Systèmes décisionnels et entrepôts de données

Table des matières

Concepts de base du décisionnel

I

Décision

« Entre les deux issues mutuellement exclusives d'une alternative, l'individu X qui choisit une issue à la suite d'un processus mental, appelé réflexion, aurait pu tout aussi bien choisir l'autre »

Nous appelons ce choix : " décision " ou plus précisément " prise de décision ".

En tant qu'être humain nous sommes tout le temps confrontés à des situations qui nécessitent une prise de décision de notre part, l'importance et les conséquences de celles-ci peuvent varier, leur effet peut être temporaire, à court ou à long terme, en revanche la manière dont la décision est prise ne doit pas changer.

En effet, savoir prendre une décision repose d'abord sur une bonne connaissance de l'environnement décisionnel, pour cela il suffit de prendre en compte toutes les données relatives à la décision et que nous jugeons importantes, car il n'est pas forcement nécessaire de s'encombrer avec des données non pertinentes, redondantes et qui ne feraient que retarder le processus de prise de décision. Après avoir rassembler toutes les données nécessaires, une analyse est de rigueur afin de choisir la décision adéquate tout en évaluant les conséquences que celle ci pourrait engendrer dans le futur.

Les Phases de la Prise de décision

On distingue quatre phases dans le processus de décision :

- l'information ou le renseignement le décideur se met en quête d'informations relatives aux questions qui le préoccupent. :
  - Quelles sont les solutions possibles
  - Que font les concurrents
  - Quelle est la pratique dans les entreprises qui ont un métier voisin ?... Comment se segmente la clientèle ?
- la conception
  - Dans un deuxième temps, le décideur construit des solutions, imagine des scénarios, ce qui peut l'amener à rechercher de l'information supplémentaire.
- le choix
  - Ensuite, le décideur choisit entre les différentes actions qu'il a été capable de construire et d'identifier pendant la phase de conception.
- l'évaluation des choix précédents.
  - Après le choix, et dans la mesure où la décision s'intègre dans un processus dynamique ; Une rétroaction (feed-back) intelligente permet de corriger bien des erreurs et, sur le déroulement d'un processus décisionnel.

Le décideur

Dans l'entreprise, le décideur peut-être le responsable de cette entreprise ou le responsable d'une fonction de cette entreprise. Nous associons le terme « Décideur

» et la responsabilité vis à vis de la pérennité de l'entreprise.

Définition : Le système d'information décisionnel

Le système d'information décisionnel est un système qui contient des outils et des informations qui sont mises à la disposition des décideurs pour la prise de décision.

Les systèmes décisionnels permettent de :

- Répondre aux besoins d'aide à la prise de décision.
- Compléter les systèmes opérationnels.
- Accéder simplement et intuitivement aux données pertinentes.
- Mettre en forme des résultats.

Pourquoi un SID?

- Environnement des entreprises difficile à appréhender
  - Mondialisation
  - Concurrence exacerbée (prix, délai, qualité, rapidité, service, personnalisation...)
- Pilotage d'entreprise : prises de décision
  - Complexes : augmentation du nombre de paramètres à prendre en compte
  - Rapides pour être réactif à l'évolution de la concurrence et de la demande.
- Évolution des systèmes d'information
  - Initialement dévolu à la production (données opérationnelles)
  - Exploiter et transformer des données en informations afin de faciliter et améliorer les prises de décision

Exemple

- Fidélisation des clients
- Gestion des ressources humaines
- Bonne connaissance du marché

Pour qui

- Décideurs de différents niveaux
  - Commerciaux, Responsables du service commercial , DRH (Gestion des ressources humaines)
- Caractéristiques des décideurs
  - Non informaticiens
  - Besoins de données consolidées, synthétiques, claires

Comment

BI (Business Intelligence) (ou informatique décsionnelle) : applications capables de transformer les données opérationnelles en informations pour la prise de décision

Outils dédiés à la prise de décision

- Extraction, transformation et chargement des données sources
- Stockage éventuel et traitement des données décisionnelles
- Restitution des données sous une forme adaptée aux décideurs

Données opérationnelles	Données décisionnelles
Orientées application, détaillées, précises au moment de l'accès	Orientée activité (thème, sujet), condensées, représentes des données historiques
Mise à jour interactive possible de la part des utilisateurs	Pas de mise à jour interactive de la part des utilisateurs
Accédées de façon unitaire par une personne à la fois	Utilisées par l'ensemble des analystes, gérées par sous-ensemble
Cohérence atomique	Cohérence globale
Haute disponibilité en continu	Exigence différente, haute disponibilité ponctuelle
Uniques (pas de redondance en théorie)	Peuvent être redondantes
Structure statique, contenu variable	Structure flexible
Petite quantité de données utilisées par un traitement	Grande quantité de données utilisée par les traitements
Réalisation des opérations au jour le jour	Cycle de vie différent
Forte probabilité d'accès	Faible probabilité d'accès
Utilisées de façon répétitive	Utilisée de façon aléatoire

Comparatif données opérationnelles / décisionnelles

Fonctions d'un SID

Un système d'information décisionnel (SID) assure quatre fonctions fondamentales, à savoir la collecte, l'intégration, la diffusion et la présentation des données. À ces quatre fonctions s'ajoute une fonction de contrôle du SID lui-même, l'administration

COLLECTE

La collecte (parfois appelée datapumping) est l'ensemble des tâches consistant à détecter, à sélectionner, à extraire et à filtrer les données brutes issues des environnements pertinents compte tenu du périmètre du SID. Les sources de données internes et/ou externes étant souvent hétérogènes tant sur le plan technique que sur le plan sémantique, cette fonction est la plus délicate à mettre en place dans un système décisionnel complexe.

Elle s'appuie notamment sur des outils d'ETL (extract-transform-load pour extraction-transformation-chargement). Cette alimentation utilise les données sources issues des systèmes transactionnels de production, le plus souvent sous forme de :

- compte-rendu d'événement ou compte-rendu d'opération : c'est le constat au fil du temps des opérations (achats, ventes, écritures comptables, ...), le film de l'activité de l'entreprise
- compte-rendu d'inventaire ou compte-rendu de stock : c'est l'image photo prise à un instant donné (à une fin de période : mois, trimestre, ...) de l'ensemble du stock (les clients, les contrats, les commandes, les encours, ...).

La fonction de collecte joue également, au besoin, un rôle de recodage. Une donnée représentée différemment d'une source à une autre impose le choix d'une représentation unique pour les futures analyses.

INTEGRATION

L'intégration consiste à concentrer les données collectées dans un espace unifié, dont le socle informatique essentiel est l'entrepôt de données. Élément central du dispositif, il permet aux applications décisionnelles de bénéficier d'une source d'information commune, homogène, normalisée et fiable, susceptible de masquer la diversité de l'origine des données.

Au passage les données sont épurées ou transformées par :

- un filtrage et une validation des données en vue du maintien de la cohérence d'ensemble (les valeurs acceptées par les filtres de la fonction de

collecte mais susceptibles d'introduire des incohérences de référentiel par rapport aux autres données doivent être soit rejetées, soit intégrées avec un statut spécial)

- une synchronisation (s'il y a nécessité d'intégrer en même temps ou à la même « date de valeur » des événements reçus ou constatés de manière décalée ou déphasée)
- une certification (pour rapprocher les données de l'entrepôt des autres systèmes « légaux » de l'entreprise comme la comptabilité ou les déclarations réglementaires).

C'est également dans cette fonction que sont effectués éventuellement les calculs et les agrégations (cumuls) communs à l'ensemble du projet.

La fonction d'intégration est généralement assurée par la gestion de métadonnées, qui assurent l'interopérabilité entre toutes les ressources informatiques, que ce soit des données structurées (bases de données accédées par des progiciels ou applications), ou des données non structurées (documents et autres ressources non structurées, manipulés par les systèmes de gestion de contenu).

DIFFUSION (OU DISTRIBUTION)

La diffusion met les données à la disposition des utilisateurs, selon des schémas correspondant au profil ou au métier de chacun, sachant que l'accès direct à l'entrepôt de données ne correspondrait généralement pas aux besoins d'un décideur ou d'un analyste. L'objectif prioritaire est de segmenter les données en contextes informationnels fortement cohérents, simples à utiliser et correspondant à une activité décisionnelle particulière.

Alors qu'un entrepôt de données peut héberger des centaines ou des milliers de variables ou indicateurs, un contexte de diffusion raisonnable n'en présente que quelques dizaines au maximum. Chaque contexte peut correspondre à un datamart, bien qu'il n'y ait pas de règles générales concernant le stockage physique.

Très souvent, un contexte de diffusion est multidimensionnel, c'est-à-dire modélisable sous la forme d'un hypercube; il peut alors être mis à disposition à l'aide d'un outil OLAP. Les différents contextes d'un même système décisionnel n'ont pas tous besoin du même niveau de détail. De nombreux agrégats ou cumuls, n'intéressant que certaines applications et n'ayant donc pas lieu d'être gérés en tant qu'agrégats communs par la fonction d'intégration, relèvent donc de la diffusion. Ces agrégats peuvent être, au choix, stockés de manière persistante ou calculés dynamiquement à la demande.

On peut distinguer trois questions à élucider pour concevoir un système de reporting :

- A qui s'adresse le rapport spécialisé ? (choix des indicateurs à présenter, choix de la mise en page)
- Par quel trajet ? (circuit de diffusion type "workflow" pour les personnes, circuits de transmission "télécoms" pour les moyens)
- Selon quel agenda ? (diffusion routinière ou sur événement prédéfini)

PRESENTATION

Cette quatrième fonction, la plus visible pour l'utilisateur, régit les conditions d'accès de l'utilisateur aux informations. Elle assure le fonctionnement du poste de travail, le contrôle d'accès, la prise en charge des requêtes, la visualisation des résultats sous une forme ou une autre. Elle utilise toutes les techniques de communication possibles (outils bureautiques, requêteurs et générateurs d'états spécialisés, infrastructure web, télécommunications mobiles, etc).

Modèles de SID

II

A. 1ère génération

Infocentre

L'infocentre sert à prendre des décisions opérationnelles basées sur des valeurs courantes

Définition

L'infocentre est une collection de données orientées sujet, intégrées, volatiles, actuelles, organisées pour le support d'un processus de décision ponctuel.

- Extrait direct des sources
- Adapté à une classe de décideurs
- Organisé selon un modèle informatique adapté aux outils décisionnels

Aide à la décision limitée

- Types de décisions
  - Décisions Opérationnelles : utilisation optimale des ressources allouées dans le gestion courante de l'entreprise (gestion des stocks, gestion de la production....) ; très court terme; chef d'équipe ou atelier
  - Décisions Tactiques : définir comment les ressources de l'entreprise doivent être utilisées pour parvenir à réaliser les décisions stratégiques ; court terme ; direction fonctionnelle
- Pas d'historisation des données
- Pas de centralisation de données décisionnelles communes

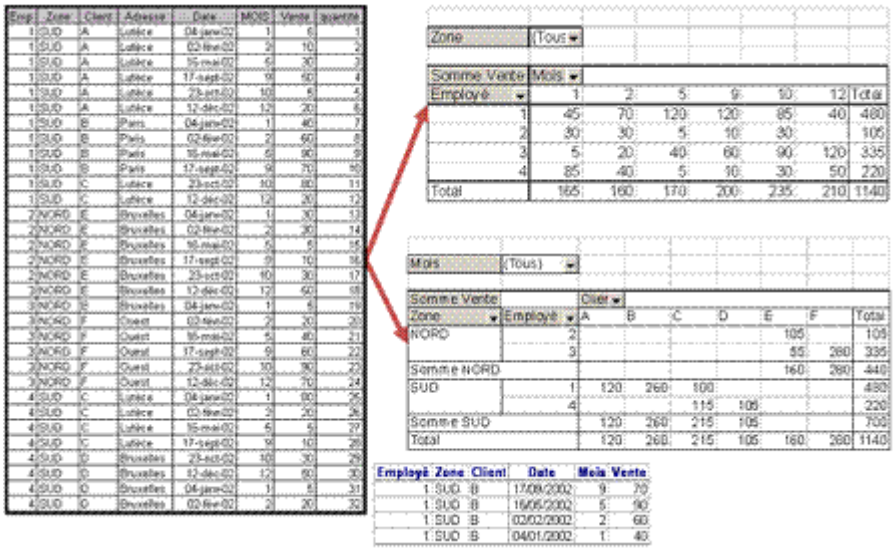
Tableur

Tableur : Outil d'aide à la décision

- Meilleur outil pour la restitution des données décisionnelles Composants
- Des fonctions de base dévolues à un tableur (calcul, graphique...)
- Des fonctions spécifiques pour l'aide à la décision (Tableaux croisés dynamiques, valeur cible, solveur)

Limites

- Un seul tableau intégrant toutes les données sources pour les traitements
- Analyses limitées



Tableur

Bases de données

- Cas 1 : Interrogation directe des données sources
  - Étude de la structure des BD sources
  - Interrogations "décisionnelles" avec le langage SQL
- Cas 2 : Spécification d'une BD extraite des sources
  - Conception de BD: E/A, Relationnel, modèle physique
  - Écriture des scripts d'alimentation de la BD
  - Interrogation langage assertionnel graphique ou textuel

Limites :

- Connaissances BD et langage d'interrogation
- Tables résultat "unidimensionnelle"

Bilan

- Prises de décisions limitées
  - Décisions opérationnelles ou tactiques
  - Pas d'historisation
  - Pas de centralisation et de croisement de données
- Est-ce un système décisionnel à part entière?
  - Pas de dissociation systèmes de production et décisionnel
  - Uniquement centré sur les outils de restitution
- Autres générations : les systèmes décisionnels

B. 2ème génération : Entrepôt de données / Magasin de données

Le concept de Data Warehouse a été formalisé pour la première fois en 1990

- Entrepôt de données ("data warehouse")
  - Lieu de stockage centralisé d'un
  - extrait des sources
  - pertinent pour les décideurs,
  - historisé
  - organisé selon un modèle informatique facilitant la gestion des données.

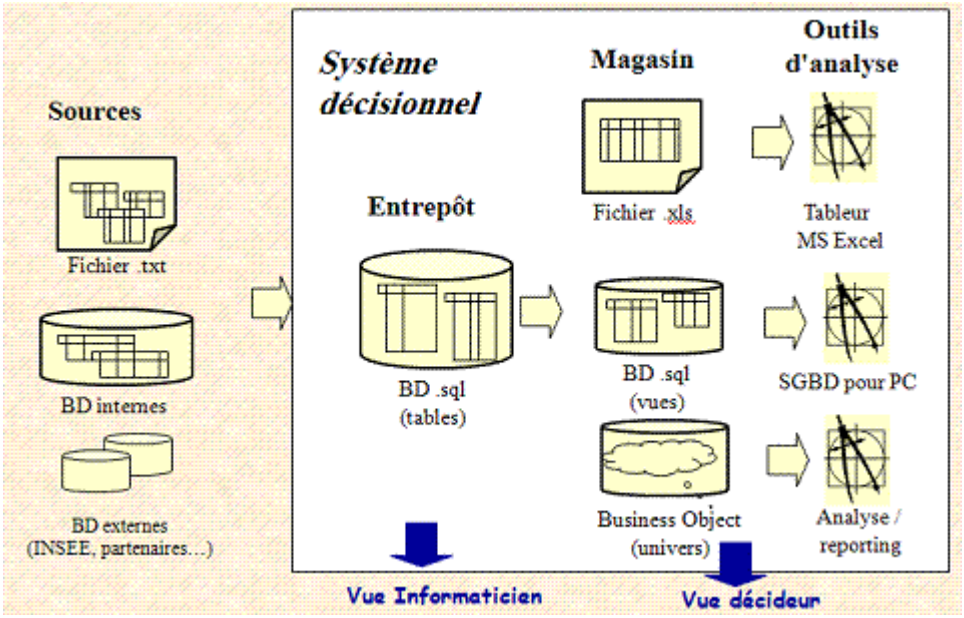
- Magasin de données ("data mart")
  - extrait de l'entrepôt
  - adapté à une classe de décideurs (ou à un usage particulier) et
  - organisé selon un modèle approprié aux outils d'analyse
  - Prise de décisions tactiques et stratégiques

Définition

"A Data Warehouse is a subject-oriented, integrated, time-variant, non-volatile, summarized collection of data in support of management decisions", Bill Inmon, "Building the Data Warehouse » , 1994

Un entrepôt de données se définit comme une collection de données

- orientées sujet,
- intégrées,
- historisées,
- non volatiles,
- résumées et
- disponibles pour l'interrogation et l'analyse"



2ème Génération (ED/MD)

Définition : ☐ Magasin de données

C'est un extrait direct de l'entrepôt de données

- Adapté à une classe de décideurs
- Organisé selon un modèle informatique
- adapté aux outils décisionnels

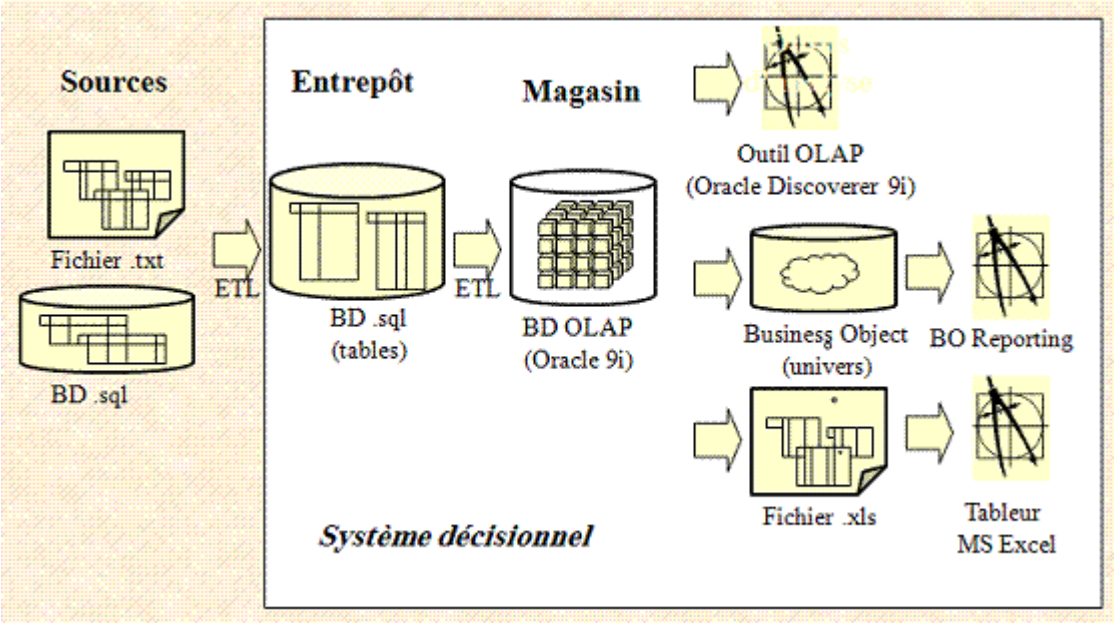
Parmi les outils associés aux magasins de données nous citons :

- Tableur : tableau uni-dimension avec déclinaison en tableaux croisés dynamiques
- Bases de données : conception et implantation d'une BD avec interrogation décisionnelle (sous-requêtes, groupements...)
- Requêteur graphique dédié à la prise de décision

C. 3ème génération OLAP

Les outils OLAP (On Line Analytical Process) reposent sur une base de données multidimensionnelle, destinée à exploiter rapidement les dimensions d'une population de données. La plupart des solutions OLAP reposent sur un même principe : restructurer et stocker dans un format multidimensionnel les données issues de fichiers plats ou de bases relationnelles. Ce format multidimensionnel, connu également sous le nom d'hypercube, organise les données le long de dimensions. Ainsi, les utilisateurs analysent les données suivant les axes propres à leur métier.

