AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur : ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite de ce travail expose à des poursuites pénales.

Contact : portail-publi@ut-capitole.fr

LIENS

Code la Propriété Intellectuelle – Articles L. 122-4 et L. 335-1 à L. 335-10
Loi n°92-597 du 1er juillet 1992, publiée au *Journal Officiel* du 2 juillet 1992
http://www.cfcopies.com/V2/leg/leg-droi.php
http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm

Université Fédérale
Toulouse Midi-Pyrénées

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Université Toulouse 1 Capitole (UT1 Capitole)

Présentée et soutenue par :
Ali HASSAN
le lundi 1 décembre 2014

Titre :

Modélisation des Bases de Données Multidimensionnelles :
Analyse par Fonctions d'Agrégation Multiples

École doctorale et discipline ou spécialité :
ED MITT : Image, Information, Hypermedia

Unité de recherche :
Institut de Recherche en Informatique de Toulouse (UMR 5505)

Directeur(s) de Thèse :

Gilles ZURFLUH et Olivier TESTE

Jury :

Jérôme DARMONT	Professeur, Université Lyon 2	Rapporteur
Bernard ESPINASSE	Professeur, Université Aix-Marseille	Rapporteur
Olivier TESTE	Professeur, Université Toulouse 2 Jean Jaurès	Co-directeur
Christine VERDIER	Professeur, Université Joseph Fourier Grenoble 1	Examinatrice
Gilles ZURFLUH	Professeur, Université Toulouse 1 Capitole	Directeur

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

UNIVERSITÉ TOULOUSE 1 CAPITOLE

Présentée et soutenue le *1er décembre 2014* par :

ALI HASSAN

Modélisation des Bases de Données Multidimensionnelles : Analyse par Fonctions d'Agrégation Multiples

École doctorale :

Mathématiques Informatique Télécommunications (MITT)

Unité de Recherche :

Institut de Recherche en Informatique de Toulouse (UMR 5505)

JURY

JÉRÔME DARMONT	Professeur, Université Lyon 2	Rapporteur
BERNARD ESPINASSE	Professeur, Université Aix-Marseille	Rapporteur
OLIVIER TESTE	Professeur, Université Toulouse 2 Jean Jaurès	Co-directeur
CHRISTINE VERDIER	Professeur, Université Joseph Fourier Grenoble 1	Examinatrice
GILLES ZURFLUH	Professeur, Université Toulouse 1 Capitole	Directeur

REMERCIEMENTS

Tout d'abord, je tiens à remercier très sincèrement Monsieur Claude CHRISMENT Professeur à l'université Toulouse 3 Paul Sabatier, Madame Josiane MOTHE Professeur à l'ESPE de Toulouse et Monsieur Gilles ZURFLUH Professeur à l'université Toulouse 1 Capitole, qui ont dirigé l'équipe « Système d'Informations Généralisées » (SIG) de l'IRIT pendant le déroulement de ma thèse, pour m'avoir si bien accueilli au sein de leur équipe afin que je puisse mener à bien cette thèse.

J'adresse mes remerciements les plus sincères à Monsieur Gilles ZURFLUH, pour avoir dirigé et encadré cette thèse, pour sa rigueur scientifique, pour ses critiques constructives ainsi que pour ses précieux conseils et encouragements. Je le remercie aussi pour la grande liberté d'action et la confiance qu'il m'a accordées. Merci également pour ses qualités d'écoute et sa bonne humeur et les excellentes conditions de travail qu'il m'a offertes. Qu'il soit assuré de ma profonde reconnaissance et de mon très grand respect.

Je tiens à exprimer ma gratitude à Monsieur Olivier TESTE Professeur à l'Université Toulouse 2 Jean Jaurès, qui a codirigé la thèse, pour le temps qu'il m'a consacré lors de la rédaction des articles, pour la lecture de mon mémoire et pour la préparation de la soutenance. Je n'oublierai pas Olivier TESTE pour ses remarques pertinentes et sa bonne humeur et ses encouragements tout au long de la préparation de ma thèse.

Je tiens à mentionner le plaisir et l'honneur que m'ont fait Monsieur Jérôme DARMONT Professeur à l'Université Lyon 2 et Monsieur Bernard ESPINASSE Professeur à l'Université d'Aix-Marseille, qui ont accepté d'être rapporteurs de mon travail. Je les remercie pour les remarques pertinentes qui ont permis d'améliorer mon mémoire et pour l'honneur qu'ils me font en participant au jury.

Je présente toute ma gratitude à Madame Christine VERDIER Professeur à l'Université Joseph Fourier Grenoble 1 pour tout l'intérêt qu'elle a manifesté envers mon travail et pour l'honneur qu'elle m'accorde en participant au jury de ma thèse. Je lui suis aussi très reconnaissant de m'avoir permis de faire mes premiers pas dans le monde de la recherche pendant la préparation de mon master à Grenoble.

Mes remerciements s'adressent également à Monsieur Franck RAVAT Professeur à l'Université Toulouse 1 Capitole et Monsieur Ronan TOURNIER Maître de Conférences à l'Université Toulouse 1 Capitole. Leurs remarques, leurs conseils et nos différents échanges m'ont été d'un grand intérêt.

Mes remerciements vont de même à l'ensemble du personnel de l'IRIT pour leur disponibilité, leur aide généreuse et leur gentillesse.

Je remercie infiniment toute ma famille. Je suis reconnaissant à mes parents pour les sacrifices qu'ils ont dû faire pendant mes longues années d'études et d'absence. J'espère rester

un sujet de fierté à leurs yeux. A mes frères Alaa, Adnan et Anas, merci pour leur soutien. A mes sœurs Abir, Ola et Nesrine, merci pour leurs encouragements.

Merci à ma belle famille qui a cru en moi, m'ont soutenu et m'ont défendu.

Enfin, je voudrais exprimer toute ma gratitude à ma petite famille, mon épouse Indaa et ma petite Almas. Indaa qui dans l'ombre m'a apporté durant ma thèse tout son soutien. Elle a su m'aider dans les moments difficiles et a fait preuve de beaucoup de patience, et ceci bien qu'elle n'ait pas toujours obtenu toute l'attention qu'elle était en droit d'attendre de ma part.

TABLE DES MATIÈRES

Table des Figures.

Table des Tableaux.

1. CHAPITRE I : CONTEXTE & PROBLÉMATIQUE

1.1 INTRODUCTION

Dans une entreprise, les systèmes d'information (SI) sont exploités par les décideurs pour diriger l'entreprise. Ces systèmes sont alimentés par des données des systèmes de production internes et de l'environnement externe de l'entreprise. Néanmoins, l'exploitation de ces informations réparties et hétérogènes à des fins décisionnelles nécessite leur transformation sous une forme adaptée à l'analyse [Kimball, 1996]. C'est pourquoi, afin de permettre aux décideurs d'optimiser leurs choix, les systèmes d'aide à la prise de décision ont été mis en place au sein des entreprises. Ces systèmes facilitent le stockage et le traitement synthétique de grands volumes de données consolidées [Inmon, 1996] en proposant des modèles et des techniques de manipulation des données. La plupart de ces systèmes repose sur une approche OLAP « On Line Analytical Processing » facilitant l'analyse interactive et la synthèse des données [Codd et al., 1993].

Plan du chapitre. Ce chapitre présente le contexte de la thèse et les systèmes d'aide à la prise de décision. La section 2 présente le contexte de ces systèmes, leur architecture et les outils qui les composent. La section 3 expose notre problématique en utilisant un exemple de motivation. Enfin, la section 4, présente le plan du mémoire de thèse.

1.2 LES SYSTÈMES D'AIDE À LA DÉCISION

Le rôle d'un système d'aide à la décision est de fournir aux décideurs l'infrastructure nécessaire pour avoir une vision globale des informations et des activités d'une entreprise afin de piloter l'entreprise. Ces systèmes sont de véritables interfaces entre les utilisateurs (décideurs avec différents niveaux de responsabilité) et les sources des données (Figure 1). Ils ont pour but de simplifier l'accès aux données, de masquer l'hétérogénéité des sources et de faciliter l'interrogation des données par les décideurs [Codd et al., 1993], [Kimball, 1996].

L'approche adoptée pour atteindre ces objectifs est de regrouper les données (hétérogènes et distribué), après les avoir pré-traitées, dans un unique espace de stockage et de les analyser grâce à des outils d'analyse interactive.

Définition 1. Un *système d'aide à la décision* (système décisionnel) est l'ensemble des outils informatiques (logiciels et matériels) qui accompagne un ou plusieurs décideurs pour piloter une entreprise. Après avoir transformé, regroupé et stocké les données issues du système d'information, le système décisionnel facilite leur analyse et manipulation en offrant des outils d'analyse, d'interrogation et de restitution.

1.2.1 Architecture des systèmes d'aide à la décision

Le système décisionnel consiste à extraire les données pertinentes pour la prise de décision de plusieurs sources, à les recopier dans un espace de stockage centralisé et à répondre aux requêtes des décideurs concernant un thème, un métier ou une analyse spécifique. Ainsi, trois catégories d'outils sont employées (Figure 1) :

- Les outils d'extraction, de transformation et de chargement ou ETL (Extract, Transform, Load) des données afin d'alimenter le système décisionnel.
- Les outils de stockage qui doivent permettre à la fois de gérer efficacement un grand volume de données (intégration des sources) et de définir des sous-ensembles de données adaptés à des classes de décideurs ou à des usages particuliers. Aussi, un système décisionnel repose sur une dichotomie d'espaces de stockage : l'entrepôt et les magasins de données [Ravat et al., 1999], [Ravat & Teste, 2000], [Teste, 2000].
- Les outils de restitution et d'analyse des données décisionnelles sous une forme adaptée aux décideurs.

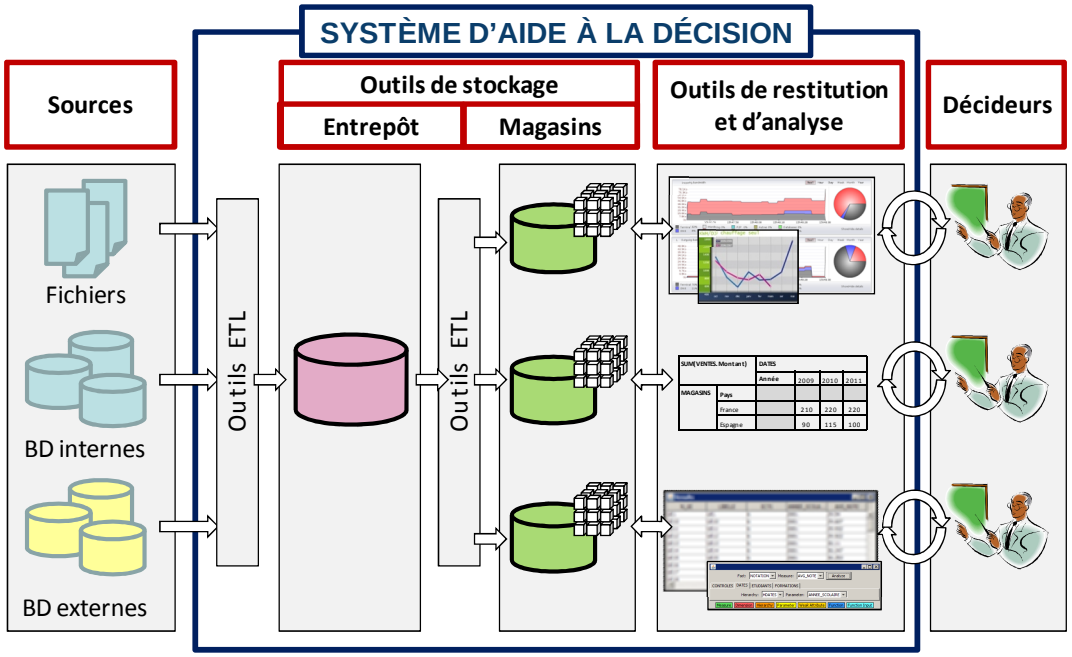


Figure 1 : Architecture d'un système d'aide à la décision

Comme l'illustre la Figure 1, l'architecture de systèmes d'aide à la décision est composée de trois niveaux :

- L'entrepôt de données : est une base de données spécifique entièrement dédiée aux décideurs [Teste, 2009]. Il est le premier niveau de stockage où les données sont regroupées, restructurées, centralisées, historialisées et matérialisées [Baril, 1999]. Le modèle de l'entrepôt doit supporter des structures complexes [Pedersen T.B. & Jensen, 1998] et supporter l'évolution des données [Pedersen T.B. & Jensen, 1999], [Yang & Widom, 2000], [Teste, 2000], [Bellahsène, 2002], [Mendelzon & Vaisman, 2003].
- Les magasins de données : constituent le deuxième niveau de stockage. Un magasin est un extrait de l'entrepôt destiné à une classe particulière de décideurs où il représente tout ou partie des données de l'entrepôt selon les besoins des décideurs.

Les données des magasins sont structurées via une modélisation multidimensionnelle et gérées par des bases de données multidimensionnelles (BDM) [Kimball, 1996]. Les magasins de données visent à supporter efficacement les processus d'interrogation et d'analyse [Ravat et al., 2001].

- La restitution et l'analyse : les données issues des magasins sont restituées aux décideurs via des outils d'analyse spécifiques : requêteurs, tableurs, outils de fouille de données et outils d'analyse OLAP (ces derniers sont les plus couramment utilisés). Ces outils permettent d'interroger et manipuler les données au travers d'interfaces graphiques.

Les modules d'extraction (ETL) permettent d'extraire les données des sources pour alimenter et rafraîchir l'entrepôt de données [Vassiliadis et al., 2002]. Les données sont souvent hétérogènes et distribuées. Elles viennent des bases de données de l'entreprise et de sources externes (sites web, emails,…). Les données sont transformées en données décisionnelles, intégrées et chargées dans l'entrepôt de données. L'évolution des données contenues dans les sources, tant au niveau des valeurs qu'au niveau de leur structure, est répercutée régulièrement dans l'entrepôt.

Dans la suite, nous détaillons les espaces de stockage, à savoir, l'entrepôt de données et les magasins.

1.2.2 Entrepôt de données

Les entrepôts de données constituent une solution adéquate pour construire un système d'aide à la décision [Widom, 1995], [Inmon, 1996]. Selon Bill Inmon, l'entrepôt de données est « une collection de données intégrées, orientées sujet, non volatiles et historisées, organisées pour le support d'un processus d'aide à la décision » [Inmon, 1996]. D'après cette définition, les données sont :

- **Intégrées** : les données proviennent de plusieurs sources. Afin de les entreposer suivant une vision homogène, il faut les nettoyer, reformater et fusionner pour réduire leur hétérogénéité.
- **Orientées sujet** : les données sont regroupées et organisées en accord avec des thèmes ou des sujets d'analyse.
- **Non volatiles** : les données de l'entrepôt sont stables, c'est-à-dire, que les données déjà intégrées sont peu modifiées mais il est toujours possible d'ajouter des nouvelles données.
- **Historisées** : l'entrepôt garde une trace de l'historique des données.

Définition 2. Un *entrepôt de données* (ED) « data warehouse » est l'espace de stockage centralisé où les données extraites des sources et pertinentes pour prendre des décisions, sont stockées, intégrées et historisées. Son organisation assure la gestion efficace des données et la conservation des évolutions.

1.2.3 Magasin de données

Les données de l'entrepôt sont restructurées dans des magasins de données.

Définition 3. Un *magasin de données* (MD) « data mart » est un extrait de l'entrepôt de données. Il regroupe les données utiles pour une classe de décideurs ou un usage particulier. Le magasin de données est organisé selon un modèle spécifique afin de faciliter l'interrogation et l'analyse décisionnelle.

Il existe classiquement deux approches pour la modélisation des magasins de données multidimensionnelles [Pedersen T.B. et al., 2001], [Torlone, 2003], [Abelló et al., 2006] :

[1] une approche reposant sur la métaphore du cube de données suivant laquelle le magasin est représentée par des cubes [Vassiliadis & Sellis, 1999].
[2] une approche dite de modélisation multidimensionnelle où le magasin est décrit par un schéma en étoile ou en constellation [Kimball, 1996].

Ces types de modélisation considèrent la donnée à analyser comme un point dans un espace à plusieurs dimensions [Gray et al., 1996], [Chaudhuri & Dayal, 1997], [Choong et al., 2003].

1.2.3.1 Métaphore du cube de données

Selon cette approche, les ***cellules*** du cube contiennent les données du sujet d'analyse. Les ***arêtes*** du cube représentent les axes d'analyse. Le nombre d'axes n'est pas limité à trois mais il peut aller jusqu'à plusieurs dizaines formant ainsi un ***hyper-cube***. Elles comportent plusieurs niveaux de granularité qui permettent d'obtenir une vision plus ou moins détaillée lors des analyses [Agrawal et al., 1995], [Li & Wang, 1996], [Gyssens et al., 1996], [Agrawal et al., 1997], [Gyssens & Lakshmanan, 1997], [Datta & Thomas, 1999].

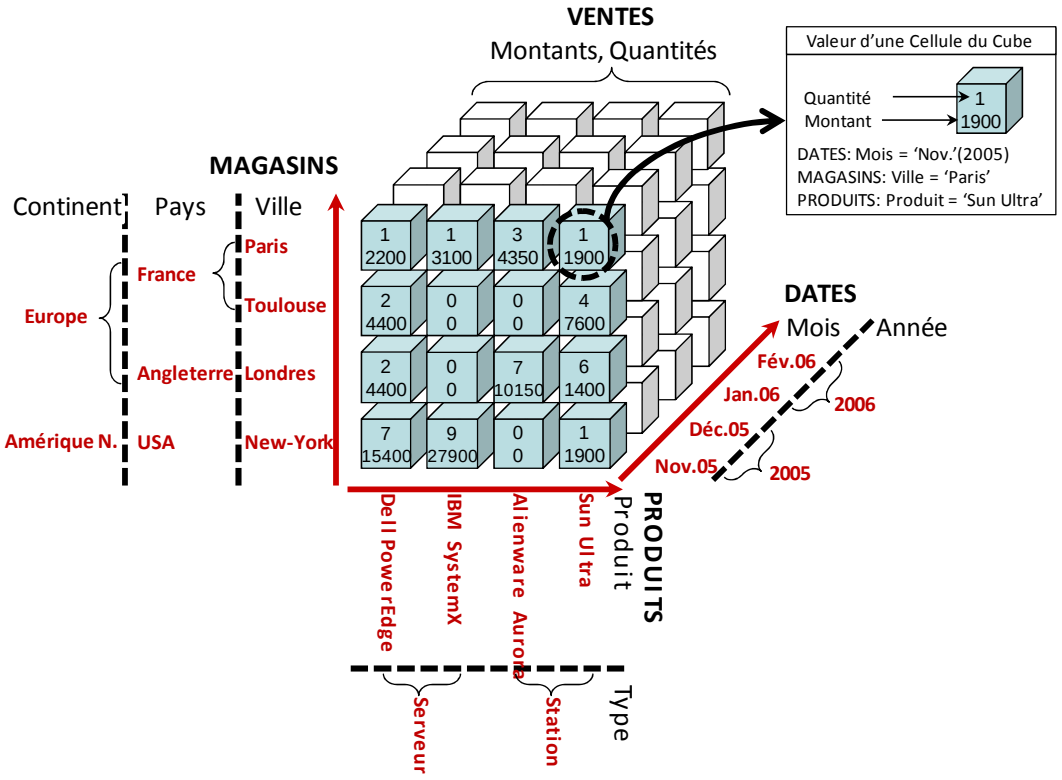


Figure 2 : Exemple de cube de données [Tournier, 2007]

La Figure 2 représente un exemple de cube tiré de [Tournier, 2007]. Dans cet exemple on analyse les 'Quantités' et les 'Montants' (indicateurs d'analyse) de 'Ventes' (sujet d'analyse) des produits informatiques en fonction de trois dimensions : les 'Magasins' où ont été effectuées les ventes, les 'Dates' correspondantes aux ventes et les 'Produits' vendus. Chaque dimension a plusieurs graduations (ville, pays, continent…).

La métaphore du cube souffre de plusieurs limites [Torlone, 2003] :

- Une séparation équivoque entre les éléments de structure et les valeurs.
- Une modélisation des axes d'analyse peu expressive notamment en raison de la difficulté à représenter l'organisation hiérarchique des données.
- Intégration d'un seul sujet d'analyse.

Afin de pallier ces limites, des structures avancées ont été définies (modélisation multidimensionnelle).

1.2.3.2 Modélisation multidimensionnelle

Cette approche permet de modéliser des sujets d'analyse appelés ***faits***, et des axes d'analyse appelés ***dimensions*** [Kimball, 1996], [Abelló et al., 2001a], [Abelló et al., 2001b]. Chaque fait comprends des indicateurs d'analyse appelés ***mesures***. Les dimensions sont composées d'attributs, appelés ***paramètres***, organisés dans des ***hiérarchies***. Les paramètres modélisent les différents niveaux de granularité sur les axes d'analyse. Ils peuvent être reliés à des attributs informationnels appelés ***attributs faibles***. Si le modèle du magasin de données est constitué d'un fait et ses dimensions associées, alors le schéma s'appellera ***schéma en étoile*** [Kimball, 1996]. Une généralisation possible du schéma en étoile est le ***schéma en constellation*** qui est constitué de plusieurs faits et de plusieurs dimensions éventuellement partagées.

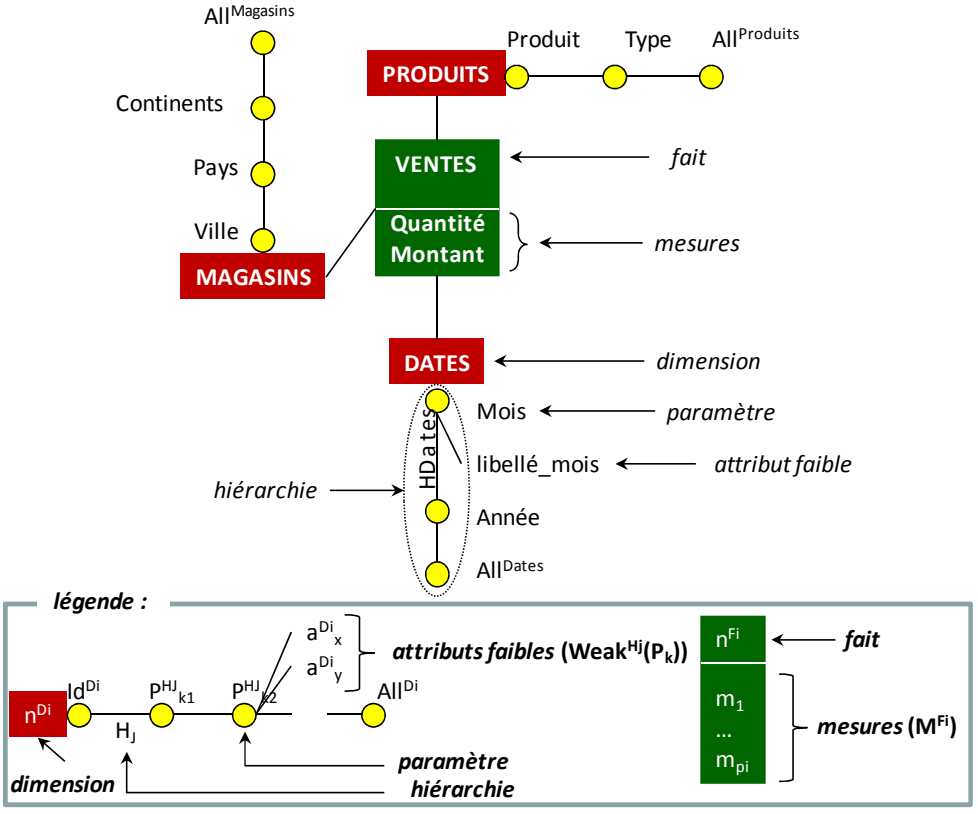


Figure 3 : Exemple de schéma en étoile

La Figure 3 représente le schéma en étoile correspondant à l'exemple précédent en utilisant les formalismes définis dans [Golfarelli et al., 1998a], [Ravat et al., 2001]. Le modèle

multidimensionnel présente mieux les différents niveaux de granularité (paramètres) et leur organisation dans les hiérarchies.

Les données des magasins sont exploitées, manipulées, interrogées et analysées par les décideurs pour prendre des décisions. Afin d'offrir cette possibilité aux décideurs, les systèmes décisionnels utilisent des outils de restitution et d'analyse décisionnelles adaptés.

1.2.4 Analyse et restitution de données

Les décideurs utilisent des outils spécifiques pour analyser les données. Les systèmes OLAP sont considérés comme les plus adaptés pour faciliter les prises de décisions [Kimball, 1996]. Les données sont interrogées au travers d'outils graphiques ou de langages textuels (MDX, OLAP SQL) [Ravat et al., 2007b], [Ravat et al., 2008]. Une requête OLAP est une requête analytique sur les données d'un magasin de données permettant d'agréger les données d'une ou de plusieurs mesures d'un fait suivant les paramètres d'une ou de plusieurs dimensions [Jerbi, 2012]. Généralement, une table bidimensionnelle est utilisée pour afficher les données résultantes de la requête [Lehner, 1998], [Tournier, 2007].

1.2.4.1 Structure de visualisation

Les décideurs manipulent et visualisent un extrait des cubes de données (généralement une tranche) au travers de tableaux à deux dimensions (ligne et colonne) [Gyssens & Lakshmanan, 1997] (Figure 4). Le choix de ces tableaux est justifié par leur simplicité d'interprétation et leur précision [Gyssens & Lakshmanan, 1997]. A partir de cette structure, appelée table multidimensionnelle (TM), le décideur peut interagir au travers d'opérations de manipulation [Ravat et al., 2007b].

Définition 4. Une *table multidimensionnelle* (TM) « multidimensional table » est une structure de visualisation d'une tranche de données à deux dimensions. Plus précisément, elle présente des instances de mesures d'un fait, en fonction des instances des paramètres des dimensions représentées en lignes et colonnes.

VENTES SOMME(Quantités)			MAGASINS		
			Continent	Europe	Amérique N.
			Pays	France	Angleterre	USA
PRODUITS	Type	Produit				
	Serveur	Dell PowerEdge		3	2	7
		IBM SystemX		1	0	9
	Station	Alienware Aurora		3	7	0
		Sun Ultra		5	6	1
DATES.Mois = 'Nov.05'						

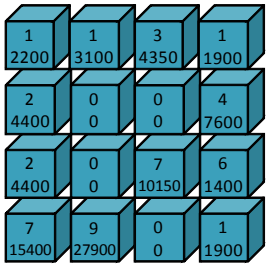


Figure 4 : Table multidimensionnelle (Visualisation d'une 'tranche' du cube)

La TM est utilisée en tant que source ou cible d'un opérateur d'une algèbre orientée décideur pour expliciter les processus d'analyse en ligne OLAP.

1.2.4.2 Opérations de manipulation OLAP

De nombreux travaux concernent la définition des opérateurs de manipulation OLAP [Gray et al., 1996], [Li & Wang, 1996], [Gyssens & Lakshmanan, 1997], [Agrawal et al., 1997], [Cabibbo & Torlone, 1997], [Cabibbo & Torlone, 1998], [Lehner, 1998], [Marcel, 1998], [Pedersen T.B. & Jensen, 1999], [Datta & Thomas, 1999], [Pedersen T.B. et al., 2001], [Abelló et al., 2003], [Franconi & Kamble, 2004], [Messaoud, 2006], [Ravat et al., 2008], [Boukraa et al., 2010], [Bimonte et al., 2012], [Golfarelli & Rizzi, 2013]. Malgré l'absence d'accord sur un

noyau d'opérateurs de manipulation OLAP, nous trouvons, dans la majorité des propositions, les trois groupes d'opérations suivants :

- **Opérations de forage.** Ces opérations consistent à modifier le niveau de granularité des données observées en navigant au moyen de la structure hiérarchique des axes d'analyses. Le forage vers le bas « DrillDown » consiste à changer le niveau de la granularité des données visualisées vers un niveau plus fin (plus détaillé). L'inverse, le forage vers le haut « RollUp » modifie le niveau de la granularité vers un niveau plus agrégé (mois détaillé).
- **Opérations de rotation.** Ces opérations permettent de réorienter l'analyse en remplaçant un axe d'analyse en cours par un autre (rotation de dimension).
- **Opérations de restriction.** Ces opérations permettent de restreindre l'ensemble des données analysées. « Slice » consiste à exprimer une restriction sur les données d'un axe d'analyse. « Dice » consiste à exprimer une restriction sur les données d'un indicateur d'analyse.

1.3 PROBLÉMATIQUE

Pour illustrer notre problématique, nous utiliserons un exemple d'analyse tiré de la météo.

1.3.1 Exemple de motivation

Dans cet exemple, les décideurs analysent les températures minimales, maximales et moyennes ainsi que les précipitations. Ces analyses sont réalisées en fonction d'informations temporelles et géographiques. La France comprend plusieurs régions et chaque région se compose de plusieurs départements. Chaque département comporte un ensemble de villes. Nous considérons qu'à proximité des villes, il y a des stations météo qui mesurent la température plusieurs fois par jour. Ces stations mesurent aussi le niveau des précipitations quotidiennement. Nous considérons également que chaque ville a un niveau administratif selon lequel elle peut être une préfecture d'un département (niveau administratif = 1), une capitale régionale (niveau administratif = 2) ou une capitale du pays (niveau administratif = 3). Si une ville n'est ni préfecture, ni capitale régionale, ni capitale du pays, alors son niveau administratif est zéro.

Les décideurs peuvent souhaiter analyser les températures selon deux méthodes différentes.

La première, **simple**, suit la même méthode utilisée par les sites de météo : pour présenter la température d'un département ou d'une région (même la température maximale et minimale), on choisit une ville représentative du département (préfecture) ou de la région (capitale régionale).

Exemple 1. La Figure 5 présente quelques captures d'écran du site de météo France[1]. Dans la Figure 5 (a), on voit la température de la région 'Midi-Pyrénées' le 23 octobre 2013 à 15h00. Nous remarquons que le site indique explicitement que la température utilisée est la température enregistrée à la station de 'Toulouse' (le rectangle rouge en pointillé) à 14h00 en sachant que 'Toulouse' est la capitale régionale de 'Midi-Pyrénées'.

[1] www.meteofrance.com.

Exemple 2. Les températures minimales (14° C) et maximales (24° C) de l'après-midi du 23 octobre 2013 dans la région 'Midi-Pyrénées' sont affichées dans la Figure 5 (b). Cependant, si l'on regarde les températures à un niveau plus détaillé (Figure 5 (c)), c'est-à-dire au niveau du département, on remarque que les températures affichées (14° et 24°) au niveau de région sont identiques aux températures du département 'Haute-Garonne' même s'il y a une température plus petite (10°) dans le département 'Ariège'. De la même manière, nous trouvons que les températures minimales et maximales du département 'Haute-Garonne' sont les températures de la ville 'Toulouse' malgré une température plus petit (9°) à 'Juzet-de-Luchon' (Figure 5 (d)). Ainsi, les températures minimales et maximales de 'Toulouse' sont utilisées pour représenter celles du département 'Haute-Garonne' et de la région 'Midi-Pyrénées', parce que 'Toulouse' est la préfecture de 'Haute-Garonne' et la capitale régionale de 'Midi-Pyrénées'.

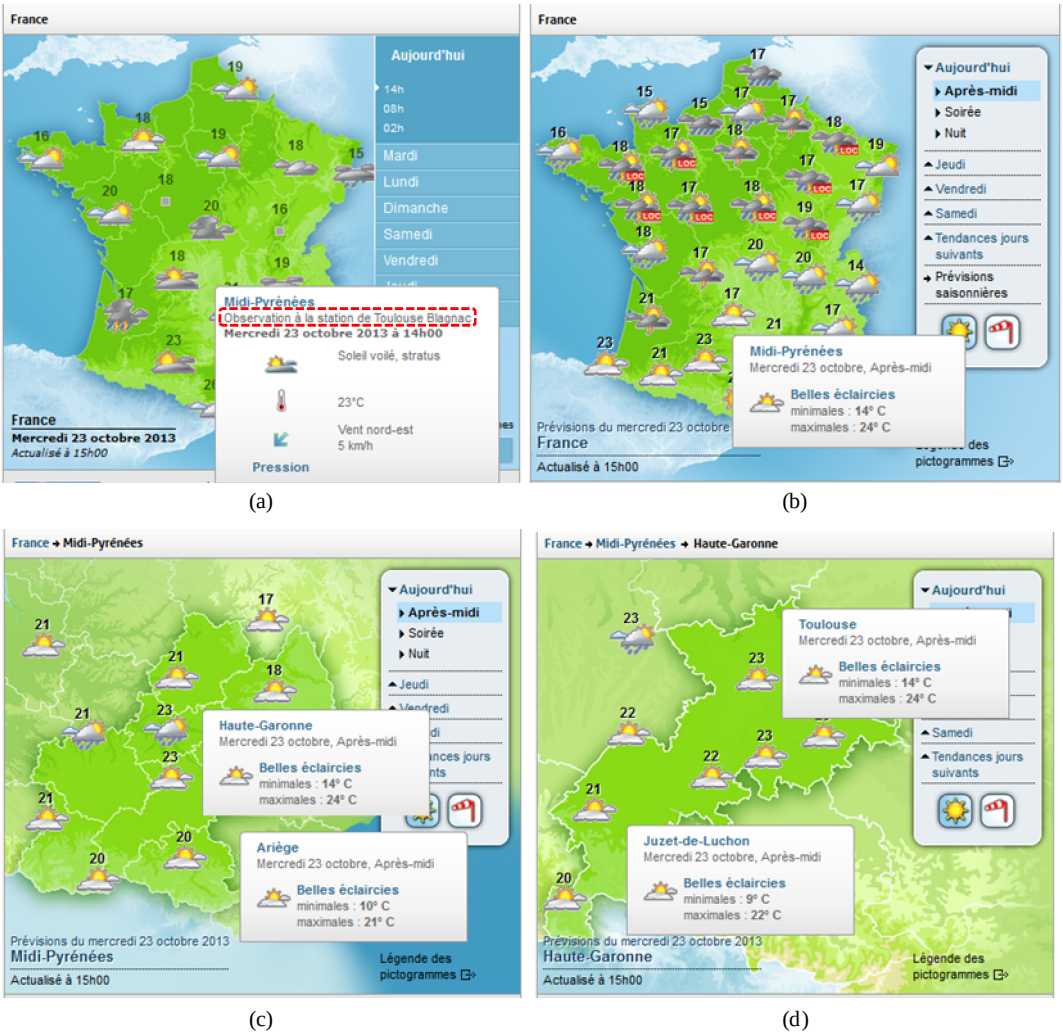


Figure 5 : Exemple de météo

La deuxième méthode, plus **scientifique**, consiste à calculer les températures d'un département, d'une région ou d'un pays en tenant compte de toutes les villes, de tous les départements ou de toutes les régions. Les météorologues ont plusieurs façons pour effectuer cette méthode d'analyse [Jones & Hulme, 1996]. Pour calculer la température dans une région [Christy et al., 2006], ils s'appuient sur un modèle de graphe orienté où les nœuds représentent

les stations météo et les arcs relient ces nœuds. Chaque arc est associé à cinq paramètres statistiques :

- La moyenne quotidienne des températures.
- Le nombre de mesures quotidiennes.
- L'écart-type des températures.
- L'autocorrélation des températures.
- L'erreur-type des températures.

D'autres météorologues préfèrent des modèles plus simples [Jones et al., 1986], [Jones & Hulme, 1996]. Le modèle le plus simple est d'interpoler les mesures des stations météo à une grille régulière. En s'appuyant sur ce modèle, le calcul de la température moyenne peut être effectué par une moyenne pondérée [Jones & Hulme, 1996] :

$$T = \sum_{i=1}^{N} w_i T_{ik} \qquad (É1)$$

Où

- (T_{ik}) est la température pendant la période (k) à la station (i) sur un échantillon de N stations.
- (W_i) est le poids de la station (i).
- (T) est la température moyenne dans le territoire considéré pendant la période (k).

Contrairement aux météorologues, qui utilisent une grille basée sur la latitude et la longitude et pour simplifier, nous utiliserons les divisions administratives. Par ailleurs, les stations sont pondérées par la superficie afin de ne pas donner aux stations avoisinantes autant de poids que celles qui sont isolées [Jones & Hulme, 1996].

Pour analyser les précipitations, nous considérons seulement la méthode scientifique. Elle est similaire à l'analyse scientifique de la température [Jones & Hulme, 1996] sauf que l'on analyse la précipitation annuelle. La précipitation annuelle est la somme des précipitations quotidiennes.

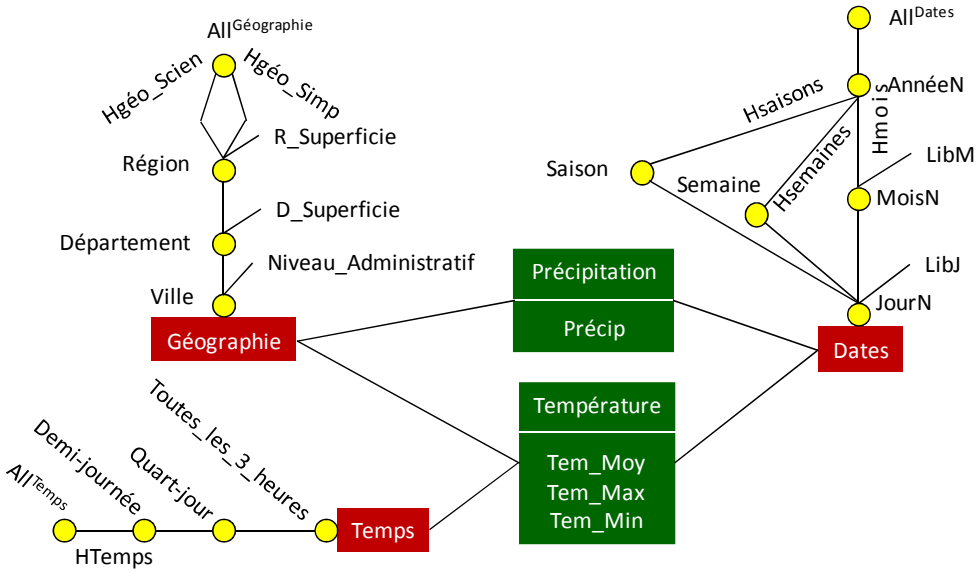


Figure 6 : Modèle multidimensionnel de l'exemple de météo

La Figure 6 décrit conceptuellement le modèle en constellation de la BDM [Golfarelli et al., 1998a], [Ravat et al., 2001] de l'exemple présenté au-dessus pour analyser la météo en France. Cette BDM vise à analyser les mesures de météo : températures moyenne 'Tem_Moy', maximale 'Tem_Max' et minimale 'Tem_Min' et la précipitation 'Précip'. Ces mesures sont organisées dans deux faits : 'Température' et 'Précipitation'. Le fait 'Précipitation' est lié à deux dimensions : 'Géographie' et 'Dates'. Le fait 'Température' est associé, en plus, à la dimension 'Temps' parce que les températures sont mesurées plusieurs fois par jour.

La dimension 'Géographie' comporte deux hiérarchies : 'Hgéo_Simp' et 'Hgéo_Scien' qui correspondent respectivement aux deux méthodes pour analyser les températures 'simple' et 'scientifique'. Sur cette dimension, chaque ville est associée à son niveau administratif 'Niveau_administratif' (ville standard (niveau zéro), préfecture, capitale régionale ou capitale du pays). Les départements et les régions sont associés à leur superficie : 'D_Superficie' et 'R_Superficie' respectivement.

La dimension 'Temps' comprend une hiérarchie 'HTemps' qui organise les niveaux des granularités horaires. La dimension 'Dates' se compose de plusieurs hiérarchies 'Hsaisons', 'Hsemaines' et 'Hmois' qui ordonnent les niveaux des granularités de la journée.

1.3.2 Illustration du problème

Ce schéma permet par exemple d'obtenir les températures moyennes (Figure 7 (a)) et les précipitations (Figure 7 (b)) par ville et jour.

Température AVG(Tem_Moy)				Géographie|Hgéo_Scien			
				Région	Midi-Pyrénées		
				Département	Haute-Garonne	Ariège	
				Ville	Toulouse	Juzet-de-Luchon	Tarascon-sur-Ariège	Porta
Dates| Hmois	Année	MoisN	JourN				
	2013	2013-10	29/10/2013	12	9	9	3
		2013-10	30/10/2013	11	9	9	2
		2013-10	31/10/2013	9	8	8	4
Géographie.Région = 'Midi-Pyrénées' and Dates.ALL= 'all' and Temps.ALL= 'all'							

(a)

Précipitation AVG(Précip)				Géographie|Hgéo_Scien			
				Région	Midi-Pyrénées		
				Département	Haute-Garonne	Ariège	
				Ville	Toulouse	Juzet-de-Luchon	Tarascon-sur-Ariège	Porta
Dates| Hmois	Année	MoisN	JourN				
	2013	2013-10	29/10/2013	3	4	4	6
		2013-10	30/10/2013	0	0	0	0
		2013-10	31/10/2013	0	0	0	0
Géographie.Région = 'Midi-Pyrénées' and Dates.ALL= 'all' and Temps.ALL= 'all'							

(b)

Figure 7 : TM visualisant les moyennes des températures et les précipitations

1.3.2.1 Problème d'agrégations multiples

Les bases de données multidimensionnelles classiques envisagent d'utiliser la même fonction d'agrégation pour calculer les valeurs d'une mesure à tous les niveaux de granularité dans l'espace multidimensionnel. Par exemple, pour calculer les températures moyennes par mois ou par année (É2), on utilise la même fonction d'agrégation 'AVG'

$$Tem\_Moy_{mois,année} = Avg(Tem\_Moy) = \frac{\sum_{i=1}^{N} Tem\_Moy}{N} \quad (É2)$$

L'utilisation de cette fonction pour agréger la température sur la dimension 'Géographie' donne un résultat incorrect car elle ne prend pas en compte les méthodes d'analyse (simple et scientifique) souhaitées par les analystes. En effet, sur la hiérarchie 'Hgéo_Simp', les températures moyennes par département, région ou par pays sont les températures de leur ville représentative (É3).

$$\{Tem\_Moy_{département,région,pays}\}_{Hgéo\_Simp} = Tem\_Moy_{villereprésentative} \quad (É3)$$

Par contre, les températures moyennes sur la hiérarchie 'Hgéo_Scien' sont calculées en utilisant la formule (É1). Afin d'adapter cette formule à notre exemple, nous considérons que :

- Le poids de chaque ville est $(\frac{1}{N})$ où N est le nombre de villes dans le département,
- Le poids de chaque département est la superficie du département divisée par la superficie de sa région,
- Le poids de chaque région est la superficie de la région divisée par la superficie du pays.

Ainsi, la température moyenne d'un département est la moyenne des températures de ses villes (É4), tandis que les températures moyennes par région ne sont pas calculées à partir des températures des villes mais sont obtenues à partir des températures des départements par une moyenne pondérée en prenant en compte les superficies des départements (É5). De la même manière, les températures moyennes du pays (niveau 'All^{Géographie}') sont obtenues à partir des températures des régions en prenant en compte la superficie de chaque région (É6).

$$\{Tem\_Moy_{département}\}_{Hgéo\_Scien} = Avg(Tem\_Moy) = \frac{\sum_{i=1}^{N} Tem\_Moy}{N} \quad (É4)$$

$$\{Tem\_Moy_{région}\}_{Hgéo\_Scien} = \frac{\sum\{Tem\_Moy_{département}\}_{Hgéo\_Scien} * D\_Superficie}{\sum D\_Superficie} \quad (É5)$$

$$\{Tem\_Moy_{pays}\}_{Hgéo\_Scien} = \frac{\sum\{Tem\_Moy_{région}\}_{Hgéo\_Scien} * R\_Superficie}{\sum R\_Superficie} \quad (É6)$$

En outre, la précipitation est agrégée sur la dimension 'Géographie' de manière similaire à l'agrégation de la température moyenne en utilisant des formules similaires aux formules (É3), (É4), (É5) et (É6). Par ailleurs, la précipitation annuelle ou mensuelle est le total des précipitations quotidiennes (É7), tandis que la précipitation générale (niveau 'All^{Dates}') est la moyenne des précipitations annuelles (É8).

$$Précip_{annuelle,mensuelle} = Sum(Précip) = \sum Précip \quad (É7)$$

$$Précip_{générale} = Avg(Précip_{annuelle}) = \frac{\sum_{i=1}^{N} Précip_{annuelle}}{N} \quad (É8)$$

1.3.2.2 Problème d'ordre entre agrégations

Un échantillon de données compatible avec notre exemple est présenté dans le Tableau 1. La température est mesurée quotidiennement, une fois en Ariège et deux fois en Haute-Garonne. Ces données simplifiées servent à analyser les températures moyennes des départements en fonction du temps selon la méthode scientifique, c'est-à-dire sur la hiérarchie 'Hgéo_Scien'.

Tableau 1 : Températures moyennes des départements par jour et temps

Région	Département	D_Superficie (km^2)	Date	Temps	Tem_Moy
Midi-Pyrénées	Haute-Garonne	6309	29/10/2013	00h00	9
Midi-Pyrénées	Haute-Garonne	6309	29/10/2013	12h00	14
Midi-Pyrénées	Ariège	4890	29/10/2013	12h00	12
Midi-Pyrénées	Haute-Garonne	6309	30/10/2013	00h00	7
Midi-Pyrénées	Haute-Garonne	6309	30/10/2013	12h00	14
Midi-Pyrénées	Ariège	4890	30/10/2013	12h00	11

Si le décideur souhaite observer, à partir des données du Tableau 1, les températures moyennes mensuelles des régions, alors deux fonctions d'agrégation doivent s'appliquer pour effectuer cette analyse :

- La moyenne pondérée de la formule (É5) pour obtenir les températures moyennes régionales à partir des températures moyennes des départements.
- La moyenne de la formule (É2) pour obtenir les températures moyennes par mois à partir des températures quotidiennes.

Si on applique la moyenne pondérée de la formule (É5) d'abord, on aura les températures moyennes régionales par jour et temps (Tableau 2). Ensuite, si on exécute la fonction moyenne de la formule (É2), on obtiendra les températures moyennes régionales mensuelles demandées (Tableau 3).

Tableau 2 : Températures moyennes des régions par jour et temps

Région	Date	Temps	Tem_Moy
Midi-Pyrénées	29/10/2013	00h00	9
Midi-Pyrénées	29/10/2013	12h00	13.1
Midi-Pyrénées	30/10/2013	00h00	7
Midi-Pyrénées	30/10/2013	12h00	12.7

Tableau 3 : Températures moyennes des régions par mois (1)

Région	Tem_Moy
Midi-Pyrénées	**10.45**

Par contre, si on exécute d'abord la fonction moyenne de la formule (É2), on obtiendra les températures moyennes départementales mensuelles (Tableau 4). Puis, si on calcule la moyenne pondérée de la formule (É5), on aura les températures moyennes régionales par mois (Tableau 5).

Tableau 4 : Températures moyennes des départements par mois

Région	Département	D_Superficie (km^2)	Tem_Moy
Midi-Pyrénées	Haute-Garonne	6309	11
Midi-Pyrénées	Ariège	4890	11.5

Tableau 5 : Températures moyennes des régions par mois (2)

Région	Tem_Moy
Midi-Pyrénées	**11.22**

Le résultat du Tableau 3 est différent de celui du Tableau 5 à cause de la non-commutativité entre les deux fonctions : la moyenne et la moyenne pondérée. Cela exige l'utilisation d'un ordre d'exécution des fonctions d'agrégation unique afin d'éviter d'avoir des résultats incorrects en raison de la non-commutativité.

1.3.3 Résumé

Les bases de données multidimensionnelles classiques, qui considèrent une seule fonction pour agréger une mesure à tous les niveaux de granularité, souffrent de plusieurs limites :

- **Variabilité de la fonction d'agrégation.** On ne peut pas faire évoluer la fonction d'agrégation suivant les axes d'analyse, les hiérarchies ou les niveaux de granularité.
  - *Changement avec les dimensionnes* : dans notre exemple, la fonction qui agrège les températures moyennes sur la dimension 'Dates' (É2) est différente des fonctions utilisées pour calculer les températures moyennes sur la dimension 'Géographie' (É3, É4, É5, É6).
  - *Changement avec les hiérarchies* : la fonction d'agrégation, qui calcule les températures moyennes sur la dimension 'Géographie', varie en fonction des hiérarchies : (É3) pour la hiérarchie 'Hgéo_Simp' et (É4, É5, É6) pour la hiérarchie 'Hgéo_Scien'.
  - *Changement avec les niveaux de granularité* : sur la dimension 'Dates', la fonction calculant la précipitation générale (É8) est différente de celle qui calcule la précipitation aux autres niveaux (É7).

  Avec l'utilisation de plusieurs fonctions d'agrégation, il faut prendre en compte l'éventuelle non-commutativité entre elles.
- **Contraintes d'agrégations.** Les BDM classiques considèrent que l'on peut toujours obtenir les valeurs d'une mesure à n'importe quel niveau d'agrégation à partir du niveau de base. Mais dans notre exemple, selon la méthode d'analyse scientifique (hiérarchie 'Hgéo_Scien'), les températures moyennes des régions et du pays ne peuvent pas être calculées directement à partir des températures des villes (niveau de base) ; il est nécessaire de calculer les températures moyennes intermédiaires par département (É4) afin de pouvoir calculer les températures moyennes régionales (É5) pour enfin obtenir les températures moyennes du pays (É6). De la même manière, la précipitation totale sur la dimension 'Dates' est obtenue à partir du calcul des précipitations annuelles ; c'est pourquoi il faut d'abord calculer la précipitation annuelle (É7) pour ensuite obtenir la précipitation totale (É8).

Notre objectif est donc de proposer un modèle multidimensionnel suffisamment expressif pour pallier ces limites et donner la possibilité d'utiliser plusieurs fonctions d'agrégation pour la même mesure en prenant en compte la non-commutativité entre elles. Ce modèle devrait de plus aborder les cas où on ne peut pas calculer une mesure à partir du niveau de base.

1.4 ORGANISATION DU MÉMOIRE

Ce mémoire s'articule de la manière suivante.

Le **chapitre 2** est consacré à l'état de l'art. Il commence par présenter les modélisations conceptuel et logique des données multidimensionnelles. Il poursuit sur les opérations d'interrogation et d'analyse OLAP. Enfin, le chapitre se termine sur un comparatif des travaux de l'intégration des fonctions d'agrégation dans les modèles multidimensionnels.

Le **chapitre 3** présente un modèle conceptuel multidimensionnel adapté pour utiliser plusieurs fonctions d'agrégation pour la même mesure et les changer en fonction des dimensions, des hiérarchies et des niveaux de granularité. Ce chapitre détaille les concepts fondamentaux pour ce modèle multifonctions avec leurs formalismes graphiques gardant la lisibilité et la compréhensibilité aux utilisateurs. Le chapitre se termine par l'introduction d'un mécanisme pour effectuer les analyses dans ce contexte multifonctions.

Le **chapitre 4** étudie les impacts de l'utilisation de plusieurs fonctions d'agrégation pour la même mesure sur le stockage de données de base (faits et dimensions) et de données d'optimisation (vues matérialisées) au niveau logique.

Le **chapitre 5** détaille la spécification de la TM et du langage d'interrogation de données multidimensionnelles adaptés au modèle multifonctions. Ce langage propose d'étendre les opérateurs OLAP afin de considérer l'utilisation de plusieurs fonctions d'agrégation pour la même mesure.

Le **chapitre 6** valide nos travaux à travers la réalisation d'un prototype. Ce prototype permet d'assister le concepteur à intégrer graphiquement les fonctions d'agrégation dans le schéma multidimensionnel. Il permet également d'assister l'analyste à spécifier graphiquement les analyses souhaitées. Ce prototype sert de plateforme expérimentale dont le chapitre détaille les expériences menées.

2. CHAPITRE II : ETAT DE L'ART

Ce chapitre présente l'état de l'art des travaux liés à ce mémoire de thèse. La première section présente les modèles conceptuels et logiques des données multidimensionnelles. Nous nous focalisons sur les opérations d'interrogation et d'analyse OLAP dans la deuxième section. En ce qui concerne la troisième section, elle détaille les travaux pour intégrer les fonctions d'agrégation dans les modèles multidimensionnels.

2.1 MODÉLISATION DES DONNÉES MULTIDIMENSIONNELLES

Plusieurs modèles ont été proposés dans la littérature pour modéliser les données multidimensionnelles. Ces modèles sont étudiés au travers des trois niveaux d'abstraction classiquement utilisés dans la conception des bases de données [Giraudin et al., 2001] :

- Niveau conceptuel : qui consiste à décrire la vision de l'utilisateur de la base de données multidimensionnelle indépendamment des choix technologiques d'implantation ;
- Niveau logique : qui consiste à décrire le modèle multidimensionnel en utilisant une technologie particulière (relationnel, objet…) ;
- Niveau physique : qui consiste à implanter le modèle logique en utilisant une plate forme spécifique (Orale, SQL Server…).

Dans la suite, nous présentons les niveaux conceptuel et logique. Le niveau physique ne se situe pas dans le cadre de nos études.

2.1.1 Niveau conceptuel

Un modèle conceptuel représente des concepts indépendamment des contraintes de plateformes d'implantation logiques ou physiques. Dans la littérature, plusieurs modèles multidimensionnels ont été proposés. Un état de l'art détaillé se trouve dans [Teste, 2009]. Les premières propositions [Agrawal et al., 1995], [Li & Wang, 1996], [Gyssens et al., 1996], [Agrawal et al., 1997], [Gyssens & Lakshmanan, 1997] se basent sur la métaphore du cube de données (*cf.* § 1.2.3.1).

Cette approche a une faiblesse dans la modélisation des constellations de faits et de dimensions partagées, pas de modélisation de l'organisation hiérarchique des données, une séparation ambiguë entre les valeurs et la structure [Torlone, 2003].

Afin de surmonter ces inconvénients, une seconde approche dite modélisation multidimensionnelle est apparue. Cette approche se base sur des concepts plus riches : faits, mesures, dimensions, hiérarchies et paramètres (*cf.* § 1.2.3.2). Nous pouvons classifier les modèles de cette approche en fonction du paradigme utilisé en trois catégories :

- Modèles basés sur le paradigme entité/association : [Sapia et al., 1998], [Tryfona et al., 1999], [Hahn et al., 2000], [Hüsemann et al., 2000], [Malinowski et al., 2006],

et [Malinowski et al., 2008]. Ces modèles représentent une dimension par un ensemble d'entités où les niveaux de granularité (paramètres) correspondent à des entités reliées par des associations. Les faits sont représentés par des associations entre les différentes dimensions.

- Modèles basés sur le paradigme objet : [Buzydlowski et al., 1998], [Trujillo et al., 1998], [Nguyen et al., 2000], [Pedersen T.B., 2000], [Pedersen T.B. et al., 2001], [Bruckner et al., 2001], [Abelló, 2002], [Trujillo et al., 2003], [Luján-Mora, 2005], [Annoni et al., 2006], [Luján-Mora et al., 2006], [Abelló et al., 2006]. Ces modèles reposent sur les concepts d'objet et de classe en utilisant les notations d'UML ou d'ODMG. Ils emploient la composition, l'agrégation et l'héritage pour représenter les relations entre les concepts dimensionnels.
- Modèles multidimensionnels spécifiques : [Golfarelli et al., 1998a], [Cabibbo & Torlone, 1998], [Cabibbo & Torlone, 2000], [Ravat et al., 2001], [Schneider, 2003], [Tournier, 2007], [Schneider, 2008]. Ces modèles séparent les éléments structurels (faits, dimensions et hiérarchies (*cf.* § 1.2.3.2)) des valeurs. Leurs avantages sont la simplicité des notations et la proximité de la vision des décideurs. Par contre, leurs notations sont spécialisées et ne s'appuient pas sur des notations standards [Torlone, 2003].

Les formalismes et les concepts utilisés dans la modélisation multidimensionnelle conceptuelle souffrent de l'absence d'un consensus standardisé [Rizzi et al., 2006].

2.1.2 Niveau logique

Plusieurs modèles sont utilisés pour implanter les schémas multidimensionnels au niveau logique :

- **Modèle R-OLAP (Relational - On Line Analytical Processing)** : ce modèle est le plus courant. Il se base sur l'implantation des schémas multidimensionnels dans un environnement relationnel [Kimball, 1996], [Dinter et al., 1998], [Mangisengi & Tjoa, 1998]. Ce modèle a plusieurs avantages : la réutilisation des mécanismes de gestion des données éprouvés, la capacité à gérer des volumes de données importants. Il transforme chaque fait et chaque dimension du modèle multidimensionnel conceptuel au niveau logique en une table relationnelle [Kimball, 1996].
- **Modèle M-OLAP (Multidimensional - On Line Analytical Processing)** : ce modèle permet de stocker les données sous une forme nativement multidimensionnelle (dans des cubes de données, des matrices ou des vecteurs à n dimensions) [Agrawal et al., 1997], [Vassiliadis, 1998], [Dinter et al., 1998]. Ce modèle se caractérise par des performances élevées grâce à la réduction des temps d'accès aux données et d'exécution des requêtes d'analyse. [Kimball, 1996]. Par contre, ce modèle nécessite de changer complètement et spécifiquement le système de gestion de données. En outre, il souffre d'une capacité limitée à gérer des volumes importants de données [Teste, 2009].
- **Modèle H-OLAP (Hybrid - On Line Analytical Processing)** : ce modèle réunit les avantages des deux modèles M-OLAP et R-OLAP en diminuant leurs inconvénients. Il est utilisé surtout dans les outils commerciaux (Oracle Application Server, Microsoft Analysis Services). Il consiste à stocker les données détaillées dans des tables relationnelles (comme le modèle R-OLAP), tandis qu'il stocke les données agrégées sous une forme multidimensionnelle (comme le modèle M-OLAP).

Par ailleurs, les systèmes d'aide à la décision impliquent des requêtes complexes sur très grandes bases de données. Étant donné que le temps de réponse doit être quelques minutes

au maximum [Harinarayan et al., 1996], l'optimisation des requêtes est essentielle. Cette optimisation consiste à pré-calculer et stocker (matérialiser) les agrégations nécessaires aux décideurs lors de leurs analyses. Les pré-agrégations sont modélisées par un treillis de pré-agrégats [Harinarayan et al., 1996] où chaque nœud représente un pré-agrégat et chaque arc représente le chemin des calculs d'agrégation.

De nombreux travaux concernent la matérialisation des pré-agrégats [Harinarayan et al., 1996], [Yang et al., 1997], [Baralis et al., 1997], [Gupta, 1997], [Theodoratos & Sellis, 1997], [Huyn, 1997], [Zhuge et al., 1997], [Zhuge et al., 1998], [Han et al., 1998], [Gupta & Mumick, 1999], [Kotidis & Roussopoulos, 1999], [Agrawal et al., 2000], [Yang & Widom, 2000], [Labio et al., 2000], [Liang et al., 2001], [Lee & Hammer, 2001], [Kotidis & Roussopoulos, 2001], [Kalnisa et al., 2002], [Yu et al., 2003], [Li et al., 2005], [Yu et al., 2005], [Lawrence & Rau-Chaplin, 2006], [Nandi et al., 2011], [Hanusse et al., 2011], [Kerkad et al., 2013], [Boukorca et al., 2013]. Ces travaux se rapportent à la sélection et la maintenance des données matérialisées.

Nous constatons que ces travaux ne considèrent pas l'utilisation de plusieurs fonctions d'agrégation pour agréger la même mesure. Il est nécessaire ainsi de réétudier le treillis d'optimisation dans un contexte multifonctions.

2.2 MANIPULATION OLAP

La manipulation OLAP consiste à aider les décideurs à analyser les données entreposées. Les premières propositions de la manipulation OLAP s'appuyaient sur les opérateurs de l'algèbre relationnelle [Gray et al., 1996], [Li & Wang, 1996], [Agrawal et al., 1997], [Gyssens & Lakshmanan, 1997],[Datta & Thomas, 1999]. En raison de l'absence de convenance de l'algèbre relationnelle dans le contexte de la manipulation multidimensionnelle, plusieurs travaux ont proposé des opérateurs afin de spécifier et manipuler un cube [Cabibbo & Torlone, 1997], [Cabibbo & Torlone, 1998], [Teste, 2000], [Pedersen T.B. et al., 2001], [Abelló et al., 2003], [Franconi & Kamble, 2004], [Ravat et al., 2007b], [Ravat et al., 2008].

Malgré l'absence d'un consensus sur les opérateurs OLAP [Ravat et al., 2007b] et l'existence de plusieurs travaux sur la visualisation OLAP [Lee & Ong, 1995], [Sifer, 2003], [Choong et al., 2003], [Maniatis et al., 2005], [Techapichetvanich & Datta, 2005], [Hanrahan et al., 2007], [Cuzzocrea et al., 2007], [Cuzzocrea & Mansmann, 2009], [Ordonez et al., 2011], la plupart des travaux s'appliquent sur un cube ou une table multidimensionnelle.

2.2.1 Table multidimensionnelle

Une table multidimensionnelle est une structure utilisée pour afficher des données d'un fait et deux de ses dimensions associées, une dimension par ligne et une dimension par colonne [Gyssens & Lakshmanan, 1997] (*cf.* § 1.2.4.1). Grâce à sa simplicité, la table multidimensionnelle est la pierre angulaire de la manipulation OLAP où tous les opérateurs OLAP s'appliquent sur une table multidimensionnelle.

2.2.2 Opérateurs OLAP

Les opérateurs OLAP permettent aux décideurs d'analyser les données, où les analyses sont réalisées par des séquences d'opérateurs qui manipulent les composants de la table multidimensionnelle. Ainsi, chaque opérateur prend en entrée une table multidimensionnelle source produit en sortie une table multidimensionnelle résultat. Donc, la fermeture des opérateurs est assurée.

De nombreux travaux définissent des opérateurs de manipulation OLAP [Gray et al., 1996], [Li & Wang, 1996], [Agrawal et al., 1997], [Gyssens & Lakshmanan, 1997], [Cabibbo & Torlone, 1997], [Thomas & Datta, 1997], [Cabibbo & Torlone, 1998], [Lehner, 1998], [Datta &

Thomas, 1999], [Vassiliadis, 2000], [Pedersen T.B., 2000], [Pedersen T.B. et al., 2001], [Abelló et al., 2003], [Franconi & Kamble, 2004], [Abelló et al., 2006], [Ravat et al., 2008], [Boukraa et al., 2010]. Les opérateurs OLAP définis dans ces travaux peuvent être classés selon leurs fonctionnalités en huit groupes : opérateurs de forage, de sélection, de rotation, de modifications du sujet d'analyse, de modifications d'une dimension, d'ordonnancements, d'agrégation et opérateurs binaires.

Opérateurs de forage : les forages vers le haut (**ROLLUP**) et vers le bas (**DRILLDOWN**) consistent à naviguer sur une dimension dans une TM afin de changer le niveau de granularité des données visualisées vers un niveau moins ou plus détaillé respectivement.

Opérateurs de rotation : ces opérateurs permettent de réorienter l'analyse par échanger un axe d'analyse contre un autre dans une TM (**DROTATE**). Ils peuvent être utilisés pour remplacer une hiérarchie par une autre appartenant à la même dimension (**HROTATE**). Ils permettent également de permuter un fait par un autre (**FROTATE**) en gardant les deux dimensions visualisées dans la TM initiale.

Opérateurs de sélection : ces opérateurs suppriment les données qui ne satisfont pas une condition (**SELECT**). Cette condition peut s'exprimer sur les valeurs d'un paramètre (Slice), ainsi que sur les valeurs de mesure (Dice). Ils permettent également d'annuler toutes ces sélections de fait et des dimensions (**UNSELECT**).

Opérateurs de modifications du sujet d'analyse : ces opérateurs permettent la modification de l'ensemble des mesures analysées par ajouter (**ADDM**) et supprimer (**DELM**) des mesures.

Opérateurs de modifications d'une dimension : ces opérateurs consistent à combiner les paramètres des dimensions avec les mesures. L'opérateur (**PUSH)** convertit les paramètres en mesures. L'opérateur (**PULL)** est l'inverse, il convertit les mesures en paramètres. Ces opérateurs permettent également d'insérer un paramètre d'une dimension non-affichée dans une dimension affichée (**NEST**) afin d'analyser les données dans la TM selon plusieurs dimension.

Opérateurs d'ordonnancements : ces opérateurs permettent de permuter deux valeurs d'un paramètre (**SWITCH**) ou bien ordonner toutes ses valeurs dans un ordre croissant ou décroissant (**ORDER**) avec répercussion sur les valeurs des paramètres de granularité inférieure.

Opérateurs d'agrégation : ces opérateurs permettent d'ajouter dans la TM une ligne ou une colonne agrégeant ses valeurs (**AGGREGATE**). Ils permettent également d'annuler cette agrégation en supprimant la ligne ou la colonne ajoutée (**UNAGGREGATE**).

Opérateurs binaires : avec ces opérateurs nous pouvons effectuer une union, une intersection, une opération de différence ou une jointure de deux TMs. Il convient d'indiquer que ces opérateurs nécessitent une forte compatibilité des deux TMs source.

Autres opérateurs : nous trouvons dans la littérature des opérateurs additionnels, nous présentons certains d'entre eux :

- **BLEND** : [Hubert & Teste, 2009] propose cet opérateur afin de permettre de regrouper lors de l'analyse les valeurs des mesures entre deux niveaux de granularité dans un seul nouveau niveau sans changer les stockage physique de données ou le modèle multidimensionnel ;
- **SHRUNK** : cet opérateur proposé par [Golfarelli & Rizzi, 2013] est applicable à un cube de données résultant d'une requête OLAP lancée. Il vise à équilibrer la précision avec la taille des données visualisées dans une TM. Il fusionne des

tranches de données similaires et il les remplace par une seule tranche représentative, jusqu'à ce que le résultat soit inférieur à un seuil donné ;

- **Opérateurs orientés sur les cellules** : ils visent à modifier les valeurs des cellules de la TM sans changer ses composants [Lehner, 1998]. Par exemple, calculer les valeurs absolues ou bien multiplier les cellules de la TM par (-1) ;
- **Opérateurs spécifique** : certains opérateurs spécifiques correspondants au contexte particulier dans lequel ils s'appliquent ont été proposés dans la littérature. Comme les opérateurs (**SPACIAL DRILL**) et (**ZOOM**) proposés par [Bimonte et al., 2012] pour les entrepôts de données spatiales. SPACIAL DRILL permet de naviguer sur une hiérarchie spatiale. ZOOM permet de modifier l'échelle ou le niveau de détail de carte de dimensions spatiales.

2.2.3 Bilan

Tableau 6 compare les travaux de recherche sur les opérateurs OLAP. Dans ce tableau, nous remarquons que [Gray et al., 1996] n'aborde pas tous les aspects d'analyse OLAP. Il propose seulement un opérateur d'agrégation afin d'afficher les données agrégées dans une nouvelle ligne ou colonne dans la TM. En ce qui concerne les autres travaux de recherche, nous trouvons qu'ils supportent tous l'opérateur ROLLUP sauf [Franconi & Kamble, 2004] qui propose au lieu de cet opérateur de définir, dès le début de la création de l'entrepôt de données, toutes les agrégations nécessaires et les stocker dans des faits agrégés.

Tableau 6 : Synthèse des travaux sur les opérateurs OLAP

Opérateurs / Travaux de recherche	Forage		Rotation			Sélection		Modification					Ordonnancement	AGGREGATE	Binaires			
								Fait		Dimension								
	ROLLUP	DRILLDOWN	DROTATE	HROTATE	FROTATE	DICE	SLICE	ADDM	DELM	PUSH	PULL	NEST			Union	Intersect	Difference	join
[Gray et al., 1996]														✓				
[Li & Wang, 1996]	✓	✓					✓					✓			✓			✓
[Agrawal et al., 1997]	✓	✓					✓	✓	✓	✓	✓				✓	✓	✓	✓
[Gyssens & Lakshmanan, 1997]	✓					✓	✓	✓	✓	✓	✓		✓		✓	✓	✓	✓
[Cabibbo & Torlone, 1997], [Cabibbo & Torlone, 1998]	✓					✓	✓											✓
[Lehner, 1998]	✓	✓					✓					✓						✓
[Thomas & Datta, 1997], [Datta & Thomas, 1999]	✓	✓				✓	✓			✓	✓				✓	✓	✓	✓
[Vassiliadis, 2000]	✓	✓					✓											
[Pedersen T.B., 2000], [Pedersen T.B. et al., 2001]	✓	✓					✓								✓		✓	✓
[Abelló et al., 2003], [Abelló et al., 2006]	✓	✓	✓		✓	✓									✓			
[Franconi & Kamble, 2004]							✓	✓							✓	✓	✓	✓
[Ravat et al., 2008]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
[Lenz & Thalheim, 2009]	✓	✓	✓			✓	✓								✓			✓
[Boukraa et al., 2010]	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓			✓	✓	✓	
[Pardillo et al., 2010]	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓				✓	✓	✓	

Les opérateurs DRILDOWN et Union sont supportés par la majorité des travaux et l'opérateur SLICE par quasiment tous les travaux. Nous remarquons également que les opérateurs d'agrégation et de rotation sont les moins supportés dans les travaux de recherche. Aucune proposition ne supporte tous les opérateurs de manipulation OLAP.

Dans le cadre de l'analyse multifonctions, il est nécessaire de reformuler la table multidimensionnelle et ces opérateurs afin de supporter l'utilisation de plusieurs fonctions d'agrégation pour agréger la même mesure dans l'espace multidimensionnelle.

2.3 FONCTIONS D'AGRÉGATION

La notion d'agrégation multidimensionnelle est apparue avec les bases de données statistiques[2] [Özsoyoglu et al., 1985], [Özsoyoglu et al., 1987]. La discision autour cette notion a repris avec les premiers modèles de cube de données [Li & Wang, 1996], [Agrawal et al., 1997], [Gyssens & Lakshmanan, 1997]. Par la suite, avec l'utilisation des hiérarchies organisant les données des axes d'analyse, de nouvelles propositions ont émergé [Jagadish et al., 1999], [Pourrabas & Rafanelli, 2000], [Pourrabas & Rafanelli, 2003].

Les fonctionnalités d'analyse OLAP sont basées sur les fonctions d'agrégation. Ainsi, les fonctions d'agrégation appliquées à un cube de données doivent être bien définies afin d'obtenir des résultats valides. Ceux-ci doivent tenir compte les dépendances entre les dimensions, et doivent obéir aux lois de l'associativité et de la commutativité [Lenz & Thalheim, 2006].

