Entrepôts de données multidimensionnelles réduites : principes et expérimentations

Faten Atigui, Franck Ravat, Jiefu Song, Gilles Zurfluh

IRIT (UMR 5505)
Institut de Recherche en Informatique de Toulouse
118 route de Narbonne
F-31062 Toulouse, France
{atigui, ravat, song, zurfluh}@irit.fr

Résumé. Notre objectif est de proposer une solution pour la réduction de données d'un Entrepôt de Données Multidimensionnelles (EDM) afin d'obtenir des schémas agrégés sur différentes périodes et de ne retenir que les informations pertinentes pour les prises de décision. Dans un premier temps, nous proposons une solution pour la modélisation des EDM réduits basée sur des états contenant des schémas en étoile et des opérateurs de réduction pour définir les schémas réduits. Dans un second temps, nous décrivons nos expérimentations et les résultats obtenus dans différents contextes : BD R-OLAP sans réduction et BD R-OLAP réduite. Nous montrons que, quelques soit le type d'analyse, les exécutions dans un contexte réduit sont plus performantes.

1 Introduction

Les systèmes d'aide à la décision sont généralement supportés par des Entrepôts de Données Multidimensionnelles (EDM). Un schéma d'EDM est basé sur des faits (sujets d'analyse) et des dimensions (axes d'analyses). Les faits contiennent des indicateurs tandis que les dimensions contiennent les paramètres d'analyse. Ces paramètres sont organisés en hiérarchies permettant de classer les paramètres du niveau de granularité minimale vers le niveau de granularité maximale.

Par définition, un EDM stocke de manière permanente les données décisionnelles et ces données sont régulièrement mises à jour (ajouts de nouvelles valeurs). De ce fait, un EDM gère des volumes de données de plus en plus importants, ce qui entraîne de nombreux problèmes en termes de performance et de stockage. Cependant, la pertinence d'une donnée décisionnelle est susceptible de décroître avec l'âge : les informations détaillées sont essentielles pour des données les plus récentes mais une information plus agrégée est souvent suffisante pour des données plus anciennes Skyt et al. (2008). Par exemple, un responsable marketing analyse les ventes à un niveau de granularité permettant de manipuler chaque produit durant les 2 ou 3 années les plus récentes mais ce niveau de granularité peut s'avérer inutile pour une période plus ancienne (la plupart de ces produits n'existant plus) ; le décideur pourra analyser

ces ventes non plus par produit, mais au niveau plus général de la catégorie des produits qui reste stable dans le temps. Faisant face à ce grand nombre de données dont une grande partie n'est pas pertinente pour la prise de décision, notre objectif est à la fois d'augmenter la performance de traitement des requêtes portant sur un gros volume de données et de faciliter la tâche de l'analyste en entreposant uniquement des informations pertinentes au cours du temps. Par conséquent, seules les informations nécessaires à ces analyses lui seront proposées. Etant donné que la pertinence d'une donnée détaillée décroit dans le temps, nous proposons une suppression sélective des données en fonction des niveaux de granularité. Cette réduction des données est basée sur l'agrégation progressive des données, autrement dit, la suppression des détails devenus obsolètes sur les données au cours du temps ; par exemple, les ventes pourront être analysées au niveau des produits jusqu'en 2010 et pour les analyses relatives aux années antérieures, nous ne proposerons qu'une analyse au niveau de la catégorie.

L'enjeu de notre proposition est donc de fournir un environnement d'analyses multidimensionnelles adapté aux besoins des décideurs en supprimant les niveaux de granularité détaillée inutile pour les périodes de temps les plus anciennes.

Cet article est composé comme suit : la section 2 propose un état de l'art sur la réduction de données. La section 3 définit notre modèle de base de données multidimensionnelle basé sur des réductions de données. La section 4 fournit une expérimentation dans différents environnements d'implantation.

2 Etat de l'art

La réduction de données permet de diminuer la quantité de données pertinentes aux décideurs tout en augmentant la qualité des analyses Udo et Afolabi (2011). Dans le contexte du décisionnel, la réduction de données est une technique initialement utilisée pour la fouille de données. Par exemple, nous pouvons citer les travaux de Okun et Priisalu (2007) et Udo et Afolabi (2011).

Dans le domaine des entrepôts de données Garcia-Molina et al. (1998) sont les premiers à proposer une solution pour la suppression de données. Plus précisément, ils ont étudié l'expiration de données dans les vues matérialisées et notamment la maintenance de ces dernières après suppression de données à l'aide d'un ensemble de vues standards prédéfinies.

Dans le domaine du multidimensionnel, Chen et al. (2002) propose une architecture pour intégrer des flots de données dans un EDM et réduire sa taille. Cette réduction est prédéfinie et automatique, les données détaillées sont disponibles pour un temps limité. De plus, elle se limite à une table de fait en utilisant une agrégation partielle du cuboïde (treillis). Skyt et al. (2008) propose une technique pour l'agrégation progressive d'un fait. Cette étude permet de spécifier des critères d'agrégation d'un fait en fonction des niveaux de granularité des dimensions. Les auteurs proposent aussi une solution pour interroger les faits agrégés. Comme mentionné dans Iftikhar et Pedersen (2011), ce travail repose sur des définitions formelles mais un exemple concret avec une solution d'implantation précise manquent. Dans Iftikhar et Pedersen (2011), une agrégation de données est proposée aussi bien au niveau conceptuel qu'au niveau

de l'implantation, et une évaluation est réalisée. Cette solution repose sur une table relationnelle contenant différentes granularités temporelles : seconde, minute, heure, mois et année. Ces travaux restent uniquement centrés sur la table de fait. Iftikhar et Pedersen (2010) et Iftikhar et Pedersen (2011) utilisent une table temporelle pour définir une réduction de données graduelle.

Notre objectif est de généraliser le mécanisme de réduction d'un fait à l'ensemble des données entreposées dans un EDM. Cette réduction doit tenir compte des besoins d'analyses du décideur. Ainsi, nous souhaitons proposer un environnement d'analyse cohérent qui facilitera la tâche du décideur en lui proposant un espace de travail compatible avec des analyses de données sémantiquement cohérentes Atigui (2013).

