

Impact de l'évolution de la nomenclature des membres de dimension sur l'entrepôt de données

Inès ZOUARI TURKI*, Faïza GHOZZI JEDIDI**, Rafik BOUAZIZ***

Laboratoire MIRACL,
Faculté des Sciences Économiques et de Gestion de Sfax,
Route de l'Aérodrome km 4.5, B.P. 1088 – 3018 Sfax, Tunisie.

*Ines.Zouari@isimsf.rnu.tn

**Jedidi_Faiza@yahoo.fr

***Raf.Bouaziz@fsegs.rnu.tn

Résumé. Les systèmes courants de gestion d'entrepôts de données (**Data Warehouses** : DW) permettent de traiter les évolutions des faits, mais pas les évolutions affectant les schémas des DW et les instances de dimension. Plusieurs solutions, basées essentiellement sur l'historisation et/ou le versionnement des éléments de DW, ont été proposées pour ces dernières. Néanmoins, peu sont les travaux qui ont traité des évolutions des nomenclatures des instances de dimension et de leurs effets sur les analyses. Dans cet article, nous proposons une classification des différents types d'évolution de nomenclature, et étudions les effets de ces évolutions sur la cohérence des analyses. Les solutions que nous envisageons à chacun de ces aspects d'évolution se situent dans le cadre d'un système de gestion de DW temporel multiversions.

1 Introduction

Un entrepôt de données, mieux connu sous le vocable anglais *Data Warehouse* (DW), est un outil précieux pour la préparation de l'information de synthèse, nécessaire au processus de prise de décision. La structuration d'un DW conformément au modèle multidimensionnel de données est constituée par un fait et un certain nombre de dimensions. Un fait comporte des mesures dont les valeurs dépendent du contexte établi par les dimensions, qui regroupent les paramètres pertinents des différentes entités d'analyse. Une dimension est souvent organisée selon une ou plusieurs hiérarchies dont chacune comporte un certain nombre de niveaux. Une hiérarchie spécifie la façon selon laquelle les mesures peuvent être agrégées. Les instances des niveaux sont appelées **membres de dimensions** (MemD). La modélisation d'un DW dépend du nombre de niveaux par dimension. Elle donne soit un modèle en étoile lorsque chaque dimension présente un seul niveau, soit un modèle en flocons de neige, lorsque certaines dimensions présentent des niveaux hiérarchisés. Par ailleurs, le modèle en constellation est d'un grand intérêt pour structurer un DW comportant plusieurs tables de fait. Il permet de mettre en relief le partage des dimensions par ces tables.

Sachant que les systèmes classiques de gestion de DW sont conçus pour traiter l'évolution des données de transaction, ils s'approprient mal à prendre en considération les changements qui peuvent affecter les dimensions, aussi bien au niveau de leurs données qu'au niveau de leurs structures. Ceci est dû à l'hypothèse de l'orthogonalité de la dimension

Impact de l'évolution de nomenclature sur les DW

Temps par rapport aux autres dimensions, imposant l'invariance de ces dimensions dans le temps (Eder et Koncilia., 2001). Mais afin de pouvoir s'adapter aux changements des besoins des utilisateurs ou aux modifications des sources de données, les constituants d'un DW doivent pouvoir évoluer au cours du temps. Face à cette problématique, plusieurs solutions ont été proposées pour les différents cas d'évolution affectant la structure du DW. De même, plusieurs aspects d'évolution des instances des dimensions de DW ont été étudiés, tels que l'insertion d'un nouveau MemD, la reclassification d'un MemD dans la hiérarchie d'une dimension, etc. Néanmoins, peu sont les travaux qui ont traité de l'évolution de la nomenclature des MemD et de son influence sur les données du DW ainsi que sur les analyses.

Nous nous proposons dans cet article d'étudier les différents aspects d'évolution relevant de la modification de nomenclature des MemD. Nous distinguons ici entre les aspects simples, caractérisés par la simple modification de la valeur de la clé de MemD, et celles complexes combinant ce type de modification avec des évolutions du niveau schéma ou du niveau instance. Par ailleurs, nous recensons les problèmes et les effets de ces différents aspects sur les données du DW et sur les analyses appliquées au DW. Face à ces problèmes, nous proposons des solutions appropriées, qui constituent une partie d'une solution généralisée pour le versionnement des schémas de DW, dont les principes de base ont été présentés dans (Zouari et Bouaziz, 2008). Il est à noter que nous ne traitons que des évolutions de nomenclatures qui risquent de perturber les analyses. D'autres évolutions, telles que l'extension ou la réduction de la taille d'un attribut clé dont les valeurs restent les mêmes, sont possibles mais ne sont pas considérées dans cet article, car elles ne perturbent pas les analyses.

La suite de cet article est organisée comme suit. Nous présentons dans la section suivante les différents types d'évolution des constituants d'un DW et les travaux de la littérature traitant de ces types. Ensuite, nous détaillons dans la section 3 les différents aspects d'évolution de nomenclature et leurs effets sur les DW, en se basant sur un cas réel. Nous proposons alors dans la section 4 les solutions que nous envisageons pour résoudre les problèmes d'évolution de nomenclature. La section 5 présente nos conclusions et perspectives.

2 Etat de l'art

Pour pouvoir s'adapter aux changements des besoins des utilisateurs ou aux modifications des sources de données, les constituants d'un DW doivent pouvoir évoluer.

2.1 Différents types d'évolution dans un DW

Les évolutions pouvant affecter les constituants d'un DW peuvent être classifiées en deux catégories : les évolutions du niveau schéma et les évolutions du niveau instance (Eder et al., 2002a). Parmi les évolutions du niveau schéma d'un DW, on trouve l'insertion et la suppression d'une dimension, d'un niveau de hiérarchie de dimension et d'une mesure, ainsi que le changement de la structure hiérarchique d'une dimension (Favre et al., 2007). Les évolutions du niveau instance peuvent concerner les données de transaction (formules de calcul de faits, unités de faits) et les données de dimension (Eder et al., 2002a). Dans le cadre de ce dernier type, on trouve l'insertion et la suppression d'un MemD, le changement de la

valeur de la clé d'un MemD, le regroupement de MemD tout en restant dans le même niveau hiérarchique, la re-classification de MemD dans la hiérarchie d'une dimension, la subdivision et la fusion de MemD et la modification de valeurs d'attributs faibles de MemD. Body et al. (2003) classifient les opérations subies par les MemD en opérations simples (il s'agit des six opérations ci-dessus évoquées) et en opérations complexes. Parmi ces dernières, on distingue la diminution (composition de la subdivision et de la suppression), l'augmentation (composition de l'insertion et de la fusion) et l'annexion partielle (composition de la subdivision et de la fusion).

2.2 Travaux existants

On distingue dans la littérature plusieurs approches traitant des évolutions des spécifications des DW. Ces approches peuvent être classées en deux catégories ; celles supportant le changement structurel de DW et celles supportant l'historisation et le versionnement de DW.