Ce type d'analyse multidimensionnelle nécessite à la fois l'accès à un grand volume de données et des moyens adaptés pour les analyser selon différents points de vue. Ceci inclut la capacité à discerner des relations nouvelles ou non prévues entre les variables, la capacité à identifier les paramètres nécessaires à manier un volume important de données pour créer un nombre illimité de dimensions et pour spécifier des expressions et conditions inter dimensions. Ces dimensions représentent les chemins de consolidation



3ème Génération OLAP

LES 12 REGLES OLAP

Afin de formaliser le concept OLAP, fin 1993, à la demande de Arbor Software, Edgar F. Codd publie un article intitulé "Providing OLAP to User Analysts" aux Etats Unis, dans lequel il définit 12 règles que tout système de pilotage multidimensionnel devrait respecter.. "Ce qu'il y a d'agréable avec ces outils OLAP", explique Eric Klusman, de Cantor Fitzgerald LP, "c'est que je suis en mesure de distribuer les données aux utilisateurs sans les obliger à apprendre des complexes formules de programmation, d'interrogation ou même à ce qu'ils aient à programmer leurs tableurs". D'une façon générale, tous affirment que l'on peut interfacer de nombreux outils d'utilisateurs avec des bases de données multidimensionnelles sans qu'il soit nécessaire de consentir de lourds efforts de formation ou des interventions importantes du service informatique.

Règles concernant l'architecture du système OLAP

1. Accessibilité à de nombreuses sources de données : Le système OLAP doit donner accès aux données nécessaires aux analyses demandées. Les outils

OLAP doivent avoir leur propre schéma logique de stockage des données physiques hétérogènes, doivent accéder aux données et réaliser n'importe quelle conversion afin de présenter à l'utilisateur une vue simple et cohérente. Ils doivent aussi savoir de quel type de systèmes proviennent les données

2. Support multi-utilisateurs : Les outils OLAP doivent supporter les accès concurrents, garantir l'intégrité et la sécurité afin que plusieurs utilisateurs accèdent au même modèle d'analyse
3. Architecture client/serveur : La plupart des données pour OLAP sont stockées sur des gros systèmes et sont accessibles via des PC. Il est donc nécessaire que les produits OLAP soient capables de travailler dans un environnement Client/Serveur.
4. Transparence du serveur OLAP à différents types de logiciel : Cette transparence se traduit pour l'utilisateur par un complément à ses outils habituels garantissant ainsi sa productivité et sa compétence. Elle s'appuie sur une architecture ouverte permettant à l'utilisateur d'implanter le système OLAP sans affecter les fonctionnalités du système central. Par ailleurs, l'utilisateur ne doit pas être concerné par l'intégration des données dans OLAP provenant d'un environnement homogène ou hétérogène.

Règles concernant les données du système OLAP

1. Vue multi-dimensionnelle : L'utilisateur a l'habitude de raisonner en vue multidimensionnelle comme par exemple lorsqu'il souhaite analyser les ventes par produit mais aussi par région ou par période. Ces modèles permettent des manipulations simples : rotation, pivot ou vues par tranche, analyse de type permutations d'axes (slice and dice) ou en cascade (drill anywhere).
2. Dimensions génériques (principe de hiérarchisation) : Toutes les dimensions doivent être équivalentes en structure et en calcul. Il ne doit exister qu'une seule structure logique pour toutes les dimensions. Toute fonction qui s'applique à une dimension doit être aussi capable de s'appliquer à une autre dimension.
3. Nombre illimité de dimension et de niveaux d'agrégation : Tout outil OLAP doit gérer au moins 15 à 20 dimensions
4. Gestion dynamique des matrices creuses : Le schéma physique des outils OLAP doit s'adapter entièrement au modèle d'analyse spécifique créé pour optimiser la gestion des matrices creuses. En effet, dans une analyse à la fois sur les produits et les régions, tous les produits ne sont pas vendus dans toutes les régions

Règles concernant la manipulation de données des systèmes OLAP

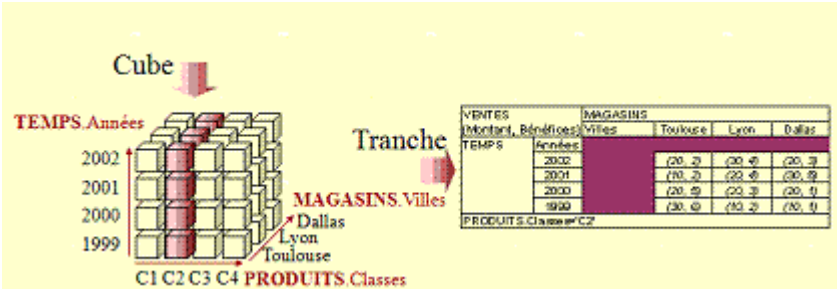
1. Manipulation intuitive des données : Toute manipulation doit être accomplie via une action directe sur les cellules du modèle sans utiliser de menus ou des chemins multiples à travers l'interface utilisateur
2. Performance du système de reporting (restitution) : L'augmentation du nombre de dimensions ou du volume de la base de données ne doit pas entraîner de dégradation visible par l'utilisateur.
3. Souplesse et facilité de construction des rapports : La création des rapports dans les outils OLAP doit permettre aux utilisateurs de présenter comme ils le désirent des données synthétiques ou des résultats en fonction de l'orientation du modèle
4. Calcul au travers des dimensions : Les opérations doivent pouvoir s'effectuer sur toutes les dimensions et ne doivent pas faire intervenir l'utilisateur pour définir un calcul hiérarchique

Remarque

D'après EF CODD & Associates, les SGBD Relationnels n'ont jamais été conçus pour fournir les puissantes fonctions de synthèse, d'analyse et de consolidation communément appelées analyse multidimensionnelle des données. Ces types de fonctions ont toujours été prévus pour être fournis par des outils séparés, orientés utilisateurs et complémentaires des SGBD Relationnels. Les tables vont être transformées en un hypercube de données. Les données vont pouvoir être visualisées sous différents angles grâce aux vues multidimensionnelles.

Modélisation multidimensionnelle

- Modélisation facilitant les prises de décisions
- Représentation des données dans un espace multidimensionnel
  - Modélisation centrée sujet d'analyse contenant des indicateurs
  - Analyse en fonction d 'axes d 'analyse



Cube de données

	OLTP On-Line Transactional Processing	OLAP On-Line Analytical Processing
Données	Exhaustives Courantes Dynamiques Orientée applications	Résumées Historiques Statiques Orientées sujets
Utilisateurs	Nombreux Variés Mises à jour et interrogations	Peu nombreux Uniquement les décideurs Interrogations

Comparatif OLAP/OLTP

Entrepôts de données

III

Définition

Un entrepôt de données est une collection de données orientées sujet, intégrées, variables dans le temps et non volatiles, en soutien au processus de prise de décisions de gestion.

Orientées sujet

Car l'entrepôt est structuré autour des principaux sujets de l'entreprise (tels que les clients, les produits et les ventes) au lieu des principaux domaines d'application (comme la facturation client, la gestion des stocks et la vente des produits). Ceci reflète le besoin de disposer de données de support à la décision et non plus simplement de données orientées application.

Intégrées,

Du fait du rassemblement de sources de données provenant de différents systèmes applicatifs à l'échelle de l'entreprise. Les données sources sont souvent incohérentes, puisqu'elles arborent des formats différents. Les sources de données intégrées doivent subir une mise en cohérence pour présenter une vue unifiée des données aux utilisateurs.

Variables dans le temps,

Parce que les données de l'entrepôt ne sont précises et valables qu'à un certain moment ou pendant un intervalle de temps donné. La variation temporelle de l'entrepôt de données est aussi montrée par la période étendue pendant laquelle les données sont conservées, par l'association implicite ou explicite du temps à toutes les données et par le fait que les données représentent une suite d'instantanés.

Non volatiles

Puisque les données ne sont pas mises à jour en temps réel, mais rafraîchies régulièrement à partir de systèmes opérationnels. Les nouvelles données viennent toujours s'ajouter en supplément à la base de données et non en remplacement. La base de données absorbe continuellement ces nouvelles données en les intégrant de manière incrémentielle (c'est-à-dire qu'il n'y a pas de modification mais bien des ajouts) aux données précédentes.

Exemple : Voici quelques exemples de la panoplie de requêtes auxquelles l'entrepôt de données d'une agence immobilière doit être capable de répondre :

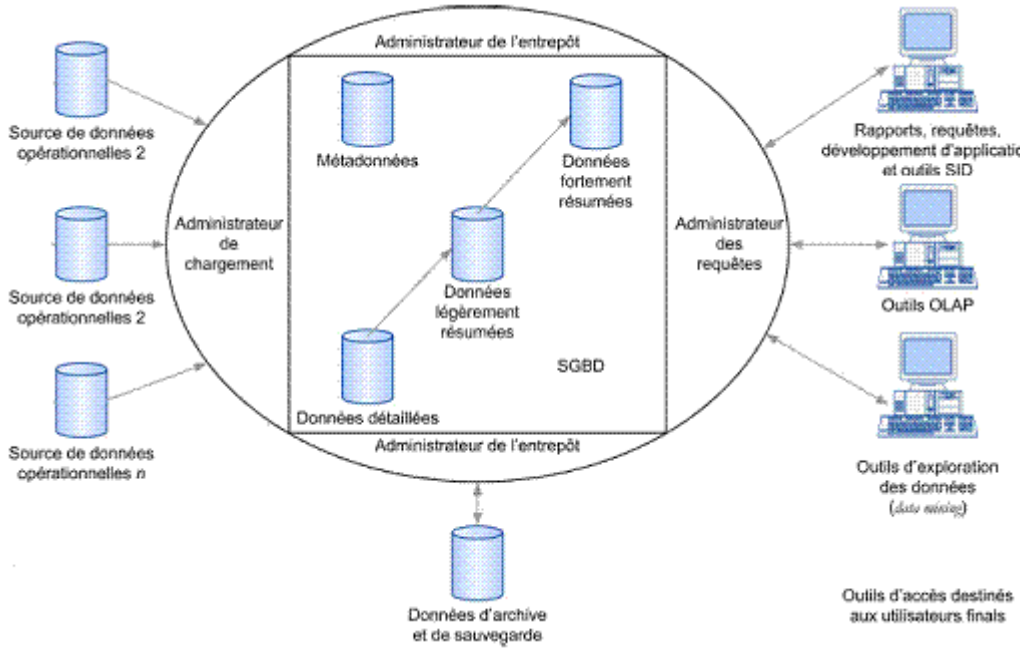
- Quel est le revenu total pour la wilaya d'Alger au troisième trimestre 2009?
- Quel est le résultat total des ventes de propriétés pour chaque type d'immeuble en Algérie en 2008?
- Quelles sont les trois zones les plus appréciées de chaque ville concernant la location de logements en 2008 et comment ces résultats se comparent-ils

par rapport aux deux années précédentes?

- Quel serait l'effet sur les ventes de propriétés dans les différentes régions de l'Algérie si les prix légaux augmentaient de 3,5 % et les taxes régionales diminuaient de 1,5 % pour les propriétés de plus de 1 00 000 000 DA?

A. Architecture d'un entrepôt de données

La figure ci dessous résume l'architecture d'un entrepôt de données :



Architecture ED

Données opérationnelles

La source des données de l'entrepôt de données est fournie par :

- Des données opérationnelles d'un gros ordinateur central, contenues dans des bases de données de première génération, de type hiérarchique ou en réseau. Des estimations ont montré que la majorité des données opérationnelles d'entreprise est encore contenue dans de tels systèmes.
- Des données départementales contenues dans des systèmes de fichiers propriétaires, tels que VSAM, RMS, et des SGBDR tels que Informix et Oracle.
- Des données privées détenues dans des stations de travail et des serveurs privés.
- Des systèmes externes, tels que l'Internet, des bases de données commerciales disponibles ou des bases de données détenues par les fournisseurs ou les clients de l'organisation.

Administrateur de chargement

L'administrateur de chargement (load manager), également dénommé composant frontal ou de frontend, effectue toutes les opérations associées à l'extraction et au chargement des données dans l'entrepôt.

Les données peuvent être extraites directement des sources de données. Les opérations effectuées par l'administrateur de chargement sont éventuellement des

transformations simples des données pour les préparer à entrer dans l'entrepôt.

La taille et la complexité de ce composant varient selon les entrepôts de données et sa construction s'effectue à l'aide d'une combinaison d'outils d'exportation et de chargement de données .

Administrateur de l'entrepôt

L'administrateur de l'entrepôt (warehouse manager) remplit toutes les tâches associées à la gestion des données dans l'entrepôt. Ce composant est édifié à l'aide d'outils d'administration des données des vendeurs et de programmes sur mesure.

Les tâches de l'administrateur de l'entrepôt :

- L'analyse des données pour en garantir la cohérence;
- La transformation et la fusion de données sources d'une zone de stockage temporaire vers des tables de l'entrepôt;
- La création d'index et de vues sur les tables de base;
- La génération de la dénormalisation si nécessaire;
- La génération d'agrégats si nécessaire;
- La sauvegarde et l'archivage des données.

Administrateur des requêtes

L'administrateur des requêtes (query manager, aussi nommé le composant de backend) se charge des opérations de gestion des requêtes des utilisateurs.

Ce composant est normalement construit à l'aide d'outils d'accès aux données par les utilisateurs, vendus par les éditeurs de logiciels, d'utilitaires de base de données et de programmes réalisés sur mesure.

La complexité de l'administrateur des requêtes est déterminée par les facilités que procurent les outils d'accès aux données par les utilisateurs finaux et la base de données. Les opérations prises en charge par ce composant sont notamment la déviation des requêtes vers les tables adéquates et la planification de l'exécution des requêtes.

Données détaillées

Cette zone de l'entrepôt stocke toutes les données détaillées dans le schéma de base de données. Dans la plupart des cas, les données détaillées ne sont pas stockées en ligne mais sont mises à disposition par l'agrégation des données jusqu'au niveau de détail suivant.

Cependant, d'une manière régulière, les données détaillées sont ajoutées à l'entrepôt en supplément des agrégats de données.

Données légèrement ou fortement résumées

Cette zone de l'entrepôt stocke toutes les données légèrement ou fortement résumées au préalable (agrégées) générées par l'administrateur de l'entrepôt. La raison d'être des informations de synthèse est d'augmenter les performances des requêtes. Même si la synthèse des données implique une augmentation des coûts opérationnels, ils sont contrebalancés par la suppression de la nécessité d'effectuer continuellement des opérations de synthèse (telles que des tris ou des regroupements) lors de la réponse à des requêtes des utilisateurs. Les données résumées sont actualisées en permanence au fur et à mesure du chargement de nouvelles données dans l'entrepôt.