3 Notre modèle

Notre objectif n'est pas de seulement s'intéresser à la réduction des données d'un fait Skyt et al. (2008), mais de spécifier des schémas agrégés dans le temps afin de ne conserver que les données pertinentes aux prises de décisions. Par conséquent, toutes les dimensions sont susceptibles de subir une réduction à différents niveaux de granularités en fonction des besoins du décideur. Seules les informations jugées inutiles du point de vue décideur sont supprimées dans l'EDM afin de lui proposer uniquement les données nécessaires à ses analyses et par conséquent d'augmenter la performance d'exécution des requêtes.

3.1 Cas d'étude

Un responsable marketing a exprimé ses besoins. Pour les 3 dernières années, l'analyse des ventes s'effectue au niveau de détail le plus fin, à savoir les produits, les clients et la date de la vente (*cf.* figure 1). Dans la période précédente (2000 à 2010), les analyses seront agrégées au niveau de la gamme de produits, date et ville de clients car les analyses se référant aux clients et aux produits sont considérées comme non judicieuses par les décideurs. En effet, tous les produits n'existent pas forcément et les analyses précises par client ne sont pas utiles aux prises de décision (*cf.* figure 2). Enfin, avant 2000, seules les analyses annuelles par gamme de produit sont considérées comme utiles pour les décideurs (*cf.* figure 3).

En fait, les trois figures suivantes représentent l'évolution de la réduction des données dans le temps. Chaque figure représente un état de l'EDM réduit. Chaque état est basé sur un sujet d'analyse (fait) connecté à différentes dimensions. Chaque fait se compose d'un ou plusieurs indicateurs. Par exemple, dans la figure 1, le fait nommé *Sales* possède deux indicateurs : *Quantity* and *Amount*. Chaque dimension représentant un axe d'analyse est composée de paramètres d'analyse. Par exemple, dans la figure 1, le fait *Sales* est connecté à 3 dimensions : *Products*, *Customers* and *Times*. Les attributs de dimensions sont organisés en hiérarchies représentant une vision d'un axe d'analyse. Chaque schéma est basé sur la notation conceptuelle proposée par Golfarelli et Rizzi (1998).

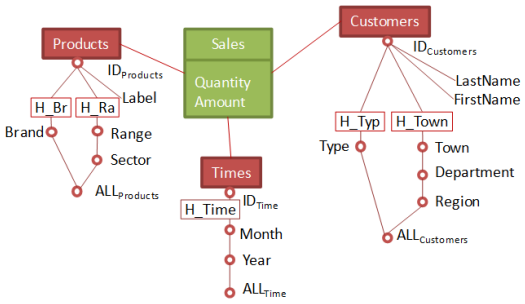

FIG. 1: Etat courant de l'EDM réduit

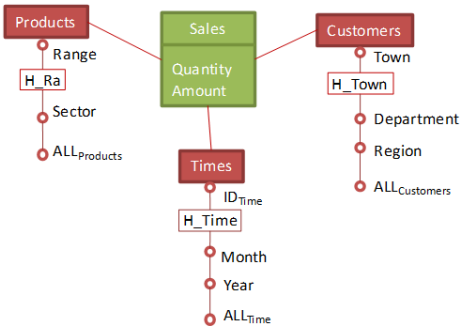

FIG. 2: Premier état réduit de l'EDM.

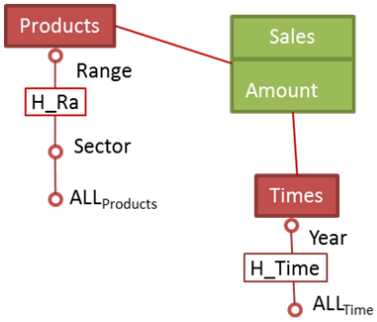

FIG. 3: Second état réduit de l'EDM.

3.2 Concepts du modèle

Nous proposons de modéliser un EDM réduit à l'aide d'états. L'état courant correspond à l'état de l'EDM valide ce jour. Les états passés correspondent aux différentes réductions de données opérées dans le temps. Chaque état repose sur un schéma multidimensionnel.

Définition 1. Un EDM est défini par $S = (n^S; \mathcal{E}; \mathcal{M}ap)$ où :
– $n^S \in N$ est le nom de l'EDM ;
– $\mathcal{E} = E_1; ...; E_n$ est un ensemble de n états composant l'EDM ;
– $\mathcal{M}ap : \mathcal{E} \rightarrow \mathcal{E} | \mathcal{M}ap(E_k) = E_{k+1}$est une fonction de dérivation définissant l'état E_{k+1} obtenu par réduction de l'état E_k.
Définissons F et D tel que $F = \{F_1, ..., F_x\}$ est un ensemble fini de faits, $x \geq 1$ et $D = \{D_1, ..., D_m\}$ un ensemble fini de dimensions, $m \geq 2$.

Définition 2. Un état est un schéma en étoile associé à une période de temps donnée tel que $Ei = (F_i; \mathcal{D}_i; T_i)$où :
– $F_i \in F$ est un fait représentant un sujet d'analyse ;
– $\mathcal{D}_i = \{D_{times}; D_1; ...; D_m\} \subseteq D$ est un ensemble de dimensions associées au fait avec nécessairement une dimension temporelle notée D_{times} ;
– $T_i = [t_{start}; t_{end}[$ est un intervalle temporel associé à un état et défini sur la dimension D_{times}.

Pour définir T_i, nous adoptons un modèle temporel numérique, linéaire et discret qui approche le temps de manière granulaire au travers d'unités temporelles d'observation (Wang, et al. 1997). Un grain temporel est un entier relativement à une unité temporelle ; nous adoptons les unités temporelles standards manipulées au travers de fonctions Year, Quarter, Month, Day... Par exemple, Year(1990) définit l'instant 1990 à l'unité temporelle année. Un instant est un grain temporel. On note T_{now} l'instant présent qui se caractérise par son caractère dynamique, c'est-à-dire que T_{now} change perpétuellement en fonction de l'écoulement du temps. Un intervalle temporel est donc défini par un couple d'instants T_{deb} et T_{fin}. Ces instants peuvent être fixes (grains temporels) ou bien dynamiques (définis relativement à l'instant Tnow).