- Les approches de la première catégorie se limitent à assurer l'évolution des schémas et/ou des instances de DW, sans assurer leur historisation. Elles ne permettent de garder, à un instant donné, qu'un seul schéma de DW (le plus récent). Dans ce contexte, Blaschka et al. (1999) ont défini quatorze opérations d'évolution de schéma de dimensions et de faits ainsi que leurs impacts sur le DW, sans évoquer l'évolution au niveau des membres de dimension. Cet aspect a été traité par Hurtado et al. (1999) qui proposent un modèle formel supportant la mise à jour des dimensions à travers des opérations d'évolution simples, touchant aussi bien le schéma que les instances des dimensions, et d'autres complexes, affectant les MemD.

- Les approches de la deuxième catégorie assurent la persistance des états historiques du DW. Elles peuvent être subdivisées en deux courants : le premier concerne l'historisation des membres de dimension et/ou des éléments de schéma de DW, et le deuxième adopte le versionnement des schémas de DW.

- Parmi les travaux du premier courant, nous citons ceux de Bliujute et al. (1998) qui proposent un schéma en étoile temporel où on remplace la dimension *Temps* par l'estampillage des membres de dimension et des instances de fait. Ceci favorise l'historisation des valeurs des attributs des membres, mais pas celles des attributs clés. C'est aussi le cas du système multidimensionnel temporel proposé par Mendelzon et Vaisman (2000), qui assure l'estampillage des instances et des schémas des dimensions, tout en gardant la dimension *Temps*.

- Quant aux travaux du deuxième courant, ils sont multiples. Nous nous limitons ici à en présenter un extrait. Eder et Koncilia (2001) proposent un système multidimensionnel temporel basé sur le versionnement du DW entier et l'historisation des MemD. Ils considèrent de la même manière l'évolution des attributs faibles et celle des attributs clés de MemD. Néanmoins, la distinction entre ces deux types d'évolution a été assurée dans une extension de ce système appelée le modèle COMET (Eder et al., 2002a), (Eder et al., 2002b). Pour assurer la conversion de données entre les versions de DW, ce modèle se limite à utiliser des fonctions de transformation basées sur des facteurs poids. Ce principe de transformation est aussi adopté dans la solution de Body et al. (2003) qui proposent de versionner les MemD, en utilisant une table de fait multiversions pour rassembler les données des différentes versions de DW, sans évoquer les problèmes d'évolution des clés des MemD. Bebel et al. (2004) jugent que l'on ne peut pas se restreindre à une table de fait multiversions pour résoudre tous les problèmes de versionnement. Il faut plutôt définir un

data warehouse multiversions (DWMV). Un modèle formel pour ce DWMV a été proposé dans (Bebel et al., 2006), en assurant à la fois le versionnement du schéma du DW et le versionnement des instances des dimensions et des faits. Cet article propose une description formelle d'un ensemble d'opérateurs supportant l'évolution de schéma et l'évolution de MemD. Néanmoins, aucun de ces opérateurs ne traite de la modification de MemD.

De notre côté, nous avons développé dans Ellouze et al. (2006) et dans Zouari et Bouaziz (2007) des solutions ad hoc aux problèmes d'évolution de schéma et d'instances recensés à partir de deux applications types. Ces solutions sont réalisables sur les systèmes de gestion de DW non temporels tels que *Microsoft Analysis Services*. Puis, dans Zouari et Bouaziz (2008), nous avons recensé les différents types d'évolution pouvant affecter le schéma et les instances d'un DW, et nous avons défini de nouveaux types d'évolution, principalement les évolutions mixtes touchant à la fois le schéma et les instances. Nous avons aussi arrêté les principes de base d'une solution généralisée pour le versionnement des schémas des DW.

2.3 Problématiques et positionnement

Peu sont donc les travaux qui ont traité des problèmes d'évolution de nomenclature de MemD. D'autre part, certains travaux ont considéré de la même manière l'évolution d'un attribut faible et celle d'un attribut clé de MemD, à l'exception de Eder et al. (2002a). En se basant sur un exemple typique (le Zaïre qui a été renommé le Congo), ces auteurs ont proposé de résoudre le problème de changement de la valeur de la clé d'un MemD en appliquant une fonction de transformation utilisant un facteur poids, égal à 1, entre les deux versions de DW obtenues suite à cette évolution. Cependant, la couverture des problèmes d'évolution de nomenclature de MemD, pouvant être engendrés dans la pratique, reste limitée. En effet, il existe des cas complexes où cette simple évolution des valeurs de clé se trouve combinée à d'autres types d'évolution d'instances ou de schémas.

La problématique traitée dans cet article consiste d'abord à voir s'il y a d'autres cas d'évolution de valeurs de clés de MemD, qui peuvent être des cas simples ou des cas complexes. Ces derniers peuvent résulter de la combinaison des évolutions de valeurs de clés à d'autres types d'évolution d'instances ou de schémas. Il s'agit ensuite d'identifier les perturbations que ces cas d'évolution risquent d'entraîner. Quels sont alors les risques engendrés et comment définir les solutions qui permettent de les écarter ? C'est dans ce cadre que se situe notre contribution. L'apport de cet article consiste essentiellement à définir de nouvelles opérations d'évolution de schémas et/ou d'instances, d'une part, et à fournir des solutions permettant d'assurer des évolutions saines, écartant tout type de perturbations des analyses, suite à toute modification de clé de MemD, d'autre part.

Pour recenser les cas d'évolution de valeurs de clés de MemD, nous présentons dans la section suivante un cas réel de DW ayant subi de tels cas, et nous discutons leurs effets.

3 Effets de changement de nomenclature sur les DW

Dans cette section, nous nous attachons à décrire les différents aspects d'évolution de nomenclature dans un DW, ainsi que leurs effets sur la cohérence des données du DW et sur les analyses. Pour illustrer ces différents aspects, nous nous référons à un cas réel issu de l'activité commerciale de la Société Nationale d'Exploitation et de Distribution des Eaux en Tunisie (SONEDE).

3.1 Exemple motivant

Considérons le DW couvrant l'activité commerciale de la SONEDE. Nous nous limitons dans cet article à la présentation d'un cube de ce DW, intitulée *Gestion Ventes Eaux*. Ce cube (cf. FIG. 1) permet d'analyser la mesure *QuantitéEauVendue* selon les dimensions *Temps*, *Branchement*, *Usage Eau* et *Catégorie Abonné*. La dimension *Temps* présente la hiérarchie suivante : *Année* ← *Trimestre* ← *Mois*, où le symbole "←" signifie "agrégé en". Les niveaux de la dimension *Branchement* sont organisés comme suit : *Direction Régionale* ← *District* ← *Centre Exploitation* ← *Branchement*. L'attribut clé *CléBranchement*, du niveau *Branchement*, est constitué par la concaténation de la clé du centre d'exploitation, du numéro de la tournée, représentant un sous ensemble du centre d'exploitation et correspondant à une journée de travail d'un releveur, et d'un numéro d'ordre relativement à cette tournée. La dimension *Usage Eau* présente la hiérarchie suivante : *ClasseUsage* ← *Usage*.

