

NF26

Data Warehouse et Décisionnel

Principes de Constitution des Entrepôts de Données¹

Pierre Morizet-Mahoudeaux

1. Paternité - Partage des Conditions initiales à l'identique :
<http://creativecommons.org/licenses/by-sa/3.0/fr/>

Table des matières

Chapitre 1 Data Warehouse et Décisionnel - Introduction	11
1. Bref aperçu de la <i>business intelligence</i> et du décisionnel	11
1.1 Concepts du décisionnel et de la BI	11
1.2 La prise de décision	11
1.2.1 Les besoins informationnels	12
1.2.2 Le processus décisionnel	12
1.3 Architecture d'un système décisionnel	13
1.3.1 Acquisition et stockage de l'information	14
1.3.2 Restitution de l'information	14
1.4 Evolution du champ d'application du knowledge management	14
2. Rôles et finalités du <i>data warehouse</i> dans un SID	16
2.1 Composants d'un SID	16
2.1.1 Les systèmes opérationnels	16
2.1.2 L'acquisition de données	16
2.1.3 Le <i>data warehouse</i>	17
2.1.4 L' <i>operational data store</i>	17
2.1.5 La diffusion de données	17
2.1.6 Les <i>data marts</i>	18
2.1.7 Gestion des meta données	18
2.2 Résumé : la nature multifonctionnelle d'un système d'information décisionnel	18
3. Les besoins des utilisateurs	19
3.1 Reporting	19
3.2 Exploration et analyse	20
3.3 Tableaux de bord et pilotage	20
3.4 Fouille de données	20
4. Les grandes étapes de conception d'un data warehouse	20
4.1 Première étape : spécification d'un <i>data mart</i>	20
4.2 Deuxième étape : spécification d'un système fonctionnel	21
4.3 Troisième étape : spécification des outils d'éditions de rapports, d'interrogation et de requêtage	21
4.4 Les différents types de serveurs : ROLAP, MOLAP, HOLAP	22
Chapitre 2 Définition des Modèles Conceptuels de Données Décisionnels	23
1. Introduction	23
1.1 Principes généraux	23
1.2 Le couple entité-relation	23
1.3 Les limites du modèle normal (3NF, CODD) pour les SID	24
1.3.1 Normalisation des entités	24
1.3.2 Identifiant des entités	25
1.3.3 Dépendance fonctionnelle	25
1.3.4 Normalisation 3FN	25
1.3.5 Pourquoi ne peut-on pas conserver intégralement cette normalisation ?	25
2. Vues, Faits et Dimensions	26

2.1	Enrichissement de la vue	27
2.2	Relier les faits et les dimensions	29
3.	Intégration des vues	29
3.1	Notion de contexte	29
3.2	Hiérarchies	30
3.3	Synthèse de contextes	32
3.4	Normalisation des contextes	34
3.4.1	Dépendances et influences	34
3.4.2	Définition des faits	34
3.4.3	Cohérence de grain	35
3.4.4	Navigation hiérarchique	35
4.	Modèle relationnel de diffusion	36
Chapitre 3 Formes Dimensionnelles Complexes		39
1.	Introduction	39
1.1	Etats et flux	40
2.	Les représentations du temps	40
2.1	Irrégularités périodiques	41
2.2	Périodes et événements	41
2.2.1	Choix période/événement	41
2.2.2	Usage période/événement	42
2.2.3	Exemple : Contexte périodique/Contexte événementiel	43
3.	Dérives dimensionnelles	44
3.1	Dérives de contenu	44
3.1.1	Propriétés et associations permanentes/changeantes	44
3.1.2	Mise en forme dimensionnelle	45
3.1.3	Mémorisation des états dimensionnels	46
3.2	Dérives de périmètre	46
3.3	Dimensions changeantes et boucles hiérarchiques	46
4.	Indicateurs qualifiés	48
5.	Les différents types de changements et leur gestion	49
6.	Méthodes de consolidation	52
Chapitre 4 Traitement des données pour l'alimentation, la diffusion et l'OLAP		53
1.	Introduction	53
2.	Systèmes intermédiaires	54
3.	Architecture de référence du SID	54
4.	Architecture et modèles de données	54
5.	Alimentation	56
5.1	Pré-traitement des données	56
5.2	Résumé des données descriptives	57
5.2.1	Mesures de la tendance moyenne	57
5.2.2	Mesure de la dispersion des données	58
5.3	Nettoyage des données	59
5.3.1	Données manquantes	59
5.3.2	Données bruitées	59
5.3.3	Procédure de nettoyage des données	61
5.4	Intégration et transformation des données	61
5.4.1	Intégration de données	61
5.4.2	Transformation	62
5.5	Réduction des données	63
5.5.1	Agrégation	63
5.5.2	Discrétisation et hiérarchisation	63
5.5.3	Formatage et standardisation	64
5.6	Génération des identifiants et des clés	64

6.	Le système de Diffusion et de Présentation	65
6.1	Modalités d'accès à l'information	65
6.1.1	Etats prédéfinis	65
6.1.2	Requêtes paramétrables	65
6.1.3	Manipulation dimensionnelle libre	65
6.1.4	Simulation	66
6.1.5	Recherche de connaissances	66
6.1.6	Alertes	66
6.1.7	Mises à jour interactives	66
6.1.8	Consultation de données opérationnelles	67
6.2	Traitement des agrégats	67
6.3	Contextes résumés et partitions	68
6.4	Optimisation des calculs sur les cubes	68
6.4.1	Exemple 4.1.	68
6.4.2	Matérialisation partielle : calcul sélectif de cuboïdes	69
6.4.3	Méthodes d'agrégation dans les cubes	70
6.5	Opérations OLAP typiques	70
6.6	Indexation des données OLAP	72
6.6.1	Exemple 4.2.	72
6.6.2	Exemple 4.3.	73
Chapitre 5	Exemples de Modèles Multidimensionnels¹	75
1.	Démarche générale	75
2.	Etude de cas de la distribution	76
3.	Les quatre étapes de cette application	76
3.1	Etape 1 : sélection du processus à modéliser	76
3.2	Etape 2 : déclaration du grain	76
3.3	Etape 3 : choix des dimensions	76
3.4	Etape 4 : Identification des faits	77
4.	Attributs de table de dimension	77
4.1	Dimension date	77
4.2	Dimension produit	78
4.3	Dimension magasin	79
4.4	Dimension promotion	80
4.5	Table de faits sans fait relative aux promotions	80
4.6	Dimension numéro de transaction dégénérée	80
4.7	Extensibilité du modèle	81
4.8	Normalisation des dimensions	81
4.9	Trop de dimensions	83
5.	Etude de cas : Les stocks dans le magasin	84
5.1	La chaîne de valeur	84
5.2	Modèles de stock	84
5.2.1	Instantané périodique de stock	84
5.2.2	Faits semi-additifs	85
5.3	Faits de stock améliorés	85
5.4	Transaction de stock	86
5.5	Instantané récapitulatif de stock	87
6.	Architecture de bus de l'entrepôt de données	87
Chapitre A	Références	89

1. Ce Chapitre est tiré de Kimball et Ross, pp. 31-90.

Table des figures

1.1	Le besoins informationnels en fonction des utilisateurs	12
1.2	Corporate information factory (d'après Mastering Data Warehouse Design, C. Imhoff, N. Gallemmo, J.G. Geiger, Wiley edts, 2003, p. 7)	13
1.3	Evolution des concepts du décisionnel (d'après Micropole, supports de cours)	15
1.4	Les trois niveaux de modélisation des données (d'après Le Projet Décisionnel, Jean-Marie Gouarné, Eyrolles Edt. 1998, pp. 35)	21
1.5	Le cycle de conception d'un <i>data warehouse</i>	22
2.1	Entités et associations	24
2.2	Association Employé-Véhicule	24
2.3	L'ajout du nom du chef de service dénormalise l'entité Employé)	25
2.4	Modèle conceptuel de données en 3ème Forme Normale (d'après Le Projet Décisionnel, Jean-Marie Gouarné, Eyrolles Edt. 1998, pp. 41)	26
2.5	Vue Frais/Employé/Véhicule/Région/Mois	27
2.6	Variante enrichie de la figure 2.5	28
2.7	Présentation tabulaire en quadri-dimensionnel de l'exemple 2.5	29
2.8	Quatre vues indépendantes	30
2.9	Exemples de hiérarchies	31
2.10	Hiérarchies périodiques multiples	32
2.11	Hiérarchies multiples sur le « <i>Client</i> »	32
2.12	Contexte « Activité commerciale »	33
2.13	Hiérarchie cyclique	35
2.14	Hiérarchie acyclique	36
2.15	Schéma en flocon	37
2.16	Schéma en étoile	38
2.17	Schéma en constellation	38
3.1	Deux contextes (périodique et événementiel) de la même activité	43
3.2	Entité mouvante (m) et entité permanente (p)	45
3.3	Propriétés changeantes dans un contexte dimensionnel	45
3.4	Identifiant entité mouvante / identifiant entité permanente	46
3.5	Double liaison Personne - Commune	47
3.6	Double liaison Personne - Commune	47
3.7	Régularisation d'une hiérarchie cyclique	48
3.8	Contexte qualifié	49
4.1	Architecture de référence d'un SID (d'après Le Projet Décisionnel, Jean-Marie Gouarné, Eyrolles Edt. 1998, pp. 115)	55
4.2	Architecture et modèles de données (d'après Le Projet Décisionnel, Jean-Marie Gouarné, Eyrolles Edt. 1998, pp. 117)	55
4.3	Exemples de différentes déviations d'une population a) normale, b et c) décalée à gauche ou à droite	58
4.4	Visualisation de boites à moustaches	58

4.5	Régression linéaire	60
4.6	Détection des <i>outliers</i> par méthode de classification.	60
4.7	Identifiants et clés	64
4.8	Tables de faits spécialisés par niveau d'agrégation	67
4.9	Treillis de cuboïdes constituant un cube 4-D. Chaque cuboïde représente un groupe-by différent. Le cuboïde de base contient les quatre dimensions <i>lieu</i> , <i>temps</i> , et <i>fournisseur</i> (d'après Data Mining Concepts and Techniques, J. Han and M. Kamber, 2001, pp. 48)	69
4.10	Exemples d'opérations OLAP typiques sur des données multidimensionnelles(d'après Data Mining Concepts and Techniques, J. Han and M. Kamber, 2001, pp. 59)	71
4.11	Liens entre la table de faits <i>ventes</i> et les tables de dimension <i>lieu</i> et <i>item</i>	73
5.1	Les deux éléments clé contribuant aux quatre étapes du processus de modélisation dimensionnelle	75
5.2	Schéma primaire des ventes en grande distribution	76
5.3	Faits mesurés du schéma vente au détail	77
5.4	Schéma d'une dimension date pour la distribution	78
5.5	Schéma d'une dimension produit pour la distribution	79
5.6	Schéma d'une dimension magasin pour la distribution	79
5.7	Schéma d'une dimension promotion pour la distribution	80
5.8	Requête sur le schéma de vente au détail	81
5.9	Extension du modèle des ventes	82
5.10	Effet de floconisation par normalisation d'une dimension	82
5.11	Table de faits mille pattes avec trop de dimensions	83
5.12	La chaîne de valeur	84
5.13	Schéma de l'instantané périodique de stock. Si on analyse les niveaux de stock dans des entrepôts plutôt que dans des magasins, la modélisation sera identique, avec entrepôt à la place de magasin.	85
5.14	Instantané de stock étendu pour supporter l'analyse de retour de marge brute sur stock	86
5.15	Modèle de transaction de stock d'entrepôt	87
5.16	Modèle récapitulatif de stock d'entrepôt	87
5.17	Les dimensions communes sur toute la chaîne de valeur	88
5.18	Les lignes de la matrice de bus correspondent à des marchés d'information	88

Liste des tableaux

2.1	Représentation relationnelle de l'association Faits-Dimensions	29
3.1	Table Clients	49
3.2	Table Clients mise à jour	50
3.3	Incrémantion des changements et modification des clés primaires	50
3.4	Enregistrement des dates de validité	50
3.5	Ajout de colonnes de changement d'état	51
3.6	Table Client	51
3.7	Table Historique Client	51
3.8	Table Client initiale (1)	51
3.9	Premier changement d'état (2)	52
3.10	Deuxième changement d'état (3)	52
4.1	Table relationnelle <i>Client</i>	72
4.2	Index binaire sur les propriétés <i>Profession</i> et <i>Ville</i>	72
4.3	Tables d'index de jointure lieu/ventes, produit/ventes, lieu/produit/vente	73
5.1	Détail d'une table de dimension date	77
5.2	Détail d'une table de dimension produit	78

Chapitre 1

Data Warehouse et Décisionnel - Introduction

1. Bref aperçu de la *business intelligence* et du décisionnel

Formalisé au début des années 1990, le concept d'entrepôt de données (*data warehouse*) est devenu un élément essentiel de ce qu'on appelle aujourd'hui **l'informatique décisionnelle** et est à la base de la **business intelligence** (BI). Cette notion, qui concerne essentiellement le management des entreprises, fait partie de celle plus large de **la prise de décision**.

Nous allons donc commencer ce chapitre d'introduction par un très court tour d'horizon du domaine général de la prise de décision. Nous donnerons ensuite un aperçu de la structure d'un système décisionnel puis les grandes étapes qui ont marqué un des champs d'application important de ce domaine qui correspond au *knowledge management* (KM). Cela nous permettra alors de justifier l'approche du développement d'un **système d'information décisionnel** (SID) par l'étude des besoins des utilisateurs.

A partir de cette présentation préliminaire, nous pourrons donner les éléments qui vont caractériser un SID par la description des tâches qu'il doit réaliser, les fonctions qu'on en attend, les traits essentiels de sa structure et ses composants de base.

Nous terminerons par une description succincte des étapes de conception d'un data warehouse, qui nous permettra d'introduire les concepts qui en constituent le modèle.

1.1 Concepts du décisionnel et de la BI

Voici deux façons de définir la *business intelligence* au travers de la vision d'une société de service (Micropole) et d'universitaires :

- Pour Micropole, la BI c'est l'art fournir des solutions d'**informations** à valeur ajoutée aux **utilisateurs** (management ou niveau opérationnel) afin de prendre les meilleures **décisions**.
- Pour Imhoff & al. la BI, dans le contexte du *Data Warehouse*, c'est la capacité pour une entreprise d'étudier ses comportements et actions passés dans le but de **comprendre** ce qu'elle a été, de **déterminer** sa situation courante et de **prédir** ou **changer** ce qui va se produire dans le futur.

Dans les deux cas présentés ci-dessus la BI a pour objectif de fournir des éléments informationnels pour aider la prise de décision. La première met en avant la notion d'information élaborée, la seconde, la notion d'historique et de modèle.

1.2 La prise de décision

La prise de décision s'appuie sur un schéma de transformation des données, d'abord en information puis en connaissances. Le passage de la donnée à l'information n'est possible que grâce à un modèle interprétatif propre au récepteur. L'acquisition de cette information et son traitement permettent alors d'élaborer une connaissance pour orienter un comportement en réponse à un stimuli (l'expression d'un besoin). Exemples :

- une décision d'approvisionnement s'appuie sur les données du volume de stock, l'historique des ventes antérieures et un modèle des flux d'entrée/sortie de produits,
- une autorisation de prêt bancaire s'appuie sur les données comptes-clients et un modèle de prise de risque,
- l'affectation d'un gène à la classe *actif* s'appuie sur des données expérimentales et un modèle en réseau de l'activité des gènes.

1.2.1 Les besoins informationnels

On peut distinguer les besoins informationnels en fonction de la situation de l'utilisateur et du type de décision requis. Les cadres supérieurs et les dirigeants doivent prendre des décisions stratégiques à partir d'information internes et externes à l'entreprise, résumées, synthétisées, qui couvrent un large spectre, peu répétitives et qui reflètent le passé, le présent et l'avenir. Les cadres moyens et les contrôleurs prennent des décisions de gestion tactique et de contrôle à l'aide d'informations pour la plupart internes, agrégées, restreintes à un domaine et relatives au présent et au passé proche. Les gestionnaires d'opérations sont en charge du contrôle opérationnel à l'aide d'informations internes, détaillées, spécifiques, répétitives et demandant un temps de réponse court et essentiellement représentatives du temps présent.

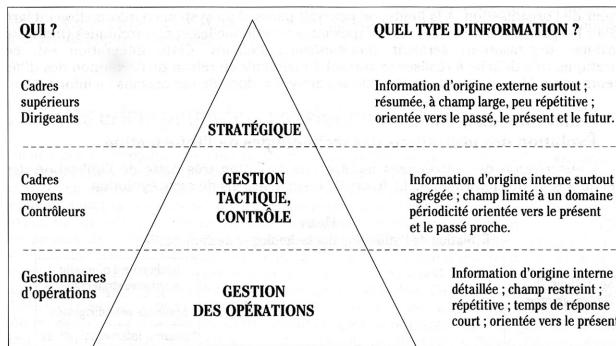


FIGURE 1.1 – Le besoins informationnels en fonction des utilisateurs

1.2.2 Le processus décisionnel

L'interprétation des données qui sous-tend un processus décisionnel dépend de trois éléments ; le modèle d'analyse dont on dispose, l'apprentissage vécu et l'expérience et enfin, la complexité de la situation et la recherche d'innovation qu'elle demande. Le processus décisionnel peut alors se décrire selon les huit étapes suivantes :

1. identification d'un problème
2. identification des critères de décision
3. attribution des priorités
4. développement des différentes options
5. analyse des options
6. sélection d'une option
7. mise en place de l'option
8. évaluation de la décision

L'élaboration de modèles décisionnels a donné lieu à de nombreux travaux dont les plus connus sont :

- a Le modèle de l'acteur unique ou mono rationnel (modèle de Harvard). Ces caractéristiques sont la rationalité des acteurs et le cartésianisme.

- b** Les observations de H. Simon (1955). Il introduit les notions de sélectivité et d'incomplétude de l'information, les limites de capacité de stockage de la mémoire (des individus), le rôle de l'expertise, l'influence du contexte qui est souvent sans lien avec la rationalité. Ces travaux mettent en évidence l'existence de biais cognitifs (sous utilisation des procédures probabilistes, généralisation à partir de cas singuliers, excès de confiance dans son jugement d'expert, implication dans une opération), du rôle du contexte (simplification de la complexité), la recherche du moindre effort cognitif, et enfin la fréquence importante de décisions sous optimales.
- c** L'hypothèse de la rationalité économique (A. Smith) à la base de l'analyse du processus décisionnel en économie et gestion reste cependant valable jusque dans les années 1970.
- d** L'hypothèse de la maximisation des profits (Walras).
- e** La théorie des jeux et hypothèse des anticipations rationnelles (Von Neumann et Morgenstern).

La remise en cause de la rationalité par H. Simon amène à concevoir le processus décisionnel non plus comme une somme de processus individuels mais comme un processus collectif. En particulier **les besoins et les attentes des utilisateurs doivent prendre une place importante dans son élaboration**. Les modèles qui ont alors ensuite été proposés ont essayé de prendre en compte les différentes observations de Simon telles que, la boucle d'apprentissage organisationnel par l'expérience (C. Argyris), la théorie des possibilités (D. Kahneman et Tversky, 1979) ou le modèle de la poubelle (garbage can theory). La prise en compte de la responsabilité fonctionnelle et opérationnelle, de la position dans la hiérarchie, des domaines de compétences, de la convergence dans les objectifs de la commande amènent alors à **étudier le processus de décision par le biais des processus au sein de l'entreprise et non plus par celui des besoins propres des services**. On cherche à obtenir des informations sur le processus de vente, le processus d'approvisionnement et de stockage ou la qualité de service et non plus sur le service achat, le service vente, le service facturation, le service approvisionnement, le service fabrication, le service après-vente, etc.

1.3 Architecture d'un système décisionnel

Les développements de ces dix dernières années ont convergé vers une architecture largement adoptée aujourd'hui pour répondre aux besoins technologiques de la BI et du processus décisionnel. L'architecture que l'on rencontre dans la plupart des environnements décisionnels s'appuie sur l'agrégation de cinq types principaux de bases de données (voir figure 1.2) : les systèmes opérationnels, le *data warehouse*, le magasin de données opérationnelles (*operational data store*), les *data marts* et les *oper marts*.

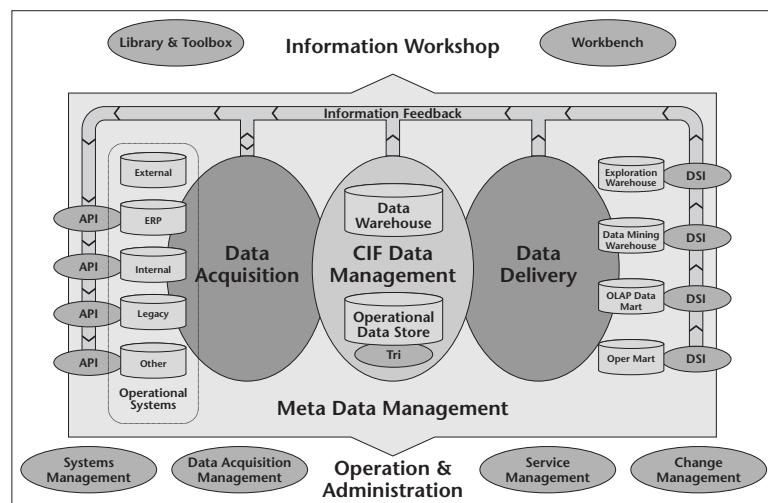


FIGURE 1.2 – Corporate information factory (d'après Mastering Data Warehouse Design, C. Imhoff, N. Gallemmo, J.G. Geiger, Wiley édts, 2003, p. 7)

L'objectif d'un SID est de fournir des solutions techniques pour réaliser un traitement de l'information à valeur ajoutée, à destination des utilisateurs afin qu'ils prennent les meilleures décisions. Pour atteindre cet objectif, l'architecture du SID se décompose en deux composants principaux, l'un pour acquérir et stocker l'information, l'autre pour restituer l'information à l'utilisateur.

1.3.1 Acquisition et stockage de l'information

L'entrée des données repose sur des processus et des bases de données impliqués dans l'acquisition de données issues de systèmes opérationnels, leur intégration, nettoyage et stockage dans des bases de données pour en permettre un usage aisé. Les composants du SID que l'on trouve pour réaliser cette fonction sont :

- les bases de données des systèmes opérationnels (systèmes sources) qui contiennent les données utilisées pour le fonctionnement courant de l'entreprise. Ce sont les sources d'information principales pour l'environnement qui aident à la prise de décision :
- interfaces avec les systèmes et données opérationnelles (le reste du système d'information : systèmes opérationnels, bases de données internes),
- saisie de données de type objectifs par les utilisateurs,
- intégration de données externes (concurrence, INSEE, ...),
- le *data warehouse*, qui est une structure d'accueil de données intégrées, détaillées, historisées pour l'aide à la décision stratégique,
- un *operational data store* qui est une structure d'accueil de données intégrées, détaillées, courantes pour l'aide à la décision tactique,
- un processus d'acquisition de données pour alimenter le *data warehouse* et le *operational data store* avec les données extraites des systèmes opérationnels. Les opérations réalisées par ce processus consistent en nettoyer, versionner, historiser, fédérer, assurer l'intégrité et la qualité de l'information avec pour objectif d'alimenter le *data warehouse* et le *data store* avec des données au format entreprise.

1.3.2 Restitution de l'information

La restitution de l'information repose sur des processus et des bases de données impliqués dans la production de données élaborées (BI) pour l'utilisateur final ou l'analyste. Les composants du SID que l'on trouve pour réaliser cette fonction sont :

- les *data marts* dérivés du data warehouse pour donner accès à différentes formes d'analyses stratégiques,
- les *opermarts* dérivés de l'ODS pour des accès multidimensionnels à des données opérationnelles courantes,
- la diffusion de données, procédé qui permet le déplacement des données du *data warehouse* vers les *marts* avec les manipulations appropriées. Ce processus s'apparente à celui de l'acquisition de données initial, mais il s'applique aux données élaborées de haut niveau qui se situent dans le *data warehouse* et l'ODS et qui sont conformes aux règles de fonctionnement de l'entreprise.

1.4 Evolution du champ d'application du knowledge management

Voici quelques dates importantes de l'évolution de l'informatique décisionnelle.

- 1970-1995 : Le KM a d'abord été global et tourné vers "l'interne"
 - Gestion des compétences (Peter F. Drucker)
 - Gestion des Best practices
 - Gestion des connaissances techniques
 - Gestion du capital intellectuel (Karl-Eric Sveiby, 1987)
 - Gestion des communautés de pratique (Etienne Wenger, 1988)
- 1995-2000 : Le KM, toujours global, s'ouvre sur "l'extérieur"
 - Apparition du modèle de transformation de connaissances tacites en connaissances explicites (Ikujiro Nonaka, Hirotaka Takeuchi, 1995)
 - Recherche des Best Practices mondiales

- Connaissance et besoins des clients (CKM : customer knowledge management)
- Connaissance et besoins des acheteurs (SKM : supplier knowledge management)
- Gestion du capital immatériel (Leif Edvinson, 1996)
- La voie japonaise du KM : le Ba (Ikujiro Nonaka, 1998)
- Le World Wide News (internet)
- 2000-2006 : Le *mixte KM* global avec apparition du niveau personnel
 - Gestion de la relation avec les clients : CRM (ventes, marketing, support clients)
 - Création de valeur pour les clients comme les collaborateurs de l'entreprise
 - Connaissance de ses concurrents, veille ou IEC
 - Gestion de la relation avec ses fournisseurs, SRM
 - Support à l'innovation
- 2006-2010 : Le Personal KM (ou *PKM* = BI + CRM + KM)
 - Emergence du concept de Personal Knowledge (Tom Davenport)
 - Convergence du besoin en information fournie par le BI & CRM
 - Convergence du besoin de gestion de l'information structurée et non structurée
 - Gestion de son information personnelle
 - Gestion de ses compétences personnelles
 - Gestion de son réseau personnel
 - Gestion de son temps
 - Apparition du concept de "workflow personnel"

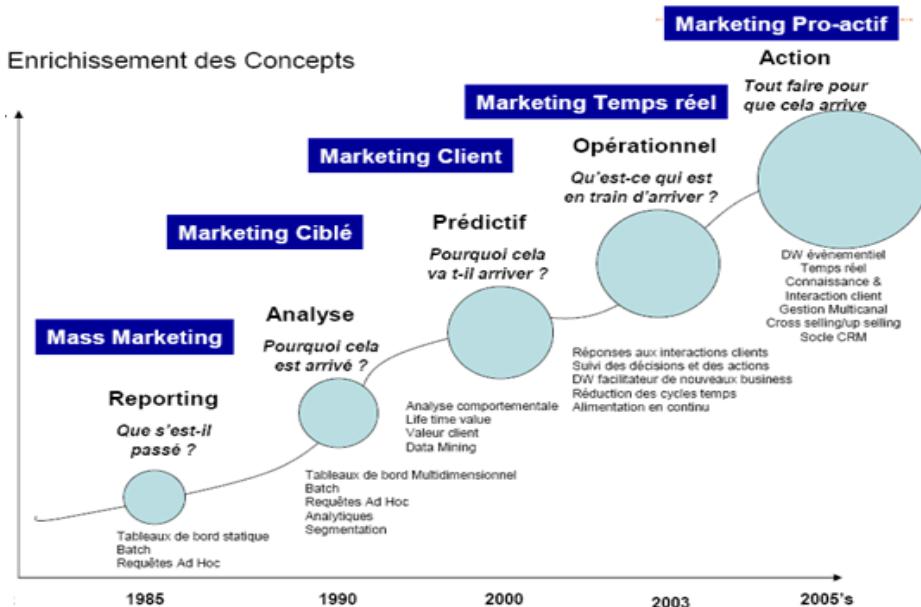


FIGURE 1.3 – Evolution des concepts du décisionnel (d'après Micropole, supports de cours)

- Le décisionnel aujourd'hui est orienté vers le pilotage et la réactivité. On est passé de la notion de
- **EPM (Enterprise Project Management)** : solution complète de gestion de projet intégrée autour d'une base de données centrale,
 - à **EPM (Enterprise Performance Management)** : ensemble de processus qui aident les entreprises à optimiser leurs performances.

C'est un cadre organisationnel, automatisé et analytique qui prend en compte la méthodologie, les processus et les systèmes qui conduisent la performance de la société. Les référentiels demandent une forte réactivité du système d'information sous la pression business / compétitivité.

Techniquement, on tend vers une simplification des architectures : on introduit le MDM (Master Data Management), qui correspond à l'ensemble de process et d'outils qui définit et gère de façon cohérente

les données non transactionnelles¹ d'une organisation. Il a pour objectif de fournir les procédures pour collecter, agréger, mettre en correspondance, vérifier la qualité, assurer la persistance et distribuer les données dans toute l'organisation pour assurer la cohérence et le contrôle de la maintenance et l'usage de l'information.

2. Rôles et finalités du *data warehouse* dans un SID

Il faut qu'un *data warehouse*, soit une zone de transfert, un entrepôt d'information, réunisse différentes compétences, fonctions et technologies, soit au niveau de détail suffisant pour tous les *data marts*, ne compromette pas la possibilité d'utiliser différentes technologies pour les *data marts* (analyses multidimensionnelles, statistiques, fouille de données, navigation, ...), puisse accueillir de nouvelles applications et technologies, puisse gérer différents modèles (schémas) de données (étoile, fichiers plats, sous-ensembles statistiques de données normalisées et ceux à venir)

Pris dans son ensemble la “Corporate Information Factory” (CIF) ou “Système d'Information Décisionnel” (SID) est *une architecture conceptuelle* qui décrit et caractérise les entrepôts d'information utilisés pour gérer trois processus organisationnels de haut niveau

- *Business operations* concerne les opérations au jour le jour. Ces processus sont pris en charge par les systèmes opérationnels de transaction et les données externes. Ils assurent des fonctions globalement statiques, stables et ne connaissent que des changements marginaux.
- *Business intelligence* concerne la recherche d'une meilleure compréhension de l'entreprise, ses produits, ses clients, ... Ce sont des processus en constante évolution à mesure que les analystes explorent l'information disponible pour développer de nouveaux produits, mesurer la fidélisation des clients, évaluer les nouveaux marchés potentiels, etc. C'est le processus stratégique de l'entreprise.
- *Business management* concerne les connaissances et nouvelles visions développées par la BI qui sont institutionnalisées et introduites dans les affaires courantes de l'entreprise.

2.1 Composants d'un SID

2.1.1 Les systèmes opérationnels

Les systèmes opérationnels sont ceux qui assurent les activités transactionnelles au jour le jour de l'entreprise. Ils sont centrés sur les processus transactionnels. Ils s'appuient sur une grande variété de technologies, architectures et processus divers, optimisés et stables, qui peuvent avoir été développé en interne, acquis auprès d'éditeurs de logiciels ou distribués par des sociétés de services.

Ce sont les sources principales des données numériques du SID. Ils sont optimisés pour être performants sur les processus de transaction qu'ils gèrent, en particulier ceux qui sont très sensibles aux temps de réponse. Les données de l'environnement opérationnels sont parfois dupliquées et rarement synchronisées. Eléments de base pour l'application des règles de fonctionnement de l'entreprise, la qualité de données qu'ils fournissent a un impact direct sur toute la chaîne de traitement de l'information.

2.1.2 L'acquisition de données

L'acquisition coordonnée et synchronisée des données transactionnelles est un étape essentielle de saisie, transformation, homogénéisation, test d'intégrité, qualité, etc. de toute la chaîne de traitement. Sa mise en œuvre consomme régulièrement au moins 70% de l'effort de développement.

La partie préparation de l'entrepôt de données s'appuie sur un ensemble de processus appelés ETC, pour *extraction, transformation, chargement* (ou **ETL**, *extraction, transformation, loading*). L'accès à cette zone est limité aux seuls développeurs et n'est en aucun cas disponible aux utilisateurs. Elle ne fournit donc aucun service de requête ou de présentation. On y trouve généralement deux types de bases de données (voir ci-dessous). Une première base, qui est la simple copie (avec quelques contrôles d'intégrité) des données transactionnelles – parfois aussi appelée **ODS** (*operational data store*) qui est une version différente de celle présentée figure 1.2. Une seconde **base de donnée de très grande**

1. les clients, les fournisseurs, les employés, les matériaux, ..., sont des exemples de données non transactionnelles

dimension, qui est le cœur du *data warehouse* et qui contient tous les éléments pour alimenter les *data marts* de la couche suivante.

L'extraction est la première étape du processus d'import de données dans l'entrepôt. Les données sont copiées dans la zone de préparation en vue de manipulations ultérieures. Une fois dans cette zone, elles peuvent être nettoyées (corrections et normalisations orthographiques, résolution de conflits, traitement d'éléments manquants, conversions, ...), combinées à d'autres données en provenance d'autres sources, et éventuellement pourvues de clés propres à l'entrepôt de données.

2.1.3 Le *data warehouse*

Une définition largement acceptée du *data warehouse* proposée par Bill Inmon dans les années 80 est “un recueil de données intégrées, dépendantes du temps, non volatiles et **orientées sujet** utilisées pour les prises de décisions stratégiques².

Il donne une **vue commune à toute l'entreprise des données** indépendamment de leur usage futur et des services. Il représente une source stable d'information historisée, qui est constante, cohérente et fiable. Il doit pouvoir grossir dans des proportions très importantes dans la mesure où il va conserver l'historique de l'entreprise sur de nombreuses années.

Il doit pouvoir alimenter toute forme de technologie analytique pour la communauté (tous les *data marts* doivent pouvoir être alimentés par le même *data warehouse*).

2.1.4 L'*operational data store*

L'*operational data store* est centré sur l'aide à la décision tactique, le *data warehouse* étant centré sur l'aide à la décision stratégique. Il est cependant

- orienté sujet comme le *data warehouse*,
- il contient des données intégrées comme le *data warehouse*,
- les données qu'il manipule sont quasi courantes (ou proches de), avec un historique minimal qui décrit l'état des entités aussi proche que le temps réel le permet,
- ses données sont volatiles et pouvant être écrasées ou mises à jour, sans trace d'un historique de valeurs,
- les données sont plus détaillées et peu agrégées par rapport au *data warehouse*,
- l'*ODS* est pourtant accessible de toute partie de l'entreprise et n'est pas spécifique d'une application (d'un service).