Exemple. La figure 4 représente les 3 états de notre cas d'étude. Elle illustre le principe des états dérivés par réduction d'un état précédent. Ainsi, l'EDM est défini comme suit : $\mathcal{E} = \{E_1; E_2; E_3\}$ avec $\mathcal{M}ap = \{(E_1, E_2); (E_2, E_3)\}$ où :

- $E_1 = (F_{sales}; \{D_{products}; D_{times}; D_{customers}\}; [Year(T_{now}) - 4; Year(T_{now})[)$;
- $E_2 = (F_{sales}; \{D_{products}; D_{times}; D_{customers}\}; [Year(T_{now}) - 14; Year(T_{now}) - 4[)$;
- $E_3 = (F_{sales}; \{D_{products}; D_{times}\}; [Year(1990); Year(T_{now}) - 14[)$;

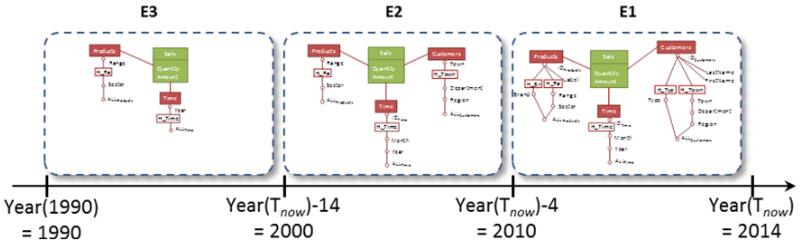

FIG. 4: Principe de la réduction d'un EDM.

L'état E_1, appelé état courant est associé à un intervalle de validité [Year(T_{now})-4 ; Year(T_{now})[correspondant à [2010 ; 2014[. Les instances de l'état correspondent uniquement aux ventes réalisées entre 2010 et 2014 conformément à la dimension D_{times}. De la même manière, l'état E_2 conserve toutes les données relatives aux ventes réalisées entre 2000 et 2010 tandis que l'état E_3 conserve les données relatives aux ventes réalisées avant 2000.

L'année 1990 est la date (instant fixe) de création de l'EDM. On remarquera le caractère dynamique (glissant au cours du temps) des intervalles définis par les instants Year(T_{now})-14, Year(T_{now})-4 et Year(T_{now}). Ainsi, l'année prochaine, Year(T_{now}) = 2015, Year(T_{now})-4 = 2011 et Year(T_{now})-14 = 2001. Au changement d'année, la mise à jour des instances contenues dans les états E_1, E_2 et E_3 est réalisée.

Définition 3. Un fait $F_i, \forall i \in$ [1..x], est défini par (n^{Fi}, M^{Fi}) où

- $n^{Fi} \in \mathcal{N}$ est le nom du fait ;
- $M^{Fi} = \{m_1, ..., m_{pi}\}$ est un ensemble de mesures (ou indicateurs).

Définition 4. Une dimension $D_i, \forall i \in [1..m]$, est définie par (n^{Di}, A^{Di}, H^{Di}), où

- $n^{Di} \in \mathcal{N}$ est le nom de la dimension ;
- $A^{Di} = \{a_1^{D_i}, ..., a_{r_1}^{D_i}\}$ est l'ensemble d'attributs de la dimension ;

– $H^{Di} = \{H_1^{Di}, ..., H_{h_1}^{Di}\}$ est l'ensemble de hiérarchies de la dimension.

Les hiérarchies organisent les attributs d'une dimension de la granularité d'analyse la plus fine (paramètre initial, ID_{Di}) vers la granularité la plus générale (paramètre final noté ALL_{Di}). Ainsi, une hiérarchie définit un schéma de navigation valide sur un axe d'analyse.

Définition 5. Une hiérarchie H_j (notation abusive de $H_j^{D_i}, \forall i \in [1..m], \forall j \in [1..h_i]$) est définie par $(n^{Hj}, P^{Hj}, \prec^{Hj}, Weak^{Hj})$, où :
– $n^{Hj} \in \mathcal{N}$ est le nom de la hiérarchie ;
– $P^{Hj} = \{P_1^{Hi}, ..., P_{q1}^{Hj}\}$ est un ensemble d'attributs appelés paramètres, $P^{Hj} \subseteq A^{Di}$;
– $\prec^{Hj} = \{(p_x^{Hj}, p_y^{Hj}) | p_x^{Hj} \in P^{Hj} \wedge p_y^{Hj} \in P^{Hj}\}$ est une relation binaire transitive antisymétrique entre paramètres. La propriété antisymétrique signifie que $(p_{k1}^{Hj} \prec^{Hj} p_{k2}^{Hj}) \wedge (p_{k2}^{Hj} \prec^{Hj} p_{k1}^{Hj}) \Rightarrow p_{k1}^{Hj} = p_{k2}^{Hj}$ tandis que la transitivité signifie que $(p_{k1}^{Hj} \prec^{Hj} p_{k2}^{Hj}) \wedge (p_{k2}^{Hj} \prec^{Hj} p_{k3}^{Hj}) \Rightarrow p_{k1}^{Hj} \prec^{Hj} p_{k3}^{Hj}$.
– $Weak^{Hj} : P^{Hj} \rightarrow 2^{A^{Di} \setminus P^{Hj}}$ est une application qui associe chaque paramètre à un ensemble d'attributs de la dimension appelé attributs faibles.

Exemple. L'état E3 est composé d'un fait et de deux dimensions, son intervalle temporel correspond à une période datée de 1990 à 2000. Le fait SALES d'E3 contient une seule mesure : AMOUNT. La dimension temporelle, intitulée D_{times}, est uniquement graduée par les paramètres YEAR et ALL_{times}. La dimension $D_{products}$ contient une hiérarchie intitulée H_Ra. Les paramètres sur cette hiérarchie sont organisés de la granularité la plus fine *Range* à la granularité la plus générale $ALL_{products}$. La représentation abstraite de E3 est la suivante :

$E_3 = (F_{sales}; \{D_{products}; D_{times}\}; [t_{1990}; t_{2000}[)$ où :

– F_{sales} = (SALES ; { Amount }) ;
– $D_{products}$ = ($P_{products}$; Range, Sector, $ALL_{products}$; H_Ra) ; ;
– D_{times} = (TIMES ; Year, ALL_{times} ; H_Time).

La hiérarchie H_Ra est définie par $(n^{H\_Ra}, P^{H\_Ra}, \prec^{H\_Ra}, Weak^{H\_Ra})$ où :
– Le nom de la hiérarchie n^{H_Ra} = H_Ra ;
– Les paramètres sur cette hiérarchie $P^{H_{Ra}}$ = Range, Sector, $ALL_{products}$;
– La relation binaire transitive antisymétrique entre paramètres $\prec^{H\_Ra}$ = (Range, Sector) ; (Sector, $ALL_{products}$) ;
– Les attributs faibles sur cette hiérarchie $Weak^{H\_Ra} = \emptyset$.