2.3.1 Classification des fonctions d'agrégation

Les fonctions d'agréation ont été classifiées dans la littérature selon des points de vue différents.

2.3.1.1 Du point de vue du mécanisme d'agrégation

Selon cette classification, les fonctions d'agrégations appartiennent à trois catégories différentes [Gray et al., 1996] :

- La première correspond aux fonctions **distributives** qui calculent les valeurs agrégées à un niveau de granularité à partir des valeurs déjà agrégées au niveau de granularité directement inférieur. Formellement, une fonction d'agrégation *f* est distributive s'il existe une fonction *g* où :

  $$f(x_1 \cup x_2 \cup \ldots \cup x_n) = g(f(x_1), f(x_2), \ldots, f(x_n))$$

  Les fonctions COUNT, MIN, MAX et SUM sont distributives où (*g* = SUM) pour la fonction COUNT et (*f* = *g*) pour les autres. Par exemple, la somme (*f* = SUM) d'un montant par année peut se calculer à partir de la somme (*g* = SUM) des montants par semestre ;
- La deuxième correspond aux fonctions **algébriques** qui calculent les valeurs agrégées à partir de résultats intermédiaires stockés. Formellement, une fonction d'agrégation *f* est algébrique s'il existe deux fonctions *g* et *h* où :

[2] Plus de détailles sur les bases de données statistiques se trouvent dans [Rafanelli & Ricci, 1983], [Lenz & Shoshani, 1997], [Shoshani, 2003].

$$f(x_1 \cup x_2 \cup \ldots \cup x_n) = g(h(x_1), h(x_2), \ldots, h(x_n))$$

Les fonctions COVARIANCE, AVG et ÉCART-TYPE sont algébriques. Par exemple, pour la moyenne (f = AVG), la fonction h enregistre les sommes (SUM) et les nombres (COUNT) des occurrences des sous-ensembles. La fonction g calcule d'abord les totaux des sommes et des nombres, puis elle fait la division pour produire la moyenne globale ;

- La troisième correspond aux fonctions **holistiques** qui ne peuvent pas être calculées à partir de résultats intermédiaires. Dans ce cas, il faut calculer les valeurs agrégées à partir des valeurs de base correspondant au niveau de granularité le plus bas. MEDIAN, MAIN et RANK sont des exemples de fonctions holistiques.

2.3.1.2 Du point de vue de l'additivité

Cette classification est une simplification de la classification précédente. Les fonctions d'agrégation font partie de deux groupes [Abelló et al., 2006], [Boulil et al., 2011] :

- Le premier contient les fonctions «**Transitives**» qui garantissent l'additivité «Summarizability» ;
- Le deuxième contient les fonctions «**Non-Transitives**» qui impliquent que l'agrégation doit toujours se calculer à partir du niveau de base.

2.3.1.3 Du point de vue de la mesure (données)

[Rafanelli & Ricci, 1983], [Lenz & Shoshani, 1997], [Golfarelli et al., 1998a], [Lehner, 1998], [Pedersen T.B. et al., 2001] distinguent trois types de fonctions d'agrégation :

- Le premier type est applicable aux données :
  - **Additives** : cela signifie que la fonction somme peut être utilisée pour agréger les mesures sur toutes les dimensions. Par exemple, le montant de ventes,
  - **Semi-additives** : cela signifie que la mesure n'est pas additive sur une ou plusieurs dimensions. Par exemple, toutes les analyses qui mesurent un niveau tel qu'un niveau de stock qui n'est pas additif sur la dimension (dates), mais il est additif sur les dimensions (produits) et (magasins).

  Les fonctions SUM, COUNT, AVG, MIN et MAX sont de ce type ;
- Le deuxième type est applicable aux données qui **peuvent être utilisées pour les calculs de moyenne**. Par exemple les températures, puisque l'addition de deux températures n'a pas de sens. Les fonctions COUNT, AVG, MIN et MAX sont classifiées dans ce type ;
- Le troisième type est applicable aux données **constantes**, c'est-à-dire qu'elles ne peuvent être que dénombrées. La fonction COUNT appartient à ce type.

2.3.1.4 Du point de vue de l'utilisation

Les fonctions sont réparties dans deux groupes [Tournier, 2007] :

- **Les fonctions classiques** : qui sont supportées par les SGBD relationnels. Elles prennent en entrée un ensemble des valeurs numériques et donnent en sortie une seule valeur numérique. Les fonctions SUM, COUNT, MIN, MAX, AVG sont parmi les fonctions le plus connues de ce groupe. Ce groupe comprend également des fonctions statistiques, par exemple, Covariance, Écart-type, Median ;
- **Les fonctions spécifiques (avancées) :** il s'agit de fonctions d'agrégation utilisées dans des contextes spécifiques :
  - *Entrepôt de données spatiales* où les données géographiques sont présentées sous forme de points, lignes et polygones. Des fonctions spécifiques adaptées

sont nécessaires pour agréger ces éléments. Par exemple, trouver le barycentre de plusieurs points ou calculer la surface moyenne de plusieurs polygones [Camossi et al., 2006], [Silva et al., 2008], [Bimonte et al., 2012]

- *Entrepôt de document* où les données à analyser sont principalement des textes et des mots. Par exemple, les fonctions AVG_KW [Ravat et al., 2007a], TOP_KEYWORD [Park et al., 2005] qui sont conçues pour agréger des ensembles de mots-clés ;
- *La fouille de données*, par exemple la fonction SKYLINE [Börzsönyi et al., 2001] qui cherche une solution maximale ou minimale pour un problème à au moins deux variables (Vector Maximisation Problem) [Kung et al., 1975].

2.3.1.5 Du point de vue de la méthode de calcul

Les fonctions d'agrégation sont de deux types [Bertinoa et al., 2003] :

- Le premier « **Sélectif** » : une fonction de ce type retourne en sortie une valeur sélectionnée parmi ses entrées. Formellement, une fonction d'agrégation f est sélective si :

$$f\{x_1, x_2, \ldots, x_n\} = x_i \mid i \in \{1, 2, \ldots, n\}$$

  Les fonctions FIRST et LAST sont de ce type ;
- Le deuxième « **Agrégat** » : les résultats des fonctions de ce type sont calculés en agrégeant au moins deux valeurs d'entrée $\{x_1, x_2, \ldots, x_n\}$. Par exemple, les fonctions SUM, AVG.

2.3.1.6 Du point de vue de la complexité de calcul

[Lenz & Thalheim, 2005] étend la notion de fonction d'agrégation pour inclure des opérations d'analyse qui demandent des agrégations complexes. Ainsi, les fonctions d'agrégation selon cette classification sont de deux catégories :

- La première **simple** où les fonctions réalisent des agrégations élémentaires comme calculer la somme totale (SUM) ou trouver la valeur maximale (MAX) ou minimale (MIN) ;
- La deuxième **complexe** où les fonctions sont utilisées pour effectuer des agrégations qui relient des sous-ensembles des données à d'autres sous-ensembles ou sur-ensembles. Par exemple, la fonction d'agrégation qui calcule le pourcentage des ventes de chaque client contribue au chiffre d'affaires total.

2.3.1.7 Bilan

Malgré l'existence de six points du vue pour classifier les fonctions d'agrégation [Gray et al., 1996], [Lenz & Shoshani, 1997], [Bertinoa et al., 2003], [Lenz & Thalheim, 2005], [Abelló et al., 2006], [Tournier, 2007], nous pouvons regrouper ces points du vue dans deux catégories différentes :

- *Classification selon le calcul* des fonctions d'agrégation (le mécanisme d'agrégation [Gray et al., 1996], l'additivité [Abelló et al., 2006] et la méthode de calcul [Bertinoa et al., 2003]).
- *Classification selon le contexte* correspondant à l'emploi des fonctions (les données [Lenz & Shoshani, 1997], le contexte de l'utilisation [Tournier, 2007] et la complexité de calcul [Lenz & Thalheim, 2005]).

Toutes ces classifications existantes estiment que l'on peut calculer l'agrégation d'une mesure pour tous les niveaux de granularité possibles à partir du niveau de base. Notre but est

d'ajouter le moyen de traiter le cas contraire (quand il n'est pas possible d'agréger la mesure à partir du niveau de base) en utilisant des *contraintes d'agrégation*.

2.3.2 Fonctions d'agrégation dans la modélisation multidimensionnelle

Plusieurs propositions existantes [Sapia et al., 1998], [Tryfona et al., 1999], [Nguyen et al., 2000], [Tsois et al., 2001], [Ravat et al., 2001], [Luján-Mora et al., 2006], [Chen et al., 2006], [Ravat et al., 2008], [Midouni et al., 2009], [Oliveira et al., 2011] considèrent qu'une mesure est associée à une fonction d'agrégation qui sera utilisée à tous les niveaux d'agrégation modélisés (paramètres). Cette fonction calcule la même agrégation pour toutes les combinaisons de tous les paramètres modélisés. Dans la suite, nous désignerons ce type de fonction par le terme *fonction générale*.

D'autres travaux traitent l'agrégation des mesures dans l'espace multidimensionnel différemment : les fonctions d'agrégation peuvent être intégrées dans la modélisation multidimensionnelle à plusieurs niveaux.

Nous présentons dans la suite, ces différentes approches ainsi que l'agrégation dans les outils commerciaux. Nous analysons ces propositions à travers l'étude de plusieurs points liés à notre problématique :

- Comment les propositions réalisent **l'intégration des fonctions d'agrégation** dans la modélisation multidimensionnelle ;
- **Multifonctions** : si les travaux autorisent l'utilisation de plusieurs fonctions d'agrégation pour la même mesure ;
- **Fonctions dimensionnelles** : si les propositions offrent la possibilité de changer la fonction d'agrégation avec les dimensions ;
- **Fonctions hiérarchiques** : si nous pouvons changer les fonctions d'agrégation selon des hiérarchies ;
- **Fonctions différenciées** : si les propositions permettent d'utiliser une fonction d'agrégation spécifique pour chaque niveau de granularité ;
- Si les travaux traitent le cas des fonctions **commutatives** et **non-commutatives** ;
- Si les propositions abordent le cas des **agrégations contraintes**, c'est-à-dire lorsque la mesure doit être calculée à partir d'un niveau différent du niveau de base.

En outre, afin de présenter les limites de ces propositions, nous allons essayer de les appliquer à notre exemple de météo (exemple de motivation *cf.* § 1.3.1).

2.3.2.1 Pré-agrégations & agrégations au cours de l'interrogation

Modèles du cube de données

La plupart des modèles du cube [Li & Wang, 1996], [Gyssens & Lakshmanan, 1997], [Agrawal et al., 1997], [Thomas & Datta, 1997], [Datta & Thomas, 1999], [Vassiliadis & Skiadopoulos, 2000], [Pardillo et al., 2010] ne précisent pas les fonctions d'agrégation pour agréger les mesures dans l'espace multidimensionnel. Seul [Lehner, 1998] précise le type de fonctions d'agrégation applicable selon le type de mesures (additives, pouvant être utilisées pour les calculs de moyenne ou constantes (*cf.* § 2.3.1.3)). Tous les travaux précédents laissent la possibilité d'utiliser, pour chaque mesure, plusieurs fonctions d'agrégation au cours de l'interrogation. Cela donne une grande flexibilité, mais laisse la possibilité de commettre des erreurs en utilisant des fonctions inappropriées.

Dans la suite, nous prenons [Vassiliadis & Skiadopoulos, 2000] comme exemple de ces travaux afin de détailler leurs limites ; en effet les auteurs précisent les fonctions d'agrégation utilisées pour calculer chaque cube de données. Ainsi, au cours de la navigation 'nav' (forage vers le haut RollUp), nous pouvons définir des nouveaux cubes calculés à partir d'autre cubes

en utilisant différentes fonctions d'agrégation. Par exemple, le cube qui calcule les mesures 'Tem_Moy', 'Tem_Max' et 'Tem_Min' du fait 'Température' au niveau le plus bas 'JourN', 'Ville' et 'Toutes_les_3_heures' est le suivant :

$Cube_0$ = ([JourN, Ville, Toutes_les_3_heures, Tem_Moy, Tem_Max, Tem_Min], AVG(Tem_Moy), MAX(Tem_Moy), MIN(Tem_Moy))

Les températures moyennes, maximales et minimales des villes par mois et toutes les trois heures peuvent être calculées en se basant sur le $cube_0$ selon (É2) :

$Cube_1$ = nav^3($Cube_0$, [MoisN, Ville, Toutes_les_3_heures, Tem_Moy_mois, Tem_Max_mois, Tem_Min_mois], AVG(Tem_Moy), MAX(Tem_Max), MIN(Tem_Min))

L'analyse de ces mesures selon l'agrégation simple (*cf.* § 1.3.1) peut être effectuée en s'appuyant sur le même $cube_0$. Par exemple, les températures des départements par jour et toute les trois heures selon (É3) :

$Cube_2$ = nav($Cube_0$, [JourN, Département, Toutes_les_3_heures, Tem_Moy_départe, Tem_Max_départe, Tem_Min_départe], SELECT_CENTER(Niveau_Administratif, Tem_Moy), SELECT_CENTER(Niveau_Administratif, Tem_Max), SELECT_CENTER(Niveau_Administratif, Tem_Min))

La fonction SELECT_CENTER(I, M) prend deux paramètres numériques. Elle rend la valeur M qui correspond au Max(I). Par exemple, si on applique SELECT_CENTER(Niveau_Administratif, Tem_Moy) au niveau région ou département, elle rend les températures de la capitale régionale ou de la préfecture (ville ayant le niveau administratif maximal).

En outre, le cube qui réalise l'analyse des températures quotidiennes des départements toutes les trois heures selon l'agrégation scientifique (*cf.* § 1.3.1) est calculé selon (É4) à partir de $Cube_0$:

$Cube_3$ = nav($Cube_0$, [JourN, Département, Toutes_les_3_heures, Tem_Moy_départe, Tem_Max_départe, Tem_Min_départe], AVG(Tem_Moy), MAX(Tem_Max), MIN(Tem_Min))

De la même façon, les températures des régions par jour et toutes les trois heures selon l'agrégation scientifique (É5) sont calculées à partir de $Cube_3$ mais en utilisant une autre fonction d'agrégation :

$Cube_4$ = nav($Cube_3$, [JourN, Région, Toutes_les_3_heures, Tem_Moy_région, Tem_Max_région, Tem_Min_région], AVG_W(Tem_Moy_départe, D_Superficie), MAX(Tem_Max_départe), MIN(Tem_Min_départe))

La fonction AVG_W (X, Y) prend deux entrées numériques. Elle retourne la moyenne des valeurs X pondérée par Y ; autrement dit, la moyenne pondérée :

$$\text{AVG\_W}\ (X, Y) = \frac{\sum (X \times Y)}{\sum Y}$$

A partir de ces cubes déjà calculés, des analyses plus avancées et plus complexes peuvent être réalisées. Par exemple les températures régionales mensuelles par demi-journée sont directement calculables de $Cube_4$:

---

[3] Nous avons simplifié la définition de l'opération de navigation 'nav' qui correspond à un forage vers le haut (RollUp), présenté dans [Vassiliadis & Skiadopoulos, 2000].

$Cube_5$ = nav($Cube_4$, [MoisN, Région, demi-journée, Tem_Moy_rég_M, Tem_Max_rég_M, Tem_Min_rég_M], AVG(Tem_Moy_région), MAX(Tem_Max_région), MIN(Tem_Min_région))

De la même manière, nous pouvons réaliser (É6), (É7) et (É8).

Ainsi, en intégrant les fonctions d'agrégation dans l'analyse OLAP, nous pouvons, en changeant les fonctions entre les opérations RollUp, effectuer toutes les analyses souhaitées. Donc, nous pouvons changer les fonctions d'agrégation avec les dimensions, les hiérarchies et les niveaux de granularité. Dans le cas de deux fonctions d'agrégation (analyse sur deux dimensions), nous pouvons profiter de la commutativité des fonctions en calculant n'importe quelle fonction avant l'autre. Par contre, avec cette méthode, il n'y a aucune contrainte pour respecter un ordre précis si les fonctions sont non-commutatives ou bien pour forcer le calcul à partir d'un niveau spécifique si les mesures ne peuvent pas s'agréger à partir de niveau de base. Par exemple, les fonctions AVG_W et AVG sont non-commutatives, il n'y a aucune obligation qui impose le calcul du $Cube_5$ à partir de $Cube_4$ et qui interdit de le calculer à partir du $Cube_1$ ce qui donne un résultat erroné (*cf.* § 1.3.2.2).

Au lieu de changer la fonction d'agrégation avec les opérations RollUp, [Franconi & Kamble, 2004] propose d'éviter l'utilisation du RollUp. Par contre, il faut définir toutes les agrégations nécessaires dans des faits agrégés. Comme ces faits agrégés sont définis dès le début de la création de l'entrepôt de données, toutes les contraintes des fonctions non-commutatives et le calcul à partir d'un niveau spécifique peuvent être respectés. Mais avec cette solution, nous ne profitons pas de la commutativité entre les fonctions. En outre, il est difficile de déterminer toutes les agrégations si l'exemple à traiter est compliqué. Par exemple, le nombre de toutes les agrégations possibles pour les températures et les précipitations selon notre exemple malgré sa simplicité est de 312 agrégations.

Modèle Dimensionnel (MD)

Un Modèle Dimensionnel (*MD*) est proposé dans les travaux [Cabibbo & Torlone, 1998]. Il diffère des propositions précédentes car il ne se base pas sur la métaphore du cube de données mais sur la méthode de modélisation multidimensionnelle (*cf.* § 1.2.3.2). Le fait *f-table* relie les dimensions avec les mesures. Les paramètres sont organisés en niveaux hiérarchiques dans les dimensions.

Le schéma de ce modèle qui fait appel à notre exemple, est présenté dans la Figure 8. Les faits sont décrits par des rectangles, les mesures par des cercles en gras liés aux faits et les dimensions par un sous-graphe encerclé en gras qui a pour titre le nom de la dimension. Chaque dimension contient un ensemble des nœuds ordonné par des flèches. Ces nœuds représentent les niveaux de granularité des hiérarchies. Les attributs faibles sont représentés par des parallélogrammes associés aux différents niveaux.

Le formalisme de ce modèle ne donne pas un nom à chaque hiérarchie, ce qui peut causer des confusions entre les parties communes à plusieurs hiérarchies. En ce qui concerne l'agrégation, ce modèle n'est pas différent des modèles de cube précédents, il ne précise pas les fonctions dans le modèle. Donc, il a les mêmes limites avec les ordres d'exécution des fonctions non-commutatives et le calcul à partir d'un niveau spécifique (agrégation contrainte).

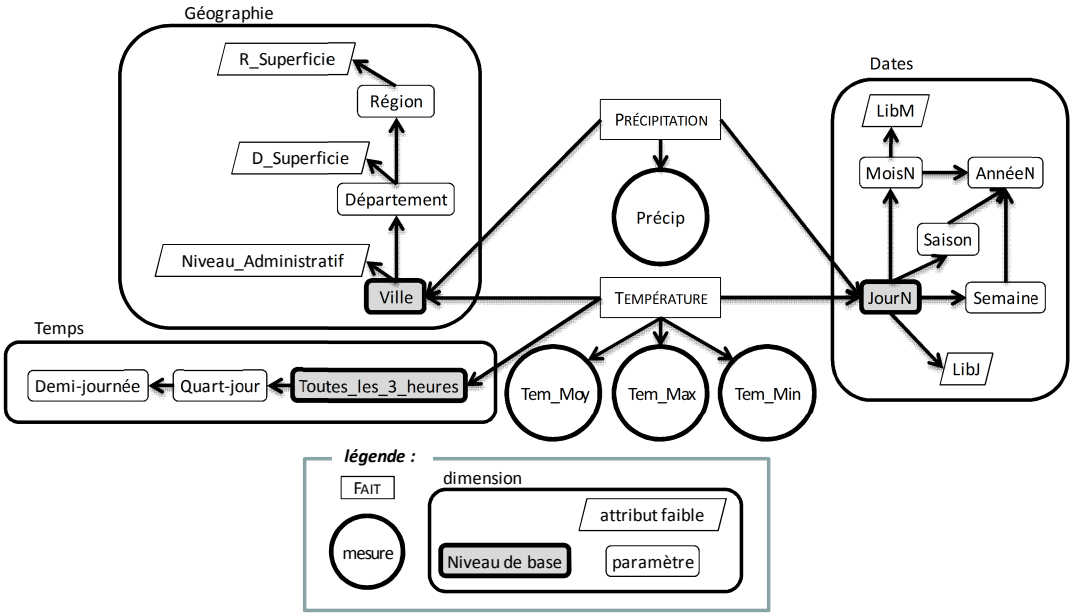


Figure 8 : Modèle *MD* [Cabibbo & Torlone, 1998]

2.3.2.2 Multifonctions générales

Extended Multidimensional Data Model (EMDM)

Les travaux [Pedersen T.B. & Jensen, 1999], [Pedersen T.B., 2000], [Pedersen T.B. et al., 2001] fournissent un modèle basé sur les concepts d'objet et de classe. Ils proposent également un langage de requête algébrique. Ce modèle utilise pour chaque fait un schéma qui relie la classe du fait avec les classes des dimensions correspondantes. Chaque dimension contient une ou plusieurs hiérarchies qui organisent les niveaux de granularité du niveau le plus détaillé ($\perp$) au niveau le plus général ($\top$).

Ce modèle traite les mesures et les dimensions d'une manière identique. Il utilise un fait sans mesures et il considère que tous les concepts qui caractérisent le fait sont des dimensions, même les attributs qui sont considérés comme mesures dans d'autres modèles. Ainsi, les mesures sont représentées par des dimensions.

Dans ce modèle, la sémantique d'agrégation est supportée en précisant pour chaque niveau de granularité (paramètre) de chaque dimension un type d'agrégation : pour les calculs additifs, pour les calculs de moyenne ou pour compter seulement (*cf.* § 2.3.1.3). Ce type lie à chaque niveau un ensemble de fonctions qui ne comprend que les fonctions valides. Néanmoins, chaque fonction est utilisée uniformément pour agréger les valeurs du niveau (comme si elle était une mesure) concerné sur toutes les dimensions et tous les niveaux des hiérarchies.

La Figure 9 présente le schéma modélisant notre exemple. Le formalisme graphique de ce modèle ne supporte pas les modèles en constellation, il faut utiliser un schéma différent pour chaque fait (la Figure 9 (a) pour le fait 'Température' et la Figure 9 (b) pour le fait 'Précipitation'). Nous remarquons que les trois mesures des températures 'Tem_Moy', 'Tem_Max' et 'Tem_Min' et la mesure des précipitions 'Précip' sont gérées comme une dimension comportant un seul paramètre 'Tem' et 'Précip' consécutivement. Ce modèle ne permet pas de définir des attributs faibles ('Niveau_Administratif', 'D_Superficie' et 'R_Superficie') ce qui empêche l'utilisation des fonctions (AVG_W et SELECT_CENTER) qui prennent ces attributs comme entrées (É3), (É5) et (É6).

Pour lier à chaque niveau de granularité un type d'agrégation, les auteurs adoptent une fonction AGGTYPE :

$$\text{AGGTYPE} : P \rightarrow \{\sum, \Phi, c\}$$

- P est l'ensemble des paramètres de schéma ;
- $\sum$ = {SUM, COUNT, AVG, MIN, MAX} est l'ensemble des fonctions utilisées pour les calculs additifs ;
- Φ = {COUNT, AVG, MIN, MAX}, est l'ensemble des fonctions utilisées pour les calculs de moyenne ;
- c = {COUNT} est l'ensemble des fonctions utilisées pour compter seulement.

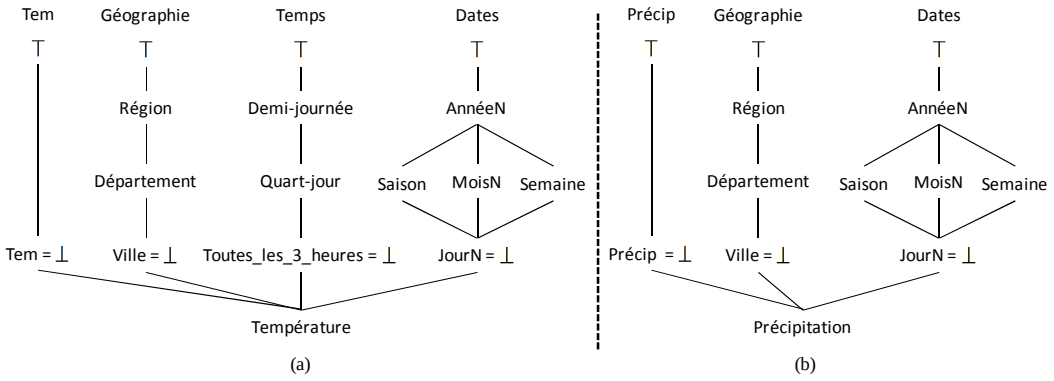


Figure 9 : Modèle EMDM [Pedersen T.B. & Jensen, 1999]

L'un des inconvénients de cette approche est qu'elle ne prend en compte que les fonctions d'agrégation SQL standard. Donc, le modèle EMDM supporte partiellement l'utilisation de plusieurs fonctions pour une seule mesure. Dans notre exemple :

AGGTYPE(Précip) =$\sum$, AGGTYPE(Tem) =Φ, AGGTYPE(Ville) = c, AGGTYPE(JourN) = c, AGGTYPE(Toutes_les_3_heures) = c, AGGTYPE(Département) = c, AGGTYPE(MoisN) = c, …

Le niveau 'Tem' est lié au type Φ, donc il peut être agrégé selon les fonctions AVG, MIN, MAX pour calculer les trois mesures 'Tem_Avg', 'Tem_Max' et 'Tem_Min' de notre exemple. Le niveau 'Précip' est lié au type $\sum$, donc nous pouvons appliquer les fonctions SUM, AVG pour réaliser (É7) et (É8). Mais il n'y a aucune contrainte qui indique que la fonction SUM doit être appliquée sur la dimension 'Dates' et pas sur la dimension 'Géographie' et de manière inversée pour la fonction AVG. Autrement dit, les fonctions associées à un niveau ne sont spécifiques ni à une dimension, ni à une hiérarchie, ni à un niveau de granularité. Cependant les auteurs supposent qu'elles vont être utilisées uniformément dans tout l'espace multidimensionnel. Par ailleurs, le modèle EMDM ne tient pas compte des cas qui nécessitent des agrégations à partir d'un niveau précis autre que le niveau de base (agrégation contrainte).

2.3.2.3 Agrégations multiples dimensionnelles

Modèle Dimensionnel des Faits (DF)

Un modèle conceptuel graphique pour les entrepôts de données a été proposé dans les travaux [Golfarelli et al., 1998a], [Golfarelli et al., 1998b] appelé modèle Dimensionnel des Faits (DF). Ces travaux proposent également une méthodologie semi-automatisée pour construire le modèle DF à partir des schémas Entité-Association décrivant un système d'information opérationnel.

Le modèle DF se base sur une structure arborescente des faits, dimensions et hiérarchies (Figure 10). D'autres caractéristiques qui peuvent être représentées sur les schémas sont les additivités des mesures sur les dimensions. Ce modèle distingue trois types de mesures (*cf.* § 2.3.1.3) : additives, semi-additives, non-additives. Dans ce modèle, les mesures sont additives sur toutes les dimensions par défaut. La semi-additivité est explicitement représentée en associant chaque mesure semi-ou non-additive aux dimensions sur lesquelles les valeurs ne peuvent pas s'ajouter. Si une fonction d'agrégation (autre que SUM) peut être utilisée, elle est indiquée de façon explicite.

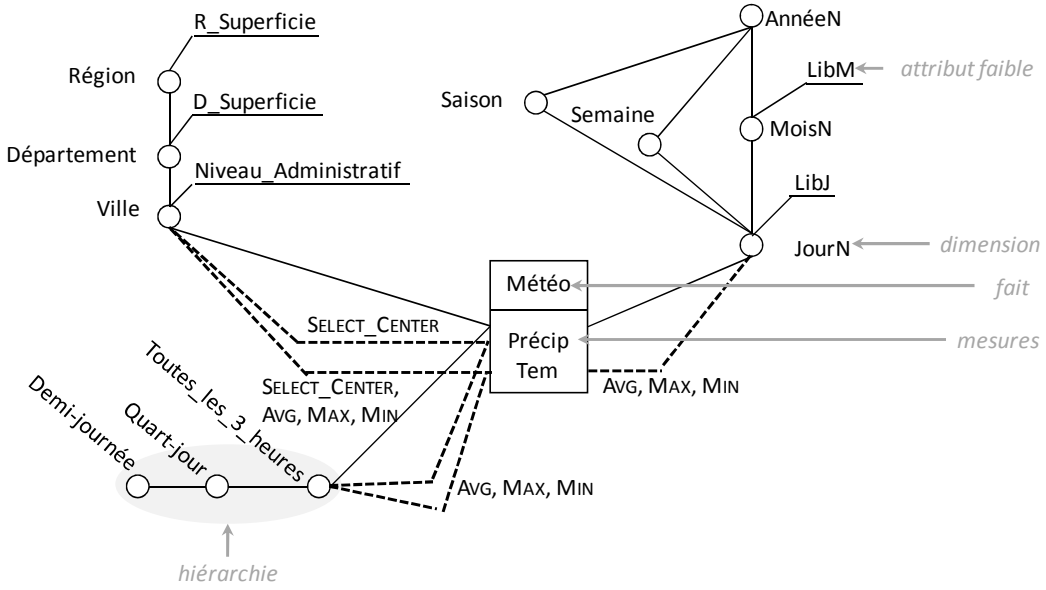


Figure 10 : Modèle DF [Golfarelli et al., 1998a]

Selon le formalisme graphique de ce modèle, nous pouvons simplifier notre exemple en utilisant un seul fait 'Météo' ayant deux mesures 'Précip' et 'Tem' (Figure 10). La mesure 'Tem' peut être agrégée sur les trois dimensions en utilisant les trois fonctions d'agrégation (AVG, MAX, MIN) pour calculer les trois mesures de notre exemple ('Tem_Moy', 'Tem_Max' et 'Tem_Min') afin d'effectuer (É2) et (É4). Nous pouvons également associer la fonction SELECT_CENTER aux mesures 'Tem' et 'Précip' pour effectuer (É3). Selon (É7), la mesure 'Précip' est additive sur la dimension 'JourN', c'est pourquoi elle n'y est pas associée par une ligne pointillée. Selon notre exemple, cette mesure n'est pas agrégée sur la dimension 'Toutes_les_3_heures', donc il existe un lien entre les deux concepts sans fonction d'agrégation.

Malgré la simplicité de l'exemple de la Figure 10 (un fait avec deux mesures), nous remarquons que la présentation de l'additivité de toutes les mesures avec les éléments structurels (les dimensions et le fait) surcharge le schéma et réduit la lisibilité. Par exemple, il existe trois liens entre le fait et les mesures d'un côté, et chacune des deux dimensions 'Ville' et 'Toutes_les_3_heures' de l'autre, avec une liste de trois ou même quatre fonctions d'agrégation parfois.

En outre, le modèle ne donne pas de noms aux hiérarchies, ce qui ne permet pas de les distinguer et peut causer des confusions entre les niveaux des parties communes à plusieurs hiérarchies. Nous ne pouvons pas définir deux types d'agrégation différentes 'Simple' et 'Scientifique' sur la dimension 'Ville'. D'autre part, même pour les parties non-communes aux hiérarchies, ce modèle ne donne pas la possibilité de changer les fonctions d'agrégation ni avec les hiérarchies ni avec les niveaux de granularité. Par exemple, nous ne pouvons ni définir des

fonctions spécifiques aux niveaux 'Région', 'All^{Géographie}' et 'All^{Dates}' pour effectuer (É5), (É6), (É8), ni préciser que la fonction AVG pour la mesure 'Tem' peut s'appliquer au niveau 'Ville' et pas au niveau 'Région'.

Il convient de noter que ce modèle permet l'utilisation de plusieurs fonctions pour la même analyse. Par contre, il ne propose rien pour traiter le cas des fonctions non-commutatives. Finalement, le modèle ne permet pas l'agrégation d'une mesure à partir d'un niveau spécifique (agrégation contrainte).

Yet Another Multidimensional Model (YAM^2)

Les travaux [Abelló, 2002], [Abelló et al., 2002], [Abelló et al., 2006] proposent un modèle conceptuel multidimensionnel O-O (Orienté Objet) appelé YAM². Ce modèle a été développé comme une extension du méta-modèle de diagramme de classe UML « Unified Modeling Language » pour faciliter sa réutilisation et tenter de combler l'absence d'un modèle standard. Il se base sur six types de nœuds :

- **Niveau** : il représente les paramètres de l'analyse. Il hérite de la classe 'Class' du méta-modèle d'UML ;
- **Descripteur** : qui modélise les attributs faibles. Il hérite de la classe 'Attribute' ;
- **Dimension** : c'est un graphe orienté qui regroupe les niveaux et les descripteurs. Il s'agit d'une spécialisation de la classe 'Classifier' ;
- **Cellule** : qui définit un ensemble des instances d'un fait pour une combinaison de niveaux des dimensions. Elle hérite de la classe 'Class' ;
- **Mesure** : c'est un attribut d'une cellule représentant les données à analyser. Elle est une spécialisation de la classe 'Attribute' ;
- **Fait** : il est un graphe orienté qui regroupe les 'Cellules' pour toutes les combinaisons de niveaux des dimensions. Il hérite de la classe 'Classifier'.

Ces nœuds se relient par des relations (arcs) orientées objet tels que l'association, l'agrégation, la généralisation, et la dérivation.

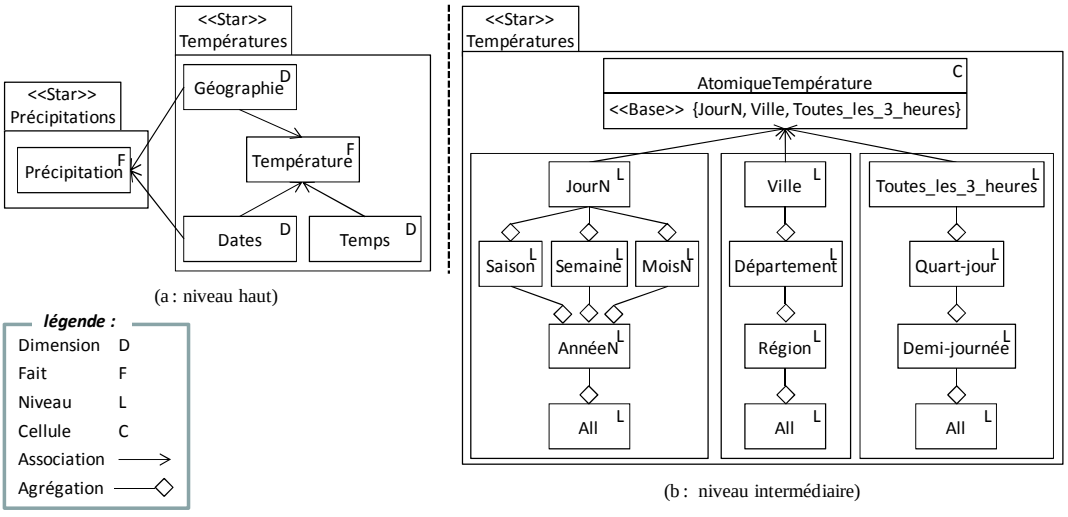


Figure 11 : Modèle YAM^2 (niveaux haut et intermédiaire) [Abelló et al., 2002]

Les différents éléments de modélisation dans YAM^2 ont été définis à trois niveaux (haut, intermédiaire et bas), de sorte qu'ils soient successivement décomposés. La définition de ces niveaux de détails permet à l'utilisateur de se concentrer sur le niveau d'abstraction désiré. Les six types de nœuds sont regroupés en trois paires. Faits et dimensions sont au niveau haut

(Figure 11 (a)). Au niveau intermédiaire, il y a les cellules et les niveaux (Figure 11 (b)). En ce qui concerne le niveau bas, il comprend les mesures et les descripteurs (Figure 12). De plus, à ce niveau, les auteurs définissent « **KindOfMeasure** » afin d'exécuter correctement les fonctions d'agrégation en indiquant comment les mesures doivent être agrégées au long de chaque dimension. Le schéma YAM^2 à ce niveau permet également d'indiquer si un niveau est une source invalide pour le calcul d'une mesure.

La Figure 11 montre les niveaux haut (a) et intermédiaire (b) de notre exemple. YAM^2 permet la modélisation de plusieurs faits en constellation. Par contre, il ne permet pas de distinguer les parties communes à plusieurs hiérarchies.

La Figure 12 montre le niveau bas où en utilisant « KindOfMeasure », nous pouvons définir une fonction d'agrégation pour une ou plusieurs dimensions. Par exemple, pour la mesure 'Tem_Moy', la fonction AVG est définie sur les dimensions 'Dates' et 'Temps' pour effectuer (É2). Egalement pour la mesure 'Précip', « KindOfMeasure » précise la fonction SUM pour la dimension 'Dates' afin d'effectuer (É7). Cependant, nous ne pouvons pas définir plusieurs fonctions (chacune pour une hiérarchie différente) pour la même mesure sur la même dimension, c'est pourquoi nous précisons sur la dimension 'Géographie' soit la fonction SELECT_CENTER (pour les mesures 'Tem_Moy', 'Tem_Max' et 'Tem_Min') afin d'effectuer (É3), soit la fonction AVG (pour la mesure 'Précip') afin d'effectuer l'équivalent de (É4).

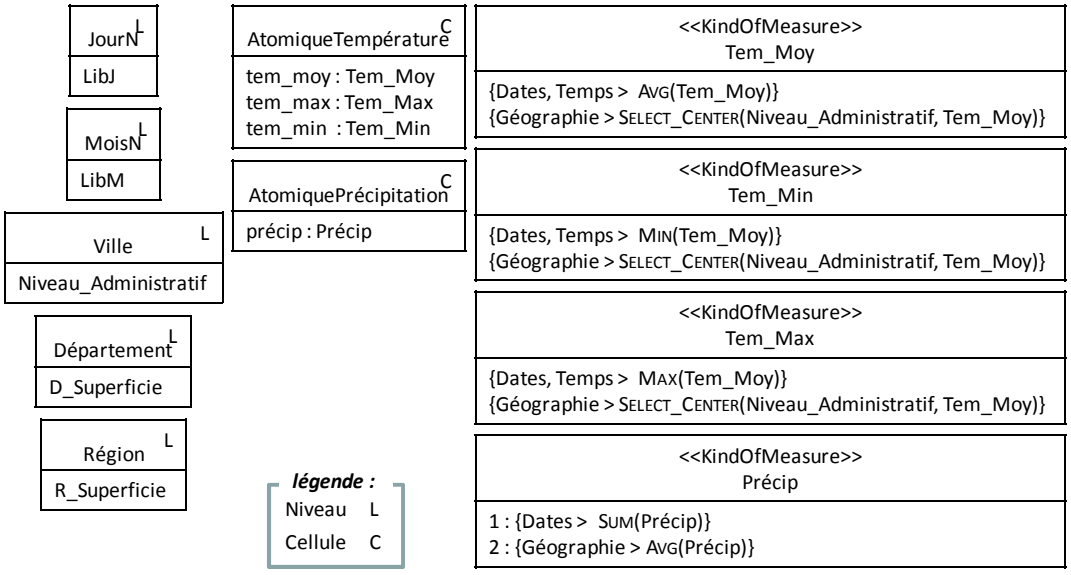


Figure 12 : Modèle YAM^2 (niveau bas) [Abelló et al., 2002]

En outre, nous ne pouvons pas préciser que la fonction AVG pour la mesure 'Précip' doit être appliquée au niveau 'Ville' et pas aux autres niveaux. Autrement dit, ce modèle n'offre pas la possibilité de spécifier une fonction d'agrégation pour chaque niveau de granularité ce qui empêche la réalisation de (É5), (É6), (É8).

Par ailleurs, l'exécution des fonctions non-commutatives est ordonnée. Par exemple, pour agréger la mesure 'Précip', il faut d'abord appliquer la fonction SUM (liée à la valeur 1) avant la fonction AVG (liée à la valeur 2). Le modèle aborde également la commutativité par l'absence de nécessité d'ordonner les fonctions si elles sont commutatives. Par exemple, les fonctions des mesures 'Avg_Moy', 'Avg_Min' et 'Avg_Max' ne sont pas ordonnées. Par contre, le modèle ne tient pas compte de l'agrégation d'une mesure à partir d'un niveau spécifique (agrégation contrainte).

2.3.2.4 Agrégations multiples dimensionnelles et différenciées

Modèle Spatio-Temporal ODMG (ST_ODMG)

La série de travaux [Camossi et al., 2006], [Camossi et al., 2008], [Bertinoa et al., 2009], [Camossi et al., 2009a], [Camossi et al., 2009b] définit une extension spatio-temporelle multi-granulaire (appelée ST_ODMG) du modèle de données ODMG (Object Data Management Group), la norme pour les bases de données orientées objet. Ce modèle se base sur les notions d'objet et de classe. Il réalise la capacité de représentation des données à différents niveaux de détail en utilisant des **granularités** qui représentent les paramètres. Le modèle ST_ODMG propose deux types paramétriques multi-granulaires : spatial et temporel.

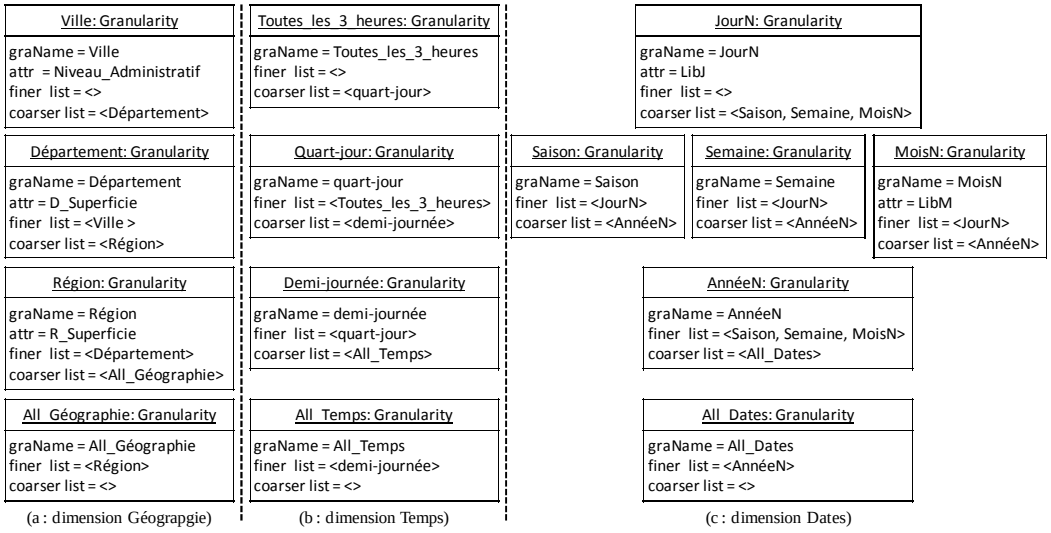


Figure 13 : Modèle ST_ODMG [Camossi et al., 2006]

Des conversions de granularité sont fournies afin de représenter les données au niveau de détail le plus approprié pour une tâche spécifique, c'est-à-dire d'augmenter ou de réduire le niveau de détail utilisé pour la représentation des données. La conversion des caractéristiques géométriques multi-granulaires est obtenue par la composition des opérateurs orientés modèle et des opérateurs de la généralisation de cartographique. Le modèle fournit également des opérateurs pour la conversion des valeurs des mesures quantitatives, spatiales et temporelles. Ces conversions sont classées selon la sémantique de l'opération effectuée: la *sélection* et *l'agrégation* (*cf.* § 2.3.1.5) qui convertissent les valeurs vers une représentation moins détaillée. Leurs fonctions inverses sont la *restriction* (RESTR) et la *fragmentation* (SPLIT) qui convertissent les valeurs des mesures vers une représentation plus fine d'après la propriété héréditaire vers le bas[4] ou selon une distribution de probabilité[5] respectivement.

[4] La propriété héréditaire vers le bas signifie que si une granularité g a une valeur v, la valeur v réfère également à n'importe quelle granularité g' inférieure incluse dans g.

[5] Une distribution de probabilité divise chaque valeur v d'une granularité g entre les granularités inférieures incluses dans g uniformément (c'est-à-dire, toutes les valeurs les plus fines seront les mêmes) ou en fonction d'une distribution non-uniforme.

Les objets représentant les granularités, qui réalisent les dimensions de notre exemple, sont illustrés dans la Figure 13. Les granularités sont liées par les attributs 'finer list' et 'coarser list'. Les granularités de base ont une liste 'finer list' vide, également les granularités les plus supérieurs ont une liste 'coarser list' vide. les attributs faibles sont définis par l'attribut 'attr'.

Selon les spécifications du modèle ST_ODMG, la classe 'Météo' qui décrit les mesures de notre exemple est définie comme suit :

```
class Météo (…){
  attribute Dates_JourN(Temps_Toutes_les_3_heures(Géographie_Ville(float))) Tem_Moy{
              <AVG_Ville → Département, AVG_W(Tem_Moy,D_Superficie)_Département → Région,
              AVG_W(Tem_Moy, R_Superficie)_Région → All_Géographie,
              RESTR_All_Géographie → Ville>,
              <AVG_Toutes_les_3_heures → All_Temps, RESTR_All_Temps → Toutes_les_3_heures>,
              <AVG_JourN → All_Dates,   RESTR_All_Dates → JourN> };
  attribute Géographie_Ville(Dates_JourN(Temps_Toutes_les_3_heures(float))) Tem_Min{
              <MIN_Toutes_les_3_heures → All_Temps, RESTR_All_Temps → Toutes_les_3_heures>,
              <MIN_JourN → All_Dates,   RESTR_All_Dates → JourN>,
              <MIN_Ville → All_Géographie, RESTR_All_Géographie → Ville> };
  attribute Géographie_Ville(Dates_JourN(Temps_Toutes_les_3_heures(float))) Tem_Max{
              <MAX_Toutes_les_3_heures → All_Temps, RESTR_All_Temps → Toutes_les_3_heures>,
              <MAX_JourN → All_Dates,   RESTR_All_Dates → JourN>,
              <SELECT_CENTER_Ville → All_Géographie, RESTR_All_Géographie → Ville> };
  attribute Géographie_Ville(Dates_JourN(int)) Précip{
              <SUM_JourN → AnnéeN, AVG_AnnéeN → All_Dates, RESTR_All_Dates → AnnéeN,
              SPLIT_AnnéeN → JourN>,
              <SELECT_CENTER_Ville → All_Géographie, RESTR_All_Géographie → Ville> };
  };
```

Donc, chaque mesure (attribut) est associée à un ensemble de fonctions qui convertissent (vers le haut et vers le bas) ses représentations. Les fonctions peuvent se changer d'une granularité à l'autre ou bien peuvent être appliquées entre plusieurs granularités successives. Ainsi, il suffit de définir une fonction entre les granularités la plus basse et la plus générale pour que cette fonction soit appliquée sur toute la dimension.

Comme la classe 'Météo' précédente le montre, cette possibilité de changer la fonction avec les granularités rend possible l'application de toutes les opérations demandées pour notre exemple (É2), (É4), (É5), (É6), (É7), (É8) et l'équivalent de (É3) qui est associé aux mesures 'Tem_Max' et 'Précip'. Mais, comme ce modèle ne distingue pas les parties communes à plusieurs hiérarchies, il ne permet pas d'agréger la même mesure selon des fonctions différentes pour la même granularité sur des hiérarchies différentes. Donc, pour une même mesure, nous ne pouvons pas associer à la fois la fonction de l'agrégation simple (É3) et les fonctions de l'agrégation scientifique (É4), (É5) et (É6). Par exemple, dans la classe 'Météo', nous associons aux granularités 'Ville', 'Département' et 'Région' soit l'agrégation simple (pour la mesure 'Précip') soit l'agrégation scientifique (pour la mesure 'Tem_Moy').

Par ailleurs, l'ordre d'exécution des fonctions dépend de l'ordre dans lequel les dimensions sont ordonnées lors de la déclaration des mesures. L'exécution commence sur la dimension qui est dans les parenthèses les plus internes. Par exemple, selon la classe 'Météo' l'agrégation de la mesure 'Tem_Moy' commence sur la dimension 'Géographie' ensuite la dimension 'Temps' et finalement la dimension 'Dates'. Par contre, le modèle ST_ODMG n'exprime pas la commutativité entre les fonctions. Par exemple, l'exécution des fonctions des mesures 'Tem_Max' et 'Tem_Min' est ordonnée bien que leurs fonctions sont commutatives.

En outre, ce modèle permet de définir partiellement des agrégations calculées à partir des niveaux spécifiques autres que les niveaux de base (agrégation contrainte). Il offre la

possibilité de définir des agrégations successives, c'est-à-dire, chaque agrégation est réalisée à partir des résultats de l'agrégation précédente. Par exemple, la température moyenne régionale ('Tem_Moy' à la granularité 'Région') est calculée à partir des températures moyennes à la granularité 'Département' qui sont à leur tour calculées à partir des températures à la granularité 'Ville'. Cependant, le modèle ne permet pas de définir dans la même hiérarchie deux agrégations (avec deux fonctions différentes) calculées à partir de la même granularité pour deux granularités différentes.

Modèle de cube de données spatiales

[Salehi, 2009] propose un cadre théorique pour identifier des contraintes d'intégrité (CI) dans les cubes de données spatiales. Afin d'interdire les agrégations erronées et proposer des agrégations correctes, l'auteur propose un ensemble de CIs d'additivité « summarizability ». Chaque CI d'additivité est définie par une combinaison d'une mesure, une fonction d'agrégation, une dimension, un niveau inférieur « *from level* » (la base de calcul), un niveau supérieur « *to level* » (le paramètre concerné) et un ensemble de mesures existantes au niveau inférieur « *measure(s) at from level* » (pour calculer les mesures dérivées).

En utilisant cette combinaison, nous pouvons définir toutes les agrégations demandées pour notre exemple comme le Tableau 7 le montre :

Tableau 7 : Définition des agrégations [Salehi, 2009]

Mesure et Agrégation	CI
Tem_Moy :É2	*measure:*"Tem_Moy", *aggregation function:*"AVG", *dimension:*"Dates"
Tem_Moy :É4	*measure:*"Tem_Moy", *aggregation function:*"AVG", *dimension:*"Géographie", *from level:*"Ville", *to level:*"Département"
Tem_Moy :É5	*measure:*"Tem_Moy", *aggregation function:*"AVG_W(Tem_Moy,D_Superficie)", *dimension:*"Géographie", *from level:*"Département", *to level:*"Région"
Tem_Moy :É6	*measure:*"Tem_Moy", *aggregation function:*"AVG_W(Tem_Moy, R_Superficie)", *dimension:*"Géographie", *from level:*"Région", *to level:*"All"
Tem_Max : Equivalent d'É2	*measure:*"Tem_Max", *aggregation function:*"Max", *measure(s) at from level:*"Tem_Moy"
Tem_Max : Equivalent d'É3	*measure:*"Tem_Max", *aggregation function:*"SELECT_CENTER", *dimension:*"Géographie", *measure(s) at from level:*"Tem_Moy"
Précip :É7	*measure:*"Précip", *aggregation function:*"SUM", *dimension:*"Dates", *to level:*"AnnéeN"
Précip :É8	*measure:*"Précip", *aggregation function:*"AVG", *dimension:*"Dates", *from level:* "AnnéeN", *to level:*"All"

D'un côté, cette combinaison permet de définir ou d'interdire des fonctions d'agrégation :

- Pour chaque dimension (Tableau 7 : É2 et équivalent d'É3),
- Pour chaque paramètre (Tableau 7 : É4, É5, É6, É8),
- Indépendamment des dimensions (Tableau 7 : équivalent d'É2).

Elle permet également de définir des agrégations à partir des niveaux spécifiques (agrégation contrainte). Cela se fait à l'aide du niveau inférieur « *from level* ». Par exemple, les CIs précédentes indiquent que les températures moyennes régionales (É5) et les précipitations générales (É8) sont calculées à partir des températures départementales et des précipitations annuelles respectivement.

D'autre part, elle ne distingue pas les parties communes à plusieurs hiérarchies. Donc, nous ne pouvons pas définir deux agrégations différentes au même paramètre pour la même mesure. Par exemple, Tableau 7 défini sur la dimension 'Géographie' soit l'agrégation

simple pour la mesure 'Tem_Max' (Equivalent d'É3) soit l'agrégation scientifique pour la mesures 'Tem_Moy' (É4, É5 et É6). L'inconvénient le plus important de cette proposition est qu'elle n'aborde pas la composition des fonctions d'agrégation, ce qui peut produire des résultats erronés si les fonctions sont non-commutatives.

2.3.2.5 Agrégations multiples dimensionnelles et hiérarchiques, et différenciées

Modèle d'agrégations statiques et dynamiques

Afin de tenir compte des caractéristiques statiques et dynamiques de l'agrégation, les travaux [Prat et al., 2010], [Prat et al., 2011] proposent de les représenter avec des objets (diagrammes de classes UML) et des règles (en langue PRR « Production Rule Representation »). Les agrégations statiques sont représentées dans les diagrammes de classes UML, combinées avec les règles de PRR qui représentent les agrégations dynamiques (c'est-à-dire comment choisir les fonctions d'agrégations et comment elles peuvent être effectuées en fonction du contexte).

Les auteurs distinguent quatre types d'agrégation (*cf.* § 2.3.1.3) : dans le premier type, toutes les fonctions d'agrégation peuvent être utilisées. Dans le deuxième, la fonction SUM ne peut pas être utilisée. Le troisième ne permet que la fonction COUNT. Le quatrième est consacré à des mesures qui ne peuvent pas être agrégées. Autrement dit, aucune fonction n'est autorisée.

Premièrement, le schéma du diagramme de classe de ce modèle permet au concepteur de l'entrepôt de données de spécifier des ensembles des fonctions d'agrégation [Prat et al., 2010] ou des types d'agrégation [Prat et al., 2011] applicables à une mesure :

- Indépendamment des dimensions et des hiérarchies,
- Pour toutes les hiérarchies d'une dimension,
- Pour une hiérarchie spécifique,
- Pour une sous-hiérarchie.

Deuxièmement, les règles d'agrégation proposées dans ce modèle sont de quatre types :

- **Les règles d'agrégation sémantiques** sont basées sur la sémantique des éléments (dimensions, mesures, les fonctions d'agrégation…) du modèle multidimensionnel. Par exemple, les mesures du type 'stock'[6] ne sont pas additives sur les dimensions temporelles ;
- **Les règles d'agrégation syntaxiques** expriment les propriétés mathématiques des fonctions d'agrégation (commutativité, distributivité, composition des fonctions…) ;
- **Les préférences de l'utilisateur** qui indiquent quelles sont les agrégations préférables dans le cas où plusieurs fonctions d'agrégation sont applicables ;
- **Les règles d'exécution d'agrégation** qui indiquent comment une fonction d'agrégation doit être exécutée une fois qu'elle a été choisie. Ces règles sont nécessaires pour faire face aux hiérarchies non-standard. Elles peuvent aussi expliquer comment les valeurs nulles sont prises en compte dans le calcul de l'agrégation.

[6] Une mesure de type stock enregistre l'état à des points spécifiques dans le temps.

La Figure 14 illustre le diagramme de classes correspondant à notre exemple. Selon le formalisme graphique de ce modèle, les hiérarchies sont construites sur des relations 'RollUp' entre les niveaux de granularité 'DimensionLevel'. Chaque niveau de granularité comprend un paramètre identifiant (indiqué par '{id}') et un ensemble d'attributs faibles.

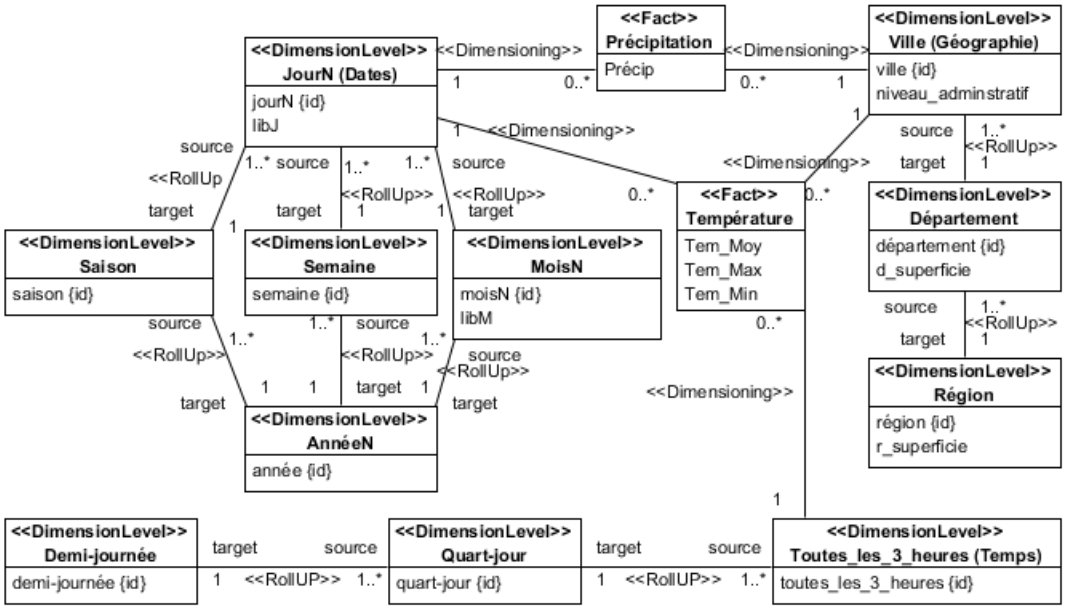


Figure 14 : Schéma conceptuel [Prat et al., 2011]

Bien que les auteurs proposent d'intégrer les fonctions dans le schéma conceptuel multidimensionnel (Figure 14), ils préfèrent les présenter séparément dans un tableau. Le Tableau 8 représente la définition des agrégations statiques liées à trois mesures 'Tem_Moy', 'Tem_Max' et 'Précip' qui effectuent toutes les opérations (É2), (É3), (É4), (É5), (É6), (É7) et (É8). Chaque niveau de granularité de la dimension 'Géographie' est lié à deux fonctions : une pour réaliser l'analyse scientifique (AVG ou AVG_W) et l'autre pour l'analyse simple (SELECT_CENTER). Les niveaux de granularité des autres dimensions sont liés à une fonction d'agrégation correspondante à la mesure concernée. Les agrégations de la mesure 'Tem_Min' sont identiques à celles de la mesure 'Tem_Max' sauf qu'elles utilisent MIN au lieu de MAX.

Tableau 8 : Définition des agrégations statiques [Prat et al., 2011]

Mesure	Niveaux de Dimension	Fonctions Définies
Tem_Moy	Toutes_les_3_heures → Quart-jour → Demi-journée → All	AVG
	JourN → MoisN → AnnéeN → All	AVG
	JourN → Semaine → AnnéeN → All	AVG
	JourN → Saison → AnnéeN → All	AVG
	Ville → Département	AVG, SELECT_CENTER
	Département → Région → All	AVG_W, SELECT_CENTER
Tem_Max	Toutes_les_3_heures → Quart-jour → Demi-journée → All	MAX
	JourN → MoisN → AnnéeN → All	MAX
	JourN → Semaine → AnnéeN → All	MAX
	JourN → Saison → AnnéeN → All	MAX
	Ville → Département → Région → All	MAX, SELECT_CENTER
Précip	JourN → MoisN → AnnéeN	SUM
	JourN → Semaine → AnnéeN	SUM
	JourN → Saison → AnnéeN	SUM

	AnnéeN → All	AVG
	Ville → Département	AVG, SELECT_CENTER
	Département → Région → All	AVG_W, SELECT_CENTER

Les règles d'agrégation sont des règles générales, c'est-à-dire, qu'elles ne sont pas liées forcement à l'exemple traité. Si une règle ne convient pas à l'exemple, les auteurs proposent de la désactiver. Les règles syntaxiques indiquent la séquence correcte des fonctions d'agrégation, sur une dimension ou entre les différentes dimensions. Par exemple, ces règles précisent si nous pouvons appliquer la fonction SUM après la fonction AVG ou calculer la moyenne des moyennes (AVG après AVG).

La Tableau 9 résume les règles syntaxiques qui peuvent être définies pour notre exemple. Il indique si les fonctions en colonnes peuvent s'appliquer (Oui) ou pas (Non) après les fonctions en lignes. Par exemple, selon la première ligne du tableau, après l'application de la fonction SUM pour calculer les précipitations mensuelles, nous pouvons appliquer soit la fonction SUM pour calculer les précipitations annuelles (É7), soit les fonctions AVG et AVG_W afin de réaliser l'analyse scientifique sur la dimension 'Géographie' (équivalant de (É4), (É5) et (É6)), soit la fonction SELECT_CENTER afin de réaliser l'analyse simple sur la dimension 'Géographie' (équivalant de (É3)).

Dans ce tableau, nous trouvons trois cas (indiqués par un point d'interrogation) où les règles syntaxiques ont du mal à préciser la séquence correcte des fonctions :

- L'application d'AVG après AVG doit être autorisée pour pouvoir calculer les températures moyennes mensuelles (É2) après avoir commencé une analyse scientifique (É4). Par contre, elle doit être interdite afin d'éviter le calcul de la moyenne (É2) des moyennes (É2) sur la dimension 'Dates' ;
- L'application d'AVG_W après AVG doit être autorisée pour continuer une analyse scientifique (calculer les températures régionales (É5) après avoir calculé les températures départementales (É4)). Cependant, elle doit être interdite afin d'éviter des résultats erronés à cause de la non-commutativité (*cf.* § 1.3.2.2) si nous calculons les températures régionales après avoir calculé les températures moyennes mensuelles ;
- L'application de SELECT_CENTER après AVG doit être autorisée pour réaliser l'analyse scientifique (É3) après calcul des températures moyennes mensuelles (É2). Et elle doit être interdite afin d'éviter une analyse simple (É3) après avoir commencé une analyse scientifique (É4).

Tableau 9 : Définition des agrégations dynamiques [Prat et al., 2011]

Est applicable après ✍	**SUM**	**AVG**	**AVG_W**	**SELECT_CENTER**	**MAX**	**MIN**
SUM	Oui	Oui	Oui	Oui	Non	Non
AVG	Non	**?**	**?**	**?**	Non	Non
AVG_W	Non	Oui	Oui	Non	Non	Non
SELECT_CENTER	Oui	Oui	Non	Oui	Oui	Oui
MAX	Non	Non	Non	Oui	Oui	Non
MIN	Non	Non	Non	Oui	Non	Oui

Ainsi, bien que le modèle supporte les changements des fonctions d'agrégation entre les dimensions, les hiérarchies et les paramètres, il aborde partiellement les fonctions non-commutatives. Par contre, il ne permet de définir que partiellement des agrégations à partir des niveaux spécifiques autres que les niveaux de base (agrégation contrainte) où les agrégations devraient être successives. Par ailleurs, il convient de noter que, après l'exécution des règles de

choix de fonctions, l'ensemble des fonctions candidates peut être vide, terminant ainsi le processus d'agrégation. Par exemple, s'il y avait 'Non' au lieu de tous les points d'interrogation dans le Tableau 9, aucune agrégation ne serait possible après une application d'une fonction AVG. Il est également possible que plusieurs fonctions restent candidates, ce qui nécessite une interaction de l'analyste pour qu'il en choisisse une parmi les candidates.

Modèle et contraintes pour les entrepôts de données spatiales

Un modèle conceptuel spatio-multidimensionnel basé sur UML étendu par un modèle d'agrégation est proposé par [Boulil et al., 2011]. Le modèle d'agrégation permet aux utilisateurs de définir des règles d'agrégation et il vérifie leur cohérence par rapport à des contraintes d'additivité sémantiques générales exprimées en OCL et valables pour tous les entrepôts de données.

Selon le méta-schéma proposé dans ce travail (Figure 15), l'utilisateur peut exprimer cinq types de règles d'agrégation :

- **AggRule** : définit les fonctions d'agrégation « Aggregator » possibles « allowed » et interdites « forbidden » pour l'ensemble des mesures d'un fait « involvedFact » ;
- **MeasureAggRule** : définit les fonctions possibles et interdites pour une mesure, indépendamment des dimensions ;
- **DimensionAggRule** : définit les fonctions possibles et interdites pour une mesure, sur une dimension ;
- **HierarchyAggRule** : définit les fonctions possibles et interdites pour une mesure, sur une hiérarchie ;
- **2LevelsAggRule** : définit les fonctions possibles et interdites qui agrègent une mesure entre deux paramètres « fromLevel » et « toLevel » d'une hiérarchie.

Selon l'attribut « distributive », les règles d'agrégation et les fonctions peuvent être distributives ou pas.

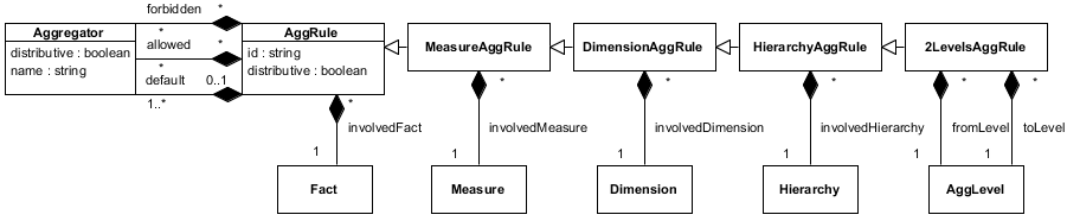


Figure 15 : Méta-modèle d'agrégation [Boulil et al., 2011][7]

Les contraintes d'additivité sémantiques de cette proposition sont équivalentes aux règles d'agrégation sémantiques et syntaxiques proposées dans le modèle d'agrégations statiques et dynamiques [Prat et al., 2010], [Prat et al., 2011] où elles expriment les compatibilités entre les mesures, les dimensions et les fonctions d'agrégation. Elles peuvent également préciser l'ordre d'exécution des fonctions d'agrégation en fonction de leurs

[7] Ce méta-schéma est simplifié. Nous ne présentons que la partie qui concerne l'agrégation. Pour plus de détails le lecteur est invité à consulter [Boulil et al., 2011].

dépendances sémantiques. D'autres types de contraintes d'additivité (de schéma et d'exhaustivité de données) sont proposés pour traiter les hiérarchies non-standard.

De manière similaire au modèle d'agrégations statiques et dynamiques [Prat et al., 2010], [Prat et al., 2011], les contraintes d'additivité sémantiques ont des limites pour aborder les fonctions non-commutatives (*cf.* Tableau 9). En outre, cette proposition permet de définir partiellement des agrégations calculées à partir des niveaux autres que les niveaux de base (agrégation contrainte). Cela se fait en utilisant « fromLevel » pour une règle « 2LevelsAggRule ». Mais dans ce cas, les agrégations devraient être successives. Plus précisément, cette proposition n'offre pas la possibilité de définir deux agrégations (avec deux fonctions différentes), pour deux paramètres différents « toLevel » situés dans la même hiérarchie et calculés à partir du même paramètre « fromLevel ».

2.3.2.6 Les outils commerciaux

En ce qui concerne les outils commerciaux, nous présentons comment les agrégations sont supportées par « Business Objects » [BO XI 3.1 SP3, 2010], [BO XI 3.1 SP6, 2013] et « Analysis Services de Microsoft » [Harinath et al., 2012] qui sont parmi les outils les plus utilisés.

« Business Objects »

« Business Objects » utilise une seule fonction d'agrégation pour chaque mesure [BO XI 3.1 SP3, 2010], [BO XI 3.1 SP6, 2013]. Cela rend difficile la possibilité de réaliser des agrégations compliquées et composées telles que celles demandées dans notre exemple. Ainsi, « Business Objects » ne permet ni de changer la fonction d'agrégation avec les dimensions, les hiérarchies ou les paramètres ni de réaliser des agrégations calculées à partir des niveaux autres que les niveaux de base (agrégation contrainte).

« Analysis Services de Microsoft »

« Analysis Services » associe normalement une fonction d'agrégation à chaque mesure mais il offre également la possibilité de plusieurs techniques d'agrégation pour traiter des cas d'agrégation particuliers [Cameron, 2009], [Smith et al., 2009], [Harinath et al., 2009], [Harinath et al., 2012].

Premièrement, l'**agrégations par-compte « By-account »** : parfois une seule mesure devrait être agrégée différemment selon les valeurs correspondantes d'un paramètre (*type de compte*). Par exemple, le montant d'un compte de revenus devrait s'ajouter sur la dimension de temps, mais le montant d'un compte d'inventaire ne le devrait pas. L'agrégation *par-compte* nous permet d'avoir des définitions d'agrégation différentes en fonction des *types de compte* différents pour la même mesure.

Dans notre exemple, nous pouvons supposer que les températures moyennes sont agrégées différemment dans les régions. Par exemple dans certaines régions, les températures sont calculées comme moyennes des températures départementales (AVG) et dans d'autres régions comme moyennes pondérées par la superficie des départements (AVG_W). Pour définir ce type d'agrégation, il faut d'abord déterminer la dimension 'Géographie' comme dimension de type '*compte*', ensuite préciser la fonction d'agrégation (AVG ou AVG_W) correspondante à chaque région ('*type de compte*'). Finalement, nous avons besoin d'appliquer la fonction d'agrégation *BYACCOUNT* à la mesure concernée ('Tem_Moy').

L'inconvénient de ce type d'agrégation est que la fonction utilisée dans chaque région va intervenir sur toutes les dimensions et pas seulement entre les paramètres 'Département' et 'Région'. En outre, nous ne pouvons pas définir une autre agrégation '*par-compte*' différente sur la même dimension pour une autre mesure. Autrement dit, toutes les mesures utilisant la

fonction d'agrégation *BYACCOUNT* utilisent les mêmes fonctions définies pour chaque '*type de compte*'.

Deuxièmement, les **mesures semi-additives** : les dimensions de date sont traitées spécialement par « Analysis Services » où certaines mesures (semi-additives) sont agrégées différemment par des fonctions d'agrégation telles que BYACCOUNT, FIRSTNONEMPTY, LASTNONEMPTY, FIRSTCHILD, LASTCHILD, AVERAGEOFCHILDREN, ou NONE. Malgré le fait que les mesures pourraient être semi-additives sur une autre dimension, la dimension de date est la seule que « Analysis Services » prenne en charge [Cameron, 2009]. Donc, « Analysis Services » permet de changer partiellement les fonctions d'agrégation entre les dimensions.