2.1.5 La diffusion de données

La diffusion de données est en général limitée à des opérations d'agrégation, de filtrage sur les dimensions ou, selon les besoins, d'applications BI. Elles réalisent un re-formatage des données pour faciliter l'accès des utilisateurs ou d'applications BI spécifiques et *in fine* assurent la transmission de données sur l'ensemble de l'entreprise. La couche de diffusion doit cependant être flexible pour s'adapter aux nouveaux besoins et applications.

La zone de diffusion des données est le lieu où les données sont organisées, stockées et rendues disponibles aux requêtes directes des utilisateurs, aux programmes de reporting et aux applications d'analyse et de fouille.

Les données doivent y être représentées, stockées et consultées sous forme de schémas multidimensionnels. Intuitivement, la notion de dimension représente un espace dont les axes servent à cumuler des entités en relation. Par exemple, si on veut évaluer la performance dans le temps d'un produit, sur certains marchés, on pense naturellement à représenter cette activité dans un espace multidimensionnel dont les axes seront le produit, le marché et le temps. On peut imaginer de faire des coupes en tranche et en dés (des projections) dans cet espace le long de chacune de ces dimensions. Les coordonnées des points de cet espace correspondent à des mesures pour chaque combinaison de produit, marché et temps.

Comme nous le verrons, la modélisation dimensionnelle est très différente de la modélisation en forme normale (3NF). On verra en particulier qu'un modélisation dimensionnelle bien conçue et bien construite garantit le déploiement d'un SID à coût justifié, dont les phases de développement restent gérables par

2. Building the Data Warehouse, Third Edition by W.H. Inmon, Wiley pub., 2001

le service informatique, dont les procédures d'alimentation sont correctement définies, dont les moyens d'accès sont clairs et dont les temps de réponse sont raisonnables.

Les données de la zone de présentation doivent être détaillées jusqu'au niveau atomique, de façon à pouvoir répondre à des requêtes imprévisibles des utilisateurs. Cela n'interdit pas qu'elles coexistent avec des données déjà agrégées ou pré-calculées de façon à répondre rapidement à des requêtes fréquentes et régulières.

2.1.6 Les *data marts*

Issus de la zone de diffusion, ils sont construits sur des sous-ensembles ou la synthèse de parties du *data warehouse*. Les données sont conditionnées pour des applications spécifiques (reporting, analyse, navigation, visualisation, analyse de KPI, ...). Ils peuvent être intégrés sur la plateforme du *data warehouse* ou être installés sur des plateformes spécifiques.

C'est un ensemble de moyens fournis aux utilisateurs pour exploiter la zone de diffusion. Un outil d'accès peut être un simple outil de requête *ad hoc* ou un algorithme complexe de fouille de données. Les outils les plus utilisés sont des applications d'analyse préfabriquées dont l'utilisateur se contente de fixer certains paramètres. La qualité des interfaces de saisie et de restitutions est essentielle dans ce cas là.

2.1.7 Gestion des meta données

C'est un ensemble des processus qui rassemblent, gèrent et distribuent les métadonnées dans le SID. On distingue trois catégories :

- les méta données techniques : elles décrivent la structure physique du SID et le processus qui déplacent et transforment les données dans l'environnement,
- les méta données sémantiques : elles décrivent les structures, éléments et règles d'usage dans des termes appropriés à la vision qu'en a l'utilisateur,
- les méta données d'administration : elles décrivent les opérations du SID incluant les traces, mesures de performance, de qualité et autres méta données statistiques.

2.2 Résumé : la nature multifonctionnelle d'un système d'information décisionnel

Nous avons présenté ci-dessus les différents éléments qui composent un SID pour répondre aux besoins des utilisateurs. On peut alors transformer l'expression de ces besoins, sous forme d'un cahier des charges, d'une part relativement aux fonctions que doit remplir l'entrepôt de données et, d'autre part, relativement à sa structure.

- L'entrepôt de données³ doit rendre les données de l'organisation facilement accessibles. Le contenu de l'entrepôt doit être facile à comprendre. Les données doivent être parlantes et leur signification évidente pour l'utilisateur. Pour être lisible, le contenu doit être étiqueté de manière significative. Les outils d'accès doivent être simples et faciles à utiliser, avec des temps de réponses minimes. Ils doivent permettre de séparer et combiner les données de toutes sortes de façons.
- Le SID doit présenter l'information de manière cohérente. Les données doivent être crédibles, même si elles sont assemblées à partir de plusieurs sources d'information. Si deux mesures portent le même nom, elles doivent vouloir dire la même chose. Inversement, si deux mesures ne veulent pas dire la même chose, elles doivent avoir des noms différents. La cohérence implique une qualité des données élevée. Elle suppose que l'on a tenu compte de toutes les données, qu'elles sont complètes.
- Centré entreprise, c'est le point de départ pour tous les *data marts* et applications d'analyse, il est utilisé par l'ensemble des départements.
- C'est un lieu d'intégration des données opérationnelles et de diffusion de données élaborées. Il est donc **séparé** des applications de production mais en reste **dépendant** pour son alimentation. Les caractéristiques techniques des applications de production et des sources externes dans lesquelles le SID puise ses données ne peuvent pas et ne doivent pas influer sur les modalités selon lesquelles l'utilisateur accède à l'information

³. Ce terme est ici pris dans son sens générique. Il couvre à la fois la structure globale (ETL + DWH + DM) et ce qu'on voit à la sortie, c'est à dire la réponse au besoin utilisateur.

- Beaucoup des traitements qu'effectue un SID ne sont pas déterminés par des algorithmes pré-établis. Ils ont pour but de permettre à l'utilisateur d'établir lui-même des rapprochements et des consolidations non pré-définis entre les données. Le **modèle de données de diffusion**, qui est un élément clé de la définition du système, doit être conçu dans cette perspective selon une approche **multidimensionnelle**.
- Le SID doit être adaptable et résistant aux changements. Les besoins des utilisateurs, les conditions d'activité, les données et la technologie sont en perpétuelle évolution. Ces modifications doivent pouvoir être prises en compte par l'entrepôt de données, sans remettre en cause les données existantes. Elles ne doivent pas invalider les données existantes et les applications ne doivent pas être modifiées ou bouleversées lorsque les utilisateurs posent de nouvelles questions ou que de nouvelles données sont ajoutées à l'entrepôt. Si les données descriptives de l'entrepôt doivent être modifiées, il faut pouvoir en rendre compte convenablement. Il faut donc penser le système indépendamment des processus, des applications et des technologies BI. Les spécifications d'un SID sont donc instables (cibles stratégiques mouvantes, modification de l'expression des besoins au fur et à mesure du déploiement du système).
- Conçu pour charger de gros volumes de données en un laps de temps très court, il faut minimiser les redondances ou données dupliquées, favoriser les fichiers plats (base de données rationnelle).
- De même, il doit être conçu pour optimiser le processus d'extraction de données par les programmes de diffusion qui alimenteront les *data marts*.
- L'information décisionnelle est **chronologique** (analyse de phénomènes évoluant dans le temps), la représentation du temps est un élément clé des modèles multidimensionnels.
- L'entrepôt de données doit être protégé. Il contient de précieuses informations sur l'entreprise, ce qu'elle vend, à qui, à quel prix, quelles sont ses interrogations, etc. Il doit donc posséder un contrôle d'accès rigoureux aux informations confidentielles de l'organisation.
- Le SID sert de socle à la prise de décision. Il doit contenir des données **objectives** et **qualifiées** servant à étayer ces décisions. Il doit correspondre à un projet d'entreprise clair.
- L'entrepôt de données doit être accepté par la communauté des utilisateurs.

Le cœur d'un SID réside dans son **modèle de données**. Nous allons donc largement développer cette problématique. Un modèle de données sans données serait une coquille vide. A la problématique de modélisation succède donc naturellement celle de **l'alimentation**. Ce n'est pas un simple problème de connectique et de transfert physique. C'est même le problème conceptuel et architectural le plus délicat du système. Il sera donc également largement étudié.

3. Les besoins des utilisateurs

Un système décisionnel se définit toujours à partir des besoins des utilisateurs. A partir de l'expression de ces besoins, on définit quatre grandes fonctions que doit remplir un système décisionnel :

- produire des rapports de masse et du reporting BI,
- permettre l'exploration et l'analyse des données descriptives des processus,
- réaliser des tableaux de bord et des systèmes intégrés de pilotage,
- posséder des outils avancés de fouille de données (*Datamining*) (corrélations et prédictions)

3.1 Reporting

Le reporting de masse et le reporting BI doit fournir une vision opérationnelle de l'activité. Il s'adresse à des managers et à leur équipes. Pour les managers, c'est un outil de management opérationnel (il s'agit plutôt de rapports numériques). Pour le directeur commercial par exemple, il doit permettre de donner le bilan des ventes par responsable de secteur, repérer qui est en retard par rapport à ses objectifs, alors que pour le responsable de production, il doit donner un état des ventes par produit, un état des stocks et un état de la production. Pour les équipes de production, c'est une aide quotidienne à leur activité (qu'elles sont les commandes à préparer ?). Pour les commerciaux c'est un outil de suivi de leur activité (classement des ventes par région/distributeur). Il s'agit d'une vision a posteriori (basée sur des données existantes), les besoins étant connus a priori.

3.2 Exploration et analyse

L'exploration et l'analyse demandent de pouvoir naviguer dans l'information depuis le niveau le plus agrégé jusqu'au niveau le plus fin sur un ensemble important d'axes d'analyse (vision pyramidale). On veut pouvoir identifier à quel endroit (produit, secteur, période), les objectifs ne sont pas atteints, émettre des hypothèses, les valider ou les invalider. Il s'agit d'une vision *a posteriori* (basée sur des données existantes). Contrairement aux rapports et tableaux de bord il s'agit d'une vision *sans a priori* (je ne sais pas où est l'information que je cherche). C'est un travail très intuitif, en interaction avec le système, qui nécessite des interfaces de qualité et des temps de réponse instantanés pour ne pas nuire à l'intuition.

3.3 Tableaux de bord et pilotage

Un tableau de bord doit fournir une vision globale et synthétique de l'activité de l'entreprise souvent en comparaison avec des objectifs pré-définis. Il s'appuie sur la métaphore du cockpit de pilotage qui affiche des indicateurs clés (KPI : *Key Performance Indicator*, voir cours sur la CRM) plutôt que des indicateurs simples. Le pilotage peut être vu comme un sous-ensemble du reporting mais, les indicateurs représentés sont souvent plus composites que les indicateurs utilisés pour le reporting ou l'analyse, la navigation se fait entre écrans plutôt que dans les données (pré-câblage) et la notion d'objectifs (suivi temporel ou comparaison par rapport à un objectif) est omniprésente.

3.4 Fouille de données

Le *datamining* fait appel à des plateformes spécifiques de calculs statistiques, l'ensemble des données doit pouvoir être facilement accessibles, y compris au niveau de détail le plus fin et sans *a priori* sur leurs structures.

4. Les grandes étapes de conception d'un data warehouse

4.1 Première étape : spécification d'un *data mart*

1. La première étape du déploiement d'un SID est **l'analyse de l'expression des besoins des utilisateurs**. Cette expression du besoin va commencer par la **sélection du processus à analyser**. On ne fait pas référence ici à un service ou une fonction d'une organisation tel que le service de vente ou le service marketing ou le service facturation, mais au processus de vente, qui implique les clients, les produits, les canaux de distribution, etc. L'activité décisionnelle s'appuie en effet sur des **mesures** déterminées par des corrélations et des consolidations sur des ensembles de données définies **indépendamment des modalités actuelles du fonctionnement** de l'entreprise et qui correspondent à ce processus. Ce sont par exemple des volumes de ventes calculés pour certaines lignes de produit, sur une période donnée et réalisées dans une région donnée. L'expression de ces besoins correspond à une cohérence sémantique de ce que l'on cherche à mesurer. La forme concrète de l'expression de ces besoins s'exprime souvent par la présentation des **rapports** que les utilisateurs voudraient obtenir.
2. La seconde étape correspond à **la définition du grain du processus**. Il définit, pour chacune des données utilisées pour la génération des rapports, le niveau de précision nécessaire le plus fin. Si les mesures à réaliser portent sur une corrélation entre des produits, des périodes, des lieux de distribution, on définira le niveau le plus précis dont on a besoin pour chacune de ces *entités* (par exemple, une entité produit unitaire, un jour, une ville).
3. La définition du grain et des besoins d'agrégation et de consolidation nécessaires pour réaliser les mesures nous amène ensuite à choisir les **dimensions** qui s'appliquent à chaque ligne de la **table de faits**. On peut en effet définir chacune de ces mesures comme une association des différentes entités qui la définissent. Dans la pratique, cela peut se faire en répondant à la question « comment les gestionnaires décrivent-ils les données qui résultent du processus concerné ? » .
4. Une fois le grain et les dimensions définies on précise alors les **faits numériques** qui vont renseigner chaque ligne de la table de faits en répondant à la question « que mesurons-nous ? ».

A la fin de ces premières étapes on a obtenu un **Modèle Conceptuel de Données (MCD)** qui correspond à un des processus analysé pour répondre à un des besoins exprimés. On le nommera **data mart**. Le MCD est une intégration de l'ensemble des vues spécifiques de chaque utilisateur dans une description qui élimine toute redondance. C'est donc une mise en forme intégrée des points de vues des utilisateurs. Il fait abstraction de toute considération liée à l'organisation ou à la technique.

On peut répéter cette étape pour d'autres besoins correspondant soit au même processus, soit à d'autres processus. On obtiendra alors plusieurs *data marts* qui seront rassemblés pour concevoir et spécifier ce qui sera la base de données : le **data warehouse**.

4.2 Deuxième étape : spécification d'un système fonctionnel

Une fois les modèles conceptuels définis, on va entrer dans une phase de spécifications fonctionnelles qui va permettre de passer des modèles conceptuels aux modèles logiques et de définir la gestion des flux de données opérationnelles qui vont venir les alimenter.

1. Le **Modèle Logique des Données (MLD)** se déduit du MCD, mais en tenant compte des contraintes d'organisation des données en rapport avec la technologie d'implémentation. Il tient compte en particulier des objectifs d'optimisation des volumes et des temps de réponse. Il ne peut donc pas être intégralement conforme aux normes qui régissent le MCD, ce qui permet de parler à son sujet de dé-normalisation.
2. Le **Modèle Physique des Données (MPD)** décrit les structures de données telles qu'elles sont enregistrées sur les supports physiques. C'est le modèle final et sa structure dépend étroitement de l'environnement matériel et logiciel d'exploitation des bases de données.

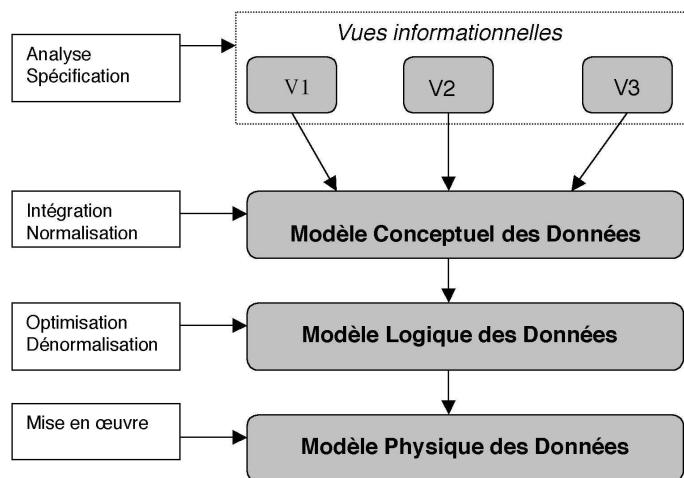
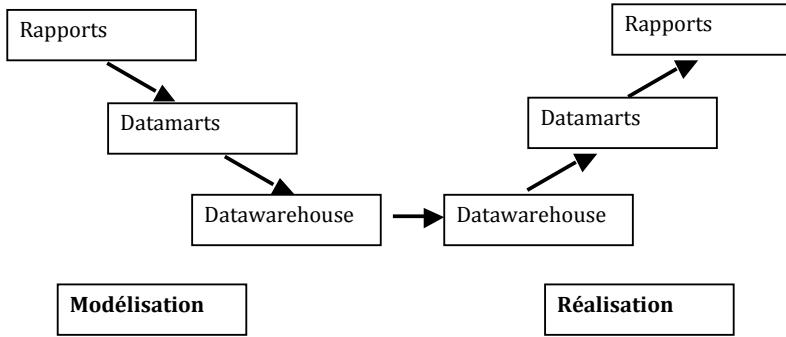


FIGURE 1.4 – Les trois niveaux de modélisation des données (d'après Le Projet Décisionnel, Jean-Marie Gouarné, Eyrolles Edt. 1998, pp. 35)

3. Les spécifications des modèles achevées il faudra définir précisément les spécifications de l'alimentation de ceux ci (procédure ETL) On définira ici les tests d'intégrité sur les données et fonctionnels pour l'alimentation de l'operating *data store* et du *data warehouse*. On définira également les modèles d'implémentation des bases de données pour le *data warehouse*, l'operating *data store* et le *data mart* (serveurs OLAP introduits ci-dessous).

4.3 Troisième étape : spécification des outils d'éditions de rapports, d'interrogation et de requêtage

La dernière étape consiste à spécifier et programmer les outils qui permettront de produire les rapports attendus, paramétrés ou non, ainsi que ceux qui permettront de naviguer dans les *data marts*, piloter des tableaux de bord, accéder aux données à des fins de fouille de données.

FIGURE 1.5 – Le cycle de conception d'un *data warehouse*

4.4 Les différents types de serveurs : ROLAP, MOLAP, HOLAP

Du point de vue utilisateur, seul le principe d'analyse en ligne (OLAP) est important indépendamment de la façon dont les données sont stockées. Cependant, l'architecture physique et l'implémentation des serveurs OLAP doivent tenir compte des problèmes de stockage de données. On trouve :

- Les serveurs OLAP relationnels (ROLAP) : ce sont des serveurs intermédiaires qui se situent entre un serveur d'arrière plan (*back-end*) et un outil client frontal (*front-end*). Ils utilisent un SGBD relationnel et des extensions OLAP (middleware). Les serveurs ROLAP ont des modules pour optimiser chaque DBMS back end, implémentent des logiques d'agrégation, et des outils et services additionnels. Ils supportent mieux les facteurs d'échelle que la technologie MOLAP. Microstrategy et Informix (Metacube) par exemple ont adopté l'approche ROLAP.
- Les serveurs multidimensionnels OLAP (MOLAP) : ces serveurs supportent les vues multidimensionnelles des données grâce à des moteurs de stockage multidimensionnels à base de tableaux. Ils projettent les vues multidimensionnelles directement sur les structures de tableaux en cubes. Par exemple, Essbase (Arbor) est un serveur MOLAP. L'avantage d'utiliser des cubes de données est qu'ils permettent l'indexation rapide de données agrégées pré-calculées. Les temps de stockage peuvent être longs, en particulier si les données sont parcimonieuses, il faut alors utiliser des techniques de compression spécifiques.
- Les serveurs hybrides OLAP (HOLAP) : dans cette approche, on combine les technologies ROLAP et MOLAP, en bénéficiant du passage à grande échelle de ROLAP et des calculs rapides de MOLAP. Par exemple, un serveur HOLAP peut accepter de grands volumes de données détaillées stockées dans une base de donnée relationnelle, alors que les agrégats sont stockés dans une base MOLAP séparée. Microsoft SQL Server est un exemple de HOLAP.
- Les serveurs SQL spécialisés : pour répondre à la demande croissante de calcul dans les bases de données relationnelles, certains éditeurs de SGBD et de *data warehouse* ont implémenté des serveurs SQL spécialisés qui fournissent des langages de requêtes avancés sur les modèles flocon et étoile.

Les modèles en flocon et en étoile seront définis au chapitre 2 section 4., et les techniques d'optimisation, d'indexation et de requêtage seront étudiées au chapitre 4, section 6.4.

Chapitre 2

Définition des Modèles Conceptuels de Données Décisionnels

1. Introduction

Le modèle conceptuel de données décisionnel est celui qui reflète la **mise à disposition** ou la **diffusion** des données. Les modèles dérivés du MCD, le MLD et le MPD, sont ensuite élaborés en liaison étroite avec la technique selon un démarche fortement tributaire des produits.

1.1 Principes généraux

Les normes d'intégration du MCD reposent sur les principes suivants :

1. Compte tenu de la nature consultative et non transactionnelle des applications, la structure des vues externes se déduit directement des requêtes des utilisateurs et non des connexions possibles entre les entités ;
2. A l'intérieur d'un domaine, il existe un ou plusieurs sous ensembles de vues liées entre elles par certains critères de cohérence sémantique et structurelle. C'est sur l'identification et la validation de ces sous-ensembles, appelés **contextes**, que repose toute la démarche de construction du MCD ;
3. Une requête décisionnelle a pour objet d'établir un rapprochement non programmé entre entités conceptuelles plus ou moins nombreuses. De ce fait, les résultats attendus sont systématiquement déterminés par des **associations**. La structure des vues reflète celle des associations possibles. Chaque vue a pour élément central une association autour de laquelle gravitent deux ou plusieurs entités, et correspond à une représentation des informations sous forme d'un **tableau** à deux ou plusieurs dimensions ;
4. La liste exhaustive des requêtes n'est jamais figée. Celle des vues qui en découle ne l'est donc pas non plus. La normalisation du MCD doit permettre d'anticiper et d'intégrer automatiquement dans chaque contexte le plus grand nombre possible de vues probables d'après la structure des vues connues ;
5. Entre deux entités d'une même vue, il doit exister un chemin et un seul de navigation sémantique, qui doit être le plus court possible.

1.2 Le couple entité-relation

Le formalisme universellement adopté pour les MCD est fondé sur le couple entité-association.

- Une entité est une « chose » ou une « idée » qui peut être identifiée comme sujet ou objet dans l'univers du discours lié au projet. Chaque entité est susceptible de posséder des **caractéristiques** ou **propriétés**.
- Une association est un lien ou un regroupement impliquant une ou plusieurs entités.

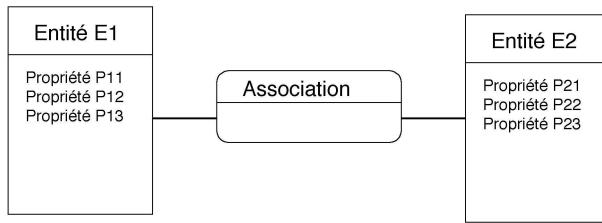


FIGURE 2.1 – Entités et associations

Une entité peut, par exemple, représenter un employé, un contrat, un produit, un véhicule, un établissement. Une entité est une représentation abstraite, qui doit être distinguée des occurrences d'entités.

L'entité « Employé » peut, par exemple, avoir comme propriétés un « salaire », un « matricule », une « couleur de cheveux ». L'entité véhicule peut avoir un « numéro d'immatriculation », une « puissance fiscale », un « nombre de places ». Si les « employés » utilisent des « véhicules » on peut dire que le « Véhicule » est un attribut de « Employé », ou bien qu'il existe une **association** entre ces deux entités. Si l'on s'intéresse à la date d'affectation d'un véhicule à un employé, la propriété « date » constitue l'association entre ces deux entités (cette date d'affectation ne peut être caractéristique d'aucune des deux entités).

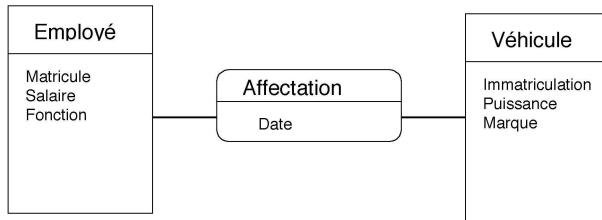


FIGURE 2.2 – Association Employé-Véhicule

Il n'existe pas de règle générale d'assemblage de propriétés.

La normalité d'un modèle ne découle pas des données elles-mêmes mais de l'usage qu'on en fait.

1.3 Les limites du modèle normal (3NF, CODD) pour les SID

Le développement des bases de données pour les systèmes d'information opérationnels (SIO) a donné lieu à la définition de normes qui président dans chaque modèle à la détermination des entités et associations. On verra qu'on ne peut pas simplement transposer ou « dé-normaliser » ces modèles pour définir un MCD, car les principes qui régissent l'un et l'autre de ces modèles ne répondent pas aux mêmes objectifs.

1.3.1 Normalisation des entités

Dans l'exemple précédent on dira que l'entité « Employé » est normalisée si tout employé est décrit par les propriétés « Matricule », « Salaire » et « Fonction ». Supposons que l'on ajoute à ces caractéristiques les diplômes et qualifications que possède un employé. Si l'un d'entre eux possède un MBA et un certificat d'études, tandis qu'un autre possède un permis de conduire, une licence en droit, un BEP de comptable et une maîtrise de philosophie, l'entité générique « Employé » possède une structure variable, puisque le nombre de qualifications change avec chaque occurrence. Pour la normaliser on va extérioriser cette propriété, en créant une entité « Qualification » et en établissant une association (1:n) entre « Employé » et « Qualification ». Ce principe valable pour les SIO reste valable pour les SID, et on retiendra donc ce premier principe de normalisation :

Une entité est normalisée si toutes ses occurrences sont décrites par les mêmes caractéristiques.

1.3.2 Identifiant des entités

Chaque entité doit posséder un **identifiant** unique. C'est une propriété ou un groupe de propriétés qui permet de distinguer de façon unique chaque occurrence. Dans beaucoup de SI l'identifiant est une propriété artificielle qui a été créée pour identifier l'entité (un numéro matricule, par exemple), et qui n'a pas d'autre signification. Certaines propriétés destinées à servir d'identifiant peuvent accessoirement contenir des éléments d'information codée. Par exemple l'immatriculation d'un véhicule indiquait le département de domiciliation du propriétaire ou le numéro de sécurité sociale donne le sexe, l'année et le département de naissance.

Nous verrons que de nombreuses entités sont créées (synthétisées) dans un *data warehouse* à partir des entités lues sur les systèmes opérationnels. Il faudra donc prévoir des fonctions spécifiques pour attribuer à ces entités des identifiants adaptés à leur exploitation dans un *data warehouse*.

1.3.3 Dépendance fonctionnelle

Il existe une dépendance fonctionnelle entre G_1 et G_2 si, à toute valeur de G_1 on ne peut associer qu'une seule et même valeur de G_2 à un instant donné.

Par exemple, on connaît le nombre de jour d'un mois si on connaît le numéro du mois et le numéro de l'année, dans le cas de février. On peut donc dire que le nombre de jours d'un mois est en dépendance fonctionnelle du couple $[numéro_année, numéro_mois]$. De même le montant de la taxe d'habitation est dépendant de la surface et du type de logement, de sa localisation, et de sa région. On distingue les dépendances fonctionnelles directes, si la valeur de G_1 détermine immédiatement celle de G_2 . Si G_1 détermine G_2 , qui à son tour détermine G_3 , il y a dépendance fonctionnelle indirecte entre G_1 et G_3 .

1.3.4 Normalisation 3FN

Une entité est dite 3FN si :

1. elle est normalisée ;
2. toutes les propriétés sont en dépendance fonctionnelle directe de l'identifiant ;
3. il n'existe dans l'entité aucune autre dépendance fonctionnelle que celle qui émane de l'identifiant.

1.3.5 Pourquoi ne peut-on pas conserver intégralement cette normalisation ?

Ce troisième principe de normalisation ne peut pas toujours être respecté dans le cas d'un SID. Par exemple, une entité « Employé » caractérisée par l'identifiant « Matricule » et les propriétés « Nom », « Prénom », « Fonction » et « Nom de Service », qui est 3FN, ne l'est plus si on lui ajoute une propriété supplémentaire « Nom du chef de Service », car il existe une DF entre « Nom du Service » et « Nom du chef de Service » (figure 2.3). Pour rester 3FN il faudrait créer une nouvelle entité « Service » et des associations avec « Employé » (l'une « est attaché à » et l'autre pour les chefs « dirige »)

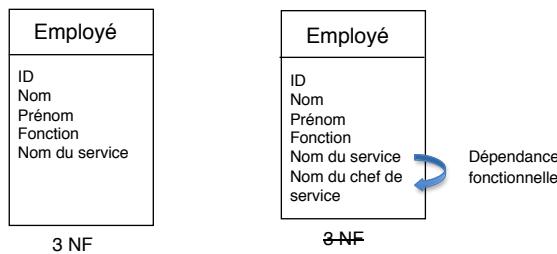


FIGURE 2.3 – L'ajout du nom du chef de service dénormalise l'entité Employé)

Le fait de créer cette nouvelle association a pour conséquence de relier les entités « Service » et « Employé » par deux chemins différents (figure 2.4). Ce qui est naturel et nécessaire dans le cas d'une base de

données d'un SIO devient source d'erreur dans le cas du MCD d'un *data mart*. En effet, l'objectif d'un SIO est d'optimiser la gestion des transactions, ce qui justifie la construction de leur modèle par les dépendances fonctionnelles. Les transactions sont généralement exécutées par des programmes d'application et ne sont donc jamais improvisées à l'initiative des utilisateurs. Le rôle du SIO est fondamentalement celui d'un automate de contrôle. Il est à chaque instant dans un certain état. Il passe d'un état à un autre par l'intermédiaire d'une transition, qui reflète un flux réel du Système Opérant.

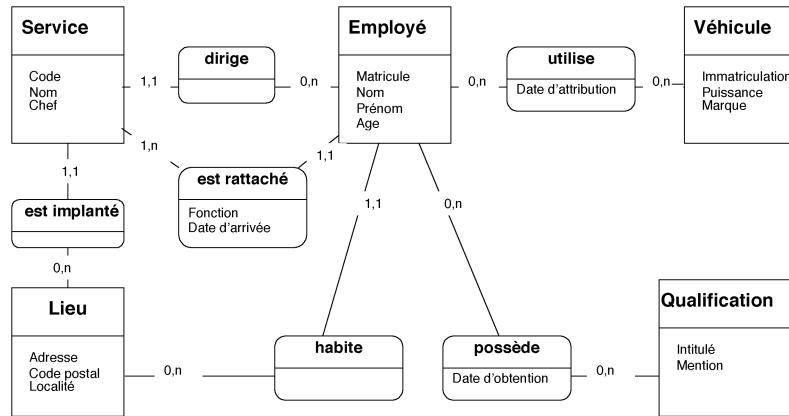


FIGURE 2.4 – Modèle conceptuel de données en 3^e Forme Normale (d'après Le Projet Décisionnel, Jean-Marie Gouarné, Eyrolles Edt. 1998, pp. 41)

La normalisation des systèmes opérationnels ainsi conçue est liée à la poursuite des objectifs suivants :

1. élimination des redondances de données, qui induisent des problèmes de cohérence et de mise à jour ;
2. performances à l'exécution des mises à jour transactionnelles ;
3. simplification des contrôles d'intégrité référentielle.

Ce n'est pas le cas pour les SID, dont l'objectif est de mesurer des cumuls, des ratios, des tendances etc. La qualité et l'unicité des résultats de ces calculs ne peuvent plus être garanties s'il y a plusieurs chemins ou des boucles qui relient les entités sur lesquels ils sont effectués. Le modèle des données ne peut donc pas se construire indépendamment de la connaissance des traitements que l'on veut rendre possibles, y compris ceux qui ne sont pas programmés à l'avance. La construction du MCD ne peut donc pas reprendre intégralement les principes énoncés ci-dessus et nécessite la définition d'un modèle spécifique.

2. Vues, Faits et Dimensions

Comme nous venons de le voir, l'approche « classique » d'élaboration du modèle de la base de données n'étant plus applicable ici, nous devons trouver une autre méthode pour construire le MCD. Le principe qui est retenu est celui qui a été énoncé à plusieurs reprises : on va construire le modèle conceptuel du système décisionnel à partir de l'étude des besoins des utilisateurs. C'est l'analyse et le regroupement de ces besoins en fonction de leur porté sémantique qui va servir de guide. Ceux-ci s'expriment en général assez facilement sous la forme de requêtes. Considérons par exemple la requête suivante :

« Quels ont été les frais de déplacement et le kilométrage des commerciaux de la région Rhône-Alpes ayant des véhicules de 12 à 14 CV en Juillet 1996 ? »

Isolée de son contexte, une telle requête ne nous indique pas le sens et la composition de chacune des entités invoquées. Nous ne savons pas si « commerciaux » sont des occurrences d'une entité « Commercial » ou bien si c'est une valeur possible d'une propriété de l'entité « Employé ».

La connaissance du **domaine**, c'est à dire des entités fondamentales du métier de l'utilisateur, est donc nécessaire pour une analyse correcte de la requête.

Cette requête associe les quatre entités suivantes : « Employé », « Véhicule », « Région », « Mois ». Les résultats demandés sont « frais de déplacement » et « kilométrage ».

Les quatre entités ne sont pas invoquées de la même manière. « Région » et « Mois » sont indiqués chacune par leur **propriété identifiante** : il n'y a qu'un seul mois de juillet 1996, et une seule région Rhône-Alpes. En revanche les entités, « Employé » et « Véhicule » sont sélectionnées par des **propriétés descriptives**, respectivement la « fonction » et la « puissance ».

La structure de la requête analysée détermine une **vue** illustrée par la figure 2.5, qui s'exprime littéralement sous la forme :

frais de déplacement, kilométrage

- / Employé (fonction)
- / Véhicule (puissance)
- / Région
- / Mois

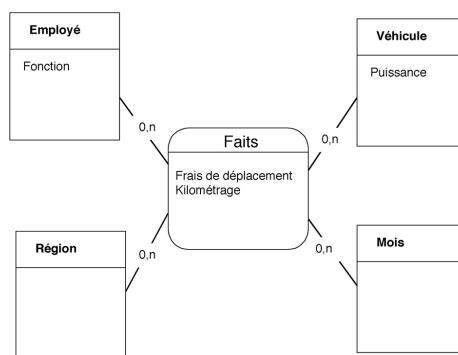


FIGURE 2.5 – Vue Frais/Employé/Véhicule/Région/Mois

Les cardinalités (0,n) ont ici une valeur par défaut, qui peut être corrigée par l'analyse. Dans la pratique on autorise très souvent des cardinalités de type (0,n), ce qui signifie que de nombreuses associations n'auront pas d'existence réelle. Par exemple tous les « employés » n'auront pas de « frais de déplacement » par Région/Mois/Véhicule, parce que certains employés ne se déplacent pas, d'autres ne vont pas dans toutes les régions ou ne se déplacent pas tous les mois, etc. C'est une première manifestation de la construction de **matrices creuses** qui peuvent poser des problèmes d'optimisation tant du point de vue du stockage de l'information, que du traitement des données.