Données archivées et sauvegardées

Cette partie de l'entrepôt emmagasine les données détaillées et résumées pour les besoins d'archivage et de sauvegarde. Même si les données résumées sont générées à partir des données détaillées, il est nécessaire de sauvegarder les

données de synthèse en ligne si ces données sont maintenues au-delà de la période de détention des données détaillées, auquel cas, il ne serait plus possible de les générer à nouveau en cas de problème. Les données sont transférées dans des stockages d'archivage tels que des bandes magnétiques ou des disques optiques.

Métadonnées

Cette partie de l'entrepôt conserve toutes les définitions de métadonnées (les données à propos des données) employées par tous les processus de l'entrepôt. Les métadonnées servent à une grande variété d'objectifs, dont :

- Les processus d'extraction et de chargement: Les métadonnées sont utilisées pour faire correspondre des sources de données à une vue commune des données au sein de l'entrepôt
- Le processus d'administration de l'entrepôt: Les métadonnées servent à automatiser la production des tables de synthèse;
- En tant qu'acteur dans le processus d'administration des requêtes: Les métadonnées sont utilisées pour diriger une requête vers la source de données la plus appropriée.

Outils d'accès des utilisateurs finaux

La principale raison d'être de l'entreposage de données est de livrer des informations aux utilisateurs commerciaux et administratifs pour les aider à prendre des décisions stratégiques. Ces utilisateurs interagissent avec l'entrepôt, grâce à des outils d'accès aux données spécifiques pour les utilisateurs finaux.

L'entrepôt de données doit par conséquent apporter son soutien aux analyses ad hoc (ou de circonstance) mais aussi de routine. Les hautes performances sont obtenues par une planification prédéfinie des exigences de jointure, de synthèse et d'états (rapports imprimés) par les utilisateurs finaux.

Nous nous intéressons à 3 catégories d'outils

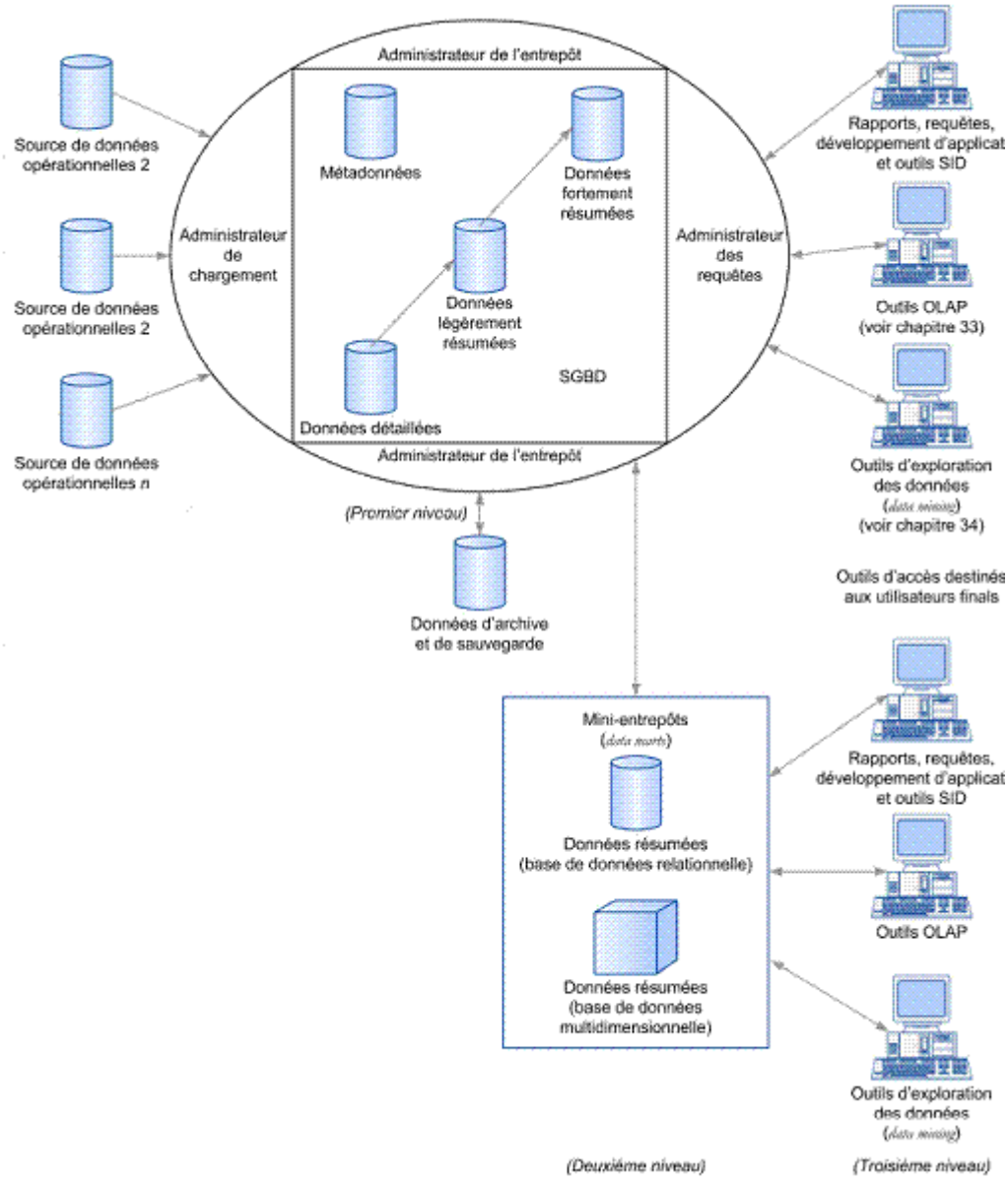
1. Les outils de rapport et de requête : comprennent les outils de génération d'états et les outils d'édition d'états. Les outils de génération d'états permettent de produire des rapports opérationnels normaux ou de supporter des lots de travaux de grand volume, tels que les commandes client, la facturation client et les feuilles de salaire. Les outils d'édition d'état sont des outils peu coûteux destinés aux utilisateurs finaux.
2. Les outils de traitement analytique en ligne (OLAP) : Les outils de traitement analytique en ligne (OLAP, Online Analytical Processing) se fondent sur le concept des bases de données multidimensionnelles et permettent à un utilisateur expérimenté et de grande compétence professionnelle d'analyser les données à l'aide de vues complexes et multidimensionnelles.
3. Les outils d'exploration des données (data mining) : c'est le processus qui consiste à découvrir des nouvelles corrélations significatives, des profils et des tendances en explorant (ou en extrayant de la mine, mining) de vastes quantités de données à l'aide de techniques statistiques, mathématiques et, même, de l'intelligence artificielle (IA).

B. Architecture ED avec magasins de données

Définition : Magasin de données (Data mart)

Un magasin de données détient un sous-ensemble des données d'un entrepôt de données, normalement sous la forme de données résumées, concernant un département ou service particulier ou encore d'une fonction professionnelle précise. Le magasin de donnée est soit autonome, soit relié à l'entrepôt de données central

de l'entreprise. À mesure de l'expansion de l'entrepôt de données, la capacité à remplir les différents besoins de l'organisation risque d'être compromise.



Architecture ED/MD

Les caractéristiques déterminantes des magasin de données

Les caractéristiques déterminantes des magasins de données par rapport aux ED sont :

- Un magasin de données se concentre sur les seules exigences des utilisateurs associés à un département ou une fonction professionnelle;
- Les magasins de données ne contiennent en principe pas de donnée opérationnelle détaillée, contrairement aux entrepôts de données;
- Comme les magasins de données contiennent moins de données, comparés aux entrepôts de données, les magasins de données sont plus aisément compris et parcourus.

C. Magasins de données indépendants

Le Data Mart est une base de données moins coûteuse que le Data Warehouse, et plus légère puisque destinée à quelques utilisateurs d'un département. Il séduit plus que le Data Warehouse les candidats au décisionnel. C'est une petite structure très ciblée et pilotée par les besoins utilisateurs. Il a la même vocation que le Data Warehouse (fournir une architecture décisionnelle), mais vise une problématique précise avec un nombre d'utilisateurs plus restreint. En général, c'est une petite base de données (SQL ou multidimensionnelle) avec quelques outils, et alimentée par un nombre assez restreint de sources de données. Son coût est nettement moins élevé que celui d'un entrepôt de donnée.

Ceci conduit à l'idée de construire des magasins de données indépendamment de l'existence d'un entrepôt central.

	Data Warehouse	Data Mart
Cible utilisateur	Toute l'entreprise	Département
Implication du service informatique	Elevée	Faible ou moyen
Base de données d'entreprise	SQL type serveur	SQL milieu de gamme, bases multidimensionnelles
Modèles de données	A l'échelle de l'entreprise	Département
Champ applicatif	Multi sujets, neutre	Quelques sujets, spécifique
Sources de données	Multiples	Quelques unes
Stockage	Base de données	Plusieurs bases distribuées
Taille	Centaine de GO et plus	Une à 2 dizaines de GO
Temps de mise en place	9 à 18 mois pour les 3 étapes	6 à 12 mois (installation en plusieurs étapes)
Coût	> 6 millions de francs	500.000 à 3 millions de francs
Matériel	Unix	NT, petit serveur Unix

Comparaison ED/MD

Construire un ou plusieurs Data Marts départementaux au lieu d'un Data Warehouse central permet de valider rapidement le concept d'informatique décisionnelle. Mais construire des Data Marts n'est pas sans risques :

- En effet, dans les entreprises, des Data Marts isolés peuvent proliférer. Ces entreprises risquent de retomber dans le piège d'une architecture composée de multiples systèmes décisionnels incohérents, contenant des informations redondantes. Cela coûte plus cher et c'est plus complexe à gérer qu'un Data Warehouse centralisé. Les entreprises américaines, plus en avance que les entreprises européennes, en ont fait les frais.
- Les Data Marts résolvent les problèmes de performance des gros Data Warehouse. Mais ils font régresser vers le vieux problème des îlots isolés. Les entreprises vont devoir affronter des problèmes techniques complexes et coûteux pour remettre en cohérence les ensembles.
- Fédérer des Data Marts ou les faire évoluer vers une structure centralisée n'est pas facile. On peut se poser la question s'il est préférable de bâtir un gros et unique Data Warehouse ou bien de concevoir un réservoir plus

modeste, nourri par les données d'un seul département.

Il est intéressant de commencer par un Data Mart, à condition de respecter certaines règles :

- Impliquer les utilisateurs.
- Ne pas construire de multiples Data Marts isolés.
- Bannir les redondances.

Attention

Donc le Data Mart peut préparer au Data Warehouse. Mais il faut penser grand, avenir, et adopter des technologies capables d'évoluer.

D. Fonctionnement d'un data warehouse

1. Processus ETC (Extraire, Transformer, Charger)

Il s'agit d'un processus d'acquisition de données qui gère les flux informationnels depuis l'acquisition de l'information des sources jusqu'à son intégration dans le data warehouse.

Ce processus comprend quatre phases :

1. Découvrir
2. Extraire
3. Transformer
4. Charger.

Découvrir

Il s'agit d'identifier les données pertinentes des sources et de qualifier leur niveau de fiabilité.

Extraire :

Une fois les données pertinentes identifiées, il faut les amener vers le data warehouse. Cela commence par une étape d'extraction qui doit prendre en considération l'hétérogénéité des bases de données sources. L'outil d'extraction doit être capable d'extraire les données de diverses sources, faute de quoi, il sera nécessaire d'écrire des extracteurs (wrapper) pour pouvoir l'alimenter.

Définition : Extracteur (wrapper)

Composant capable de traduire les requêtes et les données depuis le modèle d'une source globale vers le modèle de l'entrepôt et vice versa. Un des problèmes qui se pose, est de capturer les mises à jour locales et de générer les données nécessaires à la mise à jour de l'entrepôt. Si le SGBD gérant la base de production dispose d'un mécanisme de déclencheurs (triggers), celui-ci peut être mis à profit pour générer l'émission de la mise à jour vers l'entrepôt. Sinon, il faut être capable d'interroger périodiquement au moyen d'un logiciel spécial chaque base locale ou son journal afin de récupérer les mises à jour effectuées durant la dernière période. Un tel logiciel est appelé moniteur de source (source monitor).

Définition : Moniteur de source (source monitor) :

Composant capable de détecter les mises à jour effectuées sur la source et repérer les données à envoyer à l'entrepôt pour sa mise à niveau ultérieure

Complément

Pour extraire les données sources, il y a plusieurs technologies utilisables :

- des passerelles, fournies principalement par les éditeurs de bases de données. Ces passerelles sont généralement insuffisantes car elles sont mal adaptées aux processus de transformation complexes ;
- des utilitaires de réplication, utilisables si les systèmes de production et décisionnel sont homogènes et si la transformation à appliquer aux données est légère ;
- des outils spécifiques d'extraction. Ces outils sont certainement la solution opérationnelle au problème de l'extraction, mais leur prix relativement élevé est un frein à leur utilisation dans les premières applications.

Transformer :

Cette étape inclue la mise en correspondance des formats de données, le nettoyage, la transformation et l'agrégation, elle a pour but d'obtenir un ensemble homogène et cohérent de données.

Sachant que les données sources sont essentiellement générées par des systèmes OLTP, il est nécessaire de reconstruire ces données pour les besoins de l'entrepôt de données. La reconstruction des données implique :

- Le nettoyage des données de mauvaise qualité;
- La restructuration des données pour respecter les nouvelles exigences de l'entrepôt de données, notamment l'ajout ou la suppression de champs et la dé normalisation des données;
- La vérification de la cohérence des données sources par rapport à elles-mêmes et par rapport aux données déjà chargées dans l'entrepôt de données.

Charger :

Le chargement est la dernière phase de l'alimentation du Data Warehouse. C'est une phase délicate notamment lorsque les volumes sont importants. Pour obtenir de bonnes performances en chargement, il est impératif de maîtriser les structures du SGBD (tables et index) associées aux données chargées afin d'optimiser au mieux ces processus. Les techniques de parallélisation optimisent les chargements lourds. Pour les mettre en œuvre, des utilitaires particuliers existent chez la majorité des éditeurs de bases de données.

2. Restitution des données :

Pour pouvoir exploiter les informations contenues dans le data warehouse, l'utilisateur doit disposer d'outils de restitutions qui se chargeront d'extraire les données nécessaires. Les évolutions les plus récentes dans le domaine du data warehouse ont pour but de rendre le système le plus indépendant possible des outils de restitution, il est possible de concevoir, désormais, un model de données indépendamment des outils qui permettent de l'exploiter, voir du type d'utilisation qui en sera fait.

Trois grands types de classe existent dans le domaine de la restitution :

- La classe qui permet d'effectuer des analyses : l'utilisateur pourra effectuer des analyses sur les données de l'entrepôt afin d'améliorer une performance, et de mesurer l'impact de cette analyse sur la prise de décision.
- La classe qui permet de diffuser les informations en masse ;
- La classe qui permet un accès aux données en libre service : l'utilisateur

pourra sélectionner librement les données qu'il trouve adéquates en fonction de ses objectifs.

Le reporting

Un rapport d'entreprise doit permettre de présenter les informations de manière synthétique, l'utilisateur doit pouvoir passer au-delà de ce qui est possible avec une simple requête SQL et doit pouvoir définir et optimiser le mode d'accès aux données. L'outil de reporting doit offrir des fonctionnalités évoluées de calcul en local et de mise en page (rapport à l'allure multidimensionnelle).

Un outil de reporting doit posséder les qualités suivantes :

- Puissance de calcul et mise en forme ;
- Mécanisme de sécurité et de confidentialité ;
- Fonctionnalité de distribution des rapport.

L'accès en libre service

L'accès en libre service donne une forte autonomie à l'utilisateur : il peut accéder aux données librement à condition qu'il dispose des droits d'accès appropriés, on parle alors d'accès ad hoc ou en mode libre service.

L'outil approprié pour ce type de besoin est le requêteur.

Le datamining

Le fait de stocker simplement des informations dans un entrepôt de données n'apporte par les bénéfices qu'une organisation recherche. Pour concrétiser la valeur ajoutée d'un entrepôt de données, il est nécessaire d'extraire la connaissance enfouie au sein de celui-ci. Cependant, comme le volume et la complexité des données augmentent dans l'entrepôt de données, il devient de plus en plus difficile, voire impossible, pour les analystes financiers de dégager des tendances et des relations parmi les données à l'aide de requêtes simples et d'outils de génération de rapports et d'états. L'exploration de données constitue l'une des meilleures méthodes d'extraction de tendances et de profils significatifs, à partir d'une vaste quantité de données. L'exploration de données découvre des informations dans les entrepôts de données que les requêtes et les rapports sont incapables de révéler avec efficacité.

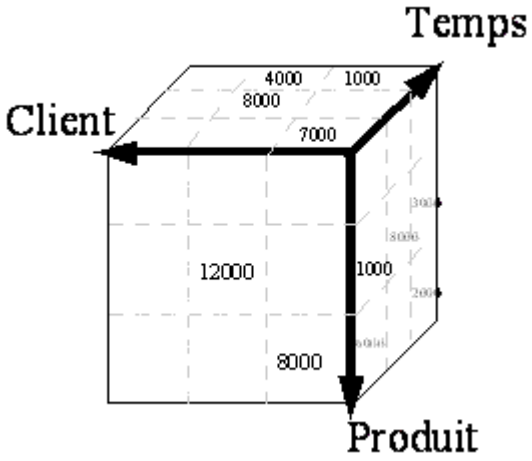
Modèle multidimensionnel

IV

Les serveurs OLAP ont été conçus pour s'intégrer dans un environnement client/serveur afin d'en retirer les possibilités offertes. Les utilisateurs disposant de postes de travail intelligents accèdent à un serveur de base de données multidimensionnelle. Celui-ci contient un hypercube prédéfini dans lequel doit être stockée la globalité des données. Ce qui nécessite de s'appuyer sur une information pré-packagée et fortement structurée.