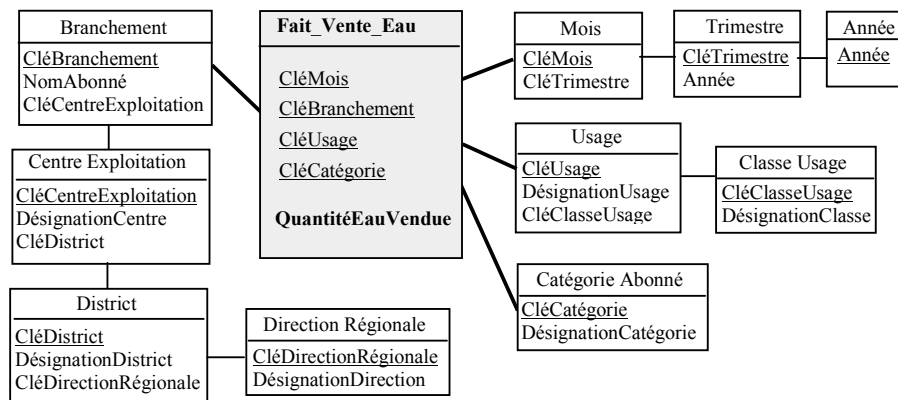


FIG. 1 – Schéma du cube *Gestion Ventes Eaux*

3.2 Effets de changements de nomenclature

Dans ce paragraphe, nous nous intéressons principalement à deux cas d'évolution qui ont affecté les deux dimensions *Usage* et *Branchement* du DW SONEDE. Il s'agit de la refonte des usages qui a eu lieu en mars 2001 et de la refonte des districts qui a eu lieu en mars 2003.

3.2.1 Modification de la valeur de la clé d'un MemD

Suite à la modification de la valeur de la clé d'un MemD, celui-ci se trouve avec deux valeurs identifiantes différentes : $Val_1_Clé$ et $Val_2_Clé$, tel que $Val_1_Clé$ l'identifie jusqu'à la date de l'évolution et $Val_2_Clé$ l'identifie à partir de cette date. Considérons comme exemple pour ce cas d'évolution, l'usage « 17 : *Autres* » qui est devenu, suite à la refonte des usages, « 19 : *Autres* », tout en restant dans la même classe usage « 10 : *Industrie* » (cf. FIG. 2). Suite à cette évolution, une requête du genre « Quelle est la quantité d'eau vendue pour l'usage *Autres* de code 17 durant toute l'année 2001 ? » risque d'avoir un résultat incorrect car le code usage 17 n'est relatif à l'usage *Autres* que jusqu'à février 2001.

3.2.2 Réaffectation d'une valeur de la clé d'un MemD

Il s'agit de réaffecter l'ancienne valeur de la clé d'un MemD, ayant subi une modification de valeur de clé ou ayant été supprimé, à un autre MemD. On se retrouve alors, suite à cette évolution, avec une valeur de clé identifiant deux MemD différents. Les analyses de données qui se basent sur cette clé, pour une période incluant la date de l'évolution T, auront des résultats erronés. En effet, pour la période qui précède T, ces analyses utilisent, pour une même valeur de clé, les données relatives à un MemD M, alors que pour la période qui suit T, elles utilisent les données d'un autre MemD M' différent de M. Tel est le cas, par exemple, de la valeur clé 17 qui était relative à l'usage *Autres* jusqu'à février 2001, alors qu'elle concerne l'usage *Activité Pétrolière* à partir de Mars 2001 (cf. FIG. 2).

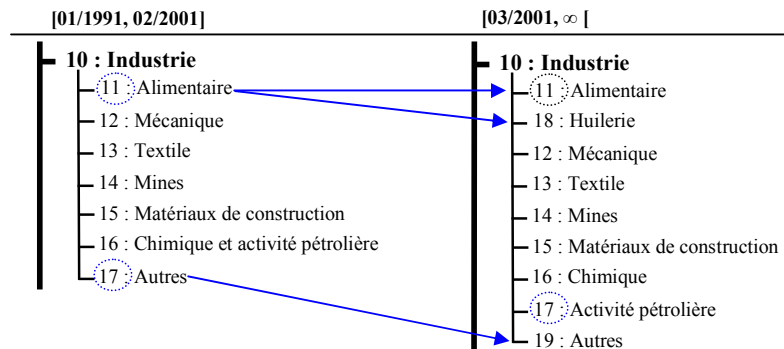


FIG. 2 – Evolution de la classe usage « 10 : Industrie »

3.2.3 Subdivision perturbatrice de MemD

Définition 1. Une subdivision perturbatrice est une subdivision d'un MemD M en n MemD (M_1, \dots, M_n) accompagnée de l'attribution de la valeur de la clé de M à l'un des MemD obtenus ($M_i, 1 \leq i \leq n$).

La valeur clé de M devient alors, suite à cette évolution, relative à un nouveau MemD, constituant tout de même une partie de l'ancien. Tel est le cas de l'évolution du MemD « 11 : Alimentaire » qui a subi, en mars 2001, une subdivision en deux nouveaux usages « 11 : Alimentaire » et « 18 : Huilerie » (cf. FIG. 2). Pour cet exemple, La quantité d'eau vendue relative au code usage 11 durant 2001, aura un résultat incorrect, puisqu'il lui manquera la quantité relative à l'usage *Huilerie* pour la période mars - décembre 2001 si l'on se rapportera à l'ancienne sémantique de la nomenclature, ou, au contraire, la quantité relative aux huileries durant janvier - février 2001 sera superflue (comptée en trop) si l'on se rapportera à la nouvelle sémantique.

3.2.4 Fusion perturbatrice de MemD

Définition 2. Une fusion perturbatrice est une fusion de n MemD (M_1, \dots, M_n) en un MemD M, accompagnée de l'attribution de la valeur de la clé de l'un des MemD fusionnés ($M_i, 1 \leq i \leq n$) à M.

C'est l'inverse du cas d'évolution précédent. Suite à cette évolution, les données relatives à la valeur clé concernée ne correspondent plus uniquement au membre M_i , mais plutôt au membre M qui regroupe tous les membres ayant participé à la fusion. Un exemple de notre cas d'étude consiste en la fusion des deux catégories d'abonnés « 03 : *Promoteur de construction* » et « 10 : *Entreprise* », donnant la catégorie « 03 : *Affaires* ».

3.2.5 Reclassification perturbatrice de MemD

Définition 3. Une reclassification perturbatrice est une reclassification d'un MemD M qui fait passer M du niveau hiérarchique X au niveau hiérarchique Y d'une même dimension (Y pouvant être supérieur ou inférieur à X), et qui est accompagnée d'une modification de la valeur de la clé de M .