Les propriétés centrales, dont la valeur est déterminée par la combinaison des quatre entités sont des **faits** et toutes les autres propriétés sont des **conditions**.

D'une manière générale :

- un **fait**, une **mesure** ou un **indicateur** est une information déterminée par la combinaison de deux ou plusieurs entités, susceptible de constituer le résultat ou un élément du résultat d'une requête.
- une **condition** est une caractéristique d'entité susceptible d'intervenir comme critère de définition d'une requête.

Une vue comporte donc toujours une **association** et **deux ou plusieurs entités**. Tous les faits sont des propriétés de l'association et toutes les conditions sont des propriétés des entités. Une requête implique nécessairement une vue, mais plusieurs requêtes peuvent s'appliquer à la même vue.

2.1 Enrichissement de la vue

Dans la structure précédente on ne peut pas obtenir en regard des frais de déplacement, le nom et le matricule de chaque employé.

Par exemple, si on reformule la requête précédente par :

Je veux la liste des noms des commerciaux de la région Rhône-Alpes ayant des véhicules de 12 à 14 CV avec, pour chacun, les frais de déplacement, le kilométrage et la marque du véhicule, pour Juillet 1996 ?

la vue appropriée devient :

```
frais de déplacement, kilométrage
/ Employé (nom, fonction)
/ Véhicule (marque, puissance)
/ Région
/ Mois
```

Ce qui amène à enrichir la vue selon le schéma de la figure 2.6 :

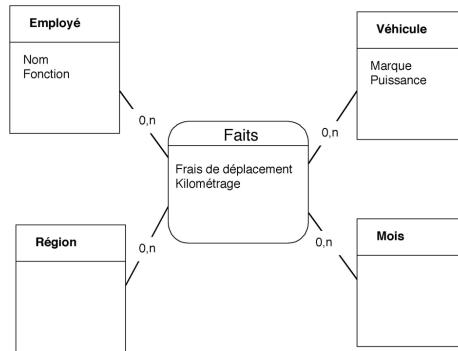


FIGURE 2.6 – Variante enrichie de la figure 2.5

Bien que demandés comme résultats, *Nom* et *Marque* figurent dans les *conditions* et non dans les *faits*.

Ce n'est pas parce qu'une information est demandée en réponse à une requête qu'elle constitue un fait. **Un fait est un résultat issu d'une association.**

D'où la règle suivante :

Règle : Si une propriété d'entité non expressément spécifiée comme critère de sélection dans une requête apparaît dans la liste des résultats demandés, il faut l'intégrer dans la vue comme s'il s'agissait d'une condition.

Une façon de distinguer un fait d'une condition est que la plupart du temps un fait est une valeur numérique calculée ou une mesure, alors qu'une condition est une valeur descriptive textuelle ou pouvant être assimilée à une valeur textuelle même si elle apparaît sous forme numérique. Par exemple une taille, bien que ce soit une valeur numérique, est une condition parce qu'elle se comporte davantage comme une description textuelle que comme une mesure numérique. La taille est une constante discrète décrivant un produit spécifique. Si un descripteur est une mesure qui peut prendre de nombreuses valeurs et participer à des calculs, c'est très certainement un fait. Si c'est une valeur plus ou moins constante qui peut participer à des contraintes, c'est alors une condition. Par exemple, le coût standard d'un produit ressemble à une condition constante du produit, mais il peut être changé si souvent que nous pouvons décider qu'il s'agit d'un fait mesuré (ce coût est en général le résultat d'un calcul sur les différents éléments qui le constituent). Une propriété peut cependant servir aussi bien comme élément de résultat que comme critère de sélection (de filtre) (on ne préjuge pas de la façon dont on va les exploiter). Par exemple on veut sortir des cumuls, avoir les 10 meilleurs commerciaux par ordre alphabétique, etc. . . .

Il existe enfin des faits que l'on peut qualifier d'*implicites* quand ils n'apparaissent pas expressément comme des propriétés nommées dans les vues. Ces faits sont des éléments de résultats pour les requêtes comportant des *comptages*. Par exemple :

«Combien de commerciaux de la région Rhône-Alpes se sont-ils déplacés avec des véhicules de 12 à 14 CV en juillet 1996 ?»

qui peut se ré-interpréter par :

«Pour combien de commerciaux de la région Rhône-Alpes existe-t-il un kilométrage et/ou des frais de déplacement non nuls avec des véhicules de 12 à 14 CV en juillet 1996 ?»

2.2 Relier les faits et les dimensions

Intuitivement une vue correspond à une *matrice* dont chaque *dimension* est décrite par une entité et dont le contenu est décrit par l'association de ces entités (table 2.1). De la même façon que la combinaison de valeurs numériques pour déterminer la position d'un point dans l'espace, les combinaisons de conditions sont les coordonnées qui déterminent des valeurs de faits . On déterminera donc autant de vues que l'on voudra construire des relations associatives entre des composantes dimensionnelles différentes.

TABLE 2.1 – Représentation relationnelle de l'association Faits-Dimensions

Employé		Véhicule		Mois	Région	...	Faits
Nom	Fonction	Marque	Puissance				

La figure 2.7 est une métaphore de ce que pourrait être une présentation tabulaire des données de l'exemple en quadri-dimensionnel, avec seulement deux régions, quatre employés, trois véhicules et trois mois. Ce type de présentation est un bon exemple de restitution d'états mais certainement pas un document de modélisation de données.

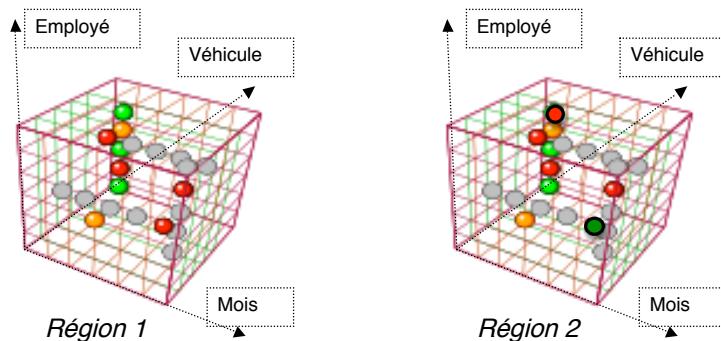


FIGURE 2.7 – Présentation tabulaire en quadri-dimensionnel de l'exemple 2.5

3. Intégration des vues

L'objectif du SID étant d'offrir une structure informationnelle intégrée et non de préparer l'exécution d'un jeu pré-défini de requêtes, les diverses vues du domaine d'analyse doivent être combinées dans le Modèle Conceptuel de Données selon des principes d'assemblage répondant à deux objectifs :

- ne jamais introduire de chemin sémantique complexe ou ambigu ;
- rester capable d'intégrer de nouvelles vues ou de modifier une vue existante sans remise en question de la structure générale du modèle.

La réalisation de ces objectifs s'appuie sur deux notions : la notion de **contexte** et celle de **hiérarchie**.

3.1 Notion de contexte

La consolidation directe de toutes les vues d'un MCD construites à partir de l'expression de tous les besoins des utilisateurs produirait inévitablement une structure trop évolutive : l'introduction de chaque nouvelle vue aurait un impact sur l'ensemble du modèle, dont la complexité augmenterait dans le temps, avec des conséquences défavorables sur les coûts de maintenance et sur la navigation.

On introduit donc un niveau intermédiaire de modélisation entre la vue et le domaine, appelé **contexte**.

Définition : Un contexte est un ensemble de faits et de dimensions assemblés selon des critères sémantiques formels de cohérence.

Un contexte est, comme une vue, caractérisé par une association unique, groupant tous les faits relevés dans les vues. Cependant, les entités qui gravitent autour ne sont pas toutes sur le même plan, sachant que certaines d'entre elles peuvent être liées par des **dépendances fonctionnelles** de type **hiérarchique**.

Soient les quatre vues suivantes :

- (1) marge / Client / Région / Produit / Jour
- (2) revenu / Pays / Mois / Marque
- (3) ventes / Canal / Gamme / Trimestre
- (4) revenu / Marque / Canal / Mois

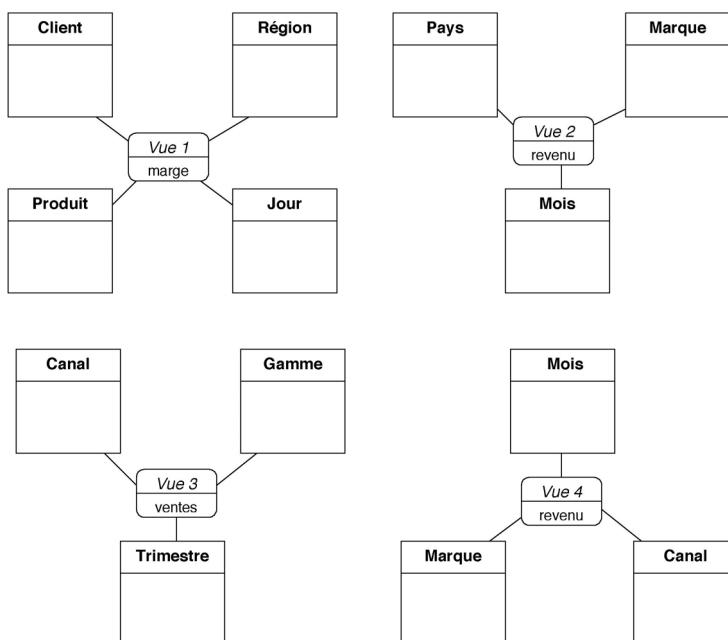


FIGURE 2.8 – Quatre vues indépendantes

En opérant un regroupement superficiel entre ces vues, on détecte deux sortes d'éléments de rapprochement :

1. certaines informations (entités ou faits) se retrouvent dans plusieurs vues,
2. certaines entités, appartenant à des vues différentes, sont fonctionnellement liées.

Nous verrons plus loin les règles de compatibilité qui permettent de décider dans quelle mesure plusieurs vues peuvent appartenir au même contexte. Pour l'instant, considérons ces quatre vues comme intégrables. Le contexte correspondant à leur intégration comporte une association porteuse des faits :

marge, revenu, ventes

Il comporte également dix entités distinctes.

3.2 Hiérarchies

Parmi les entités de cet exemple, certaines sont rattachées à d'autres par des liens d'appartenance ou de groupement hiérarchique.

Certains des chemins sont *a priori* évidents (jour, mois, année), d'autres doivent être repérés par une analyse précise du vocabulaire des utilisateurs.

On admet ici qu'après cette analyse on a identifié les trois hiérarchies symbolisées dans la figure 2.9. Elles représentent pour l'utilisateur des **chemins de consolidation** d'indicateurs.

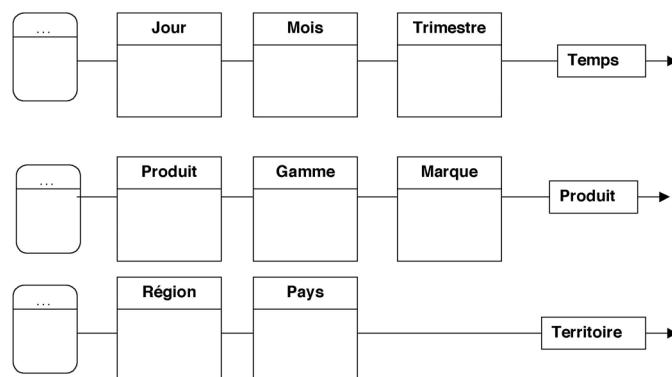


FIGURE 2.9 – Exemples de hiérarchies

Dans une simple vue, chaque entité correspond à une dimension de la matrice des résultats. Cependant dans un contexte, le nombre de dimensions peut être inférieur au nombre d'entités de toutes les vues intégrées, car certaines peuvent être à des niveaux de détail différents dans une même dimension.

L'identification conceptuelle des hiérarchies n'est pas toujours aussi évidente que dans cet exemple. Toutes les consolidations rencontrées au hasard des requêtes ne correspondent nécessairement pas à des chemins et à des niveaux hiérarchiques structurels. Dans une hiérarchie, chaque niveau est représenté par une entité. Une entité est un « objet » ayant une existence, une identité et des caractéristiques propres dans le métier de l'utilisateur, or un critère de regroupement ne correspond pas toujours à une entité.

Par exemple, si l'on fait le cumul des ventes de « produits électroménagers » sur une tranche d'âge de la clientèle, il y a bien une hiérarchie sur les produits, qui se regroupent dans la catégorie « produits électroménagers » concernée, alors que sur l'autre dimension, on applique une restriction sur une propriété, « l'âge du client ». Si l'analyse des autres requêtes confirme bien l'existence d'un concept « catégorie de produits » identifiable et possédant des caractéristiques descriptives, ce concept correspond bien à un niveau *structurel* de consolidation et doit donc apparaître comme une entité dans un chemin hiérarchique. En revanche, si les tranches d'âge de la clientèle sont toujours invoquées sous la forme “*client âgé de A₁ à A₂*” et ne sont jamais définies autrement que par leurs deux bornes, ces tranches n'ont pas d'existence perçue en tant qu'entité ; elles ne correspondent qu'à des conditions de sélection sur une entité “*Client*”. En résumé, il faut pouvoir NOMMER une consolidation de façon commune à tous les utilisateurs pour qu'elle corresponde à une entité, qui existe indépendamment de son rôle de nœud de consolidation. Dans certains cas, un intervalle peut être une entité si l'analyse de l'ensemble des requêtes confirme l'existence d'un découpage stable (par exemple, l'existence d'une classe *senior*).

En résumé, l'existence d'une entité se justifie par le fait qu'elle est porteuse d'informations qui ne se retrouvent dans aucune autre entité du même contexte.

Une hiérarchie est une dimension d'un contexte. Il s'agit d'un cas fréquent, à tel point que ces deux notions sont souvent confondues (voir figure 2.9). Cependant, une même dimension peut comporter plusieurs chemins hiérarchiques.

Si une entité A est rattachée à une entité B et une entité C, et qu'il n'existe pas de rattachement hiérarchique entre B et C, alors il existe deux chemins de consolidation possibles pour A. En dehors des hiérarchies temporelles, les hiérarchies multiples sont également fréquentes dans les dimensions liées par exemple à *l'organisation*, à la *clientèle* ou aux *produits*.

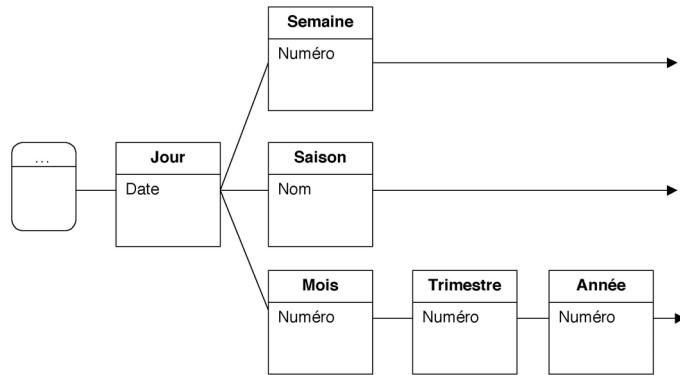


FIGURE 2.10 – Hiérarchies périodiques multiples

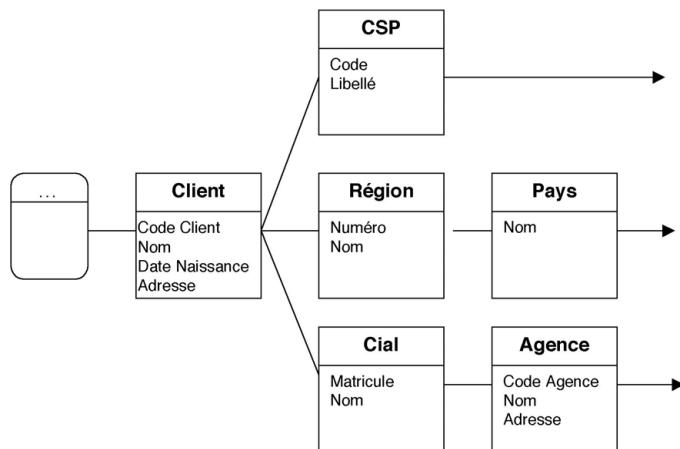


FIGURE 2.11 – Hiérarchies multiples sur le « Client »

3.3 Synthèse de contextes

La première étape de l'intégration d'un contexte consiste à :

- faire l'inventaire de tous les liens de dépendance entre entités,
- regrouper par dimensions les entités liées par des associations composition/appartenance,
- nommer les dimensions.

Les dimensions que l'on peut nommer facilement et qui ne reprennent pas le nom d'une des entités sont généralement des dimensions *robustes*. Par exemple la dimension « Temps » est caractérisée par « Jour, Mois, Trimestre ». En revanche si on donne à la dimension « Produit » le nom de son entité de base cela indique que peut-être « Gamme » et « Marque » ne sont que des niveaux conventionnels de regroupement de produits, et que la structure de cette dimension peut varier à terme.

Compte tenu de ces observations, la combinaison des quatre vues de notre exemple produit le *contexte* dont la définition littérale est :

```

Activité : marge, revenu, ventes
/ Canal      : Canal
/ Client     : Client
/ Territoire : Région   - Pays
/ Temps      : Jour      - Mois    - Trimestre
/ Produit    : Produit   - Gamme   - Marque

```

Représenté sur la figure 2.12 qui comporte 5 dimensions pour 10 entités.

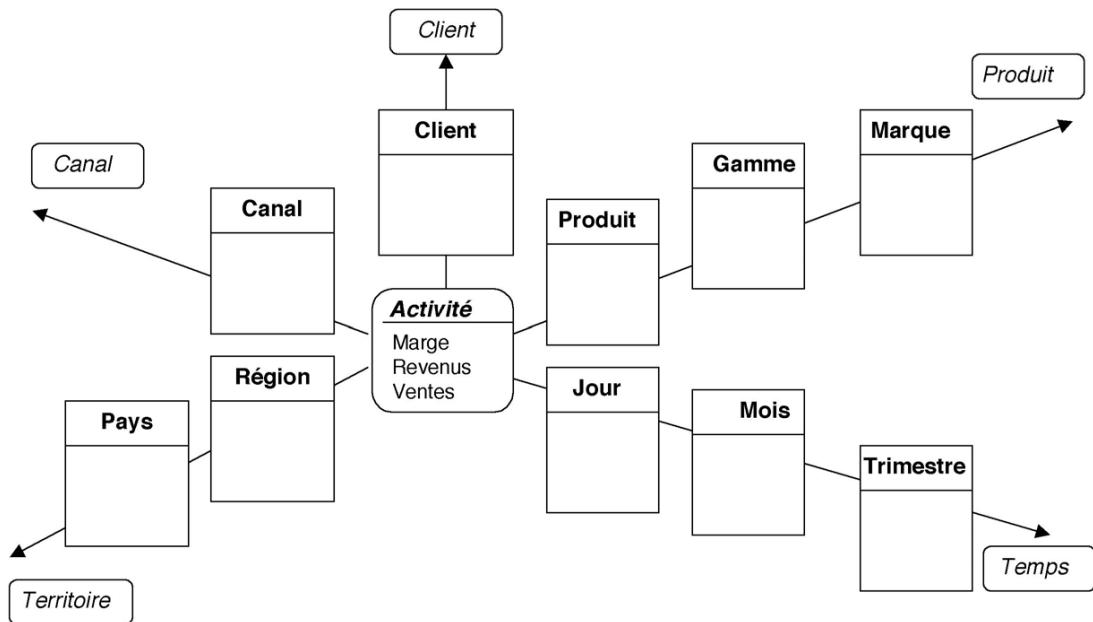


FIGURE 2.12 – Contexte « Activité commerciale »

Attention à ne pas confondre hiérarchies et dimensions, même si elles se confondent souvent.

Les dimensions que l'on rencontre fréquemment :

- les périodes calendaires (grain le plus fin, le jour),
- l'organisation (divisions hiérarchiques de l'entreprise),
- la géographie (découpage territorial des activités),
- l'offre de l'entreprise,
- la clientèle et/ou le marché (avec regroupement par segments ou secteurs économiques),
- les circuits de distribution, la logistique, les modalités de fournitures des biens et des services,
- les contrats, opérations, transactions (unités élémentaires d'activités susceptibles d'être groupées par catégories),

Les dimensions ci-dessus sont typiques de la démarche initiale de développement de *data warehouse* par des services marketing et commerciaux. On trouve également des dimensions courantes dans le domaine de la production :

- les technologies ou procédés de fabrication,
- les mesures et contrôles,
- la matière première,
- l'origine des composants ou pièces détachées,
- la sous-traitance impliquée dans le processus de fabrication,
- le conditionnement et les modalités de livraison,
- les conditions d'utilisation des produits ...

La dimension n'est pas à chercher *a priori*. Elle est détectée à partir de la définition des entités et de leurs éventuels liens de composition.

Un aspect utile des contextes est leur effet *multiplicateur de vues*. En effet, à partir du contexte construit précédemment, on peut proposer d'autres vues envisageables :

marge	/	Canal	/	Pays	/	Produit	/	Trimestre		
revenu	/	Région	/	Mois	/	Produit	/	Client	/	Canal
ventes	/	Client	/	Gamme	/	Jour				
revenu	/	Marque	/	Gamme	/	Mois	/	Région		

Cette multiplication des vues répond à l'objectif d'*anticipation des requêtes*. A partir d'un petit nombre de *vues initiales*, repérées par l'étude du domaine, on peut en déduire, sans modification du modèle de données, un grand nombre de *vues dérivées*. Il faut cependant que ces vues dérivées correspondent à des informations pertinentes, ce qui implique que les contextes soient intégrés selon des normes précises.

3.4 Normalisation des contextes

Intuitivement : un contexte est cohérent si toutes les vues qu'il autorise ont une signification dans l'univers de l'utilisateur. Un contexte *raisonnable* contient entre 4 et 12 dimensions. Il ne faut cependant pas rejeter les vues dérivées, qui rapprocheraient des variables qui n'ont *a priori* aucun rapport. C'est justement ici que se situe l'une des richesses des SID.

3.4.1 Dépendances et influences

Si dans notre exemple, l'entité Canal correspond aux filières commerciales et si la commercialisation est strictement géographique, (1 distributeur / région) alors cette entité est redondante avec Territoire.

Règle 1 : *Il ne doit pas y avoir de dépendance fonctionnelle entre deux entités appartenant à des dimensions différentes d'un même contexte.*

L'application de cette règle peut poser un problème de définition des dépendances fonctionnelles. Dans un modèle opérationnel de données, les données sont soit dépendantes, soit indépendantes. Dans un univers décisionnel cette notion est plus nuancée. Il existe une infinité de niveaux d'*influence* possibles entre la dépendance fonctionnelle pure et simple et l'indépendance.

Par exemple, un client d'une région s'adresse *a priori* au canal de distribution de sa région, mais rien de lui interdit d'en utiliser un autre. Il y a donc une *influence forte* de la dimension « Clients » sur les dimensions « Canal » et « Territoire ». Il y a donc une proportion écrasante de valeurs nulles dans la vue [ventes / Client / Région / Canal / Jour] (problème des matrices creuses).

EXTENSION : il y a une *influence* de la dimension Client sur les dimensions Canal et Territoire.

Il faut garder à l'esprit que le *data warehouse* est fait pour mettre en évidence des liens de dépendance entre dimensions.

3.4.2 Définition des faits

On ajoute la nouvelle vue :

coût de stockage / Produit / Jour

Intuitivement, on peut douter du bien-fondé de la présence d'un indicateur de *magasinage* dans un contexte à coloration *commerciale*. Mais l'intuition ne suffit pas toujours.

Cette vue n'implique que les dimensions « Temps » et « Produit » qui existent déjà. Cependant, elle apporte un nouveau fait, qui devient disponible dans le contexte pour toutes les autres vues, dont :

coût de stockage / Produit / Jour / Client / Région

Ce qui autorise des requêtes telles que : *Quel a été le coût de stockage du produit X à la date Y pour le client Z ?*

Cette question n'a pas de sens : un coût de stockage n'est pas lié à la vente d'un produit à un client. Cela est dû à ce que le fait « Coût de stockage » n'est pas défini dans les vues impliquant la dimension « Client ». Ce fait n'a donc pas le même comportement que les autres. D'où la règle suivante :

Règle 2 : *Tous les faits d'un contexte doivent être définis d'une manière cohérente pour toutes les combinaisons dimensionnelles de ce contexte.*

3.4.3 Cohérence de grain

Le grain d'une dimension est le niveau le plus fin possible dans cette dimension. L'intégration de chaque nouvelle vue est donc susceptible de modifier le grain sur une ou plusieurs dimensions. Le grain d'un contexte découle de la combinaison des grains de toutes les dimensions. Le grain du contexte de la figure 2.12 est défini par la combinaison *Produit / Jour / Client / Région / Canal*.

Ce grain s'applique à tous les faits. Si les trois indicateurs « marge », « revenu », « ventes », sont présents dans le contexte, cela signifie qu'ils ont tous un sens à tous les niveaux.

Si dans l'exemple, « Marge » n'est définie que par « Pays » et par « Mois », alors que les autres faits le sont par « Région » et par « Jour », il y aura un décalage de grain entre les faits. Ce décalage signifie que les faits n'appartiennent pas tous au même contexte. L'intégration des vues doit donc respecter la règle suivante :

Règle 3 : *Tous les faits d'un contexte doivent être définis pour le grain de ce contexte.*

Un approche simpliste qui consisterait à diviser artificiellement toute entité par le grain le plus fin amène des aberrations et des risques de pollution de la base.

3.4.4 Navigation hiérarchique

Les valeurs associées à une même entité peuvent être consolidées par des chemins différents. Un fait mesuré par jour peut, par exemple, être cumulé en suivant la hiérarchie mois-trimestre-année, ou encore par saison. Des pays peuvent être regroupés par continent ou par zone linguistique. La diversité des hiérarchies possibles doit apparaître dans le MCD, mais à condition de respecter certaines règles.

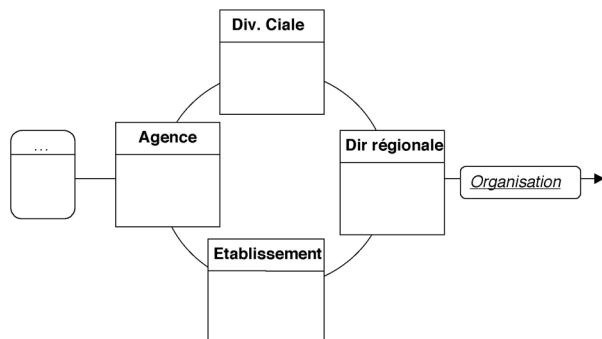


FIGURE 2.13 – Hiérarchie cyclique

Imaginons une entreprise dont l'organisation est la suivante :

- unité élémentaire sur le terrain : agence ;
- division fonctionnelle : chaque agence est rattachée à une division (admin., commercial, ...);
- répartition physique : des agences peuvent cohabiter dans un même établissement ;
- activité : les activités des divisions sont coordonnées par des directions régionales communes auxquelles, par ailleurs, sont rattachés les établissements.

Cette organisation vue comme une dimension d'analyse dans un contexte décisionnel, est représentée par la figure 2.13.

Si les consolidations de faits sont pertinentes aussi bien par filiale que par établissement, les deux chemins de consolidation doivent apparaître dans la dimension « Organisation ». Cependant, le regroupement au niveau régional pose un problème de cheminement. Si, par exemple une agence dépendant d'une division de la région A était logée dans un établissement de la région B, le résultat d'une requête sur la région dépendrait du chemin de consolidation suivi. D'où la règle :

Règle 4 : *Le graphe de chaque dimension doit être acyclique.*

Une solution au problème précédent est de constater qu'il y a un problème de définition du terme « région » ou « direction régionale » entre les utilisateurs. La création de deux concepts distincts permet de lever cette ambiguïté (Fig. 2.14).

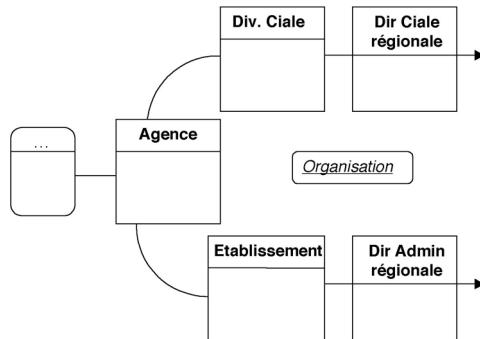


FIGURE 2.14 – Hiérarchie acyclique

D'une manière générale, une dimension multi-hiérrachisée doit avoir une structure strictement arborescente ; les deux hiérarchies ne peuvent avoir de consolidation commune.

4. Modèle relationnel de diffusion

Le MCD ayant été conçu indépendamment de toute contrainte d'implémentation, il va falloir définir le modèle correspondant à sa mise en œuvre opérationnelle (c'est à dire le MLD). Cette mise en œuvre peut s'appuyer sur un modèle de type relationnel.

Etoiles ou flocons ?

La représentation directe d'un contexte dimensionnel dans une base de données relationnelle est un réseau de tables jointes selon un *schéma en flocon*. Dans ce mode de représentation l'association conceptuelle qui contient les faits devient la *table de faits*, et chacune des entités dimensionnelles devient une table distincte.

La table de faits contient en plus des indicateurs significatifs qu'elle comporte par définition, un ensemble de *clés étrangères*, dont chacune assure la liaison avec la table du niveau le plus fin de chaque dimension.

La table des faits est généralement une très grande table, puisqu'elle comporte autant d'enregistrements qu'il existe de combinaisons pertinentes entre les tables dimensionnelles. Dans le cas de la figure 2.15 le nombre d'enregistrements de la table de faits « Activité » peut théoriquement être égal au produit du nombre d'Etablissements par le nombre de Produits et par le nombre de Jours de l'historique mémoisé. C'est une borne maximum, car il n'y a pas nécessairement eu d'activité pour chaque combinaison possible. Même si le nombre d'activités réelles est une faible proportion de ce maximum, la table de faits a pratiquement toujours une taille supérieure d'un ordre de grandeur à la taille de la plus grande table dimensionnelle. Elle occupe en général 95 à 99 % du volume total de la base de données.

La génération de clés techniques est impérative. Pour être logiquement connectée, une table de faits doit posséder une clé pour chaque dimension. Dans chaque enregistrement d'une table de faits, les clés prennent une place importante. Si la table de faits possède des centaines de milliers, voire de millions, d'enregistrements, l'espace occupé par les clés dans la base de données est loin d'être négligeable. Il faut donc chercher à minimiser cet espace.

Il ne faut donc pas chercher à utiliser les clés signifiantes, qui ne sont pas faites pour économiser de la place, mais pour signifier quelque chose. Il faut utiliser les clés techniques numériques, générées éventuellement lors du chargement de la base de diffusion (ou dans l'entrepôt de données).

Le format des clés doit être homogène, le plus petit possible compte-tenu de la cardinalité de chaque table. Il faut penser également aux possibilités d'extension de la base en volume. Supposons dans la figure 2.15 que l'historique est de trois ans (1095 jours), le nombre de produits de 2500 et le nombre

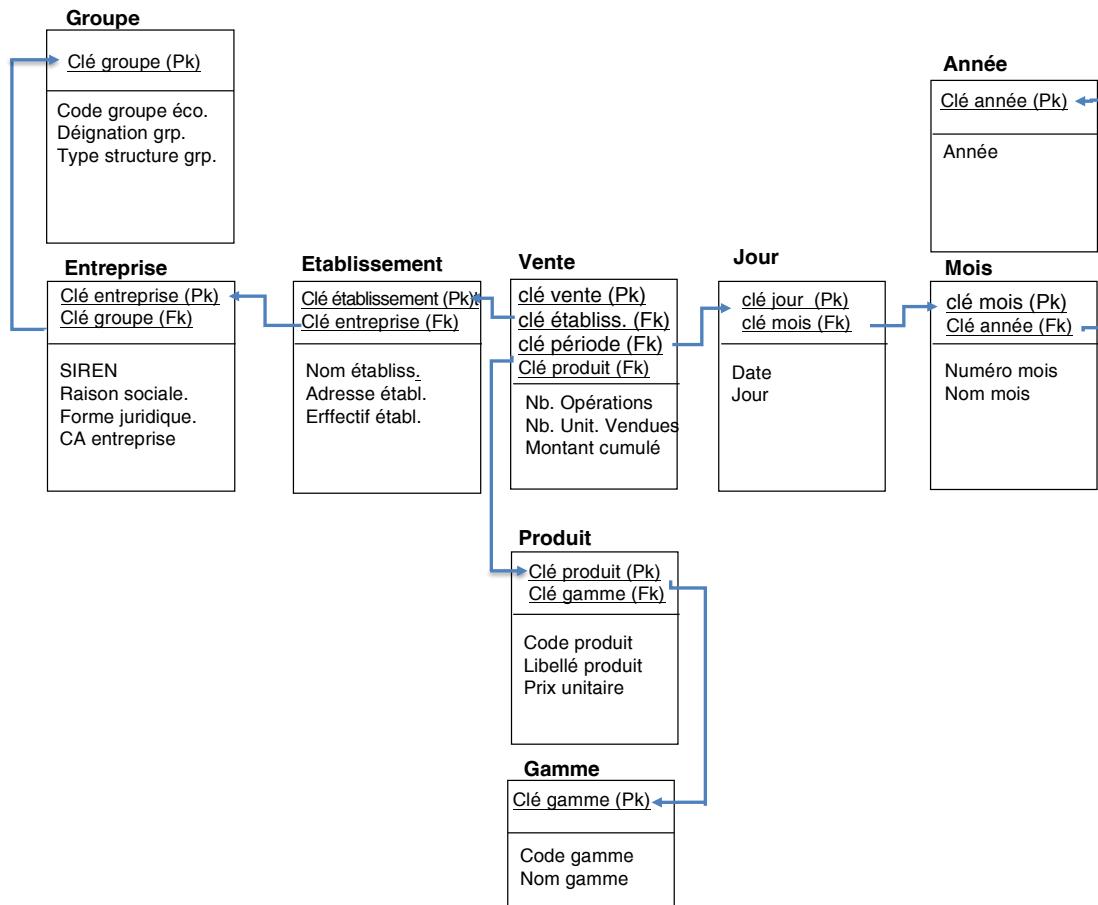


FIGURE 2.15 – Schéma en flocon

d'établissements de 84000. La clé jour peut être codée sur 2 octets, la clé produit aussi. Le nombre d'établissements nécessite au moins trois octets. Dans la pratique cette dernière clé sera codée sur 4 octets (entier long). Les clés historique et produits sans doute aussi de façon à prévoir des volumes plus importants.