Troisièmement, **RollUp personnalisé « Custom RollUp »** : l'outil « Analysis Services » offre la possibilité d'appliquer des Rollup personnalisés à une hiérarchie de plusieurs façons [Cameron, 2009], [Smith et al., 2009], [Harinath et al., 2009], [Harinath et al., 2012] :

- Par l'utilisation de la propriété « CustomRollupExpression » qui contient une expression MDX (Multidimensional Expressions) qui remplace le « RollUp » par défaut pour un paramètre ;
- Par l'utilisation de la propriété « CustomRollupColumn » qui indique, à une colonne dans une table relationnelle dans la base de données, où sont stockées les expressions MDX pour les membres (instances) d'un paramètre ;
- Par l'utilisation des opérateurs unaires avec la propriété « UnaryOperatorColumn » qui sont utilisés pour résoudre le problème de l'agrégation sur un type particulier de hiérarchie (hiérarchie d'attributs parent-enfant). Une hiérarchie parent-enfant est construite à partir d'un seul attribut parent. Un attribut parent décrit une relation de jointure réflexive dans une table de dimension principale. Un opérateur unaire est un opérateur qui prend un seul argument (instance d'un paramètre) et agrège sa valeur à son parent. Les opérateurs unaires permettent de spécifier des fonctions d'agrégation de base : ajouter, soustraire, multiplier, multiplier par un facteur, diviser et un cas particulier (sans agrégation). Comme pour « CustomRollupColumn », les opérateurs unaires doivent être stockés sous forme de colonne dans une table relationnelle.

Les deux dernières approches (« CustomRollupColumn » et les opérateurs unaires) représentent des fonctions d'agrégation mais elles ne sont liées ni à une dimension, ni à une hiérarchie, ni à un paramètre. Elles sont liées à un membre (une instance) d'un paramètre, c'est-à-dire, à une ligne dans la table de la dimension. Cela peut :

- Augmenter l'espace de stockage,
- Entraîner des difficultés en ce qui concerne la mise-à-jour des données,
- Diminue la performance [Harinath et al., 2009], [Harinath et al., 2012].

En outre, bien que « CustomRollupExpression » associe une fonction d'agrégation à un paramètre (fonction différenciée), « Analysis Services » supporte partiellement le changement des fonctions d'agrégation entre les paramètres parce que RollUp personnalisé affecte une seule dimension et toutes les mesures qui l'utilisent. Autrement dit, RollUp personnalisé agrège toutes les mesures uniformément.

Par ailleurs, une autre technique alternative est utilisée pour effectuer des agrégations personnalisées. Cette technique se base sur les **mesures calculées** (dérivées). Nous pouvons créer une mesure calculée à partir d'une mesure stockée cachée (l'utilisateur ne voit que la mesure calculée). L'expression MDX de la mesure calculée peut contrôler si elle doit effectuer ou non une combinaison d'une agrégation avec la fonction d'agrégation de la mesure de base. Les inconvénients de cette technique sont, d'une part, qu'elle ne profite pas de la commutativité entre les fonctions. D'autre part, qu'elle a besoin d'ajouter une nouvelle mesure calculée pour chaque agrégation différente. Enfin, il faut la combiner avec les autres mesures calculées. Cela

rend complexe l'expression MDX associée. En effet, si les agrégations utilisent des fonctions non-standard comme AVG_W et SELECT_CENTER de notre exemple, il faut définir cinq mesures combinées pour les températures moyennes et six pour les précipitations pour effectuer toutes les agrégations demandées (É2, É3, É4, É5, É6, É7, É8).

Le langage MDX permet la construction d'ensembles de données (qui seront regroupées par des fonctions d'agrégation) à l'aide des fonctions : PeriodsToDate, YTD, QTD, MTD, Crossjoin, Cousin, Descendants, Children, Hierarchize, Dimension, Hierarchy, Level et Members. Toutefois, cette possibilité n'est pas liée à notre problème : changer la fonction d'agrégation selon l'axe d'analyse, la hiérarchie et le paramètre.

Lorsque plusieurs calculs sont spécifiés pour une mesure, « Analysis Services » utilise un ordre spécifique pour évaluer les calculs. Il calcule d'abord la fonction régulière de la mesure. Ensuite, si les dimensions ont des Rollup personnalisés et des opérateurs unaires, tous les opérateurs unaires sont appliqués, suivis par les Rollup personnalisés selon l'ordre des dimensions dans le cube. Ainsi, « Analysis Services » aborde les fonctions non-commutatives mais il ne profite pas de la commutativité entre les fonctions et la capacité du concepteur à contrôler l'ordre d'exécution des fonctions est limitée.

Finalement, en ce qui concerne le calcul des mesures à partir d'un niveau précis (agrégation contrainte), « Analysis Services » permet de le réaliser en utilisant les Rollup personnalisés qui vont être appliqués à toutes les mesures uniformément.

2.3.2.7 Bilan

Le Tableau 10 résume les travaux sur les agrégations multidimensionnelles.

- La première colonne 'Intégration des fonctions d'agrégation' montre comment les propositions intègrent les fonctions d'agrégation dans la modélisation multidimensionnelle ;
- La deuxième colonne 'Multifonctions' précise si les propositions offrent la possibilité d'agréger la même mesure avec plusieurs fonctions d'agrégation ;
- Les colonnes 'Dimension', 'Hiérarchie' et 'Niveau' montrent si les propositions permettent de changer les fonctions d'agrégation avec les dimensions (fonction multiple dimensionnelle), les hiérarchies (fonction multiple hiérarchique) et les paramètres (fonction différenciée) ;
- Les colonnes 'Commutativité' et 'Non-Commutativité' présentent si les travaux abordent le cas des fonctions commutatives et non-commutatives ;
- La colonne 'Agrégation contrainte' précise si les travaux traitent le cas où la mesure doit être agrégée à partir d'un niveau spécifique autre que le niveau de base.

Le symbole '✓' est utilisé pour exprimer le fait que la proposition supporte parfaitement la caractéristique concernée tandis que le symbole '~' exprime un support partiel. Nous avons utilisé le symbole '×' pour montrer que dans les travaux [Pedersen T.B. & Jensen, 1999], [Pedersen T.B. et al., 2001], parce que les niveaux de granularité sont considérés comme des mesures, les fonctions sont utilisées uniformément dans tout l'espace multidimensionnel même si elles sont définies pour des niveaux de granularité spécifiques (*cf.* § Extended Multidimensional Data Model (EMDM)).

Aucune proposition ne supporte l'ensemble des caractéristiques. [Golfarelli et al., 1998a], [Golfarelli et al., 1998b], [Abelló et al., 2002], [Abelló et al., 2006] sont les seuls qui présentent les fonctions d'agrégation aux utilisateurs dans le modèle conceptuel. Toutes les propositions offrent la possibilité de plusieurs fonctions d'agrégation pour la même mesure sauf « Business Objects » et le modèle EMDM de [Pedersen T.B. & Jensen, 1999], [Pedersen T.B. et

al., 2001]. Ce dernier offre cette possibilité partiellement parce qu'il ne supporte que les fonctions standards. Tous les autres travaux permettent de changer les fonctions d'agrégations entre les dimensions. Ce changement peut être différent d'une mesure à l'autre sauf dans « Analysis Services de Microsoft » où il est uniforme pour toutes les mesures.

Nous remarquons que tous les travaux qui ne prédéfinissent pas les fonctions d'agrégation en les intégrant au cours de l'interrogation permettent de changer les fonctions avec les dimensions, les hiérarchies et les paramètres. Par contre, ils ne nécessitent pas le respect de la non-commutativité entre les fonctions et les agrégations contraintes. Nous remarquons également qu'il n'y a que les travaux [Abelló et al., 2002], [Abelló et al., 2006] qui abordent complètement à la fois la commutativité et la non-commutativité entre les fonctions. Les travaux [Prat et al., 2010], [Prat et al., 2011], [Boulil et al., 2011] mettent en œuvre l'ordre d'exécution des fonctions en général, indépendamment des mesures concernées, ce qui rend le traitement des fonctions non-commutatives limité.

Tableau 10 : Synthèse des agrégations dans l'état de l'art

	Intégration des fonctions d'agrégation	**Multifonctions**	**Dimension**	**Hiérarchie**	**Niveau**	**Commutatives**	**Non-Commutatives**	**Agrégation contrainte**
[Li & Wang, 1996]	OLAP	✓	✓	✓	✓	✓		
[Agrawal et al., 1997]	OLAP	✓	✓	✓	✓	✓		
[Gyssens & Lakshmanan, 1997]	OLAP	✓	✓	✓	✓	✓		
[Thomas & Datta, 1997], [Datta & Thomas, 1999]	OLAP	✓	✓	✓	✓	✓		
[Lehner, 1998]	OLAP	✓	✓	✓	✓	✓		
[Cabibbo & Torlone, 1998]	OLAP	✓	✓	✓	✓	✓		
[Golfarelli et al., 1998a], [Golfarelli et al., 1998b]	Modèle conceptuel	✓	✓			✓		
[Pedersen T.B. & Jensen, 1999], [Pedersen T.B. et al., 2001]	Fonction Aggtype	~			×			
[Vassiliadis & Skiadopoulos, 2000]	OLAP	✓	✓	✓	✓	✓		
[Abelló et al., 2002], [Abelló et al., 2006]	Modèle conceptuel	✓	✓			✓	✓	
[Franconi & Kamble, 2004]	Agrégation prédéfinies	✓	✓	✓	✓		✓	✓
[Camossi et al., 2006], [Bertinoa et al., 2009]	Classe du fait	✓	✓		✓		✓	~
[Salehi, 2009]	Contraintes d'intégrité	✓	✓		✓	✓		✓
[Pardillo et al., 2010]	OLAP	✓	✓	✓	✓	✓		
[Prat et al., 2010], [Prat et al., 2011]	Tableau supplémentaire	✓	✓	✓	✓	✓	~	~
[Boulil et al., 2011]	Instance de méta-schéma	✓	✓	✓	✓	✓	~	~
« Business Objects »	Modèle							
« Analysis Services de Microsoft »	Agrégation personnalisée	✓	~		~ (Instance)		~	~

En ce qui concerne l'agrégation des mesures à partir d'un niveau spécifique (agrégation contrainte), elle est la caractéristique la moins abordée dans l'état de l'art où il n'y a que [Salehi, 2009] qui la traite correctement sans demander au concepteur de prédéfinir toutes les combinaisons d'agrégations possibles comme [Franconi & Kamble, 2004]. En revanche, [Camossi et al., 2006], [Bertinoa et al., 2009], [Prat et al., 2010], [Prat et al., 2011], [Boulil et al., 2011] ne permettent d'exprimer que des agrégations successives. Et « Analysis Services de Microsoft » définit des agrégations contraintes uniformes pour toutes les mesures.

2.4 CONTRIBUTIONS

Dans le contexte des systèmes décisionnels, nous pouvons résumer les propositions de cette thèse à trois niveaux :

- **Niveau conceptuel (chapitre 3)** : le modèle conceptuel que nous souhaitons proposer doit permettre aux concepteurs d'intégrer plusieurs fonctions d'agrégation pour la même mesure et de les changer en fonction des dimensions, des hiérarchies et des paramètres. De plus, il doit à la fois contrôler la validité du calcul des fonctions non-commutatives et permettre d'exécuter les fonctions commutatives dans n'importe quel ordre. Il doit supporter l'expression des contraintes du calcul servant à agréger les mesures à partir des niveaux spécifiques qui peuvent être différents de niveaux de base. En outre, le modèle devrait présenter les agrégations aux utilisateurs en gardant la lisibilité et la compréhensibilité même avec des agrégations complexes ;
- **Niveau logique (chapitre 4)** : il s'agit d'étudier les impacts de l'utilisation de plusieurs fonctions d'agrégation pour la même mesure sur le stockage de données, surtout les pré-agrégats. A notre connaissance, ces impacts ont été peu étudiés dans la littérature. Ils comprennent le changement du nombre de pré-agrégats possibles et les modifications des relations de calcul entre ces pré-agrégats ;
- **Les opérateurs OLAP (chapitre 5)** : ils doivent compléter le modèle conceptuel et les études au niveau logique par l'adaptation d'un langage algébrique d'interrogation de données multidimensionnelles. L'utilisation de plusieurs fonctions d'agrégation pour la même mesure et la non-commutativité entre les fonctions nécessitent des changements dans les définitions et/ou dans les mécanismes de fonctionnement internes de certains opérateurs OLAP. Nous souhaitons détailler ces changements de chaque opérateur en présentant leurs causes et leurs conséquences sur la requête qui effectue l'opération demandée.

Afin de montrer la faisabilité de notre proposition, nous avons développé un prototype, appelé *OLAP-Multi-Functions* (chapitre 6). Ce prototype permet de concevoir une BDM à multifonctions ainsi que de superviser les manipulations OLAP effectuées par un analyste.

3. CHAPITRE III : MODÈLE CONCEPTUEL MULTIDIMENSIONNEL MULTIFONCTIONS

3.1 INTRODUCTION

La modélisation conceptuelle multidimensionnelle consiste à décrire les concepts de la base de données multidimensionnelles selon une vision orientée décideur [Golfarelli et al., 2002], indépendamment des contraintes d'implantation logique ou physique. Cette modélisation facilite la compréhension des données disponibles pour l'analyste [Rizzi et al., 2006]. Un modèle multidimensionnel organise les données en fonction de sujets d'étude (faits) analysés selon différents axes (dimensions) [Kimball, 1996], [Abelló et al., 2001a], [Abelló et al., 2001b], [Ravat et al., 2001]. Ces dimensions sont composées des niveaux de granularités (paramètres) organisés en hiérarchies. Cette organisation permet une manipulation et une exploitation des données rapides, efficaces et performantes [Codd et al., 1993], [Kimball, 1996].

Notre objectif est d'ajouter de nouveaux concepts afin d'intégrer les fonctions d'agrégation dans la modélisation multidimensionnelle. Notre modèle doit être suffisamment expressif pour supporter tous les types d'agrégation possibles et pour contrôler la validité des calculs, tout en facilitant les prises de décisions.

Dans ce chapitre, nous détaillons nos propositions au niveau conceptuel en présentant notre modèle conceptuel multifonctions pour un magasin de données multidimensionnelles. Nous avons choisi d'appliquer nos propositions au niveau des magasins de données afin de faciliter l'interrogation et l'analyse des données.

Plan du chapitre. Dans cette section, nous présentons notre problématique et les apports de notre modèle. La deuxième section détaille des concepts fondamentaux pour notre modèle intégrant les fonctions d'agrégation avec leurs formalismes graphiques. La troisième section précise comment les analyses sont effectuées dans notre modèle.

3.1.1 Problématique

Généralement, les utilisateurs ont besoin d'analyser des données à des niveaux d'agrégation (paramètres) différents, ce qui est réalisé au moyen des opérateurs RollUp et DrillDown. Donc, afin d'assurer des agrégations correctes et flexibles, les agrégations devraient être suffisamment précises dans les modèles multidimensionnels.

L'agrégation des mesures a été traitée différemment et différents formalismes ont été présentés dans plusieurs propositions [Sapia et al., 1998], [Luján-Mora et al., 2006], [Ravat et al., 2008], [Pedersen T.B. & Jensen, 1999], [Golfarelli et al., 1998a], [Abelló et al., 2006], [Camossi et al., 2006], [Salehi, 2009], [Prat et al., 2011], [Boulil et al., 2011].

Notre objectif est de proposer un modèle conceptuel multidimensionnel qui pallie les limites des modèles existants. Notamment, notre modèle doit mettre en lumière les agrégations de toutes les mesures le plus clairement et simplement possible en permettant de :

- Associer à la même mesure plusieurs fonctions d'agrégation compatibles ;
- Définir les hiérarchies d'une manière explicite afin de pouvoir utiliser des agrégations différentes sur les parties communes à plusieurs hiérarchies ;
- Définir des attributs faibles afin de compléter la sémantique des paramètres et de permettre l'utilisation des fonctions d'agrégation non-standard qui prennent en entrée des valeurs autres que les mesures ;
- Changer les fonctions d'agrégation selon les dimensions, les hiérarchies et les paramètres ;
- Contrôler l'ordre d'exécution des fonctions non-commutatives ;
- Agréger les mesures à partir des niveaux de granularité spécifiques.

3.1.2 Notre proposition

Nous proposons un modèle conceptuel multidimensionnel multifonctions. Ce modèle décrit le magasin des données par un schéma en constellation [Kimball, 1996] qui se compose de plusieurs faits et leurs dimensions éventuellement partagées. Afin de distinguer les hiérarchies multiples au sein des dimensions, chaque hiérarchie a un nom identifiant. Chaque paramètre peut avoir un ou plusieurs attributs faibles. Notre modèle conceptuel intègre les fonctions d'agrégation en associant à chaque mesure un ensemble de fonctions compatibles.

Cette intégration de fonctions d'agrégation au niveau conceptuel permet aux concepteurs de fournir, aux utilisateurs finaux, des systèmes décisionnels conviviaux en simplifiant l'analyse par l'automatisation du choix des fonctions d'agrégation. Cette intégration contribue également à la qualité du système d'aide à la décision, qui elle-même affecte la qualité de la décision [Clark et al., 2007]. En effet, lors de l'agrégation des données, l'effet des erreurs dues aux agrégations peut être dévastateur, car ces erreurs peuvent s'accumuler et se propager à travers le processus d'agrégation [Parssian, 2006]. Dans notre proposition, nous ne contrôlons pas la qualité des données d'entrée (les données des faits et des dimensions fournies sont supposées être cohérentes). Cependant, par la représentation des fonctions d'agrégation explicitement, nous contrôlons l'agrégation des données, ce qui minimise le risque de production d'erreurs de calcul. En outre, un autre avantage de la spécification des fonctions d'agrégation dans le modèle conceptuel est d'utiliser ces fonctions pour le calcul des cubes, c'est-à-dire pour le pré-calcul des agrégats.

Nous distinguons dans notre modèle quatre types de fonction d'agrégations :

- **Fonction d'agrégation générale** : c'est une fonction utilisée uniformément sur toutes les dimensions (correspondant aux approches des modèles multidimensionnels classiques) ;
- **Fonction d'agrégation multiple dimensionnelle** : c'est une fonction utilisée sur une dimension ;
- **Fonction d'agrégation multiple hiérarchique** : c'est une fonction utilisée sur une hiérarchie ;
- **Fonction d'agrégation différenciée** : c'est une fonction utilisée entre deux paramètres.

Afin de contrôler les combinaisons de fonctions d'agrégation, nous associons à chaque fonction un ordre d'exécution. Une fonction peut être agrégée à partir d'un niveau de granularité spécifique indiqué par une contrainte d'agrégation liée à la fonction. Afin de faciliter la lisibilité, nous séparons la présentation des éléments structurels (faits, dimensions et hiérarchies)

de la présentation des fonctions d'agrégation. Nous distinguons pour chaque mesure une présentation spécifique des fonctions d'agrégation.

3.2 MODÈLE CONCEPTUEL DE DONNÉES MULTIFONCTIONS

Dans cette section, nous définissons les différents concepts et formalismes graphiques de notre modèle conceptuel pour modéliser les BDMs. Notre modèle se base sur les concepts de fait, de dimension, de hiérarchie, de fonction d'agrégation et de schéma multidimensionnel. Nous proposons pour chaque concept un formalisme graphique.

3.2.1 Fait

Un *fait* représente un sujet d'analyse. Chaque fait comprend un ou plusieurs indicateurs d'analyse appelés *mesures*. Généralement, les mesures sont numériques [Kimball, 1996]. Elles sont agrégées par des fonctions d'agrégation selon les niveaux de granularité (paramètres) des axes d'analyse choisis par l'analyste.

Soient

- $\mathcal{N} = \{n_1, n_2, ...\}$ un ensemble fini de noms non redondants,
- $F = \{F_1, ..., F_n\}$ un ensemble fini de faits, $n \geq 1$,
- $D = \{D_1, ..., D_m\}$ un ensemble fini de dimensions, $m \geq 2$.

Définition 1. Un *fait*, noté F_i, $\forall\ i \in [1..n]$, est défini par (n^{Fi}, M^{Fi}) où

- $n^{Fi} \in \mathcal{N}$ est le nom identifiant le fait,
- $M^{Fi} = \{m_1, ..., m_{pi}\}$ est un ensemble de *mesures*.

On pose M l'ensemble des mesures :

$$M = \bigcup_{i=1}^{n} M^{Fi}$$

Formalisme graphique. Dans notre modèle, les faits sont présentés par des rectangles verts comportant deux parties, le nom du fait est contenu dans la partie haute tandis que la partie basse contient l'ensemble des mesures (Figure 16).

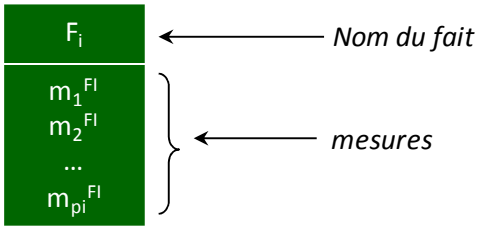


Figure 16 : Formalisme graphique d'un fait

Exemple 1. Dans notre exemple de météo, les analystes étudient les températures maximales, minimales et moyennes ainsi que le niveau des précipitations. Pour supporter ces analyses, nous établissons une BDM comportant deux faits 'Température' ayant trois mesures 'Tem_Moy', 'Tem_Max', 'Tem_Min' et 'Précipitation' ayant une mesure 'Précip' définis de la manière suivante :

- $F_{Température}$ = ('Température', {Tem_Moy, Tem_Max, Tem_Min}) ;
- $F_{Précipitation}$ = ('Précipitation', {Précip}).

La Figure 17 illustre la présentation graphique de ces faits.

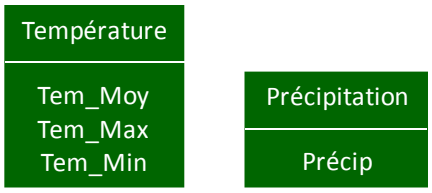


Figure 17 : Représentation graphique des faits 'Température' et 'Précipitation'

3.2.2 Dimension et hiérarchie

Une *dimension* représente un axe d'analyse selon lequel les mesures peuvent être analysées. Les dimensions fournissent des informations contextuelles aux faits [Pérez et al., 2008]. Chaque dimension comprend un ensemble d'*attributs* organisés dans des hiérarchies. Ces attributs modélisent les niveaux de granularité de la dimension.

Définition 2. Une *dimension*, notée D_i, $\forall$ i $\in$ [1..*m*], est définie par (n^{Di}, A^{Di}, H^{Di}) où

- $n^{Di} \in \mathcal{N}$ est le nom identifiant la dimension,
- $A^{Di} = \{a_1^{Di}, ..., a_{ri}^{Di}\} \cup \{Id^{Di}, All^{Di}\}$ est l'ensemble des *attributs de la dimension*,
- $H^{Di} = \{H_1^{Di}, ..., H_{si}^{Di}\}$ est un ensemble de *hiérarchies*.

On pose A l'ensemble des attributs et H l'ensemble des hiérarchies :

$$A = \bigcup_{i=1}^{m} A^{Di}$$

$$H = \bigcup_{i=1}^{m} H^{Di}$$

Au sein de chaque dimension, plusieurs hiérarchies peuvent être définies. Chaque hiérarchie comporte un ensemble d'attributs appelés *paramètres*. Les hiérarchies organisent ces paramètres de la graduation la plus fine (*paramètre racine* noté Id^{Di}) jusqu'à la graduation la plus générale (*paramètre extrémité* noté All^{Di}). Ainsi chaque hiérarchie représente un chemin de navigation valide sur un axe d'analyse en déterminant les niveaux de granularité auxquels les mesures peuvent être agrégées.

Définition 3. Une *hiérarchie*, notée H_j (notation abusive de H_j^{Di}, $\forall$ i $\in$ [1..*m*], $\forall$ j $\in$ [1..s_i]), est définie par (n^{Hj}, P^{Hj}, $\prec^{Hj}$, $Weak^{Hj}$) où

- $n^{Hj} \in \mathcal{N}$ est le nom identifiant la hiérarchie,
- $P^{Hj} = \{p_1^{Hj}, ..., p_{qj}^{Hj}\}$ est un ensemble d'attributs de la dimension appelés *paramètres*, $P^{Hj} \subseteq A^{Di}$,
- $\prec^{Hj} = \{(p_x^{Hj}, p_y^{Hj}) \mid p_x^{Hj} \in P^{Hj} \wedge p_y^{Hj} \in P^{Hj}\}$ est une relation binaire antisymétrique et transitive,
- $Weak^{Hj} : P^{Hj} \rightarrow 2^{A^{Di} \setminus P^{Hj}}$ est une application qui associe à chaque paramètre un ensemble d'attributs de dimension, appelés *attributs faibles* (2^E représente toute combinaison de l'ensemble E).

L'antisymétrie signifie que $(p_{k_1}^{H_j} \prec^{Hj} p_{k_2}^{H_j}) \wedge (p_{k_2}^{H_j} \prec^{Hj} p_{k_1}^{H_j}) \Rightarrow p_{k_1}^{H_j} = p_{k_2}^{H_j}$ tandis que la transitivité signifie que $(p_{k_1}^{H_j} \prec^{Hj} p_{k_2}^{H_j}) \wedge (p_{k_2}^{H_j} \prec^{Hj} p_{k_3}^{H_j}) \Rightarrow p_{k_1}^{H_j} \prec^{Hj} p_{k_3}^{H_j}$.

On pose P^{D_i} l'ensemble des paramètres d'une dimension D_i et P l'ensemble des paramètres de toutes les dimensions :

$$P^{D_i} = \bigcup_{j=1}^{s_i} P^{H_j}$$

$$P = \bigcup_{i=1}^{m} P^{D_i} = \bigcup_{i=1}^{m} \bigcup_{j=1}^{s_i} P^{H_j}$$

Lemme 1. Pour chaque dimension D_i, un *paramètre racine*, noté $Id^{Di} \in P^{D_i}$, existe. Il est défini comme suit. $\forall j \in [1..s_i]$, $\forall p_k^{H_j} \in P^{D_i}$, $Id^{Di} \neq p_k^{H_j} \Rightarrow Id^{Di} \prec^{Hj} p_k^{H_j}$.

Lemme 2. Pour chaque dimension D_i, un *paramètre extrémité*, noté $All^{Di} \in P^{D_i}$, existe. Il est défini comme suit. $\forall j \in [1..s_i]$, $\forall p_k^{H_j} \in P^{D_i}$, $All^{Di} \neq p_k^{H_j} \Rightarrow p_k^{H_j} \prec^{Hj} All^{Di}$.

On pose $\prec^{Di}$ est une relation qui généralise $\prec^{Hj}$ en définissant l'ordre des paramètres P^{D_i} dans la dimension D_i :

$$\prec^{Di} = \bigcup_{j=1}^{s_i} \prec^{Hj}$$

On pose W^{D_i} l'ensemble des attributs faibles d'une dimension D_i et W l'ensemble des attributs faibles de toutes les dimensions :

$$W^{D_i} = \bigcup_{\forall j \in [1..s_i], \forall k \in [1..q_j]} \text{Weak}^{Hj}(p_k^{H_j})$$

$$W = \bigcup_{i=1}^{m} W^{D_i} = \bigcup_{i=1}^{m} \bigcup_{\forall j \in [1..s_i], \forall k \in [1..q_j]} \text{Weak}^{Hj}(p_k^{H_j})$$

Lemme 3. Pour chaque dimension D_i, ses attributs de dimension sont de manière exclusive soit des paramètres, soit des attributs faibles :

$$P^{D_i} \cap W^{D_i} = \emptyset \wedge P^{D_i} \cup W^{D_i} = A^{D_i}$$

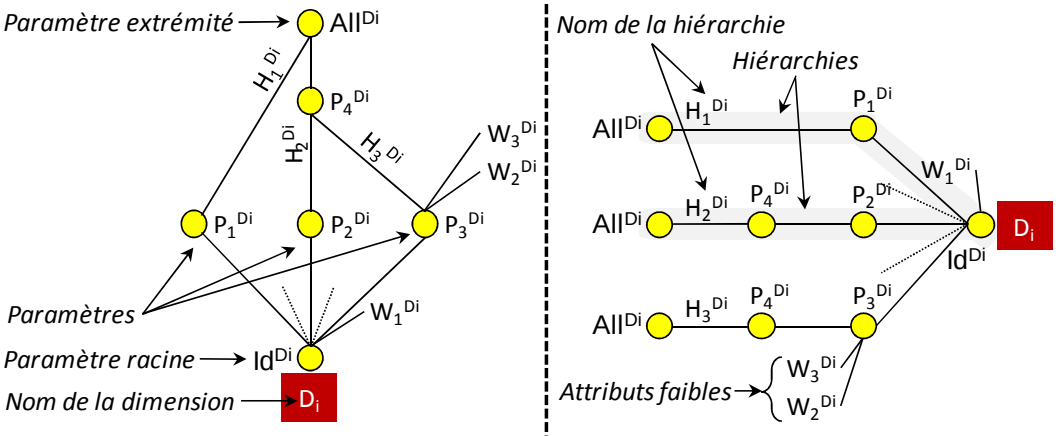
Formalisme graphique. Afin de présenter graphiquement les dimensions avec leurs hiérarchies, nous proposons un formalisme graphique inspiré de [Golfarelli et al., 1998a], [Ravat et al., 2008]. La dimension est représentée par un rectangle rouge comportant son nom (Figure 18). Chaque paramètre est représenté par un cercle jaune étiqueté par son nom ; chaque attribut faible est représenté par un segment de droite portant un nom et attaché au paramètre qu'il décrit. Les paramètres sont organisés selon la relation $\prec^{Hj}$ en une ou plusieurs hiérarchies. Les hiérarchies sont nommées. Elles peuvent être présentées en deux versions :

- **Compacte** où chaque hiérarchie est représentée par un chemin dans un treillis (Figure 18 (a)). Les nœuds du treillis sont l'ensemble des paramètres de la

dimension (P^{Di}) tandis que les arcs sont définis selon la relation ($\prec^{Di}$). Le nœud racine (borne inférieure) du treillis est le paramètre racine de la dimension (Id^{Di}) et le nœud final (borne supérieure) est le paramètre extrémité (All^{Di}) ;

- **Divisée** où les hiérarchies sont présentées séparément (Figure 18 (b)). Les parties communes à plusieurs hiérarchies sont répétées dans chaque hiérarchie à l'exception de la racine (Id^{Di}) qui reste le seul paramètre commun. Par exemple, dans la Figure 18 (b), le paramètre (P_4^{Di}) est répété dans les hiérarchies (H_2^{Di} et H_3^{Di}).

Il est nécessaire de mentionner que le paramètre racine (Id^{Di}) peut être nommé différemment par le concepteur : il prend généralement le nom d'un paramètre identifiant de la dimension.



(a) : version compacte (b) : version divisée

Figure 18 : Formalisme graphique d'une dimension et de ses hiérarchies[8]

Exemple 2. Dans notre exemple de météo, les analystes étudient les températures et les précipitations en fonction d'informations géographiques (dimension 'Géographie') et temporelles (dimensions 'Temps' et 'Dates'). Comme nous avons présenté précédemment, nous avons besoin de deux hiérarchies sur la dimension 'Géographie' ('Hgéo_scien' et 'Hgéo_simp') afin de prendre en compte les deux façons d'observer les données : simple et scientifique (*cf.* § 1.3.1). Les deux hiérarchies doivent organiser les subdivisions dans le système géographique administratif français (villes, départements, régions). Les dimensions 'Temps' avec une seule hiérarchie ('HTemps') et 'Dates' avec trois hiérarchies ('Hmois', 'Hsemaines' et 'Hsaisons') ordonnent les niveaux de granularité : horaires et des jours respectivement. Ces dimensions sont décrites formellement comme suit :

- $D_{Géographie}$ = ('*Géographie*', {a_{Ville}, $a_{Niveau\_Administratif}$, $a_{Département}$, $a_{D\_Superficie}$, $a_{Région}$, $a_{R\_Superficie}$, $All^{Géographie}$}, {$H_{Hgéo\_scien}$, $H_{Hgéo\_simp}$}) avec
  - $H_{Hgéo\_scien}$ = ('*Hgéo_scien*', {a_{Ville}, $a_{Département}$, $a_{Région}$, $All^{Géographie}$}, {(a_{Ville}, $a_{Département}$), ($a_{Département}$, $a_{Région}$), ($a_{Région}$, $All^{Géographie}$)}, {($a_{Département}$, {$a_{D\_Superficie}$}), ($a_{Région}$, {$a_{R\_Superficie}$})}), et

[8] Les lignes en pointillé indiquent que d'autres hiérarchies peuvent exister.

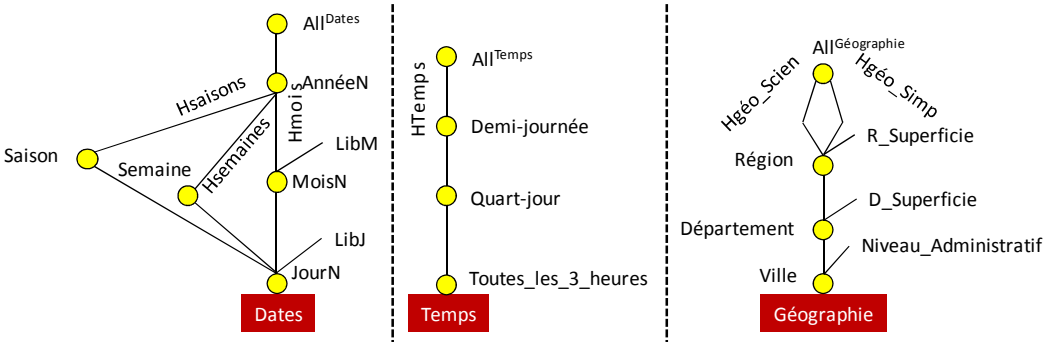
- $H_{Hgéo\_simp}$ = (‘*Hgéo_simp*’, {a_{Ville}, $a_{Département}$, $a_{Région}$, $All^{Géographie}$}, {(a_{Ville}, $a_{Département}$), ($a_{Département}$, $a_{Région}$), ($a_{Région}$, $All^{Géographie}$)}, {(a_{Ville}, {$a_{Niveau\_Administratif}$})}).

– D_{Temps} = (‘*Temps*’, {$a_{Toutes\_les\_3\_heures}$, $a_{quart\text{-}jour}$, $a_{demi\text{-}journée}$, All^{Temps}}, {H_{HTemps}}) avec
- H_{HTemps} = (‘*HTemps*’, {$a_{Toutes\_les\_3\_heures}$, $a_{quart\text{-}jour}$, $a_{demi\text{-}journée}$, All^{Temps}}, {($a_{Toutes\_les\_3\_heures}$, $a_{quart\text{-}jour}$), ($a_{quart\text{-}jour}$, $a_{demi\text{-}journée}$), ($a_{demi\text{-}journée}$, All^{Temps})}).

– D_{Dates} = (‘*Dates*’, {a_{JourN}, a_{LibJ}, a_{MoisN}, a_{LibM}, $a_{Semaine}$, a_{Saison}, $a_{AnnéeN}$, All^{Dates}}, {H_{Hmois}, $H_{Hsemaines}$, $H_{Hsaisons}$}) avec
- H_{Hmois} = (‘*Hmois*’, {a_{JourN}, a_{MoisN}, $a_{AnnéeN}$, All^{Dates}}, {(a_{JourN}, a_{MoisN}), (a_{MoisN}, $a_{AnnéeN}$), ($a_{AnnéeN}$, All^{Dates})}, {(a_{JourN}, {a_{LibJ}}), (a_{MoisN}, {a_{LibM}})}),
- $H_{Hsemaines}$ = (‘*Hsemaines*’, {a_{JourN}, $a_{Semaine}$, $a_{AnnéeN}$, All^{Dates}}, {(a_{JourN}, $a_{Semaine}$), ($a_{Semaine}$, $a_{AnnéeN}$), ($a_{AnnéeN}$, All^{Dates})}, {(a_{JourN}, {a_{LibJ}})}), et
- $H_{Hsaisons}$ = (‘*Hsaisons*’, {a_{JourN}, a_{Saison}, $a_{AnnéeN}$, All^{Dates}}, {(a_{JourN}, a_{Saison}), (a_{Saison}, $a_{AnnéeN}$), ($a_{AnnéeN}$, All^{Dates})}, {(a_{JourN}, {a_{LibJ}})}).

Les paramètres identifiants (les paramètres racines) des dimensions ‘Géographie’ notée $D_{Géographie}$, ‘Temps’ notée D_{Temps} et ‘Dates’ notée D_{Dates} sont respectivement les paramètres : ‘Ville’ noté a_{Ville}, ‘Toutes_les_3_heures’ noté $a_{Toutes\_les\_3\_heures}$ et ‘JourN’ noté a_{JourN}. Sur la hiérarchie ‘Hgéo_simp’ notée $H_{Hgéo\_simp}$ de la dimension ‘Géographie’, chaque ville est associée à son niveau administratif (ville standard, préfecture, capitale régionale ou capitale du pays) en utilisant l’attribut faible ‘Niveau_administratif’. Cette association est nécessaire afin de réaliser l’analyse simple (É3). Tandis que sur la hiérarchie ‘Hgéo_scien’, les départements et les régions sont liés à leurs superficies : les attributs faibles ‘D_Superficie’ et ‘R_Superficie’ respectivement parce que l’analyse scientifique utilise ces superficies pour calculer des moyennes pondérées (É5) et (É6).

La Figure 19 illustre la présentation graphique compacte des dimensions ‘Dates’ (a), ‘Temps’ (b) et ‘Géographie’ (c).



(a) : dimension ‘Dates’ (b) : dimension ‘Temps’ (c) : dimension ‘Géographie’

Figure 19 : Représentation graphique des dimensions ‘Dates’, ‘Temps’ et ‘Géographie’

Les concepts définis précédemment (fait, dimension, hiérarchie) représentent les éléments structurels dans un modèle des données multidimensionnelles.

3.2.3 Fonction d’agrégation

En utilisant les fonctions d’agrégation, les mesures sont agrégées selon des paramètres sélectionnés par les analystes. Les fonctions d’agrégation doivent être compatibles et peuvent changer avec les mesures, les dimensions, les hiérarchies et les paramètres. Elles peuvent être commutatives ou non entre elles.

Définition 4. Une *fonction d'agrégation*, notée f_t, est définie par $T^{f_t}(X^{f_t})$ où

- $T^{f_t} \in$ {AVG, SUM, MAX, …} est le nom de la fonction,
- $X^{f_t} = (x_1^{f_t}, x_2^{f_t}, \ldots, x_v^{f_t})$ est un ensemble ordonné d'arguments[9] dont au moins une mesure, $X^{f_t} \subseteq A \cup M, \exists i \in [1..v] \mid x_i^{f_t} \in M$.

On note $\mathcal{F}$ l'ensemble des fonctions d'agrégation :

$$\mathcal{F} = \bigcup_t f_t$$

Nous distinguons quatre types de fonction d'agrégations :

- **Fonction d'agrégation générale** : c'est la fonction que l'on utilise pour agréger les valeurs d'une mesure entre n'importe quel paramètre. Cette fonction n'est associée qu'à la mesure, sans prendre en compte ni les paramètres, ni les hiérarchies, ni les dimensions. Cette fonction représente la fonction d'agrégation dans les modèles qui agrègent une mesure uniformément dans tout l'espace multidimensionnel ;
- **Fonction d'agrégation multiple dimensionnelle** : c'est la fonction que l'on utilise pour agréger les valeurs d'une mesure entre tous les paramètres d'une même dimension. Cette fonction est associée à une mesure et à une dimension. Il est possible d'associer à une même mesure, plusieurs fonctions d'agrégation différentes selon les dimensions. Autrement dit, la fonction d'agrégation peut changer d'une dimension à l'autre ;
- **Fonction d'agrégation multiple hiérarchique** : c'est la fonction que l'on utilise pour agréger les valeurs d'une mesure entre les paramètres sur une hiérarchie. Cette fonction est associée à une mesure et à une hiérarchie. Il est possible d'associer à une même mesure, plusieurs fonctions d'agrégation, une pour chaque hiérarchie. La fonction d'agrégation peut ainsi changer d'une hiérarchie à l'autre en définissant des agrégations différentes pour les mêmes paramètres qui appartiennent à différentes hiérarchies ;
- **Fonction d'agrégation différenciée** : c'est la fonction que l'on utilise pour agréger les valeurs d'une mesure entre deux paramètres d'une hiérarchie. Elle est associée à une mesure et à un paramètre. Il est possible d'associer à une même mesure, plusieurs fonctions d'agrégation, une pour chaque paramètre. Autrement dit, cette fonction donne la possibilité d'appliquer une agrégation de manière spécifique à chaque niveau de granularité.

D'un côté, il est possible d'avoir plusieurs fonctions d'agrégation différentes sur les dimensions considérées durant une analyse. Ces fonctions sont souvent non-commutatives. Il faut donc pour contrôler la validité des résultats imposer un **ordre d'exécution**. C'est pourquoi nous associons à chaque fonction d'agrégation un nombre représentant l'ordre selon lequel la fonction concernée va être exécutée par rapport aux autres fonctions d'agrégation des autres dimensions impliquées dans l'analyse.

[9] Nous utilisons le terme (arguments) au lieu de (paramètres) afin d'éviter la confusion avec l'autre terme (paramètre) déjà utilisé pour indiquer les niveaux de granularité.

D'un autre côté, les agrégations ne s'effectuent pas toutes nécessairement de manière uniforme à partir de tous les niveaux inférieurs (contrairement au mécanisme d'agrégation prévu dans la plupart des modèles multidimensionnels existants). Par conséquent, nous introduisons un mécanisme de contrainte sur l'agrégation pour fixer le niveau d'agrégation valide permettant d'obtenir une agrégation supérieure. Ces **contraintes d'agrégation** sont réalisées en associant à chaque fonction d'agrégation un nombre négatif qui indique le niveau inférieur valide à partir duquel l'agrégation considérée doit se calculer.

Formalisme graphique. Dans notre modèle, les fonctions d'agrégation sont modélisées par des losanges bleus accompagnés de leurs noms. Le nom de la fonction est suivi par la liste des arguments qui est suivie à son tour par un nombre négatif représentant la contrainte d'agrégation possible. Chaque losange contient l'ordre d'exécution (Figure 20).

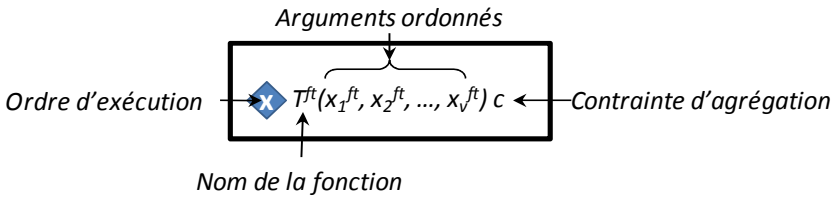
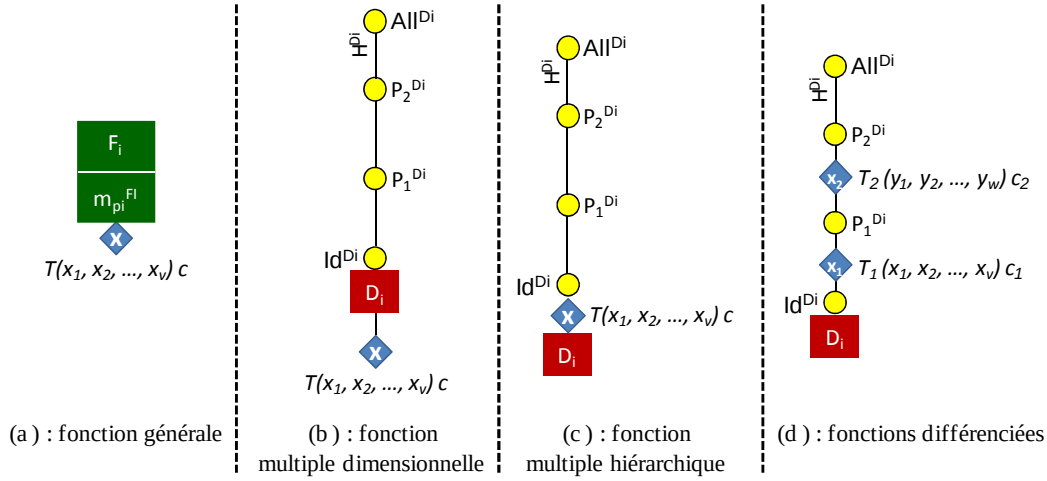


Figure 20 : Formalisme graphique d'une fonction d'agrégation

Nous utilisons le même symbole (losange) pour les quatre types de fonctions d'agrégation (générale, multiple dimensionnelle, multiple hiérarchique et différenciée) pour ne pas surcharger le schéma. Les positions des losanges dépendent du type de fonction :

- La fonction générale est représentée par un losange sur le bord du fait (Figure 21 (a)) ;
- La fonction d'agrégation multiple dimensionnelle est localisée sur un arc reliant à la dimension (Figure 21 (b)) ;
- La fonction d'agrégation multiple hiérarchique est localisée en bas de la hiérarchie (Figure 21 (c)) ;
- La fonction d'agrégation différenciée étiquette l'arc reliant deux paramètres (Figure 21 (d)).



(a) : fonction générale | (b) : fonction multiple dimensionnelle | (c) : fonction multiple hiérarchique | (d) : fonctions différenciées

Figure 21 : Formalisme graphique des types de fonctions d'agrégation

3.2.4 Schéma multidimensionnel

Nous modélisons une base de données multidimensionnelles (BDM) par une constellation qui généralise le modèle en étoile [Kimball, 1996] comportant un fait unique. Une constellation permet de modéliser plusieurs faits analysés en fonction de plusieurs dimensions éventuellement partagées. Le regroupement de plusieurs faits dans le même schéma multidimensionnel facilite la corrélation des analyses [Ghozzi, 2004].

Nous enrichissons la définition de la constellation par une fonction qui relie chaque mesure aux fonctions d'agrégation compatibles ; elle décrit comment la mesure est agrégée dans l'espace multidimensionnel.

Définition 5. Un *schéma multidimensionnel*, noté S, est défini par (F, D, Star, Aggregate) où

- $F = \{F_1,..., F_n\}$ est l'ensemble des faits, si $|F|=1$ alors le schéma multidimensionnel est appelé schéma en étoile alors que si $|F|>1$ alors le schéma est appelé schéma en constellation ;
- $D = \{D_1,..., D_m\}$ est l'ensemble des dimensions ;
- Star : $F \rightarrow 2^D$ est une fonction qui associe chaque fait à un ensemble de dimensions en fonction desquelles il peut être analysé ;
- Aggregate : $M \rightarrow 2^{\mathbb{N}^* \times \mathcal{F} \times 2^D \times 2^H \times 2^P \times \mathbb{N}^-}$ associe chaque mesure à un ensemble de fonctions d'agrégation. Elle permet de définir les différents types de fonctions d'agrégation supportés par notre modèle (générale, multiple dimensionnelle, multiple hiérarchique, différenciée) :
  - Si 2^D, 2^H et 2^P ne sont pas utilisés ($2^D = \emptyset$, $2^H = \emptyset$ et $2^P = \emptyset$), alors la fonction est une fonction d'agrégation générale. On l'utilise pour simplifier la représentation graphique au lieu d'un usage répété d'une fonction multiple dimensionnelle pour plusieurs dimensions,
  - Si 2^H et 2^P ne sont pas utilisés ($2^D \neq \emptyset$, $2^H = \emptyset$ et $2^P = \emptyset$), alors la fonction est une fonction d'agrégation multiple dimensionnelle utilisée pour agréger la mesure sur toute la dimension considérée. On l'utilise pour simplifier la représentation graphique au lieu d'un usage répété d'une fonction multiple hiérarchique pour plusieurs hiérarchies,
  - Si 2^P seul n'est pas utilisé ($2^D \neq \emptyset$, $2^H \neq \emptyset$ et $2^P = \emptyset$), alors la fonction est une fonction d'agrégation multiple hiérarchique utilisée pour agréger la mesure sur toute la hiérarchie considérée. On l'utilise pour simplifier la représentation graphique au lieu d'un usage répété d'une fonction différentiée pour plusieurs paramètres,
  - Si 2^D, 2^H et 2^P sont tous utilisés ($2^D \neq \emptyset$, $2^H \neq \emptyset$ et $2^P \neq \emptyset$), alors la fonction est une fonction d'agrégation différenciée utilisée pour agréger la mesure entre le paramètre considéré et celui directement supérieur.

$\mathbb{N}^*$ associe à chaque fonction d'agrégation un numéro d'ordre qui représente la priorité dans l'exécution. La fonction d'agrégation avec l'ordre le plus petit a la priorité la plus élevée. Si les fonctions d'agrégation sont commutatives, alors elles sont du même ordre.

$\mathbb{N}^-$ sert à contraindre une agrégation en indiquant un niveau d'agrégation spécifique à partir duquel l'agrégation considérée doit se calculer. Une agrégation non contrainte sera associée à 0 tandis qu'une agrégation contrainte sera associée à une valeur négative pour forcer le calcul à partir d'un niveau inférieur choisi par rapport au niveau considéré.

Le schéma multidimensionnel devrait supporter l'agrégation automatique, ce qui signifie que le processus d'analyse OLAP connait les fonctions d'agrégation à appliquer lors du calcul des agrégations aux différents niveau supérieurs [Rafanelli & Shoshani, 1990], [Lenz & Shoshani, 1997], [Pedersen T.B. et al., 2009].

Lemme 4. Les fonctions d'agrégation assurent la ***couverture*** du schéma multidimensionnel, c'est-à-dire qu'il ne doit pas exister de paramètres (niveaux d'agrégation) en fonction desquels nous ne connaissons pas l'agrégation à appliquer :

$$\forall i \in [1..n], \forall m_k \in M^{Fi}, \exists f \in \mathcal{F}, \exists x_1 \in \mathbb{N}^*, \exists x_2 \in \mathbb{N}^-,$$

$$\begin{cases} |(x_1, f, \{\}, \{\}, \{\}, x_2) \in Aggregate(m_k) \\ \forall D_j \in Star(F_i) | (x_1, f, \{D_j\}, \{\}, \{\}, x_2) \in Aggregate(m_k) \\ \forall H_s \in H^j | (x_1, f, \{D_j\}, \{H_s\}, \{\}, x_2) \in Aggregate(m_k) \\ \forall P_q \in P^s \setminus \{All^j\} | (x_1, f, \{D_j\}, \{H_s\}, \{P_q\}, x_2) \in Aggregate(m_k) \end{cases}$$

De manière moins formelle, la couverture du schéma est réalisée de plusieurs façons, par l'utilisation :

- d'une fonction d'agrégation générale,
- d'une fonction d'agrégation multiple dimensionnelle pour chaque dimension,
- d'une fonction d'agrégation multiple hiérarchique pour chaque hiérarchie,
- d'une fonction d'agrégation différenciée pour chaque niveau d'agrégation,
- par combinaison des fonctions d'agrégation multiples et différenciées.

Associé aux définitions formelles, nous introduisons un formalisme graphique facilitant la compréhension du schéma de la BDM. Ces représentations graphiques sont de deux niveaux. Le premier présente le schéma structurel tandis le deuxième présente les schémas d'agrégation.

3.2.4.1 Schéma structurel

Le schéma structurel permet de visualiser en constellation les éléments de structure (faits, dimensions et hiérarchies) de la BDM en masquant les mécanismes d'agrégation (les fonctions d'agrégation, l'ordre d'exécution et les contraintes d'agrégation). Cette vue globale est obtenue à partir de la fonction ***Star***.

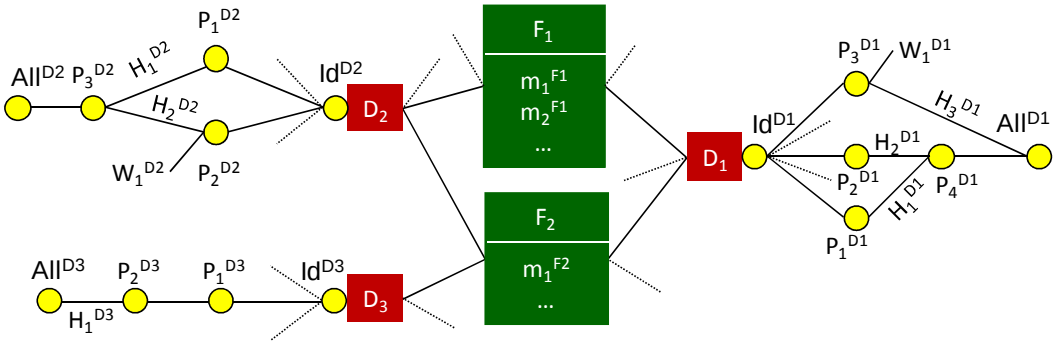


Figure 22 : Formalisme graphique d'un schéma structurel[10]

[10] Les lignes en pointillé indiquent que d'autres hiérarchies et d'autres liens entre les faits et les dimensions peuvent exister.

La constellation, inspirée de [Golfarelli et al., 1998a], [Ravat et al., 2008], se compose de tous les faits de la BDM et de l'ensemble des dimensions où chaque dimension peut être associée à un ou plusieurs faits. Le formalisme graphique du schéma structurel se représente conformément à la Figure 22. Les hiérarchies sont également représentées sous une forme compactée.

3.2.4.2 Schéma d'agrégation

Les schémas d'agrégation sont obtenus à partir de la fonction ***Aggregate***. Cette vision détaille les mécanismes d'agrégation (fonctions d'agrégation, ordre d'exécution et contraintes d'agrégation) impliqués dans le déroulement d'une analyse portant sur une mesure considérée, ceci en simplifiant les éléments structurels (faits, dimensions et hiérarchies) autant que possible.

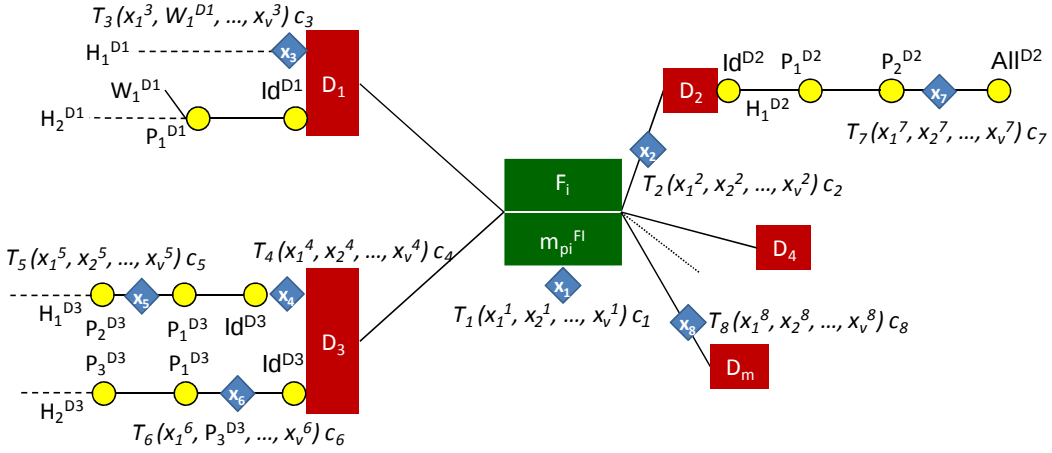


Figure 23 : Formalisme graphique d'un schéma d'agrégation[11]

Comme le montre le formalisme graphique du schéma d'agrégation, qui est illustré dans la Figure 23, un schéma d'agrégation présente plusieurs caractéristiques :

- **Un schéma par mesure** : pour chaque mesure $m_k \in F_i$, un schéma d'agrégation existe. Ce schéma montre les fonctions d'agrégation avec leur ordre d'exécution et leurs contraintes d'agrégation liés uniquement à la mesure concernée. Cette séparation entre les schémas d'agrégation des mesures différentes a pour but d'améliorer la lisibilité, de faciliter la compréhension et de ne pas surcharger le schéma de la BDM. Dans un schéma d'agrégation d'une mesure (m_1), si les fonctions d'agrégation utilisent une autre mesure (m_2) parmi leurs arguments, alors la mesure (m_1) est une mesure calculée (dérivée). C'est pourquoi il faut afficher tous les arguments de toutes les fonctions d'agrégation même s'il y a un seul argument qui est la mesure elle-même ;
- **Hiérarchies en version divisée** : les hiérarchies sont présentées en version divisée contrairement au schéma structurel (Figure 22) où elles sont présentées en version

[11] La ligne en pointillé indique que d'autres liens entre le fait et d'autres dimensions peuvent exister.

compacte. Dans une version divisée, les parties communes à plusieurs hiérarchies, même le paramètre racine (Id^{Di}), sont répétées dans chaque hiérarchie. Cette répétition a pour objectif d'éviter la confusion entre les fonctions d'agrégation sur les hiérarchies différentes ; par exemple, dans la Figure 23, la racine (Id^{D3}) est répétée dans les hiérarchies ($H_1^{D_3}$ et $H_2^{D_3}$) pour bien spécifier que la fonction multiple hiérarchique (T_4) est appliquée sur la hiérarchie ($H_1^{D_3}$) et elle n'est pas appliquée sur l'autre hiérarchie ($H_2^{D_3}$) ;

- **Simplification des éléments structurels** : les éléments structurels sont représentés dans une forme minimisée dans le schéma d'agrégation car ces éléments sont déjà présentés dans le schéma structurel. Cette simplification vise à réduire la complexité du schéma d'agrégation. Nous appliquons cette simplification à tous les niveaux :
  - **Simplification des attributs faibles** : le schéma d'agrégation ne présente que les attributs faibles utilisés pour agréger la mesure considérée. Autrement dit, les attributs faibles ne sont pas affichés dans le schéma d'agrégation sauf s'ils sont présents parmi les entrées (arguments) des fonctions d'agrégation. Par exemple, dans la Figure 23, tous les attributs faibles sont cachés sauf ($W_1^{D_1}$) qui est parmi les arguments de la fonction (T_3),
  - **Simplification des paramètres** : sur les hiérarchies qui ont des paramètres associés aux fonctions d'agrégation différenciées, il n'est pas nécessaire d'afficher les paramètres entre le niveau directement supérieur à la dernière fonction différenciée et le niveau (All^{Di}). Ces paramètres peuvent être simplifiés par une ligne en tiret. Par exemple, dans la Figure 23, les paramètres entre le paramètre ($P_2^{D_3}$) directement supérieur à la fonction différenciée (T_5) et le niveau (All^{D3}) sur la hiérarchie ($H_1^{D_3}$), sont simplifiés. Par contre, si un paramètre (ou un attribut faible associé à un paramètre), parmi les paramètres à simplifier, est utilisé en entrée d'une fonction d'agrégation, alors il faut l'afficher (avec l'attribut faible) ainsi que tous les paramètres inférieurs. Par exemple, le paramètre ($P_3^{D_3}$) sur la hiérarchie ($H_2^{D_3}$), qui un argument de la fonction (T_6), est affiché et les paramètres supérieurs sont simplifiés. Par ailleurs, si une fonction d'agrégation différenciée est associée au paramètre directement inferieur au paramètre extrémité (All^{Di}), alors tous les paramètres de la hiérarchie concernée seront affichés. Par exemple, tous les paramètres de la hiérarchie ($H_1^{D_2}$) sont présentés grâce à la fonction (T_8) qui est située entre le paramètre ($P_2^{D_2}$) et le niveau (All^{D2}),
  - **Simplification des hiérarchies** : si une hiérarchie est associée à une fonction multiple hiérarchique et aucun de ses paramètres n'est associé à une fonction différenciée, alors la hiérarchie peut être simplifiée par une ligne en tiret. Par exemple, dans la Figure 23, la hiérarchie ($H_1^{D_1}$) est simplifiée parce qu'elle n'a qu'une seule fonction d'agrégation (T_3) qui est multiple hiérarchique. En outre, si une hiérarchie ne possède aucune fonction d'agrégation, alors elle peut être cachée. Par contre, sur une hiérarchie, il faut présenter les paramètres et les attributs faibles utilisés comme arguments des fonctions d'agrégation même si la hiérarchie n'a aucune fonction d'agrégation. Par exemple, l'attribut faible ($W_1^{D_1}$) est affiché avec son paramètre ($P_1^{D_1}$) et tous les paramètres inférieurs parce qu'il est un argument de la fonction (T_3) bien que la hiérarchie ($H_2^{D_1}$) ne soit associée à aucune fonction d'agrégation,
  - **Simplification des dimensions** : si une dimension est associée à une fonction d'agrégation multiple dimensionnelle sans avoir aucune fonction multiple

hiérarchique et aucune fonction différenciée sur ses hiérarchies et ses paramètres, alors la dimension peut être simplifiée en présentant seulement son nom. Par exemple, la dimension (D_m) dans la Figure 23.

- **Présentation des dimensions sans fonctions d'agrégation** : dans un schéma d'agrégation, il faut montrer toutes les dimensions (au moins leurs noms) selon lesquelles la mesure considérée peut être analysée même si elles n'ont aucune fonction d'agrégation. Par exemple, la dimension (D_4) dans la Figure 23.

Exemple 3. Nous reprenons l'exemple de météo présenté précédemment (*cf.* § 1.3.1). Nous utilisons plusieurs fonctions d'agrégation pour agréger les précipitations et les températures moyennes, maximales et minimales. Les analystes veulent analyser les précipitations selon la géographie et la date tandis qu'ils souhaitent analyser les températures selon le temps, la date et la géographie. Nous définissons formellement une telle BDM par (F, D, Star, Aggregate) où :

- $F = \{F_{Température}, F_{Précipitation}\}$
- $D = \{D_{Géographie}, D_{Temps}, D_{Dates}\}$
- $Star = F \rightarrow 2^D$ |
  $Star(F_{Température}) = \{D_{Géographie}, D_{Dates}, D_{Temps}\}$
  $Star(F_{Précipitation}) = \{D_{Géographie}, D_{Dates}\}$
- $Aggregate = M \rightarrow 2^{N^* \times F \times 2^D \times 2^H \times 2^P \times N^-}$ |

Aggregate(Précip) = {
(1, SUM(Précip), {Dates}, {}, {}, 0)[12],
(2, AVG(Précip), {Dates}, {Hsaisons}, {AnnéeN}, -1)[13],
(2, AVG(Précip), {Dates}, {Hsemaines}, {AnnéeN}, -1),
(2, AVG(Précip), {Dates}, {MoisN}, {AnnéeN}, -1),
(1, SELECT_CENTER(Niveau_Administratif, Précip), {Géographie}, {Hgéo_simp}, {}, 0),
(2, AVG(Précip), {Géographie}, {Hgéo_Scien}, {Ville}, 0),
(3, AVG_W(Précip, D_Superficie), {Géographie}, {Hgéo_Scien}, {Département}, -1),
(3, AVG_W(Précip, R_Superficie), {Géographie}, {Hgéo_Scien}, {Région}, -1)}

Aggregate(Tem_Moy) = {
(2, AVG(Tem_Moy), {}, {}, {}, 0),
(2, SELECT_CENTER(Niveau_Administratif, Tem_Moy), {Géographie}, {Hgéo_simp}, {}, 0),
(1, AVG(Tem_Moy), {Géographie}, {Hgéo_Scien}, {Ville}, -1),
(1, AVG_W(Tem_Moy, D_Superficie), {Géographie}, {Hgéo_Scien}, {Département}, -1),
(1, AVG_W(Tem_Moy, R_Superficie), {Géographie}, {Hgéo_Scien}, {Région}, -1)}

Aggregate(Tem_Min) = {
(1, MIN(Tem_Moy), {}, {}, {}, 0),

[12] Il n'y a pas de contrainte sur l'agrégation.
[13] Les valeurs sont agrégées à partir des valeurs agrégées au niveau directement inférieur.

(1, SELECT_CENTER(Niveau_Administratif, Tem_Moy), {Géographie},
{Hgéo_simp}, {}, 0)}

Aggregate(Tem_Max) = {
(1, MAX(Tem_Moy), {}, {}, {}, 0),
(1, SELECT_CENTER(Niveau_Administratif, Tem_Moy), {Géographie},
{Hgéo_simp}, {}, 0)}

La mesure 'Précip' n'a pas une fonction d'agrégation générale. Les fonctions d'agrégation assurent la *couverture* pour cette mesure sur les deux dimensions d'analyse possibles 'Géographie' et 'Dates' en utilisant :

- Une fonction d'agrégation multiple dimensionnelle (SUM) sur la dimension 'Dates',
- Trois fonctions d'agrégation différenciées (AVG) entre le paramètre 'AnnéeN' et le niveau 'All^{Dates}' sur les trois hiérarchies 'Hmois', 'Hsaisons' et 'Hsemaines' de la dimension 'Dates',
- Une fonction d'agrégation multiple hiérarchique (SELECT_CENTER) sur la hiérarchie 'Hgéo_simp' de la dimension 'Géographie',
- Des fonctions d'agrégation différenciées (AVG et AVG_W) pour tous les paramètres sur la deuxième hiérarchie 'Hgéo_Scien' de la dimension 'Géographie'.

En ce qui concerne les mesures 'Tem_Moy', 'Tem_Max' et 'Tem_Min' :

- Elles ont toutes des fonctions d'agrégation générales (AVG, MAX, MIN respectivement). C'est suffisant pour assurer une couverture complète,
- Elles sont agrégées d'une manière identique sur la hiérarchie 'Hgéo_simp' de la dimension 'Géographie' par la fonction (SELECT_CENTER),
- Sur la hiérarchie 'Hgéo_Scien', les mesures 'Tem_Max' et 'Tem_Min' n'ont pas de fonction d'agrégation spécifique contrairement à la mesure 'Tem_Moy' qui a une fonction d'agrégation différenciée (AVG ou AVG_W) pour chaque paramètre.

Ainsi, Aggregate(Min_Note) est comparable à Aggregate(Max_Note) sauf que la fonction MIN est utilisée au lieu de la fonction MAX.

Par ailleurs, l'agrégation des mesures 'Tem_Max' et 'Tem_Min' s'appuie sur l'agrégation de 'Tem_Moy'. Cela apparait à travers l'utilisation de la mesure 'Tem_Moy' dans les fonctions d'agrégation de 'Tem_Max' et 'Tem_Min'. Pour connaître la température maximale 'Tem_Max' ou minimale 'Tem_Min' d'un département ou d'une région pour une période de temps, on doit d'abord calculer les températures 'Tem_Moy' de ce département ou de cette région pour les plus petites divisions du temps (les niveaux les plus bas des dimensions 'Dates' et 'Temps' qui sont 'JourN' et 'Toutes_les_3_heures' respectivement), ensuite on détermine la température maximale ou minimale parmi les températures obtenues.

Les contraintes sur les fonctions d'agrégation différenciées associées aux niveaux de base, ne changent rien par rapport à l'agrégation des mesures. Les fonctions d'agrégation différenciées qui agrègent les mesures entre le niveau de base et le paramètre directement supérieur peuvent être contraintes ou non parce qu'il y a un seul niveau inférieur (le niveau de base même) et les mesures ne peuvent qu'être agrégées à partir de ce niveau-là. Par exemple, la fonction différenciée de la mesure 'Tem_Moy' associée au paramètre 'Ville' sur la hiérarchie 'Hgéo_Scien' est contrainte (contrainte fixée à -1), par contre la fonction équivalente de la mesure 'Précip' liée au même paramètre n'est pas contrainte (contrainte fixée à 0).

Associées aux définitions formelles, nous introduisons les représentations graphiques facilitant la compréhension du schéma de la BDM. Le schéma structurel de ces représentations graphiques est illustré dans la Figure 24 et les schémas d'agrégation dans la Figure 25.

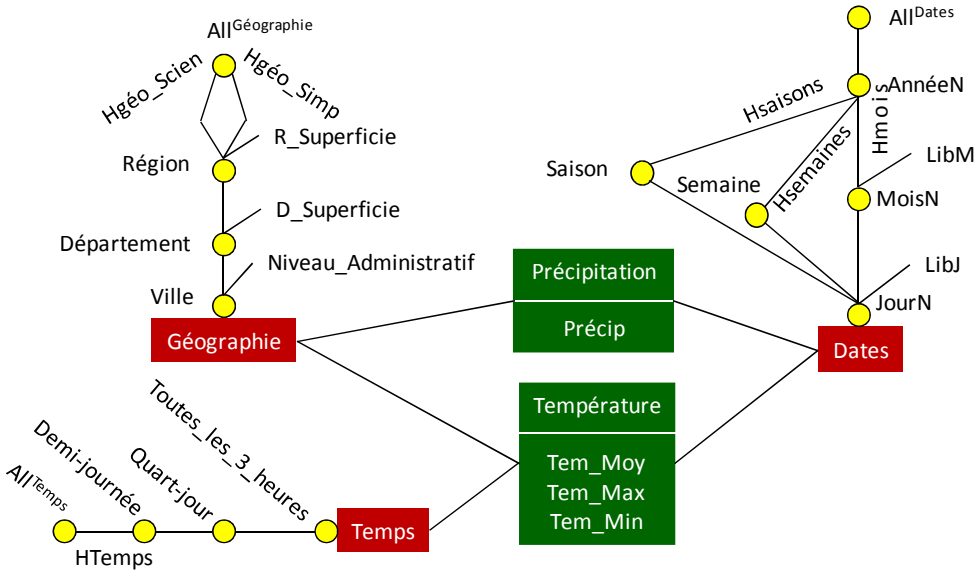


Figure 24 : Représentation graphique du schéma structurel de l'exemple de météo (Figure 6 répétée)

La Figure 24 présente la vue structurelle globale de la BDM de l'exemple de météo défini précédemment. Cette figure est identique à la Figure 6 représentant notre BDM selon [Ravat et al., 2008]. Elle présente également comment la fonction *Star* relie les faits 'Précipitation' et 'Température' (déjà présentés dans la Figure 17) aux dimensions 'Géographie', 'Dates' et 'Temps' (déjà présentées dans la Figure 19). Conformément aux définitions formelles, le fait 'Température' est lié à toutes les dimensions tandis que le fait 'Précipitation' est simplement associé aux deux dimensions 'Géographie' et 'Dates'.

La Figure 25 décrit les quatre schémas d'agrégation (a, b, c, d) correspondants aux quatre mesures 'Précip', 'Tem_Moy', 'Tem_Min' et 'Tem_Max'. Nous remarquons que les trois hiérarchies de la dimension 'Dates' (Figure 25 (a)) et les hiérarchies 'Hgéo_simp' et 'Hgéo_Scien' de la dimension 'Géographie' (Figure 25 (a et b)) sont présentées en version divisée contrairement à la version compacte à la Figure 24. Nous pouvons remarquer également la simplification des dimensions 'Temps' et 'Dates' et de la hiérarchie 'Hgéo_simp'. Les schémas d'agrégation masquent les hiérarchies qui n'ont aucune fonction d'agrégation. Par exemple, la hiérarchie 'Hgéo_Scien' est présentée dans la Figure 25 (a et b) où elle a des fonctions d'agrégation ; par contre elle est cachée dans la Figure 25 (c et d) où elle n'en a pas. La simplification des attributs faibles est appliquée en cachant les attributs faibles 'LibJ' et 'LibM' associés respectivement aux paramètres 'JourN' et 'MoisN'.