Il permettra ainsi d'analyser la répartition d'un indicateur comme le " chiffre d'affaire" en fonction des axes ou dimensions " clients ", " produit ", " temps". En outre, des hiérarchies seront définies pour chaque axe d'analyse (par exemple, l'année, puis la saison, le mois et la semaine, pour l'axe temps).

Une fois cette structure multidimensionnelle établie, l'outil OLAP propose des méthodes de navigation dans les données, comme le "drill-down" pour aller vers les informations détaillées dans une hiérarchie, le "slice and dice" pour changer d'axe d'analyse.



Cube de données

Définition

La modélisation multidimensionnelle consiste à considérer un sujet analysé comme un point dans un espace à plusieurs dimensions. Les données sont organisées de manière à mettre en évidence le sujet analysé et les différentes perspectives de l'analyse.

A. Concepts de base

Concept de fait

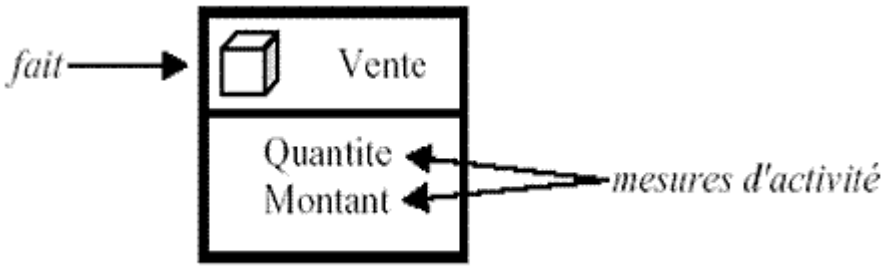
Le fait modélise le sujet de l'analyse. Un fait est formé de mesures correspondant aux informations de l'activité analysée. Les mesures d'un fait sont numériques et généralement valorisées de manière continue. L

es mesures sont numériques pour permettre de résumer un grand nombre d'enregistrements en quelques enregistrements (on peut les additionner, les dénombrer ou bien calculer le minimum, le maximum ou la moyenne).

Les mesures sont valorisées de façon continue car il est important de ne pas valoriser le fait avec des valeurs nulles. Elles sont aussi souvent additives ou semi-additives afin de pouvoir les combiner au moyen d'opérateurs arithmétiques.

Considérons le fait de Vente pouvant être constitué des mesures d'activités suivantes : quantité de produits vendus et montant total des ventes. Nous représenterons le fait par un rectangle englobant les différentes mesures d'activité qu'il contient.

En outre le symbole d'un cube estampille le fait.



Représentation d'un Fait

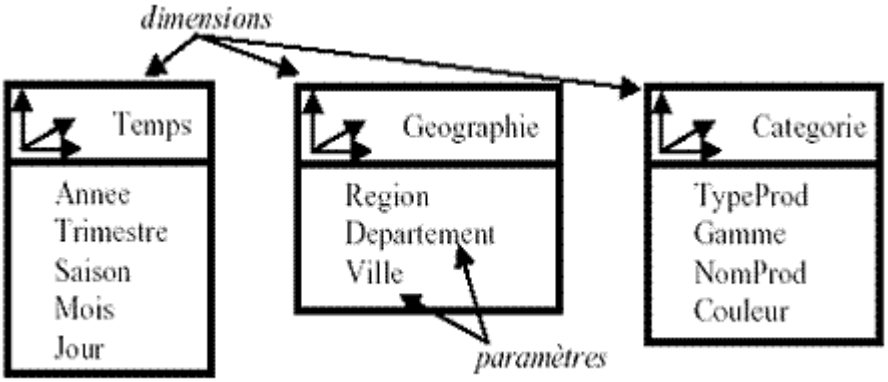
Concept de dimension

Le sujet analysé, c'est à dire le fait, est analysé suivant différentes perspectives. Ces perspectives correspondent à une catégorie utilisée pour caractériser les mesures d'activité analysées.

Une dimension se compose de paramètres correspondant aux informations faisant varier les mesures de l'activité. Les dimensions servent à enregistrer les valeurs pour lesquelles sont analysées les mesures de l'activité.

Une dimension est généralement formée de paramètres (ou attributs) textuels et discrets.

Les paramètres textuels sont utilisés pour restreindre la portée des requêtes afin de limiter la taille des réponses. Les paramètres sont discrets, c'est à dire que les valeurs possibles sont bien déterminées et sont des descripteurs constants.



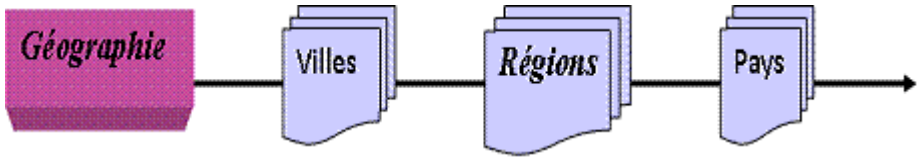
Représentation de dimension

Hiérarchie d'une dimension :

Lors de processus OLAP, les données sont généralement analysées en partant d'un faible niveau de détail vers des données plus détaillées pour « forer vers le bas ».

Pour définir ces différents niveaux de détail, chaque dimension est munie d'une (ou plusieurs) hiérarchie(s) des paramètres. La hiérarchie sert lors des analyses pour restreindre ou accroître les niveaux de détail de l'analyse .

Une hiérarchie organise les paramètres d'une dimension selon une relation « est_plus_fin » conformément à leur niveau de détail



Hiérarchie d'une dimension

Les paramètres des dimensions sont organisés suivant une hiérarchie ; les paramètres sont ordonnés par une relation "est_plus_fin ". Par exemple, pour la dimension Géographique : chaque ville appartient à une région qui est située dans un pays.

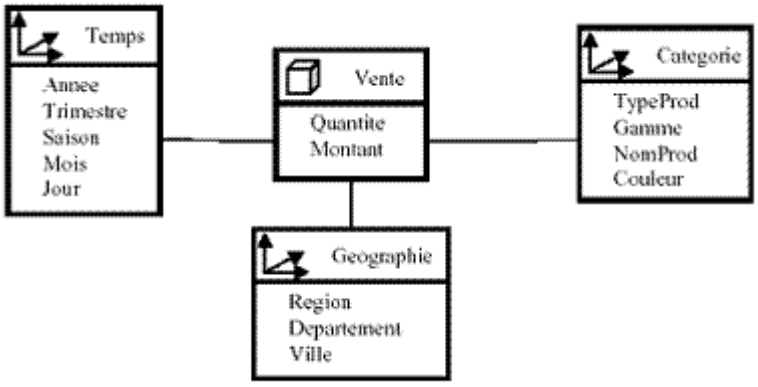
B. Types de modélisation

Nous distinguons trois principaux types de modélisation multidimensionnelle au niveau conceptuel

- Etoile
- Flocon de neige
- Constellation

1. Modèle en Etoile :

A partir du fait et des dimensions, il est possible d'établir une structure de données simple qui correspond au besoin de la modélisation multidimensionnelle. Cette structure est constituée du fait central et des dimensions. Ce modèle représente visuellement une étoile, on parle de modèle en étoile.



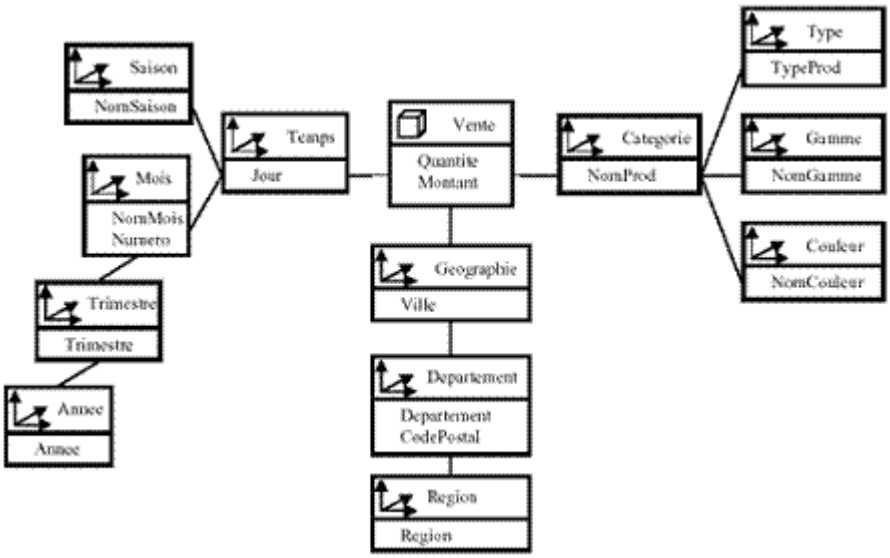
Modèle en étoile

2. Flocon de neige (snowflake)

Une modélisation en flocon consiste à décomposer les dimensions du modèle en étoile en sous hiérarchies. La modélisation en flocon est donc une émanation de la modélisation en étoile ; le fait est conservé et les dimensions sont éclatées conformément à sa hiérarchie des paramètres.

L'avantage de cette modélisation est de formaliser une hiérarchie au sein d'une dimension.

Par contre, la modélisation en flocon génère une plus grande complexité en termes de lisibilité et de gestion.

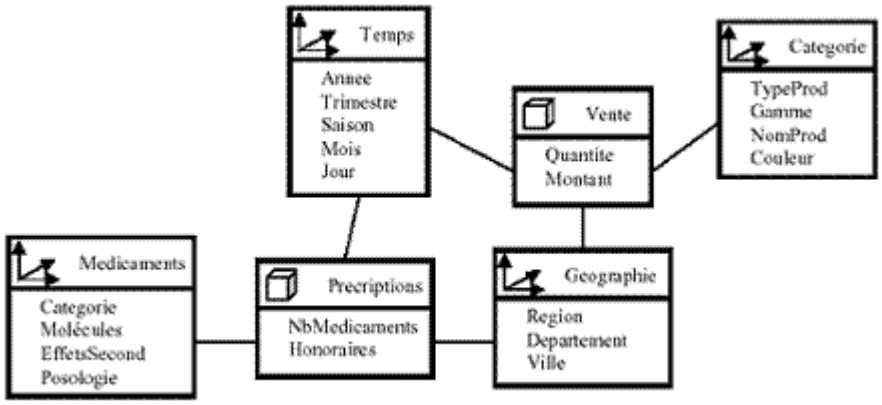


Modèle en flocon de neige

3. Constellation de faits

Une autre technique de modélisation, issue du modèle en étoile, est la modélisation en constellation. Il s'agit de fusionner plusieurs modèles en étoile qui utilisent des dimensions communes.

Un modèle en constellation comprend donc plusieurs faits et des dimensions communes ou non.



Constellation de faits

4. Faits additifs/ semi additifs

Faits additifs

Un fait additif est additionnable suivant toutes les dimensions

Par exemple : quantité vendue, chiffre d 'affaire

Faits semi additifs

Un faits semi additif est additionnable seulement suivant certaines dimensions.

Par exemple : niveau de stock, de solde (additionnable uniquement sur la dimension temps)

Faits non additifs

Un fait non additif est non additionnable quelque soit la dimension : utilisation : comptage des faits ou affichage 1 par 1

Par exemple : un attribut ratio , marge brute = 1 - Coût/CA

C. Niveau Logique

Nous distinguons 4 types de modélisation au niveau logique :

- R-OLAP : Relational-OLAP
- M-OLAP : Multidimensional-OLAP F
- OLAP : Hybrid-OLAP
- O-OLAP : Object-OLAP

1. Modélisation logique (R-OLAP)

Les dimensions et faits sont représentés par des tables relationnelles.

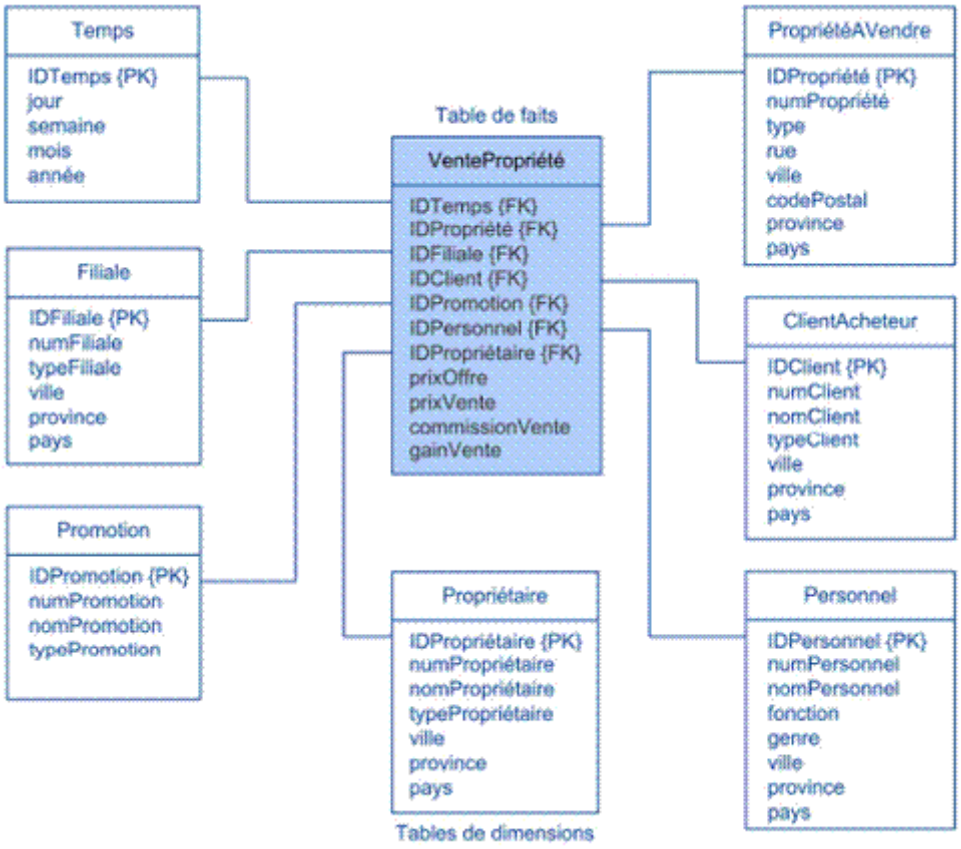
Deux cas sont à envisager, en fonction de la normalisation ou non des tables de dimension

a) Modélisation logique (R-OLAP) : DENORMALISE

Les tables de dimensions ne sont pas normalisés car les hiérarchies ne sont pas représentées

Avantage : simplicité

Inconvénients : redondance



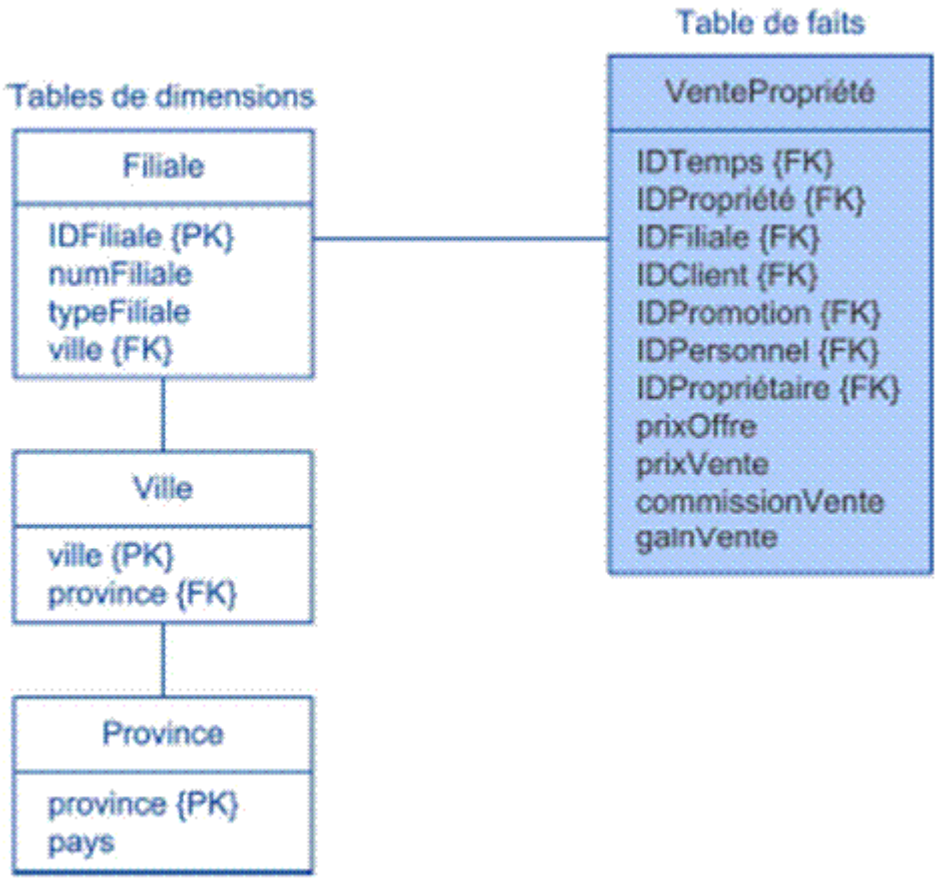
ROLAP dénormalisé

b) Modélisation logique (R-OLAP) : FLOCON NORMALISE

Les dimensions sont normalisées suivant les hiérarchies

avantages : maintenance des tables de dimensions simplifiée , réduction de la redondance