Ce modèle en flocon présente l'avantage de conserver la forme normale du MCD, mais en revanche comporte des inconvénients, notamment en ce qui concerne, d'une part, la longueur des navigations générées pour répondre à des requêtes un peu complexes et, d'autre part, le nombre important de clés techniques à générer. Par exemple dans le cas de la figure 2.15, le traitement d'une requête par Année, par Groupe et par Gamme sera bien plus complexe que celui d'une requête par Mois, par Etablissement et par Produit. La complexité et le temps de traitement d'une requête augmentent proportionnellement au nombre de tables impliquées dans la jointure. Dans la pratique on préfère une forme dénormalisée du schéma en flocon : le *schéma en étoile*. La figure 2.16 représente le schéma en étoile dérivé du même modèle que le schéma en flocon de la figure 2.15.

Le schéma en étoile ne comporte, en plus de la table de faits qu'une table par dimension. Cette simplification est obtenue au prix d'une forte dénormalisation. Dans la dimension « Etablissement », par exemple, toutes les propriétés descriptives de l'Entreprise et du Groupe sont regroupées dans la même table que les propriétés de l'Etablissement. Dans le cas d'un groupe contrôlant 100 établissements, la description du Groupe sera répétée dans 100 enregistrements.

Ce modèle est donc générateur d'une forte redondance mais :

- la redondance des données ne compromet pas la cohérence d'une base ne subissant pas de mise à jour transactionnelle ;

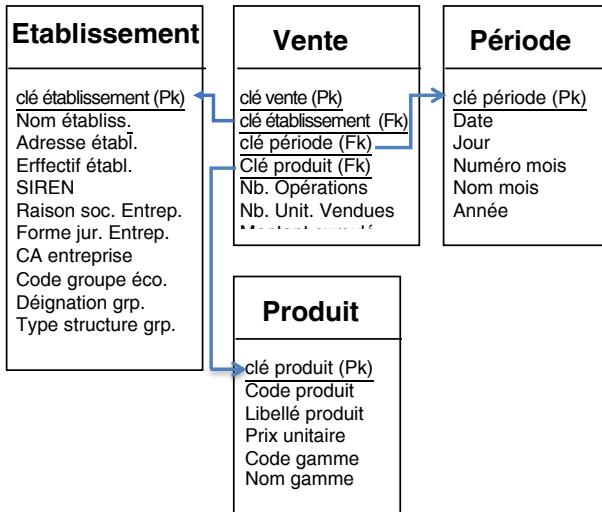


FIGURE 2.16 – Schéma en étoile

- l'espace occupé par les tables dimensionnelles est insignifiant par rapport au volume de la table de faits ;
- toutes les tables dimensionnelles ont une liaison directe avec la table de faits. Quelle que soit la complexité des dimensions, le nombre de tables pouvant être impliquées dans une requête, en plus de la table de faits, est inférieur ou égal au nombre de dimensions du contexte. Le temps d'exécution d'une requête est indépendant du niveau hiérarchique des propriétés conditionnelles invoquées.

Enfin, lorsque deux vues, c'est à dire deux tables de faits, partagent un certain nombre de tables de dimensions, on peut les rassembler en une seule structure qui prend alors la forme d'une *constellation* (figure 2.17).

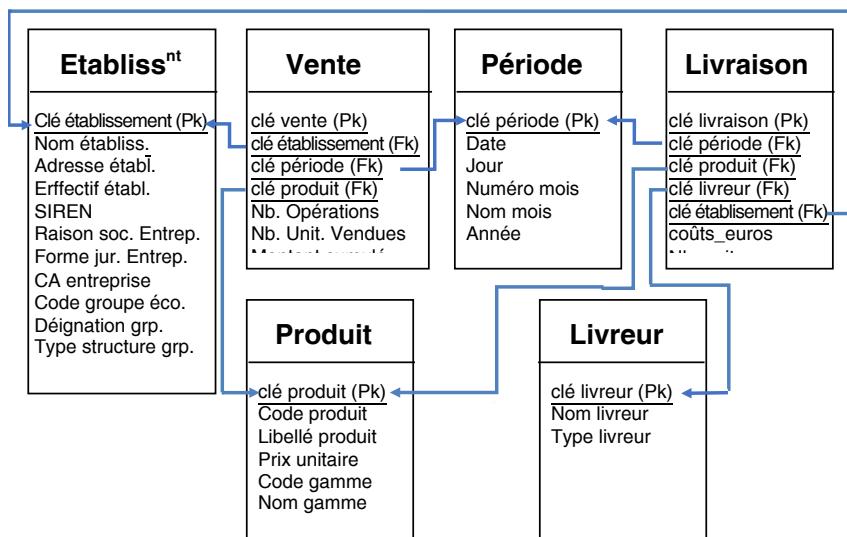


FIGURE 2.17 – Schéma en constellation

Chapitre 3

Formes Dimensionnelles Complexes

1. Introduction

Une des principales différences entre les systèmes opérationnels et les entrepôts de données est la capacité de ces derniers à capter les changements et les conserver. Les changements portent sur les dimensions qui sont amenées à varier *lentement* avec la vie des processus qui sont modélisés et représentés dans le data warehouse. La notion de variation lente fait référence à des phénomènes de modifications structurelles ou des contours des processus observés et non aux données instantanées qui sont mesurées et retransmises par les systèmes opérationnels.

Par exemple, supposons que l'on a définit les dimensions d'un data warehouse pour suivre les performances des vendeurs d'une entreprise qui vend des imprimantes et assure un soutien technique auprès de ses clients. Cette entreprise possède un système de gestion rigoureusement alimenté par les employés (techniciens, représentants, responsables de stocks, etc.). Ce système de gestion réalise les tâches courantes : consignation de l'information, récapitulatifs quotidiens, accès distant pour les nomades, facturation, etc. La production de rapports de ventes, d'analyses de ventes sur plusieurs années, du profit généré par territoire, par représentant, des activités des techniciens par territoire etc. semble relativement facile à concevoir jusqu'à ce que l'on rencontre un des événements suivants :

- le prix d'une imprimante change au milieu de l'année,
- un client déménage au milieu de l'année,
- un représentant change de territoire pour servir une autre catégorie de clients.

Les conséquences de tels changements vont être les suivantes :

- pour l'imprimante qui a changé de prix : comme les systèmes opérationnels ne prennent pas en charge la gestion de l'historique l'ancien prix sera perdu, donc les chiffres des ventes à la fin de l'année ne se baseront que sur le nouveau prix, ils seront donc faux.
- Pour le client qui déménage : la plupart des systèmes opérationnels de ventes utilisent la localisation géographique comme une partie de la clé primaire de la table « Client ». Si le client déménage (ou change de nom, ou fusionne avec une autre compagnie, etc.), on peut créer un nouveau client pour ne pas compliquer les choses. Mais, en créant un nouveau client, on « omet » toutes ses transactions précédentes. Donc si à la fin de l'année, on veut récompenser les meilleurs clients, et que le client en question déménage en juin, il n'aura que six mois de transactions à son actif. Pire, si l'on veut retirer le pourcentage de rabais aux mauvais clients et que durant les six derniers mois le client n'a pas payé à temps, ce dernier se verra perdre ses avantages, même s'il a été très rentable les six premiers mois.
- Pour le représentant qui change de territoire, on pourrait calculer la moyenne des ventes par vendeur, mais cela peut conduire à de mauvaises interprétations. Si le vendeur, qui a été muté, travaillait dans une région où le marché était très actif, se trouve maintenant dans une région où les ventes sont rares, sa performance apparaîtra bien plus élevée que celle des vendeurs de la même région, même s'ils sont aussi efficaces. On pourrait créer une deuxième instance du même vendeur, mais cela conduit à d'autres problèmes de suivi de son activité. Enfin, si on ne mesure que les ventes des territoires, un vendeur muté se verra attribuer toutes les ventes de ce territoire, même celles qui ont été faites avant son affectation.

1.1 Etats et flux

Dans un entrepôt de données, on manipule conjointement des indicateurs dont certains sont *dynamiques* et d'autres *statiques*.

Un fait dynamique représente un *flux* affectant le système observé, chaque flux élémentaire étant associé à un *événement*. Un fait statique est en revanche un élément descriptif de l'*état* du système à un instant donné. Le solde d'un compte courant ou la cotation d'un instrument financier sont par exemple des faits statiques alors que le montant d'un dépôt ou d'un retrait ainsi qu'une plus ou moins value sont des faits dynamiques.

Ces deux types de faits s'inscrivent de façon différente dans une chronologie périodique :

1. Un fait dynamique est un cumul de flux associés à une certaine catégorie d'événements survenus au cours de la période¹ de référence. Les événements survenus correspondant au flux peuvent se produire un certain nombre de fois (et éventuellement pas du tout) au cours de la période. Un fait dynamique périodique est donc par définition un *agrégat*, même si, dans la base de données du SID, il est vu comme une information élémentaire.
2. Un fait statique est un indicateur de situation ou de stock mesuré ponctuellement à un instant donné choisi une et une seule fois pour chaque période élémentaire. Quelle que soit la durée de la période de référence, un fait statique n'est déterminé que pour un point de la période.

En théorie, on peut toujours reconstituer l'histoire d'un fait statique à partir de celle d'un indicateur dynamique. A partir de la succession intégrale des opérations effectuées sur un compte courant, on peut, par exemple, retrouver le solde du compte à n'importe quel instant de son histoire. D'autre part, l'introduction d'un indicateur statique dans un contexte périodique suggère un certain degré d'arbitraire, puisqu'elle semble étendre à une période entière une valeur mesurée en un point de la période, cette valeur pouvant changer à tout instant. On pourrait donc considérer ces indicateurs comme à la fois redondants et artificiels.

En pratique, les faits statiques apportent dans certains contextes, une valeur ajoutée irremplaçable, pour plusieurs sortes de raisons :

1. la reconstitution certaine d'un état à partir des flux antérieurs n'est possible qu'à condition de disposer de l'histoire complète, précise et exacte de ces flux depuis l'origine, ce qui est une contrainte difficilement envisageable,
2. cette reconstitution, même si la mémoire antérieure des flux a été conservée, implique des coûts de recherche et de calculs exorbitants,
3. pour certains types d'analyse, les utilisateurs ne s'intéressent qu'à des échantillonnages périodiques et non aux flux correspondants, ces derniers n'étant pas forcément mémorisés,
4. la valeur périodique d'un indicateur statique n'est pas forcément si arbitraire qu'elle paraît. L'instant choisi dans la période pour prendre la mesure peut correspondre à une réalité significative. Cet instant peut être le début, la fin ou un autre point significatif unique pour la période.

Des faits statiques et dynamiques cohabitent donc souvent. Ceci contribue à l'hétérogénéité de comportement des faits dans les hiérarchies.

2. Les représentations du temps

On peut *a priori* exprimer le temps sous n'importe quel format et n'importe quel niveau de détail même si la plupart du temps la structure dimensionnelle qui semble s'imposer est de type *calendaire*, le temps étant découpé en journées éventuellement groupées par mois, semestres, années. C'est souvent le compromis le plus acceptable entre le grain voulu par les utilisateurs et les contraintes liées au volume et à la disponibilité des données primaires.

Des grains plus fins que la journée ne manqueront pas d'apparaître à mesure que l'approche dimensionnelle se généralisera à des domaines caractérisés par de fortes fluctuations horaires des phénomènes observables (marchés financiers, processus industriels, télécommunications, relevé de la consommation électrique instantanée, circulation routière, météorologie, ...).

1. Dans ce chapitre, la notion de période ne correspond pas à un phénomène qui se répète identique à lui même après un certain intervalle de temps, mais bien à un intervalle de temps pendant lequel on fait des observations ou des mesures.

2.1 Irrégularités périodiques

La période élémentaire d'un contexte est l'intervalle de temps à l'intérieur duquel il est convenu de ne pas tenir compte des variations du système observé. Une période possède au moins deux attributs distinctifs : une date de début et une date de fin. On ne doit en aucun cas considérer la date de fin comme une propriété redondante parce qu'elle est égale à la date de début de la période suivante car les périodes ne sont pas nécessairement successives et, sinon, elles peuvent être successives mais non conjointes.

Lorsque la période est de durée constante et correspond à un découpage calendaire ou horaire usuel, les bornes ne sont généralement pas explicites. Elle peut être désignée par un identifiant appelé « date ». Par exemple, « les opérations du 21 mars 2005 » désigne l'ensemble des opérations ayant eu lieu entre les heures d'ouverture et de fermeture de la journée conventionnelle du métier concerné. Que les limites soient explicites ou non, une période est par définition un *intervalle*. Cet intervalle ne coïncide pas nécessairement avec un jour, un mois ou toute autre période du calendrier. Le grain d'intervalle peut être très fin, dans le cas de phénomènes associés à des procédés industriels, ou au contraire s'étendre sur une durée de plusieurs mois, dans le cas d'observations liées à des cycles saisonniers, politiques ou économiques, par exemple.

Tout événement significatif pour l'utilisateur, se produisant à intervalles réguliers ou non, peut être considéré comme marquant la fin d'une période précédente et le début d'une période suivante : une période peut donc être de longueur variable.

L'irrégularité temporelle la plus gênante est celle qui provient non pas de la définition conceptuelle du SID, mais des contraintes de son alimentation. Les applications qui alimentent l'entrepôt de données peuvent avoir des cycles de mise à jour décalés les uns par rapport aux autres et qui ont de plus, parfois, des périodes opérationnelles de longueurs différentes. La nécessité de trouver des « points de réconciliation » pour ne pas charger des données incohérentes pèse donc lourdement sur la définition de la période de base d'un contexte.

Le grain invoqué par les utilisateurs dans leurs requêtes successives peut varier avec le temps et la position des périodes sur l'axe temporelle. Il est fréquent de ne s'intéresser aux périodes de base qu'à certaines époques de la chronologie (généralement la plus récente), et de cumuler sur des durées plus longues à d'autres moments. Ces irrégularités sont liées uniquement aux besoins des utilisateurs, qui peuvent varier avec le temps. Elles ne doivent donc pas avoir d'impact sur le MCD, qui ne doit prendre en considération que la période de base, et ne considérer les autres périodes que comme des niveaux de consolidation.

2.2 Périodes et événements

Le temps n'intervient pas obligatoirement sous forme *périodique*. Il peut aussi bien être perçu sous forme *événemmentielle*.

On définit un événement comme une transition affectant l'état du système opérant. Un SID peut donc mémoriser la chronologie aussi bien comme une succession d'événements que comme une succession de périodes.

2.2.1 Choix période/événement

Le choix entre période et événement peut se justifier de trois façons :

1. Un flux périodique est nécessairement un agrégat de flux événementiels. Sur le terrain les flux sont par nature événementiels. En les mémorisant sous forme périodique, on perd l'information sur le flux élémentaire et on ne garde trace que des montants cumulés et, éventuellement, du nombre d'événements survenus pendant la période. En revanche, certains contextes font appel à des données liées à des opérations élémentaires (par exemple, le ticket de caisse d'une grande surface qui correspond à l'événement achat par un certain client, d'un certain nombre de produits, à une certaine date, dans un certain magasin), qui eux sont par essence événementiels.
2. La distribution des événements dans le temps peut être très irrégulière par rapport au découpage périodique, quel qu'il soit. Elle peut être aléatoire ou liée au calendrier (ventes saisonnières). De ce fait, les contextes périodiques sont généralement très « creux » (tous les produits ne sont pas vendus tous les jours à tous les clients dans toutes les régions par tous les canaux). Dans une

période vide d'événement, les faits dynamiques sont nuls et les faits statiques ont généralement la même valeur que dans la période précédente. A l'inverse, un contexte à orientation événementielle est généralement très dense, puisque par définition, tout événement implique au moins un flux ou un changement d'état (en principe les deux), donc au moins une information significative ;

3. Selon le domaine d'application et le contexte d'analyse, l'utilisateur peut ne pas s'intéresser à des périodes calendaires prédéfinies, mais plutôt à la séquence des étapes d'un processus ou des mutations subies par le système.

Un contexte événementiel n'est pas limité à des faits dynamiques. On peut considérer un événement aussi bien comme porteur de changement d'état que porteur de flux. Par exemple le solde d'un compte bancaire après un dépôt (indicateur statique) est un fait tout aussi événementiel que le montant du dépôt.

2.2.2 Usage période/événement

Considérons le cas des corrélations d'achat dans une chaîne de grande distribution, c'est à dire l'étude des produits qui sont achetés ensemble (c'est à dire qui figurent sur le même ticket d'achat d'un client). Le grain d'un tel contexte ne peut pas être défini sur une base périodique telle que :

ventes / produit / site / jour

Les corrélations d'achat se mesurent sur les transactions élémentaires, puisque l'on veut savoir ce qui a été vendu en même temps au même client. Le grain utile est donc :

ventes / produit / transaction

Cette vente élémentaire est caractérisée par une date, l'idée d'une vue centrée sur les ventes par transactions par jour n'aurait donc pas de sens. Une vente n'a lieu qu'une fois et les flux associés à cette transaction sont nuls pour toutes les périodes autres que celle au cours de laquelle la transaction a eu lieu.

Le phénomène suggéré par cet exemple a une portée très générale. Une vue ne peut pas être conditionnée à la fois par *période* et par *événement*, par ce qu'un événement, s'il existe, n'est défini que pour une seule période. En d'autres termes, une dimension événementielle ne peut pas être combinée avec une dimension périodique dans une même requête. La période est fonctionnellement dépendante de l'événement. La première règle d'intégration en forme dimensionnelle normale interdit donc de les invoquer sur deux axes d'un même contexte.

Un contexte est donc soit *périodique* soit *événemementiel*, mais pas les deux à la fois (on ne peut pas croiser une période et un événement)².

On peut cependant consolider des événements sur une période. La présence d'entités périodiques est donc régulière dans un contexte événementiel, mais dans la même dimension que les événements. Une entité « Période » peut donc apparaître dans un contexte événementiel, mais uniquement en tant que niveau de consolidation hiérarchique, dans la même dimension que l'entité « Evénement ».

La manière de traiter le temps n'est pas la seule conséquence de l'alternative période-événement :

1. Les périodes ne sont pas les seules entités fonctionnellement dépendantes des événements. Une transaction élémentaire de vente, par exemple, implique également un lieu, un vendeur, un client, un moyen de paiement, une devise, La nature événementielle d'un contexte a donc pour effet d'intégrer sur un même axe des informations, qui auraient figuré sur des axes indépendants dans un contexte périodique.
2. Certains faits n'ont de sens que par période (exemple : un compteur d'événement tel que le nombre de ventes est pertinent pour un jour, mais pas pour une vente).

2. Nous verrons cependant qu'il est possible de faire coexister ces deux types de contexte au sein d'un même modèle dans son implémentation sous forme MLD

2.2.3 Exemple : Contexte périodique/Contexte événementiel

La figure 3.1 montre comment les informations liées à une activité commerciale peuvent être représentées sous forme d'un contexte périodique (Fig. 3.1 A) ou événementiel (Fig. 3.1 B).

Activité (1) : ventes, revenu, marge, montant moyen par vente

/ Clientèle : Client (Nom, Adresse)

/ Organisation : Magasin (Nom, Adresse)

/ Temps : Jour

/ Produit : Produit (Libellé, Prix unitaire)

Activité (2) : ventes, revenu, marge

/ Opérations : Vente (Date/Heure, Magasin, Commande)

- Client (Nom, Adresse)

- Magasin (Nom, Adresse)

- Jour

/ Produit : Produit (Libellé, Prix unitaire)

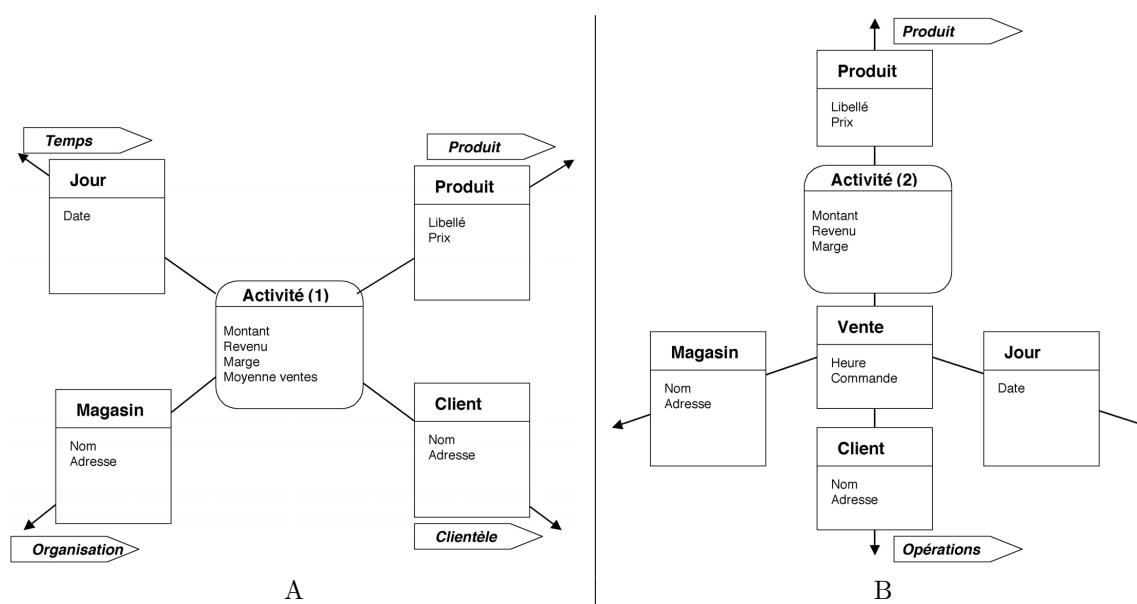


FIGURE 3.1 – Deux contextes (périodique et événementiel) de la même activité

Il existe entre les deux contextes une différence de grain puisque le premier ne prend en considération que les cumuls par Client / Magasin / Jour / Produit, alors que le second mémorise chaque vente élémentaire. Cependant, il ne s'agit pas que d'une différence de précision. Le premier contexte est périodique et le second est événementiel, ce qui implique une différence de structure. Chaque opération de vente est exécutée à une certaine date, dans un certain magasin, avec un certain client ; les entités Client, Magasin, Jour, sont donc en dépendance fonctionnelle (DF) de l'entité Vente dans la seconde structure. Comme une Vente peut impliquer un ou plusieurs produits, l'entité Produit échappe à cette DF et reste sur un axe indépendant.

Le nombre de dimensions d'un contexte événementiel est généralement réduit en forme dimensionnelle normale (FDN), sachant que l'entité événement est la source de nombreuses DF. En revanche une dimension événementielle est généralement beaucoup plus complexe qu'une dimension périodique, l'événement élémentaire comportant plusieurs propriétés pouvant orienter plusieurs chemins de consolidations. Elle est généralement porteuse de hiérarchies multiples.

3. Dérives dimensionnelles

La matière d'un domaine d'analyse est toujours constituée d'historiques, et fait référence à des données qui s'étendent sur des durées très longues par rapport aux cycles opérationnels de base. L'écoulement du temps n'a pas pour seul effet d'ajouter de nouveaux faits : à la longue, il modifie aussi les variables dimensionnelles.

Dans la plupart des situations réelles, les phénomènes de *dérive dimensionnelle* doivent être pris en compte correctement dans la structure du modèle, car autrement ils peuvent introduire des distorsions inacceptables dans les résultats. Par exemple il peut y avoir une modification des contours géographiques d'une région de consolidation, de la définition d'une gamme de produit, de la composition d'un groupe industriel par acquisition ou fusion, de la situation familiale d'un client suite à un mariage, une naissance, un décès, un déménagement, etc.

3.1 Dérives de contenu

On a vu qu'une propriété qui change, par exemple l'immatriculation d'un véhicule, ne peut être définie qu'au moyen d'une association entre deux entités dont l'une est la période. Dans notre définition des vues et des contextes, une telle propriété est un *fait*.

Il n'est pas possible d'éliminer le problème en remplaçant par des faits toutes les propriétés dimensionnelles susceptibles d'évoluer avec le temps. La classification des informations élémentaires en tant que *facts* et en tant que *conditions* présente donc un caractère de spécification fonctionnelle. Une propriété peut être changeante tout en restant un critère de sélection ou de regroupement dans une requête. Si certaines variables peuvent être à la fois des faits et des conditions, on perd toute la simplicité d'utilisation du modèle dimensionnel.

La seule méthode permettant de représenter correctement les valeurs successives des conditions sans compromettre la structure dimensionnelle du contexte est de distinguer les *propriétés permanentes* des *propriétés changeantes*.

3.1.1 Propriétés et associations permanentes/changeantes

Pour représenter les valeurs successives des *conditions* d'un *fait* qui peuvent changer avec le temps, on distingue :

- les propriétés *permanentes* ou *invariantes*, dont le contenu ne peut jamais changer (du moins dans le cadre de l'historique du contexte),
- les propriétés *mouvantes* ou *changeantes*, dont le contenu peut évoluer au cours du temps.

On peut considérer par exemple, dans le cas d'un véhicule, que la date de première mise en circulation, et la marque sont des propriétés permanentes, alors que la couleur et le numéro d'immatriculation sont des propriétés changeantes.

Il n'y a pas que les caractéristiques des objets qui peuvent changer, les associations entre entités peuvent aussi changer, par exemple le propriétaire du véhicule. Il y aura donc aussi des associations permanentes et des associations changeantes.

La représentation conceptuelle correcte de cette mobilité dimensionnelle implique donc deux entités, l'une avec les propriétés et associations permanentes, l'autre avec les propriétés et associations mouvantes, la première étant logiquement connectée à la seconde par une association « un à plusieurs » (1,n).

Dans l'exemple illustré par la figure 3.2, on répartira les propriétés permanentes (n° de série, marque, date de mise en circulation) et les propriétés mouvantes (immatriculation, couleur) respectivement dans deux entités nommées par convention « Véhicule(p) » et « Véhicule(m) ». Le « Véhicule(m) » sera associé au « Propriétaire » et le « Véhicule(p) » au « Constructeur ».

Le partitionnement d'une entité en partie fixe et partie variable permet, sans compliquer le modèle démesurément, de mémoriser un nombre quelconque d'états successifs d'un même objet. Dans l'exemple, il suffit de créer une nouvelle occurrence de « Véhicule(m) » et de la rattacher à un « Véhicule(p) » à chaque fois qu'un véhicule change de couleur ou de numéro. Ce procédé implique la présence explicite de dates dans les entités mouvantes, permettant de situer dans le temps chacun de leurs états successifs. Le couple « date début – date fin » délimite pour chaque état sa période de validité. C'est à cette condition

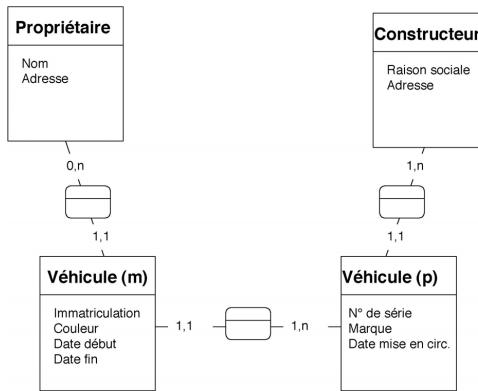


FIGURE 3.2 – Entité mouvante (m) et entité permanente (p)

qu'on pourra savoir, par exemple, que tel véhicule, à telle date, appartenait à tel propriétaire, était de telle couleur et avait tel numéro d'immatriculation.

3.1.2 Mise en forme dimensionnelle

Intégré dans un contexte dimensionnel normalisé, un tel couple d'entités correspond à un élément de structure hiérarchique. Chaque entité changeante étant rattachée à une entité permanente et une seule, la seconde peut être considérée comme un niveau de consolidation des indicateurs correspondant à la première. La mise en forme dimensionnelle va apporter quelques précisions dans l'organisation de ces entités et l'apparition des dates dans les conditions des propriétés.

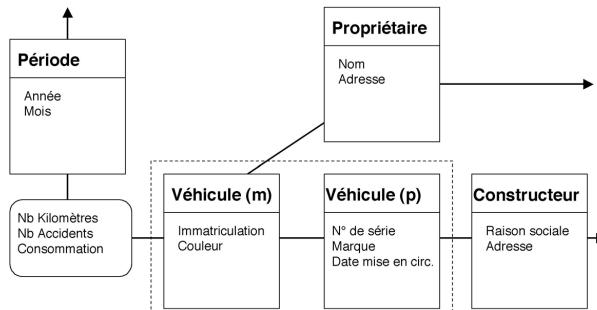


FIGURE 3.3 – Propriétés changeantes dans un contexte dimensionnel

Dans cette figure, le couple « Véhicule(m)–Véhicule(p) » est un sous-ensemble informationnel qui aurait formé une seule entité si on n'avait pas voulu mémoriser les changements d'état de chaque véhicule. Le croisement de la dimension « Véhicule » avec la dimension temporelle détermine ici des indicateurs tels que le kilométrage, le nombre d'accidents, etc. Selon le niveau hiérarchique auquel on se place, ce contexte permet, pour chaque période, de savoir quel est le nombre d'accidents pour les véhicules immatriculés dans une certaine région et d'autre part, le nombre d'accidents pour les véhicules d'une certaine marque. Le modèle permet d'enregistrer et de restituer des informations exactes et pertinentes pour les deux sortes de requêtes, même si des véhicules ont changé de département d'immatriculation au cours de la période d'analyse.

Les propriétés « Date début » et « Date fin », dans le modèle dimensionnel normalisé, disparaissent de l'entité « Véhicule(m) », la chronologie étant implicitement mémorisée par l'association avec la dimension périodique. L'occurrence de « Véhicule(m) » comportant une valeur « Couleur=rouge » n'aura d'intersection avec l'entité « Période » que pour les périodes comprises entre les deux dates de validité de cette valeur.

L'entité « Constructeur », est à un niveau hiérarchique au-dessus de « Véhicule(p) » (tout véhicule provient d'un fabriquant et tout fabricant produit plusieurs sortes de véhicules). En revanche, un véhicule pouvant avoir plusieurs propriétaires successifs, mais un seul à la fois, l'entité « Propriétaire » est un niveau de consolidation pour « Véhicule(m) » mais non pour « Véhicule(p) ». La dimension « Véhicule » comporte donc une hiérarchie double.

3.1.3 Mémorisation des états dimensionnels

Dans un modèle décisionnel, on crée une nouvelle occurrence de l'entité (m) d'une entité (p & m), à chaque modification de la propriété changeante (client : célibataire/marié; profession; adresse; ...) Chaque entité mémorisée est en réalité une *entité-état*. Il n'y a pas de problème d'incohérence dû à l'existence d'une même entité ayant plusieurs états dans la base. En effet, chaque occurrence n'est valide que pour certaines valeurs des autres dimensions, en particulier celles de la période.

De ce fait, il n'est pas nécessaire de faire figurer de date de début ni de date de fin dans les propriétés des *entités-états*, la période de validité de chacun des états successifs étant implicitement indiquée par le croisement avec la dimension temporelle du contexte.

La mémorisation des états dimensionnels n'est pas sans effet sur les *identifiants* et les *hiérarchies*. L'identifiant initial de l'entité « Client », par exemple, tel qu'il est utilisé dans les applications de produc-

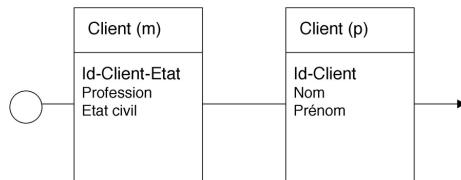


FIGURE 3.4 – Identifiant entité mouvante / identifiant entité permanente

tion ne suffit pas. Il faut créer un identifiant pour chaque état de chaque client. Il est souvent constitué de l'identifiant d'origine auquel on ajoute un code complémentaire, qui peut être un numéro chronologique d'état, par exemple. L'entité(p) peut être considérée comme un niveau hiérarchique dans lequel se consolident les informations liées aux entités-état correspondantes. Le fait de mémoriser les états successifs d'une entité n'interdit pas de suivre cette entité en tant qu'objet permanent à travers les âges. L'entité de consolidation représentant l'item permanent n'intègre que les propriétés stables.

3.2 Dérives de périmètre

Considérons l'exemple d'un groupe hôtelier pour lequel on s'intéresse à la fréquentation mesurée par le nombre nuitées par mois, au cours des deux dernières années, dans la région Ouest, pour les établissement de la marque « Doux logis » possédant 10 à 25 chambres. Pendant cette période, certains hôtels ont créé ou supprimé des chambres, voire changé de marque au sein du groupe. Il s'agit dans ce cas d'un simple changement de *contenu dimensionnel*. Mais, dans la même période, le groupe a pu modifier le périmètre de la région Ouest en lui ajoutant Loire-Atlantique. Les indicateurs de fréquentation n'ont donc pas le même sens avant et après cette modification.

Lorsque les dérives de périmètres sont assimilables à des dérives de contenu, on peut les traiter de la même façon que dans le paragraphe précédent. Si l'on veut faire une analyse à *périmètre constant* il est possible d'utiliser les entités état pour les reconstituer mais au prix de requêtes laborieuses et complexes. Lorsque la liste des périmètres constants envisagés est connue et d'une longueur raisonnable, elle peut être représentée plus efficacement par la méthode des *indicateurs qualifiés* (voir section 4.).

3.3 Dimensions changeantes et boucles hiérarchiques

La prise en charge des dimensions changeantes entraîne une relative complication des structures hiérarchiques. Certains chemins de consolidation sont valables pour les entités de type (p), d'autres seulement pour les entités de type (m).

Dans le cas des contextes illustrés par la figure 3.3, cela ne pose pas de problème, puisque les notions de « Propriétaire » et de « Fabriquant » sont nettement distinctes et non liées entre elles. En revanche, dans la pratique, il arrive souvent que l'entité permanente et l'entité mouvante soient rattachées à une même entité de consolidation, mais par des chemins différents. Par exemple, une personne peut avoir deux entités de commune ; une entité fixe, sa commune de naissance et une entité mouvante, sa commune de résidence. La commune de naissance est permanente alors que la commune de résidence peut évoluer avec le temps. Il y a donc deux associations possibles entre « Personne » et « Commune », qui peuvent être représentées selon la figure 3.5.