La Figure 25 présente les fonctions d'agrégation avec leur ordre d'exécution et leurs contraintes d'agrégation. La mesure 'Tem_Moy' est liée à la fonction d'agrégation générale (AVG) afin de réaliser l'opération (É2). Elle utilise la fonction multiple hiérarchique (SELECT_CENTER) sur la hiérarchie 'Hgéo_simp' pour effectuer l'opération (É3). Les trois fonctions différenciées (AVG et AVG_W) sur la hiérarchie 'Hgéo_Scien' exécutent les opérations (É4), (É5) et (É6). Les opérations (É7) et (É8) sur la mesure 'Précip' sont effectuées par la fonction multiple dimensionnelle (SUM) sur la dimension 'Dates' et les fonctions différenciées (AVG) associées au paramètre 'AnnéeN' sur les hiérarchies de la même dimension 'Dates' respectivement.

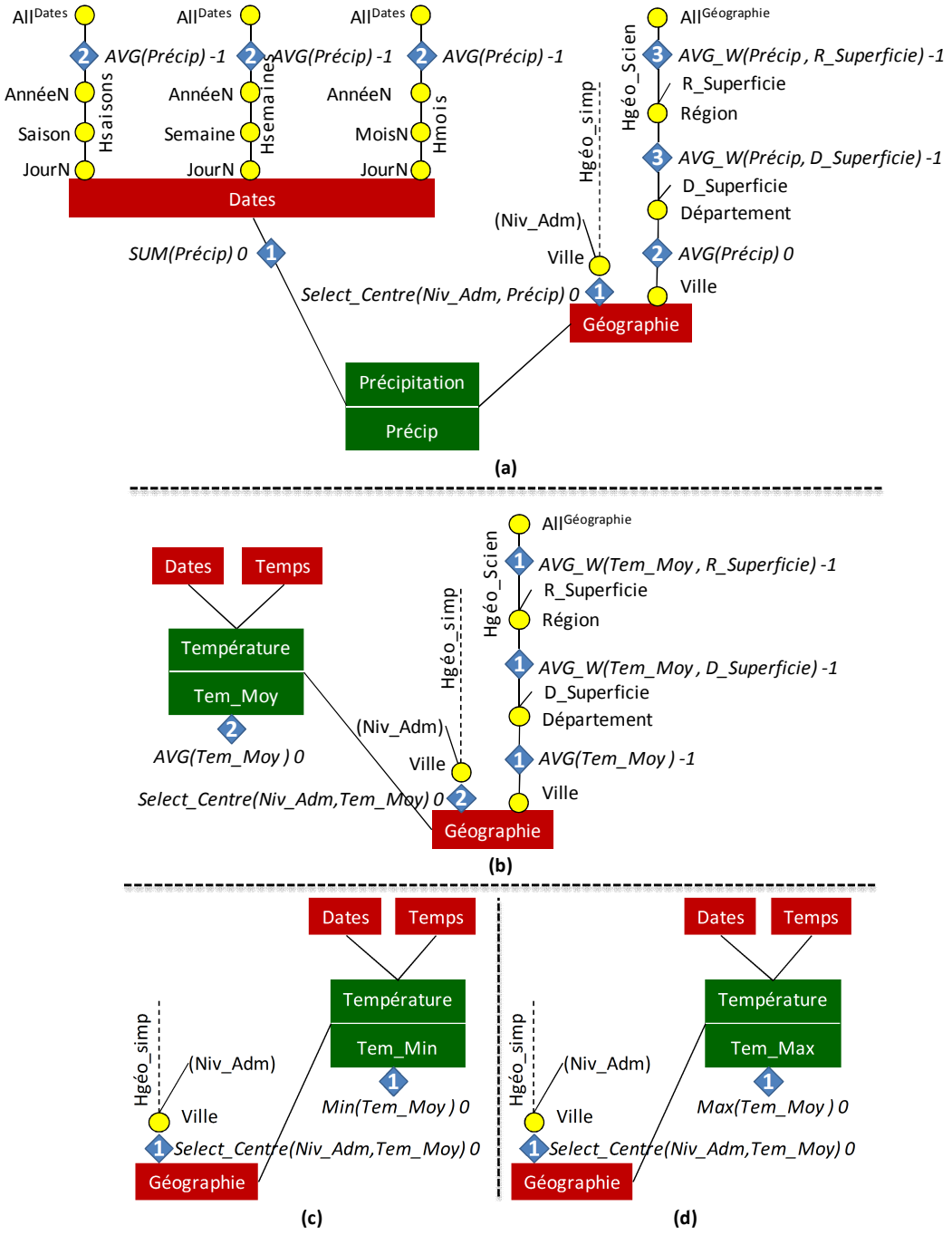


Figure 25 : Représentation graphique des schémas d'agrégation de l'exemple de météo[14]

[14] Ici nous utilisons l'abréviation (Niv_Adm) pour (Niveau_Administratif).

En ce qui concerne la commutativité dans l'ordre d'exécution, la fonction multiple hiérarchie (SELECT_CENTER) est commutative avec la fonction dimensionnelle (SUM) de la mesure 'Précip' (Figure 25 (a)) et toutes les fonctions générales : la fonction (AVG) de la mesure 'Tem_Moy' (Figure 25 (b)), la fonction (MIN) de la mesure 'Tem_Min' (Figure 25 (c)) et la fonction (MAX) de la mesure 'Tem_Max' (Figure 25 (d)).

Par ailleurs, dans notre exemple, toutes les agrégations contraintes sont calculées à partir du niveau directement inférieur (contrainte fixée à -1). Par exemple, la température moyenne par région est calculée à partir des températures par département. Dans l'hypothèse où nous aurions choisi de calculer cette température régionale à partir des températures par villes, la contrainte aurait été fixée à -2.

3.3 ANALYSE DANS LE MODÈLE MULTIFONCTIONS

Dans cette section, nous présentons comment les analyses sont effectuées dans notre modèle multifonctions. D'abord, nous mettons en lumière le rôle de l'ordre d'exécution dans l'analyse multifonctions et sa cohérence avec les contraintes d'agrégation. Ensuite, nous mettons l'accent sur la réalisation des analyses dans le contexte d'une BDM multifonctions.

3.3.1 Ordre d'exécution dans l'analyse multifonctions

Chaque fonction d'agrégation a un numéro d'ordre d'exécution. L'ordre croissant des valeurs de l'ordre d'exécution détermine l'ordre d'application des fonctions d'agrégation. Le choix d'un ordre valide dépend des besoins de l'utilisateur. Il peut différer d'un cas à l'autre, même si les fonctions sont les mêmes dans les deux cas. Ainsi, le modèle permet au concepteur de fixer l'ordre qui donne un résultat valide. Un système d'auto-vérification peut être utilisé afin de déterminer cet ordre et le valider dans des cas complexes. Un tel système ne se situe pas dans le cadre de nos recherches.

Dans le cas de **deux fonctions** d'agrégation différentes, si les fonctions sont commutatives, alors elles auront la même valeur d'ordre d'exécution, sinon la fonction ayant la valeur la plus petite est prioritaire. Par contre, dans le cas d'**une même fonction** d'agrégation qui peut être utilisée sur deux dimensions différentes (le même nom avec les mêmes arguments), l'ordre d'exécution peut définir trois calculs différents présentés dans la Figure 26 (a, b et c) qui présente trois schémas d'agrégation d'une mesure 'm' analysée selon deux dimensions 'D1' et 'D2' en utilisant la fonction (AVG).

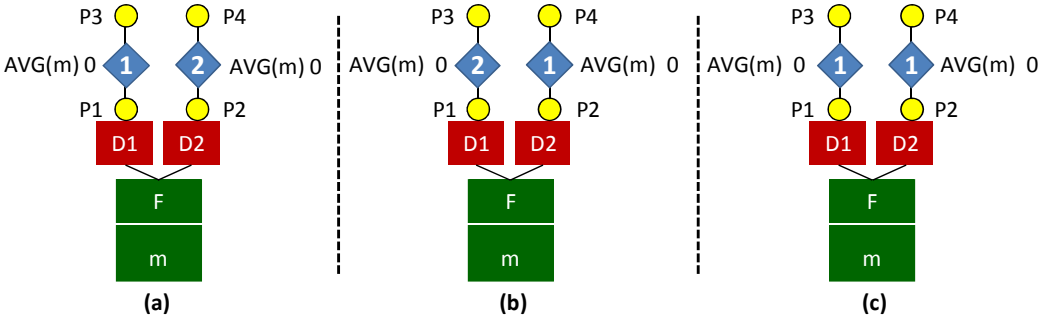


Figure 26 : Ordre d'exécution dans l'analyse multifonctions

Le premier calcul (Figure 26 (a)) : selon l'ordre d'exécution présenté, la moyenne AVG(m) doit être calculée d'abord sur la dimension 'D1' au niveau 'P3'. Ensuite il faut, à partir des valeurs résultantes, calculer la moyenne sur la dimension 'D2' au niveau 'P4'. Donc,

l'analyse de la mesure 'm' aux niveaux 'P3' et 'P4' peut être réalisée par la requête R_a suivante :

R_a :
```
SELECT AVG(m), P3, P4
FROM  ( SELECT AVG(m) AS m, P2, P3, P4
        FROM …
        WHERE …
        GROUP BY P2, P3, P4)
GROUP BY P3, P4
```

Le deuxième calcul (Figure 26 (b)) : il faut calculer les moyennes dans l'ordre inverse du premier calcul. La fonction AVG(m) est appliquée d'abord sur la dimension 'D2' pour le niveau 'P4'. Puis, elle est appliquée sur la dimension 'D1' pour le niveau 'P3' selon la requête R_b suivante :

R_b :
```
SELECT AVG(m), P3, P4
FROM  ( SELECT AVG(m) AS m, P1, P3, P4
        FROM …
        WHERE …
        GROUP BY P1, P3, P4)
GROUP BY P3, P4
```

Comme la fonction (AVG) n'est pas commutative avec elle-même, les résultats des requêtes R_a et R_b sont différents.

Le troisième calcul (Figure 26 (c)) : contrairement au cas de deux fonctions d'agrégation différentes, si la fonction a le même ordre d'exécution sur les deux dimensions, alors cela ne veut pas dire qu'elle est commutative avec elle-même mais signifie qu'il faut appliquer la fonction une seule fois pour les deux dimensions. Cette notion est mise en œuvre dans la requête R_c où une seule fonction (AVG) calcule la moyenne aux niveaux 'P3' et 'P4' à la fois.

R_c :
```
SELECT AVG(m), P3, P4
FROM …
WHERE …
GROUP BY P3, P4
```

Ce troisième calcul donne des résultats différents de ceux du premier et du deuxième calcul. Par contre, si la fonction utilisée est commutative avec elle-même (par exemple SUM), alors les résultats des trois calculs précédents sont identiques.

3.3.2 Cohérence entre les contraintes d'agrégation et l'ordre d'exécution

La notion de fonction d'agrégation est associée dans notre modèle à la contrainte d'agrégation et à l'ordre d'exécution. En utilisant les contraintes d'agrégation, nous pouvons surmonter le problème des agrégations à partir des niveaux spécifiques autres que les niveaux de base. Les contraintes d'agrégation servent à préciser pour chaque paramètre, le niveau d'agrégation à partir duquel l'agrégation considérée doit être calculée. L'agrégation à ce niveau peut être calculée à son tour par une autre fonction d'agrégation. Cela peut être considéré comme une sorte d'ordonnancement des fonctions d'agrégation entre les paramètres d'une hiérarchie. En outre, l'ordre d'exécution s'applique sur des fonctions d'agrégation situées sur des dimensions différentes. Donc, il ne doit pas y avoir de contradictions entre les contraintes d'agrégation et l'ordre d'exécution. Plus précisément, dans une analyse multifonctions, une

fonction d'agrégation ne doit pas s'appuyer sur les résultats d'une autre fonction d'agrégation ayant un ordre d'exécution supérieur.

La Figure 27 montre les différents cas de cohérence et d'incohérence entre les contraintes d'agrégation et l'ordre d'exécution.

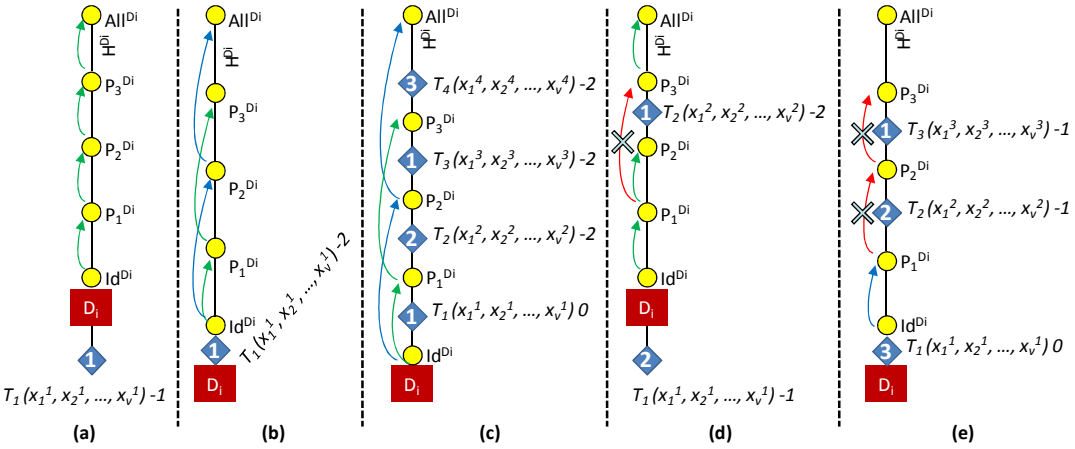


Figure 27 : Principes de cohérence entre les contraintes d'agrégation et l'ordre d'exécution

Cas de cohérences : S'il y a une seule fonction d'agrégation multiple dimensionnelle sur la dimension concernée (Figure 27 (a)) ou une seule fonction d'agrégation multiple hiérarchique sur la hiérarchie concernée (Figure 27 (b)) sans aucune autre fonction d'agrégation, alors dans ce cas, il n'y a aucune contradiction entre les contraintes d'agrégation et l'ordre d'exécution. Autrement dit, s'il y a une seule fonction d'agrégation à appliquer sur la hiérarchie, alors la cohérence entre les contraints d'agrégation et l'ordre d'exécution est garantie. C'est parce que la même fonction d'agrégation s'applique successivement tout au long de la hiérarchie en s'appuyant sur ses résultats intermédiaires. La Figure 27 (a) montre un exemple d'une fonction multiple dimensionnelle avec une contrainte d'agrégation fixée à (-1) : l'agrégation à chaque niveau est alors calculée à partir du niveau directement inférieur. La Figure 27 (b) présente le cas d'une fonction multiple hiérarchique avec une contrainte fixée à (-2) : l'agrégation à chaque niveau est calculée à partir du niveau en dessous du niveau directement inférieur s'il existe, sinon elle est calculée à partir du niveau de base. Dans la Figure 27 (b), l'agrégation au niveau P_3^{Di} est calculée à partir du niveau P_1^{Di} ; par contre, l'agrégation au niveau P_1^{Di} est calculée à partir du niveau de base Id^{Di} (parce que il n'a qu'un seul niveau inférieur qui est le niveau de base).

La Figure 27 (c) montre la cohérence entre les contraintes d'agrégation et l'ordre d'exécution dans le cas où les fonctions d'agrégation différenciées sont utilisées. Une fonction d'agrégation peut s'appuyer sur les résultats des fonctions d'agrégation ayant un ordre d'exécution égal ou inférieur à son propre ordre d'exécution. Par exemple, dans la Figure 27 (c), l'agrégation au niveau P_3^{Di} est effectuée par la fonction d'agrégation (T_3) qui a un ordre d'exécution fixé à (1), cette agrégation est calculée à partir de l'agrégation au niveau P_1^{Di} (contrainte d'agrégation fixée à -2) qui est réalisée en utilisant la fonction d'agrégation (T_1) qui a également un ordre d'exécution fixé à (1). Aussi, l'agrégation au niveau 'All^{Di}' est effectuée par la fonction d'agrégation (T_4) qui a un ordre d'exécution fixé à (3). Cette agrégation est calculée à partir de l'agrégation du niveau P_2^{Di} réalisée en utilisant la fonction d'agrégation (T_2) dont l'ordre d'exécution inférieur est fixé à (2).

Cas d'incohérences : L'incohérence entre les contraintes d'agrégation et l'ordre d'exécution est présentée dans la Figure 27 (d et e).

La Figure 27 (d) montre une contradiction entre une fonction d'agrégation différenciée et une fonction d'agrégation multiple dimensionnelle où l'agrégation au niveau P_3^{Di} est effectuée par la fonction (T_2) qui a un ordre d'exécution fixé à (1). Cette agrégation est calculée à partir de l'agrégation au niveau P_1^{Di} ; celle-ci est réalisée en utilisant la fonction d'agrégation multiple dimensionnelle (T_1) qui a un ordre d'exécution supérieur fixé à (2). Donc, l'ordre d'exécution suppose que la fonction (T_2) devrait être exécutée avant la fonction (T_1) ; mais les contraintes d'agrégation supposent l'inverse, car la contrainte de la fonction (T_2) exige l'exécution préalable de la fonction (T_1).

En outre, la Figure 27 (e) montre une contradiction entre la fonction d'agrégation différenciée (T_2) ayant un ordre d'exécution fixé à (2) et la fonction d'agrégation multiple hiérarchique (T_1) ayant un ordre d'exécution fixé à (3). La contrainte d'agrégation de la fonction (T_2) nécessite l'application de (T_1) avant (T_2), contrairement à l'ordre d'exécution qui nécessite l'application de (T_2) avant (T_1). De la même façon, une contradiction entre deux fonctions d'agrégation différenciées (T_2 et T_3) est présentée dans la Figure 27 (e).

Les contradictions entre les contraintes d'agrégation et l'ordre d'exécution dans une BDM peuvent bloquer le système d'analyse OLAP. Ces cas d'incohérence sont détectables par le système. Ce qui évite ainsi à l'analyste de disposer de combinaisons d'agrégation aboutissant à des résultats incohérents pouvant donner lieu à des prises de décision inadaptées.

3.3.3 Analyse multifonctions

Dans le contexte d'une BDM multifonctions, il est possible d'utiliser plusieurs fonctions d'agrégation pour la même analyse. Pour ce faire, le système d'analyse OLAP met en œuvre les étapes suivantes :

1. **Déterminer les niveaux d'agrégation** : plusieurs niveaux d'agrégation sont choisis par l'analyste qui indique les paramètres selon lesquels il souhaite étudier les mesures. Le système d'analyse OLAP doit ajouter à ces paramètres les niveaux 'All^{Di}' pour les dimensions que l'analyste ne prend pas en compte. Le système d'analyse OLAP considère que les mesures sont analysées aux niveaux 'All^{Di}' pour les dimensions n'intervenant pas dans l'analyse ;
2. **Déterminer les fonctions d'agrégation** : cette étape comporte deux sous-étapes :
   a) *Trouver les fonctions d'agrégation pour chaque niveau d'agrégation* : à partir des schémas d'agrégation des mesures concernées, le système d'analyse OLAP détermine les fonctions qu'il doit utiliser pour agréger les valeurs des mesures à chaque niveau d'agrégation résultant de la première étape. C'est pourquoi il cherche s'il y a une fonction d'agrégation différenciée pour le niveau concerné sinon, une fonction multiple hiérarchique sur la hiérarchie concernée sinon, une fonction multiple dimensionnelle sur la dimension concernée sinon, la fonction générale. Une fois qu'il trouve une fonction, il la considère comme la fonction à appliquer.
   b) *Traiter les contraintes d'agrégation* : le système d'analyse OLAP vérifie parmi les fonctions résultantes de l'étape précédente, s'il y a des fonctions contraintes (ayant une contrainte d'agrégation différente de zéro). Dans ce cas, il détermine les niveaux à partir desquels les fonctions contraintes sont calculées. Ensuite, il revient à l'étape 2(a) précédente afin de trouver les fonctions d'agrégation à appliquer pour calculer les valeurs des mesures à ces nouveaux niveaux d'agrégation.

Le système d'analyse OLAP répète cette étape jusqu'à ce qu'il ne trouve plus de fonctions contraintes ;

3. **Traiter l'ordre d'exécution** : le rôle principal de cette étape est de détecter l'existence de cas du **troisième calcul** présenté précédemment (*cf.* § 3.3.1) d'une fonction d'agrégation ayant le même nom, les mêmes arguments et le même ordre d'exécution sur plusieurs dimensions différentes. Dans ce cas, le système d'analyse OLAP doit supprimer la répétition de cette fonction et l'appliquer une seule fois pour toutes les dimensions concernées.
4. **Effectuer l'analyse** : finalement, le système d'analyse OLAP calcule l'analyse demandée en appliquant les fonctions d'agrégation résultantes des trois étapes précédentes et en respectant leur ordre d'exécution et les contraintes d'agrégation.

Dans notre exemple de météo, selon les schémas d'agrégation (Figure 25), les précipitations sont analysées sur la dimension 'Dates' en les agrégeant par la fonction (SUM). Une exception toutefois : au niveau 'All^{Dates}', le système d'analyse OLAP doit calculer les valeurs agrégées à partir du niveau 'AnnéeN' en utilisant une fonction différenciée (AVG). Par ailleurs, si nous analysons les températures moyennes, maximales ou minimales sur les dimensions 'Dates' et de 'Temps', le système d'analyse OLAP doit utiliser les fonctions générales (AVG, MAX et MIN respectivement) pour agréger les valeurs des mesures car il n'y a aucune autre fonction spécifique pour ces dimensions. Basée sur le même principe, les mesures 'Tem_Max' et 'Tem_Min' sont agrégées sur la hiérarchie 'Hgéo_Scien' de la dimension 'Géographie' en utilisant les fonctions générales MAX et MIN respectivement.

L'analyse des quatre mesures 'Précip', 'Tem_Moy', 'Tem_Max' et 'Tem_Min' sur la hiérarchie 'Hgéo_simp' de la dimension 'Géographie', se met en œuvre en utilisant des fonctions multiples hiérarchiques (SELECT_CENTER). Cependant, en analysant les mesures 'Précip' et 'Tem_Moy' sur la hiérarchie 'Hgéo_Scien', le système d'analyse OLAP utilise à chaque niveau d'agrégation une fonction différenciée distincte. L'agrégation est faite à partir du niveau directement inferieur (AVG pour agréger les mesures au niveau 'Département' à partir du niveau 'Ville' et AVG_W pour les autres niveaux).

En outre, si nous analysons les données sur deux dimensions ou plus, alors les fonctions d'agrégation différenciées sur la hiérarchie 'Hgéo_Scien' sont prioritaires pour la mesure 'Tem_Moy' et la fonction multiple dimensionnelle (SUM) sur la dimension 'Dates' est prioritaire pour la mesure 'Précip' ; cela signifie que le système d'analyse OLAP doit les appliquer avant les autres fonctions.

Dans la suite, nous présentons trois exemples en détaillant les quatre étapes d'analyse. La complexité de l'agrégation augmente d'un exemple à l'autre.

Exemple 4. Dans le premier exemple, les analystes souhaitent étudier les températures maximales départementales mensuelles selon l'analyse simple. Les quatre étapes pour réaliser cette analyse se présente comme suit :

1. Déterminer les niveaux d'agrégation : les choix de l'analyste sont le paramètre 'MoisN' de la dimension 'Dates' et le paramètre 'Département' de la hiérarchie 'Hgéo_simp' de la dimension 'Géographie'. En ce qui concerne la troisième dimension d'analyse 'Temps', le système d'analyse OLAP considère que les températures maximales 'Tem_Max' sont analysées au niveau 'All^{Temps}' ;
2. Déterminer les fonctions d'agrégation : selon le schéma d'agrégation de la mesure 'Tem_Max' (Figure 25 (d)), il n'y a ni fonction différenciée ni multiple hiérarchique ni multiple dimensionnelle sur les dimensions 'Dates' et 'Temps', donc la mesure est agrégée aux niveaux 'MoisN' et 'All^{Temps}' en utilisant une seule fonction d'agrégation qui est la fonction générale (MAX). En outre, il n'y a pas de fonction différenciée pour le paramètre 'Département' sur la hiérarchie

'Hgéo_simp' mais cette hiérarchie a une fonction multiple hiérarchique 'SELECT_CENTER', c'est celle que le système d'analyse OLAP utilise afin d'agréger la mesure au niveau 'Département'. Ces fonctions résultantes sont illustrées dans le Tableau 11. Parmi les fonctions résultantes, il n'y a aucune fonction contrainte, le système d'analyse OLAP effectue cette étape en une seule fois ;

Tableau 11 : Déterminer les fonctions d'agrégation de l'analyse de températures maximales départementales mensuelles (analyse simple)

Répétition	Niveaux d'agrégation	Ordre d'exécution	Fonction d'agrégation	Contrainte
1	MoisN	<1>	MAX(Tem_Moy)	0
	All^{Temps}			
	Département	<1>	SELECT_CENTER(Niv_Adm,Tem_Moy)	0

3. Traiter l'ordre d'exécution : parmi les fonctions résultantes (Tableau 11), il n'y a pas une même fonction sur plusieurs dimensions, donc cette étape ne change pas les fonctions à appliquer ;
4. Effectuer l'analyse : les fonctions à appliquer sont commutatives (elles ont le même ordre d'exécution) ; donc afin de réaliser l'analyse, les deux requêtes R_1 et R_2 peuvent être exécutées et elles donnent le même résultat. Ainsi, R_1 exécute la fonction (MAX) avant la fonction (SELECT_CENTER) et R_2 agrège la mesure au niveau 'Département' ensuite aux niveaux 'MoisN' et 'All^{Temps}'. Ces deux requêtes prennent en compte le fait que les températures maximales 'Tem_Max' sont calculées à partir des températures moyennes 'Tem_Moy'.

R_1 :

```
SELECT  MOISN, DEPARTEMENT,
       SELECT_CENTER(LEV_DATA[15](NIVEAU_ADMINISTRATIF,TEM_MOY))AS
TEM_MAX
FROM( SELECT MOISN, DEPARTEMENT, VILLE, NIVEAU_ADMINISTRATIF,
      MAX(TEM_MOY) AS TEM_MOY
      FROM …
      WHERE …
      GROUP BY MOISN, DEPARTEMENT, VILLE, NIVEAU_ADMINISTRATIF)
GROUP BY MOISN, DEPARTEMENT
```

R_2 :

```
SELECT  MAX(TEM_MOY) AS TEM_MAX, MOISN, DEPARTEMENT
FROM( SELECT MOISN, DEPARTEMENT, JOURN, TOUTES_LES_3_HEURES,
      SELECT_CENTER(LEV_DATA(NIVEAU_ADMINISTRATIF, TEM_MOY)) AS TEM_MOY
      FROM …
      WHERE …
      GROUP BY MOISN, DEPARTEMENT, JOURN, TOUTES_LES_3_HEURES)
GROUP BY MOISN, DEPARTEMENT
```

[15] La fonction d'agrégation personnalisée SELECT_CENTER reçoit un objet (TYPE Lev_Data AS OBJECT (level NUMBER, value NUMBER)) qui comprend deux valeurs : le niveau administratif et la valeur de la mesure associée. (Plus de détailles dans le chapitre 6 § 6.2.4.2).

Exemple 5. Dans le deuxième exemple, les analystes souhaitent étudier les températures moyennes annuelles régionales selon une analyse scientifique. Cette analyse est réalisée par les quatre étapes suivantes :

1. Déterminer les niveaux d'agrégation : l'analyste choisit le paramètre 'AnnéeN' de la dimension 'Dates' et le paramètre 'Région' de la hiérarchie 'Hgéo_Scien' de la dimension 'Géographie'. Comme dans l'exemple précédent, le système d'analyse OLAP considère que les températures moyennes 'Tem_Moy' sont analysées au niveau 'All^{Temps}' sur la dimension 'Temps' ;
2. Déterminer les fonctions d'agrégation : à partir du schéma d'agrégation de la mesure 'Tem_Moy' (Figure 25 (b)), le système d'analyse OLAP détermine dans la première répétition de cette étape que les températures moyennes sont agrégées aux niveaux 'AnnéeN' et 'All^{Temps}' respectivement sur les dimensions 'Dates' et 'Temps' en utilisant la fonction d'agrégation générale (AVG). Il trouve également que cette mesure est agrégée au paramètre 'Région' sur la hiérarchie 'Hgéo_Scien' de la dimension 'Géographie' par la fonction différenciée (AVG_W). Ensuite, comme cette fonction différenciée est calculée à partir du niveau 'Département' (contrainte d'agrégation fixée à -1), le système d'analyse OLAP répète cette étape une deuxième fois afin de trouver la fonction qui calcule les températures moyennes départementales. Il détermine, selon le schéma d'agrégation (Figure 25 (b)), dans cette deuxième répétition que les températures départementales sont calculées à partir des températures des villes par une fonction d'agrégation différenciée (AVG). Donc, il ajoute cette fonction (AVG) aux fonctions à appliquer trouvées dans la première répétition (AVG et AVG_W). Bien que la nouvelle fonction trouvée soit contrainte, une troisième répétition de cette étape n'ajoute pas de nouvelles fonctions à appliquer parce que le niveau 'Ville' est un niveau de base. Cela confirme l'idée présentée précédemment *"les contraintes sur les fonctions d'agrégation différenciées associées aux niveaux de base, ne change rien par rapport à l'agrégation des mesures"*. Toutes les fonctions résultantes des première et deuxième répétitions sont présentées dans le Tableau 12 ;

Tableau 12 : Déterminer les fonctions d'agrégation de l'analyse des températures moyennes annuelles régionales (analyse scientifique)

Répétition	Niveaux d'agrégation	Ordre d'exécution	Fonction	Contrainte
1	AnnéeN	<2>	AVG(Tem_Moy)	0
	All^{Temps}			
	Région	<1>	AVG_W(Tem_Moy , D_Superficie)	-1
2	Département	<1>	AVG(Tem_Moy)	-1

3. Traiter l'ordre d'exécution : parmi les fonctions résultantes (Tableau 12), la fonction (AVG(Tem_Moy)) est exécutée deux fois sur des dimensions différentes mais avec deux ordres d'exécution différents ; ceci empêche toute simplification des fonctions. Donc, cette étape ne change pas les fonctions à appliquer ;
4. Effectuer l'analyse en respectant d'un côté, les contraintes d'agrégation qui nécessitent de calculer la mesure au niveau 'Département' avant de la calculer au niveau 'Région', et de l'autre l'ordre d'exécution qui exige d'agréger la mesure sur la dimension 'Géographie' avant les autres dimensions. Une seule requête R_3 peut être exécutée afin de réaliser l'analyse.

R_3:

```
SELECT  ANNEE, REGION, AVG(TEM_MOY) AS TEM_MOY
FROM  ( SELECT  ANNEE, REGION, JourN, TOUTES_LES_3_HEURES,
```

```
        AVG_W(DATA_WEIGHTED[16](TEM_MOY, D_SUPERFICIE)) AS TEM_MOY
    FROM ( SELECT ANNEE, REGION, JourN, DEPARTEMENT, D_SUPERFICIE,
                  TOUTES_LES_3_HEURES, AVG(TEM_MOY) AS TEM_MOY
           FROM …
           WHERE …
           GROUP BY ANNEE, REGION, JourN, DEPARTEMENT,
                    D_SUPERFICIE, TOUTES_LES_3_HEURES)
    GROUP BY ANNEE, REGION, JourN, TOUTES_LES_3_HEURES)
GROUP BY ANNEE, REGION
```

Exemple 6. Dans le troisième exemple, les analystes souhaitent étudier les précipitations moyennes régionales selon une analyse scientifique. Cette analyse s'effectue comme suit :

1. Déterminer les niveaux d'agrégation : l'analyste a précisé un seul niveau d'agrégation qui est le paramètre 'Région' de la hiérarchie 'Hgéo_Scien' de la dimension 'Géographie'. Donc, le système d'analyse OLAP considère que les précipitations 'Précip' sont analysées au niveau 'All^{Dates}' sur la dimension 'Dates' ;
2. Déterminer les fonctions d'agrégation : le schéma d'agrégation de la mesure 'Précip' (Figure 25 (a)) conduit, dans la première répétition de cette étape, le système d'analyse OLAP à utiliser les fonctions d'agrégation différenciées (AVG_W) et (AVG) afin d'agréger la mesure aux niveaux 'Région' sur la hiérarchie 'Hgéo_Scien' et 'All^{Dates}' des dimensions 'Géographie' et 'Dates' respectivement. Ces deux fonctions sont contraintes (contrainte d'agrégation fixée à -1), ce qui nécessite une deuxième répétition de cette étape. Cette deuxième répétition amène le système d'analyse OLAP à préciser qu'avant l'utilisation des fonctions trouvées dans la première répétition (AVG_W) et (AVG), il devrait exécuter la fonction différenciée (AVG) et la fonction dimensionnelle (SUM) pour calculer les précipitations aux niveaux 'Département' et 'AnnéeN' respectivement. Toutes les fonctions résultantes de première et de deuxième répétitions sont présentées dans le Tableau 13 ;

Tableau 13 : Déterminer les fonctions d'agrégation de l'analyse de précipitations moyennes régionales (analyse scientifique)

Répétition	Niveaux d'agrégation	Ordre d'exécution	Fonction	Contrainte
1	All^{Dates}	<2>	AVG(Précip)	-1
	Région	<3>	AVG_W(Précip, D_Superficie)	-1
2	AnnéeN	<1>	SUM(Précip)	0
	Département	<2>	AVG(Précip)	0

3. Traiter l'ordre d'exécution : parmi les fonctions résultantes (Tableau 13), la fonction (AVG(Précip)) est répétée deux fois sur les dimensions ('Géographie' et 'Dates') avec le même ordre d'exécution (<2>). Le système d'analyse OLAP peut ainsi éviter la répétition de cette fonction et l'exécuter une seule fois pour tous les

[16] La fonction d'agrégation personnalisée AVG_W reçoit un objet (TYPE Data_Weighted AS OBJECT (value NUMBER, weight NUMBER)) qui comprend les données et leurs poids associés. (Plus de détailles dans le chapitre 6 § 6.2.4.2).

paramètres concernés (‘All^{Dates}’ et ‘Département’). Après l’accomplissement de cette étape, les fonctions d’agrégation à appliquer sont présentées dans le Tableau 14 ;

Tableau 14 : Traiter l’ordre d’exécution de l’analyse de précipitations moyennes régionales (analyse scientifique)

Niveaux d’agrégation	Ordre d’exécution	Fonction
All^{Dates}	<2>	AVG(Précip)
Département		
Région	<3>	AVG_W(Précip, D_Superficie)
AnnéeN	<1>	SUM(Précip)

4. Effectuer l’analyse : selon l’ordre d’exécution des fonctions résultantes de l’étape précédente (Tableau 14), il n’y a pas de fonctions commutatives. La requête (R_4) est la seule à pouvoir réaliser l’analyse.

R_4 :

```
SELECT  REGION, AVG_W(DATA_WEIGHTED(PRECIP, D_SUPERFICIE)) AS PRECIP
FROM  ( SELECT REGION, DEPARTEMENT, D_SUPERFICIE, AVG(PRECIP) AS PRECIP
        FROM( SELECT  ANNEE, REGION, DEPARTEMENT, D_SUPERFICIE, VILLE,
                      SUM(PRECIP) AS PRECIP
              FROM …
              WHERE …
              GROUP BY  ANNEE, REGION, DEPARTEMENT, D_SUPERFICIE, VILLE)
        GROUP BY REGION, DEPARTEMENT, D_SUPERFICIE)
GROUP BY REGION
```

3.4 CONCLUSION

Dans ce chapitre, nous avons proposé un modèle conceptuel multidimensionnel multifonctions pour répondre à notre problématique. Ce modèle reprend les avantages de la modélisation multidimensionnelle. Il représente les concepts multidimensionnels selon une vision orientée décideur [Golfarelli et al., 2002] indépendamment des contraintes d’implantation logiques ou physiques. Notre modèle conceptuel se base sur les concepts de faits, de dimensions et de hiérarchies augmentés de mécanismes d’agrégations multiples (fonctions d’agrégation, ordre d’exécution et contraintes d’agrégation). Nous proposons, pour chaque concept, un formalisme graphique dont l’objectif est de simplifier la représentation du schéma multidimensionnel.

A la manière du modèle en constellation [Ravat et al., 2008], notre modèle décrit les éléments structurels dans un schéma en étoile ou en constellation en masquant la complexité des agrégations [Hassan et al., 2012b], [Hassan et al., 2012c], [Hassan et al., 2013a], [Hassan et al., 2014]. Ce schéma regroupe un ensemble de sujets d’analyse (faits) avec leurs axes d’analyse (dimensions) éventuellement partagés afin de faciliter la corrélation entre les faits [Ghozzi, 2004]. Chaque dimension comprend une ou plusieurs hiérarchies organisant les niveaux de granularités (paramètres) des analyses du niveau le plus fin (paramètre racine) jusqu’au niveau le plus général (paramètre extrémité ‘All^{Di}’). Ces hiérarchies permettent d’associer à chaque paramètre plusieurs attributs faibles. Un même paramètre peut appartenir à plusieurs hiérarchies. Les hiérarchies sont définies explicitement dans notre modèle pour faciliter la manipulation des données multidimensionnelles selon les différents paramètres [Ghozzi, 2004] et pour pouvoir utiliser des agrégations différentes sur les parties communes à plusieurs hiérarchies.

Notre modèle est suffisamment expressif pour permettre au concepteur de combiner une même mesure avec différentes fonctions d'agrégation [Hassan et al., 2012a], [Hassan et al., 2012b], [Hassan et al., 2012c], [Hassan et al., 2013a], [Hassan et al., 2014]. Il permet de spécifier des fonctions d'agrégations multiples dimensionnelles, multiples hiérarchiques et différenciées en fonction des dimensions, des hiérarchies et des paramètres utilisés. Par contre, notre modèle ne permet pas de spécifier une fonction d'agrégation pour une instance d'un paramètre comme « Analysis Services de Microsoft ». De plus, le modèle permet de contrôler la validité des calculs des fonctions. Les contraintes d'agrégation définissent le niveau à partir duquel l'agrégation doit être calculée. L'ordre d'exécution définit l'ordre nécessaire entre les fonctions d'agrégation non-commutatives. Ces mécanismes d'agrégation liés à chaque mesure sont détaillés dans un schéma d'agrégation différent.

La présentation des formalismes graphiques en deux niveaux (un schéma structurel et des schémas d'agrégation) et l'utilisation d'un schéma d'agrégation différent pour chaque mesure, ont pour but d'améliorer la lisibilité et de faciliter la compréhension de la BDM pour l'analyste [Hassan et al., 2012b], [Hassan et al., 2012c], [Hassan et al., 2013a], [Hassan et al., 2014]. Pour ces mêmes motifs, les hiérarchies sont présentées en version divisée aux schémas d'agrégation contrairement au schéma structurel où elles sont présentées en version compacte.

Par ailleurs, les fonctions d'agrégation sont intégrées dans le modèle conceptuel afin de

- Eviter de commettre des erreurs par les analystes en utilisant des fonctions inappropriées,
- Les utiliser pour réaliser des pré-calculs des agrégats,
- Permettre leur utilisation lors de l'analyse.

Pour cette dernière raison, nous avons défini quatre étapes que le système d'analyse OLAP doit effectuer pour chaque analyse :

1. Déterminer les niveaux d'agrégation : pour définir l'analyse demandée,
2. Déterminer les fonctions d'agrégation : pour préciser les fonctions d'agrégation à appliquer et traiter les agrégations contraintes qui nécessitent une agrégation en plusieurs étapes,
3. Traiter l'ordre d'exécution : pour éviter des répétitions des fonctions d'agrégation inutiles ou incorrectes,
4. Effectuer l'analyse : pour réaliser l'analyse demandée en appliquant les fonctions d'agrégation en respectant leur ordre d'exécution et leurs contraintes d'agrégation.

Afin d'éviter tout blocage au cours de l'analyse, nous exigeons une couverture complète (*cf.* Lemme 4) du schéma multidimensionnel par les fonctions d'agrégation et une cohérence entre les contraintes d'agrégation et l'ordre d'exécution.

La proposition d'un modèle conceptuel constitue la première étape de la conception de bases de données multidimensionnelles multifonctions. L'étude des impacts de ce modèle sur le niveau logique et les pré-agrégats est réalisée dans le chapitre suivant.

4. CHAPITRE IV : MODÈLE LOGIQUE MULTIDIMENSIONNEL MULTIFONCTIONS

4.1 INTRODUCTION

Le volume de l'entrepôt de données et la complexité des requêtes peuvent provoquer des temps de réponse importants. Ce retard est souvent inapproprié dans la plupart des systèmes d'aide à la décision. L'exigence habituelle pour le temps d'exécution de la requête est de quelques secondes ou de quelques minutes au maximum [Harinarayan et al., 1996]. Il y a deux technologies utilisées afin d'atteindre ces objectifs de performance : l'optimisation par index et la matérialisation de vues.

Optimisation des index : les index sont des structures de données physiques permettant d'accélérer la recherche, le tri, la jointure et l'agrégation des données. Ils jouent un rôle essentiel dans les bases de données en général, et dans les systèmes décisionnels en particulier, où ils réduisent le temps des réponses aux requêtes. Les optimiseurs de requête et les techniques d'évaluation des requêtes peuvent gérer les agrégations [Chaudhuri & Shim, 1994], [Gupta et al., 1995] et utiliser différentes stratégies d'indexation comme index bitmap, index de jointure [O'Neiland & Graefe, 1995] et index de jointure en étoile [Red, 1997]. Ces travaux ne se situent pas dans le cadre de nos études mais présentent un axe de recherche important.

Matérialisation des vues : une technique couramment utilisée en entrepôt de données consiste à matérialiser (pré-calculer) des vues en stockant des résultats de requêtes fréquemment demandées. Les valeurs de nombreuses vues sont calculables à partir de celles d'autres vues. Ces vues sont modélisées par un treillis [Harinarayan et al., 1996] où les nœuds représentent les vues et les arcs représentent les relations de dépendance entre elles. La construction du treillis est une étape préliminaire qui permet par la suite de sélectionner l'ensemble des vues à matérialiser [Harinarayan et al., 1996], [Paraboschi et al, 2003]. Sélectionner le bon ensemble de vues à matérialiser est une des décisions les plus importantes dans la conception d'un entrepôt de données [Gupta, 1997], car en matérialisant une vue, nous pourrions être en mesure de répondre rapidement à des requêtes sur d'autres vues. Par exemple, nous pourrions vouloir matérialiser une vue peu demandée si cela contribuait à répondre rapidement à de nombreuses requêtes sur d'autres vues [Harinarayan et al., 1996].

Trois stratégies distinctes sont possibles pour choisir l'ensemble de vues à matérialiser [Ullman, 1996] :

- Matérialiser toutes les vues (nœuds du treillis) : cette approche donne le meilleur temps de réponse des requêtes. Cependant, pré-calculer et stocker toutes les vues n'est pas applicable pour les bases des données multidimensionnelles volumineuses car l'espace consommé devient excessif [Harinarayan et al., 1996] et le coût de mise-à-jour peut être très élevé [Theodoratos & Sellis, 1997]. Il est à noter que l'espace utilisé pour stocker les données est un bon indicateur du temps qu'il faut

pour créer la vue matérialisée. L'espace utilisé a également des impacts sur l'indexation et donc il augmente le coût global [Harinarayan et al., 1996].

- Ne matérialiser aucune vue : dans ce cas, nous avons besoin d'accéder aux données de base brutes et de calculer chaque requête à la demande. Cette approche ne comporte aucune optimisation pour améliorer les performances de calculs des requêtes.
- Matérialiser seulement une partie du treillis : cette stratégie est un compromis entre les deux stratégies précédentes. Plus nous matérialisons de vues, plus les performances des requêtes seront améliorées. Si nous ne pouvons matérialiser qu'une fraction des vues, en raison de contraintes d'espace de stockage, il devient essentiel de choisir la matérialisation des vues parmi toutes celles possibles.

Notre objectif est d'étudier les conséquences de l'utilisation de plusieurs fonctions d'agrégation sur le stockage de données, spécialement le treillis des pré-agrégats qui devraient être calculés en utilisant les fonctions d'agrégation prédéfinies au niveau conceptuel.

Dans ce chapitre, nous détaillons nos propositions au niveau logique en présentant comment les données des faits, des dimensions et les données pré-calculées sont stockées dans un contexte relationnel (R-OLAP).

Plan du chapitre. Dans cette section, nous présentons notre problématique et notre proposition. La deuxième section détaille la modélisation des schémas multidimensionnels au niveau logique dans un environnement relationnel. La troisième section explique comment le modèle du treillis est utilisé afin d'optimiser les systèmes décisionnels dans un contexte qui utilise une seule fonction d'agrégation pour chaque mesure (uni-fonction). La quatrième section étudie les impacts de notre modèle multifonctions sur le treillis.

4.1.1 Problématique

L'utilisation de plusieurs fonctions d'agrégation pour la même mesure au niveau conceptuel ne devrait pas influencer le stockage des données de base (faits et dimensions). Ce qui pourrait permettre d'appliquer notre modèle multifonction sur des systèmes décisionnels sans la nécessité de modifier ces données. Par contre, les données pré-agrégées sont impactées par cette utilisation de plusieurs fonctions, parce que ces fonctions définissent les calculs.

Dans la littérature, deux problèmes principaux liés aux vues matérialisées ont été traités :

- La sélection des vues matérialisées [Harinarayan et al., 1996], [Yang et al., 1997], [Baralis et al., 1997], [Gupta, 1997], [Theodoratos & Sellis, 1997], [Han et al., 1998], [Gupta & Mumick, 1999], [Kotidis & Roussopoulos, 1999], [Agrawal et al., 2000], [Liang et al., 2001], [Lee & Hammer, 2001], [Kotidis & Roussopoulos, 2001], [Kalnisa et al., 2002], [Yu et al., 2003], [Li et al., 2005], [Yu et al., 2005], [Lawrence & Rau-Chaplin, 2006], [Nandi et al., 2011], [Hanusse et al., 2011], [Kerkad et al., 2013], [Boukorca et al., 2013],
- La maintenance des vues matérialisées [Theodoratos & Sellis, 1997], [Huyn, 1997], [Zhuge et al., 1997], [Zhuge et al., 1998], [Yang & Widom, 2000], [Labio et al., 2000], [Kotidis & Roussopoulos, 2001], [Li et al., 2005], [Hanusse et al., 2011].

Tous ces travaux n'exploitent pas plusieurs fonctions d'agrégation pour la même mesure. Notre objectif est de faire une étude détaillée des effets de l'utilisation de plusieurs fonctions d'agrégation sur le treillis qui modélise les pré-agrégats. A notre connaissance, ceci constitue un travail original.

4.1.2 Notre proposition

Dans ce chapitre, nous présentons l'implantation des schémas multidimensionnels en utilisant l'approche R-OLAP [Kimball, 1996]. Ensuite, nous exploitons l'ensemble des fonctions d'agrégation dans le treillis des pré-agrégats. Cette exploitation peut affecter les pré-agrégats (les nœuds du treillis) et les relations de calcul entre ces pré-agrégats (les arcs du treillis). Nous distinguons sept impacts :

- Augmentation du nombre de nœuds,
- Typage des arcs,
- Modification d'arcs,
- Elagage du treillis,
- Blocage de la transitivité,
- Changement des données stockées,
- Différences entre les treillis d'optimisation des mesures analysées selon les mêmes dimensions.

4.2 ETOILE R-OLAP

L'implantation courante repose sur l'approche dite R-OLAP. Elle consiste à utiliser un système de gestion de données relationnelles pour implanter les schémas multidimensionnels [Kimball, 1996], [Dinter et al., 1998], [Mangisengi & Tjoa, 1998]. Cette approche procure de nombreux avantages :

- La réutilisation de mécanismes de gestion des données éprouvés,
- La capacité à gérer des volumes de données importants.

Les modèles R-OLAP transforment les faits et les dimensions du modèle multidimensionnel conceptuel au niveau logique en tables relationnelles [Kimball, 1996] comme suit :

- Chaque dimension est traduite en une table relationnelle de même nom. Les paramètres et les attributs faibles de la dimension sont représentés par les attributs de la table relationnelle (à l'exception du paramètre extrémité 'All^{Di}' qui n'est pas traduit). La clé primaire de la table relationnelle est dérivée du paramètre racine ;
- Chaque fait est traduit en une table relationnelle de même nom. Cette table comprend :
  - Des attributs correspondants aux mesures du fait du modèle conceptuel considéré,
  - Des clés étrangères référençant les tables relationnelles représentant les dimensions liées au fait considéré.

  La clé primaire de la table du fait se compose des clés étrangères (ou bien d'un attribut supplémentaire auto-incrémenté servant compteur des instances du fait).

Les tables résultantes et leurs relations forment un schéma logique dit schéma en étoile [Kimball, 1996].

Exemple 1. La Figure 28 montre un exemple d'un schéma logique R-OLAP en étoile. Ce schéma est basé sur le schéma conceptuel du fait 'Température' dans le schéma structurel de notre exemple de météo (Figure 24). Ce fait comprend trois mesures : les températures moyennes 'Tem_Moy', maximales 'Tem_Max' et minimales 'Tem_Min'. Cependant, les mesures des températures maximales 'Tem_Max' et minimales 'Tem_Min' ne sont pas stockées dans la table relationnelle correspondante au fait car elles sont calculées à partir des températures moyennes 'Tem_Moy'.

En ce qui concerne les fonctions d'agrégation, elles sont stockées dans le moteur de base de données. Par ailleurs, nous utilisons un méta-schéma (détaillé dans le chapitre 6) pour décrire le schéma multidimensionnel (faits, dimensions et hiérarchies) correspondant aux tables relationnelles R-OLAP qui stockent les données d'analyse. Il décrit également les différentes fonctions d'agrégation, l'ordre d'exécution et les éventuelles contraintes d'agrégation.

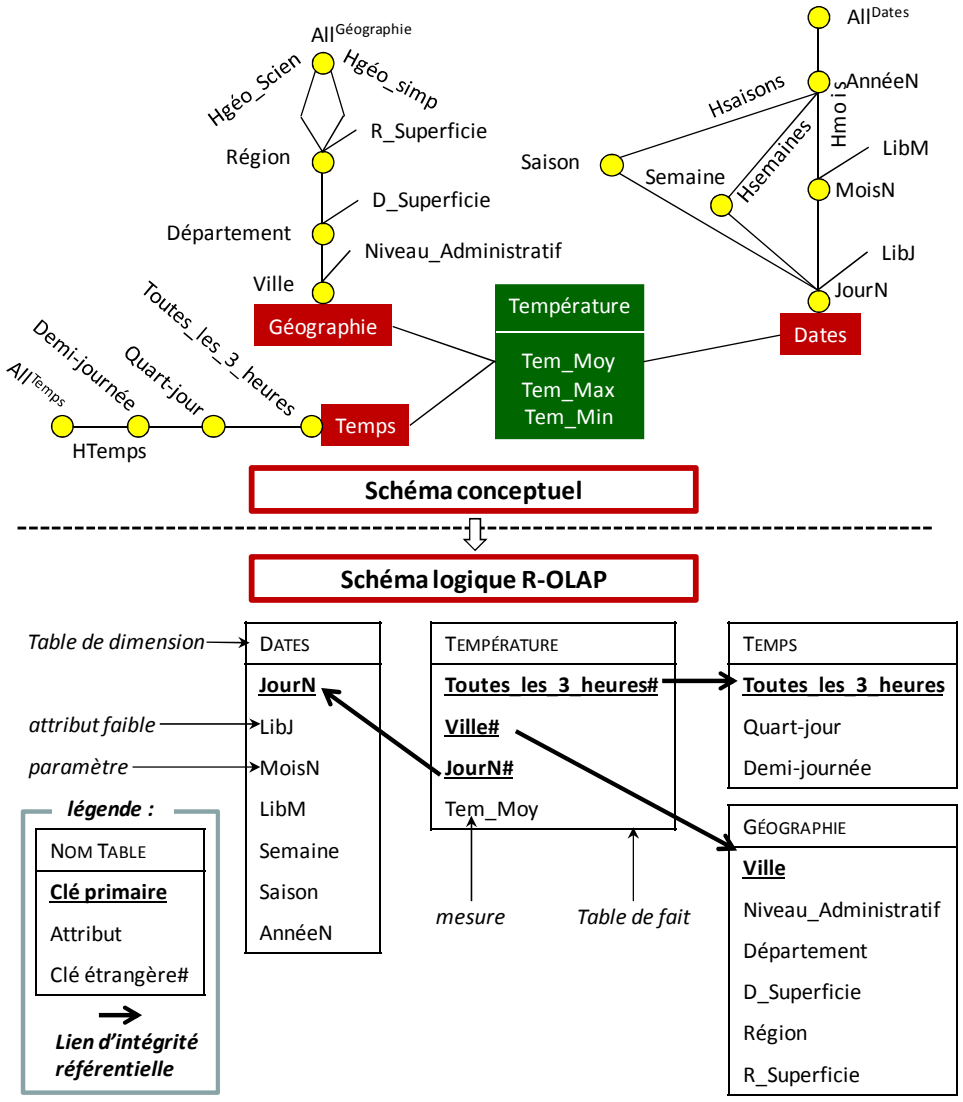


Figure 28 : Exemple de schéma logique R-OLAP en étoile

En normalisant les tables des dimensions, nous obtenons le schéma en flocon. Pour cela, il faut transformer chaque dimension du modèle conceptuel en plusieurs tables relationnelles. Chaque table relationnelle correspond à un niveau de granularité (paramètre). Elle contient :

- Une clé primaire dérivée du paramètre considéré,
- Une clé étrangère qui référence la table du paramètre directement supérieur. Par exemple, la table 'Département' référence la table 'Région',
- Des attributs correspondants aux attributs faibles associés au paramètre considéré.

Exemple 2. La Figure 29 montre le schéma logique R-OLAP en flocon traduit du même schéma conceptuel que l'exemple 1 (Figure 28).

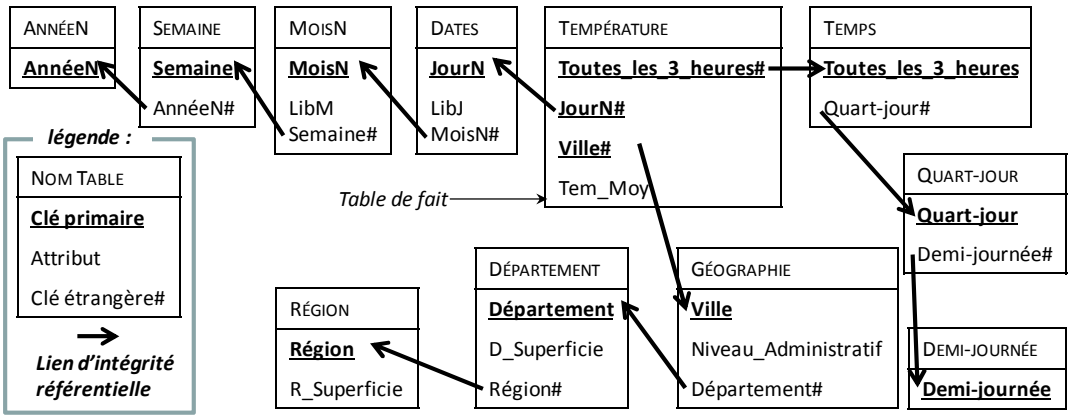


Figure 29 : Exemple de schéma logique R-OLAP en flocon

Le schéma en flocon présente les avantages suivants :

- Expliciter les hiérarchies,
- Eliminer les redondances de données, donc diminuer l'espace de stockage nécessaire,
- Permettre le partage partiel de dimensions entre les faits d'une constellation.

Par contre, le schéma en flocon utilise beaucoup plus de liens d'intégrité référentielle que le schéma en étoile afin de relier les différentes tables de niveaux de granularité. C'est pourquoi il faut effectuer beaucoup de jointures afin d'exécuter les requêtes d'analyse. Donc, avec un volume de données très important, le temps d'exécution est pénalisé dans un contexte d'analyse interactive [Kimball, 1996].

En ce qui concerne les schémas en constellation, nous pouvons exploiter les mêmes principes de transformation précédents.

4.3 ETOILE OPTIMISÉE UNI-FONCTION

La modélisation conceptuelle permet de structurer hiérarchiquement les graduations (paramètres) des axes d'analyses. Ces hiérarchies sont exploitées pour optimiser la BDM. Cette optimisation consiste à compléter le schéma multidimensionnel par un ensemble de relations pré-calculant les agrégations nécessaires aux décideurs lors de leurs interrogations et analyses OLAP. Classiquement, les pré-agrégations sont modélisées par un treillis [Harinarayan et al., 1996] qui est formé en représentant toutes les agrégations possibles en fonction de différentes combinaisons de paramètres [Nandi et al., 2011]. Chaque nœud représente un pré-agrégat (vue) et chaque arc représente une relation parent/enfant entre deux nœuds ; le nœud enfant contient un paramètre plus détaillé que celui du nœud parent [Nandi et al., 2011].

En fonction de la sémantique des relations entre les vues, nous pouvons définir trois types de treillis [Gupta, 1997] :

- Un treillis de type **ET** : ce treillis est présenté par un graphe acyclique (Figure 30 (a)) ; si un nœud (vue) parent U (supérieur) a plusieurs nœuds enfants $v_1, v_2, \ldots, v_k$ (inférieurs), alors toutes les vues $v_1, v_2, \ldots, v_k$ sont nécessaires pour calculer U et cette dépendance est présentée par un demi-cercle à travers les arcs (U, v_1), (U, v_2), …, (U, v_n), appelé Arc Et ;

- Un treillis de type **OU** : ce treillis est présenté par un graphe acyclique (Figure 30 (b)) ; contrairement au treillis de type ET, si un nœud parent U (supérieur) a plusieurs nœuds enfants v_l, v_2, . . . , v_k (inférieurs), alors U peut être calculé à partir de chaque vue $v_i \in \{v_l, v_2, \ldots, v_k\}$;
- Un treillis de type **ET-OU** : ce type est une fusion des deux types précédents. Ce treillis est présenté par un graphe acyclique (Figure 30 (c)) dont chaque nœud parent est relié aux nœuds enfants, à partir desquels il peut être calculé, par une ou plusieurs dépendances de type ET et OU.

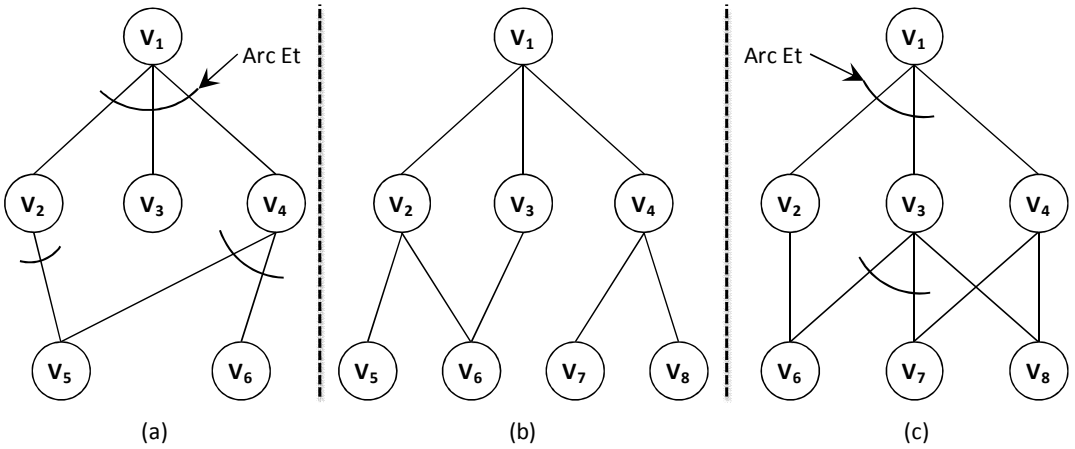


Figure 30 : Treillis de types ET, OU et ET-OU

Exemple 3. Pour éviter que le treillis soit trop complexe, nous simplifions l'exemple de météo présenté précédemment. Nous ne prenons en compte que le fait 'Précipitation' ayant une seule mesure : les précipitations 'Précip' analysée selon les deux dimensions :

- 'Géographie' avec ses deux hiérarchies 'Hgéo_Scien' pour l'analyse scientifique et 'Hgéo_Simp' pour l'analyse simple (*cf.* § 1.3.1),
- 'Dates' avec une seule hiérarchie 'Hannée'.

Comme la Figure 31 (a) l'illustre, chaque hiérarchie a trois niveaux de granularité (deux paramètres en plus du paramètre extrémité).

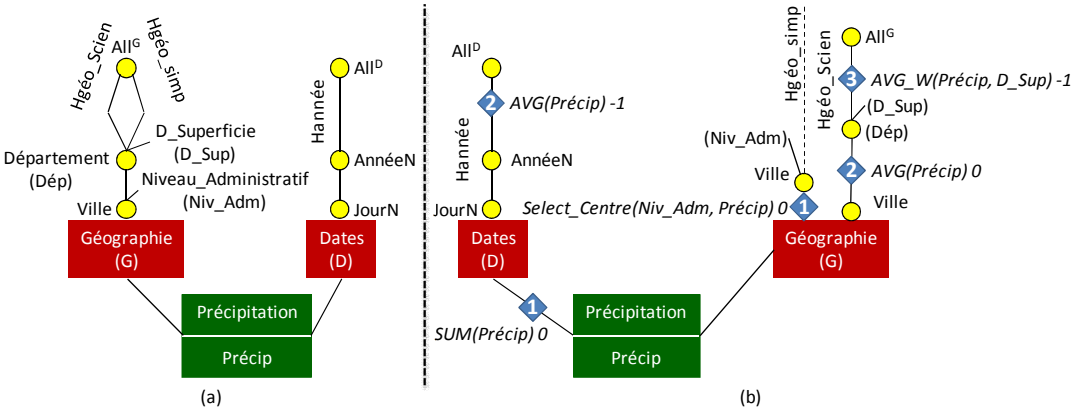


Figure 31 : Schémas structurel et d'agrégation de l'exemple simplifié

Un algorithme pour construire le treillis de pré-agrégats, en se basant sur la structure hiérarchique, afin d'inclure la sémantique des hiérarchies est défini dans [Ghozzi, 2004]. Le treillis de pré-agrégats de la mesure 'Précip' résultant d'un tel algorithme est représenté dans la Figure 32.

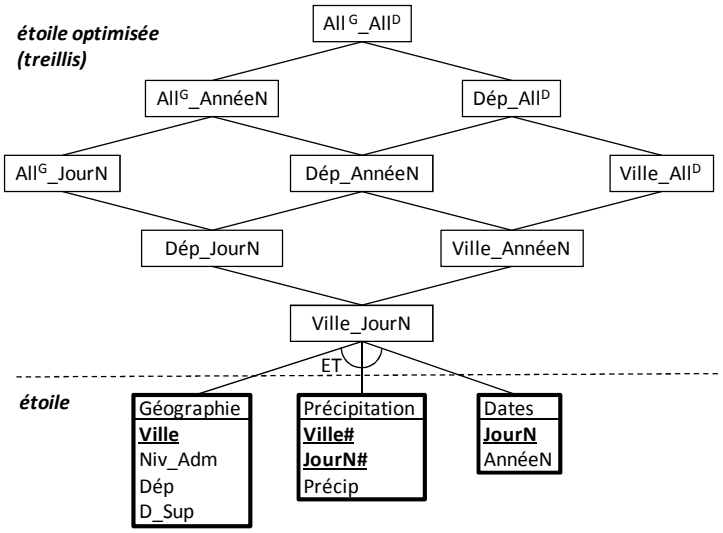


Figure 32 : Treillis d'optimisation uni-fonction[17]

Pour une mesure analysée en fonction de *m* dimensions, nous pouvons représenter chaque nœud par un n-uplet $<a_1, a_2, \ldots, a_m>$ où chaque a_i est un paramètre de l'i-ième dimension [Harinarayan et al., 1996], formellement : $a_i \in P^{Di}$. Par exemple, le nœud 'Dép_AnnéeN' représente l'agrégation des précipitations 'Précip' en fonction des paramètres 'Département' et 'AnnéeN'. Chaque arc du treillis est défini par les deux nœuds entre lesquels il se trouve. Par exemple l'arc ('Ville_AnnéeN', 'Ville_All^D') relie deux nœuds : le nœud parent 'Ville_All^D' avec le nœud enfant 'Ville_AnnéeN'.

Le nœud racine (borne inférieure) du treillis correspond à la vue représentant les données en fonction de la combinaison des paramètres racines de toutes les dimensions [Han et al., 1998]. Ce nœud peut être calculé à partir des données de base (faits et dimensions) contenues dans les tables de l'étoile R-OLAP. En outre, le nœud final (borne supérieure) correspond à la vue agrégeant totalement les données en fonction de la combinaison des paramètres extrémité (All^{Di}) de toutes les dimensions [Han et al., 1998].

Dans cette approche classique, contrairement à notre proposition, une fonction d'agrégation unique est utilisée dans l'ensemble du treillis pour la mesure 'Précip'. Lorsque la fonction d'agrégation utilisée est distributive ou algébrique [Gray et al., 1996], un agrégat est calculable directement à partir de l'agrégat inférieur direct, tandis que dans le cas d'une agrégation holistique [Gray et al., 1996], l'agrégat se calcule en cheminant jusqu'aux relations de base.

[17] Ici, nous avons utilisé les abréviations qui sont entre parenthèses dans la Figure 31.

Dans le cas d'une fonction distributive ou algébrique (à la condition de stocker des valeurs intermédiaires), nous pouvons définir la ***transitivité des calculs*** dans le treillis [Gupta, 1997] : si une vue (nœud) U peut être calculée à partir des vues V, u_l, u_2,…, u_n et la vue V peut être calculée à partir des vues v_l, v_2, …, v_k, alors la vue U peut être calculée à partir des vues v_l, v_2, …, v_k, u_l, u_2,…, u_n. Ainsi, une vue $<b_1, b_2, \ldots, b_m>$ est calculable d'une vue $<a_1, a_2,..., a_m>$ si et seulement si :

$$\forall i \in [1..m] \mid a_i \prec^{D_i} b_i \vee a_i = b_i$$

Dans nos travaux, nous supposons que les vues matérialisées sont stockées dans un environnement relationnel. Donc, chaque nœud dans la Figure 32 représente une relation. La Figure 33 décrit les relations (schéma et contenu) qui permettent d'optimiser notre exemple simplifié (Figure 31 (a)). Les données de base (faits et dimensions) d'étoile R-OLAP de cet exemple sont les suivantes :

GÉOGRAPHIE (**Ville,** Niveau_Administratif, Département, D_Superficie)

Toulouse	N3	Haute-Garonne	6309
Juzet-de-Luchon	N1	Haute-Garonne	6309

DATES (**JourN,** AnnéeN)

1-1-2013	2013
2-1-2013	2013
1-1-2014	2014

PRÉCIPITATION (**Ville#, JourN#,** Précip)

Toulouse	1-1-2013	10
Toulouse	2-1-2013	9
Toulouse	1-1-2014	6
Juzet-de-Luchon	1-1-2013	8
Juzet-de-Luchon	2-1-2013	5
Juzet-de-Luchon	1-1-2014	6

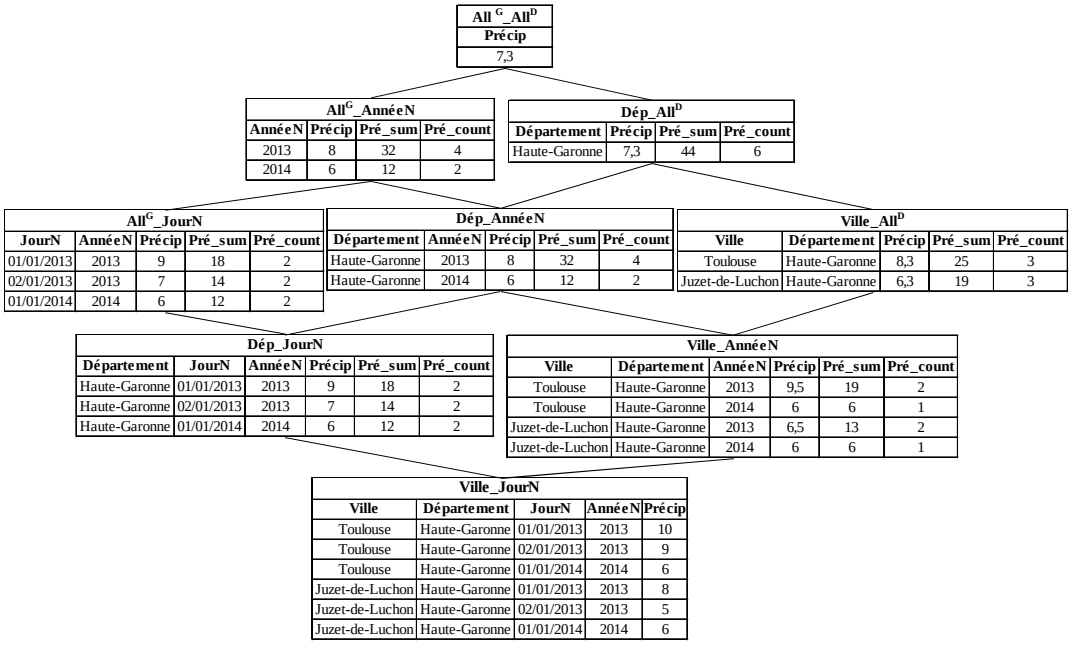


Figure 33 : Les relations du treillis d'optimisation uni-fonction

Dans ces relations, l'attribut 'Précip' représente la moyenne des précipitations calculée par la fonction d'agrégation AVG. Il s'agit, ici, d'un cas de fonction algébrique. Donc, des

valeurs intermédiaires (la somme 'Pré_sum' et le nombre 'Pré_count' des occurrences des précipitations) sont stockées et seront utilisées pour calculer les nœuds supérieurs. Par exemple, le nœud 'Dép_All^{D}' est calculable en divisant la somme de 'Pré_sum' par la somme de 'Pré_count' (SUM(Pré_sum) / SUM (Pré_count)) des nœuds 'Dép_AnnéeN' et 'Ville_All^{D}' directement inférieurs ou par la transitivité, des nœuds 'Ville_AnnéeN' et 'Dép_JourN' plus loin. Par conséquence, tous les nœuds peuvent être calculés à partir de nœud racine ('Ville_JourN') et également, le nœud final ('All^{G}_All^{D}') est calculable à partir de tous les autres nœuds [Li et al., 2005].