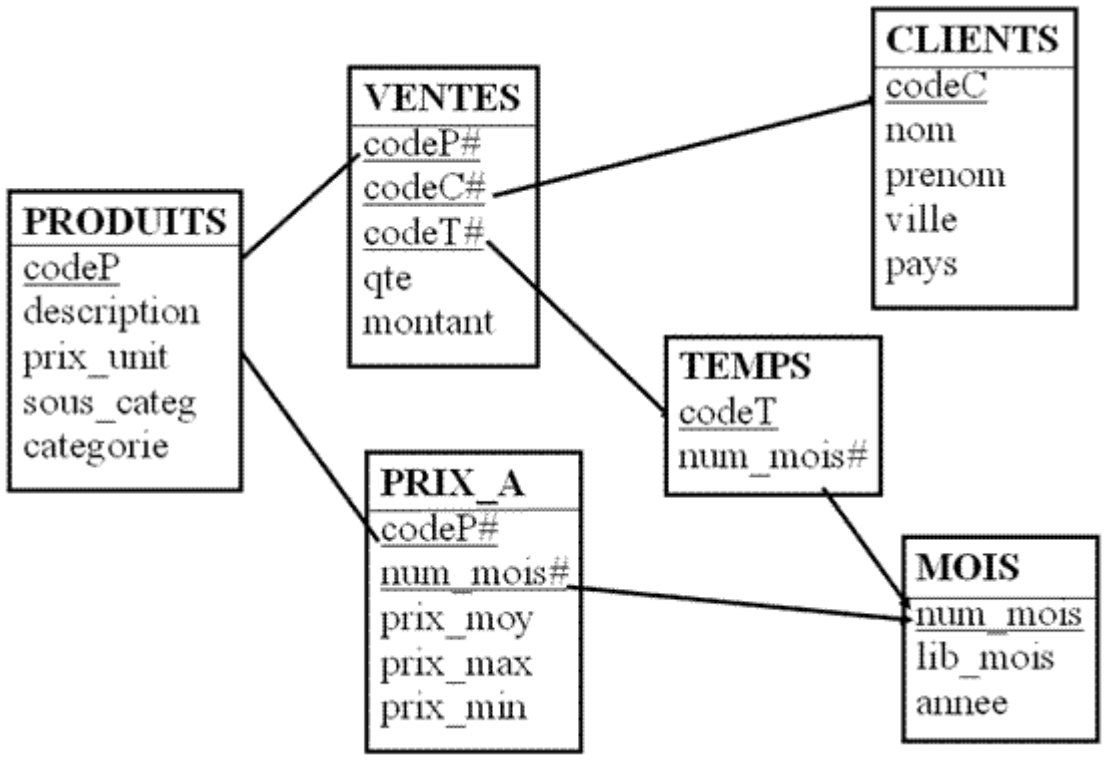
Inconvénient : navigation coûteuse



ROLAP normalisé

c) Modélisation logique (R-OLAP) : FLOCON HYBRIDE

Certaines dimensions sont normalisées, mais pas toutes.



Flocon hybride

D. Méthode de conception

Nous décrivons une méthodologie par étapes pour la conception de la base de données d'un entrepôt de données. Cette méthodologie a été initialement proposée par Kimball et s'appelle la Méthodologie à neuf étapes

Cette Méthodologie spécifie les phases requises pour la conception d'un mini-entrepôt. Cependant, elle lie aussi ensemble des mini-entrepôts de données séparés, pour que, à terme, ils fusionnent en un entrepôt de données cohérent global.

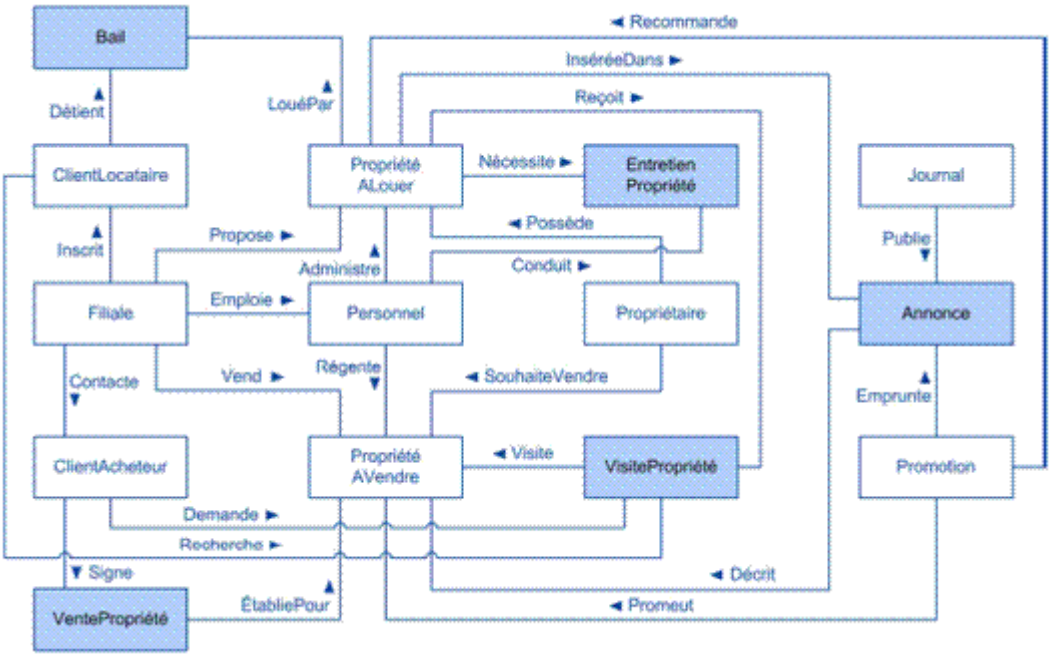
1. Étape 1 : Choisir la procédure

La procédure (ou fonction) fait référence au sujet d'un magasin de données particulier. Le premier magasin de données à construire est celui qui est susceptible d'être livré à temps, en respectant les budgets, et est destiné à répondre aux questions professionnelles les plus importantes au point de vue commercial

Exemple

Pour sélectionner le principal magasin de données d'une agence immobilière, nous commençons par déterminer que les processus métier discrets de l'agence sont notamment :

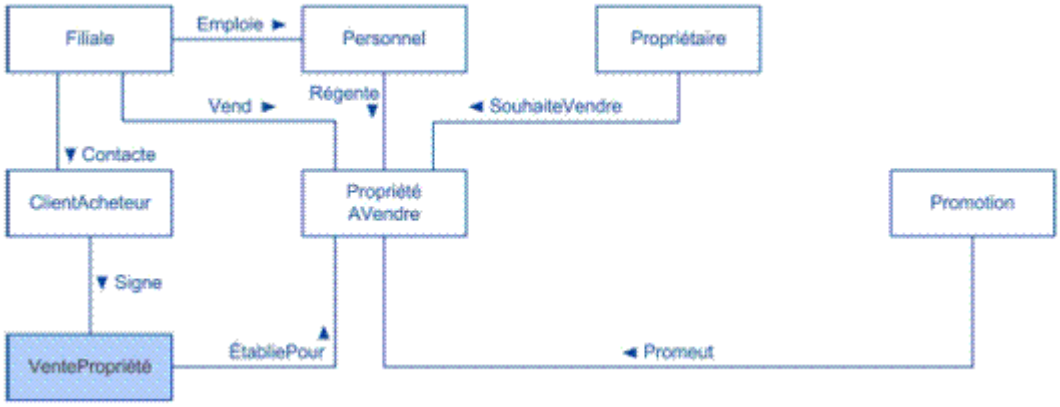
- Les ventes de propriétés;
- Les locations (baux) de propriétés;
- Les visites de propriétés;
- La publicité des propriétés;
- L'entretien des propriétés.



Modèle de données initial

Les exigences de données associées à ces processus sont exposées dans le diagramme E/A précédent. Notez que ce diagramme EA constitue une partie de la documentation du design, qui décrit les systèmes de traitement de transaction en ligne nécessaires pour soutenir les processus métier de l'agence immobilière.

Les entités assombries représentent les faits principaux pour chacun des processus métier de l'agence. Le processus métier choisi comme premier magasin de données est celui des ventes de propriétés



Fonction choisie

2. Étape 2 : Choisir le grain

Choisir le grain signifie décider exactement de ce que représente un enregistrement d'une table de faits.

Exemple

L'entité VentePropriété représente les faits relatifs à chaque vente de propriété et devient la table des faits du schéma en étoile des ventes de propriétés. Par

conséquent, le grain de la table de faits VentePropriété est une vente de propriété individuelle.

Ce n'est que lorsque nous avons choisi le grain de la table de faits que nous pouvons commencer à identifier les dimensions de la table de faits.

Par exemple, les entités Filiale, Personnel, Propriétaire, ClientAcheteur, PropriétéAVendre et Promotion deviendront les tables de dimensions du schéma en étoile des ventes de propriétés. Nous ajoutons aussi le Temps en tant que dimension principale, car elle est toujours présente dans les schémas en étoile.

La décision relative au grain pour la table de faits détermine aussi le grain de chacune des tables de dimension. Par exemple, si le grain de la table de faits VentePropriété est une vente individuelle de propriété, alors le grain de la dimension ClientAcheteur est l'ensemble des détails du client qui achète (ou a acheté) une propriété déterminée.

Exemple

Les entités Filiale, Personnel, Propriétaire, ClientAcheteur, PropriétéAVendre et Promotion deviendront les tables de dimensions du schéma en étoile des ventes de propriétés. Nous ajoutons aussi le Temps en tant que dimension principale, car elle est toujours présente dans les schémas en étoile.

La décision relative au grain pour la table de faits détermine aussi le grain de chacune des tables de dimension. Par exemple, si le grain de la table de faits VentePropriété est une vente individuelle de propriété, alors le grain de la dimension ClientAcheteur est l'ensemble des détails du client qui achète (ou a acheté) une propriété déterminée.

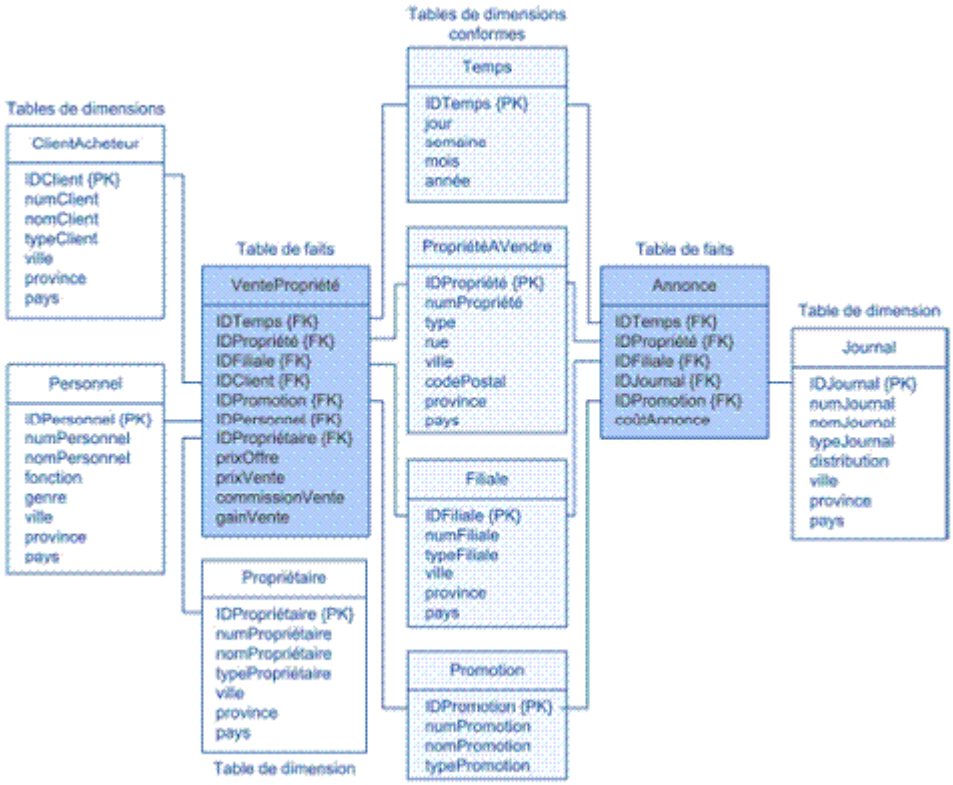
3. Étape 3 : Identifier les dimensions et s'y conformer

Les dimensions déterminent le contexte dans lequel nous pourrons poser des questions à propos des faits établis dans la table de faits. Un ensemble de dimensions bien constitué rend le magasin de données compréhensible et en simplifie l'utilisation.

Nous identifions les dimensions avec suffisance de détails, pour décrire des choses telles que les clients et les propriétés avec la granularité correcte.

Exemple

Tout client de la table de dimension ClientAcheteur est décrit par les attributs IDClient, numClient, nomClient, typeClient, ville, province et pays.



Identification des dimensions

Si une dimension, quelle qu'elle soit, apparaît dans deux magasins de données, les deux tables doivent constituer exactement la même dimension ou elles doivent former un sous-ensemble mathématique l'une de l'autre. Ce n'est que de cette manière que deux mmagasins de données peuvent partager une ou plusieurs dimensions au sein de la même application.

Lorsqu'une dimension sert dans plus d'un magasin de données, la dimension est considérée comme conforme.

Parmi les exemples de dimensions qui doivent être conformes entre les ventes de propriétés et la publicité des propriétés, citons les dimensions Temps, PropriétéAVendre, Filiale et Promotion.

Si la synchronisation de ces dimensions n'est pas assurée ou si la moindre désynchronisation est permise entre les magasins de données, alors c'est l'ensemble de l'entrepôt de données qui échouera, parce que nous serons incapables d'exploiter les deux magasins de données ensemble.

Attention

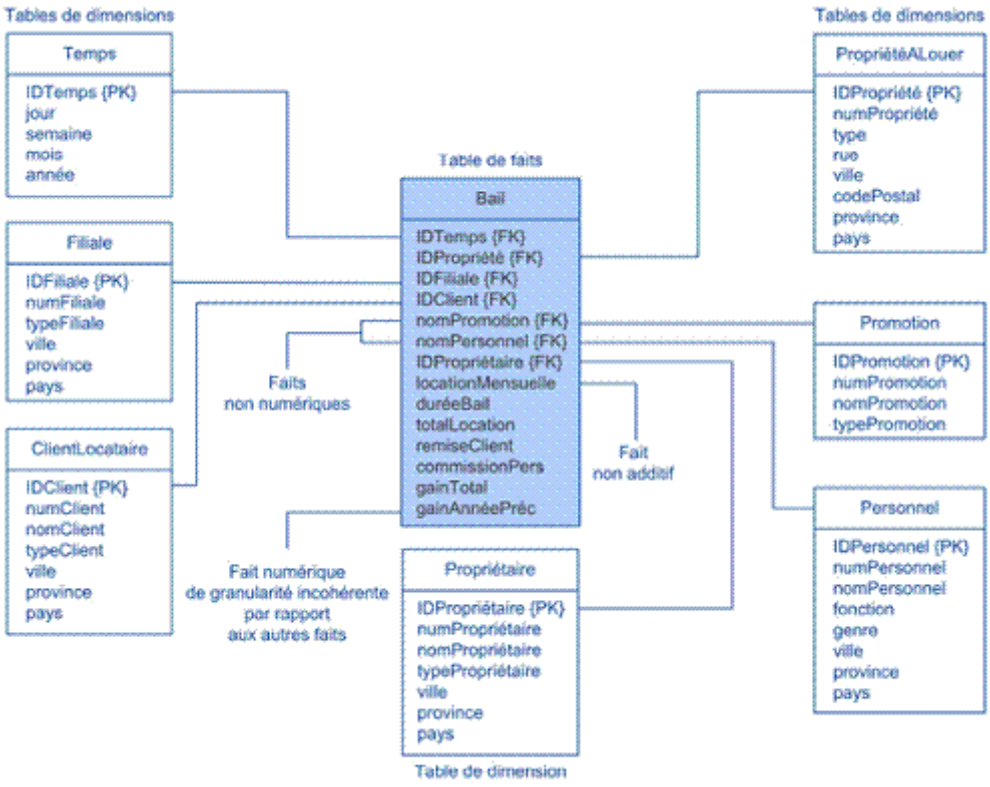
Un ensemble de dimensions mal présenté ou incomplet réduira à coup sûr l'utilité d'un magasin de données pour une entreprise.

4. Étape 4 : Choisir les mesures

Le grain de la table de faits détermine les faits utilisables dans le magasin de données. Tous les faits doivent être exprimés au niveau implicite imposé par le grain. En d'autres termes, si le grain de la table de faits est une vente de propriété bien précise, alors toutes les mesures doivent se référer à cette vente déterminée. Les mesures doivent être, outre numériques, également additifs.