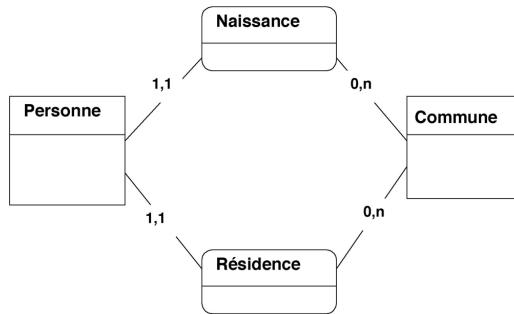


FIGURE 3.5 – Double liaison Personne - Commune

Ce schéma introduit une *ambiguïté de cheminement*. En suivant le principe énoncé précédemment l'entité « Personne » va disparaître au profit de deux entités « Personne(p) » et « Personne(m) ». Le lieu de naissance étant une propriété fixe, il est légitime de représenter la « Commune » comme niveau de consolidation de « Personne(p) ». Le lieu de résidence pouvant changer avec le temps, il est également légitime de représenter la « Commune » comme niveau de consolidation de « Personne(m) ». La même entité « Commune » apparaît alors, comme présentée sur la figure 3.6, à deux niveaux hiérarchiques à la fois, et introduit une boucle, c'est à dire de nouveau une *ambiguïté de cheminement*.

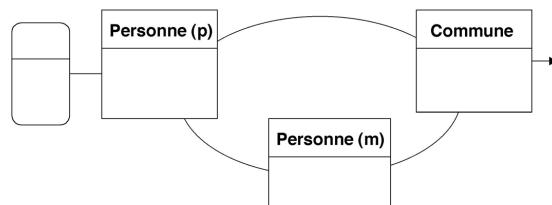


FIGURE 3.6 – Double liaison Personne - Commune

On introduit alors les questions suivantes :

1. Les vues impliquant des consolidations par commune de naissance et par commune de résidence appartiennent-elles vraiment au même contexte ?
2. La commune (de naissance et/ou de résidence) est-elle vraiment une entité à part entière, impliquant une structure distincte de "personne" ?
3. La commune de naissance et la commune de résidence sont-elles vraiment une seule et même entité (invoquera-t-on les mêmes attributs conditionnels dans l'une et dans l'autre) ?

Une réponse négative à la question (1) élimine la difficulté : il n'y a plus de hiérarchie cyclique puisque les deux rôles hiérarchiques de la commune appartiennent chacun à un contexte différent. Si la réponse à la question (2) est négative, la solution est également facile puisqu'en réalité, la commune de naissance et la commune de résidence disparaissent et sont remplacées par des propriétés intégrées respectivement dans « Personne(p) » et « Personne(m) ». Si la réponse à (1) et (2) est positive il faut bien représenter deux hiérarchies.

Il faut alors se demander si les requêtes impliquant la « Commune » en tant que commune de naissance ou en tant que commune de résidence sont susceptibles d'invoquer les mêmes attributs descriptifs de la commune. Si, selon les cas, on ne s'intéresse pas aux même propriétés on a deux structures de données, donc deux entités distinctes, situées sur deux voies hiérarchiques distinctes.

Enfin, si la réponse aux trois questions est positive, il faut recourir à un artifice de modélisation consistant à représenter deux entités distinctes, nommées différemment, même si on sait qu'elles contiennent les mêmes données (Fig. 3.7).

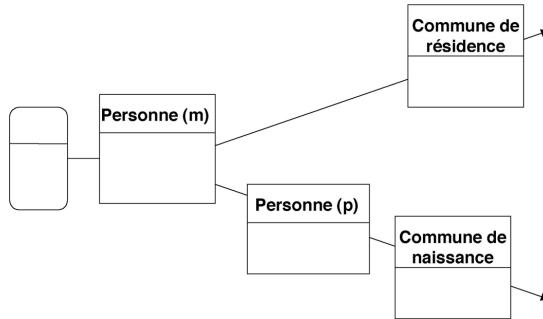


FIGURE 3.7 – Régularisation d'une hiérarchie cyclique

4. Indicateurs qualifiés

Un montant, une mesure peuvent être exprimés différemment :

- HT / TTC
- Euros / \$ / £
- valeur prévue / valeur réalisée
- différentes unités de mesure
- différents procédés de mesure
- valeur absolue ou relative (%,...)

Ce ne sont pas des faits différents, mais des *faits différemment qualifiés*. Ils sont la plupart du temps signalés par l'utilisation de qualificatifs ou d'expression (ex. « Revenu avant impôt » et « Revenu après impôt »).

Il est utile de distinguer l'indicateur fondamental de ses divers modes d'expression ou de représentation, et de ne spécifier comme faits, dans un contexte, que les indicateurs réellement distincts. En effet :

1. la détermination explicite des faits et qualifications est un excellent moyen d'affiner la définition des faits concernés et de prévenir les malentendus,
2. la représentation en tant que fait de chaque qualification de fait a pour conséquence pratique de multiplier exagérément le nombre de faits,
3. la liste des métriques évolue généralement plus vite que les indicateurs fondamentaux, d'où l'intérêt d'une description séparée des qualifications.

Pour dissocier les qualifications des faits, on ajoute des *dimensions qualificatives*. Le fait qualifié est noté une seule fois et toutes ses qualifications possibles sont définies dans ces dimensions supplémentaires. Une dimension qualificative agit alors comme un sélecteur dans les requêtes. Pour chaque requête, cette dimension qualificative permet à l'utilisateur de sélectionner un mode d'expression des résultats parmi une liste de modes d'expression.

La figure 3.8 est le graphe d'un contexte à quatre dimensions conditionnelles auxquelles s'ajoutent deux dimensions qualificatives. L'une définit la liste des devises dans lesquelles les faits peuvent être évalués, l'autre la définition budgétaire. Si dans le même contexte on avait énuméré en tant que faits toutes les combinaisons possibles, on aurait au total 32 faits ($4 \times 4 \times 2$) dans la structure.

Les qualifications ne sont cependant pas des dimensions au sens plein du terme :

1. Dans une même requête plusieurs occurrences d'une même qualification peuvent être spécifiées.
2. La notion de hiérarchie n'a pas de sens dans les dimensions qualificatives.

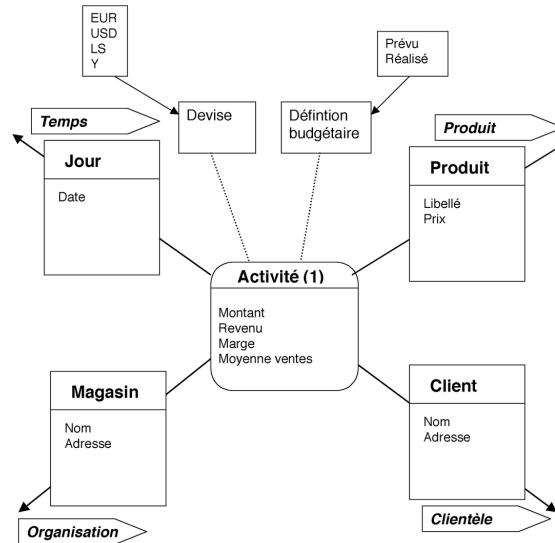


FIGURE 3.8 – Contexte qualifié

5. Les différents types de changements et leur gestion

On a vu en introduction qu'il y avait différentes façons de prendre en compte les changements dans les dimensions. Nous allons ici les répertorier et proposer pour chacun un mode de gestion approprié.

Type 0

Ce cas consiste à ne pas tenir compte d'un changement, c'est à dire que la valeur entrée au moment de la création d'une dimension ne sera jamais mise à jour, même si elle est modifiée. Cela correspond à des données dans des dimensions très spécifiques qui ne sont pas concernées par ce chapitre.

Type 1

C'est le cas le plus simple, on applique un changement de type 1 lorsqu'on ne veut pas conserver l'historique d'un champ. Pour les champs que l'entreprise ne juge pas pertinents de conserver, on se contente d'une simple mise à jour (UPDATE) dans la table de dimension, c'est à dire que l'on écrase l'ancienne valeur par la nouvelle. Par exemple, il est rare qu'une entreprise veuille conserver les changements du numéro de téléphone d'un client. C'est aussi la méthode la plus appropriée pour la correction de certaines erreurs de saisies (telles que, par exemple, un faute d'orthographe sur un nom).

Considérons la table 3.1 qui enregistre les informations d'un client dans une base de données. Dans cet exemple, **Code_Client** est la clé naturelle et **Clé_Client** est la clé primaire. Techniquement cette seconde clé n'est pas nécessaire, puisque le **Code_Client** définit de façon unique la ligne, mais les recherches et jointures seront plus efficaces avec une clé primaire qui est un entier, qu'avec une chaîne de caractères.

Clé client	Code client	Nom Client	Département Client
123	ABC	Electrotech	60

TABLE 3.1 – Table Clients

Supposons maintenant que le client déménage de l'Oise dans l'Ain, la mise à jour consiste simplement à remplacer l'ancienne valeur de département par la nouvelle (table 3.2) :

Clé client	Code client	Nom Client	Département Client
123	ABC	Electrotech	02

TABLE 3.2 – Table Clients mise à jour

Le désavantage de cette méthode de gestion est qu'il n'y a pas d'enregistrement d'historique conservé dans le data warehouse. On ne peut pas remarquer par exemple que les clients ont tendance à migrer d'un département dans l'autre. L'avantage est que c'est très facile à maintenir. Si on a calculé une table agrégat qui résume les faits par département, il faudra la recalculer quand le Département_Client aura changé.

Type 2

La méthode de type 2 repose sur le suivi des données historiques par la création, dans une table de dimension, de plusieurs enregistrements correspondant à une même clé naturelle donnée et différentes clés primaires ou différents numéros de version. Cette méthode de type 2 permet de préserver l'historique sans limite puisqu'un nouvel enregistrement est créé à chaque changement d'état.

Si l'on prend le même exemple que précédemment, la table de dimension prend la forme de la table 3.3, dans laquelle l'incrémentation des numéros de version donne la séquence de changements.

Clé client	Code client	Nom Client	Département Client	Version
123	ABC	Electrotech	60	0
124	ABC	Electrotech	02	1

TABLE 3.3 – Incrémentation des changements et modification des clés primaires

Une autre méthode utilisée couramment est d'utiliser les dates explicites de périodes de validité des valeurs.

Clé client	Code client	Nom Client	Départ. Client	Date début	Date fin
123	ABC	Electrotech	60	01-Jan-2000	21-Dec-2004
124	ABC	Electrotech	02	22-Dec-2004	null

TABLE 3.4 – Enregistrement des dates de validité

La valeur nulle de Date_fin dans une ligne indique la valeur courante du n-uplet. Dans certains cas l'utilisation d'une valeur standard (par exemple 31-Dec-9999), peut être utilisée comme date de fin, permettant à la fois d'indexer ce champ et d'éviter l'utilisation de valeurs de substitution pour les requêtes.

Les transactions qui font référence à une clé primaire Clé_Client sont donc toujours liées à des tranches de temps définies par cette ligne de la table de dimension. Une table agrégée qui résume les faits par département continue à représenter des états historiques, c'est à dire l'état dans lequel était le client au moment de la transaction, il n'y a pas nécessité de mise à jour.

S'il doit y avoir des changements rétrospectifs sur le contenu de cette dimension ou si de nouveaux attributs sont ajoutés à la table de dimension (par exemple une colonne Ventes/Représentant), qui ont des dates d'effectivité différentes de celles définies auparavant, alors il faut réaliser les mises à jour des transactions déjà enregistrées pour refléter cette nouvelle situation. Les opérations à réaliser dans la base de données risquent d'être très coûteuses, ce qui signifie que les méthodes de type 2 ne sont pas adaptées au changement du modèle des dimensions.

Type 3

La méthode de gestion des changements de type 3 est utilisée lorsqu'on veut garder un historique restreint (le dernier changement, ou les deux derniers changements). Elle consiste à ajouter autant de colonnes que de changements désirés avec les dates associées. La taille de l'historique enregistrable est limitée dans le cas du type 3, contrairement au type 2 qui est illimité, mais beaucoup plus facile à implémenter. Alors que la structure de la table originale a été peu modifiée pour le type 1 et le type 2, on ajoute des colonnes supplémentaires pour le type 3, tel qu'on le voit dans le cas de la table 3.5 :

Clé client	Code client	Nom Client	Départ. Origine	Date d'effet	Départ. courant
123	ABC	Electrotech	60	22-Dec-2004	02

TABLE 3.5 – Ajout de colonnes de changement d'état

On peut remarquer que cet enregistrement ne garde pas trace de tout l'historique des changements, tels que lorsqu'un client change deux fois de département. Une possibilité est de créer un champ **Département_Précedent** au lieu de **Département_Origine**, qui permettra de garder une trace du changement le plus récent.

Type 4

On fait souvent référence à la méthode de type 4, comme étant celle qui utilise des "tables d'historique". Une table conserve les données courantes et une table additionnelle conserve les enregistrements de certains ou de tous les changements. Dans le cas de l'exemple traité précédemment, la table originale pourrait s'appeler **Table_Client** et la table d'historique pourrait s'appeler **Historique_Client**.

Clé client	Code client	Nom Client	Département Client
123	ABC	Electrotech	02

TABLE 3.6 – Table Client

Clé client	Code client	Nom Client	Département Client	Date création
123	ABC	Electrotech	60	01-jan-2000

TABLE 3.7 – Table Historique Client

Type 6 : hybride

La méthode de type 6 combine les approches des types 1, 2 et 3 ($1 + 2 + 3 = 6$). Une explication possible de l'origine de cette appellation a été donnée par Ralph Kimball qui l'a appelée « Changements imprévisibles avec recouvrement d'une seule version » dans son livre *The Data Warehouse Toolkit*.

La table client commence par un enregistrement de notre exemple client :

Clé	Code	Nom Clt	Dpt. Cour.	Dpt. Histo.	Date déb	Date fin	Flag cour.
123	ABC	Electrotech	60	60	01-Jan-2000	31-Dec-9999	Y

TABLE 3.8 – Table Client initiale (1)

Le **Département_Courant** et le **Département_Historique** sont identiques. Le **Drapeau_Courant** indique qu'il s'agit de l'état courant ou de l'enregistrement le plus récent de ce client.

Quand Electrotech déménage de l'Oise à l'Ain, on ajoute un nouvel enregistrement, comme dans la méthode de type 2 :

Clé	Code	Nom Clt	Dpt. Cour.	Dpt. Histo.	Date déb	Date fin	Flag cour.
123	ABC	Electrotech	02	60	01-Jan-2000	21-Dec-2004	N
124	ABC	Electrotech	02	02	22-Dec-2004	31-Dec-9999	Y

TABLE 3.9 – Premier changement d'état (2)

On ré-écrit l'information Département_Courant dans le premier enregistrement (Clé_Client = 123) avec une nouvelle donnée, comme dans la méthode de Type 1, puis on enregistre l'historique dans une seconde colonne Département (Département_Historique), ce qui correspond à une méthode de Type 3.

Si dans notre exemple le client déménage à nouveau, on ajoute un nouvel enregistrement à la table de dimension client et on ré-écrit les contenus des colonnes Département_Courant et Date_Fin :

Clé	Code	Nom Clt	Dpt. Cour.	Dpt. Histo.	Date déb	Date fin	Flag cour.
123	ABC	Electrotech	80	60	01-Jan-2000	21-Dec-2004	N
124	ABC	Electrotech	80	02	22-Dec-2004	03-Feb-2008	N
125	ABC	Electrotech	80	80	04-Feb-2008	31-Dec-9999	Y

TABLE 3.10 – Deuxième changement d'état (3)

Notez que pour l'enregistrement courant (Drapeau_Courant = 'Y'), le Département_Courant et le Département_Historique sont toujours égaux.

6. Méthodes de consolidation

Dans la plupart des cas, lorsqu'il s'agit de calcul sur des collections de valeurs, la notion de *consolidation* est perçue comme synonyme de *somme*. Cependant si l'additivité est fréquente, ce n'est pas toujours le cas pour tous les indicateurs. Un indicateur peut être :

1. (universellement) additif (*les flux exprimés en montants absolus*), s'il peut être consolidé dans toutes les dimensions (ex. le « Montant des ventes » dans la figure 3.8) ;
2. non additif (*Les montants relatifs, les rapports, les variations*), s'il n'est additif dans aucune dimension (ex. « Moyenne des ventes » dans la figure 3.8) ;
3. semi-additif (*les stocks en montant absolu*), si il est cumulable dans certaines hiérarchies seulement.

Il doit toujours y avoir une méthode de consolidation et une seule pour chaque fait dans chaque hiérarchie. Cette méthode doit être spécifiée lors de l'élaboration du MCD. Tous les faits se consolident dans toutes les hiérarchies, mais chacun selon sa méthode, et, pour un fait, la méthode n'est pas forcément la même dans toutes les hiérarchies.

Un contexte n'est pas complètement défini tant que la méthode de consolidation de chaque fait dans chaque hiérarchie n'est pas spécifiée.

Chapitre 4

Traitement des données pour l'alimentation, la diffusion et l'OLAP

1. Introduction

Après avoir défini et étudié le Modèle Conceptuel de Données, notamment dans son aspect multidimensionnel, nous allons nous intéresser dans ce chapitre à sa mise en œuvre, c'est à dire son *architecture technique*. Entre, d'une part, l'environnement de requête et de présentation offrant à l'utilisateur une information conditionnée selon son propre point de vue et d'autre part, les sources de cette information (principalement les chaînes de production, éventuellement complétées par des apports externes) il existe une double distance :

- les données sources ne sont ni systématiquement cohérentes, ni synchrones, ni liées entre elles d'une manière adaptée à la perspective décisionnelle ;
- les environnements, généralement hétérogènes, d'où proviennent ces données, sont conçus et organisés autour de technologies, qui ne sont pas adaptées à l'implémentation directe d'applications décisionnelles avancées.

D'autre part, le SID doit adopter par rapport aux SIOs un profil bas de façon à ce que le déploiement d'un *data warehouse* ne perturbe pas leur fonctionnement quotidien. L'architecture du système doit donc assurer à la fois le conditionnement informationnel des données en provenance de la production et le cloisonnement entre l'environnement opérationnel et l'environnement décisionnel.

On peut définir bon nombre de composants et de modalités d'agencement de ceux-ci, pour satisfaire des performances qui peuvent varier selon la taille et le contenu des projets. Cependant, quels que soient les volumes traités, les performances requises et les périmètres concernés, la chaîne de mise à disposition des données implique, tel qu'on l'a vu dans le chapitre 1, quatre fonctions fondamentales :

- **collecte** : elle assure l'approvisionnement du SID en données primaires puisées dans le SIO et à l'extérieur ;
- **intégration** : elle assure la cohérence globale (à l'échelle d'un domaine) des données capturées, et leur mise à disposition en un point unique, conformément à des modèles unifiés et normalisés (*DWH* et *ODS*) ;
- **diffusion** : elle puise les données dans l'entrepôt central produit et maintenu par la fonction d'intégration, et les met à la disposition des applications, sous une forme *dimensionnelle, contexte* par *contexte* ;
- **présentation** : elle gère, au moyen de services logiciels, l'accès de l'utilisateur final aux données organisées par la fonction de diffusion.

L'énumération de ces fonctions ne correspond pas nécessairement à un découpage architectural ou à l'existence d'autant de dispositifs techniques distincts. En fait il n'y a jamais de coïncidence précise entre les organes physiques et les fonctions.

Avant d'étudier plus précisément les contenus et l'agencement de ces services, nous allons faire un rapide inventaire des architectures intermédiaires ou dégradées, qui sans être de véritables SID, sont souvent mises en œuvre pour produire des tableaux de bord et des présentations informationnelles des données.

2. Systèmes intermédiaires

Il y a toutes sortes d'architectures qui ont été définies et mises à disposition des utilisateurs avant l'émergence des data warehouse à proprement parler. On peut les citer rapidement sans chercher à les détailler ici.

- Les *tableaux de bord opérationnels* : les applications de production produisent le plus souvent des éditions systématiques. Ces états sont alors utilisés à des fins opérationnelles. Elles ne permettent évidemment pas de satisfaire toutes les contraintes et fonctions qui ont été définies pour un SID ;
- Les *interfaces de présentation et de dialogue* autorisent la formulation de requêtes interactives ou à exécution différée. Elles ne s'adressent cependant qu'à des données opérationnelles et les fonctions d'intégration et de diffusion ne sont pas assurées ;
- Les *systèmes de collecte et de présentation* ou *infocentres*, plus élaborés que les précédents, assurent, en plus de la fonction de présentation, quelques fonctions de collecte de base. Les requêtes portent sur des données copiées ou *répliquées* à partir des bases de production. Il y a bien une séparation physique entre les bases de production et les bases d'analyse, mais les données sont simplement juxtaposées. Il n'y a pas de modèle consolidé, les mécanismes d'alimentation ne sont pas coordonnés, les structures de données restent souvent hétérogènes, et les requêtes portent directement sur l'entrepôt central de données en l'absence d'un réel service de diffusion.
- Les *infocentres évolués* possèdent, en plus des fonctions de collecte et de présentation, une fonction d'intégration plus ou moins performante. Les utilisateurs disposent donc de données non seulement rassemblées, mais aussi unifiées et normalisées, c'est à dire organisées selon un modèle cohérent. Ces systèmes sont apparus avec le développement des SGBD relationnels et des environnements client-serveur.

Dans ces derniers systèmes, les données peuvent être organisées selon le périmètre et le vocabulaire du domaine concerné et non plus selon les points de vues disparates des applications produisant les données primaires. Cependant, même si c'est un gain notable par rapport aux systèmes précédents, les modèles de données restent du type opérationnel dans la mesure où il n'y a pas eu avant nécessité d'une démarche véritablement décisionnelle pour les construire. Il n'y a, en particulier, pas de distinction entre la fonction d'*intégration* et la fonction de *diffusion*. Enfin, comme dans les autres systèmes, les requêtes mettent directement à contribution la base de données intégrée.

3. Architecture de référence du SID

On va donc sortir du principe d'une architecture monolithique pour une architecture organisée en plusieurs « couches », celles-ci prenant en charge les quatre fonctions fondamentales qui ont été énoncées en introduction. On utilise souvent deux couches indépendantes qui assurent d'une part les fonctions de collecte et d'intégration et, d'autre part, celles de diffusion et présentation. Cette structure a l'avantage d'assurer un maximum d'isolation entre l'utilisateur et les sources de données.

Quel que soit le nombre de composants logiques et physiques effectivement mis en œuvre, on distingue deux dispositifs distincts :

- le *Système de Collecte et d'intégration* (SCI) ;
- le *Système de Diffusion et de Présentation* (SDP).

Ce découpage est lié à la cohabitation dans le système de modèles de données différents, de contraintes de fonctionnement différentes, et des liens d'interdépendance entre les fonctions.

4. Architecture et modèles de données

L'une des caractéristiques structurante d'un SID est la nécessité de gérer simultanément *trois modèles de données* :

- le *Modèle d'Intégration* (MI) ;
- le *Modèle de Diffusion* (MD) ;
- le *Modèle de Présentation* (MP).

Le Modèle Conceptuel de Données que l'on a étudié dans les chapitres précédents correspond au *modèle de diffusion*. C'est ce modèle qui représente la structure multidimensionnelle selon laquelle les

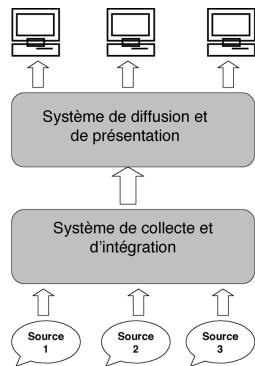


FIGURE 4.1 – Architecture de référence d'un SID (d'après Le Projet Décisionnel, Jean-Marie Gouarné, Eyrolles Edt. 1998, pp. 115)

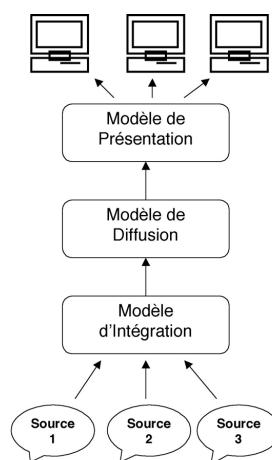


FIGURE 4.2 – Architecture et modèles de données (d'après Le Projet Décisionnel, Jean-Marie Gouarné, Eyrolles Edt. 1998, pp. 117)

données doivent être mises à disposition des applications décisionnelles.

Pendant ou après leur concentration physique par la fonction de *collecte*, les données sources sont filtrées, transformées et unifiées conformément à un modèle normalisé : le *Modèle d'Intégration*. Il effectue ses opérations sur les données brutes (homogénéisation, normalisation, synchronisation, agrégation, synthèse, ...). Sa fonction est d'unifier les données opérationnelles, et non de les structurer en contextes d'analyse décisionnelle. Son approche méthodologique de construction est fondée sur le paradigme entité-association et relève des principes de normalisation des modèles de données opérationnels.

Le MD n'est pas simplement un partitionnement technique du MI. Le découpage du MD en sous-ensembles logiques découle de sa construction à partir de vues dimensionnelles multiples. Nous verrons plus en détail les fonctions normalisatrices du MI dans la section consacrée à l'alimentation. On peut retenir pour l'instant que la distinction entre base d'intégration et base(s) de diffusion sont d'un autre ordre.

1. La distinction MI/MD permet de s'adapter à l'évolution des besoins des utilisateurs en désynchronisant les opérations collecte/intégration des opérations diffusion/présentation.
2. les deux systèmes peuvent fonctionner à des vitesses et des instants différents, limitant ainsi les problèmes d'accès aux SIOs et permettant de fournir aux utilisateurs en consultation un système toujours cohérent, puisqu'ils n'ont jamais accès aux données lors de leur mise à jour.
3. Les outils d'accès sont adaptés à chacun des deux systèmes.

Le *Modèle de Présentation* est simplement la mise à disposition d'un ensemble de moyens d'accès. Ce n'est qu'un masque plus ou moins transparent, qui recouvre le MD et en facilite l'accès au moyen de fonctions d'Interface Homme-Machine. On peut le modifier sans perturber le MD. Dans la mesure où c'est ce modèle que voit et utilise l'utilisateur, on a souvent tendance à le confondre avec les fonctions du MD. Pourtant son rôle est justement d'éviter à l'utilisateur de devoir interroger directement le MD avec son langage et son requêtage spécifiques. Il est évident qu'on ne peut pas plaquer directement un MP sur un MI, car on n'aurait pas les fonctions décisionnelles requises.

5. Alimentation

Le *Système de Collecte et d'Intégration* est la base de tout l'édifice. Il constitue, par des *traitements* sur les données du SIO, les *données* qui seront celles du *data warehouse*. Le SCI assure **deux fonctions** : la **capture sélective** (collecte) et la **mise en conformité** avec un modèle (intégration). Ces deux fonctions ne correspondent pas nécessairement à deux étapes de traitement ou des organes techniques distincts.

Les données que va collecter le SCI peuvent être bruitées, entachées d'erreur, manquantes, ou encore incohérentes, l'origine essentielle de ces problèmes étant la volumétrie importante sur laquelle on travaille. L'alimentation du SID ne se réduit donc pas à une activité de copie de données. Il y a nécessité d'un reconditionnement physique et de traitements appropriés. En réalité, les données sont plus *créées* que *copiées*. Il y a donc **différents pré-traitements et traitements** qui vont être appliqués aux données. Le nettoyage de données permet d'éliminer le bruit et les incohérences. On utilise l'intégration de données pour fusionner des données provenant de différentes sources dans un réceptacle cohérent unique. Différentes formes de transformations de données telles que la normalisation améliorent la précision et l'efficacité des algorithmes de fouille de données qui impliquent des mesures de distance. La réduction de données permet de réduire leur volume par agrégation, suppression des caractéristiques redondantes, ou classification, par exemple. Il faut ajouter à ces opérations sur les données toutes celles qui sont nécessaires à leur identification au moyen de la génération d'identifiants et de clés appropriés.

5.1 Pré-traitement des données

Dans le monde réel il peut arriver que les attributs de certains tuples n'aient pas de valeur enregistrée, ou que certaines valeurs soient erronées. Les raisons qui sont à l'origine de la "saleté" des données sont extrêmement diverses :

- Elles peuvent être incomplètes parce qu'elles n'étaient pas disponibles au moment de la saisie (données du client au moment de la transaction). Certaines n'étaient pas considérées importantes lors de la définition du datawarehouse. D'autres n'ont pas été enregistrées par erreur ou méconnaissance de la procédure, ou par défaillance de l'équipement, les données qui étaient incohérentes avec d'autres données déjà enregistrées ont été effacées, Il faut donc prévoir des procédures pour affecter des valeurs à ces attributs manquants.
- Les données bruitées correspondent à une mauvaise attribution de valeur à un attribut due à une erreur de fonctionnement des instruments de collecte, une erreur humaine ou informatique de saisie ou enfin, une erreur de transmission. Il peut y avoir aussi des origines techniques telles que la taille limitée d'un buffer qui ne permet pas une synchronisation correcte du transfert et de la consommation des données.
- Les incohérences des conventions de nommage ou d'encodage selon différentes sources de données peuvent générer des incohérences sur les données agrégées . La transgression d'une règle de dépendance fonctionnelle peut également générer des incohérences, par exemple, en modifiant une donnée liée. Enfin, les données dupliquées doivent être nettoyées.

Les données manquantes, incomplètes, bruitées ou incohérentes sont gênantes pour tous les algorithmes de fouille de données qui seront utilisés ensuite par le datawarehouse. Ces données "sales" sont génératrices de confusion dans les procédures de fouille ce qui rend les résultats non fiables. Bien que la plupart des algorithmes de fouille de données aient des procédures pour traiter les données bruitées ou manquantes, elles ne sont pas toujours robustes. Au contraire il peut arriver qu'elles cherchent surtout à les supprimer pour ne pas apporter de biais dans la fonction modélisée. La mauvaise qualité des données entraînera à coup sûr de mauvais résultats et la qualité des décisions repose sur la qualité des données. Par

exemple, des données dupliquées ou manquantes génèrent des statistiques incorrectes, voire trompeuses. En résumé, le **datawarehouse doit pouvoir faire des intégrations cohérentes de données de qualité**. L'extraction, le nettoyage et la transformation constituent la majorité du travail de construction d'un datawarehouse.

Les tâches essentielles du pré-traitement se résument de la façon suivante :

- Nettoyage des données :
 - ajouter les valeurs manquantes, lisser les données bruitées, identifier ou supprimer les outliers, résoudre les incohérences.
- Intégration des données :
 - intégration de plusieurs bases ou fichiers de données.
- Transformation de données :
 - normalisation et agrégation.
- Réduction de données :
 - génère une représentation des données réduite en volume, mais donne les mêmes résultats ou des résultats similaires.
- Discrétisation (découpage) des données :
 - C'est une forme de réduction, mais avec un impact différent, en particulier pour les données numériques (données qualitative).

5.2 Résumé des données descriptives

Une tâche de pré-traitement importante est de chercher à réduire le volume des données à traiter. En ce qui concerne les données descriptives (données numériques ou énumérables dotées d'une relation d'ordre) un résumé de celles-ci consiste à chercher des représentants calculés sur un sous-ensemble. L'objectif est d'avoir une meilleure compréhension à l'aide, par exemple, de tendance moyenne, de variation, d'étendue On utilise pour cela les outils statistiques de mesure des caractéristiques de dispersion (médiane, max, min, quantiles, outliers, variance, etc.).

5.2.1 Mesures de la tendance moyenne

- Moyenne (algébrique ou échantillon/population) : $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$, $\mu = \frac{\sum x}{N}$
 - calculée par le rapport *moyenne = somme de valeurs/nombre d'individus*
 - Moyenne arithmétique pondérée
$$\bar{x} = \frac{\sum_{i=1}^n w_i x_i}{\sum_{i=1}^n w_i}$$
- Médiane (mesure holistique)
 - Valeur médiane dans le cas d'un nombre impair de valeurs, ou moyenne des deux valeurs médianes dans l'autre cas
 - Estimée par interpolation (pour des données groupées)

$$\text{mediane} = L_1 + \left(\frac{n/2 - (\sum f)}{f_{\text{mediane}}} \right) \times c$$

où L_1 est la borne basse de la classe contenant la médiane, $(\sum f)$ la somme des fréquences des classes inférieure à la classe médiane, f_{mediane} la fréquence de la classe médiane, et c la taille de l'intervalle de la classe médiane.

- Mode
 - Valeur qui apparaît le plus souvent
 - Unimodal, bimodal, trimodal
 - Formule empirique : $\text{moyenne} - \text{mode} = 3 \times (\text{moyenne} - \text{mediane})$

La figure 4.3 est une illustration des différents cas que l'on peut rencontrer pour les paramètres moyenne, médiane, mode.

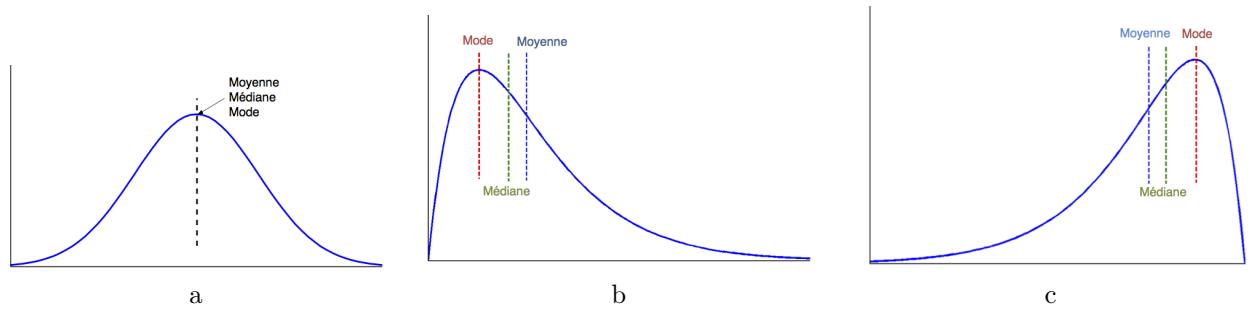


FIGURE 4.3 – Exemples de différentes déviations d'une population a) normale, b et c) décalée à gauche ou à droite

5.2.2 Mesure de la dispersion des données

- Quartiles, outliers et boîtes à moustaches
 - Quartiles : Q_1 (25 %), Q_3 (75 %)
 - Inter-quartiles : $IQR = Q_3 - Q_1$
 - Résumé : min, Q_1 , M, Q_3 , max
 - Boîte à moustaches : bornée par les quartiles, lignes médiane, moustaches et outliers
 - Outlier : en général une valeur supérieure ou inférieure à $1,5 \times IQR$
 - Variance et déviation standard
 - Variance :
- $$s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 = \frac{1}{n-1} [\sum_{i=1}^n x_i^2 - \frac{1}{n} (\sum_{i=1}^n x_i)^2]$$
- $$\sigma^2 = \frac{1}{N} \sum_{i=1}^n (x_i - \mu)^2 = \frac{1}{N} \sum_{i=1}^n x_i^2 - \mu^2$$
- déviation standard, s ou σ racines de s^2 ou σ^2
 - 68% de la population est comprise entre $\mu - \sigma$ et $\mu + \sigma$, 95% entre $\mu - 2\sigma$ et $\mu + 2\sigma$

La figure 4.4 est une illustration de boîtes à moustaches dans le cas de trois dimensions : revenu, coût, profit.