3.3 Opérateurs de réduction

La dérivation d'un schéma réduit E_{k+1} à partir du schéma d'un état E_k est effectuée grâce à la combinaison d'opérateurs de dérivation. Nous définissons l'ensemble de ces opérateurs O = $Slice^{reduce}; RollUp^{reduce}; Drop^{reduce}$ comme le noyau minimum d'opérateurs élémentaires pour définir la dérivation. Deux catégories d'opérateurs sont proposées :
– les opérateurs de réduction des instances (Tab. 1) permettant de réduire les domaines de valeurs des dimensions en maintenant les structures du schéma dérivé et
– les opérateurs de réduction du schéma (Tab. 2) effectuant des suppressions d'attributs (paramètres, attributs faibles et mesures).

L'opérateur $Slice^{reduce}$ permet d'obtenir un état réduit dans lequel le domaine de définition de la dimension spécifiée est réduit aux seules instances qui satisfont le prédicat $pred_{slice}$.

Operateurs	
$Slice^{reduce}(E_k; D_{slice}; pred_{slice}; T_{k+1}) = E_{k+1}$	
Entrées	E_k = (F_k ; $\mathcal{D}_k$; T_k) : état initial ; $D_{slice} \in \mathcal{D}_k$ est la dimension à réduire ; $pred_{slice}$ est un prédicat de restriction de D_{slice}, noté $dom(D_{slice})$.
Sortie	$E_{k+1} = (F_{k+1}; D_{k+1}; T_{k+1})$ est un état réduit tel que - $F_{k+1} = F_k$ - $D_{k+1} = D_k$ avec $dom(D_{slice}) = \{v_i \in dom(D_{slice}) \mid pred_{slice}(v_i)$ = TRUE $\}$.

TAB. 1: Opérateur de réduction d'instances.

L'opérateur $RollUp^{reduce}$ permet d'obtenir un nouvel état dans lequel la dimension spécifiée est réduite en supprimant tous les attributs inférieurs au paramètre de forage indiqué dans l'opérateur. Si le paramètre spécifié est le paramètre extrémité $ALL_{D_{rollup}}$, la dimension est alors intégralement supprimée de l'état réduit obtenu. L'opérateur $Drop^{reduce}$ permet d'obtenir un nouvel état dans lequel le fait est réduit par la suppression de la mesure spécifiée.

Operateurs	
$RollUp^{reduce}(E_k; D_{rollup}; p_{rollup}; T_{k+1}) = E_{k+1}$	
Entrées	E_k = (F_k ; $\mathcal{D}_k$; T_k) : état initial ; $D_{rollup} \in \mathcal{D}_k$: dimension dédiée à la réduction ; $p_{rollup} \in A^{Drollup}$: paramètre de réduction de la dimension D_{rollup}.
Sortie	$E_{k+1} = (F_{k+1}; \mathcal{D}_{k+1}; T_{k+1})$ est un état réduit tel que - $F_{k+1} = F_k$ - $\mathcal{D}_{k+1} = \mathcal{D}_k \setminus \{D_{rollup}\} \cup \{D_{new}\}^{(*)}$ with $D_{new} = (n^{Dnew}; A^{Dnew}; H^{Dnew})$ - $n^{Dnew} = n^{Dold}$ - $A^{Dnew} = \{a_x \in A^{Drollup} \mid a_x = p_{rollup} \vee \forall H_j \in H^{Drollup}, p_{rollup} \prec^{Hj} a_x\}$ - $H^{Dnew} = \{H_x \in H^{Drollup} \mid$ - $n^{Hx} = n^{Hj} \wedge$ - $P^{Hx} = \{p_y \in P^{Hj} \mid p_y = p_{rollup} \vee p_{rollup} \prec^{Hj} p_y\} \wedge \prec^{Hx} = \{(p^{Hj}_{x1}, p^{Hj}_{x2}) \in \prec^{Hj} \mid p^{Hj}_{x1} = p_{rollup} \in p_{rollup} \prec^{Hj} p^{Hj}_{x1}\} \wedge$ - $Weak^{Hx} := \{(p_{x1}, A^{Hx}_{x1}) \in Weak^{Hj} \mid p_y \in P^{Hj}\}$.
$Drop^{reduce}(E_k; m_{drop}; T_{k+1}) = E_{k+1}$	
Entrées	E_k = (F_k ; $\mathcal{D}_k$; T_k) : état initial ; $m_{drop} \in M_k$ est une mesure de F_k.
Sortie	$E_{k+1} = (F_{k+1}; \mathcal{D}_{k+1}; T_{k+1})$ est un état réduit tel que - $F_{k+1} = (n^{Fk}, M^{Fk} \setminus \{m_{drop}\})$; - $\mathcal{D}_{k+1} = \mathcal{D}_k$.

(*) Si $A^{Dnew} = \{ALL_{D_{rollup}}\}$ alors $\mathcal{D}_{k+1} = \mathcal{D}_k \setminus \{D_{rollup}\}$

TAB. 2: Opérateurs de réduction sur le schéma.

Exemple : Dans l'exemple précédent, les états réduits sont obtenus à partir du processus de dérivation formellement défini comme suit :
– $RollUp^{reduce}$ ($RollUp^{reduce}$ (E_1; $D_{products}$; P_{range} ; [Year(T_{now})-14 ; Year(T_{now})-4[) ; $D_{customers}$; P_{town} ; [Year(T_{now})-14 ; Year(T_{now})-4[) = E_2 ;
– $RollUp^{reduce}$($RollUp^{reduce}$($Drop^{reduce}$(E_2 ;Quantity ; [Year(1990) ; Year(T_{now})-14[) ; $D_{customers}$; $ALL_{customers}$; [Year(1990) ; Year(T_{now})-14[) ; D_{times} ; P_{year} ; [Year(1990) ; Year(T_{now})-14[) = E_3.