Exemple

La figure suivante nous présentons une table de faits mal structurée. Cette table de faits est inutilisable, du fait de la présence de faits non numériques (nomPromotion et nomPersonnel), d'un fait non additif (locationMensuelle) et d'un fait (gainAnnéePréc) dont la granularité diffère de celle des autres faits de la table.



Choix du fait

La figure suivante montre les corrections apportées à la table de faits Bail pour que la table de faits soit correctement structurée.

Les faits additionnels (ou « additifs ») s'ajoutent à tout moment à une table de faits, à condition qu'ils demeurent cohérents avec le grain de la table

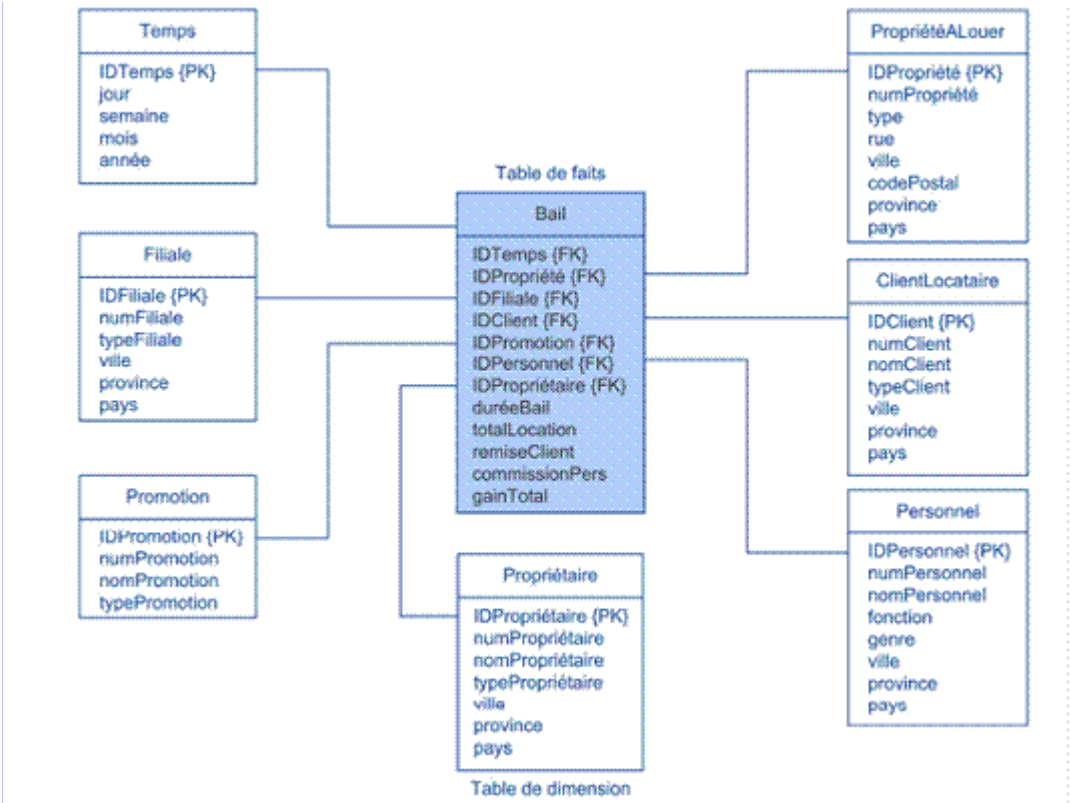


Table de fait corrigée

5. Étape 5 : Emmagasiner les calculs préliminaires dans la table des faits

Une fois que les faits ont été choisis, il est nécessaire de les réexaminer un à un, pour déterminer si des opportunités apparaissent d'exploiter des calculs préliminaires.

6. Étape 6 : Finaliser les tables de dimensions

Au cours de cette étape, nous revenons aux tables de dimensions et y ajoutons toutes les descriptions textuelles possibles aux dimensions. Les descriptions textuelles seront aussi intuitives et compréhensibles que possible pour les utilisateurs.

L'utilité d'un magasin de données est en effet déterminée aussi par la portée et la nature des attributs des tables de dimensions.

7. Étape 7 : Choisir la durée de la base de données

La durée mesure le saut dans le passé qu'une table de faits permet d'effectuer. Dans nombre d'entreprises, une exigence se manifeste de parcourir en même temps des périodes d'une ou deux années précédentes. Dans d'autres sociétés, des exigences légales imposent parfois de conserver des données vieilles de cinq ans, voire plus. Les très grandes tables de faits donnent lieu à deux soucis relatifs à la conception d'un entrepôt de données.

- Primo, plus les données sont anciennes et plus des difficultés risquent d'apparaître quant à la détermination de l'origine des données. En effet, plus les données sont anciennes, plus la probabilité augmente de faire face à des

problèmes liés à la lecture et à l'interprétation d'anciens fichiers.

- Secundo, il est impératif d'utiliser les anciennes versions des dimensions, et non les versions les plus récentes.

8. Étape 8 : Suivre les dimensions à modification lente

Le problème des dimensions à modification lente signifie par exemple que la description appropriée d'un ancien client et d'une ancienne filiale doit intervenir en accord avec un ancien historique de transaction. Souvent, l'entrepôt de données doit appliquer une clé généralisée à ces dimensions importantes, de manière à distinguer les multiples instantanés des clients et des filiales sur une certaine période de temps.

Nous pouvons distinguer trois types fondamentaux de dimensions à modification lente :

1. le Type 1, où un attribut de dimension modifié est écrasé;
2. le Type 2, où un attribut de dimension modifié provoque la création d'un nouvel enregistrement de dimension;
3. et le Type 3, où un attribut de dimension modifié provoque la création d'un attribut alternatif, pour que les deux valeurs, l'ancienne et la nouvelle, soient simultanément accessibles dans le même enregistrement de dimension.

9. Étape 9 : Décider des priorités de requêtes et des modes de requêtes

Au cours de cette étape, nous prenons en considération les soucis liés au design physique. Les soucis les plus prédominants, relatifs au design physique et qui affectent la perception du magasin de données par l'utilisateur final, sont l'ordre de tri physique de la table de faits sur disque et la présence de résumés ou d'agrégats préenregistrés

À la fin de cette la mise en pratique de cette méthodologie, nous obtenons un design d'un magasin de données qui respecte les exigences d'un processus métier déterminé et assure aussi une intégration aisée avec les autres magasin de données liés, pour constituer en définitive l'entrepôt de données de toute l'entreprise.

E. Modélisation physique (Oracle 9i, 10G, 11G)

Création des tables de faits et de dimensions

2 solutions :

1. Commande SQL (CREATE TABLE) et programme PL/SQL avec requête d'extraction pour l'alimentation
2. Vue matérialisée : vue matérialisée définissant la table de dimension et son alimentation

Syntaxe

CREATE MATERIALIZED VIEW <nomvue>

BUILD { IMMEDIATE|DEFERRED }

REFRESH { COMPLETE|FAST|FORCE|NEVER } { ON DEMAND|ON COMMIT }

AS SELECT ... ;

Options

IMMEDIATE : Création de la vue matérialisée et population de la vue

DEFERRED : Création de la vue matérialisée sans être alimentée en données. DBMS_MVIEW.REFRESH(<liste_vues>) alimente la vue

ON COMMIT : Rafraîchissement à chaque fin de transaction modifiant les tables sources

ON DEMAND : Rafraîchissement avec DBMS_MVIEW.REFRESH

COMPLETE : Recalcul complet de la vue

FAST : Application d'un rafraîchissement incrémental

FORCE : FAST si possible, COMPLETE sinon NEVER : pas de rafraîchissement

Dimensions

La création de dimensions consiste à définir la hiérarchie de chaque table de dimension (ou vue materialisée).

Syntaxe

```
CREATE DIMENSION <nomdimension>
LEVEL <niveau1> IS (<nomtable.nomattribut1>)
LEVEL <niveau2> IS (<nomtable.nomattribut2>) ...
HIERARCHY <nomhierarchie1> ( <niveaui1> CHILD OF <niveaui2> CHILD OF ...
JOIN KEY (nomtable.nomattribut) REFERENCES niveauk)
HIERARCHY <nomhierarchie2> ( <niveaui1> CHILD OF <niveaui2> CHILD OF ...
JOIN KEY (nomtable.nomattribut) REFERENCES niveauk') ...
ATTRIBUTE <niveauj1> DETERMINES <nomattributx>
ATTRIBUTE <niveauj2> DETERMINES <nomattributy> ... ;
```

Exemple

```
CREATE MATERIALIZED VIEW CLIENTS
BUILD IMMEDIATE
REFRESH COMPLETE ON DEMAND
AS SELECT codeC, nom, prenom, ville, pays
FROM CLIENTS_DW cl, PAYS_DW pa
WHERE cl.codeP=pa.codeP;
ALTER TABLE CLIENTS ADD CONSTRAINT pk_clients PRIMARY KEY(codeC);
CREATE DIMENSION CLIENTS_DIM
LEVEL niv_codeC IS (CLIENTS.codeC)
LEVEL niv_ville IS (CLIENTS.ville)
LEVEL niv_pays IS (CLIENTS.pays)
HIERARCHY CLIENTS_H ( niv_codeC CHILD OF niv_ville CHILD OF niv_pays )
ATTRIBUTE niv_codeC DETERMINES CLIENTS.nom
ATTRIBUTE niv_codeC DETERMINES CLIENTS.prenom;
```

Exemple : Dimension Produits (cas modèle en étoile)

```
CREATE MATERIALIZED VIEW PRODUITS
BUILD IMMEDIATE
REFRESH COMPLETE ON DEMAND
AS SELECT pr.codeP,pr.description,pr.prix_unit, c1.designation AS categorie
FROM PRODUITS_DW pr, CATEGORIES_DW c1
```

```
WHERE pr.codeCa=c1.codeCa ;
ALTER TABLE PRODUITS ADD CONSTRAINT pk_produits PRIMARY KEY(codeP);
CREATE DIMENSION PRODUITS_DIM
LEVEL N_codeP IS (PRODUITS.codeP)
LEVEL N_categ IS (PRODUITS.categorie)
HIERARCHY PR_H (N_codeP CHILD OF N_categ)
ATTRIBUTE N_ codeP DETERMINES PRODUITS.description
ATTRIBUTE N_ codeP DETERMINES PRODUITS.prix_unit;
```

Exemple : Dimension Produits (cas modèle en flocon de neige)

```
CREATE MATERIALIZED VIEW PRODUITS
BUILD IMMEDIATE
REFRESH COMPLETE ON DEMAND
AS SELECT pr.codeP,pr.description,pr.prix_unit, pr.codeCa AS codeCa
FROM PRODUITS_DW pr;
CREATE MATERIALIZED VIEW CATEGORIES
BUILD IMMEDIATE
REFRESH COMPLETE ON DEMAND
AS SELECT c2.codeCa,c2.designation AS categorie
FROM CATEGORIES_DW c2
ALTER TABLE CATEGORIES ADD CONSTRAINT pk_categorie PRIMARY KEY(CodeCa);
ALTER TABLE PRODUITS ADD
( CONSTRAINT pk_produits PRIMARY KEY(codeP),
CONSTRAINT fk_prod_cat FOREIGN KEY(codeCa) REFERENCES CATEGORIES(codeCa)
);
CREATE DIMENSION PRODUITS_DIM
LEVEL N_codeP IS (PRODUITS.codeP)
LEVEL N_categ IS (CATEGORIES.codeCa)
HIERARCHY PR_H ( N_codeP CHILD  OF N_categ
JOIN KEY (PRODUITS.codeCa) REFERENCES N_categ )
ATTRIBUTE N_codeP DETERMINES (PRODUITS.description, PRODUITS.prix_unit)
ATTRIBUTE N_categ DETERMINES CATEGORIES.categorie;
```

Exemple

```
CREATE MATERIALIZED VIEW TEMPS
BUILD IMMEDIATE
REFRESH COMPLETE ON DEMAND
AS SELECT DISTINCT dateC AS codeT, TO_CHAR(dateC, 'MM') AS num_mois, TO_CHAR(dateC, 'MONTH') AS lib_mois, TO_CHAR(dateC, 'YYYY') AS annee
FROM COMMANDES_DW;
ALTER TABLE TEMPS ADD CONSTRAINT pk_temps PRIMARY KEY(codeT);
CREATE DIMENSION TEMPS_DIM
LEVEL codeT IS (TEMPS.codeT)
LEVEL num_mois IS (TEMPS.num_mois)
LEVEL annee IS (TEMPS.annee)
```

```
HIERARCHY TEMPS_H ( codeT CHILD OF num_mois CHILD OF annee )
ATTRIBUTE num_mois DETERMINES TEMPS.lib_mois;
```

Exemple : Définition de la table de fait (vue matérialisée)

```
CREATE MATERIALIZED VIEW VENTES
BUILD IMMEDIATE
REFRESH COMPLETE ON DEMAND
AS SELECT codeP, codeC, dateC AS codeT, SUM(qte) AS qte, SUM(montant_sum) AS montant
FROM COMMANDES_DW co, LIGNES_COM_DW lc
WHERE co.refC = lc.refC
GROUP BY codeP, codeC, dateC;
ALTER TABLE VENTES ADD CONSTRAINT pk_ventes PRIMARY KEY(copeP, codeC, codeT);
ALTER TABLE VENTES ADD CONSTRAINT fk_ventes_produits FOREIGN KEY(copeP) REFERENCES PRODUITS(codeP);
ALTER TABLE VENTES ADD CONSTRAINT fk_ventes_clients FOREIGN KEY(copeC) REFERENCES CLIENTS(codeC);
ALTER TABLE VENTES ADD CONSTRAINT fk_ventes_temps FOREIGN KEY(copeT) REFERENCES TEMPS(codeT);
```

Exemple : Définition du fait Ventes (Table)

```
CREATE TABLE ventes (
codeP INTEGER NOT NULL,
codeT DATE NOT NULL,
codeC INTEGER NOT NULL,
qte INTEGER,
montant NUMBER(8,2),
CONSTRAINT pk_ventes PRIMARY KEY(codeP,codeT,codeC),
CONSTRAINT fk_ventes_produits FOREIGN KEY(copeP) REFERENCES PRODUITS(codeP),
CONSTRAINT fk_ventes_clients FOREIGN KEY(copeC) REFERENCES CLIENTS(codeC),
CONSTRAINT fk_ventes_temps FOREIGN KEY(copeT) REFERENCES TEMPS(codeT));
```

Exemple : Création du script PL/SQL d 'alimentation

```
CREATE OR REPLACE PROCEDURE alim_ventes IS
BEGIN
FOR nuplet IN (SELECT codeP,codeC,dateC,SUM(qte) AS qte, SUM(montant_sum) AS mnt FROM COMMANDES_DW co, LIGNES_COM_DW lc WHERE co.refC = lc.refC GROUP BY codeP, codeC, dateC) LOOP
INSERT INTO ventes VALUES ( nuplet.codeP, nuplet.dateC, nuplet.codeC, nuplet.qte, nuplet.mnt);
END LOOP;
END alim_ventes;
/
EXECUTE alim_ventes;
```

Création de cube

La procédure CWM_OLAP_CUBE.Create_Cube permet de créer un cube de données

Exemple

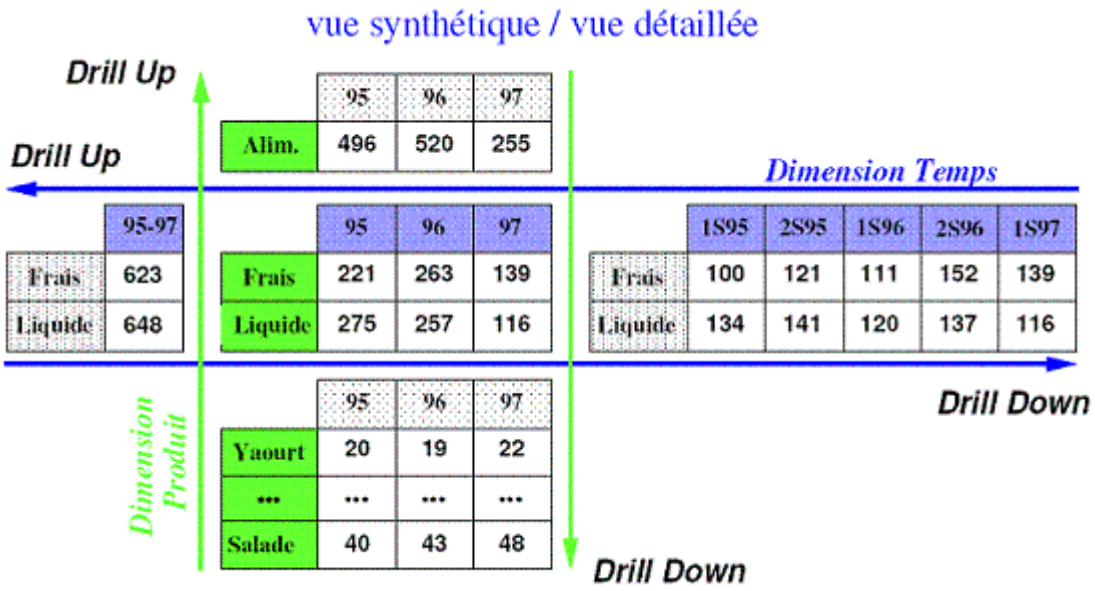
```
declare DM_CLIENTS number;
DM_TEMPS number;
tmp number;
begin
CWM_OLAP_CUBE.Create_Cube('SYSTEM','CUBE_VENTES','Ventes','');
DM_CLIENTS :=
CWM_OLAP_CUBE.Add_Dimension('SYSTEM','CUBE_VENTES','SYSTEM','CLIENTS_DIM','DM_CLIENTS');
CWM_OLAP_CUBE.Set_Default_Calc_Hierarchy('SYSTEM','CUBE_VENTES','CLIENTS_H','SYSTEM','CLIENTS_DIM','DM_CLIENTS');
CWM_OLAP_CUBE.Map_Cube('SYSTEM','CUBE_VENTES','SYSTEM','VENTES','FK_VENTES_CLIENTS','NIV_NUMCL','SYSTEM',
'CLIENTS_DIM','DM_CLIENTS');
DM_TEMPS :=
CWM_OLAP_CUBE.Add_Dimension('SYSTEM','CUBE_VENTES','SYSTEM','TEMPS_DIM','DM_TEMPS');
CWM_OLAP_CUBE.Set_Default_Calc_Hierarchy('SYSTEM','CUBE_VENTES','TEMPS_H','SYSTEM','TEMPS_DIM','DM_TEMPS');
CWM_OLAP_CUBE.Map_Cube('SYSTEM','CUBE_VENTES','SYSTEM','VENTES','FK_VENTES_TEMPS','CODET','SYSTEM','TEMPS_DIM',
'DM_TEMPS');
commit;
end;
```