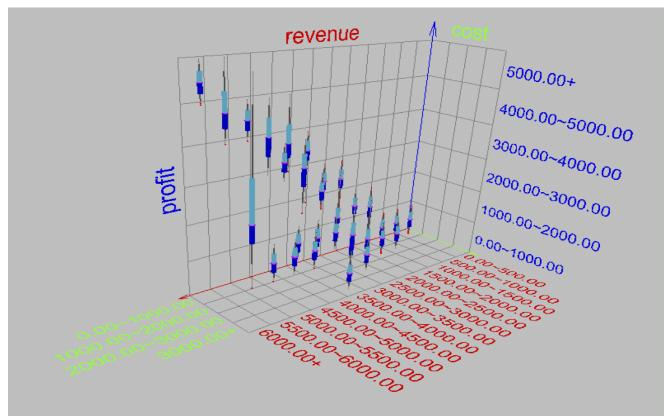


FIGURE 4.4 – Visualisation de boîtes à moustaches

Les données ayant été prétraitées par l'une des méthodes ci-dessus, on peut les analyser au moyen de différentes méthodes de visualisation ;

- Analyse d'histogrammes de fréquence de population

- Affichage de la courbe des quantiles pour des fractions croissantes de la population
- Affichage des quantiles respectifs de deux populations
- Affichage en deux dimensions pour faire apparaître des clusters
- Affichage des courbes de régression et de corrélation (voir chapitre ??).

5.3 Nettoyage des données

Le nettoyage des données est un des problèmes les plus importants à gérer dans un datawarehouse. Les procédures de nettoyage vont chercher à entrer les valeurs manquantes, lisser les données bruitées tout en identifiant les outliers, corriger les données incohérentes et résoudre les redondances générées par l'intégration de données.

5.3.1 Données manquantes

Des exemples de données manquantes ont été donnés en introduction de cette section (5.1), nous allons nous intéresser ici aux méthodes qui permettent d'y remédier. On peut :

1. ignorer tout l'enregistrement (le tuple) (possible à condition que le pourcentage du nombre de valeur manquantes par attribut ne varie pas trop),
2. entrer la valeur à la main : fastidieux, voire impossible,
3. réaliser une entrée automatique par :
 - (a) une constante globale : inconnue (nouvelle classe ?),
 - (b) la moyenne de l'attribut,
 - (c) la valeur moyenne de l'attribut des échantillons appartenant à la même classe,
 - (d) la valeur la plus probable (arbre de décision, règle d'inférence, formule de Bayes),

Les méthodes 3.b – 3.d introduisent un biais, la valeur entrée pouvant n'être pas correcte. Cependant la méthode 3.d est la plus populaire car, comparativement aux autres méthodes, elle utilise la plupart de l'information présente et passée pour prédire la valeur courante.

5.3.2 Données bruitées

Le *bruit* est une erreur aléatoire ou une variance aléatoire sur une variable mesurée. Des exemples de causes de données bruitées ont été données dans l'introduction de cette section (5.1). Le principe est de lisser les données par l'une des méthodes énumérées ci-dessous.

1. **Rangement** : On va lisser une valeur en observant ses voisines. Le principe est d'ordonner les données puis de les partitionner dans des boîtes d'égale fréquence. On va ensuite les lisser en remplaçant chaque valeur d'une boîte soit par la **moyenne de la boîte**, soit par la **médiane de la boîte**, soit par une des **bornes de la boîte**. Dans ce dernier cas, on affecte la valeur à la borne la plus proche. Plus la boîte est large, plus le lissage est important. On peut également chercher à définir des boîtes de largeur égale.

On a deux façons de construire la partition des données dans les boîtes :

- Partitionnement d'égale largeur (distance)
 - On divise l'échelle des valeurs en N intervalles de taille égale : grille uniforme
 - si A et B sont la plus grande et la plus petite valeur de l'attribut, la largeur de l'intervalle sera : $L = (B - A)/N$
 - C'est la méthode la plus simple, mais il faut faire attention aux outliers et elle ne gère pas correctement les distributions asymétriques.
- Partitionnement d'égale profondeur (fréquence)
 - On divise l'échelle de valeur en N intervalles, chacun contenant approximativement le même nombre d'échantillons.
 - Cette méthode est efficace pour l'échelonnement des données mais il peut être difficile de gérer les attributs de type catégories.

Exemple :

- ordonnancement des données de prix (en euros) : 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34
- partition d'égales fréquences :
 - boite 1 : 4, 8, 9, 15
 - boite 2 : 21, 21, 24, 25
 - boite 3 : 26, 28, 29, 34
- lissage par les moyennes de boites :
 - boite 1 : 9, 9, 9, 9
 - boite 2 : 23, 23, 23, 23
 - boite 3 : 29, 29, 29, 29
- lissage par les bornes des boites :
 - boite 1 : 4, 4, 4, 15
 - boite 2 : 21, 21, 25, 25
 - boite 3 : 26, 26, 26, 34

2. **Régression** : les données sont lissées en les faisant correspondre aux valeurs d'une fonction. Dans la régression linéaire on cherche à trouver la courbe la plus proche (la meilleure au sens d'une mesure de distance ou de proximité) de l'ensemble des points de telle sorte que l'on puisse prédire une valeur à l'aide d'une autre (figure 4.5). La régression multi-linéaire pour traiter les fonctions multi-variables et la régression logistique seront étudiées dans le chapitre ??.

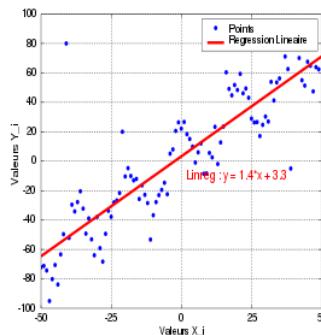


FIGURE 4.5 – Régression linéaire

3. **Classification** : les méthodes de classification permettent en particulier d'éliminer les valeurs aberrantes (*outliers*), en regroupant les valeurs similaires dans des *clusters*. Les valeurs en dehors des groupes sont des outliers.

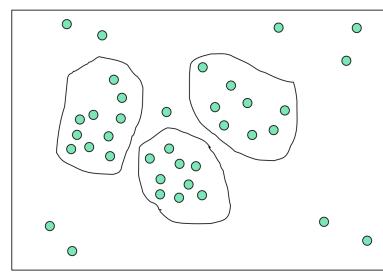


FIGURE 4.6 – Détection des *outliers* par méthode de classification.

4. **Combinaison de l'inspection humaine et par ordinateur** : une bonne façon d'identifier les outliers est de combiner l'observation aux résultats des calculs. On peut utiliser une valeur théorique pour identifier les outliers, puis les sortir dans une liste restreinte pour qu'un observateur humain décide de les affecter à une classe existante ou encore créer une classe supplémentaire.

5.3.3 Procédure de nettoyage des données

On peut organiser le nettoyage des données selon une procédure qui va faire appel à :

- la détection des écarts de valeurs :
 - utilisation des méta-données (ex., domaine, intervalle de variation, dépendance, distribution),
 - vérification des dépassements de valeurs des champs,
 - vérification des règles d'unicité, de conséquence, et de nullité,
 - utilisation d'outils commerciaux :
 - outils de nettoyage : exemple nettoyage des codes postaux,
 - outils d'analyse : découverte de règles et relations et détection des intrus (corrélations et classification pour découvrir les outliers).
- la migration et intégration des données :
 - outils de migration : permet de spécifier les transformations (exemple H/F \Rightarrow Homme/femme),
 - outils ETL : interfaces graphiques pour spécifier les transformations.
- l'intégration des deux procédures :
 - itérative et interactive.

5.4 Intégration et transformation des données

La construction d'un datawarehouse et les traitements de fouille de données associés, nécessitent la plupart du temps d'intégrer des données en provenance de plusieurs sources et de les transformer de façon à ce qu'elles soient dans un format approprié pour leur exploitation.

5.4.1 Intégration de données

L'*intégration de données* est nécessaire pour regrouper celles qui proviennent de différentes sources. Les sources étant hétérogènes, le SCI va se charger d'unifier et de rassembler les données d'origines diverses et hétérogènes et qui représentent les mêmes entités. Cette opération est presque toujours brouillée par l'existence de *synonymie* et de *polysémie*.

- des noms qui diffèrent d'une source à l'autre peuvent désigner des objets identiques ou semblables. Par exemple, un « Client » dans une application, peut être nommé un « Compte » dans une autre et un « Dossier » dans une troisième ou encore repéré par deux labels distincts (client-id \equiv client-numéro) ;
- une même entité peut être identifiée différemment selon les sources : Bill Clinton = William Clinton ;
- à l'inverse, l'appellation « Chiffre d'Affaire » peut correspondre à deux réalités différentes selon les applications ;
- l'attribut d'une même entité peut prendre des valeurs différentes selon la source dans laquelle il est évalué (problème de représentation, d'échelle, d'unité, de référence, etc.).

Les *dictionnaires de données*, qui seraient nécessaires pour traiter ces problèmes et réduire les risques d'erreurs, sont en fait bien souvent absents des applications, et il faut donc mener des enquêtes systématiques pour définir la liste des données décisionnelles recherchées et établir la liste des données opérationnelles nécessaires.

Une forme plus complexe de redondance vient de ce que certains attributs peuvent être déduits à partir d'autres (par exemple le revenu annuel). Un nombre important de redondances de ce type peut ralentir considérablement ou rendre confuses les procédures de recherche de liens de cause à effet dans un SID. L'opération de nettoyage et d'intégration doit donc aussi prévoir des étapes pour éviter ces redondances.

On peut détecter les redondances par l'analyse de **corrélations**. Par exemple, étant donnés deux attributs, une telle analyse permet d'évaluer dans quelle mesure une valeur d'un des attributs implique celle de l'autre. La corrélation entre A et B est donnée par :

$$r_{A,B} = \frac{\sum(A - \bar{A})(B - \bar{B})}{(n - 1)\sigma_A\sigma_B}$$

où n est le nombre de tuples, \bar{A} et \bar{B} sont les moyennes respectives de A et B et σ_A et σ_B , les écarts types de A et B . Selon la valeur de $r_{A,B}$ on pourra décider si A et B sont corrélés ou indépendants et s'ils sont corrélés alors ont pourra supprimer l'un des deux.

5.4.2 Transformation

Les données doivent être transformées ou consolidées pour se trouver dans un format approprié pour les opérations de fouilles de données. On va trouver comme opérations de transformation :

- **Lissage** : dont l'objectif est la suppression du bruit à l'aide de méthodes vues ci-dessus (partitionnement, classification, régression).
- **L'agrégation** : on réalise des opérations pour résumer ou agréger les données, et typiquement la construction de cubes à granularité multiple.
- **La généralisation** : dans laquelle on construit une hiérarchie de concepts qui permet de remplacer les données brutes par des catégories plus élaborées qui les généralisent (une valeur *age*, devient une catégorie, jeune, adulte, âgé).
- **La normalisation** : on est souvent amené à utiliser des algorithmes de fouille de données qui utilisent une distance (réseaux de neurones, k plus proches voisins, classification, par exemple). Les résultats obtenus par ces méthodes sont bien meilleurs si les données ont été normalisées, c'est à dire ramenées à des intervalles tels que [0.0, 1.0]. Par exemple, les données client contiennent l'attribut *âge* et *salaire annuel*. Le *salaire annuel* a un intervalle de variation bien plus important que *âge*. En conséquence, si les valeurs ne sont pas normalisées, les distances mesurées sur *salaire annuel* vont généralement écraser celles mesurées sur *âge*. Les méthodes de normalisation sont par exemple :
 - min-max,
 - réduction-centrage,
 - échelle décimale ou logarithmique.

- **La génération de nouveaux attributs** : appelée également *enrichissement*, elle est, comme l'agrégation primaire, une opération de calcul produisant des données de synthèse à partir des données opérationnelles. Cependant cette opération est plus élaborée et les objectifs ne sont pas les mêmes.

L'agrégation condense une collection homogène de données tirées d'une même structure. L'enrichissement produit des données de synthèse à partir de combinaisons de données puisées dans des structures différentes selon des formules plus ou moins complexes. Il peut avoir pour effet de réduire les volumes, mais parfois aussi de produire l'effet opposé : les données enrichies peuvent parfaitement représenter un volume supérieur à celui des données brutes.

L'enrichissement consiste à créer une donnée inconnue dans la source, pouvant être déduite de l'association de deux ou plusieurs données sources. On pouvait, par exemple, déduire du numéro d'immatriculation d'un véhicule, la région où est domicilié son propriétaire, de même on peut déduire l'âge d'une personne à une date donnée à l'aide de son numéro de sécurité sociale.

On peut enrichir des données originaires de sources internes (la production) avec des données en provenance de sources complémentaires externes (données publiques, données achetées à des spécialistes, ...). Se posent alors souvent des problèmes de cohérence, de codification et de classification.

Par exemple, la notion de client, qui est centrale dans beaucoup de projets décisionnels, est absente de beaucoup de SIO qui manipulent une information primaire liée aux *produits* et aux *activités*. Selon qu'il passe des commandes, reçoit des marchandises, ou paie des factures, le client peut être vu de différentes manières. La question n'est pas de savoir si le « vrai » client est celui de l'administration des ventes ou celui de la facturation, elle est de définir une nouvelle entité « Client » comportant une description utile pour le SID. Cette nouvelle entité possédera éventuellement deux propriétés « Adresse », une pour la livraison, l'autre pour la facturation.

Cette structure, construite à partir de données hétérogènes, peut poser des problèmes épineux lorsque le lien entre facture et client n'est plus qu'indirect. Considérons, par exemple, le cas d'une entreprise de transport rapide de colis. Un colis peut être acheminé en port payé ou en port dû. Dans le premier cas la facturation est reliée à l'expéditeur et dans le second au destinataire. Les données relatives à un expéditeur ne peuvent en fait refléter qu'une partie du volume d'affaire réalisé avec lui, le reste ayant été facturé aux destinataires en port dû. Pour reconstituer dans l'entrepôt de données le chiffre d'affaire réalisé avec cet expéditeur, indépendamment du mode de paiement, il faut, après avoir intégré les montants qui lui ont été facturés,

1. retrancher les montants associés à des colis qu'il a lui-même reçus en port dû ;

2. ajouter les données de facturation rattachées aux destinataires auxquels il a expédié des colis en port dû.

La seconde opération demande de relier destinataires et expéditeurs, ce qui peut impliquer des cheminements logiques compliqués à travers les données de routage puisées à différentes sources.

5.5 Réduction des données

Un data warehouse peut contenir des tera, voire des penta, octets de données, la réduction des données est donc impérative pour fiabiliser et optimiser leur enregistrement et leur gestion. De plus l'analyse de données complexes peut être coûteuse en temps sur des volumes complets de données.

Les techniques de réduction de données peuvent être appliquées pour obtenir une représentation réduite de l'ensemble des données qui est plus petite en volume, mais qui produit quasiment les mêmes résultats analytiques. Les méthodes les plus fréquemment rencontrées sont :

- Agrégation dans des cubes :
- définition d'un sous ensemble cohérent, représentation et langage de requêtes appropriés.
- Réduction des dimensions :
 - Sélection des attributs significatifs (arbre de décision, apprentissage, ...),
 - ACP, AFC, transformations (filtrage, ondelettes,).
- Compression des données :
 - compression des chaînes de caractères, audio, vidéo, séquences temporelle
- Réduction du nombre (modélisation)
 - représentation alternative, méthodes paramétriques ou non paramétriques, régression linéaire, histogrammes, classification, échantillonnage,
 - Discrétisation et génération de hiérarchies de concepts.

5.5.1 Agrégation

Attention, il ne faut pas confondre les valeurs agrégées du SDP, qui sont des valeurs pré-calculées (des cumuls, le plus souvent) et les agrégats constitués à partir des données des SIO et stockées dans le SCI. Ces derniers sont des données synthétiques destinées à remplacer les données opérationnelles des SIO. Les valeurs élémentaires entrant dans le calcul ne sont pas enregistrées dans l'entrepôt, soit parce qu'elles ne sont pas utiles pour les utilisateurs du SID, soit parce que leur mémorisation intégrale représenterait un coût exagéré de stockage. Les agrégats du SCI ne sont plus décomposables, alors que dans le cas du SDP on a conservé les valeurs primaires constitutives des agrégats.

Le choix des agrégats primaires du SCI est dicté par le grain des indicateurs définis dans les différents contextes dimensionnels du SDP. Il relève des spécifications fonctionnelles du projet. Il faut se méfier de la tendance naturelle à vouloir tout garder « au cas où », qui entraînerait très rapidement des volumes insupportables.

L'agrégation primaire doit être effectuée le plus en amont possible dans le SCI. Les données agrégées représentant une réduction du volume des données, il est logique de chercher à exécuter en premier lieu les traitements qui réduisent le plus le volume. L'idéal est de traiter l'agrégation au moment de la capture, sur la plate-forme technique de la source.

En résumé, l'agrégation c'est l'élimination des données opérationnelles détaillées par une opération de condensation. Les problèmes algorithmiques liés au traitement des agrégats et au requêtage dans les cubes seront traités dans les sections 5.6.2 et 5.6.4.

5.5.2 Discrétisation et hiérarchisation

Il y a trois grands types d'attributs : les attributs nominaux, qui sont des valeurs descriptives non ordonnées, les attributs ordinaux, qui sont des valeurs énumérées discrètes et ordonnées, les attributs continus qui sont des nombres entiers ou réels.

On utilise les méthodes de *discrétisation* pour réduire le nombre de valeurs d'une variable continue, en divisant son intervalle de variation en une série d'intervalles. Les labels des intervalles sont alors utilisés pour remplacer les valeurs de la variable. On peut alors utiliser des algorithmes de classification ou de regroupements hiérarchiques qui fonctionnent sur des attributs catégoriels.

La *hiérarchisation de concepts* permet de réaliser une réduction récursive des données en rassemblant et remplaçant les concepts de bas niveau (valeur de l'âge) par des concepts de plus haut niveau (jeune, adulte, senior).

5.5.3 Formatage et standardisation

Ce sont des opérations nécessaires dues à l'origine diverse des données sources et des programmes qui les ont élaborées (définition des longueurs des chaînes de caractères, des entiers, des réels, ...). Les opérations de formatage les plus simples sont la *concaténation* et la *séparation*. La première consiste à mettre bout à bout deux ou plusieurs champs de données pour en faire un seul. La seconde réalise l'opération inverse. Exemples :

- concaténation/séparation des différents formats de stockage des adresses pour n'en gérer qu'un seul,
- concaténation de champs situés dans des enregistrements successifs, voire de fichiers différents,
- troncature (extraction de la région du Code Postal, arrondi, suppression du jour, du mois, des centimes, ...)
- normalisation des noms (SOGELEC, Groupe SOGELEC, SOGELEC S.A, Société Générale d'Électricité).

5.6 Génération des identifiants et des clés

Dans un modèle de données en forme régulière, chaque entité possède toujours un *identifiant*, c'est à dire une propriété ou un ensemble de propriétés permettant de caractériser chaque occurrence. L'intégration de données au moins partiellement synthétiques nécessite la création de propriétés identifiantes qui n'existent pas dans les sources. On verra en particulier :

- la génération d'identifiants pour les objets synthétiques ;
- la génération d'identifiants pour les états successifs de certaines propriétés (id. de base + n° version) ;
- le choix d'un identifiant fédérateur pour des objets présents dans plusieurs sources ;
- la généralisation d'identifiants locaux, qui n'existaient pas dans certaines sources (par exemple, certaines entreprises n'attribuent pas à leur client un identifiant général, mais seulement un identifiant relatif à leur agence, de sorte que deux clients peuvent avoir le même numéro, s'ils sont gérés par deux agences. L'identification absolue implique alors le numéro de client et le numéro d'agence. Cet identifiant composite n'est pas un problème en soit, si ce n'est qu'il introduit un supplément de lourdeur et de complexité) ;

Le problème se pose en des termes identiques pour les clés. Rappelons qu'un identifiant est une propriété parmi d'autres, qui peut être utilisée comme condition, c'est à dire comme critère de sélection, alors qu'une clé est un identifiant dont la valeur n'a aucune signification et qui, en outre, n'est pas visible pour l'utilisateur. C'est un identifiant purement technique, interne à l'entrepôt de données, dont la valeur est attribuée par un compteur au moment de la création de chaque enregistrement. La fonction habituelle d'une clé est de matérialiser une liaison entre deux tables.

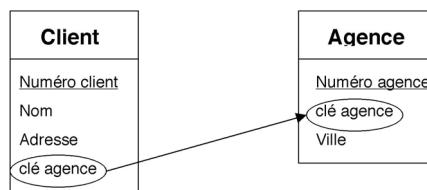


FIGURE 4.7 – Identifiants et clés

Sur la figure 4.7, Client et Agence ont chacun un identifiant et une clé. Dans la table Agence, l'attribut « clé agence » est une clé *primaire*, tandis que dans la table Client, c'est une *clé étrangère* ou *clé externe*. Il ne s'agit plus ici d'objets « Client » et « Agence » de niveau conceptuel, mais de tables ou de fichiers

physiques. Dans cet exemple d'école, on aurait pu s'abstenir de la clé technique en intégrant la propriété « Numéro Agence » dans la table Client, mais en pratique il est préférable de marquer les jointures par des clés numériques plus compactes, permettant une meilleure performance.

6. Le système de Diffusion et de Présentation

Le rôle fondamental du SDP est la *mise à disposition*, sous forme informationnelle appropriée, des données acquises par le SCI.

Le SDP s'appuie, pour son alimentation, sur une source unique et normalisée, en l'occurrence l'entrepôt central de données, créé et maintenu par le SCI. Il n'a donc pas de liaison directe avec les sources de données, de même que les utilisateurs n'ont pas de liaison directe avec le SCI.

Le SDP peut contenir une ou plusieurs base de données, chacune correspondant à des MCD différents. Ces bases supportent chacune une version du Modèle de Diffusion.

6.1 Modalités d'accès à l'information

Le SDP doit répondre aux spécifications élaborées à partir des requêtes. Pour jouer ce rôle efficacement, la structure des requêtes connues et prévisibles doit trouver un écho dans la structure des *contextes* du Modèle de Diffusion. La connaissance des vues toutefois ne suffit pas. Les conditions d'exploitation des vues, l'ergonomie de leur présentation, le degré de responsabilité laissé à l'utilisateur, le mode de restitution des résultats sont autant d'éléments de spécification qui complètent le Modèle Conceptuel de Données. Ces modalités pratiques d'utilisation relèvent de plusieurs profils types. L'architecture du SDP doit répondre de façon aussi réaliste que possible à ces profils d'exploitation dont on propose ci-dessous une classification (on a défini huit modalités d'accès) :

6.1.1 Etats prédéfinis

Même si ce n'est pas toujours justifié, le SDP se doit de reprendre les fonctions classiques d'édition systématique d'états imprimés, initialement assurées soit par les applications de production, soit par des outils de type infocentres. La reprise de ces fonctions a au moins pour intérêt de ne pas bousculer les habitudes de travail et de favoriser l'appropriation du SID.

6.1.2 Requêtes paramétrables

Les restitutions interactives qui ont connu la plus grande expansion sont les requêtes de type *semi-dirigé*. Ces sont des requêtes au format général prédéfini, mais dont certains paramètres sont choisis par l'utilisateur et dont l'exécution est déclenchée à la demande.

Ces restitutions nécessitent une interface de dialogue en présentant un schéma apparent orienté métier, qui masque le schéma de la base de diffusion et est conforme au *Modèle de Présentation*.

L'interface de dialogue offre à l'utilisateur la possibilité de choisir un certain nombre d'options de restitution et de paramètres de sélection, et d'exprimer ses requêtes dans un langage non technique. Ces requêtes sont traduites dans un langage approprié de type *SQL*, puis les données obtenues en retour sont formatées selon le cadre de présentation qui a été défini dans le Modèle.

Même si on peut envisager de laisser un certain degré de liberté à l'utilisateur, il n'a pas accès directement à la structure réelle des données, ni à la base de diffusion. Il est en interaction avec un *moteur de présentation*, qui lui propose des vues pré-organisées. Il n'est pas question d'improviser de nouveaux cadres de présentation, ni de requêtage inattendu.

6.1.3 Manipulation dimensionnelle libre

C'est le service le plus achevé d'un SID, qui est en général réservé à des utilisateurs formés. La création de requêtes non programmées par l'utilisateur implique que ce dernier dispose d'une interface montrant la structure dimensionnelle du contexte auquel il s'intéresse et lui permettant de composer ses propres vues. C'est le cas par exemple lorsqu'à partir de l'observation d'une vue dans une session d'analyse, l'utilisateur, ayant cru remarquer un phénomène intéressant, souhaite approfondir son analyse

par l'examen d'autres vues, non programmées. Cela peut se matérialiser par la descente d'un niveau hiérarchique dans chaque dimension, ou bien l'ajout d'une dimension supplémentaire, ou encore par une remontée dans une hiérarchie dans une dimension et une modification d'un paramètre dans une autre pour faire des comparaisons sur des résultats cumulés. Les manœuvres dimensionnelles de base sont :

- la navigation verticale dans les données (*drill down*, *drill up*), c'est à dire le passage d'un certain niveau de présentation d'un ensemble de données à un niveau plus détaillé ou au contraire plus agrégé ;
- la *rotation*, c'est à dire le changement d'orientation dimensionnelle dans la présentation des données, notamment par la permutation des lignes et des colonnes.

Ces opérations seront détaillées dans la section 6.4.

6.1.4 Simulation

La simulation a pour objectif d'évaluer les conséquences d'une hypothèse ou d'un scénario. Elle est caractérisée par des interrogations du type : « que ce passerait-il si ... ».

Elle nécessite donc de permettre à l'utilisateur d'entrer des données arbitraires et de tenir compte de ces données au même titre que les données authentiques. Il faut enfin que ces données arbitraires puissent être séparées des données authentiques lorsque la simulation est terminée pour être éliminées de la base.

La simulation implique donc de disposer d'une interface de dialogue appropriée pour exprimer les requêtes, restituer les résultats et saisir les données. Elle nécessite aussi de disposer d'un moteur de calcul capable d'assimiler immédiatement les données saisies par l'utilisateur.

6.1.5 Recherche de connaissances

La plupart des applications décisionnelles s'arrêtent à la mise en évidence de mesures déterminées par des associations de variables. L'étape la plus aboutie consiste à vérifier ou établir :

- des estimations ou prévisions de comportement
- des classifications, regroupements ou segmentations, ...

Ces applications relèvent du *Data Mining* et autres méthodes d'extraction de connaissance *Knowledge Mining*, *Knowledge Discovery in Data Bases*, ...

Les outils correspondants mettent en œuvre des techniques conçues autour de méthodes d'analyses statistiques (ACP, AFC, arbres de décision, réseaux neuromimétiques, régression linéaire, algorithmes génétiques, ...). Ils permettent en particulier de détecter l'existence d'influence de certaines variables sur certains indicateurs ou de trouver celles qui expliquent au mieux le comportement d'une classe particulière.

6.1.6 Alertes

Une *alerte* dans une base de données décisionnelle est liée à la présence d'une valeur ou d'une combinaison de valeurs considérées comme anormales ou remarquables et justifiant éventuellement une prise de décision.

La définition des alertes ou *exceptions* peut être déterminée par la simple comparaison d'un indicateur élémentaire avec une valeur de référence, ou encore à la répartition statistique d'un très grand nombre de valeurs élémentaires (par exemple l'enveloppe d'un nuage de points, ou la différence entre un écart instantané et un écart-type en longue période).

6.1.7 Mises à jour interactives

Bien que contraire aux principes de constitution d'un SID, il est parfois nécessaire que l'utilisateur puisse effectuer des saisies ou des modifications interactives de données persistantes.

Les saisies de ce type peuvent être considérées comme des sources de données parmi d'autres. Il faut cependant utiliser cette possibilité avec les précautions qui s'imposent pour ne pas risquer d'introduire d'incohérences.

6.1.8 Consultation de données opérationnelles

Au cours d'une session d'analyse, l'utilisateur peut avoir besoin de consulter des données reflétant directement les opérations de transactions, par exemple, lors de la découverte d'un fait exceptionnel. Il n'est pas envisageable de permettre ou de chercher à mettre en place un accès aux bases de données transactionnelles via le SDP, mais en revanche on peut autoriser la consultation des données du SCI. Celle-ci est acceptable dans les conditions suivantes :

- les procédures d'alimentation et de maintenance du SCI ont toujours priorité sur la consultation ;
- la consultation de la base de donnée du SCI ne produit que de la visualisation de données. Il n'est pas envisagé de fournir un système de requêtage quel qu'il soit ;
- les requêtes portent en général plutôt sur l'ODS que le DWH.

6.2 Traitement des agrégats

Le problème de cumuls pré-calculés complique toujours les applications associées à des base de données relationnelles. Les performances de SGBD relationnels classiques sont très mauvaises en matière de calcul de cumuls sur de très grandes collections de valeurs. Lorsqu'une requête demande le calcul d'un cumul (somme, moyenne, comptage) sur plusieurs centaines de milliers d'enregistrements de la table de faits, le délai d'attente est incompatible avec une utilisation interactive des données. Une solution est de calculer ce cumul (ou un ensemble de cumuls partiels dont il se déduit facilement) à l'avance et de le ranger dans la base de données.

Le stockage de ces agrégats pose divers problèmes dont celui de l'apparition de données redondantes. Pour gérer ce problème on a deux solutions :

- les tables de faits multiples ;
- les tables de faits uniques à plusieurs niveaux (contient les cumuls pré-calculés).

Pour représenter ces deux procédés considérons le schéma en étoile de la figure 2.16. Dans cet exemple, le grain est défini par la combinaison Produit / Client / Jour. Supposons que l'on veuille pré-enregistrer des cumuls par Groupe / Groupe économique / Mois.

La première option consiste à traiter chaque catégorie d'agrégats comme si elle correspondait au grain d'un contexte particulier, et à mettre en place une table de faits spéciale pour ce grain. On a donc deux tables de faits chacune correspondant à un niveau de consolidation comme le montre la figue 4.8.

Cette structure facile à représenter est cependant complexe à mettre en œuvre.

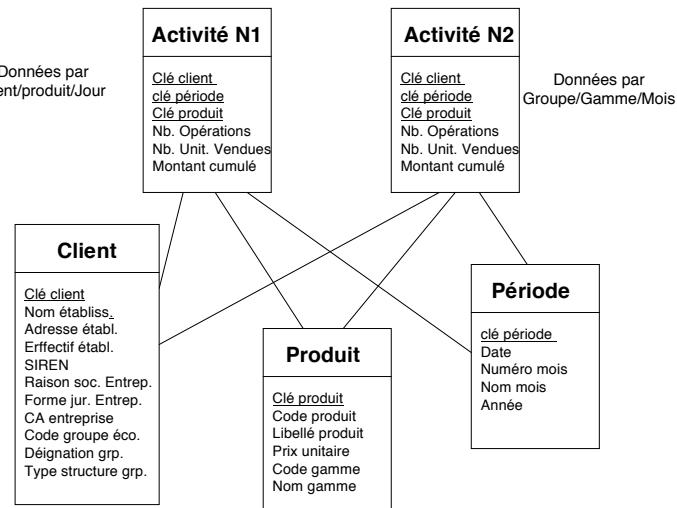


FIGURE 4.8 – Tables de faits spécialisés par niveau d'agrégation

Il y a une grande variété de niveaux envisageables, si on veut représenter tous les cumuls possibles. La pluralité des tables de faits complique l'utilisation de la base de données. En effet, les applications

doivent « savoir » choisir quelle table utiliser pour quelle requête.

La seconde option consiste à placer tous les cumuls pré-calculés dans la même table que les faits élémentaires. Il n'y a alors plus qu'une table de faits, mais celle-ci devient hétérogène.

6.3 Contextes résumés et partitions

Le pré-enregistrement de cumuls n'est pas le seul procédé d'optimisation logique. On peut également avoir recours à deux autres sortes de dénormalisation :

- les contextes résumés
- les partitions

Un contexte résumé est le reflet d'un contexte de base avec un niveau de grain plus synthétique. C'est donc une table de faits agrégés associée à la même structure dimensionnelle qu'un contexte détaillé. Par exemple, si dans une campagne d'analyse de comportement, 90% des requêtes sont focalisées sur des clients ayant acheté des produits électroménagers au cours de l'année 1996, on peut réduire le temps d'exécution de ces requêtes en créant un contexte dans lequel seuls figurent les clients ayant acheté ces produits dans cette période.

Le partitionnement consiste à découper le contexte sur un ou plusieurs segments. Par exemple, si une entreprise a ses forces de ventes organisées en 20 régions, un directeur régional n'a pas à s'intéresser au détail des affaires concernant les clients n'appartenant pas à sa région. On peut alors *partitionner* le contexte sur le critère régional, divisant par 20 la taille de la table de faits.

6.4 Optimisation des calculs sur les cubes

Le cœur de l'analyse multidimensionnelle est le calcul efficace d'agrégations sur plusieurs ensembles de dimensions. En langage SQL, ces agrégations sont référencées par des opérateurs **group-by**. Une approche de calcul sur les cubes consiste à étendre SQL à un opérateur **compute-cube**. L'opérateur **compute-cube** agrège tous les éléments des dimensions spécifiées dans l'opération.

6.4.1 Exemple 4.1.

Supposons qu'on ait créé un cube sur les ventes en Euros d'une entreprise qui contient les données suivantes : *produits*, *lieu*, *temps*, *fournisseur*, et *ventes_en_euros*. On souhaiterait pouvoir analyser les données avec des requêtes du type :

- Calculer la somme des ventes, groupées par produit et par lieu,
- Calculer la somme des ventes, groupées par produit,
- Calculer la somme des ventes, groupées par lieu.