4 Expérimentations

4.1 Données

Il existe plusieurs bancs d'essais pour les BD R-OLAP, tel que TPC-DS [1]. En s'appuyant sur une seule BD ROLAP prédéfinie, ces bancs d'essais exécutent un ensemble de requêtes afin de mesurer la performance des machines hébergeant un SGBD Darmont et al. (2007). Dans le cadre de nos expérimentations dont le but est de démontrer l'efficacité du schéma réduit et non pas une machine particulière, un ensemble de schémas multidimensionnels multi-états serait plus adéquat. Etant donné que les bancs d'essais de TPC ne permettent pas d'évaluer les impacts de différentes modélisations d'EDM sur un système donné Darmont et al. (2007), nous avons décidé de proposer notre propre environnement d'essais en utilisant des données synthétiques qui s'avèrent plus pertinentes pour la démonstration.

Afin d'expérimenter notre solution, nous avons deux types d'implantation de l'EDM défini précédemment avec le SGBD Oracle.

Le premier type d'implantation consiste en la définition d'une BD non réduite, appelée *Global Star*. Cette BD contient toutes les données implantées sous un format R-OLAP dénormalisé (chaque fait et chaque dimension est implanté avec une seule table). Le schéma de cette BD est le suivant :
– *Sales* (Quantity, Amount, IDProducts#, IDTime#, IDCustomers#)
– *Customers* (IDCustomers, Lastname, Firstname, Town, Department, Region, Type)
– *Products* (IDProducts, Range, Sector, Brand)
– *Times* (IDTime, Month, Year)
La population de la BD Global Star a été effectuée comme suit :
– La dimension *Times* contient toutes les dates de 1990 à 2013.
– Les deux autres dimensions contiennent des données aléatoires définies par génération de données synthétiques. Pour éviter tout biais, l'affectation des données aléatoires a été faite de telle manière que les attributs père d'une hiérarchie n'ont pas tous le même nombre de fils tout en respectant les contraintes d'intégrité des hiérarchies strictes (tout attribut fils d'une hiérarchie possède un et seul attribut père).
Nous avons également proposé plusieurs versions de la BD non réduite en faisant varier le nombre de tuples des tables *Customers* et *Products*, autrement dit |*Customers*| et |*Products*| varient de 10 à 40 tuples.

1. http://www.tpc.org/tpcds/

– |*Customers*| = 10, 20, 30, 40 tuples
– |*Products*| = 10, 20, 30, 40 tuples
– |*Times*| = 8401 tuples (du premier Janvier 1990 au 31 Décembre 2013)
– |*Sales*| = |*Customers*| x |*Products*| x |*Times*| = 840 100 to 13 441 600 tuples.

Dans le tableau ci-dessous, sont décrites les différentes valeurs associées aux attributs des tables contenant des données variables :

|*Customers*| x |*Products*|	Contenu de la table *Customers*	Contenu de la table *Products*
10*10	2 *Towns*, 2 *Departments* , 1 *Region*, 2 *Types*	2 *Ranges*, 2 *Sectors*, 2 *Brands*
20*20	4 *Towns*, 3 *Departments*, 2 *Regions*, 4 *Types*	4 *Ranges*, 3 *Sectors*, 4 *Brands*
30*30	6 *Towns*, 4 *Departments*, 2 *Regions*, 6 *Types*	6 *Ranges*, 4 *Sectors*, 6 *Brands*
40*40	8 *Towns*, 5 *Departments*, 3 *Regions*, 8 *Types*	8 *Ranges*, 5 *Sectors*, 8 *Brands*

TAB. 3: Détail d'implantation des versions de Global Star.

Le second type d'implantation, appelé *Reduced Star*, contient les 3 états définis précédemment. Le schéma relationnel de ces 3 états est décrit dans le tableau suivant.

Etat E1	*Sales* (Quantity, Amount, IDProducts#, IDTime#, IDCustomers#) *Customers* (IDCustomers, Lastname, Firstname, Town, Department, Region, Type) *Products* (IDProducts, Range, Sector, Brand) *Times* (IDTime, Month, Year)
Etat E2	*Sales* (Quantity, Amount, Range#, IDTime#, Town#) *Customers* (Town, Department, Region) *Products* (Range, Sector) *Times* (IDTime, Month, Year)
Etat E3	*Sales* (Amount, Range#, Year#) *Products* (Range, Sector) *Times* (Year)

TAB. 4: Schémas R-OLAP des 3 états.

La génération des données de la BD *Reduced Star* a consisté à répartir les données en fonction des critères définis en figure 4.

|Customers| x|Products|	BD R-OLAP Global Star	BD R-OLAP Reduced Star		
		E_1	E_2	E_3
10x10	|SALES|=840,100	|SALES|=109,600	|SALES|=32,877	|SALES|=30
20x20	|SALES|=3,360,400	|SALES|=438,400	|SALES|=91,325	|SALES|=50
30x30	|SALES|=7,560,900	|SALES|=986,400	|SALES|=178,997	|SALES|=70
40x40	|SALES|=13,441,600	|SALES|=1,753,600	|SALES|=32,877	|SALES|=90

TAB. 5: Nombre de tuples pour les BD Global Star et Reduced Star.

4.2 Protocole

Notre expérimentation consiste à comparer les résultats obtenus lors de l'application de requêtes sur les BD non réduites et réduites. Cette expérimentation tient compte de 3 critères :

- Volumétrie de la BD : comme indiqué précédemment, nous allons appliquer des requêtes sur 4 versions de BD pour les deux types de BD.
- Typologie des requêtes
  - Requêtes ne contenant que des jointures et pas de critères de sélection sur les dimensions non temporelles lors de la manipulation d'états (manipulation de toutes des données des états de la BD réduite)
  - Requêtes contenant des conditions de restrictions sur les données (manipulation de certaines données de certains états)
  - Requêtes écrites avec différents types de groupement SQL (Group By, Group By Cube, Group By Roll Up).
- Portée des requêtes
  - requêtes manipulant une ou plusieurs tables de dimensions
  - requêtes intervenant sur 1, 2 ou 3 états.

4.3 Requêtes avec jointures : résultats et discussion

La table ci-dessous liste les requêtes implantées en SQL pour notre expérimentation.