F. Opérations multidimensionnelles

Roll up (drill-up)/Drill down (roll down)

Roll up (drill-up) : résumer, agréger des données en montant dans une hiérachie ou en oubliant une dimension

Drill down (roll down): inverse de roll-up En descendant dans une hiérachie ou en ajoutant une dimension



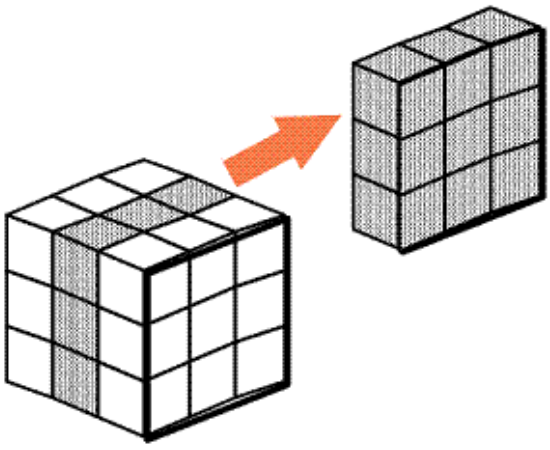
Drill up drill down

Slice and Dice

Sélection et Projection sur un sous ensemble de dimensions

		1995	1996	1997
Frais	IdF	220	265	284
	Province	225	245	240
Liquide	IdF	163	152	145
	Province	187	174	184

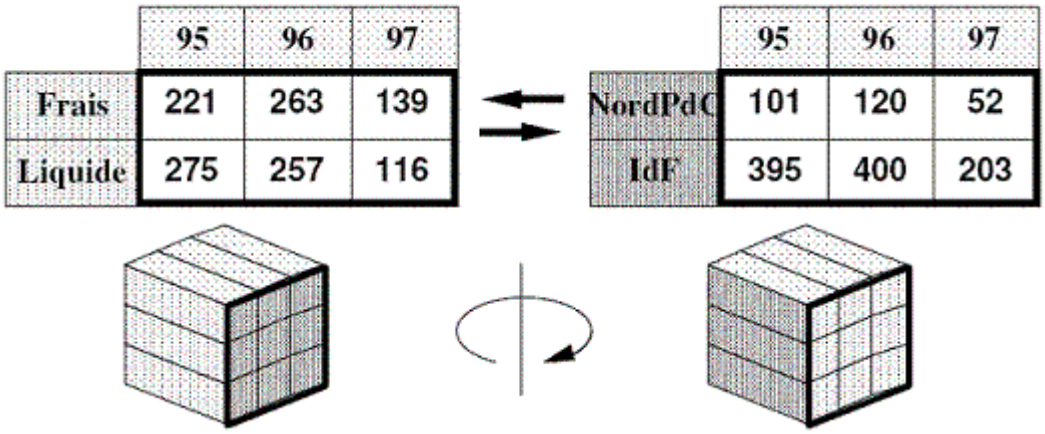
		1996
Frais	IdF	265
	Province	245
Liquide	IdF	152
	Province	174



Slicing

Rotate (Pivot)

Tourner le cube pour visualiser une face



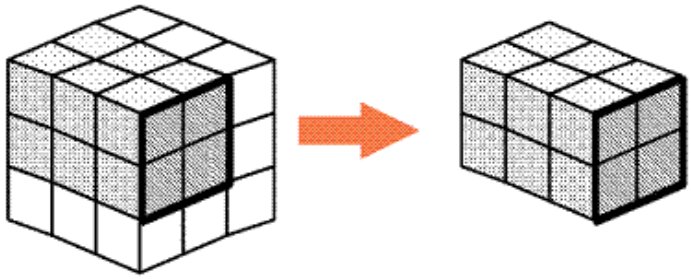
Rotate

Scoping

Sélection d'un sous cube

		1995	1996	1997
Frais	IdF	220	265	284
	Province	225	245	240
Liquide	IdF	163	152	145
	Province	187	174	184

		1995	1996
Frais	IdF	220	265
	Province	225	245



Scoping

G. Extensions SQL pour OLAP

GROUPING SETS

GROUP BY multiples en précisant quelles UNION sont souhaitées. L'imbrication d'attributs permet de séparer les GROUP BY simples de l'UNION de GROUP BY

GROUP BY GROUPING SETS ((jour, ville, pièce))	≡	GROUP BY jour, ville, pièce
GROUP BY GROUPING SETS (jour, ville, pièce)	≡	GROUP BY jour UNION GROUP BY ville UNION GROUP BY pièce
GROUP BY GROUPING SETS (jour,(ville,pièce))	≡	GROUP BY jour UNION GROUP BY ville, pièce

Grouping sets

ROLLUP

permet à une instruction SELECT de calculer plusieurs niveaux de sous-totaux parmi un groupe de dimensions spécifié. ROLLUP se glisse dans la clause GROUP BY d'une instruction SELECT selon le format suivant :

SELECT . . .

GROUP BY ROLLUP(listeDeColonnes)

ROLLUP crée des sous-totaux qui peuvent porter sur le niveau de détail, jusqu'au niveau du total, en fonction de la liste de colonnes spécifiée dans la clause ROLLUP.

ROLLUP calcule d'abord les valeurs d'agrégat standard spécifiées dans la clause GROUP BY, puis calcule progressivement les sous-totaux des niveaux supérieurs et se déplace parmi les colonnes de la liste jusqu'à aboutir à un total général.

ROLLUP crée ainsi des sous-totaux à n + 1 niveaux, où n est le nombre de colonnes de regroupement. Par exemple, si une requête spécifie un ROLLUP sur les colonnes de regroupement typePropriété, annéeMois et ville (donc, n = 3), alors l'ensemble de résultats comporte des lignes de 4 niveaux d'agrégation.

Exemple

Afficher les ventes totales d'appartements ou de maisons réalisées par les agences sises à Genève, Montréal ou Paris pour les mois de septembre et octobre 2004.

Dans cet exemple, nous avons besoin d'identifier d'abord les agences (filiales) des villes de Genève, Montréal et Paris, puis de calculer les agrégats des ventes totales obtenues pour les appartements et les maisons que les agences de ces villes ont vendus en septembre et octobre 2004.

SELECT typePropriété, annéeMois, ville, SUM(montantVente) AS totalVentes

FROM Filiale, PropriétéAVendre, VentesPropriétés

WHERE Filiale.numFiliale = VentesPropriétés.numFiliale AND PropriétéAVendre.numPropriété = VentesPropriétés.numPropriété AND VentesPropriétés.annéeMois IN ('2004-09', '2004-10') ☐ AND Filiale.ville IN ('Genève', 'Montréal', 'Paris')

GROUP BY ROLLUP(typePropriété, annéeMois, ville);

Cette requête renvoie les ensembles de lignes décrits comme suit:

Les lignes d'agrégation qui proviendraient du GROUP BY, sans la présence de

ROLLUP.

Les sous-totaux de premier niveau obtenus sur la ville, pour chaque combinaison de typePropriété et annéeMois.

Les sous-totaux de second niveau obtenus pour l'agrégation sur annéeMois et ville, pour chaque valeur de typePropriété.

Une ligne de total général

typePropriété	annéeMois	ville	totalVentes
Appartement	2004-09	Genève	115432
Appartement	2004-09	Montréal	236573
Appartement	2004-09	Paris	7664
Appartement	2004-09		359669
Appartement	2004-10	Genève	123780
Appartement	2004-10	Montréal	323100
Appartement	2004-10	Paris	8755
Appartement	2004-10		455635
Appartement			815304
Maison	2004-09	Genève	77987
Maison	2004-09	Montréal	135670
Maison	2004-09	Paris	4765
Maison	2004-09		218422
Maison	2004-10	Genève	76321
Maison	2004-10	Montréal	166503
Maison	2004-10	Paris	4889
Maison	2004-10		247713
Maison			466135
			1281439

Rollup

CUBE

CUBE prend l'ensemble de colonnes de regroupement spécifiées et crée des sous-totaux pour chacune des combinaisons possibles. CUBE se place dans la clause GROUP BY d'une instruction SELECT, selon la forme suivante :

SELECT . . .

GROUP BY CUBE(listeDeColonnes)

Exprimée en termes d'analyse multidimensionnelle, CUBE génère tous les sous-totaux que l'on pourrait calculer pour un cube de données des dimensions spécifiées.

Ainsi, si nous indiquons CUBE(typePropriété, annéeMois, ville), l'ensemble de résultats contient toutes les valeurs que donnerait une instruction ROLLUP équivalente, plus toutes les combinaisons supplémentaires.

Dans l'exemple précédent, les totaux sur ville pour les combinaisons de types de propriétés ne sont pas calculés par la clause ROLLUP(typePropriété, annéeMois, ville), tandis qu'ils le sont par la clause CUBE(typePropriété, annéeMois, ville). Si la clause CUBE comporte n colonnes, alors les résultats renvoient 2 n combinaisons de

sous-totaux. L'exemple que nous venons de choisir illustre un cube tridimensionnel.

CUBE est utilisable dans toute situation qui nécessite l'obtention de tableaux croisés dynamiques. Les données nécessaires pour les tableaux croisés peuvent être générées en un seul SELECT contenant une clause CUBE. Comme ROLLUP, CUBE s'avère utile surtout lorsqu'il est nécessaire d'obtenir des tableaux de synthèse.

CUBE est particulièrement indiqué dans les requêtes qui emploient des colonnes de dimensions multiples et non de colonnes représentant des niveaux différents d'une seule et même dimension. Par exemple, un tableau croisé peut être souvent demandé pour connaître les sous-totaux de toutes les combinaisons de typePropriété, annéeMois et ville. Ce sont là trois dimensions indépendantes et l'analyse de toutes les combinaisons possibles de sous-totaux présente un intérêt pratique

Par contre, un tableau croisé présentant toutes les combinaisons possibles de l'année, du mois et du jour offrirait des valeurs d'un intérêt assez limité, puisqu'il existe une hiérarchie naturelle au sein de la dimension temps.

Exemple

Afficher tous les sous-totaux possibles des ventes de propriétés par les agences immobilières de Genève, Montréal et Paris sur les mois de septembre et d'octobre 2004.

```
SELECT typePropriété, annéeMois, ville, SUM(montantVente) AS totalVentes
FROM Filiale, PropriétéAVendre, VentesPropriétés
WHERE Filiale.numFiliale = VentesPropriétés.numFiliale AND PropriétéAVendre.numPropriété = VentesPropriétés.numPropriété AND VentesPropriétés.annéeMois IN ('2004-09', '2004-10') AND Filiale.ville IN ('Genève', 'Montréal', 'Paris')
GROUP BY CUBE(typePropriété, annéeMois, ville);
```

Les lignes mises en évidence sont celles obtenues en commun dans les tables de résultats des deux fonctions ROLLUP et CUBE.

La clause CUBE(typePropriété, annéeMois, ville), où n = 3, produit toutefois 23 = 8 niveaux d'agrégation, tandis que dans l'exemple précédant, la clause ROLLUP(typePropriété, annéeMois, ville), où n = 3, n'a produit que 3 + 1 = 4 niveaux d'agrégation.

Rank

Il faut bien comprendre la différence entre le classement (ranking) et le tri (ordering) : le classement trie un ensemble de résultats en fonction d'une fréquence statistique calculée sur une colonne ou une série de valeurs, tandis que le tri permet d'ordonner les résultats en fonction des occurrences d'une ou plusieurs colonnes.

Une fonction de classement calcule la position d'un enregistrement par rapport aux autres enregistrements d'un ensemble de données, selon les valeurs d'une série de mesures. Un certain nombre de fonctions de classement sont disponibles, comme RANK et DENSE_RANK. Leur syntaxe est la suivante :

```
RANK( ) OVER (ORDER BY listeDeColonnes)
DENSE_RANK( ) OVER (ORDER BY listeDeColonnes
```

Ces syntaxes sont incomplètes mais suffisent pour expliquer et illustrer l'utilité de ces fonctions.

La différence entre RANK et DENSE_RANK réside dans le fait que DENSE_RANK ne laisse aucun « trou » dans la séquence de classement, quand il y a des ex-æquo dans un classement.

typePropriété	annéeMois	ville	totalVentes
Appartement	**2004-09**	**Genève**	**115432**
Appartement	**2004-09**	**Montréal**	**236573**
Appartement	**2004-09**	**Paris**	**7664**
Appartement	**2004-09**		**359669**
Appartement	**2004-10**	**Genève**	**123780**
Appartement	**2004-10**	**Montréal**	**323100**
Appartement	**2004-10**	**Paris**	**8755**
Appartement	**2004-10**		**455635**
Appartement		Genève	239212
Appartement		Montréal	559673
Appartement		Paris	16419
Appartement			**815304**
Maison	**2004-09**	**Genève**	**77987**
Maison	**2004-09**	**Montréal**	**135670**
Maison	**2004-09**	**Paris**	**4765**
Maison	**2004-09**		**218422**
Maison	**2004-10**	**Genève**	**76321**
Maison	**2004-10**	**Montréal**	**166503**
Maison	**2004-10**	**Paris**	**4889**
Maison	**2004-10**		**247713**
Maison		Genève	154308
Maison		Montréal	302173
Maison		Paris	9654
Maison			**466135**
	2004-09	Genève	193419
	2004-09	Montréal	372243
	2004-09	Paris	12429
	2004-09		578091
	2004-10	Genève	200101
	2004-10	Montréal	489603
	2004-10	Paris	13644
	2004-10		703348
		Genève	393520
		Montréal	861846
		Paris	26073
			1281439

Cube

Par exemple si trois filiales détiennent ex-æquo la deuxième place en matière de montant total des ventes de propriétés, DENSE_RANK les identifie tous les trois en deuxième place et la filiale suivante en troisième place, tandis que, si la fonction RANK identifie aussi les trois filiale en deuxième place, la filiale suivante est considérée comme étant à la (2 + 3 =) cinquième place.

Exemple

Classer les ventes totales de propriétés pour les filiales situées à Montréal. Pour résoudre ce problème, nous calculons d'abord les totaux des ventes des propriétés de chaque agence immobilière de Montréal, puis nous classons les résultats.

La requête accède aux tables Filiale et VentesPropriétés.

Nous illustrons la différence entre les comportements des fonctions RANK et DENSE_RANK dans la requête suivante

SELECT numFiliale, SUM(montantVente) AS totalVentes,

RANK() OVER (ORDER BY SUM(montantVente)) DESC AS classement,

DENSE_RANK() OVER (ORDER BY SUM(montantVente)) DESC AS classement_dense

```
FROM Filiale, VentesPropriétés
WHERE Filiale.numFiliale = VentesPropriétés.numFiliale AND Filiale.ville = 'Montréal'
GROUP BY(numFiliale);
```

numFiliale	totalVentes	classement	classement_dense
F009	120 000 000	1	1
F018	92 000 000	2	2
F022	92 000 000	2	2
F028	92 000 000	2	2
F033	45 000 000	5	3
F046	42 000 000	6	4

Rank

Calculs fenêtrés

Les calculs fenêtrés portent ce nom car ils interviennent sur des « fenêtres » de données, prises dans une plage d'enregistrements sélectionnée en fonction de certains critères. La fenêtre de données peut ensuite se déplacer en fonction d'un élément qui représente le temps, présent dans l'une des colonnes du jeu de données, soit de façon explicite (valeurs discrètes), soit implicite (champ dérivé).

Les calculs fenêtrés peuvent ainsi servir à calculer des agrégats cumulatifs, mobiles et centrés. Ils renvoient une valeur pour chaque ligne de la table, dépendant des autres lignes de la fenêtre correspondante.