Notre cas d'étude comporte un tel exemple d'évolution, suivi d'une subdivision. Il s'agit de l'évolution de l'usage « 31 : *Commerce* », appartenant auparavant à la classe usage « 30 : *Collectif* ». Cet usage a été reclassifié pour devenir une classe usage et a reçu la clé 40. Il a été aussi subdivisé en sept nouveaux usages : « 41 : Commerces courants, 42 : Exposition commerciale, 43 : Restauration, ... ». Cette évolution aura certainement un impact important sur les analyses de données, car, à partir de mars 2001, on ne peut pas suivre les consommations d'eau relatives à l'usage *Commerce* en se basant sur sa clé 31 sachant que cette dernière devient, suite à cette évolution relative, à un nouvel usage « *Bouche d'incendie* », alors que *Commerce* devient une classe d'usage ayant 40 comme valeur clé.

3.2.6 Suppression perturbatrice d'un niveau hiérarchique

Définition 4. Une suppression perturbatrice d'un niveau hiérarchique N_i (... $N_{i-1} \leftarrow N_i \leftarrow N_{i+1}$...) est une opération qui supprime N_i , rattache le niveau hiérarchique N_{i+1} au niveau hiérarchique N_{i-1} , et modifie les valeurs des clés des MemD du niveau N_{i+1} .

La refonte de la dimension *Branchement* constitue un exemple de ce cas d'évolution. Il s'agit de la suppression du niveau *Centre Exploitation*, du rattachement direct des MemD du niveau *Branchement* aux MemD du niveau *District*, et de l'affectation de nouvelles valeurs clés aux branchements relatifs au district de *Sfax* (subdivisé en trois districts) (cf. FIG. 3). En effet, la refonte a consisté en un nouveau découpage de la population des abonnés en de nouvelles tournées, et en une affectation de nouveaux numéros d'ordre. La valeur clé de tout branchement correspond, depuis, à la concaténation de la valeur de la clé de son district (et non plus de son centre d'exploitation), du numéro de sa tournée et de son numéro d'ordre. Par exemple, l'identifiant du branchement *C35_178_005* est devenu *D32_251_015*. Un problème se pose donc pour l'agrégation des consommations des abonnés lorsque la période désirée contient l'instant de la refonte t_{ref} , en tant qu'un point différent de ses bornes.

4 Solutions proposées

Nos solutions pour les problèmes d'évolution de nomenclature se situent dans le cadre du système de gestion de data warehouse multiversions que nous avons défini dans (Zouari et Bouaziz, 2008). Les principes de base de ce système sont rappelés dans la section suivante. Afin de pouvoir formaliser ces solutions, nous développons dans cette même section une expression formelle des constituants de notre système.

Impact de l'évolution de nomenclature sur les DW

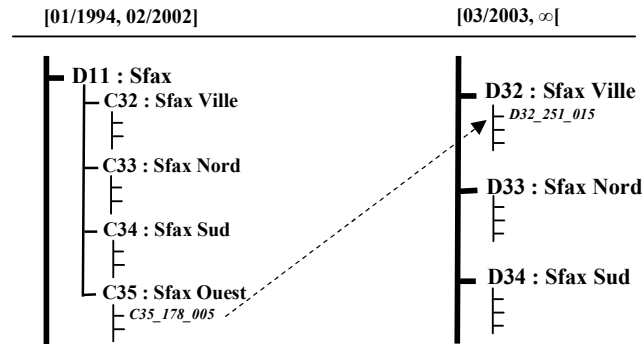


FIG. 3 – Evolution du district « D11 : Sfax »

4.1 Système de gestion de data warehouse multiversions (DWMV)

Ce système se base sur le versionnement des différents éléments à historiser dans le DW : les cubes, les dimensions, les niveaux et les MemD. Chacune des versions de ces éléments est caractérisée par un numéro et un intervalle de temps d'application [TDA, TFA], où TDA constitue le temps de début d'application de cette version et TFA constitue le temps de fin d'application. Par ailleurs, notre système permet, en appliquant un estampillage par les temps d'application, d'historiser les attributs faibles, les mesures, ainsi que les relations entre niveaux et celles entre MemD. Les DW sont alors versionnés à travers leurs constituants ; on ne leur tient que des versions virtuelles : la version de DW appliquée à un instant donné est constituée par les versions réelles en application à cet instant de chacun de ses éléments.

Afin de pouvoir analyser les données relatives à une période donnée conformément à une version d'un DW, notre système permet de réaliser des transformations de données entre les différentes versions de ce DW, et ce en définissant des fonctions de transformation que nous classifions en deux types : celles qui se basent sur des formules mathématiques, à travers l'application de facteurs poids, d'une manière similaire à l'approche de Eder et Koncilia (2001), et celles qui se basent sur des règles de calcul spécifiques, exprimées par des requêtes de transformation. Une telle requête utilise une table de correspondance qui fournit pour chaque MemD relatif à une ancienne version son correspondant dans la nouvelle version.

Le schéma de notre système de gestion de DWMV est défini par :

1. L'ensemble \mathbb{DW} de DW, $\mathbb{DW} = \{DW_1, \dots, DW_m\}$ où chaque DW DW_i est défini comme suit : $DW_i = \langle ID_DW_i, Nom_DW_i, \zeta_i \rangle$. ID_DW_i et Nom_DW_i représentent l'identifiant unique et le nom du DW. ζ_i est l'ensemble des cubes appartenant à DW_i . ζ_i est un sous ensemble de \mathbb{C} , ci-dessous défini.
2. L'ensemble \mathbb{C} de cubes, $\mathbb{C} = \{C_1, \dots, C_n\}$, où chaque cube C_j est défini comme suit : $C_j = \langle ID_Cub_j, Nom_Cub_j, VC_Cub_j, \mathcal{V}\mathcal{H}_Cub_j \rangle$. ID_Cub_j et Nom_Cub_j représentent l'identifiant et le nom du cube C_j . VC_Cub_j est la version courante (en application) de ce cube. Elle est définie par $VC_Cub_j = \langle ID_Cub_j, Vers_Cub_j, \mathcal{D}_j, Mes_j, ValFait_j, TDA_j \rangle$, où $Vers_Cub_j$ et TDA_j sont le numéro de la version courante et sa date de début d'application. $\mathcal{D}_j = \{D_{j1}, \dots, D_{jc}\}$ est l'ensemble des dimensions constituant cette version, et $Mes_j = \{Mes_{j1}, \dots, Mes_{jd}\}$ est l'ensemble de ses mesures. $ValFait_j$ est la fonction de valorisation des occurrences de fait ; elle associe à chaque combinaison de MemD relatifs

au plus bas niveau des hiérarchies des dimensions de \mathcal{D}_j , une valeur pour chaque mesure de Mes_j . Elle est définie comme suit :

$$\begin{aligned} ValFait_j : \mathcal{M}_1, \dots, \mathcal{M}_c &\rightarrow \text{dom}(Mes_1), \dots, \text{dom}(Mes_d) \\ (Mem_1, \dots, Mem_c) &\mapsto Val_1, \dots, Val_d \end{aligned}$$

\mathcal{M}_i est l'ensemble des versions en application des MemD concernés. $\text{Dom}(Mes_i)$ est le domaine de la mesure Mes_i . $\mathcal{VH_Cub}_j$ est l'ensemble des versions historisées du cube C_j . Une version historisée de ce cube admet, en plus des éléments définissant une version courante, le TFA indiquant sa date de fin d'application. Il en est de même pour toutes les versions historisées des autres éléments versionnés de chaque DW.