	Requêtes	Etats	Dimensions
Q1	Montant des ventes pour les 3 dernières années	E1	Time
Q2	Montant et quantité des ventes pour 2008	E2	Time
Q3	Montant annuel des ventes réalisées avant 2000	E3	Time
Q4	Montant des ventes par villes de 2010 à 2012	E1	Time, Customers
Q5	Montant des ventes mensuelles par départements de 2000 à 2005	E2	Time, Customers
Q6	Montant des ventes annuelles par secteurs avant 2000	E3	Time, Products
Q7	Montant des ventes par villes, secteurs et mois en 2012	E1	Time, Products, Customers
Q8	Montant annuel des ventes par secteurs et départements de 2000 à 2005	E2	Time, Products, Customers
Q9	Montant des ventes mensuelles depuis 2000	E1 ; E2	Time
Q10	Montant des ventes annuelles par villes de 2002 à 2012	E1 ; E2	Time, Customers
Q11	Montant des ventes par année et catégorie de 1990 à 2009	E2 ; E3	Time, Products
Q12	Montant des ventes par villes et secteurs de 2002 à 2012	E1 ; E2	Time, Products, Customers
Q13	Montant des ventes annuelles	E1 ; E2 ; E3	Time
Q14	Montant des ventes annuelles par catégorie	E1 ; E2 ; E3	Time, Products

TAB. 6: Requêtes sans restriction de données

Remarque. Il est impossible de définir une requête manipulant 3 états avec 3 dimensions car l'état E3 est composé de seulement 2 dimensions.

Le premier test compare les coûts d'exécution théorique (fournis par « Explain Plan » du SGBD Oracle) en faisant varier la taille de l'EDM selon le protocole défini précédemment. Le coût d'exécution n'a pas d'unité particulière, ce n'est qu'un indice pondéré qui sert à mesurer différents coûts de plans d'exécution.

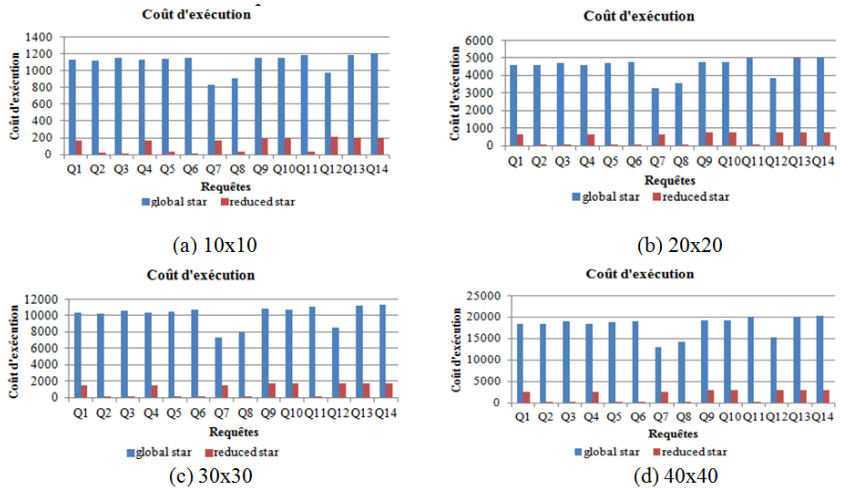

FIG. 5: Coût d'exécution des requêtes Q1 à Q14.

A la première lecture, quelle que soit la volumétrie des BD, les coûts d'exécution des requêtes dans un environnement réduit (en rouge dans la figure ci-dessus) sont plus faibles que dans un environnement non réduit (en bleu dans la figure ci-dessus). Dans chacune des versions, la moyenne des gains se situent de 89,23% en taille 10*10 à 90,51% en taille 40*40.

Dans la figure ci-dessous, nous analysons les cardinalités des résultats afin de vérifier si la volumétrie des résultats est liée aux temps d'exécution des requêtes. Même si, nous obtenons des volumes très différents, les proportions restent similaires d'une version de BD à une autre. Comme nous pouvons le voir dans la figure ci-dessous,

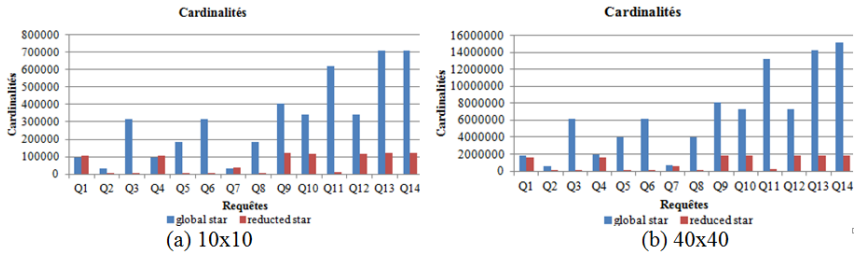

FIG. 6: Cardinalités des résultats pour Q1 à Q14.

– Quelle que soit la volumétrie de la BD le gain en temps d'exécution est pratiquement constant et important (de l'ordre de 90%) ; *cf* . courbe rouge ;
– Il faut également noter que malgré l'augmentation importante de la volumétrie de la table de fait (courbe verte dans le graphique ci-dessous) induisant une légère augmentation de la cardinalité du résultat des requêtes (courbe bleue), le gain relatif au temps d'exécution des requêtes reste quasi stable au fur à mesure que l'on augmente la volumétrie en passant de 89,23% pour une BD de taille 10*10 à 90,51% pour une BD de taille 40*40.

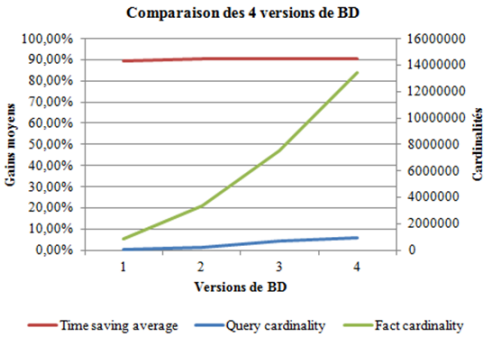

FIG. 7: Comparaison des 4 versions de BD.