Le treillis présenté dans cette section ne tient pas compte la combinaison de plusieurs fonctions d'agrégation sur la même mesure. Il considère que les précipitations sont agrégées par une seule fonction d'agrégation (AVG). Autrement dit, ce treillis ne prend pas en compte le schéma d'agrégation présenté dans la Figure 31 (b). Dans la section suivante, nous étudions, en considérant le schéma d'agrégation, les impacts de cette combinaison de plusieurs fonctions sur le treillis.

4.4 ETOILE OPTIMISÉE MULTIFONCTIONS

L'expressivité des mécanismes d'agrégation (les différentes fonctions d'agrégation, les contraintes d'agrégation et l'ordre d'exécution) que nous avons introduite dans le modèle conceptuel peut être exploitée au niveau du treillis.

4.4.1 Augmentation du nombre de nœuds

Dans notre modèle multidimensionnel conceptuel multifonctions, en utilisant les fonctions d'agrégation multiples hiérarchiques et/ou les fonctions différenciées, nous pouvons associer (pour une mesure concernée) le même paramètre dans des hiérarchies différentes à des fonctions d'agrégation différentes. Cela donne des résultats d'agrégation différents pour la même analyse selon la hiérarchie utilisée. Ainsi, des nouveaux nœuds compatibles avec les résultats de toutes ces agrégations possibles se produiront dans le treillis (Figure 34).

Exemple 4. Dans notre exemple simplifié, selon le schéma d'agrégation (Figure 31 (b)), les précipitations 'Précip' par département et par jour peuvent être calculées en utilisant la fonction d'agrégation multiple hiérarchique 'SELECT_CENTER(Niv_Adm, Précip)' sur la hiérarchie 'Hgéo_simp' ou bien en utilisant la fonction d'agrégation différenciée 'AVG(Précip)' sur la hiérarchie 'Hgéo_Scien'. Evidemment, chaque fonction donne des résultats différents. Donc, deux nœuds pour les précipitations départementales quotidiennes seront dans le treillis. Afin de distinguer ces deux nœuds, nous ajoutons aux noms des paramètres, dans les noms des nœuds, des abréviations des hiérarchies correspondantes : 'HSi' pour 'Hgéo_simp' et 'HSc' pour 'Hgéo_Scien' (Figure 34).

Le nombre total de nœuds du treillis (la taille du treillis) dépend du nombre des paramètres des hiérarchies dans les dimensions. Pour un treillis uni-fonction, (c'est-à-dire, sans considération des schémas d'agrégation) construit en fonction de *m* dimensions (Figure 32), ce nombre est calculé en multipliant les nombres de paramètres de chaque dimension [Han et al., 1998], [Ghozzi, 2004], [Li et al., 2005] :

$$Le\ nombre\ de\ nœuds_{uni-fonction} = \prod_{i=1}^{m} |P^{D_i}|$$

Mais dans notre modèle multifonctions, en supposant que chaque hiérarchie ou chaque paramètre, possède sa propre fonction d'agrégation pour agréger la mesure concernée, le nombre de nœuds du treillis (Figure 34) est calculé en multipliant les sommes (des nombres de paramètres de chaque hiérarchie) dans chaque dimension. Ici, nous devons être prudents pour ne pas compter le paramètre racine d'une dimension plusieurs fois avec les différentes hiérarchies :

$$Le\ nombre\ de\ n\text{œ}uds_{multifonctions} = \prod_{i=1}^{m}(\sum_{j=1}^{S_j}(|P^{H_j}| - 1) + 1)$$

Nous appliquons cette formule à notre exemple simplifié (Figure 31) où chaque hiérarchie a trois paramètres ; ainsi :

le nombre de nœuds = $((3\text{-}1)_{\text{Hgéo\_Scien}}+(3\text{-}1)_{\text{Hgéo\_Simp}}+1)_{\text{Géographie}} \times ((3\text{-}1)_{\text{Hannée}}+1)_{\text{Dates}} = 15$.

Dans le cas où, un ou plusieurs paramètres communs à plusieurs hiérarchies ont les mêmes fonctions d'agrégation sur les différentes hiérarchies, alors le nombre de nœuds du treillis est situé entre les deux cas précédents :

$$\prod_{i=1}^{m}|P^{D_i}| \leq le\ nombre\ de\ n\text{œ}uds \leq \prod_{i=1}^{m}(\sum_{j=1}^{S_j}(|P^{H_j}| - 1) + 1)$$

La sélection de l'ensemble de vues à matérialiser devient plus difficile avec l'augmentation du nombre de nœuds du treillis. En outre, cette augmentation du nombre de nœuds peut correspondre à la modification de sa forme. Si un niveau extrémité 'All^{Di}' est associé à deux fonctions d'agrégation différentes sur deux hiérarchies distinctes pour agréger la même mesure, alors le treillis aura deux nœuds finaux. Donc, le treillis d'optimisation devient un graphe. Par soucis de cohérence avec de précédents travaux, nous gardons le terme *treillis* dans la suite du mémoire.

Exemple 5. Dans notre exemple simplifié, les précipitations au niveau 'All^{Géographie}' sont agrégées selon deux fonctions d'agrégation ('AVG_W(Précip, D_Superficie)' et 'SELECT_CENTER(Niv_Adm, Précip)') sur les deux hiérarchies 'Hgéo_Scien' et 'Hgéo_Simp'. Donc, le treillis (Figure 34) a deux nœuds extrémités afin de combiner l'agrégation des précipitations au niveau 'All^{Dates}' avec ces deux agrégations différentes au niveau 'All^{Géographie}'.

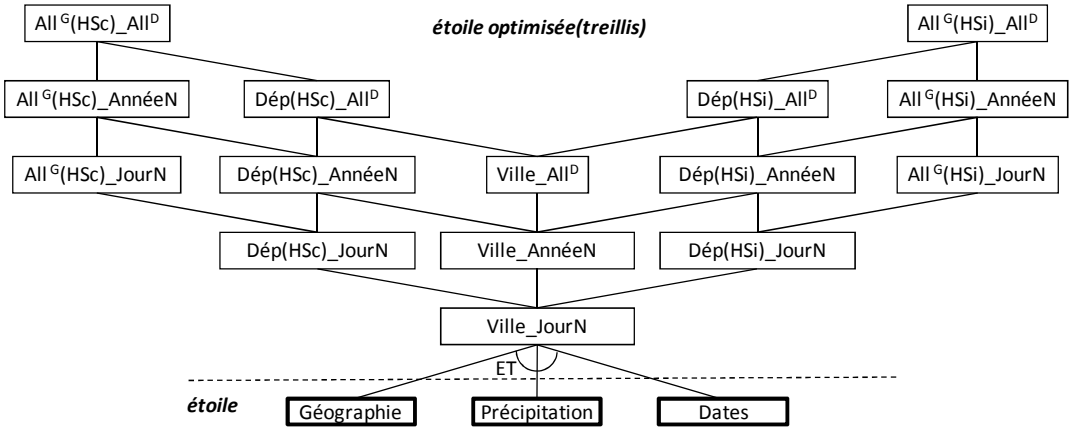


Figure 34 : Treillis d'optimisation multifonctions[18]

[18] Ici, nous avons utilisé les abréviations qui sont entre parenthèses dans la Figure 31 et 'HSi' pour 'Hgéo_simp' et 'HSc' pour 'Hgéo_Scien'.

Ainsi, le nombre de nœuds extrémités est calculé par la multiplication des nombres d'agrégations différentes au niveau 'All^{Di}' sur chaque dimension.

4.4.2 Typage des arcs

Comme l'illustre la Figure 35, la possibilité pour une même mesure d'utiliser les fonctions d'agrégation multiples et différenciées nécessite de typer les arcs du treillis, contrairement au treillis uni-fonction qui considère une fonction d'agrégation unique. Ce typage permet d'indiquer entre deux nœuds la fonction d'agrégation correspondante.

Exemple 6. Selon le schéma d'agrégation de notre exemple simplifié (Figure 31 (b)), cinq fonctions d'agrégation sont utilisées afin d'agréger les précipitations. Pour le calcul des précipitations annuelles nous utilisons la somme (SUM), tandis que le calcul des précipitations générales (niveau 'All^{Dates}') sur la dimension 'Dates' se fait en utilisant la moyenne (AVG). Dans le treillis, il faut donc distinguer les arcs qui relient les nœuds faisant intervenir le paramètre 'JourN', aux nœuds faisant intervenir le paramètre 'AnnéeN' (qui utilisent la fonction SUM), de ceux qui relient les nœuds faisant intervenir le paramètre 'AnnéeN', aux nœuds faisant intervenir le niveau 'All^{Dates}' (qui utilisent la fonction AVG). De la même manière, il faut distinguer :

- Les arcs entre les paramètres ('Ville' et 'Département'), qui utilisent la fonction moyenne (AVG), des arcs entre les paramètres ('Département' et 'All^{Géographie}'), qui utilisent la moyenne en pondérant par la superficie des départements (AVG_W), sur la hiérarchie 'Hgéo_Scien',
- Les arcs entre les paramètres ('Ville' et 'Département') et les paramètres ('Département' et 'All^{Géographie}') sur la hiérarchie 'Hgéo_Simp', qui utilisent la fonction (SELECT_CENTER) afin de réaliser l'agrégation simple, des autres arcs qui utilisent les fonctions somme, moyenne et moyenne pondérée.

Dans la Figure 35, les arcs correspondant aux fonctions d'agrégation moyenne (AVG), moyenne pondérée (AVG_W) et somme (SUM) apparaissent respectivement en trait simple, en double et triple trait tandis que les arcs correspondant à la fonction (SELECT_CENTER) apparaissent en tiret.

Lorsque plusieurs chemins sont possibles pour calculer une vue, le chemin le moins coûteux est préféré. La fonction de coût, que nous ne détaillons pas, privilégie les temps de calcul les plus efficaces [Kotidis & Roussopoulos, 1999]. Nous pouvons néanmoins remarquer que l'utilisation de fonctions d'agrégation différentes sur chaque arc rend l'estimation du coût plus complexe que dans les treillis uni-fonctions habituels.

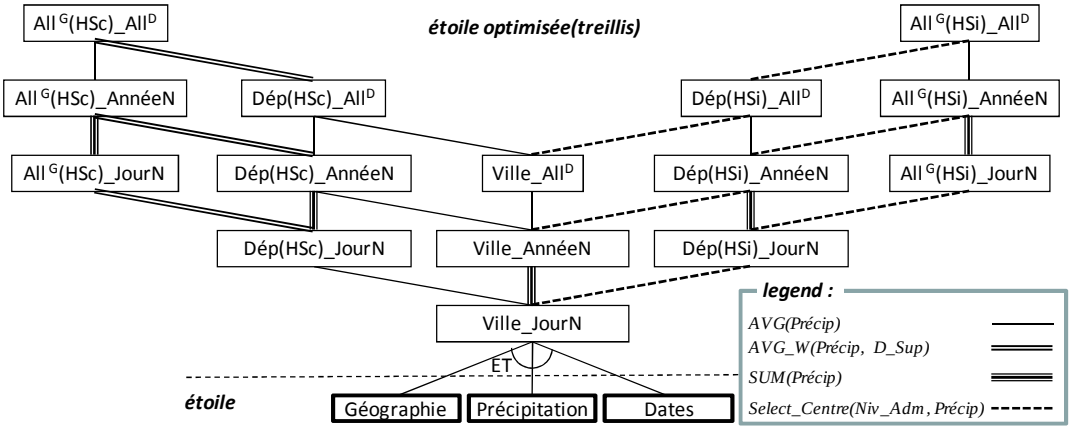


Figure 35 : Treillis d'optimisation multifonctions avec des arcs typés

4.4.3 Modification d'arcs

Dans notre modèle, nous avons proposé un mécanisme de contrainte sur l'agrégation pour fixer le niveau d'agrégation valide à partir duquel se calcule une agrégation supérieure. Ce niveau valide n'est pas forcément le niveau directement inférieur. Nous exprimons ce cas lorsque nous utilisons une valeur de la contrainte différente de 0 ou -1, où :

- Pour la valeur (0), l'agrégation est calculable à partir de n'importe quel niveau inférieur,
- Pour la valeur (-1), l'agrégation est calculable uniquement à partir du niveau directement inférieur.

Les contraintes différentes de 0 et -1 induisent des changements de chemins du calcul (arcs) dans le treillis : les arcs correspondant aux fonctions ayant des contraintes autres que 0 et -1 ne relient pas les nœuds concernés aux nœuds directement inférieurs mais ils les relient aux nœuds plus éloignés qui correspondent aux paramètres valides pour le calcul.

Exemple 7. Dans notre exemple (Figure 31), les précipitations générales (niveau 'All^{Dates}') sont calculées à partir des précipitations annuelles (valeur de contrainte = -1) par la fonction moyenne (AVG). Dans l'hypothèse où nous aurions choisi de calculer ces précipitations générales à partir des précipitations quotidiennes, la contrainte de la fonction moyenne aurait été fixée à -2. Donc, tous les arcs représentant cette fonction et reliant les nœuds correspondants au niveau 'All^{Dates}' aux nœuds correspondants au paramètre 'AnnéeN', seraient modifiés et relieraient les nœuds correspondants au niveau 'All^{Dates}' aux nœuds correspondants au niveau 'JourN' directement. Alors, le treillis correspondrait à la Figure 36 où les arcs modifiés sont présentés par des lignes courbes.

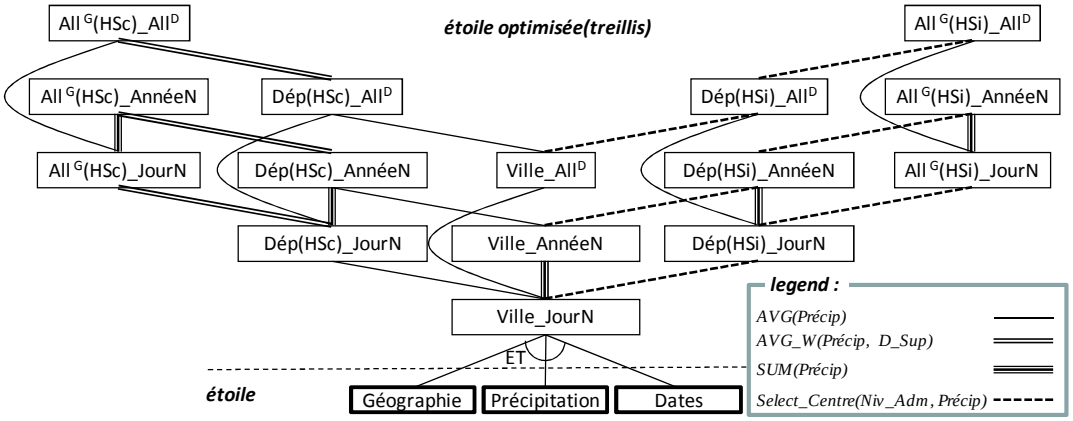


Figure 36 : Treillis d'optimisation multifonctions avec contraint = -2

Ici, il est important d'indiquer que l'application des deux derniers impacts (le typage des arcs et la modification d'arcs) est nécessaire pour réaliser les prochains impacts (le blocage de la transitivité et l'élagage du treillis). Ces deux derniers s'appuient sur l'ordre des fonctions d'agrégation et l'ordre des valeurs d'ordre d'exécution dans le treillis. D'un côté, le typage des arcs définit l'ordre des fonctions d'agrégation dans le treillis. D'un autre côté, la modification d'arcs change cet ordre.

Par ailleurs, les valeurs d'ordre d'exécution peuvent être définies dans le treillis au cours du typage des arcs. L'ordre de ces valeurs peut être différent de l'ordre des fonctions d'agrégation car :

- Deux fonctions d'agrégation différentes peuvent avoir la même valeur d'ordre d'exécution. Par exemple, les fonctions 'SUM' et 'SELECT_CENTER' dans notre exemple simplifié (Figure 31) ont la même valeur d'ordre d'exécution (1),
- Une fonction d'agrégation peut être définie deux fois dans le schéma d'agrégation avec deux valeurs d'ordre d'exécution différentes.

Dans la Figure 37, les ordres d'exécution ont été associés aux arcs afin de faciliter la compréhension des prochains impacts (le blocage de la transitivité et l'élagage du réseau).

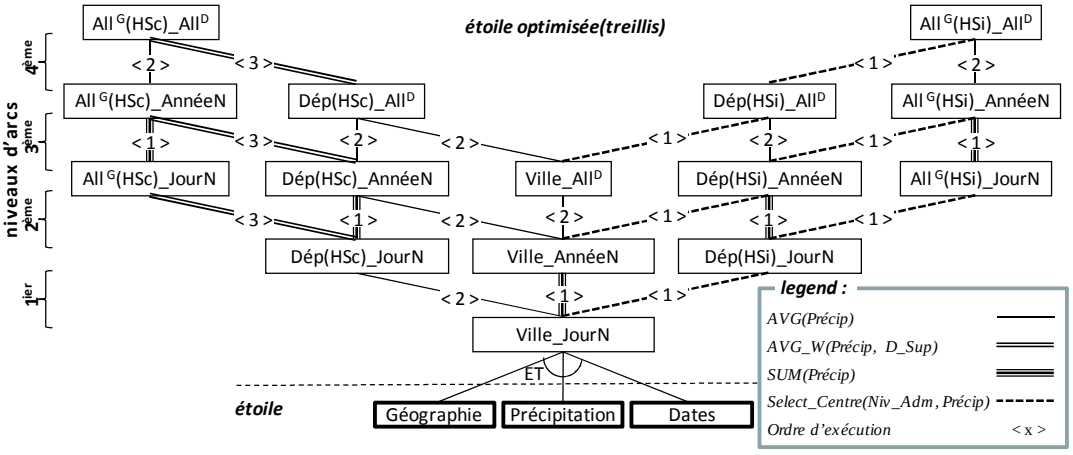


Figure 37 : Treillis d'optimisation multifonctions avec l'ordre d'exécution

4.4.4 Elagage du treillis

Dans notre modèle conceptuel multidimensionnel multifonctions, il y a un ordre entre les fonctions d'agrégation. Il s'agit de l'ordre avec lequel les fonctions d'agrégation doivent être appliquées. Cet ordre d'exécution est utilisé pour éliminer les combinaisons des fonctions d'agrégation invalides qui donnent des résultats incohérents. Ces combinaisons interdites correspondent aux chemins ou arcs dans le treillis. Par conséquent ces arcs sont invalides et peuvent ainsi être éliminés pour réduire le treillis. Cette élimination diminue le nombre de chemins alternatifs dans le treillis. Cela facilite la tâche de la fonction de coût pour trouver le chemin le moins coûteux.

Exemple 8. Dans notre exemple (Figure 31), on ne peut pas appliquer la fonction 'SUM(Précip)' sur la dimension 'Dates' (avec un ordre d'exécution de valeur 1) après la fonction 'AVG(Précip)' sur la hiérarchie 'Hgéo_Scien' de la dimension 'Géographie' (avec un ordre d'exécution de valeur 2) car cela donnerait un résultat invalide. Ainsi, pour obtenir le nœud 'Dép(HSc)_AnnéeN' (les précipitations départementales annuelles selon l'agrégation scientifique), on ne peut pas le calculer à partir du nœud 'Dép(HSc)_JourN' (les précipitations départementales quotidiennes selon l'agrégation scientifique). L'arc entre les nœuds 'Dép(HSc)_JourN' et 'Dép(HSc)_AnnéeN' peut donc être supprimé.

Afin d'éliminer tous les arcs invalides, nous pouvons nous appuyer sur le treillis d'optimisation multifonctions avec l'ordre d'exécution (Figure 37). Sur tous les chemins possibles, à partir du nœud racine (représentant les données en fonction des niveaux de base de toutes les dimensions [Han et al., 1998]) jusqu'au nœud final (représentant l'agrégation totale des données), il est interdit de passer d'un arc associé à un ordre d'exécution supérieur à un arc associé à un ordre d'exécution inférieur. Pour effectuer cet élagage, il faut donc vérifier si les arcs précédents pour un certain arc sont associés à des ordres d'exécution supérieurs, alors cet arc doit être supprimé. Ainsi, aucun arc au premier niveau (qui relie le nœud racine aux nœuds directement supérieurs (Figure 37)) ne doit être supprimé. Ce processus de vérification doit être

réalisé simplement à partir du deuxième niveau d'arcs du treillis vers le haut, niveau après niveau.

En appliquant ce processus d'élagage au treillis de notre exemple (Figure 37) :

- Au deuxième niveau d'arcs : l'arc ('Dép(HSc)_JourN', 'Dép(HSc)_AnnéeN') doit être supprimé parce qu'il est associé à un ordre d'exécution de valeur 1 et il est précédé par un seul arc ayant un ordre d'exécution de valeur 2 ;
- Au troisième niveau d'arcs : de la même manière, il faut supprimer les arcs entre les nœuds 'All^G(HSc)_JourN' et 'All^G(HSc)_AnnéeN' et les nœuds 'Ville_All^D' et 'Dép(HSi)_All^D' ;
- Au quatrième niveau d'arcs : après la suppression des arcs au troisième niveau, les deux arcs ('All^G(HSc)_AnnéeN', 'All^G(HSc)_All^D') et ('Dép(HSi)_All^D', 'All^G(HSi)_All^D'), ayant l'ordre d'exécution de valeur 2 et 1 respectivement, deviennent précédés par un seul arc ('Dép(HSc)_AnnéeN', 'All^G(HSc)_AnnéeN') et ('Dép(HSi)_AnnéeN', 'Dép(HSi)_All^D') respectivement qui ont des ordres d'exécution supérieurs 3 et 2 respectivement. Donc, ils doivent être éliminés.

La Figure 38 présente le treillis après suppression des arcs invalides.

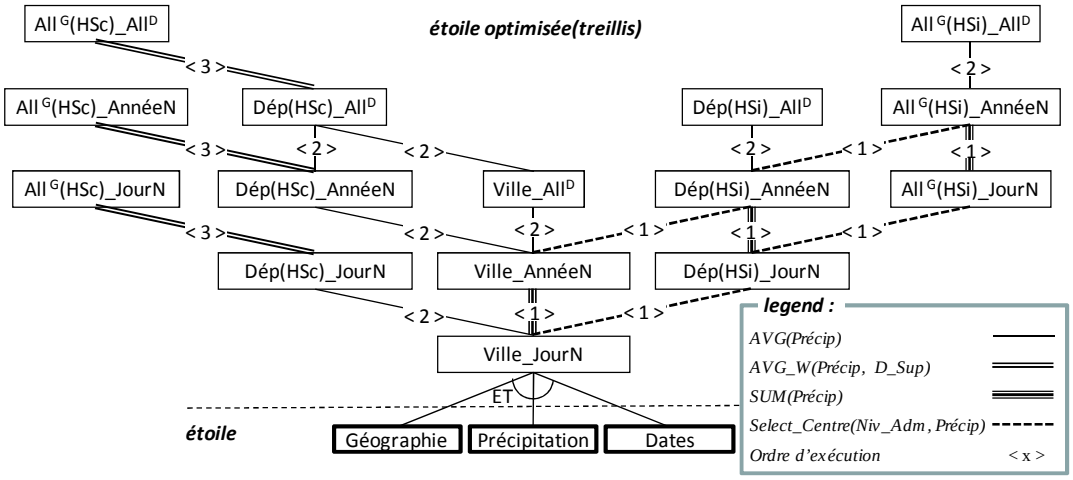


Figure 38 : Treillis d'optimisation multifonctions avec des arcs élagués

Par ailleurs, si nous ne respectons pas la condition de la cohérence entre les contraintes d'agrégation et l'ordre d'exécution (*cf.* § 3.3.2), l'élagage risque de diviser le treillis. Ce qui rend le calcul de certains nœuds impossible.

La Figure 39 présente un exemple d'un treillis sans cette cohérence. La Figure 39 (a) montre le schéma d'agrégation à partir duquel le treillis de la Figure 39 (b) est construit. Dans ce schéma d'agrégation les fonctions F1 et F3 ne respectent pas la condition de cohérence entre les contraintes d'agrégation et l'ordre d'exécution où la fonction F3 a un ordre d'exécution de valeur 1 qui est inférieur à l'ordre d'exécution de la fonction F1 de valeur 2, par contre les contraintes d'agrégation exigent d'appliquer F1 avant F3. A cause de ces valeurs d'ordre d'exécution et contraintes d'agrégation, les arcs en gris dans la Figure 39 (b) devraient être supprimés selon le processus d'élagage présenté au-dessus. Le treillis se divise alors en deux parties rendant une partie des calculs impossibles.

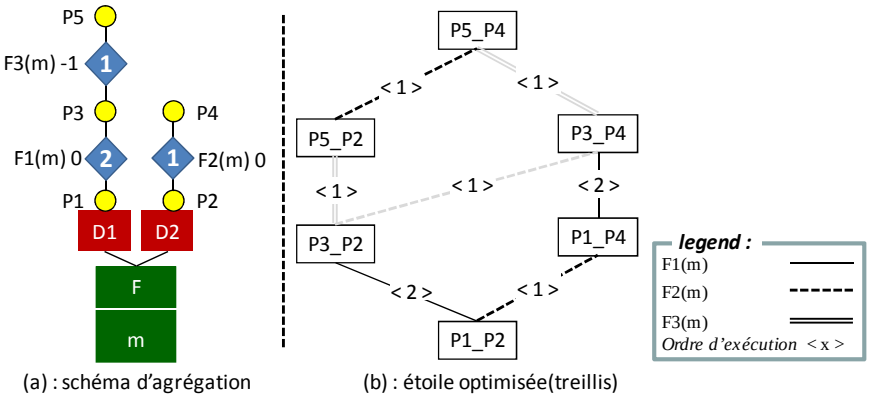


Figure 39 : Treillis sans la cohérence entre les contraintes d'agrégation et l'ordre d'exécution

4.4.5 Blocage de la transitivité

Les contraintes (spécification d'un niveau précis à partir duquel l'agrégation considérée doit être calculée) associées aux fonctions d'agrégation ont une autre répercussion sur le treillis : les arcs obtenus à partir de ces fonctions contraintes, qui ont une contrainte d'agrégation autre que zéro, imposent de calculer un nœud à partir d'un nœud précis. Il est alors interdit de calculer un nœud supérieur par transitivité des nœuds inférieurs comme cela est possible dans le treillis uni-fonction [Gupta, 1997]. Ainsi les chemins de calcul sont bloqués dès qu'un arc contraint intervient.

Exemple 9. Dans le treillis de notre exemple (Figure 40), le nœud 'All^G(HSi)_JourN' est calculable à partir du nœud inférieur direct 'Dép(HSi)_JourN', par transitivité, il est également calculable à partir du nœud inférieur 'Ville_JourN'. Par contre, l'arc contraint issu de la contrainte de la fonction 'AVG_W(Précip, D_Superficie)' qui opère sur l'arc ('Dép(HSc)_JourN', 'All^G(HSc)_JourN') bloque la transitivité des calculs. Donc, le nœud 'All^G(HSc)_JourN' est calculable à partir du nœud inférieur direct 'Dép(HSc)_JourN' mais il ne l'est pas par transitivité à partir du nœud inférieur 'Ville_JourN'.

De la même manière, le changement des ordres d'exécution ou des fonctions entre les arcs provoque un blocage de la transitivité. Autrement dit, si tous les arcs précédents pour un arc spécifique correspondent à des fonctions et/ou des ordres d'exécution différents, alors cet arc est non transitif.

Exemple 10. Dans la Figure 38, l'arc ('Ville_AnnéeN', 'Dép(HSc)_AnnéeN') correspond à la fonction 'AVG(Précip)' avec un ordre d'exécution de valeur 2. Cet arc a un seul arc précédent ('Ville_JourN', 'Ville_AnnéeN') qui correspond à la fonction 'SUM(Précip)' avec un ordre d'exécution de valeur 1. A cause de la différence entre les ordres d'exécution l'arc ('Ville_AnnéeN', 'Dép(HSc)_AnnéeN') est non transitif (Figure 40). Donc, le nœud 'Dép(HSc)_All^D' est calculable par transitivité à partir du nœud 'Ville_AnnéeN' mais il n'est pas calculable par transitivité à partir du nœud 'Ville_JourN'. En effet, le schéma d'agrégation (Figure 31 (b)) impose à l'ordre d'exécution de calculer d'abord les précipitations annuelles (nœud 'Ville_AnnéeN') pour pouvoir calculer ensuite les précipitations départementales selon l'agrégation scientifique (nœud 'Dép(HSc)_All^D').

La Figure 40 décrit le treillis d'optimisation multifonctions contrôlé final dans lequel les arcs étiquetés par un cercle barré sont obtenus soit à partir des contraintes d'agrégation, soit à partir du changement d'ordre d'exécution ou de fonction d'agrégation entre les arcs.

Dans ce treillis, nous pouvons distinguer deux comportements différents pour les arcs contraints par rapport aux arcs des niveaux supérieurs :

- Si l'arc est suivi, au niveau supérieur, par un arc ayant le même type de fonction d'agrégation et le même ordre d'exécution, alors la transitivité du calcul est bloquée au-dessous de l'arc considéré. Par exemple, l'arc contraint ('Ville_AnnéeN', 'Dép(HSi)_AnnéeN') est suivi par l'arc ('Dép(HSi)_AnnéeN', 'All^{G}(HSi)_AnnéeN'). Ces deux arcs correspondent au même type de fonction d'agrégation 'SELECT_CENTER(Niv_Adm, Précip)' et même ordre d'exécution de valeur 1. Donc, le nœud 'All^{G}(HSi)_AnnéeN' peut être calculé par transitivité à partir de nœud 'Ville_AnnéeN' qui se trouve en bas de l'arc contraint considéré mais il ne peut pas être calculé à partir de nœuds inférieurs comme 'Ville_JourN' ;
- Si l'arc est suivi, au niveau supérieur, par un arc d'autre type de fonction d'agrégation et/ou d'autre ordre d'exécution, alors la transitivité du calcul est bloquée à partir de l'arc considéré lui-même. Par exemple, l'arc contraint ('Dép(HSi)_JourN', 'Dép(HSi)_AnnéeN') est suivi par le même arc de l'exemple précédent ('Dép(HSi)_AnnéeN', 'All^{G}(HSi)_AnnéeN'). Les deux arcs correspondent aux types de fonction d'agrégation différents 'SUM(Précip)' et 'SELECT_CENTER(Niv_Adm, Précip)' respectivement. Donc, le nœud 'All^{G}(HSi)_AnnéeN' peut être calculé à partir d'un nœud inférieur direct 'Dép(HSi)_AnnéeN' mais il ne peut pas être calculé par transitivité à partir de nœud inférieur 'Dép(HSi)_JourN' qui se trouve en bas de l'arc contraint considéré.

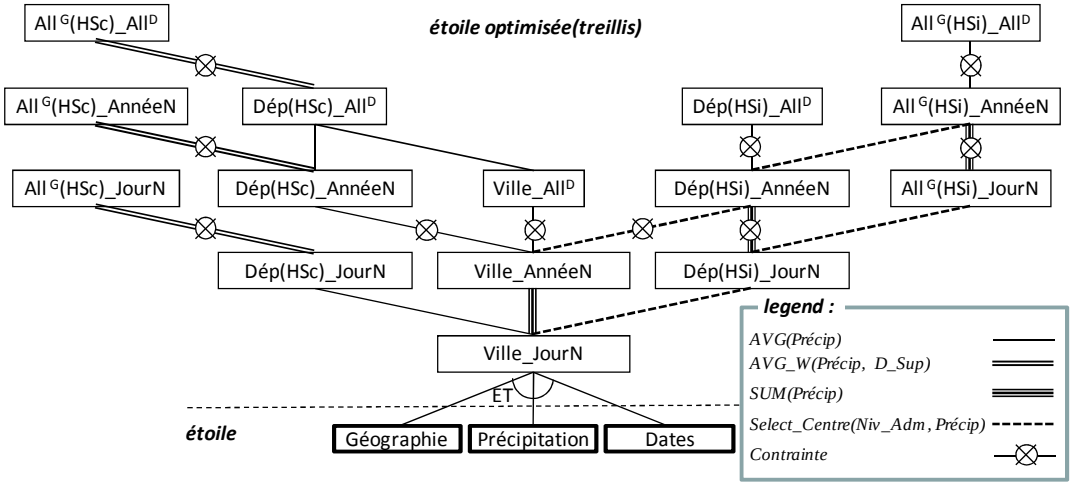


Figure 40 : Treillis d'optimisation multifonctions contrôlé (avec des arcs contraints)

4.4.6 Changement des données stockées

A l'instar du stockage des vues matérialisées du treillis uni-fonction, nous supposons que les données des nœuds du treillis multifonctions sont stockées dans des relations. La Figure 41 décrit ces relations (schéma et contenu) d'optimisation de notre exemple simplifié (Figure 31).

En comparant les données des vues matérialisées du treillis uni-fonction (Figure 33) avec les données des vues matérialisées du treillis multifonctions (Figure 41), nous pouvons remarquer que l'utilisation de plusieurs fonctions d'agrégation pour la même mesure peut affecter également les données (le contenu des nœuds).

D'un côté, dans ce treillis multifonctions, de nouvelles données ('Niveau_Administratif' et 'D_Superficie') sont stockées dans certains nœuds. Ces données

correspondent aux arguments des fonctions d'agrégations (AVG_W, SELECT_CENTER) qui suivent ces nœuds (c'est-à-dire, appliquées à ces nœuds).

D'un autre côté, bien que la fonction algébrique (AVG) soit utilisée sur sept arcs, il n'y a que deux nœuds ('Ville_All^D' et 'Dép(HSc)_AnnéeN') dans lesquels les valeurs intermédiaires (Pré_sum et Pré_count) sont stockées. Car, pour que le stockage des valeurs intermédiaires d'une fonction algébrique soit utile, il faut avoir au moins deux arcs successifs correspondant à cette fonction à condition que l'arc au niveau supérieur soit transitif (non contraint). Ainsi, les contraintes d'agrégation, l'élagage du treillis et l'utilisation de plusieurs fonctions d'agrégation, qui produisent des changements de types entre les arcs successifs, rendent plus difficile l'atteinte de cette condition, ce qui peut diminuer les données intermédiaires stockées.

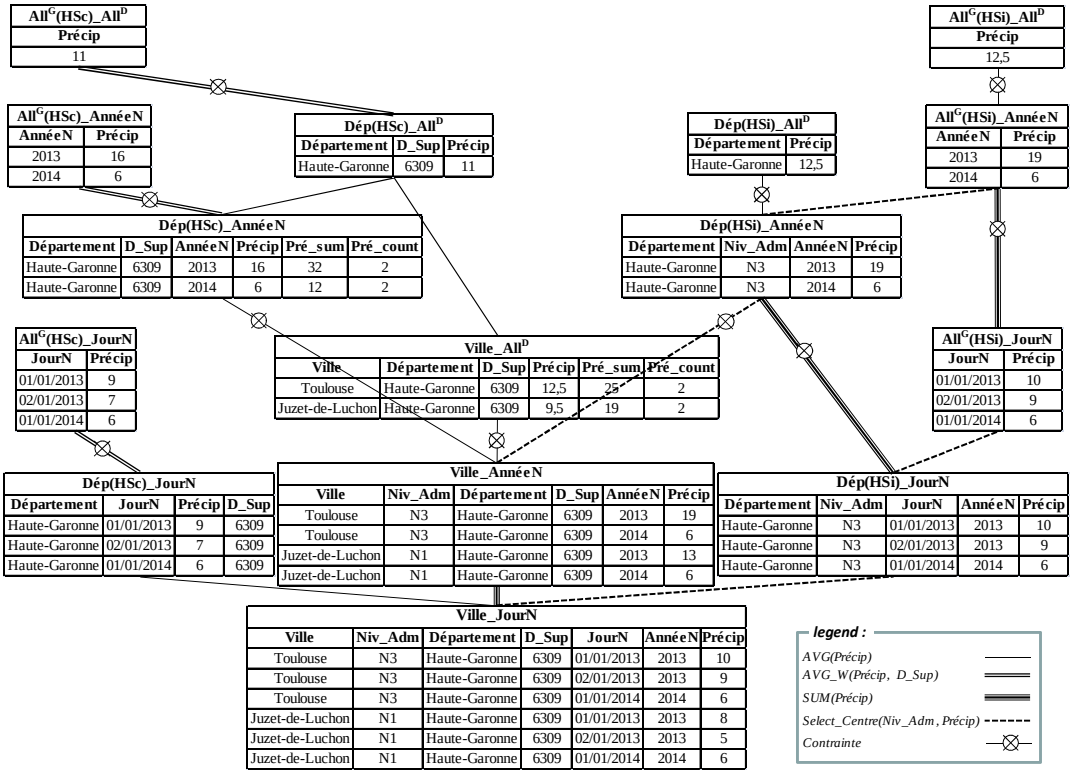
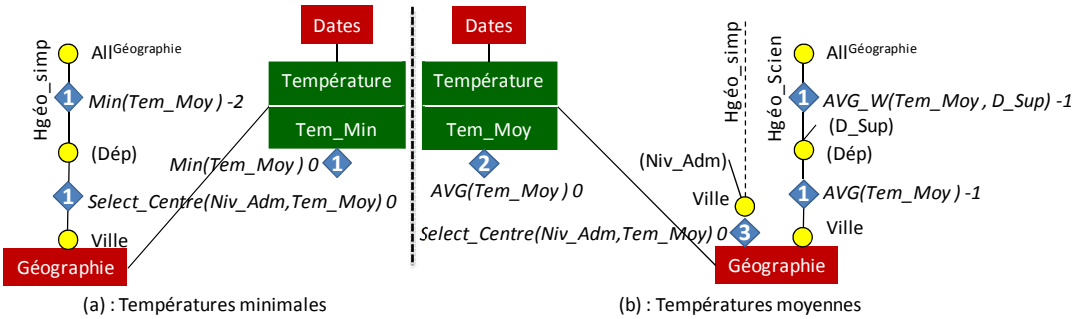


Figure 41 : Les relations du treillis d'optimisation multifonctions contrôlé

4.4.7 Différences entre les treillis d'optimisation des mesures analysées selon les mêmes dimensions

Dans le cas de plusieurs mesures analysées selon les mêmes dimensions, les treillis uni-fonctions de toutes ces mesures sont identiques [Li et al., 2005]. Dans notre modèle multifonctions, les différences entre les fonctions d'agrégation de chaque mesure génèrent forcement des différences entre les treillis d'optimisation. C'est pourquoi nous proposons d'utiliser un treillis d'optimisation différent pour chaque mesure.

Afin d'illustrer les différences entre ces treillis, nous utilisons un fait 'Température' ayant deux mesures : températures minimales 'Tem_Min' et températures moyennes 'Tem_Moy'. Le schéma structurel correspondant à ce fait est similaire au schéma structurel de l'exemple simplifié présenté précédemment Figure 31 (a) mais avec le fait 'Température' à la place du fait 'Précipitation'. Les schémas d'agrégation des mesures 'Tem_Min' et 'Tem_Moy' sont présentés dans la Figure 42 (a, b) respectivement.



(a) : Températures minimales (b) : Températures moyennes

Figure 42 : Schémas d'agrégation des températures simplifiés[19]

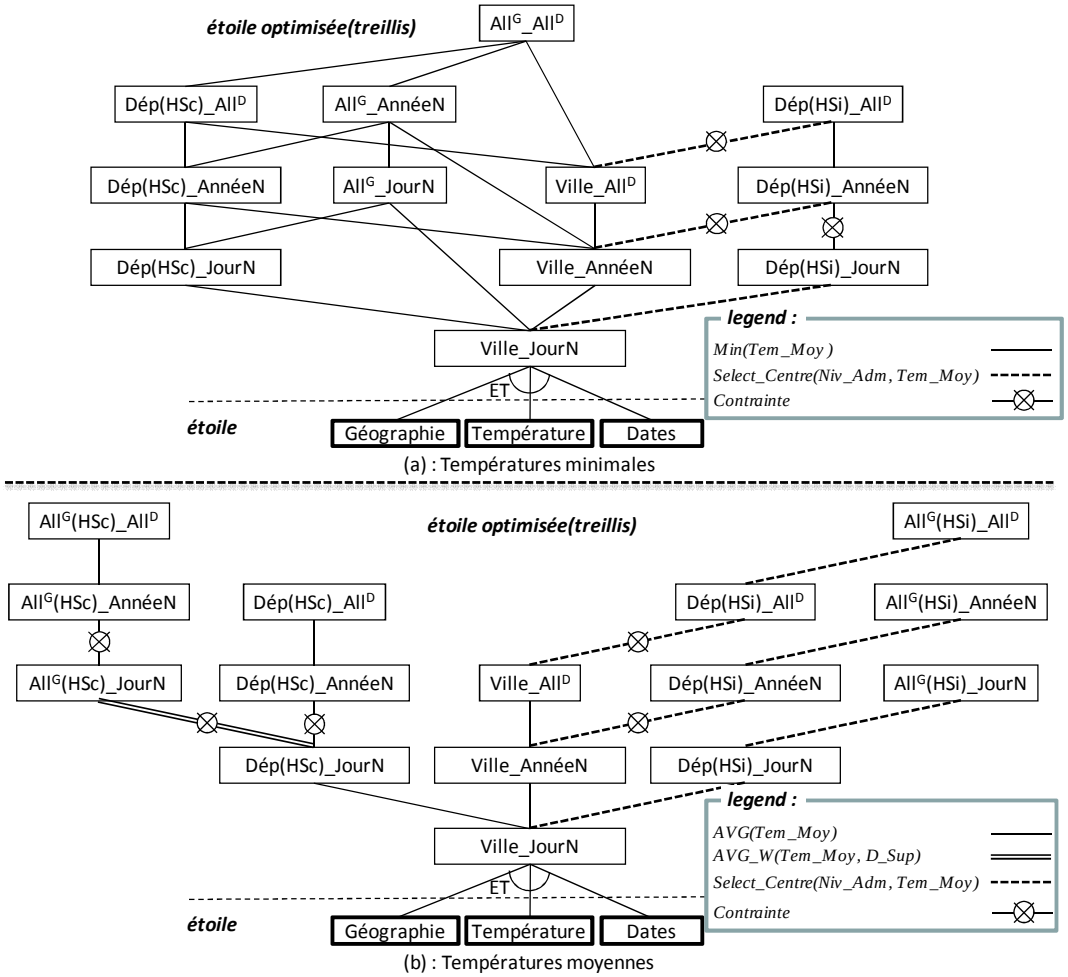


Figure 43 : Treillis d'optimisation des températures moyennes et minimales

[19] Ici, nous avons utilisé les abréviations qui sont entre parenthèses dans la Figure 31.

En s'appuyant sur ces schémas, nous comparons les treillis d'optimisation des mesures 'Tem_Min' (Figure 43 (a)), 'Tem_Moy' (Figure 43 (b)) et 'Précip' de l'exemple simplifié présenté précédemment (Figure 40).

Dans le schéma d'agrégation de la mesure 'Tem_Min' (Figure 42 (a)), deux fonctions d'agrégation sont utilisées au niveau 'Département' :

- La fonction différenciée 'SELECT_CENTER(Niv_Adm,Tem_Moy)' sur la hiérarchie 'Hgéo_simp',
- La fonction générale 'MIN(Tem_Moy)' sur la hiérarchie 'Hgéo_Scien'.

Par contre, les températures minimales sont agrégées au niveau 'All$^{\text{Géographie}}$' sur les deux hiérarchies 'Hgéo_simp' et 'Hgéo_Scien' par la même fonction d'agrégation qui est la fonction 'MIN(Tem_Moy)'. C'est ce qui rend les valeurs des températures minimales 'Tem_Min' d'un côté, différentes entre les deux hiérarchies au niveau 'Département' et de l'autre, identiques au niveau 'All$^{\text{Géographie}}$'. Donc, deux nœuds pour les températures minimales départementales en fonction de chaque paramètre de la dimension 'Dates' coexistent dans le treillis contrairement à un seul nœud pour les températures minimales générales (correspondant au niveau 'All$^{\text{Géographie}}$') (Figure 43 (a)). En outre, selon le schéma d'agrégation des températures moyennes 'Tem_Moy' (Figure 42 (b)), les agrégations aux niveaux 'Département' et 'All$^{\text{Géographie}}$' sont différentes entre les deux hiérarchies 'Hgéo_simp' et 'Hgéo_Scien'. Donc, deux nœuds pour chaque paramètre de la dimension 'Géographie' en fonction de chaque paramètre de la dimension 'Dates' seront présents dans le treillis d'optimisation.

Comme le montre la Figure 43, à cause des différences entre les fonctions d'agrégation des mesures du fait 'Température', le nombre de nœuds et la forme du treillis d'optimisation peuvent être différents d'une mesure à l'autre. Le treillis des températures minimales 'Tem_Min' a un seul nœud final (borne supérieure) avec 12 nœuds tandis que le treillis des températures moyennes 'Tem_Moy' a deux nœuds finaux avec 15 nœuds.

D'un autre côté, même si les treillis ont le même nombre de nœuds et la même forme, il peut exister d'autres différences concernant les relations du calcul entre les nœuds (les arcs). Les contraintes sur les fonctions d'agrégation d'une mesure peuvent modifier les arcs du treillis autrement que ceux d'une autre mesure analysée selon les mêmes dimensions. De la même façon, à cause de l'ordre d'exécution, des arcs peuvent être supprimés dans le treillis d'optimisation d'une mesure autrement que ceux du treillis d'une autre mesure analysée selon les mêmes dimensions. Enfin, le blocage de transitivité de calcul peut varier d'un treillis à l'autre. Nous pouvons remarquer de telles différences en comparant le treillis des précipitations 'Précip' (Figure 40) avec le treillis des températures moyennes 'Tem_Moy' (Figure 43 (b)) en sachant que les deux treillis ont le même nombre de nœuds et la même forme. Il convient de noter ici que nous pouvons comparer les treillis de ces deux dernières mesures, malgré qu'elles ne soient pas du même fait, parce que les deux mesures sont analysées selon les mêmes dimensions ('Géographie' et 'Dates').

Par ailleurs, si les fonctions d'agrégation sur chaque dimension ont un ordre d'exécution différent des autres dimensions, alors le treillis d'optimisation n'est plus un graphe, mais un arbre. Par conséquent, pour calculer chaque nœud, il n'y a qu'un seul chemin d'accès à partir du nœud racine. Cela réduit le nombre d'arcs et rend certains nœuds critiques car de nombreux nœuds nécessitent le calcul de ces nœuds. Donc, ces nœuds deviennent des candidats plus importants à matérialiser.

Exemple 11. Dans le schéma d'agrégation des températures moyennes 'Tem_Moy', il n'y a aucune valeur d'ordre d'exécution commune sur les deux dimensions 'Géographie' et 'Dates'. La dimension 'Géographie' est associée aux fonctions d'agrégation ayant des ordres d'exécution de valeurs 1 et 3 tandis que la fonction d'agrégation appliquée sur la dimension 'Dates' a un ordre d'exécution de valeur 2. Donc, le treillis d'optimisation des températures

moyennes devient un arbre, comme le montre la Figure 43 (b). Par conséquence, les nœuds 'Dép(HSc)_JourN' et 'Ville_AnnéeN' deviennent vitaux.

4.5 CONCLUSION

Dans ce chapitre, nous avons étudié les conséquences de l'utilisation de plusieurs fonctions d'agrégation prédéfinies au niveau conceptuel sur le stockage de données de base (faits et dimensions) et de données d'optimisation (vues matérialisées) au niveau logique.

A l'instar du stockage R-OLAP [Kimball, 1996], [Dinter et al., 1998], [Mangisengi & Tjoa, 1998], nous avons proposé de stocker les données des faits et des dimensions du modèle multifonctions dans des tables relationnelles [Hassan et al., 2012a], [Hassan et al., 2012b], [Hassan et al., 2012c], [Hassan et al., 2013a], [Hassan et al., 2014]. Chaque dimension et chaque fait sont stockés dans une table relationnelle où les tables des faits ont des clés étrangères référençant les tables des dimensions formant ainsi un schéma logique dit schéma en étoile [Kimball, 1996]. Un schéma en flocon peut être obtenu en normalisant les tables des dimensions pour transformer chaque dimension en plusieurs tables relationnelles en fonction de ses paramètres. Ainsi, grâce à l'absence d'influence de la pluralité des fonctions d'agrégation sur le stockage des données de base, notre modèle multifonction peut s'appliquer sur les systèmes décisionnels sans modifier ces données.

Contrairement aux données de base, notre modèle multifonction a des impacts sur les données d'optimisation qui sont les pré-agrégats calculés et stockés (vues matérialisées). Toutes les agrégations possibles en fonction de différentes combinaisons de paramètres peuvent être modélisées par un treillis [Harinarayan et al., 1996], [Nandi et al., 2011], où les nœuds représentent les pré-agrégats et les arcs représentent les calculs des agrégations. Nous avons distingué plusieurs impacts de l'exploitation des mécanismes d'agrégation (les fonctions d'agrégation, les contraintes d'agrégation et l'ordre d'exécution) sur le treillis d'optimisation [Hassan et al., 2012a], [Hassan et al., 2012b], [Hassan et al., 2012c], [Hassan et al., 2013a], [Hassan et al., 2014] :

- **Augmentation du nombre de nœuds** : l'existence des fonctions d'agrégation différentes d'une hiérarchie à l'autre sur une partie commune à plusieurs hiérarchies, augmente le nombre de nœuds du treillis. En outre, si cette partie comprend un niveau 'All^{Di}', alors la forme du treillis change à cause de plusieurs nœuds extrémités (bornes supérieures). Cette augmentation du nombre de nœuds peut rendre le choix de vues à matérialiser plus difficile ;
- **Typage des arcs** : comme les arcs correspondent aux calculs de pré-agrégats, les types des arcs changent selon les fonctions d'agrégation qui réalisent ces calculs. Ce typage rend l'estimation du coût plus complexe que dans les treillis uni-fonctions ;
- **Modification d'arcs** : une fonction ayant une valeur de contrainte d'agrégation différente de 0 ou -1, exprime le cas d'une agrégation calculée à partir d'un niveau qui n'est pas directement inférieur. Donc, les arcs correspondants relient les nœuds concernés aux nœuds plus éloignés. Ces derniers correspondent au paramètre valide pour le calcul ;
- **Elagage du treillis** : cet élagage est rendu possible par l'utilisation de l'ordre d'exécution. Grâce à l'interdiction de l'application d'une fonction d'agrégation après une autre fonction ayant un ordre d'exécution supérieur, il est possible de supprimer tous les arcs précédés par des arcs associés aux ordres d'exécution supérieurs. Cette suppression facilite la tâche pour trouver le chemin le moins coûteux ;
- **Blocage de la transitivité** : la transitivité du calcul est bloquée sur les arcs correspondants aux fonctions d'agrégation contraintes (ayant une contrainte

d'agrégation autre que 0) et sur les arcs précédés par des arcs correspondants à des fonctions et/ou des ordres d'exécution différents ;

- **Changement des données stockées** : contrairement au treillis uni-fonction, les valeurs intermédiaires d'une fonction algébrique ne devraient pas être stockées dans tous les nœuds mais seulement dans les nœuds qui sont entre deux arcs correspondant à la fonction algébrique à condition que l'arc supérieur soit transitif ;
- **Différences entre les treillis d'optimisation des mesures analysées selon les mêmes dimensions** : comme chaque mesure a un schéma d'agrégation différent, il est normal que les impacts sur le treillis soient différents d'une mesure à l'autre.

La modification d'arcs, le blocage de la transitivité et le changement des données stockées influencent le choix de vues à matérialiser.

L'étude des impacts de notre modèle multifonctions sur les opérateurs d'analyse OLAP est réalisée dans le chapitre suivant.

5. CHAPITRE V : MANIPULATIONS OLAP MULTIFONCTIONS

5.1 INTRODUCTION

L'exploitation des systèmes décisionnels repose sur l'utilisation d'un langage d'interrogation dédié aux données multidimensionnelles [Marcel, 1998]. Ce langage offre à l'analyste la possibilité de manipuler les indicateurs (mesures) selon plusieurs dimensions. Les décideurs peuvent utiliser les opérateurs OLAP [Ravat et al., 2008] pour étudier les mesures en fonction des différents niveaux de granularité (paramètres) définis sur les dimensions. Les données sont ainsi regroupées selon les niveaux sélectionnés et calculées en utilisant des fonctions d'agrégation. Ces opérateurs permettent d'abord de spécifier une analyse et en ensuite de la modifier.

Dans ce chapitre, nous focalisons notre étude sur l'analyse multidimensionnelle OLAP [Codd et al., 1993] appliquée à une BDM multifonctions afin d'étudier les conséquences de l'utilisation de plusieurs fonctions d'agrégation sur la table multidimensionnelle (TM) et les opérateurs d'analyse OLAP.

Plan du chapitre. Dans ce chapitre, nous présentons d'abord notre problématique et notre contribution pour adapter les opérateurs OLAP à notre modèle multifonctions. La deuxième section détaille la spécification de la TM et notre langage d'interrogation de données multidimensionnelles en considérant l'existence de plusieurs fonctions d'agrégation pour la même mesure.

5.1.1 Problématique

Au cours d'une analyse, les données sont synthétisées en faisant intervenir des fonctions d'agrégation. Le langage d'interrogation devrait assurer la fiabilité des résultats d'analyse [Ghozzi, 2004]. Plusieurs langages d'interrogation ont été proposés dans la littérature [Gray et al., 1996], [Li & Wang, 1996], [Agrawal et al., 1997], [Gyssens & Lakshmanan, 1997], [Thomas & Datta, 1997], [Cabibbo & Torlone, 1997], [Cabibbo & Torlone, 1998], [Lehner, 1998], [Datta & Thomas, 1999], [Vassiliadis, 2000], [Pedersen T.B., 2000], [Pedersen T.B. et al., 2001], [Abelló et al., 2003], [Franconi & Kamble, 2004], [Abelló et al., 2006], [Ravat et al., 2008], [Lenz & Thalheim, 2009], [Boukraa et al., 2010], [Pardillo et al., 2010].

Ces langages proposent un ensemble d'opérateurs interactifs facilitant la navigation entre les données multidimensionnelles analysées [Abelló et al., 2003]. Ces opérateurs sont présentés succinctement dans le Tableau 15.

Tableau 15 : Opérateurs OLAP classiques

Opérateur	Tâche
DISPLAY(F,{$f_1(m_1)$, $f_2(m_2)$, …}, D_L, H_L, D_C, H_C)	Construire une TM présentant les données des mesures (m_1, m_2,…) du fait (F) agrégées par les fonctions (f_1, f_2,…). Cet opérateur se positionne sur le niveau de granularité le plus haut des hiérarchies (H_L, H_C) des dimensions (D_L, D_C) présentées en ligne et en colonne respectivement.
DRILLDOWN(T_{SRC}, D, p_{inf})	Changer le niveau de la granularité dans une TM (T_{SRC}) vers un niveau plus fin (p_{inf}) sur la hiérarchie courante de la dimension (D).
ROLLUP(T_{SRC}, D, p_{sup})	Modifier le niveau de la granularité sur la hiérarchie courante de la dimension (D) vers un niveau moins détaillé (p_{sup}) dans une TM (T_{SRC}).
SELECT(T_{SRC}, pred)	Réduire aux données qui satisfont la condition (pred).
UNSELECT(T_{SRC})	Annuler toutes les sélections.
DROTATE(T_{SRC}, D_{old}, D_{new} [,H_{new}])[20]	Remplacer un axe d'analyse (D_{old}) par un autre (D_{new}) dans une TM (T_{SRC}). Il est possible de préciser la hiérarchie (H_{new}) de la nouvelle dimension pour l'utiliser dans la TM résultante.
HROTATE(T_{SRC}, D, H_{new})	Changer la hiérarchie actuelle par une autre (H_{new}) de la dimension (D) en ligne ou en colonne.
FROTATE (T_{SRC}, F_{new}, {$f'_1(m'_1)$, $f'_2(m'_2)$, …})	Utiliser un nouveau fait (F_{new}) dans la TM (T_{SRC}), en préservant toutes les caractéristiques actuelles des axes d'analyse.
ADDM(T_{SRC}, $f_i(m_i)$)	Ajouter une mesure (m_i) agrégée par la fonction (f_i) dans le fait visualisé par la TM (T_{SRC}).
DELM(T_{SRC}, $f_i(m_i)$)	Supprimer une mesure(m_i) du fait visualisé par la TM (T_{SRC}).
NEST(T_{SRC}, D, p_i, D_{nested}, p_{nested})	Insérer un paramètre (p_{nested}) d'une dimension non-présentée (D_{nested}) dans une dimension présentée (D) dans la TM (T_{SRC}).
PUSH(T_{SRC}, D, p_i)	Convertir un paramètre (p_i) d'une dimension (D) en mesure dans le fait courant de la TM (T_{SRC}).
PULL (T_{SRC}, D, $f_i(m_i)$)	Convertir une mesure (M_i) agrégée par la fonction (f_i) en paramètre dans la dimension courante (D).
SWITCH(T_{SRC}, D, p_i, v_1, v_2)	Permuter deux valeurs (v_1, v_2) d'un paramètre (p_i) d'une dimension courante (D).
ORDER(T_{SRC}, D, p_i, ord)	Ordonner les valeurs d'un paramètre (p_i) d'une dimension courante (D) dans un ordre croissant ou décroissant (ord□{'asc', 'dsc'}).
AGGREGATE(T_{SRC}, D, $f_i(p_i)$)	Agréger les valeurs d'un paramètre (p_i) par une fonction d'agrégation (f_i) dans une nouvelle ligne ou colonne d'une TM (T_{SRC}).
UNAGGREGATE(T_{SRC})	Annuler les agrégations précédentes

D'une part, aucun accord sur une formalisation de ces opérateurs n'est unanimement reconnue, aucun standard ne s'est imposé. D'autre part, parmi toutes les propositions, seuls les travaux de [Abelló et al., 2003] et [Abelló et al., 2006] exploitent plusieurs fonctions d'agrégation pour la même mesure. Néanmoins ces travaux n'étudient pas en détails les effets de l'utilisation de plusieurs fonctions sur le langage d'interrogation.

Notre objectif est les conséquences sur les opérateurs d'analyse OLAP en reformulant la table multidimensionnelle et ces opérateurs afin de supporter notre modèle conceptuel multidimensionnel multifonctions.

[20] Nous utilisons la notation [ ] pour indiquer un paramètre optionnel dans la définition des opérateurs.

5.1.2 Notre proposition

Il y a des changements nécessaires sur certains opérateurs d'analyse en raison de caractéristiques de notre modèle conceptuel des bases de données multidimensionnelles multifonctions qui sont :

- L'automatisation de choix des fonctions d'agrégation ;
- La différence entre les fonctions utilisées sur les hiérarchies de la même dimension ;
- L'exigence d'un ordre d'exécution entre les fonctions d'agrégation de la même mesure.

Ces changements ne touchent pas la fonctionnalité mais ils peuvent toucher la définition et/ou le mécanisme interne. Ce mécanisme interne correspond à la requête qui effectue l'opération demandée. Ces changements peuvent affecter la TM ; ils peuvent aussi nécessiter d'adapter la TM afin de supporter plusieurs fonctions d'agrégation.

Afin de prendre en compte les mécanismes d'agrégation définis dans notre modèle, nous allons définir la table multidimensionnelle et les opérateurs d'analyse OLAP multifonctions :

- En évitant aux analystes de commettre des erreurs à cause d'une utilisation des fonctions inappropriées et facilitant l'analyse grâce à l'automatisation de choix des fonctions d'agrégation ;
- Assurant la fiabilité des données en respectant les contraintes d'agrégation et l'ordre d'exécution prédéfinis au niveau conceptuel, ce qui minimise le risque de production d'erreurs au cours de calcul des agrégations.

5.2 LANGAGE D'INTERROGATION DES DONNÉES MULTIDIMENSIONNELLES MULTIFONCTIONS

Un langage d'interrogation multidimensionnel a été proposé dans [Ravat et al., 2008]. Cette proposition se base sur un modèle en constellation, une structure de visualisation de table multidimensionnelle et un ensemble d'opérateurs de manipulation (Tableau 15). Inspiré de ces travaux, nous étendons la table multidimensionnelle et les opérateurs d'analyse pour les adapter à notre modèle multifonctions.

5.2.1 Table multidimensionnelle multifonctions

Une table multidimensionnelle (TM) est une structure de visualisation pour afficher les données résultantes d'une requête [Lehner, 1998], [Tournier, 2007]. Elle est une table à deux dimensions, utilisée en tant que source ou cible d'une opération d'analyse. Ce choix est justifié par le fait que, lors de leurs analyses, les décideurs manipulent couramment les données au travers de tableaux à deux dimensions (ligne et colonne) à cause de leur simplicité d'interprétation et leur précision [Gyssens & Lakshmanan, 1997].

Dans de précédents travaux [Lee & Ong, 1995], [Gyssens & Lakshmanan, 1997], [Sifer, 2003], [Choong et al., 2003], [Maniatis et al., 2005], [Techapichetvanich & Datta, 2005], [Hanrahan et al., 2007], [Cuzzocrea et al., 2007], [Ravat et al., 2008], [Cuzzocrea & Mansmann, 2009], [Ordonez et al., 2011], une TM n'affiche qu'une seule fonction d'agrégation pour chaque mesure. Dans le cadre de nos recherches, nous étendons le concept de TM pour qu'il puisse

supporter les principes du modèle conceptuel multifonctions, notamment en y intégrant les fonctions d'agrégation dans sa définition. Plus précisément, nous ajoutons dans la définition d'une TM, les dimensions non explicitées et les fonctions d'agrégations associées car elles ont une influence sur les calculs des mesures affichées.

Une TM multifonctions est donc définie formellement comme suit[21] :

TM = (F,
<(D_L, H_L, <p_{L1}, p_{L2}, …>),
(D_C, H_C, <p_{C1}, p_{C2}, …>),
(D_3, H_3, < >), …
(D_m, H_m, < >) >,
<{Aggregate(m_1)}, {Aggregate(m_2), …}>,
Pred)

Où :

- F : fait à analyser ;
- D_L, D_C : dimensions affichées respectivement en ligne et en colonne ;
- D_3, …, D_m : autres dimensions non-détaillées ;
- H_L, H_C : hiérarchies utilisées pour naviguer respectivement en ligne et en colonne ;
- H_3, …, H_m : hiérarchies courantes des dimensions non-détaillées ;
- p_{L1}, p_{L2}, … : les paramètres affichés en ligne où p_{L1} est supérieur à p_{L2} ;
- p_{C1}, p_{C2}, … : les paramètres affichés en colonne où p_{C1} est supérieur à p_{C2} ;
- Aggregate(m_1), Aggregate(m_2), … : fonctions d'agrégation associées aux mesures affichées (m_1, m_2, …) ;
- Pred : le prédicat de sélection sur les données du fait et/ou des dimensions pour limiter l'ensemble de valeurs à analyser.

Exemple 1. Selon le schéma structurel de notre exemple de météo (Figure 24) et le schéma d'agrégation de la mesure 'Tem_Moy' (Figure 25 (b)), la définition d'une table multidimensionnelle pour analyser les températures moyennes par mois et par département selon l'analyse scientifique (en utilisant la hiérarchie 'Hgéo_Scien') dans la région 'Midi-Pyrénées' est :

TM = (Température,
<(Dates, Hmois, <Année, MoisN>),
(Géographie, Hgéo_Scien, <Région, Département>),
(Temps, HTemps>, <>) >,
<{ (2, AVG(Tem_Moy), {}, {}, {}, 0),
(1, AVG(Tem_Moy), {Géographie}, {Hgéo_Scien}, {Ville}, 0)}>,
Géographie.Région = 'Midi-Pyrénées')

Les représentations graphiques des TMs classiques et multifonctions de cette définition sont illustrées dans la Figure 44 (a et b). La TM classique montre une seule fonction d'agrégation (AVG) dans la cellule qui affiche la mesure (Tem_Moy) [Ravat et al., 2008], parce qu'elle est la seule fonction utilisée pour agréger les valeurs de la mesure. Dans notre modèle, où nous avons plusieurs fonctions d'agrégation pour la même mesure, nous ne pouvons pas

[21] Une définition fonctionnelle d'une TM est déjà présentée (*cf.* § 1.2.4.1)

suivre cette méthode parce que cela donne une fausse idée de l'agrégation des valeurs de la mesure.

Température AVG(Tem_Moy)			**Géographie|Hgéo_Scien**		
			Région	Midi-Pyrénées	
			Département	Haute-Garonne	Ariège
Dates| Hmois	**Année**	**MoisN**			
	2012	2012-1		1	2
		2012-2		5	4
Géographie.Région='Midi-Pyrénées' and Dates.All='all' and Temps.All='all'					

(a) TM Classique

Température <2> AVG(Tem_Moy)			**Géographie|Hgéo_Scien**		
			Région	Midi-Pyrénées	
			Département <1> AVG(Tem_Moy)	Haute-Garonne	Ariège
Dates| Hmois	**Année**	**MoisN**			
	2012	2012-1		4	3
		2012-2		6	5
Géographie.Hgéo_Scien.Région='Midi-Pyrénées' and Dates.Hmois.All='all' and Temps.HTemps.All='all'					

(b) TM multifonctions

Figure 44 : Représentation graphique d'une TM

Afin d'adapter la visualisation de la TM pour présenter plusieurs fonctions d'agrégation qui peuvent être utilisées pour une seule mesure, la TM permet d'afficher les fonctions d'agrégation avec leurs arguments, leurs ordres d'exécution et leurs contraintes d'agrégation :

- La fonction générale est affichée à la place de la mesure dans la cellule du fait,
- La fonction multiple dimensionnelle est affichée à côté du nom de la dimension correspondante,
- La fonction multiple hiérarchique est affichée à côté du nom de la hiérarchie correspondante,
- La fonction différenciée est affichée dans la cellule du paramètre correspondant,
- Les arguments d'une fonction sont affichés entre parenthèses '()' après son nom,
- L'ordre d'exécution d'une fonction est affiché entre '<>' avant son nom,
- La contrainte d'agrégation d'une fonction est affichée après leurs arguments, à la fin de la fonction.

La visualisation est basée sur la simplification des fonctions autant que possible :

- Simplification des contraintes d'agrégation : si une fonction n'est pas contrainte (elle a une contrainte de valeur 0) alors la TM n'affiche pas cette valeur ;
- Simplification des ordres d'exécution : si toutes les fonctions d'agrégation affichées ont le même ordre d'exécution, la TM cache l'ordre. Par exemple, si nous analysons les températures moyennes mensuelles par département selon l'analyse simple (en utilisant la hiérarchie 'Hgéo_Simp'), les fonctions d'agrégation utilisées (la fonction générale 'AVG(Tem_Moy)' et la fonction multiple hiérarchique 'SELECT_CENTER(Niv_Adm, Tem_Moy)') ont le même ordre d'exécution (2). Alors, la TM résultante est présentée dans la Figure 45 ;
- Réduire le nombre des fonctions affichées :
  - La TM ne montre que les fonctions d'agrégation utilisées pour obtenir les valeurs affichées. Autrement dit, la TM n'illustre que les fonctions d'agrégation correspondantes au calcul des valeurs des mesures aux paramètres affichés les plus détaillés sur les deux dimensions ;
  - Si le paramètre affiché le plus détaillé sur une dimension a une fonction différenciée, la TM ne présente ni la fonction multiple hiérarchique ni la fonction multiple dimensionnelle sur la dimension considérée ;
  - Si une hiérarchie affichée a une fonction d'agrégation, la TM ne présente pas la fonction multiple dimensionnelle de la dimension considérée ;
  - Si les deux dimensions affichées ont une fonction multiple dimensionnelle ou une fonction multiple hiérarchique ou encore une fonction différenciée pour le paramètre affiché le plus détaillé, la TM ne présente pas la fonction générale.

Température AVG(Tem_Moy)			Géographie|Hgéo_Simp Select_Center(Niv_Adm, Tem_Moy)		
			Région	Midi-Pyrénées	
			Département	Haute-Garonne	Ariège
Dates| Hmois	Année	MoisN			
	2012	2012-1		2	3
		2012-2		4	4.5
Géographie.Hgéo_Simp.Région='Midi-Pyrénées' and Dates.Hmois.All='all' and Temps.HTemps.All='all'					

Figure 45 : Représentation graphique d'une TM avec un ordre d'exécution simplifié

5.2.2 Opérateurs d'analyse OLAP multifonctions

Une analyse OLAP est une interrogation dynamique et interactive du schéma multidimensionnel [Ghozzi, 2004]. Les opérateurs d'analyse de notre langage d'interrogation s'appuient sur notre modèle conceptuel multifonctions. Nous adaptons ces opérateurs à la possibilité d'utiliser plusieurs fonctions d'agrégation pour la même mesure. Nous pouvons classer les opérateurs multifonctions selon leurs fonctionnalités en neuf groupes : opérateur de construction, opérateurs de forage, de sélection, de rotation, de modifications du sujet d'analyse, de modifications d'une dimension, d'ordonnancements, d'agrégation et d'affichage d'agrégation.

D'un côté, le rôle principal de l'opérateur de construction est de permettre de spécifier une analyse multidimensionnelle à partir des éléments structurels (faits, dimensions, hiérarchies) du schéma multidimensionnel. D'un autre côté, les autres opérateurs permettent aux analystes de modifier et d'affiner cette analyse, ce qui leur offre une meilleure interprétation des données qu'ils observent, en assurant une abstraction complète de toutes les implantations logique et physique [Tournier, 2007].

Chaque opérateur prend en entrée une table multidimensionnelle source T_{SRC} (sauf l'opérateur de construction qui prend en entrée la BDM) et produit en sortie une table multidimensionnelle résultat T_{RES} (Figure 46). La fermeture des opérateurs est ainsi assurée.