3. L'ensemble \mathbb{D} de dimensions, $\mathbb{D} = \{D_1, \dots, D_o\}$ où chaque dimension D_k est définie comme suit : $D_k = \langle ID_Dim_k, Nom_Dim_k, VC_Dim_k, \mathcal{VH_Dim}_k \rangle$. ID_Dim_k et Nom_Dim_k sont respectivement l'identifiant de la dimension D_k et son nom. VC_Dim_k est la version courante de cette dimension, alors que $\mathcal{VH_Dim}_k$ est l'ensemble de ses versions historisées. $VC_Dim_k = \langle ID_Dim_k, Vers_Dim_k, \mathcal{N}_k, \mathcal{HN}_k, TDA_k \rangle$, où $Vers_Dim_k$ et TDA_k représentent le numéro de la version courante et sa date de début d'application. \mathcal{N}_k est l'ensemble des niveaux appartenant à cette version de dimension. \mathcal{HN}_k est l'ensemble des hiérarchies selon lesquelles ces niveaux sont organisés. \mathcal{N}_k et \mathcal{HN}_k sont respectivement inclus dans \mathbb{N} et \mathbb{HN} , ci-après présentés.
4. L'ensemble \mathbb{N} de niveaux de hiérarchie, $\mathbb{N} = \{N_1, \dots, N_p\}$ où chaque niveau N_e est défini comme suit : $N_e = \langle ID_Niv_e, Nom_Niv_e, VC_Niv_e, \mathcal{VH_Niv}_e \rangle$. ID_Niv_e et Nom_Niv_e représentent l'identifiant du niveau N_e et son nom. VC_Niv_e est la version courante de ce niveau, alors que $\mathcal{VH_Niv}_e$ est l'ensemble de ses versions historisées. $VC_Niv_e = \langle ID_Niv_e, Vers_Niv_e, \mathcal{A}_e, TDA_e \rangle$, où $Vers_Niv_e$ et TDA_e représentent le numéro de la version courante du niveau et sa date de début d'application. \mathcal{A}_e est l'ensemble des attributs relatifs à cette version de niveau. \mathcal{A}_e est un sous ensemble de \mathbb{A} ci-dessous défini.
5. L'ensemble \mathbb{HN} de relations hiérarchiques entre niveaux, $\mathbb{HN} = \{HN_1, \dots, HN_g\}$ où chaque relation HN_h est définie comme suit : $HN_h = \langle ID_HN_h, Dim_h, Ordre_Niv_h, TDA_h, TFA_h \rangle$. ID_HN_h , Dim_h et $[TDA_h, TFA_h]$ sont respectivement l'identifiant de HN_h , sa dimension et l'intervalle durant lequel elle est en application dans le DW concerné. $Ordre_Niv_h$ est un ensemble de tuples qui décrit les niveaux appartenant à la hiérarchie HN_h ainsi que leur ordre. $Ordre_Niv_h$ est défini comme suit : $Ordre_Niv_h = \{(ID_HN_h, Niv_i, ordre_i), 1 \leq i \leq N\}$. Niv_i , défini par $(N.ID_i, N.Vers_i)$, est une version de niveau appartenant à la hiérarchie HN_h et admettant l'ordre $ordre_i$ dans cette hiérarchie.
6. L'ensemble \mathbb{A} d'attributs, $\mathbb{A} = \{A_1, \dots, A_r\}$ où chaque attribut A_b est défini comme suit : $A_b = \langle ID_Att_b, Nom_Att_b, Type_b, TDA_b, TFA_b \rangle$. ID_Att_b , Nom_Att_b et $Type_b$ sont respectivement l'identifiant de l'attribut, son nom et son type. $[TDA_b, TFA_b]$ est l'intervalle de temps d'application de l'attribut A_b .
7. L'ensemble \mathbb{Mes} de mesures, $\mathbb{Mes} = \{Mes_1, \dots, Mes_s\}$ où chaque mesure Mes_m est définie comme suit : $Mes_m = \langle ID_Mes_m, Nom_Mes_m, TDA_m, TFA_m \rangle$. ID_Mes_m et Nom_Mes_m représentent l'identifiant de la mesure Mes_m et son nom. $[TDA_m, TFA_m]$ est l'intervalle de temps d'application de cette mesure.

Les instances de notre système de gestion de DWMV sont définies par :

1. L'ensemble \mathbb{Mem} de membres de dimension, $\mathbb{Mem} = \{Mem_1, \dots, Mem_g\}$ où chaque membre Mem_g est défini comme suit : $Mem_g = \langle ID_Mem_g, ID_Niv_g, VC_Mem_g, \mathcal{VH_Mem}_g \rangle$. ID_Mem_g est l'identifiant de ce membre, ID_Niv_g est l'identifiant du niveau

Impact de l'évolution de nomenclature sur les DW

auquel ce MemD est attribué. VC_Mem_g est la version courante de ce MemD, alors que $\mathcal{V}H_Mem_g$ est l'ensemble de ses versions historisées. $VC_Mem_g = \langle ID_Mem_g, Clé_g, Vers_Mem_g, Niv_g, ValA_g, TDA_g \rangle$, où $Clé_g$, $Vers_Mem_g$ et TDA_g constituent respectivement la clé de ce MemD sous sa version courante, le numéro de cette version et sa date de début d'application. Niv_g , égal à $(Niv.ID_g, Niv.Vers_g)$, est la version de niveau à laquelle VC_Mem_g est relative. $ValA_g$ est l'ensemble de n-uplets constituant les valeurs des attributs relatives à la version courante du membre Mem_g : $ValA_g = \{(A_g^i.ID, val)\}$, où $A_g^i.ID$ est l'identifiant de l'attribut considéré et val est sa valeur.