Nous focalisons maintenant notre étude sur la version de BD la plus volumineuse (40*40) pour analyser les gains liés aux temps d'exécution entre la BD non réduite et la BD réduite. Comme indiqué par la droite linéaire estimant les gains, nous pouvons remarquer que plus le nombre de tables manipulées augmente, plus le gain est important tout en restant à des proportions similaires. Néanmoins, en affinant notre étude, nous pouvons remarquer que plus le nombre d'états manipulés par une requête est important, plus le gain faiblit mais restent dans le même ordre de grandeur :

– pour les requêtes manipulant un seul état (Q1 à Q8), les gains variant de 80,99% (Q7) à 99,98% (Q3) pour un gain moyen de 93,43%
– pour les requêtes manipulant deux états (Q9 à Q12), les gains variant de 84,99% (Q9) à 98,41% (Q11) pour un gain moyen de 87,18%
– pour les requêtes manipulant trois états (Q13, Q14), les gains variant de 85,45% (Q13) à 85,59% (Q14) pour un gain moyen de 85,52%

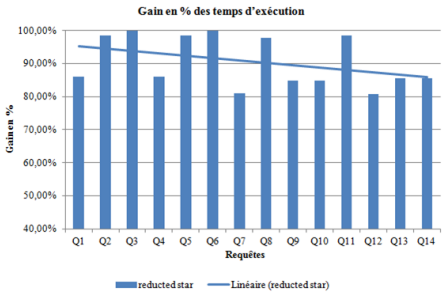

FIG. 8: Gain en % des temps d'exécution entre Global Star et Reduced Star

4.4 Requêtes avec sélection : résultats et discussion

Ce second test vise à analyser l'impact des conditions de sélection sur le temps d'exécution. Etant donné que la volumétrie de la BD n'influence que légèrement le gain de temps, nous avons effectué ces tests uniquement sur une BD 40*40. Vous trouverez ci-dessous les requêtes retenues pour ce test.

	Requêtes	Etats	Dimensions
Q1	Montant des ventes pour un client X de 2010 à 2012	E1	Customers
Q2	Montant des ventes pour une Ville Y de 2010 à 2012	E1	Customers
Q3	Montant des ventes pour un département Z de 2010 à 2012	E1	Customers
Q4	Montant des ventes mensuelles pour les produits de la catégorie X pour la ville Z depuis 2000	E1, E2	Products, Customers
Q5	Montant des ventes mensuelles pour les produits d'un secteur W dans une ville X depuis 2000	E1, E2	Products, Customers
Q6	Montant des ventes mensuelles pour tous les produits vendus dans une ville X depuis 2000	E1, E2	Products, Customers
Q7	Montant des ventes annuelles des produits de la catégorie X	E1 ; E2 ; E3	Products
Q8	Montant des ventes annuelles des produits de la catégorie Y (3fois moins nombreux que la catégorie X)	E1 ; E2 ; E3	Products
Q9	Montant des ventes annuelles pour le secteur X	E1 ; E2 ; E3	Products

TAB. 7: Requêtes avec restriction de données

Contrairement à nos attentes, les gains de performances entre une BD non réduite et une BD réduite restent dans les mêmes proportions que l'on applique ou pas des conditions de sélection sur les requêtes. En effet, pour des requêtes contenant des sélection, ce gain varie de 78,01% (requête Q5) à 88,03% (requête Q1) avec une moyenne de 85,31% alors que lors des tests avec les requêtes ne proposant pas de sélection le gain moyen était de 90,51%. De plus, quelle que soit la portée de la condition de sélection (attribut clé primaire, attribut contenant beaucoup ou peu de valeurs différentes), l'écart type n'est pas très important (0.08).

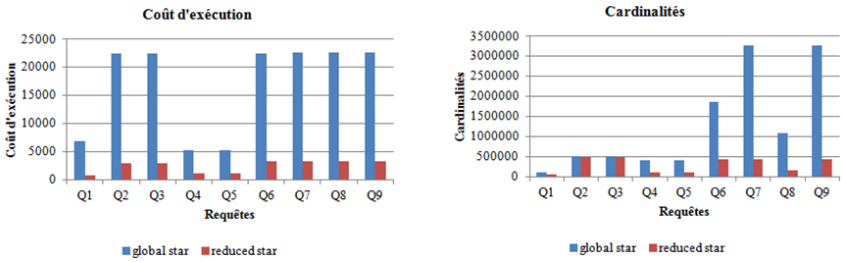

FIG. 9: Coûts d'exécution et cardinalités des requêtes Q1 to Q9.

De plus, si on compare les résultats des requêtes Q7 et Q8 dans la figure ci-dessus, nous pouvons remarquer que même si la cardinalité du résultat de la requête Q7 est trois fois supérieure

à celle de la requête Q8, le coût d'exécution des deux requêtes est identique. Cela est dû au fait que quel que soit le nombre de lignes sélectionnées par la requête, le SGBD est obligé de parcourir tous les tuples des tables afin d'obtenir le résultat de la requête.

4.5 Requêtes avec différents « group by » : résultats et discussion

Dans cette troisième série de tests, nous avons étudié l'impact des fonctions de groupement proposées par Oracle.

	Requêtes	Etats	Dimensions
Q1	Montant des ventes mensuelles pour les 3 dernières années (group by)	E1	Time
Q2	Montant des ventes mensuelles pour les 3 dernières années (group by rollup)	E1	Time
Q3	Montant des ventes annuelles et mensuelles par villes de 2010 à 2012 (group by)	E1	Time, Customers
Q4	Montant des ventes annuelles et mensuelles par villes de 2010 à 2012 (group by rollup)	E1	Time, Customers
Q5	Montant des ventes annuelles et mensuelles par villes de 2010 à 2012 (group by Cube)	E1	Time, Customers
Q6	Montant des ventes mensuelles par ville, catégorie et secteurs pour les 3 dernières années (group by)	E1	Time, Products, Customers
Q7	Montant des ventes mensuelles par ville, catégorie et secteurs pour les 3 dernières années (group by rollup)	E1	Time, Products, Customers
Q8	Montant des ventes mensuelles par ville, catégorie et secteurs pour les 3 dernières années (group by Cube).	E1	Time, Products, Customers
Q9	Montant des ventes annuelles en fonction des villes et secteurs de 2002 à 2012 (group by)	E1, E2	Time, Products, Customers
Q10	Montant des ventes annuelles en fonction des villes et secteurs de 2002 à 2012 (group by rollup)	E1, E2	Time, Products, Customers
Q11	Montant des ventes annuelles en fonction des villes et secteurs de 2002 à 2012 (group by cube)	E1, E2	Time, Products, Customers
Q12	Montant des ventes annuelles par catégorie et par secteur (group by)	E1, E2, E3	Time, Products,
Q13	Montant des ventes annuelles par catégorie et par secteur (group by rollup)	E1, E2, E3	Time, Products,
Q14	Montant des ventes annuelles par catégorie et par secteur (group by cube)	E1, E2, E3	Time, Products
Q15	Montant des ventes annuelles par catégorie et par secteur (partial group by cube)	E1, E2, E3	Time, Products,

TAB. 8: Requêtes reposant sur différents regroupements

Nous avons donc testé le groupement classique (Group By) avec des groupements contenant des calculs de sous-totaux (Group By, RollUp et Group By Cube). Avant d'effectuer ces tests, notre intution était que les temps d'éxécution entre ces deux modes d'agrégation allaient être différents et modifieraient les gains. Or, les temps d'exécution sont pratiquement similaires pour les groupes de requêtes similaires avec des fonctions d'agrégations différentes.