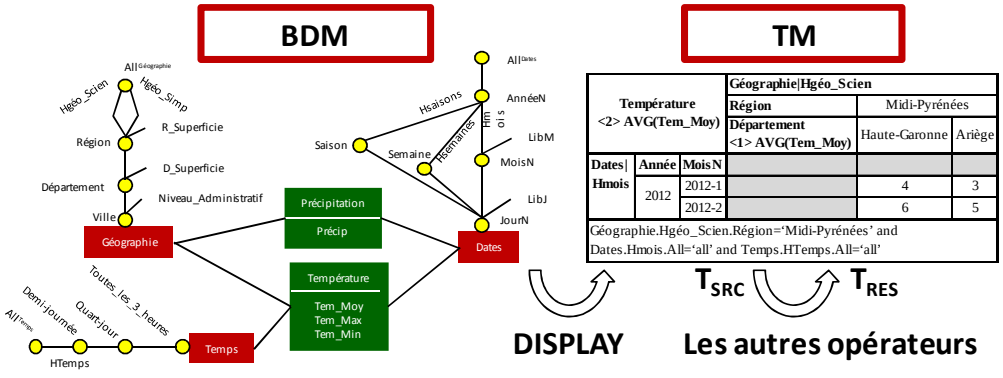


Figure 46 : Séquence d'opérateurs

L'adaptation des opérateurs d'analyse classiques (Tableau 15) à notre modèle multifonctions est due à plusieurs extensions détaillées à la suite.

L'automatisation du choix des fonctions d'agrégation : nous avons proposé de visualiser les fonctions d'agrégation avec leur ordre d'exécution et leurs contraintes d'agrégation directement dans le modèle conceptuel multidimensionnel afin d'éviter :

- Les erreurs de l'analyste qui risque de choisir une fonction d'agrégation inappropriée ;
- Les erreurs au cours du calcul des agrégations.

Ainsi, chaque mesure est liée aux fonctions qui agrègent ses valeurs dans l'espace multidimensionnel. Le système d'analyse peut donc choisir automatiquement la fonction d'agrégation à appliquer. Tout ce qui reste à faire par l'analyste est de choisir la mesure et les niveaux d'agrégation appropriés. Sur la base de ce qui précède, sont modifiés tous les opérateurs, dont la définition classique (Tableau 15) demande de choisir la fonction d'agrégation. Ainsi, le choix de la fonction disparaît.

La différence entre les fonctions utilisées sur les hiérarchies de la même dimension : l'utilisation des fonctions multiples hiérarchiques et des fonctions différenciées permet d'agréger les mesures de différentes manières en fonction de la hiérarchie considérée. Cela conduit à la possibilité d'obtenir des valeurs différentes de la même mesure pour le même paramètre (même le niveau 'All^{Di}') selon la hiérarchie. Il faut donc que l'analyste précise la hiérarchie même pour les dimensions non-détaillées dans la TM afin de choisir les valeurs des mesures souhaitées. Pour ajouter cette possibilité aux analystes, les définitions et les mécanismes internes de certains opérateurs seront changés.

L'existence d'un ordre d'exécution entre les fonctions d'agrégation : le rôle principal de l'ordre d'exécution est d'imposer un ordre précis dans l'application des fonctions d'agrégation pour ne pas avoir un résultat erroné à cause de la non-commutativité. Pour effectuer les opérateurs qui changent un niveau d'agrégation pour un autre niveau supérieur, il faut appliquer de nouvelles fonctions d'agrégation après avoir appliqué les fonctions précédentes qui donnent la TM en cours. Or, les ordres d'exécution des fonctions précédentes peuvent être supérieurs aux ordres d'exécution des nouvelles fonctions, c'est-à-dire, qu'il n'est pas possible d'appliquer les nouvelles fonctions après avoir appliqué les fonctions précédentes. Cela nécessite la mise en place de mécanismes internes plus coûteux pour ces opérateurs.

Dans la suite, nous détaillons nos opérateurs d'analyse multifonctions.

5.2.2.1 Opérateur de construction

Cet opérateur construit la première TM à partir du schéma d'une BDM initiant un processus d'analyse OLAP. Cette TM comprend une ou plusieurs mesures (sélectionnées par l'analyste) analysées selon deux dimensions. L'analyste choisit les deux dimensions en fonction desquelles le contenu sera affiché en lignes et en colonnes. Les autres dimensions ne sont pas détaillées dans la TM résultante. Cet opérateur est défini comme suit :

DISPLAY(F, {m_1, m_2,…}, D_L, H_L, D_C, H_C, D_3, H_3, … D_m, H_m) = T_{RES}

Où :

- F est le fait à analyser ;
- m_1, m_2,… sont les mesures à afficher dans la TM ;
- D_L est la dimension affichée en ligne ;
- H_L est la hiérarchie sélectionnée pour afficher ses paramètres en ligne ;
- D_C est la dimension affichée en colonne ;
- H_C est la hiérarchie sélectionnée pour afficher ses paramètres en colonne ;
- D_3, …, D_m sont les dimensions non-détaillées ;
- H_3, …, H_m sont les hiérarchies courantes des dimensions non-détaillées ;
- T_{RES} est la table multidimensionnelle résultante (Figure 44) telle que :

T_{RES} = (F, < (D_L, H_L, < $P^{H_L}_{max}$ >), (D_C, H_C, < $P^{H_C}_{max}$ >),
(D_3, H_3, < >), … , (D_m, H_m, < >) >,
<{Aggregate(m_1)}, {Aggregate(m_2), …}>,
D_L. H_L.All= 'all' ∧ D_C. H_C.All= 'all' ∧
D_3.H_3.All= 'all' ∧ … ∧ D_m.H_m.All= 'all')

Cet opérateur positionne l'analyse par défaut sur les niveaux de granularité ($P^{H_L}_{max}$ et $P^{H_C}_{max}$) directement inférieurs aux paramètres extrémités des hiérarchies H_L et H_C. En ce qui concerne les dimensions non-détaillées (D_3, …, D_m), les mesures sont agrégées aux paramètres extrémités 'All^{Di}' de leurs hiérarchies H_3, …, H_m.

Exemple 2. Pour que l'analyste puisse étudier les températures moyennes mensuelles par département selon l'analyse scientifique, il doit d'abord initialiser l'analyse par l'opérateur **DISPLAY** afin d'afficher les températures moyennes en fonction des hiérarchies 'Hgéo_Scien' et 'Hmois' des dimensions 'Géographie' et 'Dates'.

T_0 = **DISPLAY**(Température, {Tem_Moy},
Dates, Hmois,
Géographie, Hgéo_Scien,
Temps, HTemps)

Le système d'analyse doit consulter le schéma structurel (Figure 24) et le schéma d'agrégation de la mesure 'Tem_Moy' (Figure 25 (b)) afin de déterminer les niveaux de granularité directement inférieurs aux paramètres extrémités des hiérarchies 'Hmois' et 'Hgéo_Scien' et les fonctions d'agrégation qui leur correspondent. Il définit alors formellement T_0 comme suit :

T_0 = (Température,
<(Dates, Hmois, <Année>),
(Géographie, Hgéo_Scien, <Région>),
(Temps, HTemps>, <>) >,
<{ (2, AVG(Tem_Moy), {}, {}, {}, 0),
(1, AVG_W(Tem_Moy, D_Superficie), {Géographie}, {Hgéo_Scien},
{Département}, -1)}>,
Dates.Hmois.All = 'all' ∧ Géographie.Hgéo_Scien.All = 'all' ∧
Temps.HTemps.All = 'all')

La Figure 47 illustre la représentation graphique de T_0.

Température <2> **AVG(Tem_Moy)**		**Géographie|Hgéo_Scien**		
		Région <1> **AVG_W(Tem_Moy, D_Superficie) -1**	Alsace	Midi-Pyrénées	Rhône-Alpes
Dates| Hmois	**Année**				
	2011		12	14	11
	2012		11	14	12
	2013		12	15	11
Dates.Hmois.All='all' and Géographie.Hgéo_Scien.All='all' and Temps.HTemps.All='all'					

Figure 47 : Exemple d'une TM après l'application de l'opérateur DISPLAY (T_0)

Opérateur de construction et les fonctions d'agrégation

Une première conséquence de l'utilisation de plusieurs fonctions d'agrégation pour la même mesure concerne la spécification de l'opérateur de construction **DISPLAY**.

En effet, la visualisation des données multidimensionnelles après application de cet opérateur nécessite d'agréger les données aux niveaux de granularité ($P_{max}^{H_L}$ et $P_{max}^{H_C}$) des hiérarchies courantes en ligne et en colonne (H_L et H_C). Dans notre système multifonctions, cette agrégation peut donner des résultats différents en fonction des hiérarchies utilisées sur les dimensions non-détaillées dans la TM en raison des différentes fonctions utilisées sur les hiérarchies de la même dimension. C'est pourquoi nous offrons aux analystes la possibilité de déterminer ces hiérarchies en ajoutant (D_3, H_3, … D_m, H_m) dans la définition de l'opérateur **DISPLAY**.

Exemple 3. L'initialisation de l'analyse de la mesure 'Tem_Moy' selon les hiérarchies 'HTemps' et 'Hmois' des dimensions 'Temps' et 'Dates' nécessite de calculer les températures moyennes par année et demi-journée. Cette opération peut être effectuée dans un contexte uni-fonction par la requête SQL R_1 suivante :

R_1:
```
SELECT D.ANNEE, T.DEMI-JOURNEE, AVG(TEM_MOY) AS TEM_MOY
FROM DATES D, TEMPS T, TEMPERATURE TT
WHERE TT.ID_TEMPS= T.ID_TEMPS
  AND TT.ID_DATE = D.ID_DATE
GROUP BY ANNEE, DEMI-JOURNEE
```

Dans cette requête simple, la dimension 'Géographie' n'intervient pas. En revanche, dans un contexte multifonctions, cette opération donne deux résultats différents selon la hiérarchie considérée sur la dimension 'Géographie' non présente dans la TM.

Le premier résultat correspond à la hiérarchie 'Hgéo_Scien' (agrégation scientifique). Il est affiché dans la table multidimensionnelle T'_0 (Figure 48 (a)). Il convient de noter ici que dans une TM, les hiérarchies courantes des dimensions non-détaillées sont indiquées dans la zone consacrée au prédicat de sélection (Figure 48). Ce résultat peut être calculé par la requête R_2 suivante :

R_2:
```
SELECT ANNEE, DEMI-JOURNEE, AVG(TEM_MOY) AS TEM_MOY
FROM( SELECT  ANNEE, JourN, DEMI-JOURNEE, TOUTES_LES_3_HEURES,
              AVG_W(DATA_WEIGHTED(TEM_MOY, R_SUPERFICIE)) AS TEM_MOY
      FROM( SELECT  ANNEE,JourN,DEMI-JOURNEE,TOUTES_LES_3_HEURES,
                    REGION, R_SUPERFICIE,
                    AVG_W(DATA_WEIGHTED(TEM_MOY,D_SUPERFICIE)AS TEM_MOY
            FROM( SELECT  D.ANNEE,D.JourN, T.DEMI-JOURNEE,
                          T.TOUTES_LES_3_HEURES, G.REGION,
                          G.DEPARTEMENT, G.R_SUPERFICIE, G.D_SUPERFICIE,
                          AVG(TT.TEM_MOY) AS TEM_MOY
                  FROM DATES D, GEOGRAPHIE G, TEMPS T, TEMPERATURE TT
                  WHERE TT.ID_TEMPS= T.ID_TEMPS
                    AND TT.ID_VILLE= G.ID_VILLE
                    AND TT.ID_DATE = D.ID_DATE
                  GROUP BY  D.ANNEE,D.JourN, T.DEMI-JOURNEE,
                            T.TOUTES_LES_3_HEURES, G.REGION,
                            G.DEPARTEMENT,G.R_SUPERFICIE,G.D_SUPERFICIE)
            GROUP BY  ANNEE,JourN, DEMI-JOURNEE,TOUTES_LES_3_HEURES,
                      REGION, R_SUPERFICIE)
      GROUP BY ANNEE, JourN, DEMI-JOURNEE, TOUTES_LES_3_HEURES)
GROUP BY ANNEE, DEMI-JOURNEE
```

Le système d'analyse peut générer cette requête en effectuant les quatre étapes de l'analyse multifonctions (*cf.* § 3.3.3). Dans cette requête, les températures moyennes sont agrégées selon le schéma d'agrégation (Figure 25 (b)) :

- Premièrement, sur la hiérarchie 'Hgéo_Scien' de la dimension 'Géographie' :
  - D'abord, au niveau 'Département' par la fonction AVG,
  - Puis, en calculant les températures moyennes régionales par la fonction AVG_W(Tem_Moy, D_Superficie),
  - Ensuite, au niveau 'All$^{\text{Géograhie}}$' en utilisant la fonction AVG_W(Tem_Moy, R_Superficie),
- Deuxièmement, sur les dimensions 'Dates' et 'Temps', aux niveaux 'Année' et 'Demi-journée' en appliquant la fonction 'AVG'.

Le deuxième résultat correspond à la hiérarchie 'Hgéo_Simp' (agrégation simple). La table multidimensionnelle T"$_0$ (Figure 48 (b)) affiche ce résultat qui est calculable par la requête R_3 suivante :

R_3 :

```
SELECT  ANNEE, DEMI-JOURNEE,
        SELECT_CENTER(LEV_DATA(NIVEAU_ADMINISTRATIF,TEM_MOY))AS TEM_MOY
FROM( SELECT D.ANNEE, T.DEMI-JOURNEE, G.VILLE, G.NIVEAU_ADMINISTRATIF,
             AVG(TT.TEM_MOY) AS TEM_MOY
      FROM DATES D, GEOGRAPHIE G, TEMPS T, TEMPERATURE TT
      WHERE TT.ID_TEMPS= T.ID_TEMPS
        AND TT.ID_VILLE= G.ID_VILLE
        AND TT.ID_DATE = D.ID_DATE
      GROUP BY D.ANNEE, T.DEMI-JOURNEE, G.VILLE,G.NIVEAU_ADMINISTRATIF)
GROUP BY ANNEE, DEMI-JOURNEE
```

Cette requête calcule les températures moyennes d'abord aux niveaux 'Année' et 'Demi-journée' des dimensions 'Dates' et 'Temps'. Ensuite, elle applique l'agrégation simple au niveau 'All$^{\text{Géographie}}$'.

T'$_0$=DISPLAY(Température,{Tem_Moy}, Dates, Hmois, Temps, HTemps, Géographie, Hgéo_Scien)

Température AVG(Tem_Moy)		Temps|Htemps		
		Demi-journée	jour	nuit
Dates| Hmois	Année			
	2011		20	4
	2012		21	5
	2013		22	6
Dates.Hmois.All='all' and Temps.HTemps.All='all' and Géographie.Hgéo_Scien.All='all'				

(a) T'$_0$

T"$_0$=DISPLAY(Température,{Tem_Moy}, Dates, Hmois, Temps, HTemps, Géographie, Hgéo_Simp)

Température AVG(Tem_Moy)		Temps|Htemps		
		Demi-journée	jour	nuit
Dates| Hmois	Année			
	2011		18	3
	2012		19	4.5
	2013		20.5	5.5
Dates.Hmois.All='all' and Temps.HTemps.All='all' and Géographie.Hgéo_Simp.All='all'				

(b) T"$_0$

Figure 48 : Exemple de l'intervention d'une dimension non-présentée dans une TM

Il y a une deuxième implication sur la spécification de cet opérateur grâce à l'automatisation du choix des fonctions d'agrégation qui ne nécessite pas de préciser les fonctions utilisées pour agréger la mesure au moment de l'interrogation.

5.2.2.2 Opérateurs de forage

Ces opérateurs permettent aux analystes de changer, les niveaux de granularité des données multidimensionnelles visualisées dans une TM par un niveau plus ou moins détaillé. Il s'agit de passer d'un paramètre donné à un autre paramètre de la même hiérarchie. Nous étendons les deux opérateurs de forage classiquement définis.

5.2.2.2.1 Le forage vers le bas (DRILLDOWN)

Grâce à cet opérateur, les analystes peuvent analyser les données de manière plus détaillées. L'opérateur augmente le nombre de paramètres affichés en ajoutant un paramètre plus détaillé à une hiérarchie courante d'une dimension visualisée dans la TM (Figure 49). Sa syntaxe est la suivante :

DRILLDOWN(T_{SRC}, D, P_{inf}) = T_{RES}

Où :

- D $\in$ {D_L, D_C} est la dimension sur laquelle s'applique le forage ;
- P_{inf} est un paramètre de la hiérarchie courante de la dimension concernée (D). Ce paramètre doit être inférieur à tous les paramètres de la même hiérarchie affichés dans la table multidimensionnelle source (T_{SRC}) ;
- T_{RES} est la table résultante où la seule modification concerne la liste des paramètres de la dimension (D) affichés en ligne ou en colonne ; il s'agit d'ajouter le paramètre (P_{inf}) :
  - Si D = D_L, la liste de paramètres en ligne est <$P^{H_L}_{max}$, …, $P^{H_L}_{min}$, P_{inf} >,
  - si D = D_C, la liste de paramètres en colonne est <$P^{H_C}_{max}$, …, $P^{H_C}_{min}$, P_{inf} >.

Les paramètres intermédiaires entre le paramètre inférieur de la dimension concernée D de la table source ($P^{H_L}_{min}$ ou $P^{H_C}_{min}$) et le nouveau paramètre (P_{inf}) ne sont pas affichés.

Forage vers le bas et les fonctions d'agrégation

Il n'y a pas de répercussion liée à l'utilisation de plusieurs fonctions d'agrégation pour la même mesure sur cet opérateur autre que l'application des quatre étapes de l'analyse multifonctions (*cf.* § 3.3.3).

5.2.2.2.2 Le forage vers le haut (ROLLUP)

Cet opérateur consiste à analyser les données de manière plus globale. Il réduit le nombre de paramètres affichés en retirant un ou plusieurs paramètres d'une hiérarchie courante d'une dimension visualisée dans la TM (Figure 49). La syntaxe de l'opérateur est la suivante :

ROLLUP(T_{SRC}, D, P_{sup}) = T_{RES}

Où :

- D $\in$ {D_L, D_C} est la dimension sur laquelle s'applique le forage ;
- P_{sup} est un paramètre de la hiérarchie courante de la dimension concernée (D) déjà affiché dans la table source T_{SRC}. Il ne doit pas être le paramètre le plus inférieur ;
- T_{RES} est la table résultante où la modification sera de supprimer tous les paramètres affichés inférieurs au paramètre (P_{sup}) de la dimension (D) affichés en ligne ou en colonne :
  - Si D = D_L, la liste de paramètres en ligne est <$P^{H_L}_{max}$, …, P_{sup} >,
  - si D = D_C, la liste de paramètres en colonne est <$P^{H_C}_{max}$, …, P_{sup} >.

Forage vers le haut et les fonctions d'agrégation

L'utilisation de plusieurs fonctions d'agrégation pour la même mesure influence le mécanisme interne de cet opérateur. Afin d'étudier cet impact, nous nous basons sur la TM pour

analyser les températures moyennes des départements par mois dans la région 'Midi-Pyrénées' (Figure 44). A partir de cette TM, nous effectuons une opération de **ROLLUP** pour analyser les températures par régions (Figure 49).

Premièrement, nous prenons en compte ***le modèle multidimensionnel classique*** où il n'y a qu'une seule fonction pour agréger la mesure. La requête SQL qui réalise l'analyse des températures moyennes départementales mensuelles dans la région 'Midi-Pyrénées' est :

```
R4: Result1 =
SELECT  G.DEPARTEMENT, D.MOISN, AVG(TT.TEM_MOY) AS TEM_MOY,
        SUM(TT.TEM_MOY) AS sum_Tem_Moy,
        COUNT(TT.TEM_MOY) AS count_Tem_Moy
FROM DATES D, GEOGRAPHIE G, TEMPERATURE TT
WHERE TT.ID_VILLE = G.ID_VILLE
  AND TT.ID_DATE = D.ID_DATE
  AND G.REGION = 'Midi-Pyrénées'
GROUP BY G.DEPARTEMENT, D.MOISN;
```

Pour exécuter le forage **ROLLUP** souhaité, on peut profiter des résultats de la TM précédente si la fonction d'agrégation est distributive ou algébrique (dans ce dernier cas, il est nécessaire de stocker des valeurs intermédiaires) [Gray et al., 1996]. Dans notre exemple, la fonction est algébrique (AVG). Les valeurs intermédiaires demandées sont la somme des températures moyennes mensuelles des départements (sum_Tem_Moy) et le nombre des occurrences (count_Tem_Moy). Ainsi, la requête R_5, qui réalise le **ROLLUP** correspondant à la TM (T_2) de la Figure 49, bénéficie des résultats (Result1) de la requête précédente R_4. Autrement dit, pour mettre en œuvre une opération de forage vers le haut (**ROLLUP**), on n'a pas besoin d'accéder et de charger les valeurs des mesures, mais on peut utiliser les valeurs intermédiaires présentées dans la TM.

```
R5:
SELECT REGION, MOISN, SUM(sum_TEM_MOY)/SUM(count_TEM_MOY) AS TEM_MOY
FROM GEOGRAPHIE G, Result1 R4
WHERE R4.DEPARTEMENT = G.DEPARTEMENT
GROUP BY REGION, MOISN
```

Deuxièmement, dans ***notre modèle multifonctions***, puisqu'il y a plusieurs fonctions qui agrègent la mesure, la requête SQL qui réalise l'analyse des températures moyennes des départements (sur la hiérarchie 'Hgéo_Scien') par mois dans la région 'Midi-Pyrénées' (T_1 de la Figure 49) devient plus complexe :

```
R6: Result2 =
SELECT  MOISN, DEPARTEMENT, AVG(TEM_MOY) AS TEM_MOY,
        SUM(TEM_MOY) AS sum_Tem_Moy, COUNT(TEM_MOY) AS count_Tem_Moy
FROM( SELECT  D.MOISN, G.DEPARTEMENT, D.JourN, T.TOUTE_LES_3_HEURES,
              AVG(TT.TEM_MOY) AS TEM_MOY
      FROM DATES D, GEOGRAPHIE G, TEMPERATURE TT, TEMPS T
      WHERE TT.ID_TEMPS = T.ID_TEMPS
        AND TT.ID_VILLE = G.ID_VILLE
        AND TT.ID_DATE = D.ID_DATE
        AND G.REGION = 'Midi-Pyrénées'
      GROUP BY D.MOISN, G.DEPARTEMENT, D.JourN, T.TOUTE_LES_3_HEURES)
GROUP BY MOISN, REGION, DEPARTEMENT
```

Pour effectuer une opération de forage vers le haut dans un modèle multifonctions, à cause de l'ordre d'exécution entre les fonctions d'agrégation, nous pouvons distinguer deux cas selon les fonctions d'agrégation correspondant au forage demandé :

Le premier cas, où tous les ordres d'exécution des fonctions qui agrègent la mesure entre le paramètre actuel et le paramètre demandé, sont supérieurs ou égaux aux ordres d'exécution des fonctions qui agrègent la mesure entre les paramètres de base et les paramètres actuels, y compris les niveaux 'All^{Di}' des dimensions qui n'apparaissent pas dans la TM. Par exemple, si nous voulons faire un forage vers le haut pour analyser les températures moyennes des départements par années. La fonction qui agrège les températures moyennes entre le niveau 'moisN' et 'Année' est la fonction générale 'Avg(Tem_Moy)'. Cette fonction a un ordre d'exécution de valeur 2 qui est égal ou supérieur aux ordres d'exécution des fonctions qui agrègent les températures moyennes entre les niveaux de base et les niveaux 'Département', 'moisN', 'All^{Temps}'. Dans ce cas, de la même manière que le **ROLLUP** dans un contexte uni-fonction, nous pouvons profiter des valeurs de la mesure présentées dans la TM comme dans la requête R_7.

R_7:
```
SELECT ANNEE,DEPARTEMENT,SUM(sum_TEM_MOY)/SUM(count_TEM_MOY)AS TEM_MOY
FROM DATES D, Result2 R6
WHERE R6.MOISN = D.MOISN
GROUP BY ANNEE, DEPARTEMENT
```

Le deuxième cas, où un ordre d'exécution d'une fonction, qui agrège la mesure entre le paramètre actuel et le paramètre demandé, est inférieur à un ordre d'exécution d'une fonction qui agrège la mesure entre les paramètres de base et les paramètres actuels. Par exemple, nous voulons faire un forage vers le haut pour analyser les températures moyennes des régions par mois (T_2 de la Figure 49) à partir des températures moyennes des départements par mois (T_1 de la Figure 49). La fonction qui agrège les températures moyennes entre les niveaux 'Département' et 'Région' sur la hiérarchie 'Hgéo_Scien' est la fonction Avg_W(Tem_Moy, D_Superficie). Cette fonction a un ordre d'exécution de valeur 1 qui est inférieur à l'ordre d'exécution de la fonction générale qui agrège les températures moyennes entre le niveau de base et le niveau 'moisN'. Ceci signifie qu'il faut calculer les températures moyennes par région avant de calculer les températures moyennes par mois. Dans ce cas, nous ne pouvons pas profiter des valeurs de la mesure présentées dans la TM. Il faut donc recalculer les valeurs de la mesure à partir des niveaux de base comme la requête R_8.

R_8: Result3 =
```
SELECT MOISN, REGION, AVG(TEM_MOY) AS TEM_MOY
FROM( SELECT  MOISN, REGION, TOUTE_LES_3_HEURES, JourN,
              AVG_W(DATA_WEIGHTED(TEM_MOY, D_SUPERFICIE)) AS TEM_MOY
      FROM( SELECT  D.MOISN, G.REGION, G.DEPARTEMENT,
                    T.TOUTE_LES_3_HEURES, D.JourN, G.D_SUPERFICIE,
                    AVG(TT.TEM_MOY) AS TEM_MOY
            FROM DATES D, GEOGRAPHIE G, TEMPERATURE TT, TEMPS T
            WHERE TT.ID_TEMPS = T.ID_TEMPS
              AND TT.ID_VILLE = G.ID_VILLE
              AND TT.ID_DATE = D.ID_DATE
              AND G.REGION = 'Midi-Pyrénées'
            GROUP BY  D.MOISN, G.REGION, G.DEPARTEMENT,
                      T.TOUTE_LES_3_HEURES, D.JourN, G.D_SUPERFICIE)
      GROUP BY MOISN, REGION, TOUTE_LES_3_HEURES, JourN)
GROUP BY MOISN, REGION
```

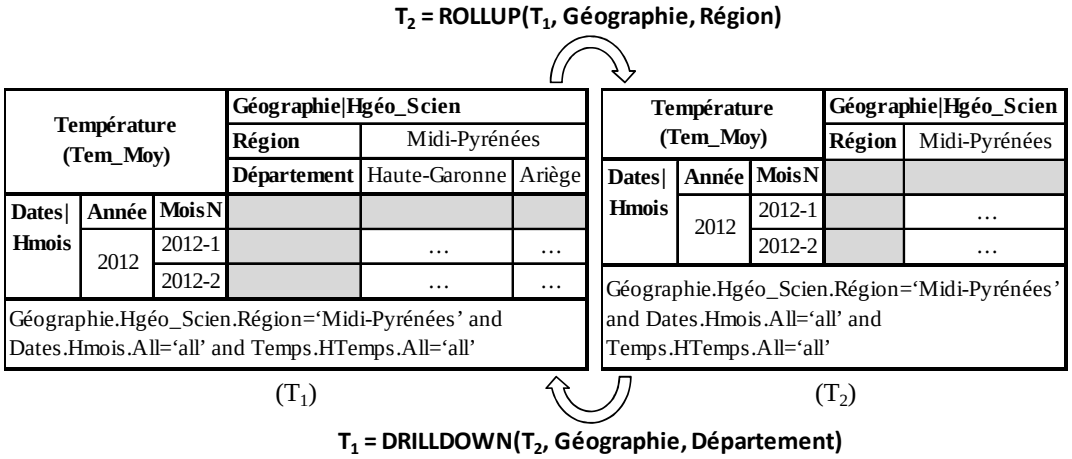


Figure 49 : Opérateurs de forage[22]

5.2.2.3 Opérateurs de sélection

Ces opérateurs sont utilisés pour réduire le nombre de données affichées dans la TM. L'analyste peut effectuer une opération de sélection en spécifiant une condition (prédicat) de restriction. Cette restriction peut être appliquée sur le fait et/ou les dimensions. Toutes les valeurs qui ne satisfont pas le prédicat sont retirées de la TM (Figure 50). Ces opérateurs permettent également d'annuler toutes les sélections appliquées précédemment.

5.2.2.3.1 SELECT

Cet opérateur permet à l'analyste de spécifier le prédicat de restriction. La syntaxe de l'opérateur est la suivante :

SELECT(T_{SRC}, $pred_{new}$) = T_{RES}

Où :

- $pred_{new} = pred'_1 \wedge pred'_2 \wedge \ldots$ est un prédicat de sélection des données du fait et/ou des dimensions ;
- T_{RES} est la table résultante qui a les mêmes caractéristiques que la table source (T_{SRC}) sauf son prédicat ($pred_{old} = pred_1 \wedge pred_2 \wedge \ldots$) qui est remplacé par le nouveau prédicat ($pred_{new}$).

5.2.2.3.2 UNSELECT

Cet opérateur consiste à annuler toutes les sélections de fait et de dimensions. Sa syntaxe est la suivante :

[22] Nous ne présentons pas les fonctions d'agrégation et les valeurs de la mesure pour adapter les TMs aux deux contextes (uni-fonction et multifonctions).

UNSELECT(T_{SRC}) = T_{RES}

Où T_{RES} est la table résultante. Elle est construite à partir de toutes les caractéristiques de la table source (T_{SRC}) mais sans restrictions.

Exemple 4. La Figure 50 présente un exemple d'application des opérateurs de sélection (**SELECT** et **UNSELECT**) montrant le changement du prédicat de la TM avec cette application. Dans cet exemple nous analysons les températures moyennes annuelles régionales selon l'analyse simple (en utilisant la hiérarchie 'Hgéo_Simp'). La TM (T_3) visualise les températures de toutes les régions tandis que grâce à l'opérateur **SELECT** la TM (T_4) ne montre que les températures de la région 'Midi-Pyrénées'. Un retour à la TM (T_3) est possible en utilisant l'opérateur **UNSELECT.**

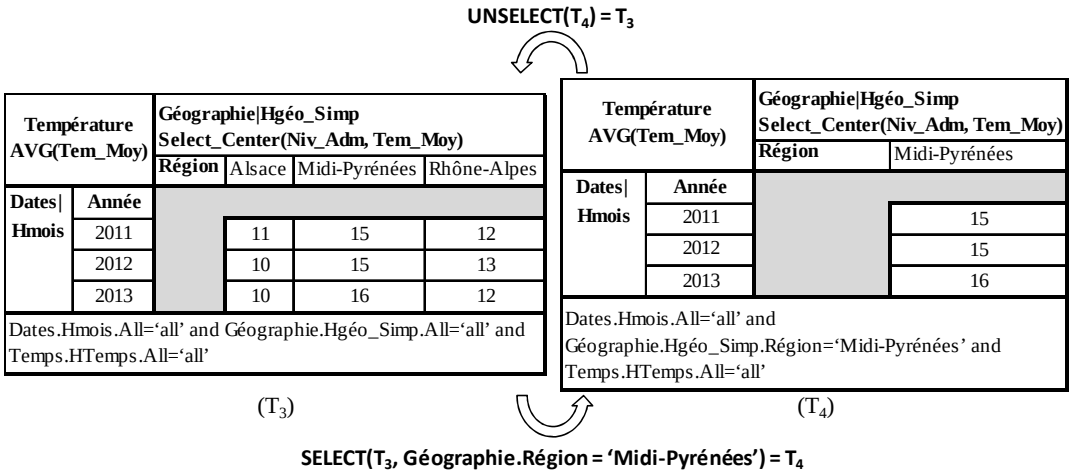


Figure 50 : Opérateurs de sélection

Les opérateurs de sélection et les fonctions d'agrégation

Ces opérateurs sont spécifiés et mis en œuvre dans un contexte multifonctions de manière identique au contexte uni-fonction.

5.2.2.4 Opérateurs de rotation

Le rôle principal des opérateurs de rotation est d'offrir aux analystes la possibilité de réorienter l'analyse en permutant deux dimensions ou deux hiérarchies afin de changer l'agrégation et la visualisation des données dans la TM. Ils permettent également dans un contexte d'un schéma en constellation de remplacer un fait par un autre afin de changer les données analysées.

5.2.2.4.1 La rotation des dimensions (DROTATE)

Au cours d'une analyse, l'analyste pourrait vouloir changer la perspective des données. Dans ce cas, il peut exploiter l'opérateur de rotation des dimensions qui lui permet de remplacer un axe d'analyse affiché en ligne ou en colonne dans une TM par un autre qui doit être associé au fait analysé dans le schéma conceptuel multidimensionnel. La syntaxe de l'opérateur est la suivante :

DROTATE(T_{SRC}, D_{old}, D_{new}, H_{new}) = T_{RES}

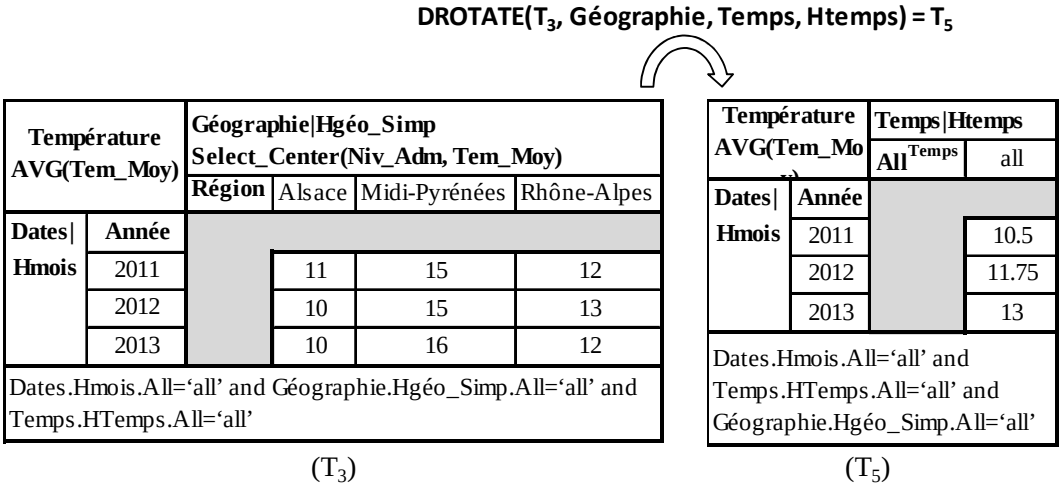
Où :

- $D_{old} \in \{D_L, D_C\}$ est la dimension à remplacer ;
- D_{new} est la nouvelle dimension remplaçante ;

- H_{new} est la hiérarchie courante de la nouvelle dimension ;
- T_{RES} est la table résultante où la modification concerne la dimension affiché en ligne (si $D_{old} = D_L$) ou en colonne (si $D_{old} = D_C$).

Le paramètre affiché de la hiérarchie H_{new} est par défaut le paramètre extrémité ($All^{D_{new}}$) (Figure 51).

Exemple 5. Dans la Figure 51, l'analyste peut étudier les températures moyennes selon la date et le temps à partir de celles analysées selon la date et la géographie en remplaçant la dimension 'Géographie' de la TM (T_3) par la dimension 'Temps' dans la TM (T_5). Cette application de l'opérateur **DROTATE** ne change pas la hiérarchie courante ('Hgéo_Simp') de la dimension qui n'est plus détaillée dans la TM ('Géographie').



Température AVG(Tem_Moy)		Géographie|Hgéo_Simp Select_Center(Niv_Adm, Tem_Moy)		
		Région	Alsace	Midi-Pyrénées	Rhône-Alpes
Dates| Hmois	Année				
	2011		11	15	12
	2012		10	15	13
	2013		10	16	12

Dates.Hmois.All='all' and Géographie.Hgéo_Simp.All='all' and Temps.HTemps.All='all'

(T_3)

Température AVG(Tem_Mo…)		Temps|Htemps	
		All^{Temps}	all
Dates| Hmois	Année		
	2011		10.5
	2012		11.75
	2013		13

Dates.Hmois.All='all' and Temps.HTemps.All='all' and Géographie.Hgéo_Simp.All='all'

(T_5)

Figure 51 : Opérateur de rotation des dimensions (DROTATE)

L'opérateur de rotation des dimensions (DROTATE) et les fonctions d'agrégation

Les implications d'utilisation de plusieurs fonctions d'agrégation pour la même mesure sur l'opérateur **DROTATE** s'apparentent à celles de l'opérateur **ROLLUP**. Pour effectuer les deux opérateurs, il faut changer un niveau d'agrégation vers un niveau supérieur :

- Pour l'opérateur **ROLLUP** : du niveau actuel au niveau cible ;
- Pour l'opérateur **DROTATE** : du niveau actuel au niveau 'All^{Di}' de la dimension qui va disparaitre de la TM.

C'est pourquoi la réalisation de **DROTATE** suit les principes de la réalisation de **ROLLUP** selon lesquels nous pouvons profiter ou pas des valeurs de la mesure présentées dans la TM précédente. Cette exploitation dépend de l'ordre d'exécution des fonctions d'agrégation qui effectuent l'opération **DROTATE** par rapport à celles des fonctions déjà appliquées.

Exemple 6. La Figure 52 présente l'agrégation nécessaire pour réaliser l'opération **DROTATE** de l'exemple 5 précédent. Les schémas multidimensionnels de gauche et de droite représentent la visualisation des éléments structurels dans les TMs (T_3 et T_5) avant et après la réalisation de **DROTATE**. Les dimensions et les paramètres affichés dans les TMs sont présentés en couleur et en gras. Les dimensions non-détaillées dans les TMs sont présentées en gris. Nous remarquons que le niveau d'agrégation considéré sur la dimension 'Temps' ('All^{Temps}'), qui va apparaitre dans la TM (T_5) après l'application de **DROTATE,** reste le même

avant et après cette application. Par contre, pour la dimension 'Géographie' affichée dans (T_3) et qui va disparaitre dans (T_5), le niveau d'agrégation change de 'Région' vers 'All^{Géographie}'. C'est le seul changement de niveaux d'agrégation qui peut être réalisé de manière identique à un **ROLLUP** entre les deux niveaux ('Région' et 'All^{Géographie}').

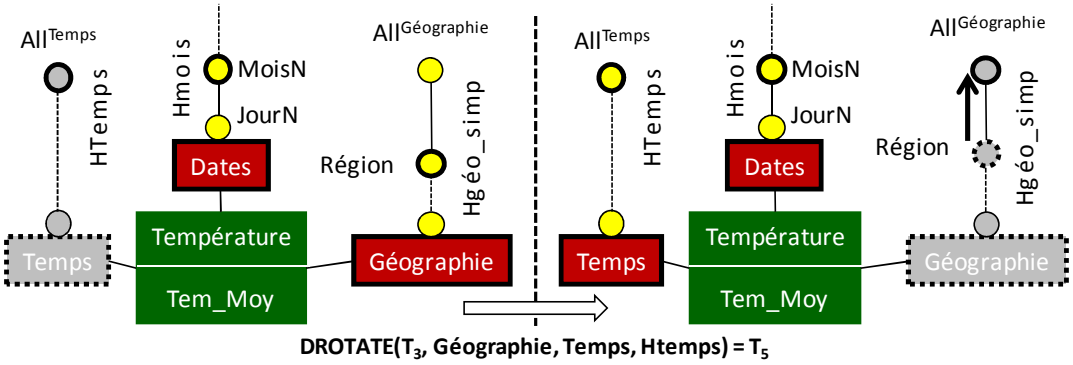


Figure 52 : DROTATE et les fonctions d'agrégation

5.2.2.4.2 La rotation des hiérarchies (HROTATE)

Cet opérateur consiste à changer une hiérarchie courante dans une TM par une autre appartenant à la même dimension. La syntaxe de l'opérateur est la suivante :

HROTATE(T_{SRC}, D, H_{new}) = T_{RES}
Où :
- D est une dimension liée au fait visualisé ;
- H_{new} est la nouvelle hiérarchie courante de la dimension D ;
- T_{RES} est la table résultante où la hiérarchie (H_{new}) remplace la hiérarchie courante de la dimension (D).

Similairement à l'opérateur **DROTATE**, l'opérateur **HROTATE** positionne l'analyse sur le paramètre extrémité (All^D) de la hiérarchie H_{new}.

L'opérateur de rotation des hiérarchies (HROTATE) et les fonctions d'agrégation

Nous distinguons deux implications d'utilisation de plusieurs fonctions d'agrégation sur le mécanisme interne et la spécification de l'opérateur **HROTATE** :

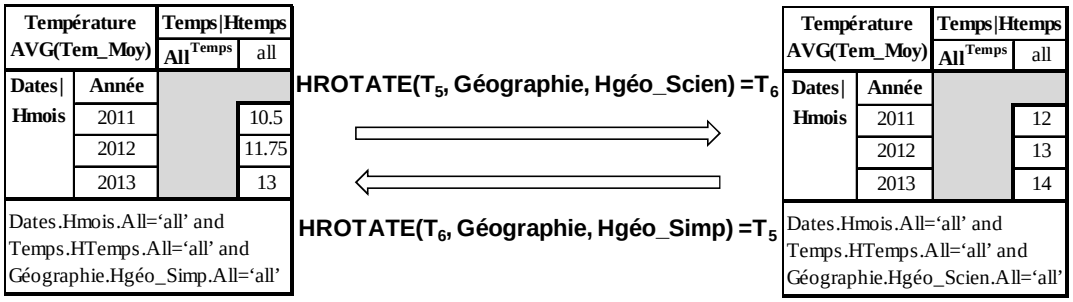
Premièrement, de la même manière que l'opérateur **ROLLUP** et **DROTATE**, dans un contexte uni-fonction, nous pouvons toujours profiter des résultats de la TM précédente afin de réaliser l'opérateur **HROTATE**. Toutefois dans notre contexte multifonctions :

- S'il y a une différence entre les agrégations sur la hiérarchie courante et la nouvelle hiérarchie, ou
- Si les ordres d'exécution des fonctions, qui agrègent la mesure entre le paramètre actuel et le paramètre 'All^{Di}' de la hiérarchie courante, ne sont pas tous supérieurs ou égaux aux ordres d'exécution des fonctions déjà appliquées.

Dans ces cas, nous ne pouvons pas exploiter les valeurs de la TM précédente.

Deuxièmement, contrairement au contexte uni-fonction, l'application de cet opérateur dans notre contexte multifonctions peut changer les valeurs de la mesure affichées même s'il est appliqué sur une hiérarchie d'une dimension non-détaillée dans la TM. Ce changement est dû à la différence entre les fonctions utilisées sur les hiérarchies de cette dimension.

Exemple 7. Le changement de hiérarchie courante de la dimension 'Géographie' modifie les températures moyennes analysées en fonction des dimensions 'Dates' et 'Temps' (Figure 53). En effet, cette permutation entre les hiérarchies 'Hgéo_Simp' et 'Hgéo_Scien' (en appliquant l'opérateur **HROTATE**) correspond à la permutation entre les agrégations des données scientifique et simple au niveau 'All$^{\text{Géographie}}$'. C'est pourquoi les valeurs de la mesure changent, bien que la dimension 'Géographie' ne soit pas détaillée dans les TMs (T_5 et T_6).



(T_5) (T_6)

Figure 53 : Opérateur de rotation des hiérarchies (HROTATE)

5.2.2.4.3 La rotation des faits (FROTATE)

Notre modèle multidimensionnel permet de définir un schéma en constellation qui comporte plusieurs faits dans le même schéma. L'opérateur de rotation de faits permet de permuter les faits du schéma en constellation en conservant les caractéristiques des dimensions utilisées. La syntaxe de cet opérateur est la suivante :

FROTATE(T_{SRC}, F_{new}, {m'_1, m'_2, ...}, D_3, H_3,..., D_m, H_m) = T_{RES}

Où :

- F_{new} est le nouveau fait ;
- m_1, m_2 sont les nouvelles mesures à analyser ;
- D_3, …, D_m sont les dimensions non-détaillées associés au nouveau fait ;
- H_3, …, H_m sont les hiérarchies courantes des dimensions non-détaillées ;
- T_{RES} est la table résultante qui utilise le nouveau fait (F_{new}).

L'opérateur préserve toutes les caractéristiques des axes d'analyse en ligne et en colonne. Le nouveau fait doit ainsi partager avec le fait initial au moins les deux dimensions visualisées (D_L et D_C) (Figure 54).

Exemple 8. A partir de la TM (T_3) qui affiche les températures moyennes régionales annuelles, nous pouvons analyser directement les précipitations régionales annuelles en appliquant l'opérateur de rotation des faits (**FROTATE**) comme le montre la Figure 54. Cette application est possible parce que les deux faits 'Température' et 'Précipitation' partages les deux dimensions affichées ('Géographie' et 'Dates'). Contrairement au fait 'Précipitation', le fait 'Température' est associé dans le schéma multidimensionnel à une troisième dimension 'Temps'. C'est pourquoi, afin de remplacer le fait 'Précipitation' par le fait 'Température', il est nécessaire d'indiquer la hiérarchie courante de la dimension 'Temps'. En revanche, la permutation inverse n'a pas besoin de faire cela (Figure 54).

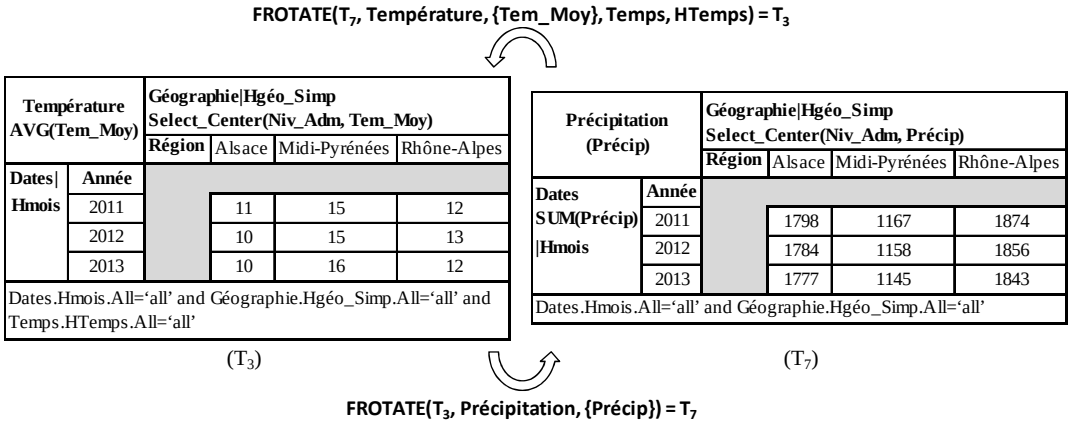


Figure 54 : Opérateur de rotation des faits (FROTATE)

L'opérateur de rotation des faits (FROTATE) et les fonctions d'agrégation

De la même manière que l'opérateur de construction **DISPLAY**, le contexte multifonctions a deux impacts sur la spécification de l'opérateur de rotation des faits **FROTATE** :

- Grâce à l'automatisation du choix des fonctions d'agrégation, il n'est pas nécessaire de spécifier les fonctions d'agrégation des mesures analysées.
- A cause de la différence entre les agrégations utilisées sur les hiérarchies d'une même dimension, il est nécessaire de préciser les hiérarchies utilisées sur les dimensions non-détaillées dans la définition de l'opérateur **FROTATE**.

5.2.2.5 Opérateurs de modifications du sujet d'analyse

Ces opérateurs consistent à modifier l'ensemble des mesures affichées dans une TM en ajoutant et supprimant des mesures.

5.2.2.5.1 L'ajout de mesures (ADDM)

L'opérateur **ADDM** permet à l'analyste d'ajouter des nouvelles mesures à analyser dans la TM (Figure 55). La syntaxe de l'opérateur est la suivante :

ADDM(T_{SRC}, m_i) = T_{RES}

Où :

- m_i est la nouvelle mesure à afficher. Cette mesure doit appartenir au même fait visualisé dans la table source (T_{SRC}) ;
- T_{RES} est la table résultante qui affiche toutes les mesures de la table source (T_{SRC}) avec la nouvelle mesure (m_i) : {m_1, m_2, …, m_i}.

5.2.2.5.2 Suppression de mesures (DELM)

L'opérateur **DELM** permet à l'analyste de supprimer des mesures d'une TM (Figure 55). La syntaxe de l'opérateur est la suivante :

DELM(T_{SRC}, m_i) = T_{RES}

Où :

- m_i est la mesure à supprimer de la table source (T_{SRC}) ;

- T_{RES} est la table résultante qui affiche toutes les mesures de la table source (T_{SRC}) sans la mesure (m_i) qui n'est pas nécessairement la dernière mesure ajoutée à table source : {m_1, m_2, …, m_{i-1}, m_{i+1}, …}.

Les opérateurs de modifications du sujet d'analyse et les fonctions d'agrégation

L'automatisation du choix des fonctions d'agrégation simplifie la spécification des opérateurs de modification du sujet d'analyse, où il ne faut pas préciser les fonctions d'agrégation utilisées pour agréger les nouvelles mesures à analyser ou les mesures à supprimer (Figure 55).

Température <2> AVG(Tem_Moy)			Géographie|Hgéo_Scien	
			Région <1> AVG_W(Tem_Moy, D_Superficie) -1	Midi-Pyrénées
Dates| Hmois	Année	MoisN		
	2012	2012-1		2.4
		2012-2		4.2
Géographie.Hgéo_Scien.Région='Midi-Pyrénées' and Dates.Hmois.All='all' and Temps.HTemps.All='all'				

ADDM(T_2, Tem_Max) =T_8 ⇩ (T_2) ⇧ **DELM(T_8, Tem_Max) =T_2**

Température <2> AVG(Tem_Moy), MAX(Tem_Moy)			Géographie|Hgéo_Scien	
			Région <1> AVG_W(Tem_Moy, D_Superficie) -1	Midi-Pyrénées
Dates| Hmois	Année	MoisN		
	2012	2012-1		(2.4), (5)
		2012-2		(4.2), (8.7)
Géographie.Hgéo_Scien.Région='Midi-Pyrénées' and Dates.Hmois.All='all' and Temps.HTemps.All='all'				

(T_8)

Figure 55 : Opérateurs de modifications du sujet d'analyse

Exemple 9. L'analyse des températures maximales avec les températures moyennes est toujours possible en ajoutant la mesure 'Tem_Max' à une TM qui affiche les valeurs de la mesure 'Tem_Moy'. La TM (T_8) de la Figure 55 montre l'analyse scientifique régionale mensuelle de ces mesures après cet ajout. La requête qui réalise une telle analyse peut profiter des valeurs de la mesure actuelle (température moyenne) déjà calculée par la requête qui réalise l'analyse actuelle (R_8). Il reste donc à calculer les valeurs de la nouvelle mesure (température maximale). Ensuite, elle fait la jointure entre les deux comme le montre la requête R_9 suivante :

R_9 :

```
SELECT A.REGION, A.MOISN, TEM_MOY, TEM_MAX
FROM( SELECT  G.REGION, D.MOISN, MAX(TT.TEM_MOY) AS TEM_MAX
      FROM DATES D, GEOGRAPHIE G, TEMPERATURE TT
      WHERE TT.ID_VILLE = G.ID_VILLE
        AND TT.ID_DATE = D.ID_DATE
        AND G.REGION = 'Midi-Pyrénées'
      GROUP BY G.REGION, D.MOISN) A, Result3 R8
WHERE A.REGION = R8.REGION
  AND A.MOISN = R8.MOISN
```

5.2.2.6 Opérateurs de modifications d'une dimension

Ces opérateurs permettent de modifier la liste des paramètres affichés en ligne ou en colonne en insérant un paramètre dans la TM ou en combinant les mesures avec les paramètres des dimensions.

5.2.2.6.1 L'imbrication (NEST)

L'opérateur d'imbrication (**NEST**) permet à l'analyste d'insérer un paramètre dans une dimension affichée en ligne ou en colonne. La syntaxe de l'opérateur est la suivante :

NEST(T_{SRC}, D, p_i, D_{nested}, p_{nested}) = T_{RES}

Où :

- D $\in\{D_L, D_C\}$ est la dimension dans laquelle le paramètre sera ajouté ;
- p_i est un paramètre affiché en ligne (si $D = D_L$) ou en colonne (si $D = D_C$) ;
- D_{nested} est la dimension contenant le paramètre à afficher. Elle doit être associée au fait analysé dans la table source (T_{SRC}) ;
- p_{nested} est le paramètre à insérer dans la dimension (D) ;
- T_{RES} est la table résultante où le nouveau paramètre est positionné comme granularité directement inférieure du paramètre (p_i) en ligne ou en colonne :
  - Si $D = D_L$, la liste de paramètres en ligne est $<P_{max}^{H_L}, \ldots, P_i^{H_L}, p_{nested}, \ldots, P_{min}^{H_L}>$,
  - Si $D = D_C$, la liste de paramètres en colonne est $<P_{max}^{H_C}, \ldots, P_i^{H_C}, p_{nested}, \ldots, P_{min}^{H_C}>$.

Selon la combinaison de dimensions et paramètres (D, p_i, D_{nested}, p_{nested}), l'opérateur **NEST** joue un rôle différent :

- $D_{nested} \in\{D_L, D_C\}$: si $D_{nested} = D$, alors les paramètres (p_i, p_{nested}) doivent appartenir à la même hiérarchie. Nous pouvons distinguer deux cas différents :
  - p_{nested} n'est pas affiché dans la TM source (T_{SRC}), et il est inférieur à tous les paramètres de la hiérarchie affichée dans la table source (T_{SRC}) : dans ce cas, il faut recalculer les valeurs des mesures comme si nous faisions un forage vers le bas (**DRILLDOWN**) au niveau du paramètre p_{nested}. Par exemple, la première application de l'opérateur **NEST** dans la Figure 56 insère le paramètre 'Département' (qui n'est pas affiché dans la table source (T_4)) dans la dimension 'Géographie' de la TM (T_9),
  - p_{nested} est affiché dans la TM source (T_{SRC}), ou il n'est pas inférieur à tous les paramètres de la hiérarchie affichée : alors aucun nouveau calcul n'est nécessaire. Il suffit de modifier la visualisation des données. Par exemple, dans la Figure 56, l'opérateur **NEST**, qui produit la TM (T_{10}), est appliqué sur un paramètre p_{nested} ('Région') déjà affiché dans la table source (T_9). Cela change l'ordre des paramètres affichés dans la table résultante où le paramètre 'Région' devient inférieur au paramètre 'Département' dans la TM (T_{10}),
- $D_{nested} \notin\{D_L, D_C\}$: l'opérateur **NEST** insère donc un paramètre d'une dimension non-affichée dans une dimension affichée dans la TM ce qui permet l'utilisation de plusieurs dimensions dans l'espace 2D de la TM. Par exemple, la TM (T_{11}) de la Figure 56 est produite en insérant le paramètre 'Demi-journée' de la dimension 'Temps' (qui n'est pas affichée dans la table source (T_9)) dans la dimension 'Dates'. La réalisation d'une telle opération nécessite de recalculer les mesures comme si nous faisions un forage vers le bas (**DRILLDOWN**) du paramètre 'All^{Di}' de la dimension D_{nested} ('All^{Temps}') vers le paramètre p_{nested} ('Demi-journée').

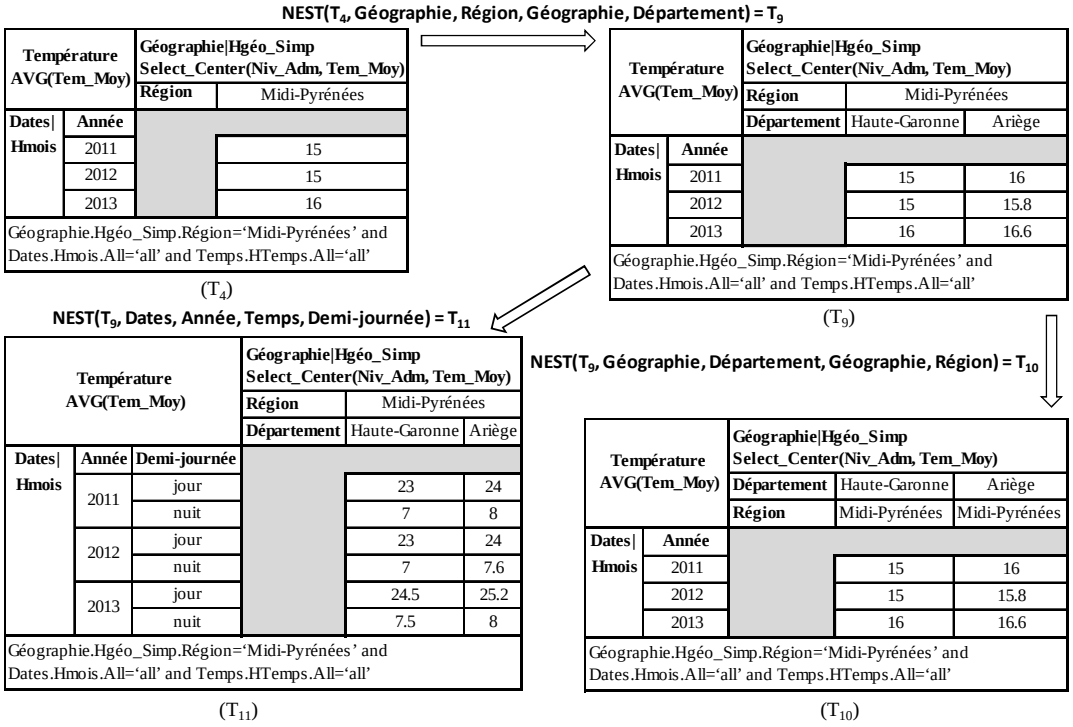


Figure 56 : Opérateur d'imbrication (NEST)

5.2.2.6.2 PUSH

Cet opérateur consiste à convertir un paramètre d'une dimension affichée en mesure dans le fait analysé dans une TM. La syntaxe de l'opérateur est la suivante :

PUSH(T_{SRC}, D, p_i) = T_{RES}

Où :

- D $\in \{D_L, D_C\}$ est la dimension de laquelle le paramètre sera retiré. Elle doit être présentée au moins avec deux paramètres ;
- p_i est le paramètre à convertir en mesure. Il ne doit pas être le paramètre le plus inférieur ;
- T_{RES} est la table résultante affichant le paramètre (p_i) dans la liste des mesure : {m_1, m_2, …, p_i}. Ce paramètre n'est plus affiché en ligne ou en colonne :
  - Si D = D_L, la liste de paramètres en ligne est $<P^{H_L}_{max}, \ldots, P^{H_L}_{i+1}, P^{H_L}_{i-1}, \ldots, P^{H_L}_{min}>$,
  - Si D = D_C, la liste de paramètres en colonne est $<P^{H_C}_{max}, \ldots, P^{H_C}_{i+1}, P^{H_C}_{i-1}, \ldots, P^{H_C}_{min}>$.

5.2.2.6.3 PULL

L'opérateur **PULL** permet de convertir une mesure en paramètre. La syntaxe de l'opérateur est la suivante :

PULL(T_{SRC}, D, m_i) = T_{RES}

Où :

- T_{SRC} est la table source qui doit avoir au moins deux mesures ;

- D $\in \{D_L, D_C\}$ est la dimension dans laquelle la mesure sera ajoutée ;
- m_i est la mesure à convertir en paramètre ;
- T_{RES} est la table résultante. Elle n'affiche plus la mesure (m_i) dans la liste des mesures : $\{m_1, m_2, \ldots, m_{i-1}, m_{i+1}, \ldots\}$. Mais elle l'affiche comme granularité minimale des paramètres affichés de la dimension (D) en ligne ou en colonne :
  - Si D = D_L, la liste de paramètres en ligne est $<P_{max}^{H_L}, \ldots, P_{min}^{H_L}, m_i>$,
  - Si D = D_C, la liste de paramètres en colonne est $<P_{max}^{H_C}, \ldots, P_{min}^{H_C}, m_i>$.

Exemple 10. La Figure 57 montre un exemple d'application des opérateurs **PUSH** et **PULL**. L'opérateur **PUSH** transforme le paramètre 'Région' en mesure dans le fait 'Température'. L'opérateur **PULL** transforme une mesure en paramètre. Afin de replacer le paramètre 'Région' à la position initiale, nous utilisons l'opérateur **NEST**, sans quoi il serait inférieur au paramètre 'Département' comme dans la TM (T_{10}) de la Figure 56.

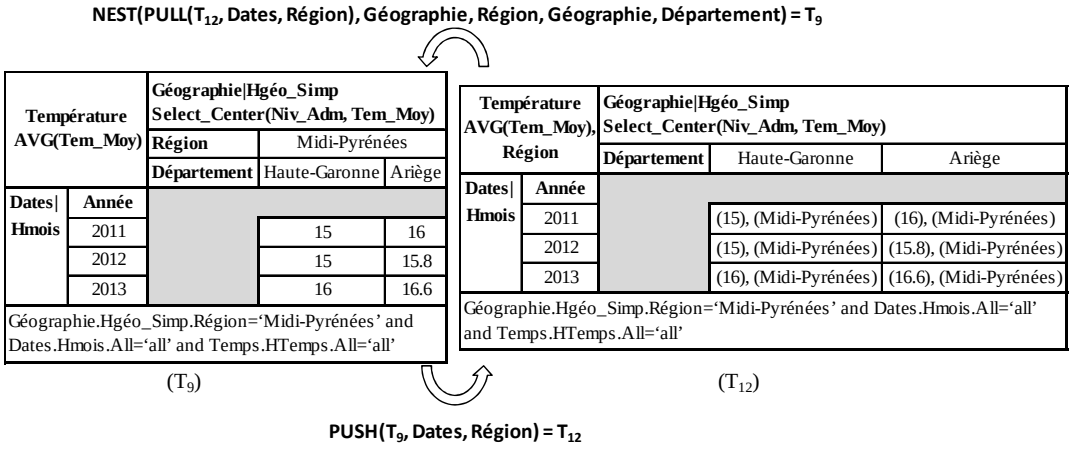


Figure 57 : Opérateurs de modifications d'une dimension (PUSH et PULL)

Les opérateurs de modifications d'une dimension et les fonctions d'agrégation

Grâce à l'automatisation du choix des fonctions d'agrégation, il ne faut pas indiquer dans la spécification de l'opérateur **PULL**, les fonctions d'agrégation associées à la mesure à convertir en paramètre. En outre, l'opérateur **PUSH** ne change que la visualisation de données, c'est pourquoi aucune nouvelle agrégation n'est demandée pour réaliser l'opérateur. Cet opérateur est spécifié et effectué dans un contexte multifonctions de manière identique au contexte uni-fonction. Par ailleurs, il n'y a pas de répercussions d'utilisation de plusieurs fonctions d'agrégation pour la même mesure sur l'opérateur **NEST** (même s'il ajoute un nouveau niveau d'agrégation) autre que l'application des quatre étapes de l'analyse multifonctions (*cf.* § 3.3.3).

5.2.2.7 Opérateurs d'ordonnancements

Ces opérateurs permettent à l'analyste de réorganiser une TM en ordonnant les valeurs affichées ou en permutant leurs positions dans la table.

5.2.2.7.1 La permutation (SWITCH)

L'opérateur **SWITCH** permet un ordre spécifique des valeurs affichées en permutant les positions de deux valeurs d'un paramètre affiché en ligne ou en colonne. Ce qui conduit à

une permutation de deux lignes ou deux colonnes correspondantes avec des répercussions sur les valeurs des paramètres de granularité inférieure. La syntaxe de l'opérateur est la suivante :

SWITCH(T_{SRC}, D, p_i, v_1, v_2) = T_{RES}
Où :
- D ∈{D_L, D_C} ;
- p_i est un paramètre affiché en ligne (si D = D_L) ou en colonne (si D = D_C) ;
- v_1, v_2 sont les deux valeurs du paramètre (p_i) à permuter ;
- T_{RES} est la table résultante.

Exemple 11. L'analyste peut utiliser l'opérateur **SWITCH** afin de mettre en avant les températures moyennes correspondantes à la région 'Midi-Pyrénées' comme le montre la Figure 58. Une deuxième application de la même opération sur la table résultante (T_{13}) retourne les valeurs telles qu'elles étaient auparavant (Figure 58).

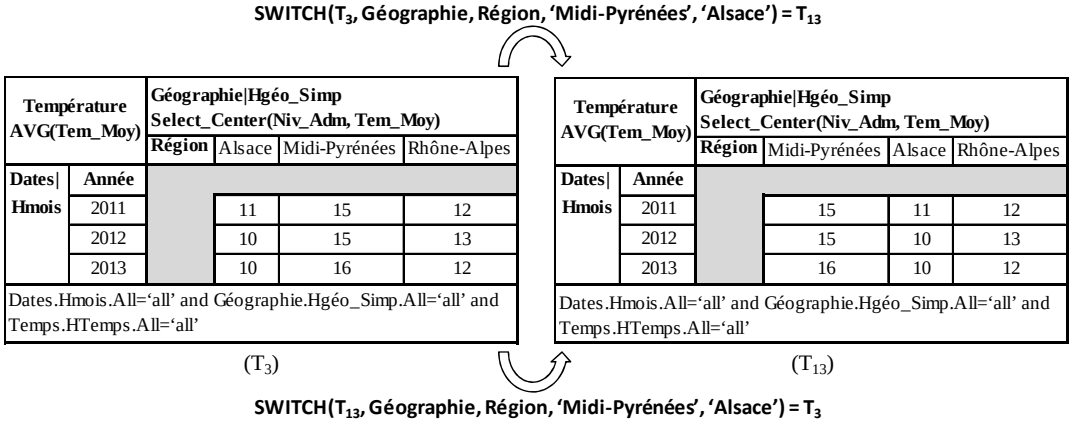


Figure 58 : Opérateur de la permutation (SWITCH)

5.2.2.7.2 ORDER

Cet opérateur consiste à ordonner les valeurs d'un paramètre affiché en ligne ou en colonne dans un ordre croissant ou décroissant avec des répercussions sur les paramètres inférieures. La syntaxe de l'opérateur est la suivante :

ORDER(T_{SRC}, D, p_i, ord) = T_{RES}
Où :
- D ∈{D_L, D_C} ;
- p_i est le paramètre à ordonner ses valeurs. Il est affiché en ligne (si D = D_L) ou en colonne (si D = D_C) ;
- ord ∈ {'asc', 'dsc'}. 'asc' correspond à l'ordonnancement croissant (ascendant) tandis que 'dsc' correspond à l'ordonnancement décroissant (descendant) ;
- T_{RES} est la table résultante qui affiche les valeurs du paramètre (p_i) ordonnées en fonction de la valeur de (ord).

Exemple 12. Si l'analyste veut visualiser les températures moyennes selon un ordre chronologique ou inversement chronologique, il peut appliquer l'opérateur **ORDER** à un paramètre de la dimension 'Dates' comme le montre la Figure 59.

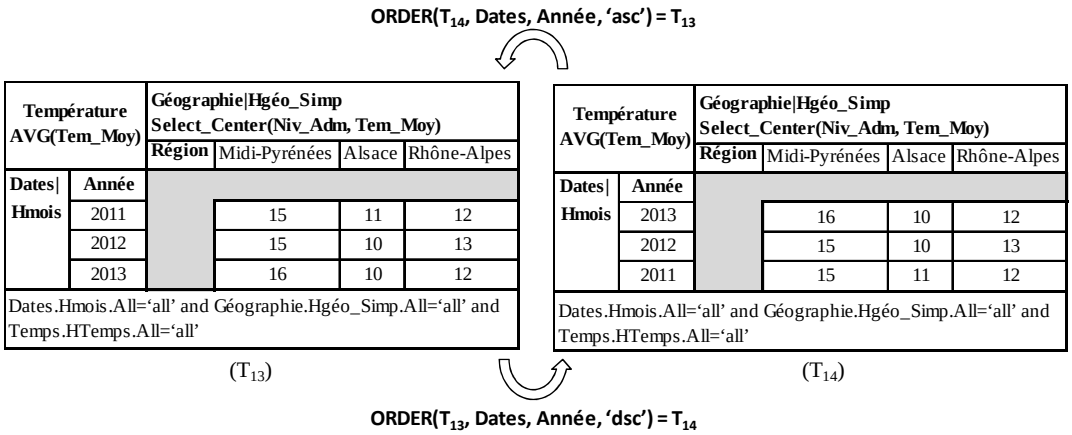


Figure 59 : Opérateur de d'ordonnancements (ORDER)

Les opérateurs d'ordonnancements et les fonctions d'agrégation

La réalisation de ces opérateurs n'a besoin d'aucun calcul parce qu'ils ne changent que la visualisation de données. Par conséquence, il n'y a pas de répercussions d'utilisation de plusieurs fonctions d'agrégation pour la même mesure, ni sur les spécifications de ces opérateurs, ni sur leurs mécanismes internes.

5.2.2.8 Opérateurs d'agrégation

Ces opérateurs permettent à l'analyste d'ajouter (ou supprimer) à une TM une ligne ou une colonne agrégeant les valeurs de la TM.

5.2.2.8.1 AGGREGATE

Cet opérateur permet d'agréger les valeurs des mesures par une fonction d'agrégation (f) dans une nouvelle ligne ou colonne d'une TM. Il réalise l'opérateur Cube proposé par [Gray et al., 1996]. La syntaxe de l'opérateur est la suivante :

AGGREGATE(T_{SRC}, D, p_i, f) = T_{RES}

Où :

- $D \in \{D_L, D_C\}$;
- p_i est un paramètre de la dimension (D) en fonction duquel les valeurs des mesures sont agrégées ;
- f ∈{SUM, COUNT, MAX, MIN,…} est la fonction d'agrégation à utiliser ;
- T_{RES} est la table résultante agrégeant les valeurs dans une nouvelle ligne (si $D = D_C$) ou colonne (si $D = D_L$).

5.2.2.8.2 UNAGGREGATE

Cet opérateur annule toutes les applications précédentes de l'opérateur **AGGREGATE**. La syntaxe de l'opérateur est la suivante :

UNAGGREGATE(T_{SRC}) = T_{RES}

Où T_{RES} est la table résultante qui fait disparaître les agrégations précédentes en supprimant les lignes ou/et les colonnes contenant les calculs d'agrégation.

Les opérateurs d'agrégation et les fonctions d'agrégation

Le contexte multifonctions n'a pas de répercussions sur ces opérateurs parce que leurs opérations d'agrégation représentent des opérations supplémentaires qui ne sont pas liées directement à l'agrégation attendue. Autrement dit, la fonction d'agrégation utilisée (f) n'est pas forcement prédéfinie dans les schémas d'agrégation.

Exemple 13. L'analyste peut souhaiter utiliser l'opérateur **AGGREGATE** afin de présenter les plus petites températures moyennes selon les années et les départements. Ce qui ajoute une ligne et une colonne à la TM comme le montre la Figure 60.