2. L'ensemble HM de relations de hiérarchie entre membres, $HM = \{HM_1, \dots, HM_u\}$ où chaque relation de hiérarchie HM_a est définie comme suit : $HM_a = \langle ID_HM_a, Mem_d^f, Mem_d^p, TDA_a, TFA_a \rangle$. ID_HM_a est l'identifiant de la relation hiérarchique. Mem_d^f , égal à $(Mem_d^f.ID, Mem_d^f.Vers)$, représente une version de membre. Mem_d^p , égal à $(Mem_d^p, Mem_d^p.Vers)$, est le membre père de Mem_d^f . Mem_d^p est valorisé à \emptyset si Mem_d^f est relatif au niveau le plus haut des hiérarchies de la dimension concernée. $[TDA_a, TFA_a]$ est l'intervalle durant lequel cette relation est en application dans le DW considéré.

Notre système de gestion de DWMV englobe aussi l'ensemble F de fonctions de transformation, $F = \{F_1, \dots, F_e\}$ assurant la transformation entre les versions de MemD. Une fonction F_i est définie comme suit :

$$F_i : \mathcal{V}Mem_D \times \mathcal{V}Mem_D \rightarrow [0, 1]$$

$$(VM_i, VM_j) \mapsto \omega$$

$\mathcal{V}Mem_D$ est l'ensemble des versions des MemD d'une dimension D . ω est la valeur associée du facteur poids. Nous remarquons qu'il faut aussi définir les fonctions inverses F_i^{-1} .

4.2 Solutions aux problèmes d'évolution de nomenclature

4.2.1 Actions générales d'historisation

Proposition 1.

- La création d'une nouvelle version d'un MemD ou d'une nouvelle relation hiérarchique s'accompagne de l'affectation de l'instant de sa mise en application à son TDA. Cet instant est déterminé par l'administrateur du DW considéré. Par défaut, il est égal à l'instant courant de l'horloge du système.

- L'historisation d'une version d'un MemD ou d'une relation hiérarchique, qui doit cesser d'être en application, s'accompagne par l'affectation à son TFA de l'instant $T - Q$, où :

- T est l'instant de la décision de fin d'application. Il est déterminé par l'administrateur du DW considéré. Par défaut, il est égal à l'instant courant de l'horloge du système.

- Q représente la granularité la plus fine prise pour la dimension *Temps*.

Proposition 2.

- La création d'une nouvelle version V_{i+1} d'un MemD M (V_{i+1}_M) s'accompagne de la création d'une nouvelle relation hiérarchique reliant V_{i+1}_M au MemD père M^p , s'il existe, sous sa version courante V_k ($V_k_M^p$). Elle s'accompagne également de la création d'une nouvelle relation hiérarchique reliant V_{i+1}_M à chacun de ses MemD fils M^f , s'il y en a, sous sa version courante V_e ($V_e_M^f$). Il s'agit, le cas échéant, du même père et des mêmes fils de l'ancienne version V_i_M .

- L'historisation d'une version V_i d'un MemD M (V_i_M) s'accompagne de l'historisation

de la relation hiérarchique reliant V_i_M à son MemD père M^p , s'il existe, sous sa version courante V_k ($V_k_M^p$). Elle s'accompagne également de l'historisation de chaque relation hiérarchique reliant V_i_M à chacun de ses MemD fils M^f , s'il y en a, sous sa version courante V_e ($V_e_M^f$).

4.2.2 Solution proposée pour la modification de la valeur clé d'un MemD

Proposition 3. Soit M un MemD ayant comme version courante V_i et subissant une modification de la valeur de sa clé ($Val_2_Clé$ au lieu de $Val_1_Clé$). Pour écarter toute incohérence d'analyse suite à un tel cas d'évolution, il faut :

- Créer pour M une nouvelle version V_{i+1} comportant la nouvelle valeur clé $Val_2_Clé$.
- Historiser l'ancienne version courante V_i de M (V_i_M).
- Appliquer les actions définies dans *Proposition 1* et *Proposition 2*.
- Définir une fonction de transformation F entre V_i_M et V_{i+1_M} , ainsi que la fonction inverse F^{-1} , comme suit : $F(V_i_M, V_{i+1_M}) = 1$ et $F^{-1}(V_{i+1_M}, V_i_M) = 1$.

4.2.3 Solution proposée pour la réaffectation de clé de MemD

Proposition 4. Suite à la réaffectation d'une ancienne valeur de la clé d'un MemD M (V_i_M) à un nouveau MemD, il faut :

- Créer un nouveau MemD M' , sous sa première version (V_1_M'), comportant la clé réaffectée.
- Appliquer les actions définies dans *Proposition 1* et *Proposition 2*, en remplaçant V_{i+1_M} par V_1_M' .
- Définir une fonction de transformation F entre V_i_M et V_1_M' , ainsi que la fonction inverse F^{-1} , comme suit : $F(V_i_M, V_1_M') = 0$ et $F^{-1}(V_1_M', V_i_M) = 0$. Ceci signifie qu'il n'y a en fait aucune correspondance entre les deux versions de MemD V_i_M et V_1_M' .

L'application des propositions 3 et 4 à l'exemple des sections 3.2.1 et 3.2.2 donne le résultat présenté en figure 4.

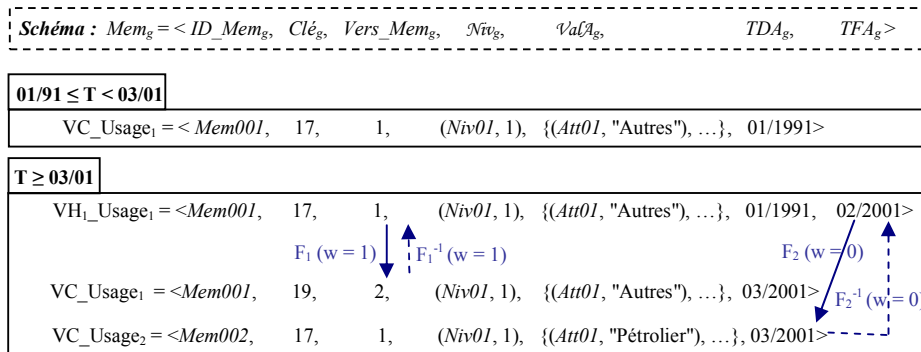


FIG. 4 – Illustration de notre solution pour une évolution saine suite à la modification de la valeur clé d'un MemD et à la réaffectation de cette valeur à un autre MemD

4.2.4 Solution proposée pour la subdivision perturbatrice de MemD

Proposition 5. Suite à la subdivision d'un MemD M , ayant comme version courante V_i (V_i_M), en n MemD M_j , $1 \leq j \leq n$, et l'attribution de la valeur clé de M à l'un des MemD résultats (M_k), il faut :

- Créer une nouvelle version de M (V_{i+1_M}) correspondant à M_k .
- Créer les membres M_j , $1 \leq j \leq n$ et $j \neq k$, sous leur première version ($V_1_M_j$).
- Historiser l'ancienne version courante V_i de M (V_i_M).
- Appliquer les actions définies dans *Proposition 1* et *Proposition 2*.
- Définir une fonction de transformation F_k entre V_i_M et V_{i+1_M} , ainsi que la fonction inverse F_k^{-1} , comme suit : $F_k(V_i_M, V_{i+1_M}) = \omega_k$ et $F_k^{-1}(V_{i+1_M}, V_i_M) = 1$. Le facteur poids $\omega_k \in [0, 1]$, est égal à la proportion de V_i_M qui doit être affectée à V_{i+1_M} .
- Définir de la même manière les fonctions F_j , $1 \leq j \leq n$ et $j \neq k$, entre V_i_M et $V_1_M_j$, ainsi que les fonctions inverses F_j^{-1} .