Le groupe 1 contient Q1 et Q2, le groupe 2 contient Q3, Q4 et Q5, le groupe 4 contient les requêtes Q6, Q7, Q8, le groupe 5 contient les requêtes Q9 à Q11 et le groupe 6 contient les requêtes Q12 à Q15. Quel que soit le groupe, le gain moyen est toujours du même ordre 86% et il n'y a pas de différence dans les temps d'exécution des requêtes qu'elles contiennent ou pas des calculs de sous-totaux.

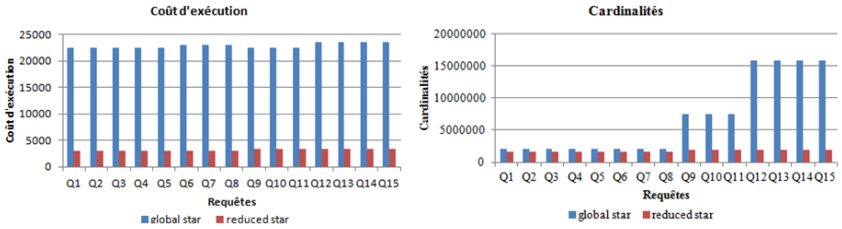

FIG. 10: Coûts d'exécution et Cardinalité des requêtes Q1 à Q15

5 Conclusion

Cet article se situe dans le contexte des EDM. Plus précisément, nous apportions une solution pour la réduction des données décisionnelles. Dans un premier temps, nous avons défini un modèle conceptuel reposant sur des états. Chaque état est représenté avec un schéma multidimensionnel, un intervalle temporel de validité et une fonction de dérivation pour faire le lien des états. Pour définir ces fonctions de dérivation, nous avons défini noyau minimum d'opérateurs élémentaires $\{RollUp^{reduce}; Drop^{reduce}; Slice^{reduce}\}$ pour effectuer aussi bien de la réduction de schémas que d'instances.

Dans un second temps, nous avons expérimenté notre solution dans un environnement R-OLAP. Nous avons montré qu'en faisant varier le nombre de tuples de la table de fait (de 840,100 to 13,441,600 tuples), le gain moyen entre une BD non réduite et un EDM réduit était de 90% et restait quasi stable en augmentant le volume des données. Pour l'EDM le plus grand, ce gain variait de 93,43% (manipulation d'un seul état) à 85,52% (manipulation de 3 états). De plus, nous avons démontré que le temps d'exécution n'était pas lié à la cardinalité du résultat. Lors de l'application de requêtes contenant des conditions de sélection spécifique, le gain reste important avec un gain de plus de 85%. Enfin, nous avons remarqué que l'application des requêtes avec les différents types de groupement n'influençait pas les temps d'exécution des requêtes d'analyses décisionnelles.

Une des perspectives à ce travail consiste à proposer un outil graphique permettant l'interrogation d'une BD R-OLAP multi-états. Cette extension nécessite la définition d'une algèbre et

d'un langage graphique applicable à un EDM multi-états et des solutions de représentation des résultats d'analyses multi-états.

Références

Atigui, F. (2013). *Approche dirigée par les modèles pour l'implantation et la réduction d'entrepôts de données*. Ph. D. thesis, Toulouse 1.

Chen, Y., G. Dong, J. Han, J. Pei, B. W. Wah, et J. Wang (2002). Online analytical processing stream data : Is it feasible ? In *DMKD*. Citeseer.

Darmont, J., F. Bentayeb, et O. Boussaid (2007). Benchmarking data warehouses. *International Journal of Business Intelligence and Data Mining 2*(1), 79–104.

Garcia-Molina, H., W. Labio, et J. Yang (1998). Expiring Data in a Warehouse. In *Proceedings of 24th International Conference on Very Large Data Bases, VLDB*, New York City, New York, USA, pp. 500–511.

Golfarelli, M. et S. Rizzi (1998). Methodological Framework for Data Warehouse Design. In *Proceedings of the 1st International Workshop on Data warehousing and OLAP, DOLAP*, Bethesda, Maryland, USA, pp. 3–9.

Iftikhar, N. et T. B. Pedersen (2010). Using a Time Granularity Table for Gradual Granular Data Aggregation. In *Proceedings of the 14th East European Conference Advances in Databases and Information Systems, ADBIS*, Novi Sad, Serbia, pp. 219–233.

Iftikhar, N. et T. B. Pedersen (2011). A rule-based tool for gradual granular data aggregation. In *Proceedings of the 14th International Workshop on Data Warehousing and OLAP, DOLAP*, Glasgow, United Kingdom, pp. 1–8.

Okun, O. et H. Priisalu (2007). Unsupervised data reduction. *Signal Processing 87*(9), 2260–2267.

Skyt, J., C. S. Jensen, et T. B. Pedersen (2008). Specification-based data reduction in dimensional data warehouses. *Information Systems Journal 33*(1), 36–63.

Udo, Ifiok, J. et B. Afolabi (2011). Hybrid Data Reduction Technique for Classification of Transaction Data. *Journal of Computer Science and Engineering 6*(2), 12–16.

Summary

Our aim is to elaborate a multidimensional database reduction process which will specify aggregated schema applicable over a period of time as well as retain useful data for decision support. Firstly, we describe a multidimensional database schema composed of a set of states. Each state is defined as a star schema composed of one fact and its related dimensions. We introduce associated reduction operators to define reduced schema. Secondly, we describe our experiments and the associated results. Evaluating our solution implies executing different requests in various contexts: relational star schema without reductions and reduced star schema. We show that queries are more efficiently executed in a reduced star schema.