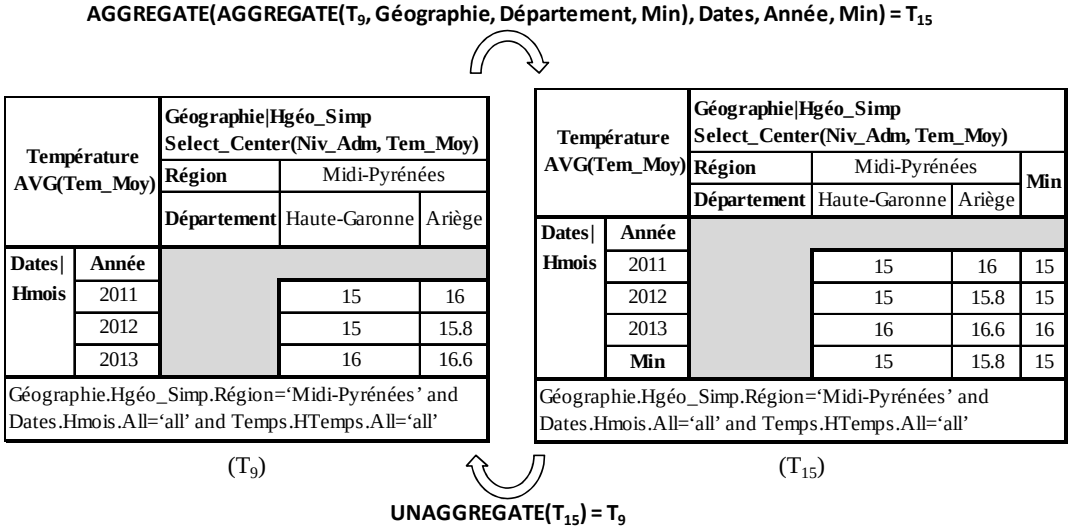


Figure 60 : Opérateurs d'agrégation

5.2.2.9 Opérateurs d'affichage d'agrégation

Ces opérateurs consistent à présenter et cacher les fonctions d'agrégation utilisées pour obtenir les valeurs affichées dans une TM.

5.2.2.9.1 DISPLAYAGGREGATE

Cet opérateur permet d'afficher à l'analyste comment les mesures sont agrégées en présentant leurs fonctions d'agrégation (avec leurs entrées, leurs ordres d'exécution et leurs contraintes d'agrégation). Cette présentation sert à montrer comment l'analyse courante est calculée. La syntaxe de l'opérateur est la suivante :

DISPLAYAGGREGATE(T_{SRC}, [m_i]) = T_{RES}

Où :

- m_i est la mesure à présenter ses fonction d'agrégation ;
- T_{RES} est la table résultante qui affiche les fonctions associées à la mesure m_i (Aggregate(m_i)) utilisées pour agréger la mesure aux niveaux d'agrégation actuels.

L'utilisation de la notation [ ] indique que le nom de la mesure (m_i) est optionnel dans la définition de l'opérateur. Un opérateur **DISPLAYAGGREGATE** sans préciser une mesure, affiche les fonctions d'agrégation de toutes les mesures analysées (Figure 61). Cet opérateur respecte les règles de la simplification des fonctions d'agrégation discutées précédemment (§ 5.2.1).

5.2.2.9.2 HIDEAGGREGATE

La présentation des fonctions d'agrégation pour plusieurs mesures dans une seule TM peut perturber les analystes surtout s'il y a des mesures qui sont calculées à partir d'autres mesures comme dans la TM (T_8) de la Figure 55. Dans ce cas nous proposons d'utiliser l'opérateur **HIDEAGGREGATE** (Figure 61). Cet opérateur est l'inverse de l'opérateur précédent. Il cache les fonctions d'agrégation utilisées pour une mesure analysée dans une TM. La syntaxe de l'opérateur est la suivante :

HIDEAGGREGATE(T_{SRC}, [m_i]) = T_{RES}

Où :

- m_i est la mesure à cacher ses fonction d'agrégation ;
- T_{RES} est la table résultante qui affiche le nom de la mesure ('m_i') au lieu de ses fonctions d'agrégation (Aggregate(m_i)).

De la même manière que l'opérateur **DISPLAYAGGREGATE, HIDEAGGREGATE** sans préciser une mesure, cache les fonctions d'agrégation de toutes les mesures analysée (Figure 61).

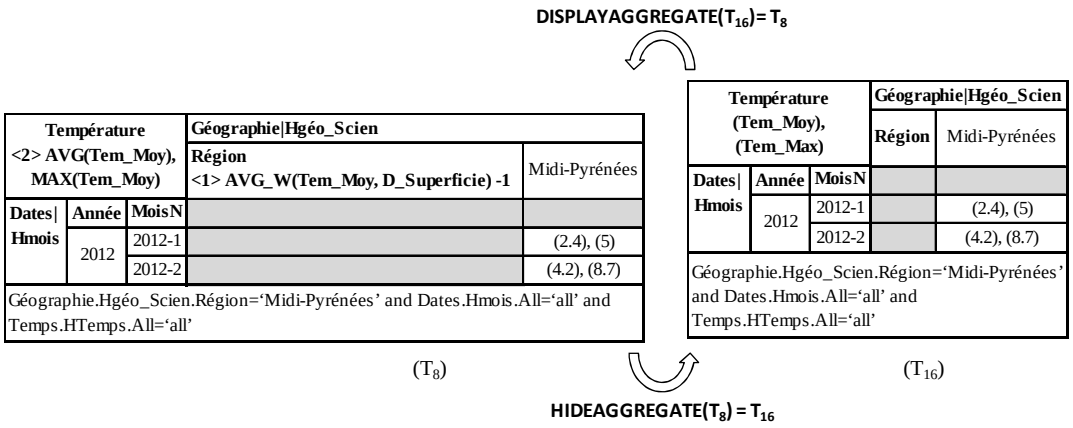


Figure 61 : Opérateurs d'affichage d'agrégation

Les opérateurs d'affichage d'agrégation et les fonctions d'agrégation

Ces opérateurs sont particulièrement utilisés dans un contexte multifonction. Ils ne changent que la visualisation des fonctions d'agrégation. Ce qui ne demande aucun calcul de données.

5.3 CONCLUSION

Dans ce chapitre, nous avons présenté une définition de TM et un langage d'interrogation des données adaptés à notre modèle multidimensionnel multifonctions. Notre proposition se base sur la structure de visualisation de la TM et l'ensemble des opérateurs de manipulation des bases de données multidimensionnelles proposé dans [Ravat et al., 2008].

Nous avons adapté la définition de la TM au contexte multifonctions en ajoutant les fonctions d'agrégations associées aux mesures analysées (Aggregate(m_i)) et toutes les dimensions associées au sujet d'analyse même si elle ne sont pas affichées dans la TM [Hassan et al., 2013b], [Hassan et al., 2013c]. Cette adaptation consiste à simplifier autant que possible la présentation des contraintes d'agrégation des fonctions d'agrégation et leur ordre d'exécution, et à réduire le nombre des fonctions affichées.

Les opérateurs de notre langage d'interrogation permettent de spécifier une analyse multidimensionnelle à partir des éléments structurels (faits, dimensions, hiérarchies) en utilisant l'opérateur de construction. Ensuite, l'analyste utilise les autres opérateurs interactivement pour la modifier et l'affiner.

Nous avons adapté nos opérateurs d'analyse OLAP à l'utilisation de plusieurs fonctions d'agrégation pour la même mesure. Ces extensions sont justifiées par les raisons suivantes :

- **L'automatisation du choix des fonctions d'agrégation** qui simplifie la spécification de plusieurs opérateurs d'analyse OLAP où les fonctions d'agrégation utilisées ne doivent plus être spécifiées ;
- **La différence entre les fonctions utilisées sur les hiérarchies de la même dimension** influence la spécification de certains opérateurs parce qu'elle nécessite de préciser les hiérarchies utilisées sur l'ensemble des dimensions associées au sujet d'analyse (même si la dimension n'est pas affichée dans la TM) ;
- **L'existence d'un ordre d'exécution entre les fonctions d'agrégation** a des impacts sur le mécanisme interne des opérateurs qui changent un niveau d'agrégation par un autre niveau supérieur. Ces impacts limitent la possibilité de profiter des valeurs déjà agrégées dans la TM précédente [Hassan et al., 2013b], [Hassan et al., 2013c].

Dans le cadre de cette adaptation, nous proposons deux opérateurs (**DISPLAYAGGREGATE** et **HIDEAGGREGATE**) à la liste d'opérateurs classiques afin de montrer et cacher les fonctions d'agrégation dans la TM.

Le Tableau 16 présente une synthèse de cette adaptation.

Tableau 16 : Synthèse d'adaptation des opérateurs OLAP au contexte multifonctions

Opérateur classique	Cause	Changement
DISPLAY(F,{$f_1(m_1)$, $f_2(m_2)$, …}, D_L, H_L, D_C, H_C)	-l'automatisation d'agrégation -la différence entre les fonctions sur les hiérarchies	-spécification □ DISPLAY(F, {m_1, m_2, … }, D_L, H_L, D_C, H_C, D_3, H_3,…, D_m, H_m)
ROLLUP(T_{SRC}, D, p_{sup})	-l'ordre d'exécution	-mécanisme interne ROLLUP(T_{SRC}, D, p_{sup})
DROTATE(T_{SRC}, D_{old}, D_{new} ,H_{new})	-l'ordre d'exécution	-mécanisme interne DROTATE(T_{SRC}, D_{old}, D_{new} ,H_{new})
HROTATE(T_{SRC}, D, H_{new})	-la différence entre les fonctions sur les hiérarchies -l'ordre d'exécution	-mécanismes internes HROTATE(T_{SRC}, D, H_{new})
FROTATE (T_{SRC}, F_{new}, {$f'_1(m'_1)$, $f'_2(m'_2)$, …})	-l'automatisation d'agrégation -la différence entre les fonctions sur les hiérarchies	-spécification □ FROTATE(T_{SRC},F_{new},{m'_1, m'_2, …}, D_3, H_3,…, D_m, H_m)
ADDM(T_{SRC}, $f_i(m_i)$)	-l'automatisation d'agrégation	-spécification □ ADDM(T_{SRC}, m_i)
DELM(T_{SRC}, $f_i(m_i)$)	-l'automatisation d'agrégation	-spécification □ DELM(T_{SRC}, m_i)
PULL (T_{SRC}, D, $f_i(m_i)$)	-l'automatisation d'agrégation	-spécification □ PULL(T_{SRC}, D, m_i)
-	-	DISPLAYAGGREGATE(T_{SRC})
-	-	HIDEAGGREGATE(T_{SRC})

6. CHAPITRE VI : IMPLANTATION ET VALIDATION

6.1 INTRODUCTION

Ce chapitre a pour objectif de présenter le prototype que nous avons développé, afin de montrer la faisabilité de notre approche et de valider nos propositions. L'objectif de notre prototype, appelé *OLAP-Multi-Functions*, est de fournir un système d'analyses OLAP multifonctions qui permet de concevoir une BDM à agrégations multiples et différenciées, ainsi que de superviser les manipulations OLAP effectuées par un analyste.

Plan du chapitre. Ce chapitre est organisé comme suit. La deuxième section présente notre prototype en décrivant les modules composant l'architecture du prototype. Nous détaillions les principaux modules : les interfaces, le méta-schéma et le générateur des requêtes. *OLAP-Multi-Functions* nous sert de plateforme à partir de laquelle nous pouvons mener des expériences ; nous détaillons les expériences menées dans la troisième section.

6.2 PROTOTYPE *OLAP-MULTI-FUNCTIONS*

La fonctionnalité principale d'*OLAP-Multi-Functions* est de visualiser et de faciliter l'intégration des fonctions d'agrégation dans le schéma multidimensionnel. Notre prototype s'appuie sur des représentations graphiques de la BDM. Un langage graphique de conception permet d'avoir une vision synthétique et proche de la réalité modélisée, rendant les représentations plus claires que celles d'autres types de langages (textuels ou tabulaires) [Le Parc, 1997]. Afin de développer ce prototype, nous avons utilisé Java 7 au-dessus du système de gestion de bases de données relationnelles Oracle 12c.

Dans la suite, nous présentons l'architecture générale de notre prototype.

6.2.1 Architecture d'*OLAP-Multi-Functions*

Notre prototype ne se charge pas de construire le modèle logique R-OLAP. Il considère que les données de base (faits et dimensions) sont déjà stockées dans des tables relationnelles et que le schéma structurel (faits, dimensions et hiérarchies) est déjà défini. Le prototype *Graphic-OLAP* [Ravat et al., 2008] développé dans l'équipe permet d'effectuer ces tâches et produit une BDM compatible avec *OLAP-Multi-Functions.*

La Figure 62 illustre l'architecture générale de notre prototype. Selon cette architecture le prototype est organisé en deux niveaux :

- **Le niveau « Interfaces »** qui repose sur un ensemble d'interfaces graphiques :
  - Le module *Constructeur* permet d'intégrer graphiquement les fonctions d'agrégation dans le schéma multidimensionnel,

- Le module *Visualisation* affiche le schéma structurel (*cf.* § 3.2.4.1) et les schémas d'agrégation (*cf.* § 3.2.4.2) aux utilisateurs,
- Le module *Analyseur* aide à spécifier graphiquement les analyses souhaitées et affiche leurs résultats.

– **Le niveau « Stockage »** qui comprend trois espaces de stockage :

- Une base de données *R-OLAP* stocke la BDM (faits et dimensions). Elles représentent l'implantation du schéma au niveau logique présentée précédemment dans le chapitre 4 (*cf.* § 4.2),
- Une méta-base décrit le schéma structurel et les schémas d'agrégation de la BDM,
- Un *Générateur SQL* élabore les requêtes SQL correspondantes aux analyses exprimées par les utilisateurs. Il est implanté sous la forme de procédures stockées PL/SQL.

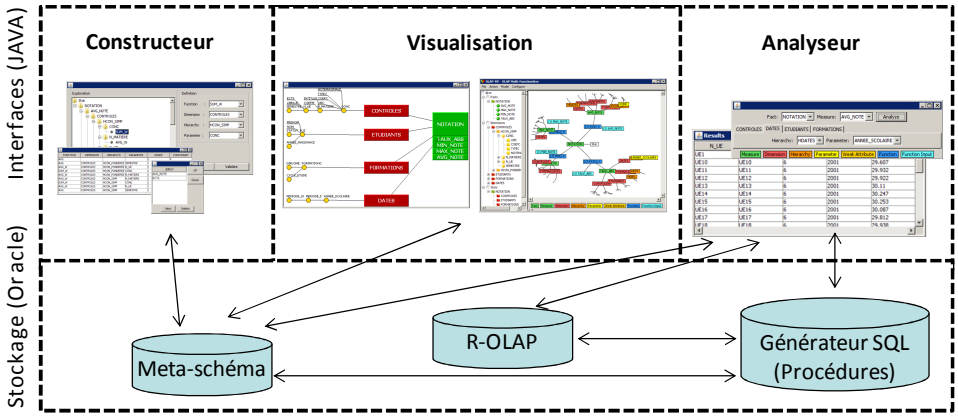


Figure 62 : Architecture de prototype *OLAP-Multi-Functions*

Dans la suite, nous détaillons les interfaces et les espaces de stockage.

6.2.2 Interfaces

Les interfaces permettent aux utilisateurs d'interagir avec la BDM. Les interfaces des modules de constructeur et de visualisation sont exploitées par les concepteurs de la BDM, tandis que les analystes utilisent les interfaces des modules de visualisation et de l'analyseur.

6.2.2.1 Constructeur

Il comprend un ensemble d'interfaces graphiques permettant de définir les quatre types de fonctions d'agrégation (générale, multiple dimensionnelle, multiple hiérarchique et différentiée), leur ordre d'exécution et les éventuelles contraintes d'agrégation.

Les fonctions d'agrégation peuvent s'intégrer dans le schéma multidimensionnel en utilisant l'interface graphique de la Figure 63. Après avoir choisi la mesure concernée dans l'arborescence de gauche de l'interface de la Figure 63, nous définissons :

- La fonction générale en ne déterminant que la fonction d'agrégation ;
- La fonction multiple dimensionnelle en définissant la fonction et la dimension ;
- La fonction multiple hiérarchique en spécifiant la fonction, la dimension et la hiérarchie ;
- La fonction différentiée en déterminant la fonction, la dimension, la hiérarchie et le paramètre.

La Figure 63 montre la définition de la fonction multiple dimensionnelle (SUM) de la mesure 'Précip'. Cette interface offre également la possibilité de modifier une fonction d'agrégation en changeant la fonction utilisée ou bien de supprimer une fonction en choisissant un nom de fonction vide.

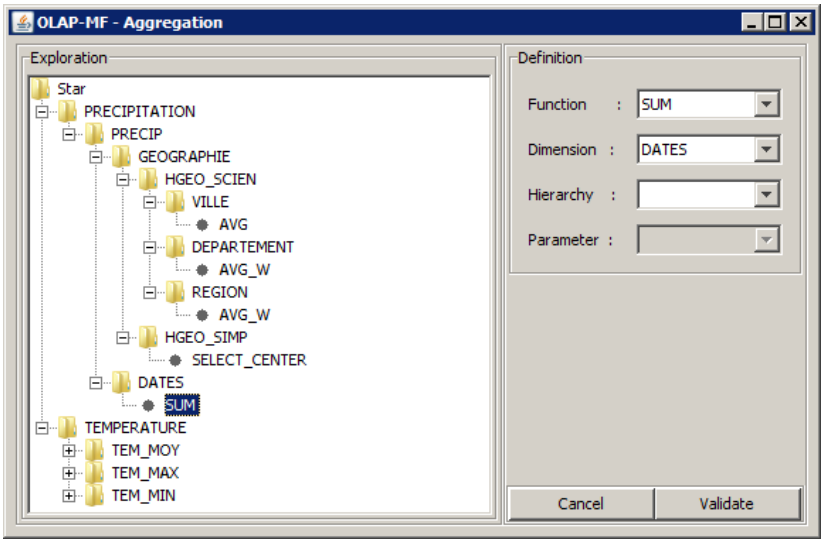


Figure 63 : Définition des fonctions d'agrégation

Nous déterminons les contraintes d'agrégation et l'ordre d'exécution des fonctions d'agrégation d'une mesure en utilisant une autre interface graphique (la fenêtre en arrière plan de la Figure 64). Dans le cas des fonctions qui prennent en entrée des valeurs autres que les mesures, les arguments des fonctions peuvent être spécifiés par la fenêtre en premier plan de la Figure 64. Les boutons (Up et Down) servent à ordonner ces arguments.

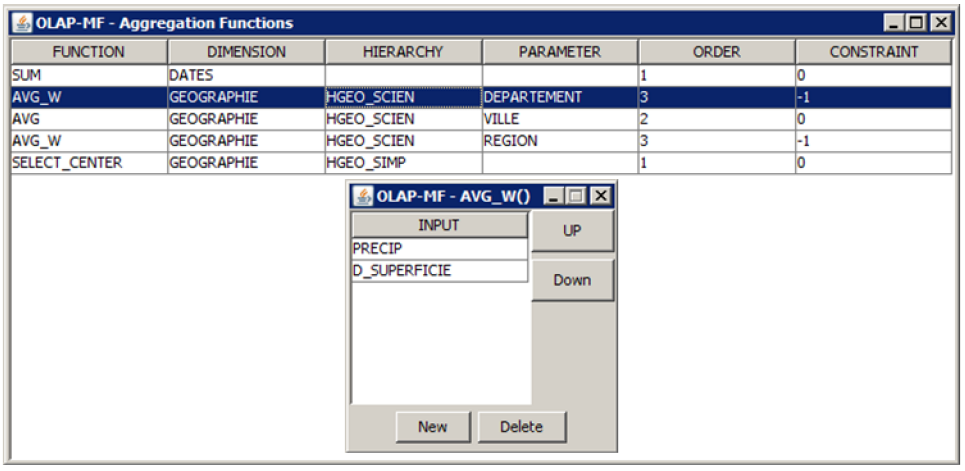


Figure 64 : Définition des arguments, contraintes et ordre d'exécution des fonctions

6.2.2.2 Visualisation

Le schéma structurel est visualisé sous la forme d'un graphe en constellation (Figure 65) basé sur des formalismes graphiques des faits, des dimensions et des hiérarchies introduits dans [Ravat et al., 2007b] et [Ravat et al., 2008]. Les différents schémas d'agrégation sont quant

à eux, visualisés sous la forme d'un graphe hyperbolique (Figure 66) qui permet de se concentrer sur une partie du schéma, sans détailler les autres parties en gardant toujours la possibilité de changer la partie détaillée. Par exemple, la Figure 66 détaille l'agrégation de la mesure 'Précip' sur la dimension 'Dates' et la hiérarchie 'Hgéo_Simp' de la dimension 'Géographie', tandis que l'agrégation sur la hiérarchie 'Hgéo_Scien' n'est pas détaillée.

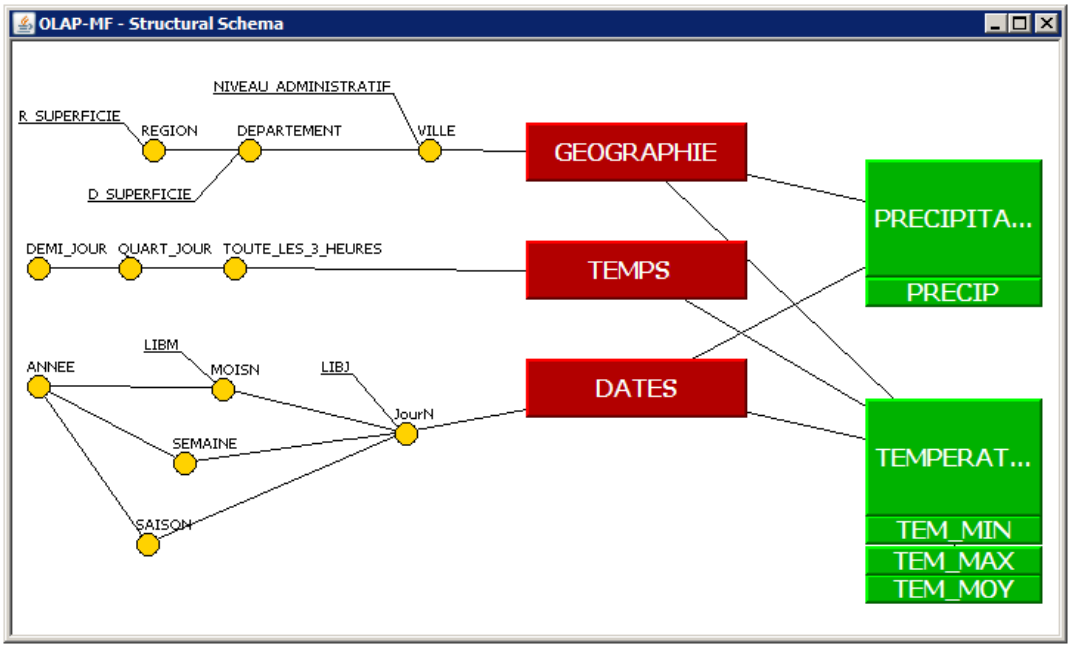


Figure 65 : Schéma structurel

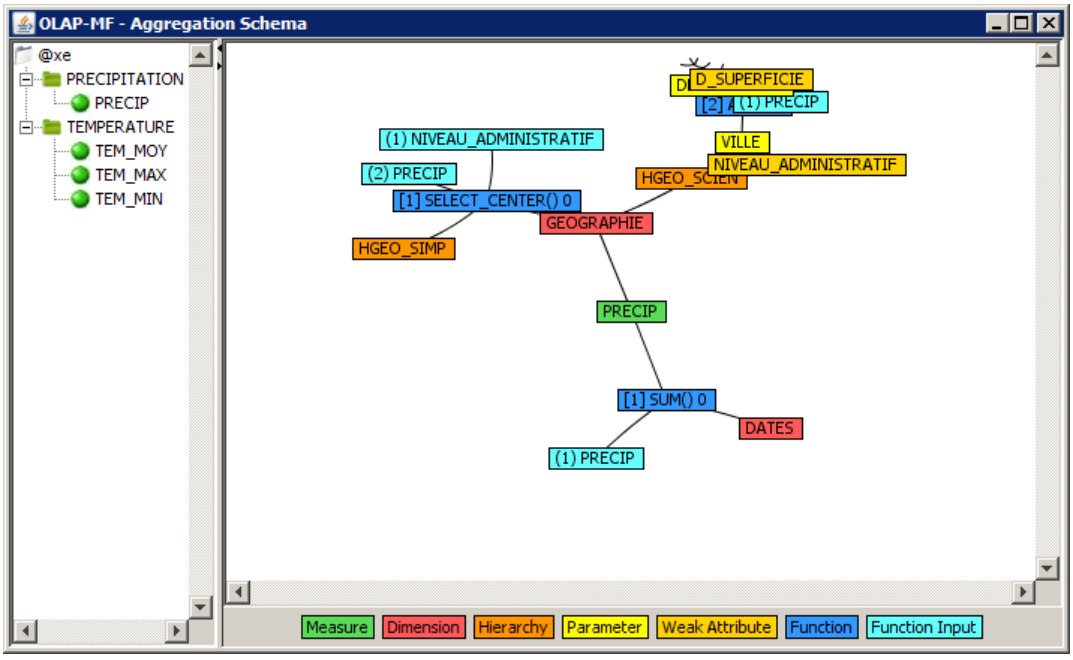


Figure 66 : Schéma d'agrégation

6.2.2.3 Analyseur

Le rôle principal de l'analyseur est de faciliter l'interrogation des données. Or, l'analyste ne manipule que des concepts multidimensionnels. Ainsi, la complexité de l'agrégation et la structure logique de la BDM sont cachées.

L'analyste sélectionne graphiquement dans l'interface de l'analyse (la fenêtre au premier plan de la Figure 67), la mesure et le niveau d'agrégation souhaité pour chaque dimension. L'analyseur transfère les interactions d'analyste au générateur de requêtes SQL. Les requêtes sont envoyées au SGBD. Puis, elles sont calculées et les résultats sont affichés dans une fenêtre séparée sous forme d'une table (à gauche de la fenêtre, arrière plan de la Figure 67). La fenêtre de résultats présente également l'agrégation correspondante à l'analyse réalisée (à droite de la fenêtre, arrière plan de la Figure 67).

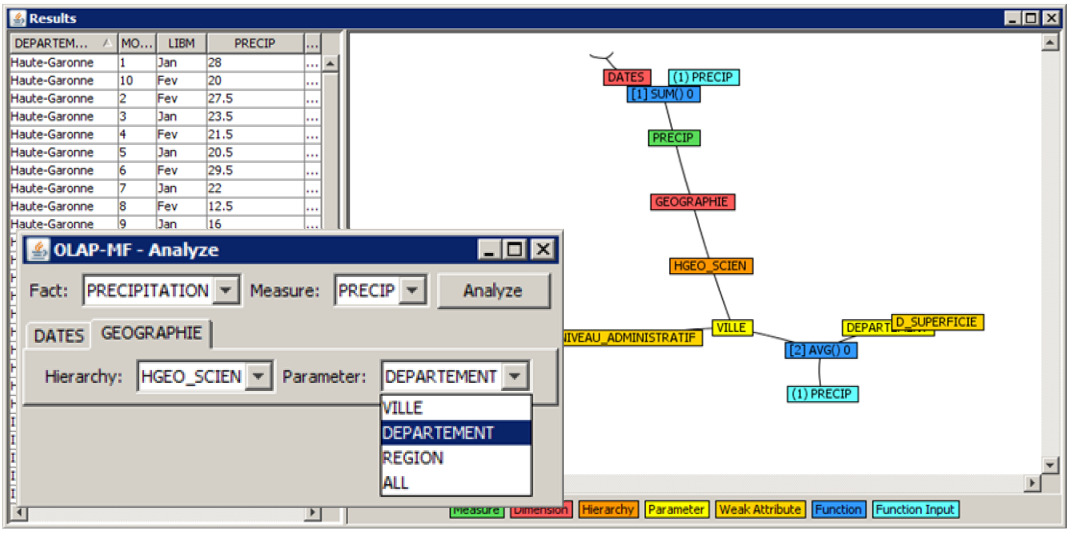


Figure 67 : Analyseur

L'interface de l'analyse (la fenêtre en avant de la Figure 67) offre aux analystes la possibilité d'effectuer plusieurs manipulations graphiques simples. Ces manipulations peuvent être équivalentes à plusieurs opérateurs d'analyse OLAP. Dans la suite, nous présentons certaines d'entre elles.

- Le changement d'un paramètre sélectionné par un autre paramètre inférieur est équivalent à une opération du forage vers le bas (**DRILLDOWN**) ;
- Le changement d'un paramètre sélectionné par un autre paramètre supérieur est équivalent à une opération du forage vers le haut (**ROLLUP**) ;
- Le changement à la fois
  - d'un paramètre sélectionné par le paramètre extrémité All^{Di} sur une dimension D_i, et
  - d'un paramètre extrémité All^{Dj} sélectionné par un paramètre inférieur sur une autre dimension D_j,

  est équivalent à une opération de rotation des dimensions (**DROTATE**) ;
- Le changement d'une hiérarchie sélectionnée par une autre est équivalent à une opération de rotation des hiérarchies (**HROTATE**) ;
- Le changement du fait analysé par un autre est équivalent à une opération de rotation des faits (**FROTATE**).

D'autres manipulations concernant la visualisation de données (comme l'ordonnancement des données (**ORDER**) et la permutation des colonnes (**NEST**)) peuvent être effectuées dans l'interface de résultats.

6.2.3 Stockage

Dans cette section, nous ne détaillions que le méta-schéma et le générateur de requêtes SQL. Les tables relationnelles stockant les données des faits et des dimensions, implantées selon la méthode R-OLAP, sont présentées en détail dans le chapitre 4. Ces tables peuvent former un schéma en flocon ou en étoile [Kimball, 1996] (*cf.* § 4.2).

6.2.3.1 Méta-Schéma

Ce méta-schéma décrit les structures du schéma multidimensionnel (faits, dimensions et hiérarchies) ainsi que les fonctions d'agrégation, l'ordre d'exécution et les contraintes d'agrégation pour construire des requêtes SQL valides et cohérentes.

La Figure 68 montre ce méta-schéma selon le diagramme de classes UML. Les classes de couleur claire décrivent les éléments structurels (schéma structurel, *cf.* § 3.2.4.1), et les classes bleues/grisées décrivent les mécanismes d'agrégations (schémas d'agrégation, *cf.* § 3.2.4.2)).

Selon ce méta-schéma, un fait se compose d'une ou plusieurs mesures. Il peut être analysé selon plusieurs dimensions. Chaque dimension comprend des hiérarchies, des paramètres et des attributs faibles. Un paramètre peut appartenir à plusieurs hiérarchies qui sont ordonnées en niveaux d'agrégation. Chaque niveau peut être associé à un ensemble d'attributs faibles.

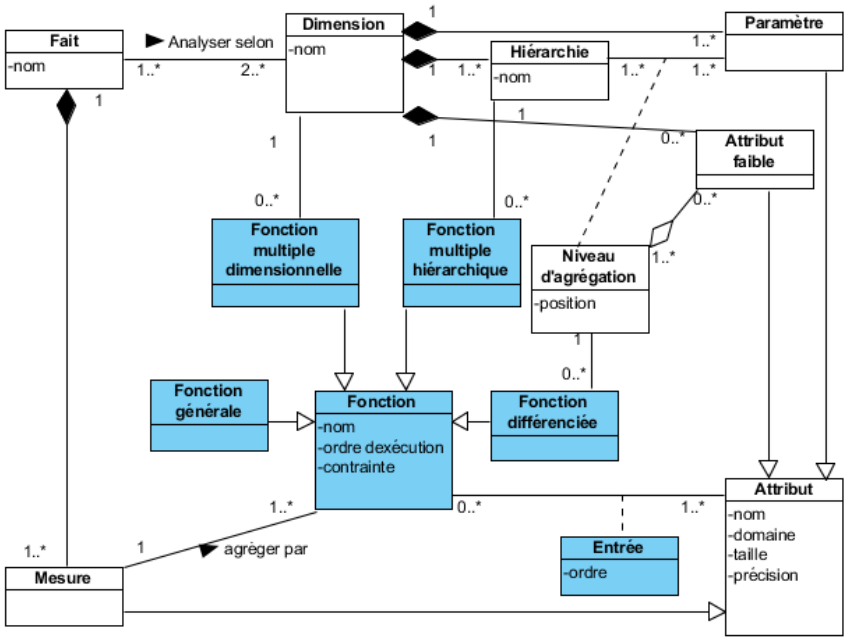


Figure 68 : Méta-schéma

Une mesure peut être agrégée par plusieurs fonctions d'agrégation (classe *Fonction*) mais une fonction d'agrégation est associée à une seule mesure. La classe *Fonction* possède un attribut *contrainte* qui est utilisé pour forcer l'agrégation en indiquant le niveau inférieur à partir

duquel l'agrégation considérée doit être calculée. L'attribut *ordre d'exécution* est utilisé pour ordonner l'exécution des fonctions d'agrégation non-commutatives. Une fonction prend au moins une entrée. Cette entrée est, soit une mesure, soit un paramètre, soit un attribut faible.

Quatre types de fonctions d'agrégation héritent de la classe *Fonction* :

- *Fonction générale* est liée uniquement à une mesure (une mesure ayant au maximum une fonction d'agrégation générale). Cette fonction est utilisée pour agréger les valeurs de mesure dans l'espace multidimensionnel où il n'y a aucune autre fonction d'agrégation spécifique ;
- *Fonction multiple dimensionnelle* est liée à une mesure et une dimension sur laquelle cette fonction est appliquée ;
- *Fonction multiple hiérarchique* est liée à une mesure et une hiérarchie sur laquelle nous appliquons cette fonction ;
- *Fonction différenciée* est liée à un niveau d'agrégation pour agréger les valeurs de mesure entre ce niveau et le niveau directement supérieur dans la même hiérarchie.

Toutefois, chaque dimension, chaque hiérarchie et chaque niveau d'agrégation peuvent avoir plusieurs fonctions d'agrégation ; une pour chaque mesure différente.

6.2.3.2 Générateur de requêtes SQL

Pour superviser les analyses, le prototype dispose d'un générateur de requêtes SQL. Nous l'avons réalisé sous forme d'une procédure stockée. Le générateur traduit les interactions des analystes en générant un script SQL exécutable dans le contexte d'implantation R-OLAP. L'analyste paramètre le calcul d'agrégation qu'il souhaite réaliser : il doit préciser la mesure et les niveaux d'agrégation souhaités. Notre prototype les stocke dans une table temporaire 'niveaux d'agrégation'. La procédure du générateur pour accomplir sa tâche accède à cette table temporaire, au méta-schéma et au schéma R-OLAP.

Le processus de génération comprend quatre étapes correspondant aux étapes de l'analyse multifonctions (*cf.* § 3.3.3), comme l'illustre la Figure 69 en diagramme BPMN (Business Process Modeling Notation).

- *Détecter les tables du modèle logique R-OLAP* : à partir du nom de fait à analyser, cette étape identifie les tables relationnelles utilisées pour stocker les données pour l'analyse et les liens d'intégrité référentielle entre elles. Ensuite, elle stocke ces informations dans une table temporaire ;
- *Déterminer les fonctions d'agrégation* : à partir du méta-schéma et des niveaux d'agrégation demandés, cette étape identifie les fonctions d'agrégation à appliquer pour calculer les données de l'analyse. Puis elle les stocke avec leur ordre d'exécution et leurs contraintes d'agrégation dans une nouvelle table temporaire. Cette étape correspond à l'étape (déterminer les fonctions d'agrégation) de l'analyse multifonctions ;
- *Simplifier les fonctions d'agrégation* : cette étape consiste à détecter d'éventuels calculs redondants, c'est-à-dire une répétition inutile ou invalide d'une fonction d'agrégation. Pour cela il faut supprimer les répétitions de la même fonction dans la table temporaire, c'est-à-dire supprimer les fonctions qui ont le même nom, les mêmes arguments et le même ordre d'exécution. Cette étape correspond à l'étape (traiter l'ordre d'exécution) de l'analyse multifonctions ;
- *Obtenir script SQL* : à partir du méta-schéma et les tables temporaires, cette étape génère la requête SQL finale. Le prototype envoie cette requête au SGBD qui la calcule et restitue les données. Cette étape correspond à l'étape (Effectuer l'analyse) de l'analyse multifonctions.

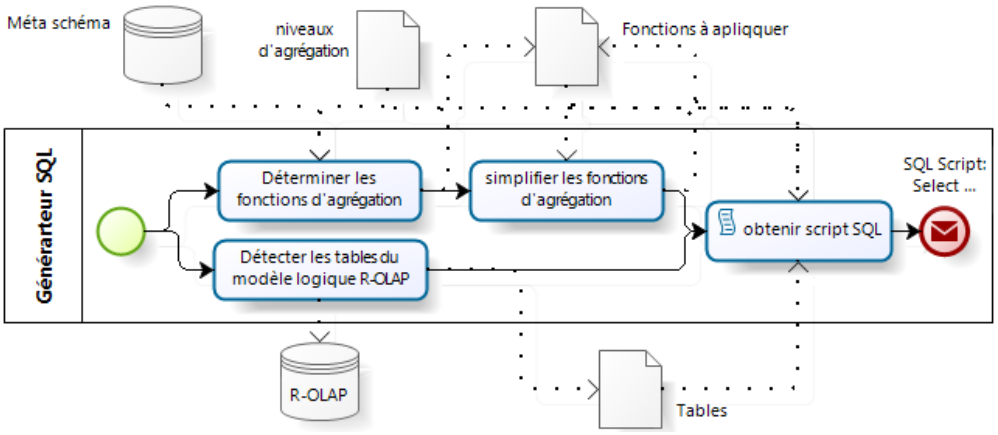


Figure 69 : Générateur de requêtes SQL

6.2.4 Discussion

Dans cette section, nous présentons les avantages de notre prototype *OLAP-Multi-Functions* par rapport à un des outils commerciaux les plus utilisés: « Business Objects » (BO).

6.2.4.1 « Business Objects »

A notre connaissance, la limite principale de BO est l'utilisation d'une seule fonction d'agrégation pour chaque mesure [BO XI 3.1 SP3, 2010], [BO XI 3.1 SP6, 2013]. Pour savoir jusqu'à quel point nous pouvons surmonter ce problème, nous avons appliqué notre exemple de météo (exemple de motivation *cf.* § 1.3.1) en utilisant BO. Nous avons associé la fonction d'agrégation (AVG) à la mesure 'Tem_Moy'. Nous pouvons ainsi effectuer toutes les agrégations possibles sur les dimensions 'Dates' et 'Temps' ; par exemple, nous pouvons analyser les températures moyennes mensuelles par ville et demi-journée.

Cependant pour agréger les températures moyennes sur la dimension 'Géographie', nous utilisons d'autres fonctions ; par exemple, nous utilisons une fonction d'agrégation non-standard, qui est la moyenne pondérée (AVG_W), pour calculer les températures régionales (au niveau 'Région' de la hiérarchie 'Hgéo_Scien'). Pour résoudre ce problème, il y a deux propositions :

- **L'utilisation des mesures calculées (dérivées)** [BO XI 3.1 SP3, 2010] : cette proposition implique de définir une nouvelle mesure 'Tem_Moy_Région' calculée par la formule (É5), définie dans la section 1.3.2.1. Le problème de cette proposition est que cette formule (« Select: » dans la Figure 70) ne sera pas utilisée pour calculer la mesure au niveau 'Région' mais au niveau de base 'Ville', puis pour calculer la mesure au niveau 'Région', sa fonction d'agrégation propre sera utilisée pour agréger les valeurs ;
- **L'utilisation des variables** [BO XI 3.1 SP6, 2013] : l'avantage de cette proposition est que la variable peut utiliser des valeurs agrégées contrairement à la mesure calculée qui utilisent uniquement les valeurs de base. Le problème est que si la variable utilise des valeurs autres que la mesure, ces valeurs doivent être utilisées dans l'analyse, sinon il y aura des erreurs. Par exemple, la variable 'Moy_Var' (Figure 71) est calculée par la formule (É5) mais les valeurs de 'D_Superficie' ne sont pas présentées dans l'analyse, donc il y a des erreurs (Figure 71). Pour surmonter ce problème et obtenir les résultats demandés, nous pouvons définir deux nouvelles mesures : la première M_1 = SUM(D_Superficie * Tem_Moy), la deuxième M_2 = SUM(D_Superficie) et la variable devient alors Moy_Var = M_1 / M_2.

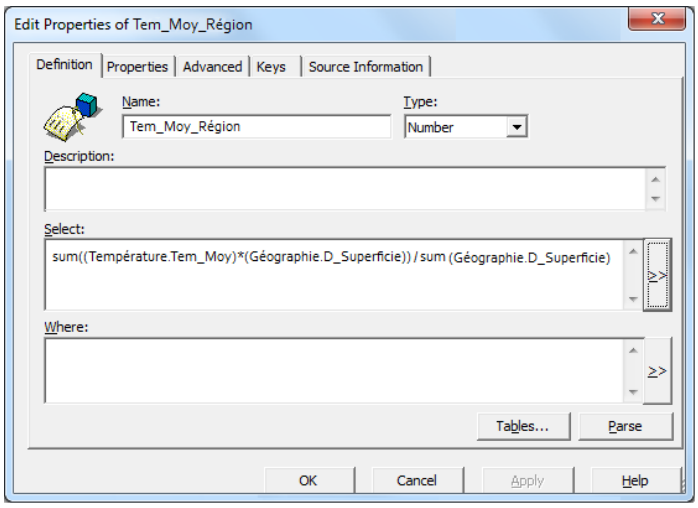


Figure 70 : L'utilisation des mesures calculées dans BO

Date	2011		2012		2013	
Région	Tem_Moy	Moy_Var	Tem_Moy	Moy_Var	Tem_Moy	Moy_Var
Alsace	12,00	#IERR	12,67	#IERR	12,00	#IERR
Midi-Pyrénées	13,00	#IERR	13,00	#IERR	15,00	#IERR
Rhône-Alpes	11,00	#IERR	12,00	#IERR	12,00	#IERR

Figure 71 : L'utilisation des variables dans BO

Ainsi, en utilisant les variables, on peut calculer :

[1] Une fonction non-standard,
[2] Une deuxième agrégation à partir des résultats de la fonction d'agrégation associée à la mesure. Ceci est similaire à l'association de deux fonctions d'agrégation avec une seule mesure.

Lorsque une variable est utilisée pour un niveau spécifique (comme la variable 'Moy_Var' qui est utilisée pour le niveau 'Région'), il n'y a pas de contrainte qui interdit de l'utiliser pour un autre niveau, ce qui donnera un résultat erroné. Une autre limite d'utilisation des variables est que nous ne pouvons pas utiliser une variable pour calculer une autre variable sans afficher la première dans l'analyse, sinon il y aura des erreurs (Figure 71). Ainsi, nous ne pouvons pas nous baser sur la variable 'Moy_Var' pour calculer les températures moyennes au niveau 'All^{Géographie}'.

6.2.4.2 OLAP-Multi-Functions

Notre prototype intègre plusieurs fonctions d'agrégation dans le modèle multidimensionnel. Il pallie les limites principales de BO : l'utilisation des fonctions d'agrégation non-standard et l'utilisation de plusieurs fonctions d'agrégation pour la même mesure.

Pour utiliser les fonctions d'agrégation non-standard : la moyenne pondérée (utilisée pour l'agrégation scientifique) et la sélection de ville représentative (utilisée pour l'agrégation simple) (*cf.* § 1.3.2.1), nous avons implanté nos fonctions d'agrégation (AVG_W et SELECT_CENTER) décrites précédemment (*cf.* § 2.3.2.1) :

- Nous avons créé deux types (un pour chaque fonction) d'objet Oracle (classe) qui implémentent les quatre méthodes de l'interface ODCIAggregate : ODCIAggregateInitialize, ODCIAggregateIterate, ODCIAggregateMerge et ODCIAggregateTerminate. Ces méthodes correspondent aux opérations internes que chaque fonction d'agrégation accomplit (*Initialize*, *Iterate*, *Merge*, *Terminate*) [Oracle 12c, 2013] ;
- Nous avons ensuite créé nos fonctions d'agrégation AVG_W et SELECT_CENTER en reposant sur nos précédents types d'objet.
  - AVG_W reçoit un paramètre (TYPE Data_Weighted AS OBJECT (valeur NUMBER, poids NUMBER)) composé de la donnée à agréger et de sa pondération,
  - SELECT_CENTER reçoit un paramètre (TYPE Lev_Data AS OBJECT (level NUMBER, value NUMBER)) composé de la donnée à agréger associée à son niveau administratif.

L'utilisation des types d'objet (Data_Weighted et Lev_Data) est nécessaire parce que les fonctions d'agrégation en Oracle ne supportent en entrée qu'un unique paramètre [Oracle 12c, 2013].

Afin d'utiliser plusieurs fonctions d'agrégation dans la même analyse, notre générateur de requêtes SQL peut générer des requêtes imbriquées. Par exemple, la requête SQL générée par notre prototype pour analyser les précipitations départementales mensuelles (selon l'agrégation simple) est la suivante :

```
SELECT  MOISN, DEPARTEMENT,
        SELECT_CENTER(LEV_DATA(NIVEAU_ADMINISTRATIF, PRECIP)) AS PRECIP
FROM( SELECT  DATES.MOISN, GEOGRAPHIE.DEPARTEMENT, GEOGRAPHIE.VILLE,
              GEOGRAPHIE.NIVEAU_ADMINISTRATIF,
              SUM(PRECIPITATION.PRECIP) AS PRECIP
      FROM DATES, GEOGRAPHIE, PRECIPITATION
      WHERE PRECIPITATION.ID_VILLE = GEOGRAPHIE.ID_VILLE
        AND PRECIPITATION.ID_DATE = DATES.ID_DATE
      GROUP BY  DATES.MOISN, DATES.LIBM, GEOGRAPHIE.DEPARTEMENT,
                GEOGRAPHIE.VILLE, GEOGRAPHIE.NIVEAU_ADMINISTRATIF)
GROUP BY MOISN, LIBM, DEPARTEMENT
```

6.3 ETUDES EXPÉRIMENTALES

Nous avons réalisé une série d'expériences. Nous présentons ces expériences en deux groupes : expériences sur le treillis d'optimisation et expériences sur la performance. Toutes les données utilisées dans nos études expérimentales sont des données de synthèse produites automatiquement par un générateur de données.

6.3.1 Etudes expérimentales sur le treillis d'optimisation

Ces études ont pour but d'étudier les conséquences de nos propositions sur la complexité du treillis.

Expérience 1.

Nous étudions les relations entre les priorités des fonctions d'agrégation (ordre d'exécution) d'une mesure et la complexité de son treillis d'optimisation.

Collection de données :

Nous utilisons trois schémas multidimensionnels avec quatre, cinq et six dimensions. Nous considérons que :

- Chaque dimension n'a qu'une seule hiérarchie avec cinq niveaux de granularité (paramètres), et
- Les fonctions d'agrégation utilisées sur une même dimension ont le même ordre d'exécution.

Protocole :

Tout d'abord, toutes les dimensions ont le même ordre d'exécution. Puis nous changeons l'ordre d'exécution des dimensions une par une jusqu'à ce que chaque dimension ait son propre ordre d'exécution. Nous suivons l'évolution du nombre d'arcs du treillis.

Résultats :

La Figure 72 montre que le nombre d'arcs du treillis diminue lorsque le nombre d'ordres d'exécution différents augmente. Nous remarquons que dans les trois cas (quatre dimensions (4D), cinq dimensions (5D) et six dimensions (6D)), cette diminution est linéaire. Ces résultats montrent que l'ordre d'exécution permet de réduire significativement le nombre d'arcs du treillis (jusqu'à environ 75% et 80% d'arcs sont supprimés dans les cas de cinq et six dimensions respectivement).

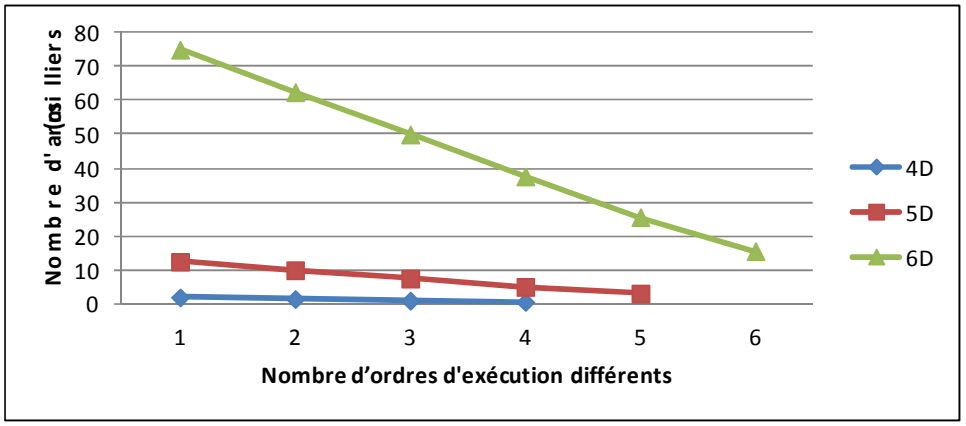


Figure 72 : Nombre d'arcs en fonction du nombre d'ordres d'exécution (Expérience 1)

Expérience 2.

Nous étudions les relations entre le nombre d'axes d'analyse (dimensions) et la complexité du treillis d'optimisation d'une mesure.

Collection de données :

Nous nous basons sur deux schémas multidimensionnels avec des dimensions de cinq paramètres. Dans le premier schéma, toutes les dimensions ont la même priorité, c'est-à-dire une valeur unique de l'ordre d'exécution pour toutes les fonctions d'agrégation. Le second schéma a des valeurs d'ordre d'exécution différentes pour chaque dimension.

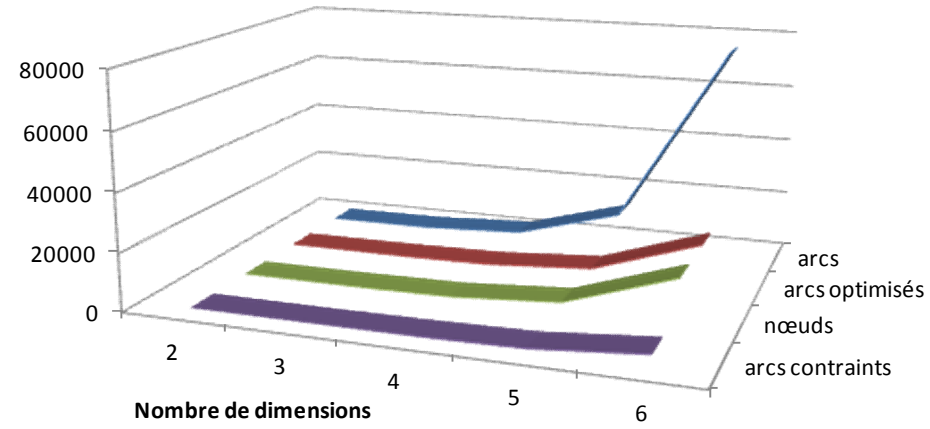
Protocole :

Nous observons le nombre de nœuds, d'arcs et d'arcs non transitifs en fonction du nombre de dimensions (de deux à six).

Résultats :

La Figure 73 montre que,

- **Si toutes les dimensions ont la même valeur d'ordre d'exécution**, le nombre d'arcs (la courbe d'*arcs*) augmente beaucoup plus rapidement que le nombre de nœuds (la courbe de *nœuds*).
- **Si chaque dimension a une valeur différente de l'ordre d'exécution**, l'augmentation du nombre d'arcs non transitifs (la courbe d'*arcs contraints*) est nettement inférieure à l'augmentation du nombre total des arcs (la courbe d'*arcs optimisés*). Dans le cas étudié, selon les valeurs affichées dans la table de données qui est au-dessous des courbes (Figure 73), le taux d'arcs contraints par rapport aux arcs totaux est environ 25%. Nous pouvons remarquer également que la courbe d'*arcs optimisés* est identique à la courbe de *nœuds* : selon la table des données, le nombre d'*arcs optimisés* = le nombre de *nœuds* -1. Ce qui indique qu'il n'y a qu'un arc pour chaque nœud. Ainsi, si chaque dimension possède un ordre d'exécution différent, le treillis d'optimisation n'est plus un graphe, mais un arbre (*cf.* § 4.4.7).



	2	3	4	5	6
arcs	40	300	2000	12500	75000
arcs optimisés	24	124	624	3124	15624
nœuds	25	125	625	3125	15625
arcs contraints	4	28	152	776	3900

Figure 73 : Nombre de nœuds et d'arcs en fonction du nombre de dimensions (Expérience 2)

Ces deux premières expériences montrent que la complexité du treillis peut être atténuée en réduisant significativement le nombre d'arcs grâce à l'ordre d'exécution.

Expérience 3.

Cette expérience étudie les effets des arcs non transitifs (contraints) sur le temps de la création du treillis.

Collection de données :

Afin de réaliser cette expérience, nous nous appuyons sur deux versions différentes du treillis de la Figure 74 (b). Ce treillis est défini à partir du schéma structurel et du schéma d'agrégation de la mesure 'Tem_Moy' de la Figure 74 (a). La première version est identique à la Figure 74 (b), où environ 42 % des arcs sont contraints. La seconde est différente de la première, car elle ne comprend aucun arc non transitif.

Pour éviter toute ambiguïté résultant de la différence entre les fonctions standard et les fonctions personnalisées, nous n'utilisons qu'une seule fonction d'agrégation (AVG) sur tout le treillis.

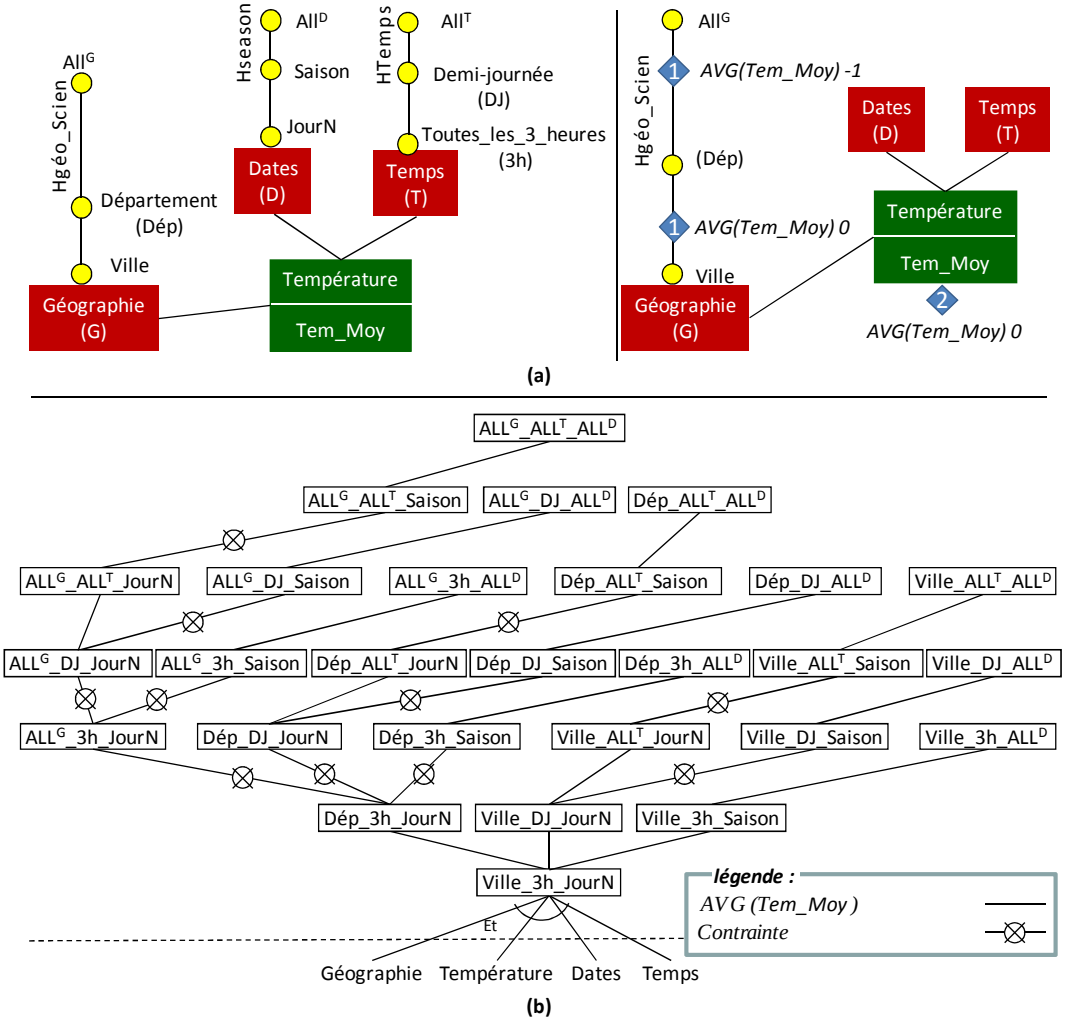


Figure 74 : Treillis d'optimisation multifonctions contrôlé (avec 42% des arcs contraints)

Protocole :

Nous observons le temps nécessaire pour créer le treillis entier en fonction du nombre de tuples du fait (de deux à huit millions).

Nous pouvons créer le treillis selon deux stratégies différentes :

[1] Tous les nœuds peuvent être calculés à partir du nœud de base,

[2] Chaque nœud peut être créé à partir d'un nœud directement inférieur. Comme la fonction considérée (AVG) est une fonction algébrique [Gray et al., 1996], il faut que chaque nœud (sauf la borne inférieure et la borne supérieure) stocke des valeurs intermédiaires (la somme et le nombre des occurrences) pour pouvoir calculer les nœuds supérieurs de manière transitive.

Résultats :

Les arcs contraints n'influencent pas le temps nécessaire pour calculer un nœud à partir d'un nœud directement inférieur. Par conséquence, les temps nécessaires pour construire les deux versions du treillis selon la deuxième stratégie sont identiques (la courbe *séquentiel* dans la Figure 75).

Cependant, les arcs contraints augmentent de manière significative le temps nécessaire pour la construction d'un nœud à partir du nœud de base. Ceci est clairement observable dans la Figure 75, où le temps supplémentaire nécessaire pour construire le treillis avec 42 % des arcs non transitifs (la courbe *avec blocage*) est jusqu'à environ 75% du temps nécessaire pour construire le treillis sans arcs non transitifs (la courbe *sans blocage*).

Ainsi, cette expérience montre que le temps de la création du treillis augmente significativement avec l'augmentation de nombre d'arcs non transitifs.

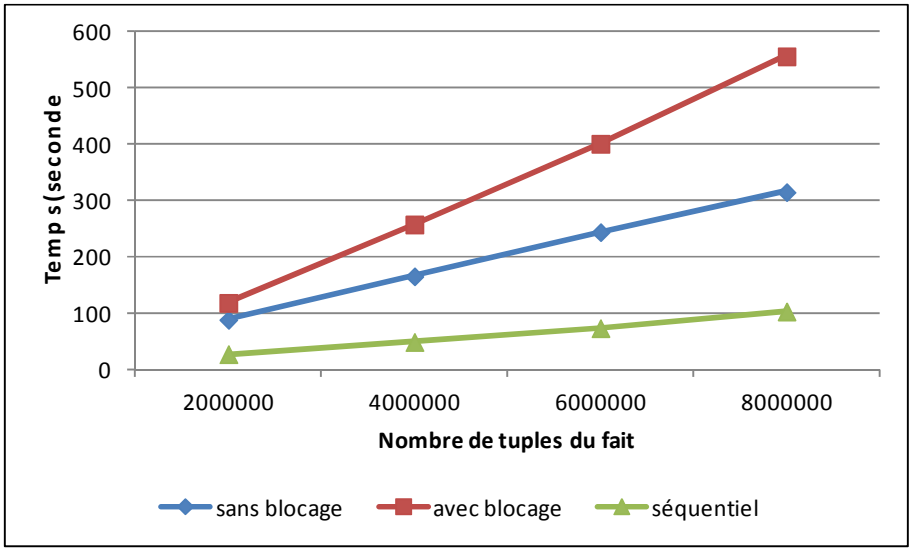


Figure 75 : Temps de la création du treillis selon le nombre de tuples du fait (Expérience 3)

6.3.2 Etudes des performances

Le générateur de requêtes SQL nous sert de plateforme expérimentale pour laquelle nous montrons une série d'expériences. Cette série d'expériences permet d'évaluer la performance de l'analyse multifonctions.

Expérience 4.

Cette expérience vise à étudier les conséquences de nos propositions (l'utilisation de plusieurs fonctions d'agrégation pour la même mesure) sur le temps d'exécution des requêtes d'interrogations et d'analyses OLAP. Plus précisément, nous étudions la relation entre le nombre de fonctions d'agrégation utilisées dans la même analyse et le temps d'exécution.

Collection de données :

Nous travaillons sur notre exemple de météo dont la taille de regroupement de données sur la dimension 'Géographie' est de valeur 5, c'est-à-dire, chaque instance d'un niveau supérieur correspond à cinq instances de niveau inférieur (par exemple, chaque région comprend cinq départements).

Protocole :

Nous observons les temps d'exécution (seconde) de trois requêtes, en fonction du nombre de tuples contenus dans le fait (de deux à dix millions) :

- La première requête agrège les températures moyennes au niveau 'Département'. Elle repose sur une seule fonction d'agrégation (comme dans le modèle classique) ;
- La deuxième requête agrège les températures moyennes au niveau 'Région'. Elle repose sur deux fonctions d'agrégation ;
- La troisième requête agrège les températures moyennes au niveau 'All$^{\text{Géographie}}$'. Elle nécessite d'appliquer trois fonctions d'agrégation.

Nous avons choisi ces trois requêtes pour présenter l'impact de l'utilisation de plusieurs fonctions d'agrégation (deuxième et troisième requêtes) par rapport au modèle classique qui utilise une seule fonction d'agrégation (première requête).

Afin d'éviter toute ambiguïté résultant de la différence entre les fonctions standard et les fonctions personnalisées, nous n'utilisons dans cette expérience qu'une seule fonction d'agrégation personnalisée (AVG_W).

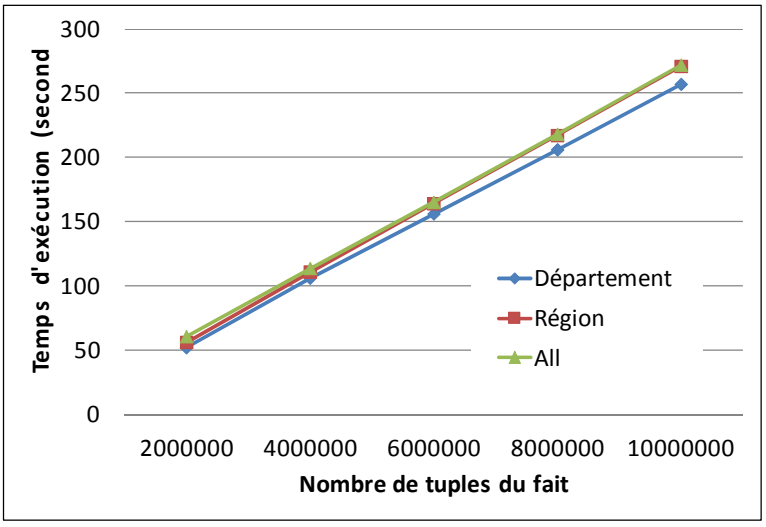


Figure 76 : Temps d'exécution selon le nombre de fonctions d'agrégation (Expérience 4)

Résultats :

La Figure 76 montre les temps d'exécution des trois requêtes. Ce temps augmente linéairement en fonction du nombre de tuples. La distance entre les courbes des première et deuxième requêtes (les courbes *Département* et *Région* respectivement) représente le temps supplémentaire demandé pour appliquer la deuxième fonction. Ce temps est environ de 6% du temps total pour exécuter la requête. Le temps supplémentaire pour appliquer la troisième fonction semble être non-remarquable (la courbe de la troisième requête *All* est presque au-dessus de la courbe de la deuxième requête *Région*). En réalité, ce phénomène est lié au volume des données qui décroît avec les fonctions précédemment appliquées. Ainsi lors du calcul de la

troisième fonction, le volume des données est proportionnellement fortement réduit par rapport au volume initialement impliqué.

Expérience 5.

Cette expérience étudie la relation entre le temps d'exécution, la taille de regroupement de données et le nombre de fonctions d'agrégation utilisées. Nous entendons par taille de regroupement de données le nombre de valeurs d'un paramètre inférieur qui sont regroupées en une valeur d'un paramètre supérieur.

Collection de données :

Nous travaillons sur deux versions différentes de notre exemple de météo. La première avec une taille de regroupement de données sur la dimension 'Géographie' de valeur 2, et la deuxième, avec une taille de regroupement de données de valeur 5.

Protocole :

Nous étudions le temps d'exécution (exprimé en seconde), en fonction du nombre de tuples du fait (de deux à dix millions), des quatre requêtes :

- deux requêtes au niveau 'Département' (une avec une taille de regroupement de données à 2 et l'autre à 5) qui utilisent une seule fonction d'agrégation, et
- deux requêtes supplémentaires au niveau 'All$^{\text{Géographie}}$' (avec une taille de regroupement de données à 2 et à 5) qui utilisent trois fonctions d'agrégation.

Comme dans l'expérience précédente, nous n'utilisons que la fonction d'agrégation personnalisée (AVG_W).

Résultats :

La Figure 77 montre les temps d'exécution des quatre requêtes. Les courbes correspondantes à une taille de regroupement de valeur 2 (ou 5) sont indiquées dans la légende par le chiffre deux (ou cinq) entre parenthèses.

Selon ces courbes, le temps d'exécution des requêtes avec une taille de regroupement à 5 est moindre que celui des requêtes avec une taille de regroupement à 2. Nous remarquons que le temps d'exécution des requêtes semble principalement influencé par la taille de regroupement. Ainsi, la requête avec une taille de regroupement à 2 et une seule fonction d'agrégation (la courbe *Département (2)*) est plus coûteuse en temps de calcul que la requête avec une taille de regroupement à 5 malgré trois fonctions d'agrégation (la courbe *All (5)*).

Nous pouvons conclure que la taille de regroupement paraît avoir un impact primordial sur le temps d'exécution, plus que le nombre de fonctions d'agrégation utilisées.

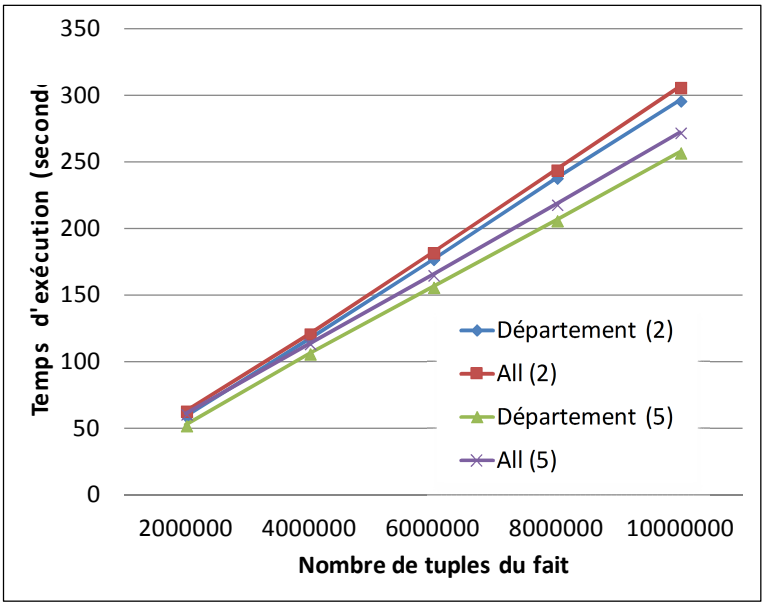


Figure 77 : Temps d'exécution selon le nombre de fonctions d'agrégation et la taille de regroupement de données (Expérience 5)

Expérience 6.

Cette dernière expérience concerne les opérateurs d'analyse OLAP. Nous examinons le temps d'exécution requis par nos opérateurs étendus par rapport aux opérateurs classiques. Nous étudions l'opérateur RollUp parce qu'il est parmi les opérateurs les plus utilisés.

Collection de données :

Cette expérience se base sur notre exemple de météo, où les températures sont enregistrées huit fois par jour (toutes les trois heures). La taille de regroupement de données sur la dimension 'Géographie' est de valeur 5.

Protocole :

Nous observons le temps d'exécution, en fonction de nombre de tuples du fait (de deux à huit millions), des cinq requêtes détaillées précédemment dans le chapitre 5 (*cf.* § 5.2.2.2.2) :

- **R4** : réalise l'analyse mensuelles des températures moyennes départementales dans un contexte uni-fonction ;
- **R5** : réalise une opération de **ROLLUP** pour analyser les températures moyennes régionales mensuelles et bénéficie des résultats de la requête précédente R4 dans un contexte uni-fonction (**ROLLUP** classique) ;
- **R6** : réalise l'analyse mensuelles des températures moyennes départementales dans un contexte multifonctions contrairement à R4 ;
- **R7** : réalise une opération de **ROLLUP** pour analyser les températures moyennes départementales annuelles et bénéficie des résultats de la requête précédente R6 dans un contexte multifonctions (**ROLLUP** étendu) ;
- **R8** : réalise une opération de **ROLLUP** pour analyser les températures moyennes régionales mensuelles, sans bénéficier des résultats de la requête R6 dans un contexte multifonctions (**ROLLUP** étendu) ;

La dernière requête R8 comprend une fonction d'agrégation personnalisée (AVG_W), ce qui affecte le temps d'exécution, car la fonction n'est pas optimisée contrairement aux fonctions

standard (SUM, AVG, COUNT, MAX, MIN, …). Par conséquent, nous étudions également une sixième requête (R9) identique à R8, mais avec la fonction standard (AVG) au lieu de (AVG_W).

La Figure 78 illustre les relations des calculs entre les six requêtes à étudier. Les flèches indiquent si une requête est calculée à partir du résultat d'une autre requête ou directement à partir de la BDM.

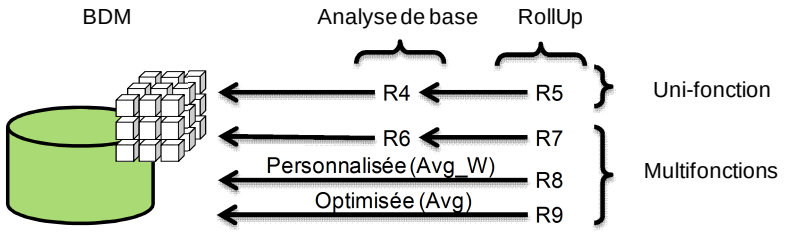


Figure 78 : Dépendances entre les requêtes à étudier

Résultats :

La Figure 79 montre les courbes correspondant aux six requêtes précédentes. Le temps nécessaire pour exécuter l'analyse de base dans un contexte multifonctions (la courbe *R6*) est supérieur à celui dans un contexte uni-fonction (la courbe *R4*) en raison de la complexité de la requête R4 (utilisation de plusieurs fonctions d'agrégation).