4.2.5 Solution proposée pour la fusion perturbatrice de MemD

Proposition 6. Afin de traiter la fusion de n MemD ($V_r_M_1, \dots, V_y_M_n$) en un MemD M (r, \dots, y représentent les versions utilisées par ces n MemD, respectivement) et l'attribution de la valeur de la clé de l'un des MemD fusionnés ($V_r_M_k$) à M , il faut :

- Créer M comme une nouvelle version V_{t+1} du MemD M_k ($V_{t+1_M_k}$).
- Historiser les versions courantes V_s des MemD M_i ($V_s_M_i$), $1 \leq i \leq n$ et $s \in \{r, \dots, y\}$.
- Appliquer les actions définies dans *Proposition 1* et *Proposition 2*.
- Définir des fonctions de transformation F_i , $1 \leq i \leq n$, ainsi que les fonction inverses F_i^{-1} , $1 \leq i \leq n$, comme suit : $F_i(V_s_M_i, V_{t+1_M_k}) = 1$ et $F_i^{-1}(V_{t+1_M_k}, V_s_M_i) = \omega_i$. Le facteur poids $\omega_i \in [0, 1]$ traduit la proportion de $V_{t+1_M_k}$ qui doit être affectée à $V_s_M_i$.

4.2.6 Solution proposée pour la reclassification perturbatrice de MemD

Proposition 7. Suite à la reclassification d'un MemD M , sous la version V_i , d'un niveau N vers un autre niveau N' de la même dimension et la modification de sa valeur clé ($Val_2_Clé$ au lieu de $Val_1_Clé$), il faut :

- Créer un nouveau MemD M' , sous sa première version (V_1_M'), appartenant au niveau N' et comportant la nouvelle valeur clé $Val_2_Clé$.
- Historiser l'ancienne version courante V_i du MemD M (V_i_M).
- Appliquer les actions définies dans *Proposition 1* et *Proposition 2*.
- Créer une fonction de transformation F entre V_i_M et V_1_M' , ainsi que la fonction inverse F^{-1} , comme suit : $F(V_i_M, V_1_M') = 1$ et $F^{-1}(V_1_M', V_i_M) = 1$.
- Dans le cas où M'_V_1 admet des MemD fils ($V_1_M_k$, $1 \leq k \leq n$), il faut définir des fonctions de transformation F_k , $1 \leq k \leq n$, entre V_i_M et $V_1_M_k$, ainsi que les fonctions inverses F_k^{-1} , comme suit : $F_k(V_i_M, V_1_M_k) = \omega_k$ et $F_k^{-1}(V_1_M_k, V_i_M) = 1$. ω_k est égal à la proportion de V_i_M qui doit être affectée à $V_1_M_k$.

4.2.7 Solution proposée pour la suppression perturbatrice d'un niveau hiérarchique

Proposition 7. Suite à la suppression d'un niveau hiérarchique N_i , au rattachement des MemD M_f du niveau fils N_{i-1} aux MemD M_p du niveau père N_{i+1} et à la modification des

valeurs des clés des MemD M_f , il faut :

- Créer une nouvelle version pour chaque MemD M_f , chacune comportant la nouvelle clé qui lui correspond.
- Historiser les anciennes versions courantes des MemD M_f .
- Appliquer les actions définies dans *Proposition 1* et *Proposition 2*.
- Créer une table de correspondance entre l'ancienne clé et la nouvelle de tout MemD M_f .
- Créer, sur la base de cette table et de la table de faits, une requête de transformation permettant de déterminer (sous la forme d'une vue), pour une période donnée, la valeur d'un fait relatif à un MemD ayant subi l'évolution de suppression perturbatrice.

4.3 Prototype

Nous avons envisagé une validation par prototypage en deux phases :

- La première phase, déjà réalisée, a consisté en la résolution des problèmes d'évolution de nomenclature dans le cadre d'une solution ad hoc assurant le versionnement du DW de la SONEDE (cf. §3.1), développée sous la version 8.0.1.94 de *Microsoft SQL Server 2000 Analysis Services* (Ellouze et al., 2006), (Zouari et Bouaziz, 2007). A titre d'exemple, le cube *Gestion Ventes Eaux* a subi des évolutions touchant les dimensions *Usage* (comportant 38 enregistrements) et *Branchement* (comportant 29.103 enregistrements), avec des modifications de nomenclature. Il s'agit de tous les cas présentés au §3.2. Notre solution ad hoc a permis d'assurer l'exploitation de la table de fait *Fait_Vente_Eau* de ce cube, avec deux versions pour la dimension *Usage* et deux versions pour la dimension *Branchement*, dans des conditions tout à fait satisfaisantes : les résultats fournis sont corrects et les temps de réponse sont acceptables, sachant que l'expérimentation a été réalisée avec 191.506 enregistrements de *Fait_Vente_Eau*, correspondant à trois années. Les figures 5 et 6 montrent un extrait des données du cube *Gestion Ventes Eaux* suite à la subdivision perturbatrice qui a affecté l'usage « 16 : Chimique et activités pétrolières ».
- La deuxième phase, en cours de réalisation, vise le développement d'un prototype basé sur la solution généralisée que nous avons définie pour le versionnement des constituants des DW, y compris les MemD ayant subi des évolutions de nomenclature.