Combien peut-on calculer de cuboïdes, ou de groupe-by, pour ce cube ? Si l'on prend les quatre attributs, *produits*, *lieu*, *temps*, et *fournisseur*, comme les quatre dimensions et *ventes_en_euros* comme mesure, le nombre total de cuboïdes, ou de groupe-by, est $2^4 = 16$. Les groupe-by possibles sont : $\{(produits, lieu, temps, fournisseur), (produits, lieu, temps), (produits, lieu, fournisseur), (produits, temps, fournisseur), (lieu, temps, fournisseur), (produits, lieu), (produits, temps), (lieu, temps), (produits, fournisseur), (lieu, fournisseur), (temps, fournisseur), (produits), (lieu), (temps), (fournisseur), ()\}$. Ces groupe-by forment un treillis de cuboïdes pour ce cube, tel que dans la figure 4.9. Le cuboïde qui contient le plus bas niveau d'agrégation est appelé cuboïde de base. Par exemple, sur la figure 4.9, le cuboïde de base contient les quatre dimensions *produits*, *lieu*, *temps*, et *fournisseur*. Il permet de retourner le total des ventes pour toute combinaison des quatre dimensions. Le cuboïde de dimensions 0-D, qui contient le plus haut niveau d'agrégation, est appelé cuboïde d'Apex. Dans l'exemple il représente la somme totale des ventes cumulées sur les quatre dimensions.

Une requête SQL qui ne contient aucune instruction groupe-by telle que, “calculer la somme totale des ventes”, est une *opération en dimension zéro*. Une requête SQL telle que “calculer la somme de ventes, groupées par lieu”, qui contient donc un groupe-by, est une *opération en dimension un*. Un opérateur sur un cube de dimension n est équivalent à un ensemble d'instructions groupe-by pour chaque sous ensemble des n dimensions. Un opérateur sur un cube est donc une généralisation n -dimensionnelle de l'opérateur **groupe-by**.

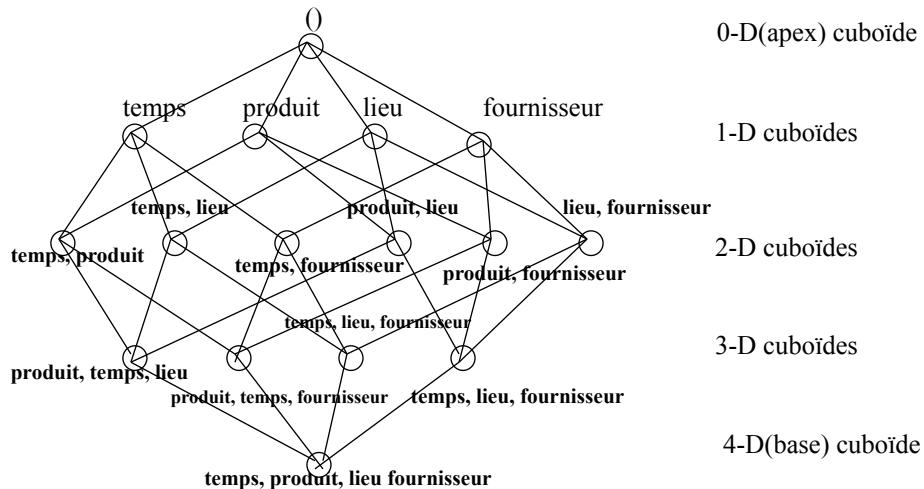


FIGURE 4.9 – Treillis de cuboïdes constituant un cube 4-D. Chaque cuboïde représente un groupe-by différent. Le cuboïde de base contient les quatre dimensions *lieu*, *temps*, et *fournisseur* (d'après Data Mining Concepts and Techniques, J. Han and M. Kamber, 2001, pp. 48)

Considérons, par exemple, le langage DMQL, qui est un langage de requêtes multi-dimensionnelles construit sur SQL et contenant des primitives pour définir les datawarehouses et les datamarts. La définition d'un cube et d'une dimension suit la syntaxe suivante :

```
define cube <nom_du_cube> [<liste_des_dimensions>] : <liste_des_mesures>
define dimension <nom_de_dimension> as (<liste_attributs_ou_sous-dimension>).
```

Ce qui donnera dans cet exemple :

```
define cube ventes [produit, lieu, temps, fournisseur] : sum(ventes_en_euros)
```

Et l'instruction :

```
compute cube ventes
```

demande explicitement au système de calculer les agrégats de ventes pour les seize sous-ensembles de *{produits, lieu, temps, fournisseur}*, l'ensemble vide inclus.

C'est donc sur cette base que sont calculés à l'avance l'ensemble des agrégats d'un cube, permettant ensuite de réaliser les opérations de navigation dans le cube très rapidement. Le problème est que la mémoire de stockage requise peut exploser si tous les cuboïdes du cube sont pré-calculés, en particulier si le cube a des dimensions associées à des hiérarchies.

En effet, un cube de dimension n possède 2^n cuboïdes. Cependant, si les n dimensions possèdent chacune L_i niveaux de hiérarchies, alors le nombre total de cuboïdes que l'on peut générer est donné par :

$$T = \prod_{i=1}^n (L_i + 1).$$

Pour un cube de dimension 10, avec chacune 4 niveaux de hiérarchies, le nombre total de cuboïdes serait $5^{10} \approx 9,8 \times 10^6$. Il n'est donc pas envisageable de les calculer tous, une option est d'en faire une *matérialisation partielle*, c'est à dire, de n'en calculer que quelques uns. Selon l'option choisie, on parlera de non-matérialisation, matérialisation partielle ou matérialisation totale.

6.4.2 Matérialisation partielle : calcul sélectif de cuboïdes

Il y a donc trois possibilités de matérialisation de cube pour un cuboïde donné :

1. **Pas de matérialisation** : on ne pré-calcule aucun des cuboïdes sauf le cuboïde de base.
2. **Matérialisation totale** : on pré-calcule tous les cuboïdes et on obtient dans ce cas un *cube complet*. Ce choix nécessite de disposer d'un volume mémoire très élevé pour enregistrer tous les cuboïdes pré-calculés.

3. Matérialisation partielle : on pré-calcule une sélection d'un sous-ensemble de tous les cuboïdes possibles. Une autre alternative est de ne calculer qu'un sous ensemble du cube, qui ne contiendrait que les cellules qui satisfont un critère de sélection, tel que ne garder que celles dont le total (i.e. count) est supérieur à un seuil. On parlera dans ce cas de *sous-cube*.

La sélection du sous-ensemble de sous-cubes ou cuboïdes à matérialiser doit prendre en compte les requêtes les plus fréquentes et leur coûts d'accès. Il faut considérer en plus les coûts de mises à jour incrémentales et les contraintes de stockage. Le contexte général de la conception physique de la base de donnée, telle que la génération et la sélection des indices, doit être aussi pris en compte. Des règles heuristiques peuvent également être utilisées, telles que ne matérialiser que les cuboïdes qui sont fréquemment utilisés par d'autres cuboïdes.

Enfin, une stratégie que l'on rencontre fréquemment consiste à ne matérialiser les cubes que sur deux ou trois dimensions. Pour les requêtes qui utilisent les autres dimensions, on complètera le calcul à la volée.

Une fois que les cuboïdes ont été matérialisés, il faut définir des stratégies pour choisir ceux qui sont pertinents pour une requête donnée, comment utiliser des structures d'index appropriées et comment exécuter les opérations OLAP sur les cuboïdes sélectionnés. Ces points sont brièvement discutés dans la section 6.5.

6.4.3 Méthodes d'agrégation dans les cubes

On peut avoir dans certain cas besoin de faire une matérialisation totale des cuboïdes. Il faut donc trouver des algorithmes optimaux pour réaliser les calculs, en prenant en compte la taille de la mémoire disponible et le temps d'exécution.

Dans le cas du OLAP relationnel (ROLAP) la structure de base des données est une table relationnelle et le calcul se fait sur des tuples, alors que dans le cas du OLAP multidimensionnel (MOLAP), la structure de base est un tableau multidimensionnel. On s'attend donc à ce que les techniques de calcul ne soient pas les mêmes.

Le calcul sur les cubes ROLAP s'appuie sur les techniques d'optimisation suivantes :

- les opérations d'ordonnancement, de hachage et de groupement sont appliquées sur les attributs des dimensions pour ré-ordonner et grouper les tuples correspondants,
- les groupements sont réalisés sur des sous-agrégats en "étapes de regroupements partiels". Ces groupements partiels peuvent être utilisés pour accélérer le calcul d'autres sous-agrégats,
- Les agrégats sont calculés à partir des sous-agrégats plutôt que par calcul direct sur les bases de faits.

Alors que le ROLAP utilise un accès de type clé-valeur pour ses stratégies de recherche, le MOLAP utilise un adressage direct, dans lequel on accède aux valeurs des dimensions par les coordonnées ou l'index de leur position dans le cube. Un MOLAP ne peut donc pas utiliser de ré-ordonnancement comme dans le cas du ROLAP. Les techniques d'optimisation sont donc les suivantes :

- partition du tableau en morceaux. Chaque morceau est un sous-cube qui est suffisamment petit pour réaliser les calculs en mémoire. Les morceaux sont compressés pour réduire l'espace perdu par les cellules vides (structures et algorithmes optimisés pour les données parcimonieuses),
- calcul des agrégats en accédant aux cellules (c'est à dire aux valeurs) du cube. On optimise le nombre de fois où chaque cellule est visitée.

6.5 Opérations OLAP typiques

Dans le modèle multidimensionnel, les données sont organisées en plusieurs dimensions et chaque dimension peut contenir plusieurs niveaux d'abstraction définis par les hiérarchies de concepts. Cette organisation offre aux utilisateurs une certaine flexibilité pour voir les données selon différentes perspectives. Il existe donc différentes opérations OLAP sur les données des cubes qui matérialisent ces différents points de vues, permettant le requêtage interactif, et l'analyse des données accessibles.

- **Roll-up** : le *roll-up* (ou encore *drill-up*) est une opération d'agrégation soit en remontant le long d'une hiérarchie de concept dans une dimension, soit par réduction de dimensions. Dans l'exemple de la figure 4.10, l'opération de roll-up est réalisée en remontant la hiérarchie *location* (localisation). Cette hiérarchie a été définie par un ordre total sur *rue < ville < province_ou_etat < pays*.

L'agrégation est réalisée par ascendance de la hiérarchie localisation du niveau *ville* au niveau *pays*. En d'autres termes le cube résultat groupe les données par pays au lieu de les grouper par ville comme dans le cube initial.

Un roll-up par suppression de dimension consisterait à ne pas considérer la dimension *temps* dans le cube et ainsi calculer les agrégats des ventes totales seulement par *localisation* et *catégorie*.

- **Drill-down** : le *drill-down* (ou encore *roll-down*) est l'inverse du roll-up et permet de naviguer du niveau le moins détaillé au niveau le plus détaillé, soit en descendant le long d'une hiérarchie de concept dans une dimension, soit en introduisant des dimensions additionnelles. La figure 4.10 montre le résultat d'une opération drill-down appliquée au cube central après avoir descendu le long de la hiérarchie *temps* définie par *jour < mois < trimestre < année*. Dans cet exemple on est passé du *trimestre* au *mois*.

Comme le drill-down ajoute plus de détails à un ensemble de données, il peut être réalisé par l'ajout de dimensions à un cube. Par exemple, un drill-down sur le cube central peut être réalisé en ajoutant une dimension telle que *type_consommateur*.

- **Slice et dice** : l'opération *slice* (projection) effectue la sélection d'une dimension du cube initial pour générer un sous-cube. La figure 4.10 illustre la sélection des données sur la dimension *temps* pour lesquelles on a fixé la période comme étant égale au premier trimestre (*time = Q1*). L'opération *dice* (sélection) définit une sélection sur deux ou plusieurs dimensions. Sur cette même figure on a sélectionné le sous-cube pour lequel (*location = Toronto or Vancouver*) and (*time = Q1 or Q2*) and (*item = home_entertainment or computer*).
- **Pivot (rotation)** : L'opération *pivot* (ou rotation) est une opération de visualisation dans laquelle on permute les axes pour obtenir une présentation alternative des données. Une autre possibilité consiste à transformer un cube en 3-D, en une série de plans en 2-D.
- **Autres opérations** : on peut trouver d'autres opérations telles que *drill-across* qui exécute des requêtes sur plus d'une table de faits, ou encore *drill-through* qui utilise les propriétés du SQL relationnel pour naviguer d'un cube aux tables relationnelles qui ont permis de le construire. On peut également réaliser des opérations qui vont ranger les *N* premiers ou *N* derniers items d'une liste, ou calculer des moyennes mobiles, des accroissements, intérêts, dépréciations, rendements, conversions de devises, et fonctions statistiques.

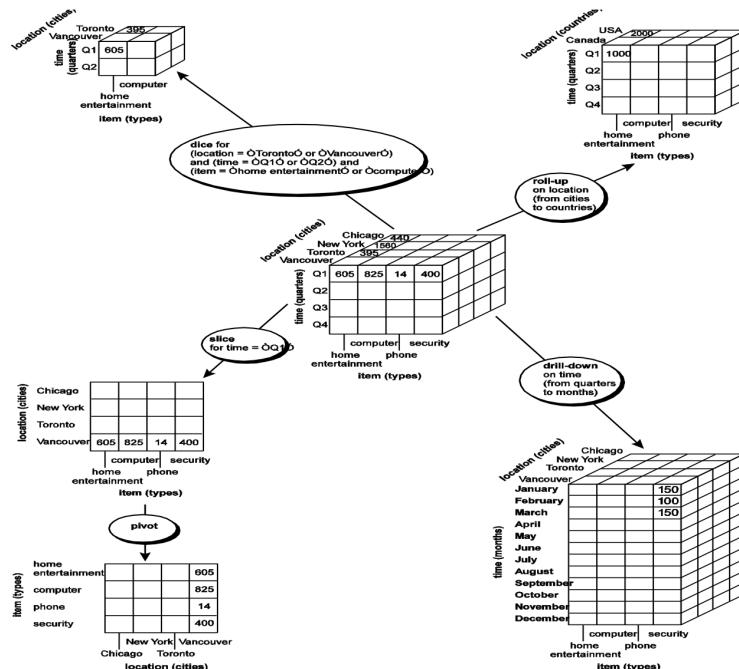


FIGURE 4.10 – Exemples d'opérations OLAP typiques sur des données multidimensionnelles (d'après Data Mining Concepts and Techniques, J. Han and M. Kamber, 2001, pp. 59)

L'OLAP offre des possibilités de modélisation analytique, y compris des moteurs de calcul pour dériver des ratios, variances, etc., et pour calculer des mesures sur plusieurs dimensions. Il peut générer des résumés, des agrégats, des hiérarchies à tout niveau de granularité et toute intersection de dimension. L'OLAP propose également des modèles fonctionnels de prévision, d'analyse de tendance et d'analyse statistique.

6.6 Indexation des données OLAP

Nous venons d'étudier ci-dessus les méthodes de sélection des cuboïdes à pré-calculer, nous examinons ici deux méthodes d'indexation des données OLAP, *l'indexation bitmap* et *l'indexation par jointure*.

L'indexation bitmap est très utilisée en OLAP car elle permet une recherche rapide dans les cubes de données. Un index bitmap est une alternative aux listes d'enregistrements *record_ID (RID)*. Pour un attribut donné dans un index bitmap, on définit un vecteur binaire, Bv , pour chaque valeur v du domaine de l'attribut. Si le domaine d'un attribut donné contient n valeurs, alors il faut avoir n bits pour chaque entrée de l'index bitmap (c'est à dire qu'on a des vecteurs n binaires). Si un attribut d'une ligne de la table de données a la valeur v , alors le bit représentant cette valeur est mis à 1 dans la ligne correspondante de l'index bitmap. Tous les autres bits sont mis à zéro.

6.6.1 Exemple 4.2.

Supposons qu'une dimension d'un *Client* ait six valeurs représentant des catégories socio-professionnelles : *Agriculteur*, *Employé*, *Ouvrier*, *Cadre*, *Artisan*, et *Commerçant*. Chaque valeur de *Client* (par exemple, *Employé*) est représentée par un vecteur binaire dans la table d'index binaires. Supposons que le cube soit stocké dans une table de relations de 100 000 lignes. Puisque le domaine *Client* comporte six valeurs, l'index binaire nécessite d'avoir des vecteurs six bits. La table 4.1 est un extrait d'une base contenant les dimensions *Profession* et *Ville*, et ses projections dans des tables d'index binaire 4.2 pour chacune des dimensions.

La méthode d'indexation binaire est avantageuse comparée aux méthodes de hachage et d'arbres particulièrement quand la cardinalité du domaine de variation est faible, parce que les opérations de comparaisons, jointures et agrégations sont réduites à une arithmétique binaire, qui raccourci les temps de calcul de façon significative. De plus les opérations d'entrée/sortie n'utilisent qu'un faible espace mémoire car les chaînes de caractères sont remplacées par un seul bit.

Code Client	Profession	Ville
AZ001	Agric.	Roye
XB425	Cadre	Compiègne
ZS456	Ouvrier	Compiègne
KX223	Employé	Roye
BF322	Artisan	Roye
AB652	Cadre	Compiègne
GC278	Agriculteur	Roye
FG054	Commerçant	Compiègne

TABLE 4.1 – Table relationnelle *Client*

Code Client	Agric.	Empl.	Ouvrier	Cadre	Artisan	Comm.
AZ001	1	0	0	0	0	0
XB425	0	0	0	1	0	0
ZS456	0	0	1	0	0	0
KX223	0	1	0	0	0	0
BF322	0	0	0	0	1	0
AB652	0	0	0	1	0	0
GC278	1	0	0	0	0	0
FG054	0	0	0	0	0	1

Code Client	Roye	Compiègne
AZ001	1	0
XB425	0	1
ZS456	0	1
KX223	1	0
BF322	1	0
AB652	0	1
GC278	1	0
FG054	0	1

TABLE 4.2 – Index binaire sur les propriétés *Profession* et *Ville*

L'**indexation par jointure** vient de son utilisation dans les procédures de requête dans les bases de données relationnelles. La jointure traditionnelle projette la valeur d'une colonne donnée sur une liste de lignes qui contiennent cette valeur. Au contraire, l'indexation par jointure enregistre les lignes qui peuvent être jointes par deux relations d'une base de données relationnelle. Par exemple, si deux relations $R(RID, A)$ et $S(B, SID)$ se joignent sur les attributs A et B , alors l'enregistrement des index joints contient les paires (RID, SID) , où RID et SID sont respectivement les identifiants d'enregistrement des relations R et S . En conséquence, l'indexation de jointure permet d'identifier deux tuples qui peuvent être joints sans réaliser d'opération de jointure coûteuse. Elle est particulièrement utile pour maintenir une relation entre une clé étrangère et sa clé primaire correspondante dans une relation de jointure.

Le modèle en étoile des datawarehouses est particulièrement intéressant pour les recherches à travers une table, car les liens entre un fait de la table de faits et les tables des dimensions correspondantes sont justement les clés étrangères de la table de faits et la clé primaire de la table de dimension. L'indexation de jointure maintient les relations entre les valeurs des attributs d'une dimension et les lignes correspondantes de la table de faits. L'index de jointure peut s'étendre sur de multiples dimensions pour former les **index de jointure composés**. Ils peuvent être utiles pour identifier des sous-cubes présentant un intérêt.

6.6.2 Exemple 4.3.

Considérons un schéma en étoile correspondant aux ventes de produits électroniques défini dans l'exemple 4.1. : *ventes_etoile* [*produit*, *lieu*, *temps*, *fournisseur*] : *euros_vendus* =**sum**(*ventes_en_euros*). Un exemple d'index de jointure sur la relation entre la table de faits *ventes* et les tables de dimension *lieu* et *produit* est donné dans la figure 4.11. Par exemple la valeur “*Rue Principale*” de la dimension *lieu* est reliée aux tuples T57, T238 et T884 de la table de faits *ventes*. De même, la valeur “*TV-Sony*” de la dimension *produit* est reliée aux tuples T57 et T459 de la table de faits *ventes*. Les tables 4.3 donnent les correspondances d'index de jointure.

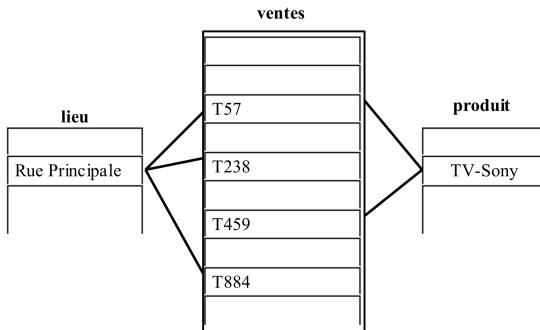


FIGURE 4.11 – Liens entre la table de faits *ventes* et les tables de dimension *lieu* et *item*

Supposons que l'on ait 360 valeurs de temps, 100 produits, 50 fournisseurs, 30 lieux et 10 millions de tuples ventes dans le cube en étoile. Si la table de faits n'a enregistré des ventes que pour 30 produits, alors les 70 autres produits ne participeront pas aux jointures.

lieu	clé vente	produit	clé vente	lieu	produit	clé vente
...
Rue Principale	T57	TV-Sony	T57	Rue Principale	TV-Sony	T57
Rue Principale	T238	TV-Sony	T459
Rue Principale	T884
...

TABLE 4.3 – Tables d'index de jointure *lieu/ventes*, *produit/ventes*, *lieu/produit/vente*

Chapitre 5

Exemples de Modèles Multidimensionnels¹

1. Démarche générale

Quelle que soit l'application envisagée, la conception d'une base de données multidimensionnelle doit passer par 4 étapes :

- *Sélectionner le processus de l'entreprise à modéliser* : on ne fait pas référence à un service ou une fonction d'une organisation (un modèle pour les commandes client et pas un modèle pour le service vente et un pour le service marketing) ;
- *Déclarer le grain du processus* : exemple une ligne d'un ticket d'achat, un instantané de stock, un instantané mensuel d'un compte en banque ;
- *Choisir les dimensions qui s'appliquent à chaque ligne de la table de faits* : en répondant à la question « comment les gestionnaires décrivent-ils les données qui résultent du processus concerné ? ». Si le grain est clair les dimensions sont claires ;
- *identifier les faits numériques qui vont renseigner chaque ligne de la table de faits* : en répondant à la question « que mesurons-nous ? » : on veut faire la mesure de performance sur les valeurs enregistrées au cours du processus.

Il faut tenir compte à chacune de ces étapes, à la fois des besoins des utilisateurs et des données sources effectivement disponibles.

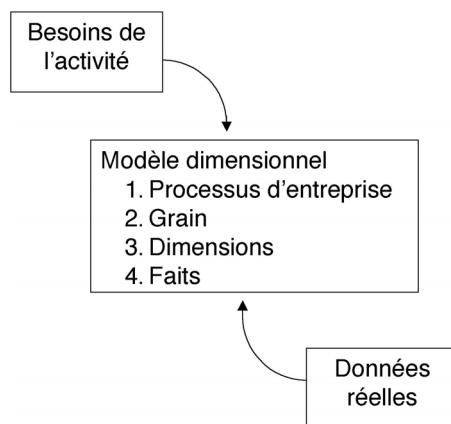


FIGURE 5.1 – Les deux éléments clés contribuant aux quatre étapes du processus de modélisation dimensionnelle

1. Ce Chapitre est tiré de Kimball et Ross, pp. 31-90.

2. Etude de cas de la distribution

- On considère le cas d'une entreprise qui comprend 100 magasins d'alimentation répartis sur 5 régions.
- Chaque magasin est un supermarché classique avec épicerie, produits laitiers, boucherie, surgelés, fruits et légumes, boulangerie, fleurs, parapharmacie.
- Chaque magasin a sur ses étagères environ 60 000 produits appelés unités de stock : US.
 - 55 000 US viennent de fabricants extérieurs avec un code barre imprimé sur l'emballage (CUP : code universel de produits). Chaque variante d'emballage a un CUP différent et est une US différente, le grain du CUP est donc le même que l'US.
 - Les 5000 US restants proviennent de rayons tels fruits et légumes, fleurs ou boulangerie. Ils n'ont pas de CUP, mais on leur attribue des étiquettes scannérables, qui ne sont pas des CUP mais qu'on peut assimiler à des US.
- La collecte des données s'effectue au niveau du système TPV (terminaux points de vente)
- Objet de l'étude : analyse de l'activité du magasin pour définir une politique de prix (promotions, marge, ...), de logistique de commandes, de stockage, de coûts de gestion (abaissement du prix d'achat pour atteindre l'objectif de vendre le plus cher possible, en achetant et en dépensant le moins cher possible).

3. Les quatre étapes de cette application

3.1 Etape 1 : sélection du processus à modéliser

Modélisation des terminaux point de vente (TPV) pour mieux comprendre les achats des clients : quels produits se vendent, dans quels rayons, quels magasins, quels jours, dans quelles conditions de promotion. Un autre point d'observation du processus pourrait être, par exemple, les livraisons.

3.2 Etape 2 : déclaration du grain

On va chercher à rester au niveau le plus atomique possible : la donnée la plus granulaire est une ligne de transaction. On va mettre l'analyse du grain en regard des questions que l'on peut se poser : ventes du lundi par rapport aux autres jours, d'une marque par rapport au stock, ou des promotions, ...

3.3 Etape 3 : choix des dimensions

- Les dimensions qui s'imposent sont : temps, produit et magasin.
- On ajoute la dimension promotion, car c'est un élément important de caractérisation de l'activité
- on est dans une modélisation périodique. On pourrait envisager d'inclure une information sur la transaction (le numéro du ticket), qui correspondrait à une modélisation événementielle. On ajoutera cette dimension de façon très particulière plus tard dans l'étude.

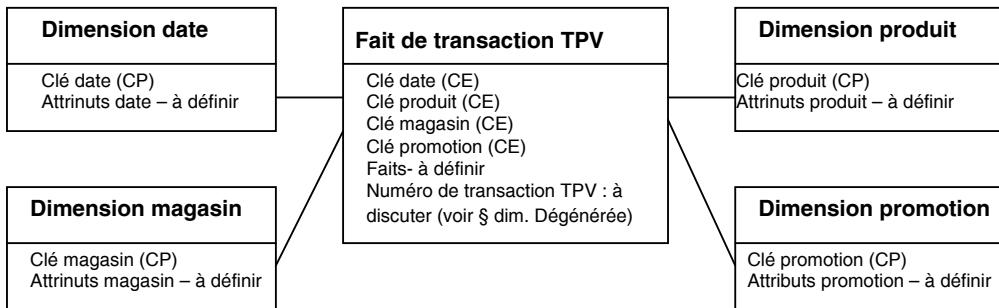


FIGURE 5.2 – Schéma primaire des ventes en grande distribution

3.4 Etape 4 : Identification des faits

Le choix sera fonction du grain : la ligne individuelle d'article d'une transaction TPV

- quantité vendue (nombre de boîtes de petits pois) pour chaque prix unitaire
- montant de la vente (en euros) = prix unitaire × quantité
- coût du produit livré par le fournisseur

On pourrait ajouter le *bénéfice brut* = *prix - coût*, auquel cas tous les éléments seraient additifs dans toutes les dimensions. On lui préférera la *marge brut* = (*bénéfice brut*) ÷ (*chiffre d'affaire*). Le chiffre d'affaire peut-être calculé sur un ensemble de produits, de magasins ou de jours. Ce fait ne sera pas additif.

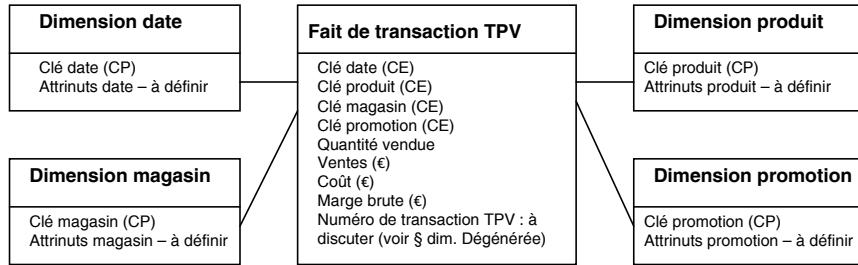


FIGURE 5.3 – Faits mesurés du schéma vente au détail

Remarque : le prix unitaire des produits n'est pas additif.

4. Attributs de table de dimension

4.1 Dimension date

- C'est une table de taille relativement modeste (dix ans = 3650 lignes)
- Elle peut être construite à l'avance (on peut prévoir 5 ou 10 ans de lignes)
- le contenu de chaque colonne (tableau 5.1) dépend du jour particulier que la ligne représente :
 - jour de la semaine permet de comparer *lundi / jeudi*;
 - quantième du jour permet de comparer des jours à la même distance du début du mois. On pourrait faire de même pour les numéros de mois dans l'année ;
 - les noms de mois et numéro années sont utiles pour les états, idem pour les trimestres.
- la notation *férié* ou *non férié* est plus facile à manipuler pour les états que OUI/NON qui ne sont pas des valeurs explicites,
- la colonne *saison* sert à marquer des périodes particulières telles que Noël, Pâques, St. Valentin. De même la colonne *événements* sert à marquer des événements sportifs, sociaux, etc.
- L'heure de la transaction n'apparaît pas ici car, d'une part, elle engendrerait une table de taille démesurée (5 256 000 lignes pour 10 ans) et, d'autre part, l'heure de transaction peut-être considérée indépendante de la date. On peut alors croiser une table de 3560 lignes de dates avec une table de 1440 lignes de minutes si on veut pouvoir rendre compte de phénomènes tels que rush en soirée, changement d'équipe, etc.

Clé date	Date	Date description complète.	Jour de la semaine	Mois calendrier	Année calendrier	Année-mois d'exercice	Indicateur de jour férié	Indicateur j. de sem.
1	01/01/2002	1er janvier 2002	mardi	janvier	2002	E2002-01	férié	j. de sem.
2	02/01/2002	2 janvier 2002	mercredi	janvier	2002	E2002-01	non férié	j. de sem.
3	03/01/2002	3 janvier 2002	jeudi	janvier	2002	E2002-01	non férié	j. de sem.
4	04/01/2002	4 janvier 2002	vendredi	janvier	2002	E2002-01	non férié	j. de sem.
5	05/01/2002	5 janvier 2002	samedi	janvier	2002	E2002-01	non férié	week-end
6	06/01/2002	6 janvier 2002	dimanche	janvier	2002	E2002-01	non férié	week-end
7	07/01/2002	7 janvier 2002	lundi	janvier	2002	E2002-01	non férié	j. de sem.
8	08/01/2002	8 janvier 2002	mardi	janvier	2002	E2002-01	non férié	j. de sem.

TABLE 5.1 – Détail d'une table de dimension date

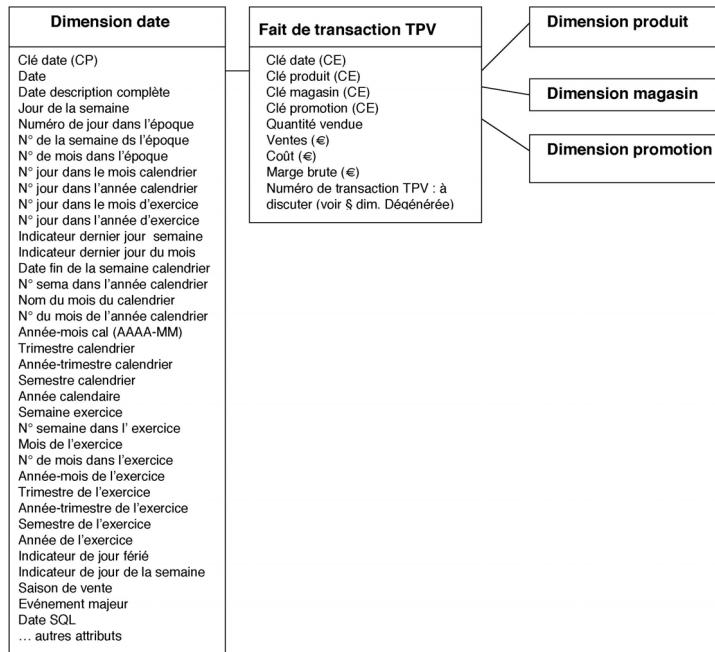


FIGURE 5.4 – Schéma d'une dimension date pour la distribution

4.2 Dimension produit

- Chaque magasin peut avoir 60 000 produits en stock mais, compte tenu des différences de merchandising d'un magasin à l'autre et des produits qui ne sont plus disponibles, la dimension peut atteindre au moins 150 000 lignes, voire un million.
- La dimension est obtenue à partir d'un fichier maître produit de l'application opérationnelle. Dans ce fichier sont définis les nouveaux produits, les règles d'attribution des US aux produits tels que la boulangerie, fruits et légumes, etc. Ces mises à jour sont extraites à intervalles réguliers pour les placer dans la dimension produit.
- On trouve dans cette dimension une hiérarchie, construite à partir des informations fournies par le fichier maître, qui permet les regroupements de produits en marques, en catégories, en rayons.
- tous les niveaux de la hiérarchie sont bien définis pour chaque produit. On peut donc avoir 150 000 valeurs différentes de description de produits, mais seulement 50 valeurs de rayons. Cette valeur sera donc répétée en moyenne 3000 fois. Cette redondance est sans importance.
- certains attributs ne font pas partie de la hiérarchie ; l'emballage par exemple.