Nous pouvons remarquer que le temps requis pour exécuter les opérations de **ROLLUP** qui profitent des résultats des requêtes précédentes (les courbes *R5* et *R7*) est remarquablement faible (environ 0,1 seconde) parce que les données ont déjà été fortement agrégées. Ici, nous ne remarquons aucune différence entre l'opérateur classique (uni-fonction) et étendu (multifonctions).

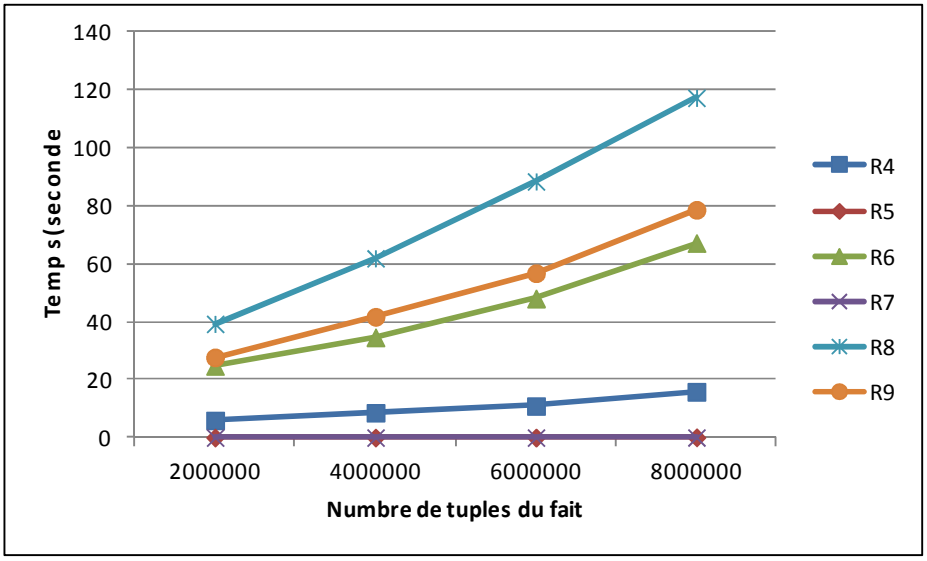


Figure 79 : Temps d'exécution de six requêtes selon le nombre de tuples du fait

Ces courbes montrent également que le temps nécessaire pour exécuter l'opération **ROLLUP** étendu (multifonctions) en l'absence de la possibilité de bénéficier des valeurs intermédiaires (la courbe *R8*) est largement supérieur au temps requis pour exécuter l'analyse de base dans un contexte multifonctions (la courbe *R6*). Cependant, une grande partie de ce temps

est due au fait que la fonction personnalisée AVG_W n'est pas optimisée, c'est ce que nous voyons clairement par la différence entre les courbes *R8* (**ROLLUP** étendu avec une fonction personnalisée) et *R9* (**ROLLUP** étendu avec une fonction optimisée).

Ces résultats montrent que le temps supplémentaire requis pour exécuter les opérateurs étendus (par rapport aux opérateurs classiques) est important. Dans le cas étudié au-dessus, ce temps est environ de 320% du temps d'exécution des opérateurs classiques. Il est donc nécessaire d'optimiser les requêtes et les fonctions d'agrégation utilisées et de bénéficier autant que possible des valeurs précédemment calculées.

Discussion

En regardant les trois expériences sur l'étude des performances (expérience 4, 5 et 6), nous remarquons l'augmentation notable des temps d'exécution des requêtes uni-fonction et multifonctions dans l'expérience 6 par rapport aux expériences 4 et 5. Cette augmentation est due au fait que les effets de regroupement de données sur le temps d'exécution est plus efficace dans le calcul uni-fonction que le calcul multifonction :

- Les données dans **l'expérience 4** sont agrégées aux niveaux :
  - 'All$^{\text{Géographie}}$' : avec une taille regroupement = 5 (entre les niveaux 'Ville' et 'Département') * 5 (entre les niveaux 'Département' et 'Région') * 5 (entre les niveaux 'Région' et 'All$^{\text{Géographie}}$') = 125,
  - 'JourN' et 'Toutes_les_3_heures' sans agrégation (parce qu'ils sont des niveaux de base) ;
- Les données dans **l'expérience 5** sont faiblement agrégées aux niveaux :
  - 'All$^{\text{Géographie}}$' : avec une taille regroupement = 2 (entre les niveaux 'Ville' et 'Département') * 2 (entre les niveaux 'Département' et 'Région') * 2 (entre les niveaux 'Région' et 'All$^{\text{Géographie}}$') = 8,
  - 'JourN' et 'Toutes_les_3_heures' sans agrégation ;
- Les données dans **l'expérience 6** sont fortement agrégées aux niveaux :
  - 'Département' avec une taille regroupement de valeur 5 (entre le niveau 'Ville' et 'Département'),
  - 'MoisN' avec une taille regroupement de valeur 30 (entre le niveau 'JourN' et 'MoisN'),
  - 'All$^{\text{Temps}}$' avec une taille regroupement de valeur 8 (entre le niveau 'Toutes_les_3_heures' et 'All$^{\text{Temps}}$').

Le Tableau 17 compare la différence entre les requêtes uni-fonction et multifonctions par rapport à la taille de regroupement de données dans les trois expériences. Cette différence est étudiée pour un nombre de tuples de huit millions dans le fait. Il convient d'indiquer que toutes les requêtes multifonctions étudiées dans ce tableau utilisent trois fonctions d'agrégation.

Dans ce tableau, nous constatons que les temps d'exécution des requêtes uni-fonction et multifonctions diminuent avec l'augmentation de taille de regroupement. Cette diminution est plus importante dans les requêtes uni-fonction que dans les requêtes multifonctions.

Par ailleurs, les fonctions d'agrégation utilisées dans les requêtes des expériences 4 et 5 sont des fonctions personnalisées contrairement aux fonctions standard utilisées dans les requêtes de l'expérience 6 considérées. C'est pourquoi les requêtes de l'expérience 6 sont beaucoup plus rapides que celles des expériences 4 et 5.

Tableau 17 : Temps d'exécution des requêtes uni-fonction et multifonctions dans les expériences 4, 5 et 6 pour huit millions tuples du fait

Expérience	Taille de regroupement totale	Temps d'exécution (Seconde)		Différence
		uni-fonction	multifonction	
5	2*2*2 = 8	238	244	6 secondes (2.5%)
4	5*5*5 = 125	206	218	12 secondes (5.8%)
6	5*30*8 = 1200	16	67	51 secondes (319%)

6.4 CONCLUSION

Dans ce chapitre, nous avons présenté notre prototype *OLAP-Multi-Functions* dont l'objectif est de montrer la faisabilité de notre approche d'analyse multifonctions. Il repose sur un ensemble de modules [Hassan et al., 2013a] :

- Le module **Constructeur** définit les fonctions d'agrégation afin de les intégrer dans le schéma multidimensionnel ;
- Le module **Visualisation** montre le schéma structurel et les schémas d'agrégation ;
- Le module **Analyseur** facilite l'interrogation des données effectuées graphiquement en cachant la complexité induite par l'agrégation et les structures d'implantation ROLAP sous-jacentes de la BDM.

Notre prototype comprend trois espaces de stockage [Hassan et al., 2013a] :

- Une **Base relationnelle *R-OLAP*** stockant les données des faits et des dimensions ;
- Une **Méta-base** décrivant les structures du schéma multidimensionnel (faits, dimensions et hiérarchies) ainsi que les fonctions d'agrégation, l'ordre d'exécution et les contraintes d'agrégation ;
- Un **Générateur de requêtes SQL** qui supervise les analyses et traduit les interactions d'analyste en des scripts SQL exécutables dans le contexte d'implantation R-OLAP.

Nous avons mené une série d'expériences organisées en deux groupes :

Etudes expérimentales sur le treillis d'optimisation : ces expériences montrent que l'augmentation de la complexité du treillis induite par l'augmentation du nombre de nœuds (*cf.* § 4.4.1) peut être atténuée en réduisant le nombre d'arcs grâce à l'ordre d'exécution ; cette réduction s'effectue jusqu'à ce que le nombre d'arcs devienne dans le meilleur cas égal au nombre de nœuds [Hassan et al., 2014].

Ces expériences démontrent également la nécessité d'utiliser des pré-agrégats notamment dans le cas d'un grand nombre d'arcs contraints dans le treillis d'optimisation [Hassan et al., 2014].

Etudes des performances : nous avons observé que le temps supplémentaire pour appliquer plusieurs fonctions d'agrégation dans la même analyse est acceptable grâce à la réduction du volume des données avec l'application de chaque fonction d'agrégation [Hassan et al., 2013a].

Les résultats des études de la relation entre le temps d'exécution, la taille de regroupement de données et le nombre de fonctions d'agrégation utilisées montre que la taille de regroupement de données a des impacts sur le temps d'exécution plus importants que le nombre de fonctions d'agrégation utilisées [Hassan et al., 2013a]. Ces impacts sont moins efficaces dans notre contexte multifonctions que dans le contexte uni-fonction. Nos expériences montrent l'importance d'optimiser les requêtes d'analyse en exploitant autant que possible des valeurs précédemment calculées [Hassan et al., 2013b].

7. CHAPITRE VI : CONCLUSION ET PERSPECTIVES

7.1 CONCLUSION GÉNÉRALE

Les travaux de recherche présentés dans ce mémoire de thèse se situent dans le cadre des systèmes d'aide à la décision. Ces systèmes reposent sur un processus d'analyse en ligne (OLAP) facilitant l'analyse interactive et la synthèse des données [Codd et al., 1993]. Les systèmes d'aide à la décision adoptent généralement la modélisation multidimensionnelle adaptée aux analyses [Kimball, 1996]. Jusqu'à présent, peu de travaux considèrent la modélisation de l'agrégation des données dans l'espace multidimensionnel. La plupart des propositions existantes considèrent généralement une même fonction d'agrégation pour déterminer les valeurs d'une mesure aux différents niveaux de granularité de l'espace multidimensionnel. Ces travaux supposent qu'il est toujours possible de calculer l'agrégation d'une mesure directement à partir des niveaux de base.

Dans cette thèse, notre premier objectif a été de pallier ces limites en proposant un nouveau **modèle conceptuel multidimensionnel multifonctions**. Ensuite, nous avons étudié l'impact de notre proposition au **niveau logique** et sur l'ensemble d'**opérateurs d'analyse OLAP**. La **validation** de nos propositions a été effectuée par le développement d'un prototype permettant l'analyse multidimensionnelle en utilisant plusieurs fonctions d'agrégation pour la même mesure.

Modèle conceptuel multidimensionnel multifonctions. Notre modèle est indépendant des contraintes de plateformes d'implantation logiques ou physiques. Il permet au concepteur d'intégrer les fonctions d'agrégation en associant à chaque mesure un ensemble de fonctions compatibles. Notre modèle permet de faire évoluer les fonctions d'agrégation suivant les axes d'analyse (fonction multiple dimensionnelle), les hiérarchies (fonction multiple hiérarchique) ou les niveaux de granularité (fonction différenciée). La spécification des fonctions d'agrégation au niveau conceptuel a pour but de contrôler l'agrégation des données (ce qui minimise le risque de production d'erreurs de calcul) et d'utiliser ces fonctions pour le calcul des pré-agrégats.

Notre modèle est suffisamment expressif pour permettre au concepteur de contrôler la validité des combinaisons de fonctions d'agrégation en associant à chaque fonction un ordre d'exécution. Grâce à cet ordre d'exécution, nous pouvons à la fois contrôler l'exécution des fonctions non-commutatives et permettre d'exécuter les fonctions commutatives dans n'importe quel ordre. Des contraintes d'agrégation sont associées aux fonctions d'agrégation afin d'indiquer un niveau de granularité spécifique à partir duquel l'agrégation doit être calculée.

L'originalité de notre modèle est la présentation des formalismes graphiques en deux niveaux afin d'améliorer la lisibilité du schéma de la BDM :

- Un schéma structurel comprend les éléments structurels (faits, dimensions et hiérarchies)
- Un schéma d'agrégation différent pour chaque mesure comprend les différents mécanismes d'agrégation (fonctions d'agrégation, ordre d'exécution et contraintes d'agrégation)

Niveau logique. Nous avons étudié les conséquences de la prédéfinition de plusieurs fonctions d'agrégation pour la même mesure sur le stockage de données de base (faits et dimensions) et de données d'optimisation (vues matérialisées). Nous avons proposé d'implanter les schémas multidimensionnels en utilisant l'approche R-OLAP [Kimball, 1996] qui transforme les faits et les dimensions au niveau logique en tables relationnelles. Cette transformation n'est pas influencée par la pluralité des fonctions d'agrégation, ce qui permet d'appliquer notre modèle multifonction sur les systèmes décisionnels sans modifier ces données.

Par contre, notre modèle multifonctions a plusieurs impacts sur les données d'optimisation modélisées par un treillis [Harinarayan et al., 1996], où les nœuds représentent les pré-agrégats et les arcs représentent les chemins des calculs des agrégations. Ces impacts rendent les treillis d'optimisation différents d'une mesure à l'autre même si les mesures concernées sont analysées selon les mêmes dimensions. Les impacts peuvent toucher les données stockées dans les nœuds ou la structure de treillis (la forme et des arcs du treillis peuvent être modifiés). D'autres impacts (comme le typage des arcs et le blocage de la transitivité du calcul) rendent l'estimation du coût du calcul plus complexe. Par ailleurs, l'augmentation de nombre de nœuds peut engendrer une augmentation de la complexité du treillis. Cette dernière est cependant atténuée par la réduction du nombre d'arcs suite à l'élagage du treillis.

A notre connaissance, l'exploitation de plusieurs fonctions d'agrégation pour la même mesure dans le treillis d'optimisation n'a fait l'objet d'aucun travail de recherche par ailleurs.

Opérateurs d'analyse OLAP. Dans notre contexte multifonctions, nous avons proposé un langage d'interrogation des données adaptés à notre modèle. Ce langage se base sur la structure de visualisation de la table multidimensionnelle (TM) et l'ensemble des opérateurs de manipulation des bases de données multidimensionnelles OLAP [Ravat et al., 2008]. Notamment, nous avons adapté la définition et la visualisation de la TM au contexte multifonctions. Nous avons étendu également les opérateurs d'analyse OLAP afin de prendre en compte l'utilisation de plusieurs fonctions d'agrégation pour la même mesure. Ces extensions peuvent concerner la spécification de plusieurs opérateurs en raison de l'automatisation du choix des fonctions d'agrégation et de la différence entre les fonctions utilisées sur les hiérarchies de la même dimension. En outre, à cause de l'existence d'un ordre d'exécution entre les fonctions d'agrégation, ces extensions peuvent toucher également le mécanisme interne des opérateurs qui changent un niveau d'agrégation par un autre niveau supérieur.

Validation. Afin de valider nos contributions, nous avons développé le prototype *OLAP-Multi-Functions*. Ce prototype permet d'intégrer graphiquement les fonctions d'agrégation dans le schéma multidimensionnel ainsi que de superviser les manipulations OLAP. Le prototype consiste à faciliter l'interrogation graphique des données en cachant les complexités de l'agrégation et la structure logique de la BDM. Notre prototype se base sur un méta-schéma décrivant les structures du schéma multidimensionnel (faits, dimensions et hiérarchies) ainsi que les fonctions d'agrégation, l'ordre d'exécution et les contraintes d'agrégation utilisés pour construire des requêtes SQL valides et cohérentes. Ces requêtes sont générées automatiquement par un générateur de requêtes SQL. Ce générateur traduit les interactions d'analystes en scripts SQL exécutables dans le contexte d'implantation R-OLAP.

Notre prototype sert de plateforme expérimentale au travers de laquelle nous avons mené une série d'expériences dont les résultats montrent que la taille de regroupement de données influence le temps d'exécution de manière plus prépondérante que le nombre de fonctions d'agrégation utilisées.

7.2 PERSPECTIVES

A **court terme** nous prévoyons de poursuivre nos travaux en développant plusieurs extensions.

Environnement d'intégration de fonctions d'agrégation. Nous avons proposé des fonctions d'agrégation personnalisées (AVG_W et SELECT_CENTER) adaptées aux calculs spécifiques (la moyenne pondérée et la sélection de ville représentative). D'autres fonctions pourraient être utilisées. Nous envisageons de proposer un environnement d'intégration pour des nouvelles fonctions. Cet environnement permettrait au concepteur d'ajouter des fonctions, adaptées à ces besoins spécifiques d'analyse.

Une extension de notre langage d'interrogation des données multidimensionnelles. Nous avons proposé un langage d'interrogation de données multidimensionnelles. Ce langage comprend un ensemble d'opérateurs d'analyse OLAP qui permet de construire et de modifier une table multidimensionnelle (TM). Une extension possible de ces opérateurs consiste à proposer des opérateurs binaires permettant de comparer et fusionner deux TMs (union, intersection, …) afin de faciliter la corrélation entre les analyses.

En outre, la différence entre les mesures en termes de fonctions d'agrégations (où leur ordre d'exécution où leurs contraintes d'agrégation) entraine des difficultés à agréger plusieurs mesures à la fois. Cela peut nécessiter de chercher un chemin du calcul partagé entre les mesures agrégées ou bien d'agréger chaque mesure seule pour fusionner les résultats intermédiaires. Cela induit des changements au niveau des mécanismes internes des opérateurs d'analyse. Pour répondre à ce point, nous envisageons d'optimiser notre langage d'interrogation afin de l'adapter à cette analyse multi-mesures multifonctions.

La matérialisation d'une partie des vues. Cette matérialisation évite au système de coûteux temps de calculs. D'un côté, nous avons étudié les impacts de notre modèle multifonctions sur le treillis de vues à matérialiser. D'un autre côté, nos expériences ont démontré que notre contexte multifonctions a besoin de l'optimisation plus que le contexte uni-fonction. Donc, nous envisageons de poursuivre nos travaux :

- En revisitant les algorithmes de calcul des pré-agrégats en les adaptant à notre modélisation multifonctions ;
- En étudiant les effets des changements dans le treillis contrôlé (changements de forme, d'arcs et des données) lors de la sélection de nœuds pour améliorer les performances ;
- En définissant une fonction de coût (en termes d'espace occupé, d'économie et de temps de calcul) pour l'implantation et la maintenance d'une BDM multifonctions optimisée.

D'autres perspectives à plus **long terme** sont envisageables.

Gestion des versions. L'évolution d'un schéma multidimensionnel représente une perspective liée à notre problématique. Les besoins des décideurs peuvent évoluer dans le temps (ajout, suppression ou modification des éléments du schéma multidimensionnel). Des travaux au sein de notre équipe proposent de prendre en compte ces changements au travers de la gestion des versions du schéma multidimensionnel [Ravat et al., 2006], [Ravat & Teste, 2006]. Plus précisément, la gestion de l'historique des mesures et des paramètres voire la gestion de l'historique des faits et des dimensions [Bellahsène, 2002]. Certes, l'étude de l'évolution des

schémas multidimensionnels a fait l'objet de plusieurs travaux comme présentés dans [Golfarelli & Rizzi, 2009]. Cependant, l'évolution dans ces travaux ne considère que les éléments structurel (faits, dimensions, hiérarchies). Nous envisageons la gestion des versions de schéma multidimensionnel qui permettront de faire face aux évolutions des mécanismes d'agrégation (fonctions, ordre d'exécution et contraintes d'agrégation). Nous envisageons également de développer un langage d'interrogation adaptée qui permet d'interroger à la fois plusieurs versions du schéma.

Mégadonnées « Big Data ». Nous envisageons de généraliser notre approche en proposant de nouvelles techniques d'analyse décisionnelle pour les mégadonnées « Big Data ». Les techniques actuelles de traitement, de stockage et d'analyse de données doivent être reconsidérées et étendues, voire redéfinies compte tenu de la règle dite « des 3V » (volume, vitesse et variété)[23] de ce domaine d'étude :

- **Volume** : Les mégadonnées sont généralement associées à cette caractéristique. Les volumes massifs sont ainsi supportés par une organisation massivement distribuées des espaces de stockage et des traitements appliqués aux données ainsi distribuées ;
- **Vitesse** : des flux croissants de données doivent être analysés en temps réel pour une utilisation immédiate ou différée ;
- **Variété :** Il ne s'agit pas de données relationnelles traditionnelles, ces données sont brutes, semi-structurées voire non structurées. Ce sont des données complexes provenant du web, au format texte et images. Ce qui les rend difficilement utilisables avec les outils traditionnels.

Nous souhaitons ainsi investir ces nouveaux paradigmes afin d'implanter nos propositions dans des contextes non plus exclusivement SQL, mais « Not Only SQL ».

[23] http://www.journaldunet.com/solutions/expert/51696/les-3-v-du-big-data---volume--vitesse-et-variete.shtml

8. BIBLIOGRAPHIE

[Abelló, 2002] Alberto Abelló, "*YAM^2: a multidimensional conceptual model*", thèse de doctorat, Université Polytechnique de Catalogne (Espagne), avril 2002.

[Abelló et al., 2001a] Alberto Abelló, José Samos, Fèlix Saltor, "Understanding Facts in a Multidimensional Object-Oriented Model", *4th ACM Intl. Workshop on Data Warehousing and OLAP (DOLAP)*, ACM Press, p. 32-39, 2001.

[Abelló et al., 2001b] Alberto Abelló, José Samos, Fèlix Saltor, "Understanding Analysis Dimensions in a Multidimensional Object-Oriented Model", 3rd Intl. Workshop on Design and Management of Data Warehouses (DMDW), CEUR Workshop proceedings Vol. (39), CEUR-WS.org, p. 41-49, 2001.

[Abelló et al., 2002] Alberto Abelló, José Samos, Fèlix Saltor, "YAM^2 (Yet Another Multidimensional Model): An Extension of UML", *Intl. Symposium on Database Engineering and Applications (IDEAS)*, Washington (USA), p. 172-181, 2002.

[Abelló et al., 2003] Alberto Abelló, José Samos, Fèlix Saltor, "Implementing operations to navigate semantic star schemas", *6th ACM Intl. Workshop on Data Warehousing and OLAP (DOLAP)*, ACM Press, p. 56-62, 2003.

[Abelló et al., 2006] Alberto Abelló, José Samos, Fèlix Saltor, "YAM^2: a multidimensional conceptual model extending UML", *Information Systems (IS)*, Vol. (31), N. (6), Elsevier, p. 541-567, septembre 2006.

[Agrawal et al., 1995] Rakesh Agrawal, Ashish Gupta, Sunita Sarawagi, "*Modeling Multidimensional Databases*", IBM Research Report, http://rakesh.agrawal-family.com/papers/icde97olap_rj.pdf, 1995.

[Agrawal et al., 1997] Rakesh Agrawal, Ashish Gupta, Sunita Sarawagi, "Modeling Multidimensional Databases", *13th Intl. Conf. on Data Engineering (ICDE'97)*, IEEE Computer Society, Birmingham (U.K.), p. 232-243, 1997.

[Agrawal et al., 2000] Sanjay Agrawal, Surajit Chaudhuri, Vivek R. Narasayya, "Automated selection of materialized views and indexes in sql databases" *26th Intl. Conf. on Very Large Data Bases (VLDB)*, Le Caire (Egypte), p. 496-505, 2000.

[Annoni et al., 2006] Estella Annoni, Franck Ravat, Olivier Teste, Gilles Zurfluh, "Towards Multidimensional Requirement Design", *8th Intl. Conf. on Data Warehousing and Knowledge Discovery (DaWaK)*, LNCS 4081, Springer, p. 75-84, 2006.

[Baralis et al., 1997] Elena Baralis, Stefano Paraboschi, Ernest Teniente, "Materialized View Selection in a Multidimensional Database", *23rd Intl. Conf. on Very Large Data Bases (VLDB)*, Morgan Kaufmann, Athens (Greece), p. 156-165, août 1997.

[Baril, 1999] X. Baril, "*Historisation dans les entrepôts de données*", mémoire de DEA 2IL (Informatique de l'Image et du Langage), Université Paul Sabatier Toulouse 3 (France), juin 1999.

[Bellahsène, 2002] Z. Bellahsene, "Schema Evolution in Data Warehouses". *Knowledge and Information Systems*, Springer-Verlag London Ltd, Vol. (4), N. (3), p. 283-304, juillet 2002.

[Bertinoa et al., 2003] Elisa Bertinoa, Elena Ferrarib, Giovanna Guerrinic, Isabella Merlo, "T-ODMG: an ODMG compliant temporal object model supporting multiple granularity management", *Journal Information Systems*, Vol. (28), N. (8), p. 885-927, December 2003.

[Bertinoa et al., 2009] Elisa Bertino, Elena Camossi, Michela Bertolotto, "Multi-granular Spatio-temporal Object Models: Concepts and Research Directions", *2nd Intl. Conf. On Objects and DataBases (ICOODB)*, LNCS 5936, Springer-Verlag Berlin Heidelberg 2010, Zurich (Suisse), p. 132-148, juillet 2009.

[Bimonte et al., 2012] Sandro Bimonte, Michela Bertolotto, Jérôme Gensel, Omar Boussaid, "Spatial OLAP and Map Generalization Model and Algebra", *Intl. Journal of Data Warehousing and Mining (ijDWM)*, Vol. (8), N. (1), p. 24-51, 2012.

[BO XI 3.1 SP6, 2013] "SAP BusinessObjects Desktop Intelligence Access and Analysis Guide", SAP BusinessObjects XI 3.1 Service Pack 6 (17/05/2013) http://help.sap.com/

[BO XI 3.1 SP3, 2010] "Universe Designer", SAP BusinessObjects XI 3.1 Service Pack 3 (07/05/2010) http://help.sap.com/

[Börzsönyi et al., 2001] Stephan Börzsönyi, Donald Kossmann, Konrad Stocker, "The Skyline Operator", *17th Intl. Conf. on Data Engineering (ICDE)*, IEEE Computer Society, p. 421-430, 2001.

[Boukorca et al., 2013] Ahcène Boukorca, Ladjel Bellatreche, Sid-Ahmed Benali Senouci, Zoé Faget, "Votre Plan d'Exécution de Requêtes est un Circuit Intégré : Changer de Métier", *9èmes journées francophones sur les Entrepôts de Données et l'Analyse en ligne (EDA)*, Blois (France), p. 133-148, 2013.

[Boukraa et al., 2010] Doulkifli Boukraa, Omar Boussaid, Fadila Bentayeb, "OLAP Operators for Complex Object Data Cubes", *14th East-European Conf. on Advances in Databases and Information Systems (ADBIS)*, LNCS 6295, Springer, p. 103-116, 2010.

[Boulil et al., 2011] Kamal Boulil, Sandro Bimonte, François Pinet, "Un modèle UML et des contraintes OCL pour les entrepôts de données spatiales. De la représentation conceptuelle à l'implémentation", *Journal Ingénierie des Systèmes d'Information (ISI)*, Vol. (16), N. (6), p. 11-39, 2011.

[Bruckner et al., 2001] Robert M. Bruckner, Beate List, Josef Schiefer, A. Min Tjoa, "Modeling Temporal Consistency in Data Warehouses", *1st Intl. Workshop on Knowledge Extraction for Enterprise Services (KEES), 12th Intl. Workshop on Database and Expert Systems Applications (DEXA Workshop)*, IEEE Computer Society, p. 901-905, 2001.

[Buzydlowski et al., 1998] J.W. Buzydlowski, I.Y. Song, L. Hassell, "A Framework for Object-Oriented On-line Analytical Processing", *1st Intl. Workshop on Data Warehousing and OLAP (DOLAP)*, Bethesda (Maryland, USA), ACM, p. 10-15, novembre 1998.

[Cabibbo & Torlone, 1997] Luca Cabibbo, Riccardo Torlone, "Querying Multidimensional Databases", *6th Intl. Workshop Database Programming Languages (DBPL)*, LNCS 1369, Springer, p. 319-335, 1997.

[Cabibbo & Torlone, 1998] Luca Cabibbo, Riccardo Torlone, "From a Procedural to a Visual Query Language for OLAP", *10th Intl. Conf. on Scientific and Statistical Database Management (SSDBM)*, IEEE Computer Society, p. 74-83, 1998.

[Cabibbo & Torlone, 2000] Luca Cabibbo, Riccardo Torlone, "The Design and Development of a Logical System for OLAP", *2nd Intl. Conf. on Data Warehousing and Knowledge Discovery (DaWaK)*, LNCS 1874, Springer, p. 1-10, 2000.

[Cameron, 2009] Scott Cameron, Hitachi Consulting, "*Microsoft SQL Server 2008 Analysis Services Step by Step*", Microsoft Press, ISBN : 978-0735626201, avril 2009.

[Camossi et al., 2006] Elena Camossi, Michela Bertolotto, Elisa Bertino, "A multigranular object-oriented framework supporting spatio-temporal granularity conversions", *Intl. Journal of Geographical Information Science (IJGIS)*, Vol. (20), N. (5), p. 511-534, mai 2006.

[Camossi et al., 2008] Elena Camossi, Michela Bertolotto, Elisa Bertino, "Querying Multigranular Spatio-temporal Objects", *19th Intl. Conf. Database and Expert Systems Applications (DEXA)*, LNCS 5181, Springer-Verlag Berlin Heidelberg, Turin (Italy), p. 390-403, septembre 2008.

[Camossi et al., 2009a] Elena Camossi, Elisa Bertino, Giovanna Guerrini, Michela Bertolotto, "Adaptive Management of Multigranular Spatio-Temporal Object Attributes", *11th Intl. Symposium on Spatial and Temporal Databases (SSTD)*, LNCS 5644, Springer-Verlag Berlin Heidelberg, Aalborg (Denmark), p. 320-337, juillet 2009.

[Camossi et al., 2009b] Elena Camossi, Michela Bertolotto, Elisa Bertino, "*Spatio-temporal Multi-granularity: Modelling and Implementation Challenges*", rapport technique, University College Dublin, School of Computer Science and Informatics (UCD-CSI), août 2009.

[Chaudhuri & Dayal, 1997] Surajit Chaudhuri, Umeshwar Dayal, "An Overview of Data Warehousing and OLAP Technology", *Intl. Conf. on Management of Data (SIGMOD)*, Vol. (26), N. (1), ACM Press, p. 65-74, mars 1997.

[Chaudhuri & Shim, 1994] Surajit Chaudhuri, Kyuseok Shim, "Including Group-By in Query Optimization", *20th Intl. Conf. on Very Large Data Bases (VLDB)*, Morgan Kaufmann, p. 354-366, 1994.

[Chen et al., 2006] Lei Chen, Raghu Ramakrishnan, Paul Barford, Bee-Chung Chen, Vinod Yegneswaran, "Composite Subset Measures", *32nd Intl. conf. on Very large data bases (VLDB)*, Seoul (Korea), p. 403-414, Septembre 2006.

[Choong et al., 2003] Y. W. Choong, D. Laurent, P. Marcel, "Computing appropriate representations for multidimensional data", *Data & Knowledge Engineering Journal*, Vol. (45), N. (2), p. 181-203, mai 2003.

[Christy et al., 2006] John R. Christy, William B. Norris, Kelly Redmond, Kevin P. Gallo, "Methodology and Results of Calculating Central California Surface Temperature Trends: Evidence of Human-Induced Climate Change?", *Journal of climate*, Vol. (19), p. 548-563, 15 février 2006.

[Clark et al., 2007] Thomas D. Clark, Mary C. Jones, Curtis P. Armstrong, "The dynamic structure of management support systems: theory development, research focus, and direction", *Intl. Journal MIS Quarterly*, Vol. (31), N. (3), p. 579-615, septembre 2007.

[Codd et al., 1993] E.F. Codd, S.B. Codd, C.T. Salley, "*Providing OLAP (On Line Analytical Processing) to user analyst: an IT mandate*", rapport technique, E.F. Codd and associates, (white paper de Hyperion Solutions Corporation), 1993.

[Cuzzocrea & Mansmann, 2009] Alfredo Cuzzocrea, Svetlana Mansmann, "OLAP Visualization: Models, Issues, and Techniques", *Encyclopedia of data warehousing and mining*, 2nd Edition, p. 1439-1446, 2009.

[Cuzzocrea et al., 2007] Alfredo Cuzzocrea, Domenico Saccà, Paolo Serafino, "Semantics-aware Advanced OLAP Visualization of Multidimensional Data Cubes", *Intl. Journal of Data Warehousing and Mining (ijDWM)*, IGI Global, D. Taniar, Vol. (3), N. (4), p. 1-30, janvier 2007.

[Datta & Thomas, 1999] Anindya Datta, Helen Thomas, "The cube data model: a conceptual model and algebra for on-line analytical processing in data warehouses", *Decision Support Systems (DSS)*, Vol. (27), N. (3), Elsevier, p. 289-301, décembre 1999.

[Dinter et al., 1998] B. Dinter, C. Sapia, G. Höfling, M. Blaschka, "The OLAP Market: State of the Art and Research Issues", *1st Intl. Workshop on Data Warehousing and OLAP (DOLAP)*, Bethesda (Maryland, USA), ACM, p. 22-27, novembre 1998.

[Franconi & Kamble, 2004] Enrico Franconi, Anand Kamble, "The GMD Data Model and Algebra for Multidimensional Information", *16th Intl. Conf. on Advanced Information Systems Engineering (CAiSE)*, LNCS 3084, Springer, p. 446-462, 2004.

[Ghozzi, 2004] Faiza Ghozzi, "*Conception et Manipulation de Bases de Données Dimensionnelles à Contraintes*", thèse de doctorat, Université Paul Sabatier Toulouse 3 (France), novembre 2004.

[Giraudin et al., 2001] Jean-Pierre Giraudin, Monique Chabre-Peccoud, Dominique Rieu, Cristophe Saint-Marcel, "Informatique et Systèmes d'Information, Information - Commande - Communication" Chapitre 3, *Modèles de spécification pour l'ingénierie des systèmes d'information*, HERMES Science Publications, ISBN : 2-7462-0219-0, p. 61-92, 2001.

[Golfarelli & Rizzi, 2009] Matteo Golfarelli, Stefano Rizzi, "A Survey on Temporal Data Warehousing", *Intl. Journal of Data Warehousing and Mining (IJDWM)*, Vol. (5), N. (1), p. 1-17, 2009.

[Golfarelli & Rizzi, 2013] Matteo Golfarelli, Stefano Rizzi, "Honey, I Shrunk the Cube B", *17th East-European Conf. on Advances in Databases and Information Systems (ADBIS)*, LNCS 8133, Springer, Gênes (Italie), p. 176-189, 2013.

[Golfarelli et al., 1998a] Matteo Golfarelli, Dario Maio, Stefano Rizzi, "The Dimensional Fact Model: A Conceptual Model for Data Warehouses", *Intl. Journal of Cooperative Information Systems (IJCIS)*, Vol. (7), N. (2-3), World Scientific Publishing, p. 215-247, juin & septembre 1998.

[Golfarelli et al., 1998b] Matteo Golfarelli, Dario Maio, Stefano Rizzi, "Conceptual Design of Data Warehouses from E/R Schemes", *31st Annual Hawaii Intl. Conf. on System Sciences (HICSS'98)*, Vol. (7), p. 334-343, 6-9 janvier 1998.

[Golfarelli et al., 2002] Matteo Golfarelli, Stefano Rizzi, Ettore Saltarelli, "WAND: A CASE Tool for Workload-Based Design of a Data Mart", *Decimo Convegno Nazionale su Sistemi Evoluti per Basi di Dati (SEBD)*, p. 422-426, 2002.

[Gray et al., 1996] Jim Gray, Adam Bosworth, Andrew Layman, Hamid Pirahesh, "Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Total", *12th Intl. Conf. on Data Engineering (ICDE)*, IEEE Computer Society, p. 152-159, 1996.

[Gupta, 1997] Himanshu Gupta, "Selection of Views to Materialize in a Data Warehouse", *6th Intl. Conf. Delphi, Greece (ICDT)*, p. 98-112, 8-10 janvier, 1997.

[Gupta & Mumick, 1999] Himanshu Gupta, Inderpal Singh Mumick, "Selection of Views to Materialize under a Maintenance-Time constraint", *7th Intl. Conf. on Database Theory (ICDT)*, p. 453-470, 1999.

[Gupta et al., 1995] Ashish Gupta, Venky Harinarayan, Dallan Quass, "Aggregate-Query Processing in Data Warehousing Environments", *21st Intl. Conf. on Very Large Data Bases (VLDB)*, Morgan Kaufmann, p. 358-369, 1995.

[Gyssens & Lakshmanan, 1997] Marc Gyssens, Laks V. S. Lakshmanan, "A Foundation for Multi-dimensional Databases", *23rd Intl. Conf. on Very Large Data Bases (VLDB)*, Morgan Kaufmann, Athens (Greece), p. 106-115, 25-29 août 1997.

[Gyssens et al., 1996] Marc Gyssens, Laks V. S. Lakshmanan, Iyer N. Subramanian, "Tables as a Paradigm for Querying and Restructuring", *15th SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'96)*, ACM Press, Montreal (Canada), p. 93-103, 1996.

[Hahn et al., 2000] Karl Hahn, Carsten Sapia, Markus Blaschka, "Automatically Generating OLAP Schemata from Conceptual Graphical Models", *3rd ACM Intl. Workshop on Data Warehousing and OLAP (DOLAP)*, ACM Press, p. 9-16, 2000.

[Han et al., 1998] Jawei Han, Nebojsa Stefanovic, Krzysztof Koperski, "Selective Materialization: An Efficient Method for Spatial Data Cube Construction", *2nd Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD)*, p. 144-158, 1998.

[Hanrahan et al., 2007] P. Hanrahan, C. Stolte, J. Mackinlay, "*Visual Analysis for Everyone: Understanding Data Exploration and Visualization*", (White Paper de Tableau Software Inc), 2007.

[Hanusse et al., 2011] Nicolas Hanusse, Sofian Maabout, Radu Tofan, "Revisiting the Partial Data Cube Materialization", *15th East-European Conf. on Advances in Databases and Information Systems (ADBIS)*, LNCS 6909, Springer, p. 70-83, 2011.

[Harinarayan et al., 1996] Venky Harinarayan, Anand Rajaraman, Jeffrey D. Ullman, "Implementing Data Cubes Efficiently", *Intl. Conf. on Management of Data (SIGMOD)*, Vol. (25), N. (2), ACM Press, p. 205-216, juin1996.

[Harinath et al., 2009] Sivakumar Harinath, Matt Carroll, Sethu Meenakshisundaram, Robert Zare, Denny Guang-Yeu Lee, "*Professional Microsoft SQL Server Analysis Services 2008 with MDX*", Wiley Publishing, Inc., ISBN : 978-0-470-24798-3, mars 2009.

[Harinath et al., 2012] Sivakumar Harinath, Ronald Pihlgren, Denny Guang-Yeu Lee, John Sirmon, Robert M. Bruckner, "*Professional Microsoft SQL Server 2012 Analysis Services with MDX and DAX*", John Wiley & Sons, ISBN : 978-1-118-10110-0, octobre 2012.

[Hassan et al., 2012a]* **Ali Hassan, Frank Ravat, Olivier Teste, Ronan Tournier, Gilles Zurfluh, "Agrégations multiples différentiées dans les bases de données multidimensionnelles", *30ème congrès INFormatique des ORganisations et Systèmes d'Information et de Décision (INFORSID)*, Montpellier (France), p.447-462, 2012.**

[Hassan et al., 2012b]* **Ali Hassan, Frank Ravat, Olivier Teste, Ronan Tournier, Gilles Zurfluh, "Modélisation des bases de données multidimensionnelles à agrégations multiples et différentiées", *8èmes journées francophones sur les Entrepôts de Données et l'Analyse en ligne (EDA)*, Bordeaux (France), p. 57-71, 2012.**

[Hassan et al., 2012c]* **Ali Hassan, Frank Ravat, Olivier Teste, Ronan Tournier, Gilles Zurfluh, "Differentiated Multiple Aggregations in Multidimensional Databases", *14th Intl. Conf. on Data Warehousing and Knowledge Discovery (DAWAK)*, LNCS 7448, Springer, Vienna (Austria), p.93-104, 2012.**

[Hassan et al., 2013a]* **Ali Hassan, Frank Ravat, Olivier Teste, Ronan Tournier, Gilles Zurfluh, "Agrégations multiples différenciées dans les bases de données multidimensionnelles",** ***Journal Ingénierie des Systèmes d'Information (ISI),*** **Vol. (18), N. (2), p. 75-102, 2013.**

[Hassan et al., 2013b]* **Ali Hassan, Frank Ravat, Olivier Teste, Ronan Tournier, Gilles Zurfluh, "OLAP in Multifunction Multidimensional Databases",** ***17th East European Conference on Advances in Databases and Information Systems (ADBIS),*** **LNCS 8133, Springer, Gênes (Italie), p. 190-203, 2013.**

[Hassan et al., 2013c]* **Ali Hassan, Frank Ravat, Olivier Teste, Ronan Tournier, Gilles Zurfluh, "Opérateurs OLAP dans les bases de données multidimensionnelles multifonctions",** ***9èmes journées francophones sur les Entrepôts de Données et l'Analyse en ligne (EDA),*** **Blois (France), p. 69-78, 2013.**

[Hassan et al., 2014]* **Ali Hassan, Frank Ravat, Olivier Teste, Ronan Tournier, Gilles Zurfluh, "Multidimensional Databases Modeling with Differentiated Multiple Aggregations",** ***Journal of Decision Systems (JDS),*** **Vol. (23), N. (4), p. 437-459, 2014.**

[Hubert & Teste, 2009] Gilles Hubert, Olivier Teste, "Analyse multigraduelle OLAP", Journées Francophones Extraction et Gestion de Connaissances (EGC), p. 241-252, Strasbourg, 27-30 janvier 2009.

[Hüsemann et al., 2000] B. Hüsemann, J. Lechtenbörger, G. Vossen, "Conceptual data warehouse modeling", *2nd Intl. Workshop on Design and Management of Data Warehouses (DMDW'00),* Stockholm (Sweden), juin 2000.

[Huyn, 1997] Nam Huyn, "Multiple-View Self-Maintenance in Data Warehousing Environments", *23rd Intl. Conf. on Very Large Data Bases (VLDB),* Athens (Greece), p. 26-35, 1997.

[Inmon, 1996] W.H. Inmon, "*Building the Data Warehouse*", John Wiley and Sons, New York, NY, ISBN : 0764599445, 1996 (2nd ed.), 4th ed. 2005.

[Jagadish et al., 1999] H. V. Jagadish, Laks V. S. Lakshmanan, Divesh Srivastava, "What can Hierarchies do for Data Warehouses?", *25th Intl. Conf. on Very Large Data Bases (VLDB),* Morgan Kaufmann, p. 530-541, 1999.

[Jerbi, 2012] Houssem Jerbi, "*Personnalisation d'analyses décisionnelles sur des données multidimensionnelles*", thèse de doctorat, Université Toulouse 1 Capitole (UT1 Capitole) (France), janvier 2012.

[Jones & Hulme, 1996] P.D. Jones, M. Hulme, "Calculating regional climatic time series for temperature and precipitation: Methods and illustrations", *Intl. Journal of Climatology,* Vol. (16), p. 361-377, 1996.

[Jones et al., 1986] P. D. Jones, S. C. B. Raper, T. M. L. Wigley, "Southern Hemisphere Surface Air Temperature Variations: 1851–1984", *Journal of Climate and Applied Meteorology,* Vol. (25), p. 1213-1230, 1986.

[Kalnisa et al., 2002] Panos Kalnisa, Nikos Mamoulisb, Dimitris Papadias, "View selection using randomized search", *Data & Knowledge Engineering (DKE),* Vol. (42), N. (1), p. 89-111, 2002.

[Kerkad et al., 2013] Amira Kerkad, Ladjel Bellatreche, Dominique Geniet, "La Fragmentation Horizontale Revisitée: Prise en Compte de l'Interaction de Requêtes", *9èmes journées francophones sur les Entrepôts de Données et l'Analyse en ligne (EDA),* Blois (France), p. 117-131, 2013.

[Kimball, 1996] Ralph Kimball, "*The data warehouse toolkit: Practical Techniques for Building Dimensional Data Warehouses*", John Wiley and Sons, ISBN : 0-471-15337-0, 1996, 2nd ed. : Ralph Kimball, Margaery Ross, "*The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*", 2nd Edition, John Wiley & Sons, 2002.

[Kotidis & Roussopoulos, 1999] Yannis Kotidis, Nick Roussopoulos, "DynaMat: A Dynamic View Management System for Data Warehouses", *Intl. Conf. on Management of Data (SIGMOD)*, Vol. (28), N. (2), ACM Press, p. 371-382, juin 1999.

[Kotidis & Roussopoulos, 2001] Yannis Kotidis, Nick Roussopoulos, "A case for dynamic view management", *Database Systems Journal*, Vol. (26), N. (4), p. 388-423, 2001.

[Kung et al., 1975] H. T. Kung, E Luccio, E P. Preparata, "On finding the maxima of a set of vectors", *Journal of the ACM (JACM)*, ACM Press, Vol. (22), N. (4), p. 469-476, 1975.

[Labio et al., 2000] Wilburt Juan Labio, Jun Yang, Yingwei Cui, Hector Garcia-Molina, Jennifer Widom, "Performance Issues in IncrementalWarehouse Maintenance", *26th Intl. Conf. on Very Large Data Bases (VLDB)*, Le Caire (Egypte), p. 461-472, 2000.

[Lawrence & Rau-Chaplin, 2006] Michael Lawrence, Andrew Rau-Chaplin, "Dynamic View Selection for OLAP", *8th Intl. Conf. on Data Warehousing and Knowledge Discovery (DaWaK)*, LNCS 4081, Springer, p. 33-44, 2006.

[Le Parc, 1997] Annig Le Parc, "*Une algèbre et un langage graphique pour les bases de données objet intégrant le concept de version*", thèse de doctorat, Université Paul Sabatier - Toulouse 3 (France), Décembre 1997.

[Lee & Hammer, 2001] Minsoo Lee, Joachim Hammer, "Speeding up materialized view selection in data warehouses using a randomized algorithm", *Intl. Journal of Cooperative Information Systems (IJCIS)*, Vol. (10), N. (3), p. 327-353, 2001.

[Lee & Ong, 1995] Hing-Yan Lee , Hwee-Leng Ong, "A New Visualisation Technique for Knowledge Discovery in OLAP", *1st Intl. Workshop on Integration of Knowledge Discovery in Databases with Deductive and Object-Oriented Databases*, p. 23-25, 1995.

[Lehner, 1998] Wolfgang Lehner, "Modelling Large Scale OLAP Scenarios", *6th Intl. Conf. on Extending Database Technology - Advances in Database Technology (EDBT)*, LNCS 1377, Springer, p. 153-167, 1998.

[Lenz & Shoshani, 1997] Hans-Joachim Lenz, Arie Shoshani, "Summarizability in OLAP and Statistical Data Bases", *9th Intl. Conf. on Scientific and Statistical Database Management (SSDBM)*, IEEE Computer Society, p. 132-143, 1997.

[Lenz & Thalheim, 2005] Hans-Joachim Lenz, Bernhard Thalheim, "OLAP Schemata for Correct Applications", *VLDB Workshop Trends in Enterprise Application Architecture (TEAA)*, LNCS 3888, Springer-Verlag Berlin Heidelberg, Trondheim (Norvège), p. 99-113, août, 2005.

[Lenz & Thalheim, 2006] Hans-Joachim Lenz, Bernhard Thalheim, "Warning-cube may mislead", *4th Intl. Conf. on Computer Science and Information Technology (CSIT)*, Vol. (2), p. 7-16, Amman, Jordanie, 4-5 avril 2006.

[Lenz & Thalheim, 2009] Hans-J. Lenz, Bernhard Thalheim, "A Formal Framework of Aggregation for the OLAP-OLTP Model", *Journal of Universal Computer Science*, Vol. (15), N. (1), p. 273-303, 2009.

[Li & Wang, 1996] Chang Li, Xiaoyang Sean Wang, "A Data Model for Supporting On-Line Analytical Processing", *5th Intl. Conf. on Information and Knowledge Management (CIKM'96)*, ACM Press, Rockville (Maryland, USA), p. 81-88, 12-16 novembre 1996.

[Li et al., 2005] Hongsong Li, Houkuan Huang, Shijin Liu, "PCM: Select Materialized Cells in Data Cubes", *7th Intl. Conf. on Data Warehousing and Knowledge Discovery (DaWaK)*, LNCS 3589, Springer, Copenhagen (Danemark), p. 168-178, août 2005.

[Liang et al., 2001] Weifa Liang, Hui Wang, Maria E. Orlowska, "Materialized view selection under the maintenance time constraint," *Data & Knowledge Engineering (DKE)*, Vol. (37), N. (2), p. 203-216, 2001.

[Luján-Mora, 2005] Sergio Luján-Mora, "*Data Warehouse Design with UML"*, thèse de doctorat, Université d'Alicante (Espagne), juin 2005.

[Luján-Mora et al., 2006] Sergio Luján-Mora, Juan Trujillo, Il-Yeol Song, "A UML profile for multidimensional modeling in data warehouses", *Data & Knowledge Engineering (DKE)*, Vol. (59), N. (3), Elsevier, p. 725-769, décembre 2006.

[Malinowski et al., 2006] E. Malinowski, E. Zimanyi, "Hierarchies in a multidimensional model: From conceptual modeling to logical representation", *Data & Knowledge Engineering*, Vol. (59), N. (2), p. 348-377, 2006.

[Malinowski et al., 2008] E. Malinowski, E. Zimanyi, "A conceptual model for temporal data warehouses and its transformation to the ER and the object-relational models", *Data & Knowledge Engineering*, Vol. (64), N. (1), p. 101-133, 2008.

[Mangisengi & Tjoa, 1998] O. Mangisengi, A.M. Tjoa, "A multidimensional modeling approach for OLAP within the framework of the relational model based on quotient relations", *1st Intl. Workshop on Data Warehousing and OLAP (DOLAP)*, Bethesda (Maryland, USA), p. 40-46, (1998).

[Maniatis et al., 2005] Andreas Maniatis, Panos Vassiliadis, Spiros Skiadopoulos, Yannis Vassiliou, George Mavrogonatos, Ilias Michalarias, "A Presentation Model & Non-Traditional Visualization for OLAP", *Intl. Journal of Data Warehousing & Mining (ijDWM)*, Vol. (1), N. (1), p. 1-36, 2005.

[Marcel, 1998] P. Marcel "*Manipulation de Données Multidimensionnelles et Langages de Règles*", thèse de Doctorat, Institut des Sciences Appliquées, Lyon (France), 1998.

[Mendelzon & Vaisman, 2003] A.O. Mendelzon, A.A. Vaisman, "Time in Multidimensional Databases", Chapitre VI, *Multidimensional Databases: Problems and Solutions*, Maurizio Rafanelli (Ed.), Idea Publishing Group (IGP), ISBN : 1-59140-053-8, p. 166-199, juin 2003.

[Messaoud, 2006] Riadh Ben Messaoud, "*Couplage de l'analyse en ligne et de la fouille de données pour l'exploration, l'agrégation et l'explication des données complexes*", thèse de doctorat, Université Lumière Lyon 2 (France), novembre 2006.

[Midouni et al., 2009] Sid-Ahmed-Djallal Midouni, Jérôme Darmont, Fadila Bentayeb, "Approche de modélisation multidimensionnelle des données complexes : application aux données médicales", *5èmes Journées francophones sur les Entrepôts de Données et l'Analyse en ligne (EDA)*, Montpellier (France), p. 155-166, 2009.

[Nandi et al., 2011] Arnab Nandi, Cong Yu, Phil Bohannon, Raghu Ramakrishnan, "Distributed Cube Materialization on Holistic Measures", *27th Intl. Conf. on Data Engineering (ICDE)*, p. 183-194, 2011.

[Nguyen et al., 2000] Thanh Binh Nguyen, A. Min Tjoa, Roland Wagner, "An Object Oriented Multidimensional Data Model for OLAP", *1st Intl. Conf. on Web-Age Information Management (WAIM)*, LNCS 1846, Springer, p. 69-82, 2000.

[O'Neiland & Graefe, 1995] Patrick O'Neiland, Goetz Graefe, "Multi-table joins through bitmapped join indexes", *Intl. Conf. on Management of Data (SIGMOD)*, Vol. (24), N. (3), ACM Press, p. 8-11, septembre 1995.

[Oliveira et al., 2011] Rui Oliveira, Fatima Rodrigues, Paulo Martins, Jõao Paulo Moura, "Extending the Dimensional Templates Approach to Integrate Complex Multidimensional Design Concepts", 13th *Intl. Conf. on Data Warehousing and Knowledge Discovery (DaWaK)*, LNCS 6862, Springer, p. 26-38, 2011.

[Oracle 12c, 2013] "Oracle® Database Data Cartridge Developer's Guide, 12*c* Release 1 (12.1)", Chapitre 22, *User-Defined Aggregate Functions Interface*, http://docs.oracle.com/cd/E57425_01/121/ADDCI/toc.htm , avril 2013.

[Ordonez et al., 2011] Carlos Ordonez, Zhibo Chen, Javier García-García, "Interactive Exploration and Visualization of OLAP Cubes", *14th intl. workshop on Data Warehousing and OLAP (DOLAP)*, Glasgow (Scotland, UK), ACM Press, p. 83-88, October 2011.

[Özsoyoglu et al., 1985] Gültekin Özsoyoglu, Zehra Meral Özsoyoglu, Francisco Mata, "A Language and a Physical Organization Technique for Summary Tables", *Intl. Conf. on Management of Data (SIGMOD)*, Vol. (14), N. (4), ACM Press, p. 3-16, décembre 1985.

[Özsoyoglu et al., 1987] Gültekin Özsoyoglu, Zehra Meral Özsoyoglu, Victor Matos, "Extending Relational Algebra and Relational Calculus with Set-Valued Attributes and Aggregate Functions", *ACM Transactions on Database Systems (TODS)*, Vol. (12), N. (4), ACM Press, p. 566-592, 1987.

[Paraboschi et al, 2003] Stefano Paraboschi, Giuseppe Sindoni, Elena Baralis, Ernst Teniente, "Materialized Views in Multidimensional Databases", Chapitre VIII, *Multidimensional Databases: Problems and Solutions*, Maurizio Rafanelli (Ed.), Idea Publishing Group (IGP), ISBN 1-59140-053-8, p. 222-251, 2003.

[Pardillo et al., 2010] Jesús Pardillo, Jose-Norberto Mazón, Juan Trujillo, "Extending OCL for OLAP querying on conceptual multidimensional models of data warehouses", *Journal Information Sciences*, Vol. (180), N. (5), p. 584-601, mars 2010.

[Park et al., 2005] Byung-Kwon Park, Hyoil Han, Il-Yeol Song, "XML-OLAP: A Multidimensional Analysis Framework for XML Warehouses", *7th Intl. Conf. on Data Warehousing and Knowledge Discovery (DaWaK)*, LNCS 3589, Springer, p. 32-42, 2005.

[Parssian, 2006] Amir Parssian, "Managerial decision support with knowledge of accuracy and completeness of the relational aggregate functions", *Intl. Journal Decision Support Systems*, Vol. (42), N. (3), p. 1494-1502, décembre, 2006.

[Pedersen T.B., 2000] Torben Bach Pedersen, "*Aspects of Data Modeling and Query Processing for Complex Multidimensional Data*", thèse de doctorat, Université d'Aalborg (Danemark), 2000.

[Pedersen T.B. & Jensen, 1998] Torben Bach Pedersen, Christian S. Jensen, "Research Issues in Clinical Data Warehousing", *10th Intl. Conf. on Scientific and Statistical Database Management (SSDBM'98)*, Capri, Italie, p. 43-52, juillet 1998.

[Pedersen T.B. & Jensen, 1999] Torben Bach Pedersen, Christian S. Jensen, "Multidimensional Data Modeling for Complex Data", *15th Intl. Conf. on Data Engineering (ICDE'99)*, Sydney, Australie, p. 363-345, mars 1999.

[Pedersen T.B. et al., 2001] Torben Bach Pedersen, Christian S. Jensen, Curtis E. Dyreson, "A foundation for capturing and querying complex multidimensional data", *Information Systems (IS)*, Vol. (26), N. (5), Elsevier, p. 383-423, juillet 2001.

[Pedersen T.B. et al., 2009] Torben Bach Pedersen, Junmin Gu, Arie Shoshani, Christian S. Jensen, "Object-extended OLAP querying", *Intl. Journal Data & Knowledge Engineering*, Vol. (68), N. (5), p. 453-480, mai 2009.

[Pérez et al., 2008] Juan Manuel Pérez, Rafael Berlanga, Maria José Aramburu, Torben Bach Pedersen, "Integrating Data Warehouses with Web Data : A Survey", *Journal IEEE Transactions on Knowledge and Data Engineering*, Vol. (20), N. (7), p. 940-955, juillet 2008.

[Pourrabas & Rafanelli, 2000] Elaheh Pourabbas, Maurizio Rafanelli, "Hierarchies and Relative Operators in the OLAP Environment", *Intl. Conf. on Management of Data (SIGMOD)*, Vol. (29), N. (1), ACM Press, p. 32-37, 2000.

[Pourrabas & Rafanelli, 2003] Elaheh Pourabbas, Maurizio Rafanelli, "Hierarchies", Chapitre IV, *Multidimensional Databases: Problems and Solutions*, Maurizio Rafanelli (Ed.), Idea Publishing Group (IGP), ISBN 1-59140-053-8, p. 91-115, 2003.

[Prat et al., 2010] Nicolas Prat, Isabelle Comyn-Wattiau, Jacky Akoka, "Representation of aggregation knowledge in OLAP systems", *18th European Conf. on Information Systems (ECIS)*, AIS Electronic Library, 2010.

[Prat et al., 2011] Nicolas Prat, Isabelle Comyn-Wattiau, Jacky Akoka, "Combining objects with rules to represent aggregation knowledge in data warehouse and OLAP systems", *Journal Data & Knowledge Engineering*, Vol. (70), N. (8), p. 732-752, 2011.

[Rafanelli & Ricci, 1983] Maurizio Rafanelli, Fabrizio L. Ricci, "Proposal of a Logical Model for Statistical Databases", *2nd Intl. Workshop on Statistical Database Management (SSDBM)*, Los Altos, CA, ISBN: 1-87654-234-X, p. 264-272, 1983.

[Rafanelli & Shoshani, 1990] Maurizio Rafanelli, Arie Shoshani, "STORM: a statistical object representation model", *5th Intl. Conf. on Statistical and Scientific Database Management (SSDBM)*, p. 14-29, 1990.

[Ravat, 2007] Franck Ravat, "*Modèles et outils pour la conception et la manipulation de systèmes d'aide à la décision*", mémoire de HDR (Habilitation à diriger des recherches), Université Toulouse 1 Capitole (UT1 Capitole) (France), décembre 2007.

[Ravat & Teste, 2000] Franck Ravat, Olivier Teste, "Object-Oriented Decision Support System", *2nd Intl. Conf. on Enterprise Information Systems (ICEIS'00)*, Stafford, UK, eds. B. Sharp, J. Cordeiro, J. Filipe, ISBN : 972-98050-1-6, p. 79-84, juillet 2000.

[Ravat & Teste, 2006] Franck Ravat, Olivier Teste, "Supporting Data Changes in Multidimensional Data Warehouses", Intl. Review on Computers and Software, Vol. (1), N. (3), Praize Worthy Prize, Wantag - USA, p. 251-259, novembre 2006.

[Ravat et al., 2006] Franck Ravat, Olivier Teste, Gilles Zurfluh, "A Multiversion-based Multidimensional Model", *8th Intl. Conf. on Data Warehousing and Knowledge Discovery (DAWAK)*, LNCS 4081, Springer, p. 75-84, 2006.

[Ravat et al., 1999] Franck Ravat, Olivier Teste, Gilles Zurfluh, "Towards Data Warehouse Design", *8th Intl. Conf. on Information and Knowledge Management (CIKM'99)*, ACM Press Susan Gauch, Kansas City (Missouri, USA), p. 359-366, novembre 1999.

[Ravat et al., 2001] Franck Ravat, Olivier Teste, Gilles Zurfluh, "Modélisation multidimensionnelle des systèmes décisionnels". *1ères Journées d'Extraction et de Gestion des Connaissances (EGC'01)*, Revue des Sciences et Technologies de l'Information, Série RIA-ECA (Extraction des Connaissances et Apprentissage), Nantes, Vol. (1), N. (1-2), ISBN : 2-7462-0216-6, p. 201-212, 17-19 janvier 2001.

[Ravat et al., 2007a] Franck Ravat, Olivier Teste, Ronan Tournier, "OLAP Aggregation Function for Textual Data Warehouse", *Intl. Conf. on Enterprise Information Systems (ICEIS)*, Funchal, Madeira - Portugal, 12-17 juin 2007, Vol. DISI, INSTICC Press, p. 151-156, juin 2007.

[Ravat et al., 2007b] Franck Ravat, Olivier Teste, Ronan Tournier, Gilles Zurfluh, "Graphical Querying Multidimensional Databases", *11th East-European Conf. on Advances in Databases and Information Systems (ADBIS)*, LNCS 4690, Springer, p. 298-313, 2007.

[Ravat et al., 2008] Franck Ravat, Olivier Teste, Ronan Tournier, Gilles Zurfluh, "Algebraic and graphic languages for OLAP manipulations", *Intl. Journal of Data Warehousing and Mining (ijDWM)*, Idea Publishing Group (IGP), D. Taniar, Vol. (4), N. (1), p.17-46, 2008.

[Red, 1997] Red Brick Systems Inc., "*Star Schema Processing for Complex Queries*", White paper, juillet 1997.

[Rizzi et al., 2006] Stefano Rizzi, Alberto Abelló, Jens Lechtenbörger, Juan Trujillo, "Research in data warehouse modeling and design: dead or alive? ", *9th Intl. Workshop on Data Warehousing and OLAP (DOLAP)*, Arlington (Virginia, USA), p. 3-10, novembre 2006.

[Salehi, 2009] Mehrdad Salehi, "*Developing a Model and a Language to Identify and Specify the Integrity Constraints in Spatial Datacubes*", thèse de doctorat, Faculté des études supérieures de l'Université Laval, Canada, 2009.

[Sapia et al., 1998] Carsten Sapia, Markus Blaschka, Gabriele Höfling, Barbara Dinter, "Extending the E/R Model for the Multidimensional Paradigm", *Advances in Database Technologies, ER'98 Workshops on Data Warehousing and Data Mining, Mobile Data Access, and Collaborative Work Support and Spatio-Temporal Data Management (ER Workshops)*, LNCS 1552, Springer, p. 105-116, 1998.

[Schneider, 2003] Michel Schneider, "Well-formed data warehouse structures", *5th Intl. Workshop on Design and Management of Data Warehouses (DMDW)*, CEUR Workshop Proceedings, Vol. (77), CEUR-WS.org, p. 2.1-2.13, 2003.

[Schneider, 2008] Michel Schneider, "A general model for the design of data warehouses", *Intl. Journal of Production Economics*, Elsevier, Vol. (112), N. (1), p. 309-325, 2008.

[Shoshani, 2003] Arie Shoshani, "Multidimensionality in Statistical, OLAP, and Scientific Databases. Multidimensional Databases", Chapitre II, *Multidimensional Databases: Problems and Solutions*, Maurizio Rafanelli (Ed.), Idea Publishing Group (IGP), ISBN 1-59140-053-8, p. 46-68, 2003.

[Sifer, 2003] Mark Sifer, "A Visual Interface Technique for Exploring OLAP Data with Coordinated Dimension Hierarchies", *12th Intl. Conference on Information and Knowledge Management*, p. 532-535, 2003.

[Silva et al., 2008] Joel da Silva, Valeria C. Times, Ana Carolina Salgado "A Set of Aggregation Functions for Spatial Measures", *11th intl. workshop on Data warehousing and OLAP (DOLAP)*, Napa Valley, (California, USA), ACM, p. 25-32, octobre 2008.

[Smith et al., 2009] Bryan C. Smith, C. Ryan Clay, Hitachi Consulting, "*Microsoft SQL Server 2008 MDX Step by Step*", Microsoft Press, ISBN : 978-0735626188, février 2009.

[Techapichetvanich & Datta, 2005] Kesaraporn Techapichetvanich, Amitava Datta, "Interactive Visualization for OLAP", *Intl. Conference on Computational Science and its Applications (ICCSA)*, LNCS 3482, Springer, p. 206-214, 2005.

[Teste, 2000] Olivier Teste, "*Modélisation et Manipulation d'Entrepôts de Données Complexes et Historisées*", thèse de doctorat, Université Paul Sabatier Toulouse 3 (France), décembre 2000.

[Teste, 2009] Olivier Teste, "*Modélisation et manipulation des systèmes OLAP : de l'intégration des documents à l'usager*", mémoire de HDR (Habilitation à diriger des recherches), Université Paul Sabatier Toulouse 3 (France), décembre 2009.

[Theodoratos & Sellis, 1997] Dimitri Theodoratos, Timos Sellis, "Data Warehouse Configuration", *23rd Intl. Conf. on Very Large Data Bases (VLDB)*, Athens (Greece), p. 126-135, 1997.

[Thomas & Datta, 1997] Helen Thomas, Anindya Datta, "A Conceptual Model and Algebra for On-Line Analytical Processing in Data Warehouses", *7th Workshop on Information Technologies and Systems (WITS)*, p. 91-100, 1997.

[Torlone, 2003] Riccardo Torlone, "Conceptual Multidimensional Models", Chapitre III, *Multidimensional Databases: Problems and Solutions*, Maurizio Rafanelli (Ed.), Idea Publishing Group (IGP), ISBN 1-59140-053-8, p. 69-90, 2003.

[Tournier, 2007] Ronan Tournier, "*Analyse en ligne (OLAP) de documents*", thèse de doctorat, Université Paul Sabatier Toulouse 3 (France), décembre 2007.

[Trujillo et al., 1998] J. Trujillo, M. Palomar, "An Object-Oriented Approach to Multidimensional Database Conceptual Modeling", *1st Intl. Workshop on Data Warehousing and OLAP (DOLAP)*, Bethesda (Maryland, USA), p. 16-21, novembre 1998.

[Trujillo et al., 2003] J. Trujillo, S. Luján-Mora, I. Song, "Applying UML for designing multidimensional databases and OLAP applications", *Advanced Topics in Database Research*, Vol. (2), Hershey, PA: Idea Group Publishing, p. 13-36, 2003.

[Tryfona et al., 1999] Nectaria Tryfona, Frank Busborg, Jens G. Borch Christiansen, "starER: A Conceptual Model for Data Warehouse Design", *2nd ACM Intl. Workshop on Data Warehousing and OLAP (DOLAP)*, ACM Press, p. 3-8, 1999.

[Tsois et al., 2001] Aris Tsois, Nikos Karayannidis, Timos Sellis, "MAC: Conceptual data modeling for OLAP", *3rd Intl. Workshop on Design and Management of Data Warehouses (DMDW)*, Interlaken (Switzerland), juin 2001.

[Ullman, 1996] Jeffrey D. Ullman, "Efficient implementation of data cubes via materialized views", *2nd Intl. Conf. on Knowledge Discovery and Data Mining (KDD)*, p. 386-388, 1996.

[Vassiliadis, 1998] Panos Vassiliadis, "Modelling Multidimensional Databases, Cubes and Cube Operations", *10th Intl. Conf. on Scientific and Statistical Database Management (SSDBM'98)*, Capri, Italie, p. 53-62, juillet 1998.

[Vassiliadis, 2000] Panos Vassiliadis, "*Data Warehouse Modeling and Quality Issues*", thèse de doctorat, Université d'Athens (États-Unis), 2000.

[Vassiliadis & Sellis, 1999] Panos Vassiliadis, Timos Sellis, "A Survey of Logical Models for OLAP Databases", *Intl. Conf. on Management of Data (SIGMOD)*, Vol. (28), N. (4), ACM Press, p. 64-69, décembre 1999.

[Vassiliadis & Skiadopoulos, 2000] Panos Vassiliadis, Spiros Skiadopoulos, "Modelling and optimisation issues for multidimensional databases", *Intl. Conf. on Advanced Information Systems Engineering CAISE*, LNCS 1789, Springer, p. 482-497, 2000.

[Vassiliadis et al., 2002] Panos Vassiliadis, Alkis Simitsis, Spiros Skiadopoulos, "Modeling ETL activities as graphs", *Intl. Workshop on Design and Management of Data Warehouses (DMDW)*, CAISE Workshops, p. 52-61, 2002.

[Widom, 1995] J. Widom, "Research problems in data warehousing", *4th Intl. Conf. on Information and Knowledge Management (CIKM'95)*, ACM, Baltimore, Maryland, USA, p. 25-30, novembre 1995.

[Yang & Widom, 2000] J. Yang, J. Widom, "Temporel View Self-Maintenance in a warehousing Environment", *7th Intl. Conf. on Extending Database Technology (EBDT)*, Konstanz, Allemagne, p. 395-412, mars 2000.

[Yang et al., 1997] Jian Yang, Kamalakar Karlapalem, Qing Li, "Algorithms for materialized view design in a-data warehousing environment", *23rd Intl. Conf. on Very Large Data Bases (VLDB)*, Morgan Kaufmann, Athens (Greece), p. 136-145, août 1997.

[Yu et al., 2003] Jeffrey Xu Yu, Xin Yao, Chi-Hon Choi, Gang Gou, "Materialized View Selection as Constrained Evolutionary Optimization", *IEEE transactions on Systems, Man and Cybernetics, part C*, Vol. (33), N. (4), p. 458-467, 2003.

[Yu et al., 2005] Songmei Yu, Vijayalakshmi Atluri, Nabil Adam, "Selective View Materialization in a Spatial Data Warehouse", *7th Intl. Conf. on Data Warehousing and Knowledge Discovery (DaWaK)*, LNCS 3589, Springer, Copenhagen (Danemark), p. 157-167, août 2005.

[Zhuge et al., 1997] Yue Zhuge, Janet L. Wiener, Hector Garcia-Molina, "Multiple view consistency for data warehousing", *13th Intl. Conf. on Data Engineering*, p. 289-300, 1997.

[Zhuge et al., 1998] Yue Zhuge, Hector Garcia-Molina, Janet L. Wiener, "Consistency Algorithms for Multi-Source Warehouse View Maintenance", *Technical Report, Stanford InfoLab, Distributed and Parallel Databases*, Vol. (6), N. (1), p. 7-40, 1998.