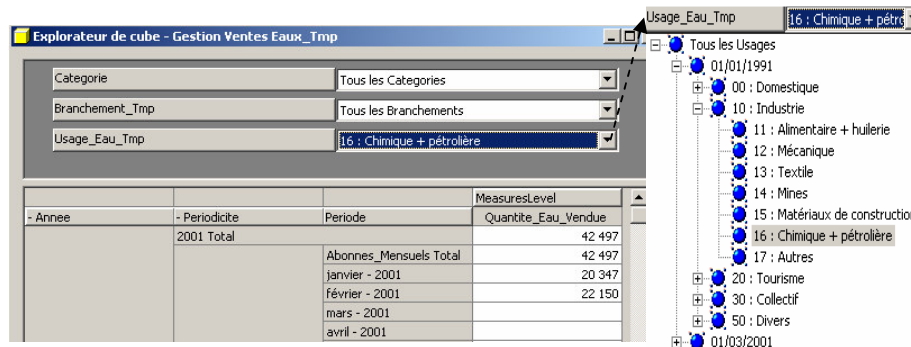
5 Conclusion

Les évolutions pouvant affecter les MemD ne se limitent pas à des opérations simples, telles que l'insertion ou la suppression d'un MemD, mais elles s'étendent à des opérations complexes, telles que les opérations découlant des modifications touchant la nomenclature de MemD. Ces dernières n'ont pas été suffisamment étudiées dans la littérature. Cet article s'est intéressé alors au recensement des différents aspects et problèmes d'évolution de nomenclature de MemD. Face à ces problèmes, nous avons proposé des solutions appropriées dans le cadre d'un système de gestion de DW multiversions. Les fonctions de transformation permettant d'assurer la transformation des données entre les versions des MemD, qu'il faut créer suite à une évolution de nomenclature de MemD, ne se limitent pas à l'utilisation d'un facteur poids, mais utilisent, si nécessaire des tables de correspondance et des requêtes de transformation. L'implémentation d'une solution ad hoc nous a permis de montrer la faisabilité de nos propositions dans le cadre du DW versionnable que nous avons

Impact de l'évolution de nomenclature sur les DW

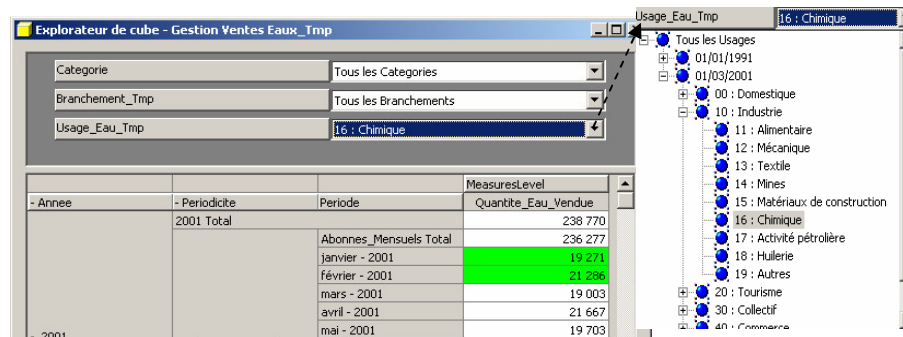
réalisé pour un cas réel.

L'implémentation d'un prototype basé sur une solution généralisée nous permettra de confirmer la faisabilité de nos propositions dans le cadre d'un système de gestion de DW multiversions. Nous envisageons également de définir formellement les différentes opérations d'évolution de nomenclature proposées, de procéder à une validation formelle de ces opérations et d'étudier les contraintes d'application.



- Année	- Périodicité	Période	MeasuresLevel	Quantite_Eau_Vendue
2001	Total			42 497
		Abonnes_Mensuels Total		42 497
		janvier - 2001		20 347
		février - 2001		22 150
		mars - 2001		
		avril - 2001		

FIG. 5 – Extrait des données du cube Gestion Ventes Eaux avant la subdivision perturbatrice de l'usage « 16 : Chimique et activités pétrolières »



- Année	- Périodicité	Période	MeasuresLevel	Quantite_Eau_Vendue
2001	Total			238 770
		Abonnes_Mensuels Total		236 277
		janvier - 2001		19 274
		février - 2001		21 286
		mars - 2001		19 003
		avril - 2001		21 667
		mai - 2001		19 703

FIG. 6 – Extrait des données du cube Gestion Ventes Eaux après la subdivision perturbatrice de l'usage « 16 : Chimique et activités pétrolières »

Références

- Bebel B., Eder J., Koncilia C., Morzy T., Wrembel R. (2004). Creation and Management of Versions in Multiversion Data Warehouse. *Proc. of the ACM Symposium on Applied Computing (SAC)*, Nicosia, Cyprus, 717-723.
- Bebel B., Krolkowski Z., Wrembel R. (2006). Formal Approach to modelling a multiversion data warehouse. *Bulletin of the polish academy of sciences*, vol. 54, No. 1.

- Blaschka M., Sapia C., Hofling G. (1999). On Schema Evolution in Multidimensional Databases. *Proceeding de la Conférence DaWaK'99*, Italie, 153-164.
- Bliujute R., Saltenis S., Slivinskas G., Jensen C. S. (1998). Systematic Change Management in Dimensional Data Warehousing. *A Time Center Technical Report*.
- Body M., Miquel M., Bédard Y., Tchounikine A. (2003) Handling Evolutions in Multidimensional Structures. *IEEE 19th International Conference on Data Engineering (ICDE)*, Bangalore, India, 581–591.
- Eder J., Koncilia C. (2001). Changes of Dimension Data in Temporal Data Warehouses. *Proceeding de la Conférence DaWaK'01*, Munich, Allemagne, 284-293.
- Eder J., Koncilia C., Kogler H. (2002a). Temporal Data Warehousing : Business Cases and Solutions. *ICEIS 2002*, 81-88.
- Eder J., Koncilia C., Morzy T. (2002b). The COMET Metamodel for a temporal Data Warehouse. *Proceeding de la Conférence CAISE' 02*, Toronto, Canada, 83-99.
- Ellouze L., Zouari I., Bouaziz R. (2006). Versionner les entrepôts de données sous les systèmes courants. *Atelier INFORSID'06*, Hammamet, TUNISIE.
- Favre C., Bentayeb F., Boussaid O. (2007). Evolution de modèle dans les entrepôts de données : existant et perspectives. *III^{èmes} journées francophones sur les Entrepôts de Données et l'Analyse en ligne (EDA'07)*, France, volume B-3 de la revue RNTI, 21-35.
- Hurtado C., Mendelzon A. O., Vaisman A. (1999). Maintaining data cubes under dimension updates. *Proceedings of IEEE/ICDE'99*, 68-80.
- Mendelzon A.O., Vaisman A.A. (2000). Temporal Queries in OLAP. *Proc. of the VLDB Conference*, Egypt, 242-253.
- Zouari I., Bouaziz R. (2007). Versionnement ad hoc d'un entrepôt de données de la SONEDE. *Atelier des Systèmes décisionnels (ASD'07)*, Sousse, Tunisie.
- Zouari I., Bouaziz R. (2008). Vers le versionnement des schémas des entrepôts de données. *I^{ère} Conférence Internationale sur les Systèmes d'Information et Intelligence Economique (SIIE'2008)*, Hammamet, Tunisie, Vol. 2, 628-644.

Summary

Current data warehouse (DW) management systems support evolutions of facts, but not those which can affect DW schema and dimension instances. Several solutions to these evolutions are proposed. They are mainly based on the historization and/or the versioning of DW components. However, few are the works which support the evolutions of dimension instance nomenclature and their impact on analysis. In this paper, we propose a classification of several kinds of nomenclature evolutions and we study the effects of these evolutions on analysis coherence. The solutions that we consider to each of these evolution aspects define some components of a multiversion data warehouse management system.