Clé produit	Description du produit	Description marque	Description de la catégorie	Description du rayon	Contenu en mat. grasses
1	Pain frais bien cuit au levain	Bien cuit	Pain	Boulangerie	Réduit
2	Au froment complet en tranches	Soufflé	Pain	Boulangerie	Normal
3	Au froment complet léger en tranches	Soufflé	Pain	Boulangerie	Réduit
4	Petits pains à la canelle sans matière grasse	Léger	Pain sucré	Boulangerie	Sans mat. gr.
5	5 litres glace vanille légère	Paq Froid	Desserts glacés	Surgelés	Sans mat.gr.
6	Demi-litre glace à la noix de pécan	Froidure	Desserts glacés	Surgelés	Réduit
7	1 litre glace des amis du chocolat	Superchoco	Desserts glacés	Surgelés	Normal
8	Demi-litre glace fraise	Le glacier	Desserts glacés	Surgelés	Normal
9	Sandwiches à la crème glacée	Le glacier	Desserts glacés	Surgelés	Normal

TABLE 5.2 – Détail d'une table de dimension produit

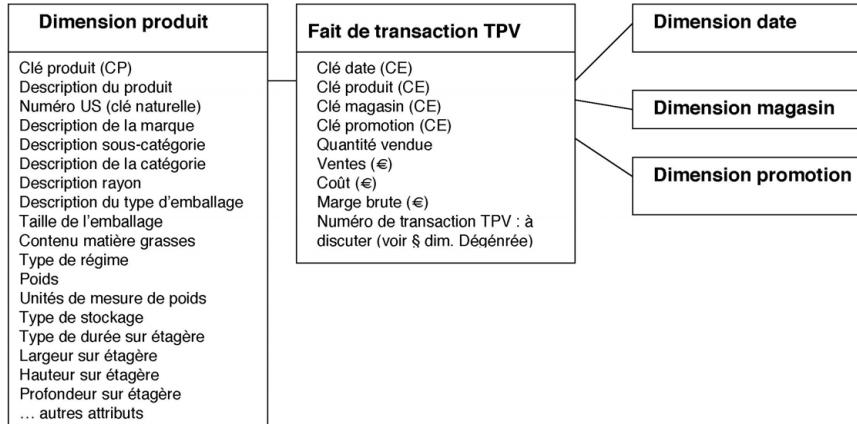


FIGURE 5.5 – Schéma d'une dimension produit pour la distribution

4.3 Dimension magasin

- il n'y a pas toujours de fichier maître magasin, comme pour les produits
 - alors que le fichier maître produit doit être téléchargé dans tous les magasins à chaque fois qu'il y a un changement, les systèmes TPV n'ont pas besoin d'un fichier maître des magasins.
 - l'élaboration de cette dimension demande de saisir l'information à différentes sources.
 - c'est une dimension géographique. Les magasins correspondent à un emplacement géographique (code postal, département ou région). Ils peuvent par ailleurs être regroupés par districts ou zones régionales définis par l'entreprise. Il y a donc deux hiérarchies possibles.
 - il y a un certain nombre de descripteurs textuels explicites : type agencement de la surface, type de développement photo, type de services commerciaux.
 - la surface est une valeur numérique, additive, mais c'est bien un attribut constant de magasin, pas un fait, qui peut servir de contrainte ou d'intitulé. Elle est donc mise dans la dimension magasin
 - La date du premier jour d'ouverture et la date de la dernière rénovation sont typiquement des clés de jointure vers des copies de la table de dimension date. Ces copies de la dimension date sont déclarées en SQL par une construction VIEW et sont sémantiquement distinctes de la clé primaire de la dimension date. La déclaration VIEW pourrait prendre la forme :
- ```
CREATE VIEW PREMIERE_DATE_OUVERTURE (PREMIER_JOUR_OUVERT, PREMIER_MOIS_OUVERT, ...)
AS SELECT NUMERO_JOUR, MOIS, ...
FROM DATE
```

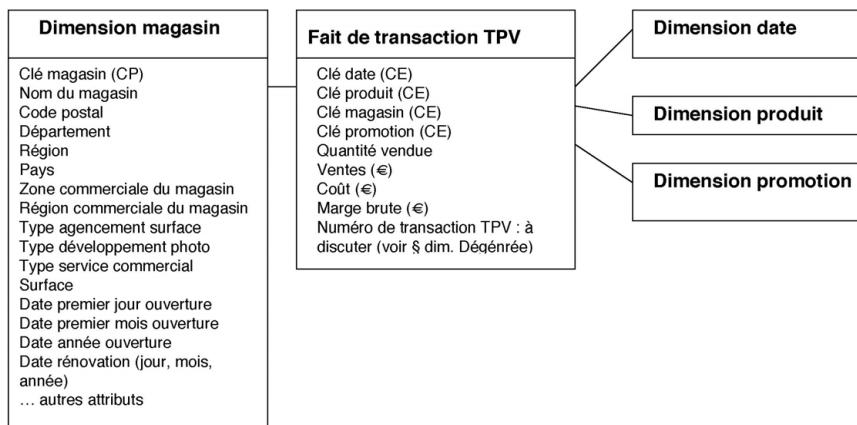


FIGURE 5.6 – Schéma d'une dimension magasin pour la distribution

#### 4.4 Dimension promotion

- Décrit chacune des modalités de promotion de chaque produit (réduction de prix temporaire, présentation en tête de gondole, annonce dans les journaux, ...)
- on veut pouvoir évaluer l'effet d'une promotion selon des critères tels que :
  - y a-t-il eu augmentation des ventes au cours de la promotion (*gain*) ? C'est une mesure à partir des ventes *normales* sur la même période, construites à partir de l'historique des ventes antérieures.
  - y a-t-il eu baisse de ventes avant ou après promotion ?
  - y a-t-il eu diminution d'autres produits simultanément (voisinage de rayon) ?
  - tous les produits de la catégorie en promotion ont-ils connu une augmentation sur les mêmes périodes de mesure ?
  - la promotion a-t-elle été profitable ?

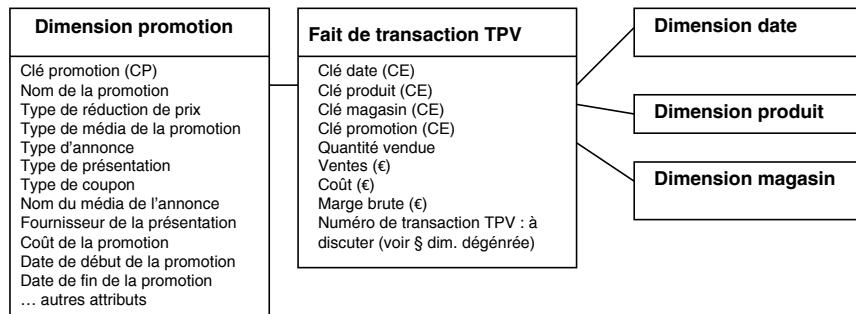


FIGURE 5.7 – Schéma d'une dimension promotion pour la distribution

- les modalités de promotion ne peuvent pas toujours être saisies au niveau du TPV. Certaines oui (le coupon de réduction), d'autres non (la publicité dans un journal). C'est donc une alimentation multisource.
- les modalités de promotion sont fortement corrélées, elles doivent donc être dans la même dimension. Par exemple sur une année, il peut y avoir 1000 annonces, 5000 réductions de prix temporaires et 1000 présentations en tête de gondole, mais seulement 10 000 combinaisons de ces trois types de promotion pour un produit. On peut donc prévoir une ligne avec les trois types simultanés et trois lignes avec seulement deux.
- on pourrait cependant imaginer une dimension spécifique pour les types de promotion (une dimension de type indicateur qualifié?)
- il faut prévoir une clé *vente sans promotion*, qui concerne la majorité des ventes, pour éviter une valeur nulle dans la table promotion pour l'immense majorité des ventes ordinaires.

#### 4.5 Table de faits sans fait relative aux promotions

Comment mesurer les produits qui étaient en promotion et qui n'ont pas été vendus (on ne mesure que ce qui a été vendu) ? On crée donc une table de faits qui associe les clés des produits, des dates, des magasins et des promotions, mais qui ne rend compte que de cette association, sans mesure de vente. Elle est donc sans fait, mais servira à mesurer l'efficacité d'une promotion.

#### 4.6 Dimension numéro de transaction dégénérée

Le numéro de transaction TPV devrait en principe être utilisé pour construire un autre contexte de type événementiel. Dans un tel contexte ce numéro servirait de clé pour enregistrer une transaction qui réunit toutes les informations valides d'une transaction, telles que la date et l'identifiant du magasin. Dans le modèle dimensionnel qui a été construit, les identifiants de ces informations figurent dans les tables de dimensions et non dans la table de faits. Cependant le numéro de transaction reste utile car il permet de regrouper tous les produits achetés au cours d'une même transaction.

Bien que le numéro de transaction TPV apparaisse comme une clé de dimension dans une table de faits, tous les items descriptifs d'une transaction qui devraient apparaître dans la table de dimension correspondante ont été retirés. Cette dimension étant donc vide, on réfère le numéro de transaction TPV en tant que *dimension dégénérée*. Le ticket de caisse figure dans le table de fait à titre de numéro de transaction TPV sans jointure vers une table de dimension.

Dans notre exemple, la clé primaire d'une vente dans la table de fait est constituée de la clé dégénérée du numéro de transaction TPV et de la clé produit.

La figure 5.8 est une illustration d'une requête associant les produits d'une transaction.

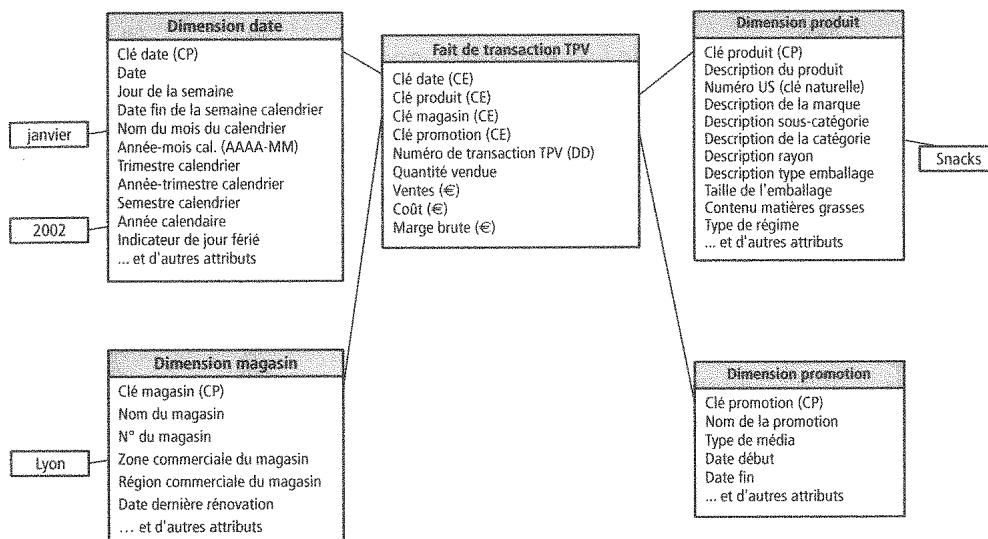


FIGURE 5.8 – Requête sur le schéma de vente au détail

## 4.7 Extensibilité du modèle

Il est intéressant de s'interroger sur les capacités de notre modèle à subir des transformations qui ne remettent pas en cause sa conception initiale. Si on décide de mettre en place un programme de fidélisation qui permet d'identifier des profils de consommateurs. On est plus intéressé par exemple de savoir que tel consommateur achète régulièrement tel ensemble de produits toutes les semaines plutôt que de connaître exactement le contenu d'un caddy.

La gestion de ce type de besoin à partir du modèle initial est assez simple :

- on crée une table de dimension consommateur régulier,
- on ajoute la clé étrangère correspondante dans la table de faits,
- comme on ne peut pas demander aux consommateurs de rapporter leurs anciens tickets de caisse pour relier leur nouvel identifiant d'acheteur régulier à l'historique des ventes, on devra substituer une clé "consommateur avant programme acheteur régulier" à la clé acheteur dans la table d'historique. De même, tous les clients ne seront pas des consommateurs réguliers et n'auront pas de carte de fidélité, il faudra ajouter une clé "consommateur régulier non identifié". Elle a pour effet d'éviter d'avoir des clés qui pointent vers des enregistrements nuls dans la table de faits.

L'ajout de cette nouvelle dimension peut être complété par l'ajout d'une dimension de l'heure du jour et du vendeur associés à la transaction, comme indiqué dans la figure 5.9.

## 4.8 Normalisation des dimensions

La dénormalisation de toutes les tables peut poser quelques problèmes aux concepteurs de modèles. Revenons sur la table de dimension produits. Les 150 000 produits sont enregistrés dans 50 départements

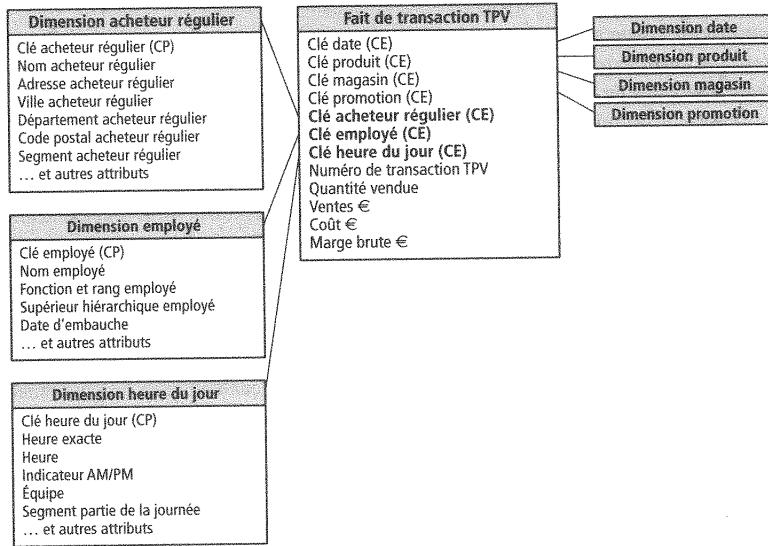
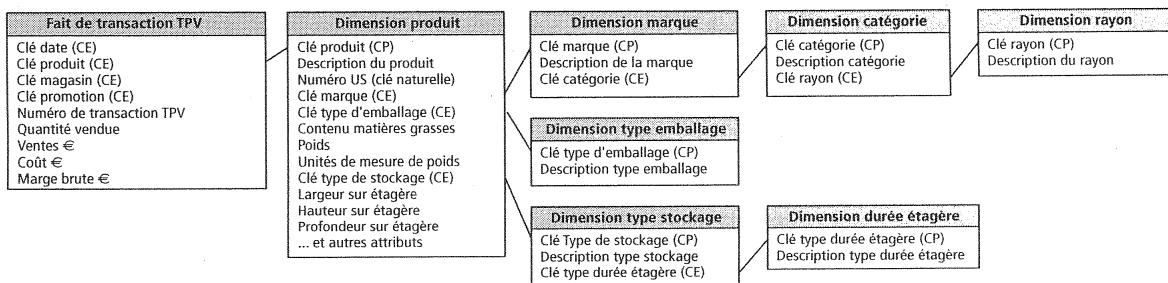


FIGURE 5.9 – Extension du modèle des ventes

distincts. Plutôt que d'enregistrer une description du département sur 20 octets, on pourrait enregistrer le code département sur 2 octets et créer une nouvelle dimension pour chaque département. L'argument en faveur de cette normalisation est que l'on n'enregistre que des codes cryptés dans la table de 150 000 lignes et non de longs descripteurs.

Un autre argument est que la normalisation facilite la maintenance. Si la description d'un département change, il n'y a qu'une occurrence à changer plutôt que ses 3000 répétitions dans la table originale. Cependant il faut se méfier de cet argument maintenance en faveur de la normalisation, car il n'est pas nécessairement compatible avec la notion de conservation d'historique.

La normalisation va avoir un effet de *floconisation*. Les attributs redondants sont supprimés de la dimension dénormalisée et placés dans des tables de dimension secondaires normalisées. La figure 5.10 est une illustration d'une floconisation partielle avec la dimension produits. La table de faits reste identique à celle de la figure 5.8, mais le nombre de tables de la dimension produits devient rapidement pléthorique.



Dimension produit partiellement en flocons de neige

FIGURE 5.10 – Effet de floconisation par normalisation d'une dimension

Bien que la floconisation puisse sembler une extension légitime du modèle dimensionnel, il faut garder à l'esprit que :

- la multiplication des tables en flocons complexifie le modèle et sa lisibilité,
- le nombre de jointures et la complexité des requêtes sont largement augmentés,

- l'espace disque sauvegardé reste faible. Si on remplace la description sur 20 octets des 150 000 lignes de la table de dimension départements par des codes sur 2 octets on a bien sauvegardé 2,7 MO ( $150\,000 \times 18$  octets), mais la table de faits occupe 10 GO. Les tables de dimension sont dans une proportion géométrique de la table de faits et les efforts de normalisation pour gagner de l'espace disque sont de peu d'effet,
- la floconisation est un obstacle à la navigation à l'intérieur d'une dimension, car celle-ci suppose de contraindre un ou plusieurs attributs d'une dimension pour en observer un autre en présence de ces contraintes. La navigation a, en effet, pour objectif de comprendre les relations entre les valeurs d'attributs,
- enfin, la floconisation empêche l'utilisation d'indexées binaires, qui eux, apportent un véritable gain en performance.

## 4.9 Trop de dimensions

La table de faits est naturellement très normalisée et compacte. On ne peut pas normaliser plus cette table car par définition les relations entre les clés de la table de faits portent sur des dimensions qui ne sont pas correlées. Tôt ou tard, la plupart des produits finissent par être vendus en promotion dans presque tous ou tous les magasins.

On peut être tenté de dénormaliser la table de faits en y ajoutant des clés étrangères pour les éléments qui sont fréquemment analysés dans la hiérarchie des produits tels que les marques, les sous-catégorie, les catégories ou les départements. De même la clé date peut donner lieu à un ensemble de clés pour joindre les semaines, les mois, les trimestres et les années de la dimension date. On risque alors d'arriver à une architecture telle que présentée sur la figure 5.11 qui fait penser à un mille pattes. Bien entendu cette architecture n'est pas à recommander car elle augmentera l'espace disque nécessaire au stockage de façon significative.

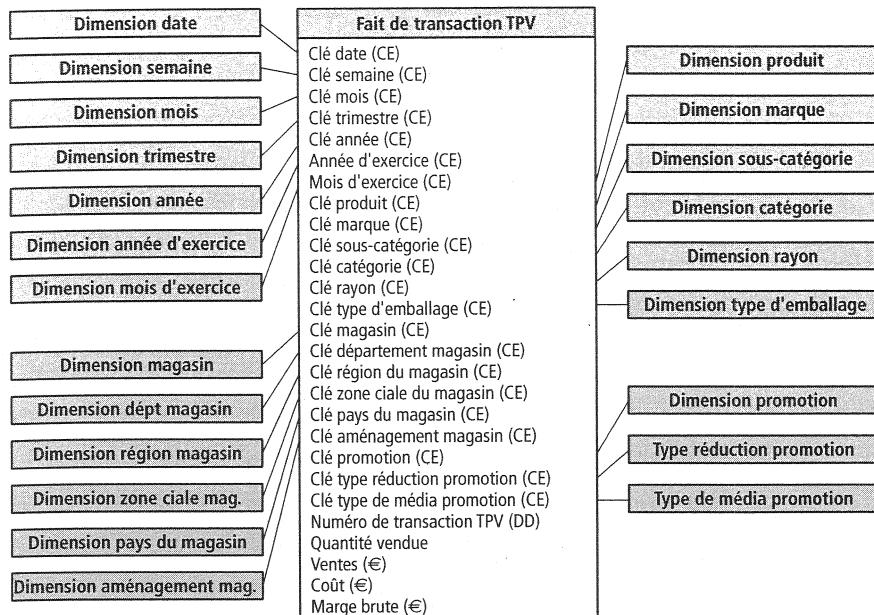


Table de faits mille-pattes avec un trop grand nombre de dimensions

FIGURE 5.11 – Table de faits mille pattes avec trop de dimensions

## 5. Etude de cas : Les stocks dans le magasin

Cet exemple s'applique à une grande variété de cas (distribution, production, etc.). On peut considérer des cas de distribution mais également la gestion d'entrepôts etc.

### 5.1 La chaîne de valeur

La plupart des organisations ont une chaîne de valeurs qui décrit les étapes clé de leur procédure. Elle identifie les flux naturel des activités essentielles de l'organisation. Dans le cas d'un système de distribution, une entreprise émet un ordre de commande à un fabricant, les produits sont livrés dans un entrepôt où ils sont inventoriés. Ils sont ensuite acheminés vers un magasin de détail où ils sont achetés par des clients. Un exemple de chaîne de valeur est donné figure 5.12.

Un objectif premier d'un système d'aide à la décision est de donner des éléments qui permettent de surveiller la performance de chaque étape de processus clé. Chaque processus ayant sa propre métrique, ses intervalles de temps, sa granularité et sa dimension, donne lieu à une table de fait. Dans ce but, la chaîne de valeur donne une image globale de l'ensemble du datawarehouse de l'entreprise.

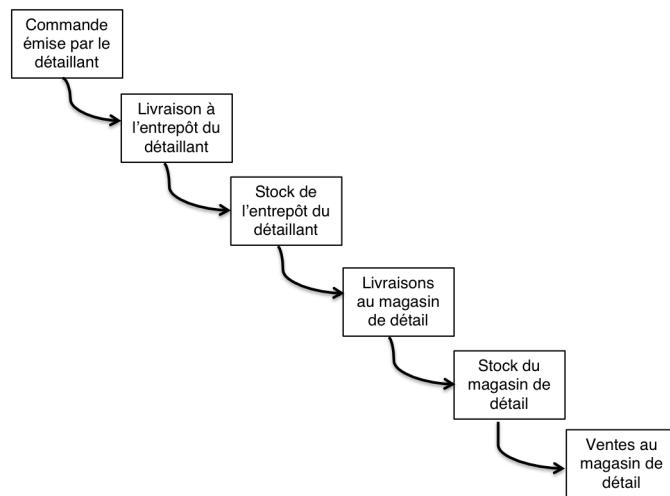


FIGURE 5.12 – La chaîne de valeur

### 5.2 Modèles de stock

Il y a trois types de modèles de stock : l'instantané périodique, l'enregistrement des transactions modifiant les niveaux de stocks, l'instantané récapitulatif de stock qui garde la trace du produit tant qu'il est sur le lieu de stockage.

#### 5.2.1 Instantané périodique de stock

On peut utiliser la même démarche que pour le magasin. Optimiser l'inventaire des produits peut avoir un impact sur l'optimisation du profit ou du fonctionnement des magasins. A l'inverse, une rupture de stock va générer un ensemble de problèmes.

- Les quatre étapes du modèle de dimensionnel
- processus : stock d'un magasin de vente
- grain : stock journalier, par produit dans chaque magasin
- dimensions : date, produit, magasin
- la dimension date est identique à celle de la distribution

- la dimension produit est également identique mais nécessite quelques attributs supplémentaires (quantité minimale pour le réapprovisionnement, descripteurs des unités de produits, ...).
- la dimension magasin est identique au cas précédent, mais peut nécessiter des attributs supplémentaires tels que la surface de stockage des produits surgelés ou des produits réfrigérés par exemple.
- faits : la quantité disponible

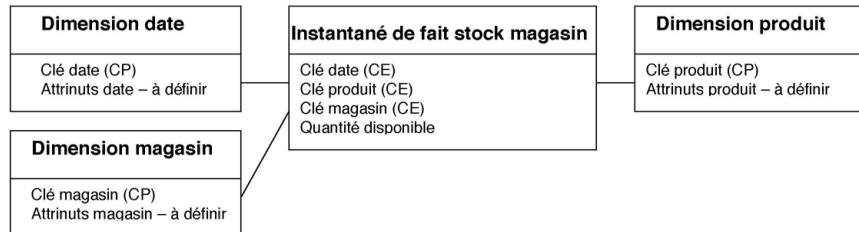


FIGURE 5.13 – Schéma de l'instantané périodique de stock. Si on analyse les niveaux de stock dans des entrepôts plutôt que dans des magasins, la modélisation sera identique, avec entrepôt à la place de magasin.

- Alors que la table de fait des ventes était relativement épars (10% des produits sont vendus par chaque magasin chaque jour), la quasi totalité des produits est présente chaque jour.
- 60 000 produits dans 100 magasins génèrent 6 000 000 de lignes. Si la largeur de la table est de 14 octets (une ligne) la table de fait augmente de 84 Mo à chaque mise à jour. En un an on obtient alors 30 Go.
- On peut ne conserver le niveau de détail que pour les trois dernières années et ne garder que des valeurs agrégées (mois par mois, ...) pour les années précédentes.

### 5.2.2 Faits semi-additifs

- On peut cumuler les quantités stockées sur des produits et des magasins différents. Cependant, dans le cas d'une vente, un produit vendu n'est pas revendu (si on ne tient pas compte des retours contre remboursement). Le nombre de ventes est donc additif dans toutes les dimensions. Dans le cas des stocks, les produits sont reconduits en partie de jour en jour. On ne peut donc pas les cumuler sur des dates (si on a 100 produits d'un type un jour, 50 le jour suivant, 50 le jour d'après et 100 les deux jours qui suivent, on n'a pas 400 produits de ce type au bout des cinq jours!).
- On peut en revanche calculer une moyenne sur plusieurs jours. Cependant il faut éviter le piège de la fonction calcul de la moyenne AVG de SQL, qui ne s'applique pas aux dates. En effet, pour une requête qui cherche à donner la moyenne d'inventaire pour une classe de trois produits présents dans quatre magasins sur sept jours (moyenne sur une semaine d'une marque dans une région, par exemple), la fonction AVG divisera la somme des inventaires par 84 (3 produits × 4 magasins × 7 dates) alors que le résultat correct est obtenu en divisant l'inventaire total par 7, qui est le nombre de jours sur la période de mesure. Il est donc nécessaire d'isoler la contrainte date du calcul de l'inventaire, puis de calculer sa cardinalité pour effectuer la division à la fin.

### 5.3 Faits de stock améliorés

- Si on ajoute à chaque ligne la quantité vendue (ou la quantité prélevée ou expédiée d'un lieu de stockage), on peut calculer la rotation journalière et le nombre de jours d'approvisionnement.
- rotation = quantité vendue / quantité en stock (sur une journée et, par extension sur une année : quantité totale vendue / moyenne annuelle).
- nbr jours appro = quantité finale de stock / quantité moyenne vendue sur une période.
- profit brut = (prix de vente - prix coûtant) × quantité livrée
- marge brute = bénéfice brut / le dernier prix de vente

- retour de marge brute sur stock (RMBSS) = rotation × marge brute  

$$= \frac{(quantité totale livrée) \times [(valeur au dernier prix de vente) - (valeur à prix coutant)]}{(quantité moyenne journalière en stock) \times (valeur au dernier prix de vente)}$$
- un ratio élevé indique que le produit circule rapidement : il y a beaucoup de tours, et on gagne beaucoup d'argent (marge brute élevée)
- un ratio faible indique que le produit circule lentement : il y a peu de tours, et on gagne peu d'argent (marge brute faible)
- on doit inclure dans la table de faits, la quantité livrée, la valeur au coût et la valeur au dernier prix de vente.
- les faits ajoutés sont additifs dans toutes les dimensions et sont au grain retenu initialement.
- le RMBSS n'est pas dans la table de fait, il peut être calculé sur les périodes voulues au moment nécessaire.



FIGURE 5.14 – Instantané de stock étendu pour supporter l'analyse de retour de marge brute sur stock

## 5.4 Transaction de stock

On enregistre chaque transaction affectant le stock.

- réception du produit
- mise du produit à l'inspection
- sortie d'inspection
- retour au fournisseur pour défaut à l'inspection
- placement du produit dans le casier de stock
- autorisation du produit à la vente
- prélèvement du produit dans le casier de stockage
- emballage du produit pour expédition
- envoi du produit à un client
- réception du produit retourné par le client
- remise d'un produit en stock suite à un retour client
- suppression d'un produit du stock

Chaque transaction de stock identifie la date, le produit, l'entrepôt, le fournisseur, le type de transaction et un montant indiquant l'incidence de la transaction sur le stock.

On peut répondre à des questions telles que :

- combien de fois avons nous mis un produit dans un casier de stockage un jour donné et prélevé le produit dans le même casier le même jour ?
- combien de livraisons distinctes avons nous reçues d'un fournisseur donné et quand les avons-nous reçues ?
- sur quels produits avons nous eu des rejets à l'inspection répétés, provoquant des retours au fournisseur ?

Ce modèle pratique pour ces questions ne l'est pas pour celles plus courantes relatives à la gestion des stocks.

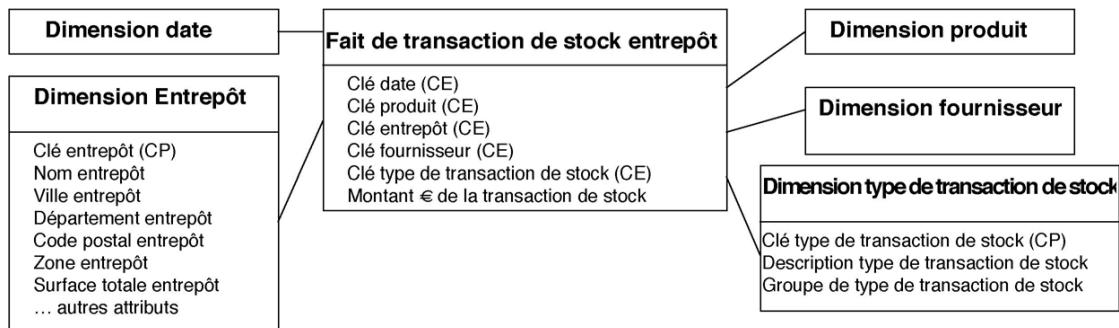


FIGURE 5.15 – Modèle de transaction de stock d'entrepôt

## 5.5 Instantané récapitulatif de stock

Dans ce modèle la table de faits comporte une ligne par expédition de produit parvenu à l'entrepôt. Dans cette même ligne on suit l'évolution du contenu de l'expédition jusqu'à ce qu'il ait quitté l'entrepôt. Cela demande à avoir une identification précise de chaque produit au moyen des numéro de série, numéro d'emballage, ..., qui permettent de le distinguer d'un produit identique arrivé plus tard.

On va donc noter les différentes étapes de passage à la réception, inspection, mise en casier, autorisation de vente, prélèvement, emballage etc. de façon à avoir un état permanent à jour de la situation des produits dans l'entrepôt.

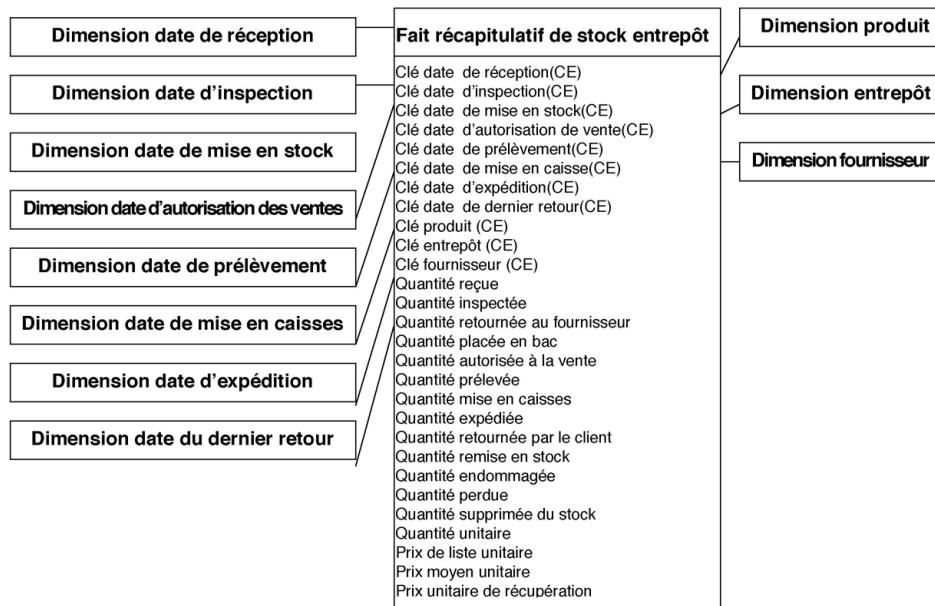


FIGURE 5.16 – Modèle récapitulatif de stock d'entrepôt

## 6. Architecture de bus de l'entrepôt de données

L'objectif est de faire apparaître clairement d'une part les processus qui seront étudiés, en commençant par des processus simples, et d'autre part les dimensions qui seront en regard (figure 5.17).

En ce qui concerne les dimensions, c'est le moyen de les définir de telle sorte qu'elles soient valides (conformes) pour tous les contextes qui seront définis. Soit se sont exactement les mêmes, soit se sont des

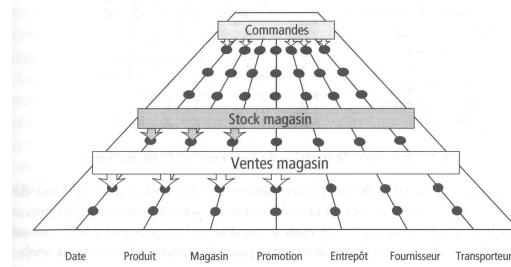


FIGURE 5.17 – Les dimensions communes sur toute la chaîne de valeur

sous ensembles (au niveau granulaire ou au niveau des composants) (figure 5.18).

| PROCESSUS D'ENTREPRISE       | DIMENSIONS COMMUNES |         |         |           |          |             |         |              |
|------------------------------|---------------------|---------|---------|-----------|----------|-------------|---------|--------------|
|                              | Date                | Produit | Magasin | Promotion | Entrepôt | Fournisseur | Contrat | Transporteur |
| Ventes au détail             | X                   | X       | X       | X         |          |             |         |              |
| Stock vente détail           | X                   | X       | X       |           |          |             |         |              |
| Livraisons pour vente détail | X                   | X       | X       |           |          |             |         |              |
| Stock entrepôt               | X                   | X       |         |           | X        | X           |         |              |
| Livraisons entrepôt          | X                   | X       |         |           | X        | X           |         |              |
| Commandes                    | X                   | X       |         |           | X        | X           | X       | X            |

FIGURE 5.18 – Les lignes de la matrice de bus correspondent à des marchés d'information

## Annexe A

### Références

1. Le Projet Décisionnel, Jean-Marie Gouarné, Eyrolles Edt. 1998
2. Entrepôts de Données, Ralph Kimball et Margy Ross, Vuibert Edt., 2003
3. Mastering Data Warehouse Design : Relational and Dimensional Techniques, Claudia Imhoff, Nicholas Galembo, Jonathan G. Geiger, Wiley Publishing, Inc. 2003
4. Data Warehousing Fundamentals for IT Professionals, Paulraj Ponniah, Wiley Publishing, Inc. 2010
5. Data Mining Concepts and Techniques, Jiawei Han, Micheline Kamber, Jian Pei, – 3rd ed. 2011
6. [http://en.wikipedia.org/wiki/Slowly\\_changing\\_dimension](http://en.wikipedia.org/wiki/Slowly_changing_dimension)
7. <http://grim.developpez.com/articles/concepts/slow-changing-dimension/>
8. Micropole, supports de cours