Par exemple, le fenêtrage permet de calculer des sommes cumulatives, des sommes mobiles, des moyennes mobiles, des valeurs minimales ou maximales mobiles, ainsi que d'autres mesures statistiques.

Les fonctions d'agrégat donnent accès à plus d'une ligne d'une table sans jointure réflexive et ne peuvent être utilisées que dans les clauses SELECT et ORDER BY d'une requête.

Exemple

Afficher les chiffres mensuels, les moyennes et les sommes mobiles à trois mois des ventes totales de propriétés de la filiale F003 pour les six premiers mois de 2004.

Nous calculons d'abord la somme des ventes de propriétés pour chacun des six premiers mois de 2004 au sein de la filiale F003, puis nous exploitons ces chiffres pour déterminer les moyennes mobiles à trois mois et les sommes mobiles à trois mois.

Autrement dit, nous calculons la moyenne mobile et la somme mobile des ventes totales de propriétés par la filiale F003 pour le mois de calcul et les deux mois qui précèdent le mois de calcul, puis nous passons au mois suivant et ainsi de suite pour atteindre le sixième mois.

La requête accède uniquement à la table VentesPropriétés. Pour la démonstration, nous créons une fenêtre mobile de trois à l'aide de la fonction ROWS 2 PRECEDING

(c'est-à-dire la ligne en cours plus les 2 précédentes) dans la requête qui suit

```
SELECT annéeMois, SUM(montantVente) AS ventesMensuelles, AVG(SUM(montantVente))
OVER (ORDER BY annéeMois, ROWS 2 PRECEDING) AS Moy mobile à 3 mois,
SUM(SUM(montantVente)) OVER (ORDER BY annéeMois ROWS 2 PRECEDING)
AS Somme mobile à 3 mois
FROM VentesPropriétés
WHERE numFiliale = 'F003'
AND annéeMois BETWEEN ('2004-01' AND '2004-06')
    GROUP BY annéeMois
    ORDER BY annéeMois;
```

Calcul fenêtré exemple

annéeMois	ventesMensuelles	Moy mobile à 3 mois	Somme mobile à 3 mois
2004-01	210000	210000	210000
2004-02	350000	280000	560000
2004-03	400000	320000	960000
2004-04	420000	390000	1170000
2004-05	440000	420000	1260000
2004-06	430000	430000	1290000

Calcul fenêtré

Notez que les deux premières lignes des calculs de la moyenne et de la somme mobiles à trois mois qui apparaissent dans la table résultante se basent sur un intervalle de plus petite taille que ce que nous avons demandé (sur les six premiers mois de 2004) car le calcul fenêtré ne peut atteindre des données au-delà de l'ensemble de données obtenu dans la requête.

Il est donc nécessaire de considérer les différentes tailles de fenêtres trouvées aux frontières des ensembles de résultats.

Covariance

Soit avgexp1 le résultat de AVG(expression1) et soit avgexp2 le résultat de AVG(expression2),

Alors COVARIANCE(expression1, expression2) = AVG((expression1 - avgexp1) * (expression2 - avgexp2)

Exemple

SELECT COVARIANCE(SALARY, BONUS)

FROM EMPLOYEE WHERE

WORKDEPT = 'A00'

Correlation

CORRELATION(expression1,expression2)=COVARIANCE(expression1,expression2)/ (STDDEV(expression1)*STDDEV(expression2))

Exemple

```
SELECT CORRELATION(SALARY, BONUS)
FROM EMPLOYEE
WHERE WORKDEPT = 'A00'
```

Data mining

Dans ce qui suit nous présenterons les concepts de base du data mining

A. Concepts de base

Définition

Le data mining est un procédé d'exploration et d'analyse de grands volumes de données en vue d'une part de les rendre plus compréhensibles et d'autre part de découvrir des corrélations significatives, c'est-à-dire des règles de classement et de prédiction dont la finalité ultime la plus courante est l'aide à la décision.

Définition

Le data mining est un procédé de production de connaissance. En terme de logique philosophique traditionnelle, le data mining consiste à produire des jugements (toutes les personnes sont x, la moyenne des y des personnes vaut tant, etc. : c'est l'étape de description et de compréhension des données) et des règles de raisonnements (si toutes les personnes sont « a » alors elles seront « b » : c'est l'étape modélisation qui permet la prédiction).

Pourquoi la naissance du data mining ?

- Augmentation des capacités de stockage des données (disques durs de giga octets).
- Augmentation des capacités de traitements des données (facilité d'accès aux données : il n'y a plus de bandes magnétiques ; accélération des traitements).
- Maturation des principes des bases de données (maturation des bases de données relationnelles).
- Croissance exponentielle de la collecte des données (scanners de supermarché, internet, etc.)
- Croissance exponentielle des bases de données : capacités atteignant le terabits et émergence des entrepôts de données : data warehouse, rendant impossible l'exploitation manuelle des données.
- Plus grande disponibilité des données grâce aux réseaux (intranet et internet).

Intérêt du data mining

Les entreprises sont inondées de données (scanners des supermarchés, internet, bases de données, etc.). Ces données languissent dans des entrepôts de données (ou référentiels, ou data warehouse). Le data mining permet d'exploiter ces données pour améliorer la rentabilité d'une activité.

- Le data mining permet ainsi d'augmenter le retour sur investissement des systèmes d'information.
- IL permet de comprendre et décider, savoir et prévoir (la raison et la volonté) .
- Le data mining est un outil qui permet de produire de la connaissance :
  - dans le but de comprendre les phénomènes dans un premier temps : SAVOIR
  - dans le but de prendre des décisions dans un second temps : PREVOIR pour DECIDER.

Méthodes du data mining

- Du bon sens. Il s'agit d'abord d'analyser les données avec du bon sens et un peu d'outillage mathématiques et statistiques élémentaire. ·
- Des algorithmes de calculs statistiques. Il s'agit ensuite d'appliquer des algorithmes de calculs à des données. Ces algorithmes sont plus ou moins complexes à mettre en oeuvre. Ils permettent de classer les données et de prédire des valeurs inconnues.

Les techniques du data mining

La production de règles de raisonnement se fait à partir de plusieurs techniques plus ou moins spécifiques au data mining. Ces techniques mixent à la fois des statistiques et de l'algorithmique. Globalement, on peut dire que certaines techniques visent à classer, d'autres visent à prédire.

1. Applications du data mining

Publics

- Le scientifique : pour comprendre certains phénomènes.
- L'analyste : pour produire des rapports pour les décideurs.
- Le décideur (au sens large) : pour l'aide à la décision.

Gestion de la relation client : Customer Relationship Management (CRM)

Principe : amélioration de la rentabilité par l'amélioration de la connaissance du client.

Matière première : les données sur le client.

Le CRM se divise en deux parties :

1. CRM analytique : collecte et analyse des données.
2. CRM opérationnel : choix des campagnes marketing (stratégie) et gestion des canaux de communication (forces commerciales, centres d'appel téléphoniques, internet, télévision, etc.)

Difficulté : tirer partie de la masse de données. Ne pas se noyer dedans.

Objectif : on ne veut plus seulement savoir : « combien de clients ont acheté tel produit pendant telle période ?», mais on veut savoir « quel est leur profil ? », « quels autres produits les intéresseront ? », « quand seront-ils de nouveau intéressés ? ».

Avec le data mining, on va s'intéresser à l'historique des données, autrement dit à un film du comportement de la variable étudiée (par exemple, le client) et pas seulement à une photographie. Le développement des systèmes d'informations avec des bases de données favorise la collecte de l'historique des données et les études de data mining. Et inversement : c'est parce qu'on pourra tirer quelque chose de ces historiques qu'on aura intérêt à les conserver.

Les systèmes d'information permettent de connaître le comportement singulier de chaque client. Le data mining permet de découvrir des corrélations statistiques grâce à l'étude des comportements de tous les clients, et ainsi de catégoriser le client et de pouvoir établir un marketing individualisé (one-to-one) et proactif (proposer ce qui est intéresse au moment où ça intéresse).

Exemple

- Une société de cosmétique de luxe détecte automatiquement ses meilleurs clients dès les premières transactions dans sa base de données pour les

traiter avec le plus d'égards possibles.

- Un opérateur de télévision par abonnement détecte les clients les plus sensibles à des offres de chaînes complémentaires à partir des appels téléphoniques des clients.
- Un directeur de théâtre segmente son public par des critères d'ancienneté, de durée et de fréquence de fréquentation (forme de la consommation), mais aussi par genre de spectacle (contenu de la consommation) pour adapter son offre et sa publicité.

Autres grands domaines d'application

- Secteur bancaire : le scoring, pour mieux cibler les propositions de prêts et éviter les surendettements (et donc les mauvais payeurs).
- Secteur de la téléphonie : prédiction de l'attrition (usure, churn en anglais), c'est-à-dire le changement d'opérateur.
- Grande distribution : analyse du panier de la ménagère pour déterminer les produits achetés simultanément.
- Web mining et e-commerce : 50% des clients d'un constructeur de machine achètent ses machines à travers le web. Mais seulement 0,5% des visiteurs du site deviennent clients. L'idée est de stocker les séquences de click des visiteurs et d'analyser les caractéristiques des acheteurs pour adapter le contenu du site.
- Text mining pour analyser les lettres de réclamation.
- Scientifique : identification et classification d'objets célestes.
- Médical : analyse de résultat d'une expérimentation
- Sécurité informatique : recherche de transactions frauduleuses par la police ; suivi des opérations des traders.

2. Distinctions entre statistiques et data mining

- La quantité de données : On a tendance à considérer que le data mining traite plus de données que les statistiques. C'est une idée partiellement juste. Dans l'absolu, les deux techniques peuvent en traiter autant. Dans la pratique, le data mining en traite plus.
- L'origine des données : Les statisticiens peuvent travailler sur des populations entières, mais le plus souvent ils travaillent sur des échantillons. Leur travail consistera à construire la représentativité de ces échantillons. Souvent, le statisticien est donc amené à collecter des données qui n'existent pas encore. Le data miner travaille plutôt sur des données qui existent déjà (issues de bases de données ou de data warehouse). De ce fait (et aussi du fait des techniques de modélisation qu'il utilise) il a tendance à travailler sur la population entière et pas sur des échantillons.
- L'analyse des données : Le data mining utilise des techniques issues de l'intelligence artificielle. Certaines techniques du data mining n'appartiennent qu'au data mining et ne font pas partie de la panoplie des techniques de l'analyse de données. C'est le cas des techniques de réseaux de neurones et d'arbres de décision. Le data mining tend à travailler avec moins d'a priori que les statistiques traditionnelles (moins d'hypothèses de départ). D'où la tendance à en faire un produit donnant des résultats miraculeux !
- Théorie vs pratique : Le data mining est orienté pratique et pas théorie : d'où le mépris des statisticiens ! Le data mining ne s'intéressent pas, contrairement aux statisticiens, aux lois générales de la statistique : c'est un domaine directement appliqué.

Le data mining recherche parfois plus la compréhensibilité des modèles que leur précision. Les modèles du data mining sont en général plus localisés (c'est-à-dire limité à une population très spécifique) que ceux des statisticiens

3. Les tâches du data mining

Contrairement aux idées reçues, le data mining n'est pas le remède miracle capable de résoudre toutes les difficultés ou besoins de l'entreprise. Cependant, une multitude de problèmes d'ordre intellectuel, économique ou commercial peuvent être regroupés, dans leur formalisation, dans l'une des tâches suivantes :

- Classification,
- Estimation,
- Prédiction,
- Groupement par similitudes,
- Segmentation (ou clusterisation),
- Description,

Afin de lever toute ambiguïté sur des termes qui peuvent paraître similaires, il semble raisonnable de les définir.

LA CLASSIFICATION

La classification se fait naturellement depuis déjà bien longtemps pour comprendre et communiquer notre vision du monde (par exemple les espèces animales, minérales ou végétales).

" La classification consiste à examiner des caractéristiques d'un élément nouvellement présenté afin de l'affecter à une classe d'un ensemble prédéfini. "

Dans le cadre informatique, les éléments sont représentés par un enregistrement et le résultat de la classification viendra alimenter un champ supplémentaire.

La classification permet de créer des classes d'individus (terme à prendre dans son acception statistique). Celles-ci sont discrètes : homme / femme, oui / non, rouge / vert / bleu, ...

Les techniques les plus appropriées à la classification sont :

- les arbres de décision,
- le raisonnement basé sur la mémoire,
- éventuellement l'analyse des liens.

L'ESTIMATION

Contrairement à la classification, le résultat d'une estimation permet d'obtenir une variable continue. Celle-ci est obtenue par une ou plusieurs fonctions combinant les données en entrée. Le résultat d'une estimation permet de procéder aux classifications grâce à un barème.

Par exemple, on peut estimer le revenu d'un ménage selon divers critères (type de véhicule et nombre, profession ou catégorie socioprofessionnelle, type d'habitation, etc ...). Il sera ensuite possible de définir des tranches de revenus pour classifier les individus.

Un des intérêts de l'estimation est de pouvoir ordonner les résultats pour ne retenir si on le désire que les n meilleures valeurs. Cette technique sera souvent utilisée en marketing, combinée à d'autres, pour proposer des offres aux meilleurs clients potentiels. Enfin, il est facile de mesurer la position d'un élément dans sa classe si celui ci a été estimé, ce qui peut être particulièrement important pour les cas limitrophes.

La technique la plus appropriée à l'estimation est : les réseaux de neurones.

LA PREDICTION

La prédiction ressemble à la classification et à l'estimation mais dans une échelle temporelle différente. Tout comme les tâches précédentes, elle s'appuie sur le passé et le présent mais son résultat se situe dans un futur généralement précisé.

La seule méthode pour mesurer la qualité de la prédiction est d'attendre !

Les techniques les plus appropriées à la prédiction sont :

- L'analyse du panier de la ménagère
- Le raisonnement basé sur la mémoire
- Les arbres de décision
- Les réseaux de neurones

LE REGROUPEMENT PAR SIMILITUDES

Le regroupement par similitudes consiste à grouper les éléments qui vont naturellement ensembles. La technique la plus appropriée au regroupement par similitudes est : L'analyse du panier de la ménagère

L'ANALYSE DES CLUSTERS

L'analyse des clusters consiste à segmenter une population hétérogène en sous-populations homogènes. Contrairement à la classification, les sous populations ne sont pas préétablies.

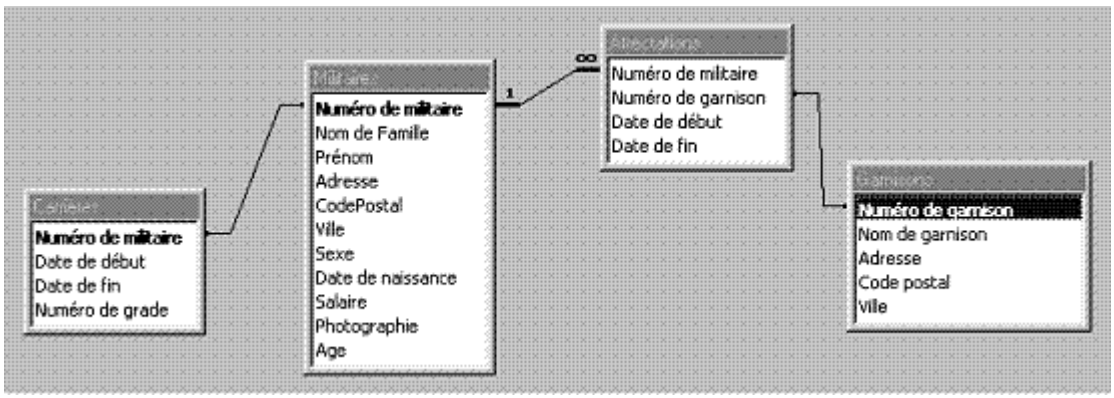
LA DESCRIPTION

C'est souvent l'une des premières tâches demandées à un outil de data mining. On lui demande de décrire les données d'une base complexe. Cela engendre souvent une exploitation supplémentaire en vue de fournir des explications.

La technique la plus appropriée à la description est : L'analyse du panier de la ménagère

Exemple

Prenons une base de données simple de suivi des militaires, du point de vue carrière et affectations, établie selon le modèle suivant :



Exemple

Nous retrouvons les fichiers militaires, affectations, garnisons, et carrières. Sur cet exemple, voyons comment utiliser chaque tâche du data mining. Nous supposons que le travail de préparation des données est fait.

La classification : Déterminer le grade en fonction du sexe, de l'âge, l'ancienneté, le salaire et les affectations. Déterminer le sexe en fonction de l'âge, l'ancienneté, le salaire et les affectations.

L'estimation : se fait sur des variables continues : Estimer l'âge en fonction du

grade, sexe, ancienneté et affectations Estimer le salaire en fonction de l'âge, sexe, ancienneté et affectations

La prédiction : Dans cet exemple, nous pouvons prédire par exemple quelle sera la prochaine affectation d'un militaire.

Le regroupement par similitudes : En utilisant cette technique, nous pouvons déterminer des règles de type : le militaire qui est sergent entre 25 et 30 ans sera lieutenant colonel entre 45 et 50 ans (fiabilité de n %).

La segmentation (ou clusterisation) Nous pouvons essayer de segmenter les militaires en fonction de leurs parcours (suivi de la carrière) et affectations.

La description : Dans cet exemple, la description se fera surtout autour des indicateurs statistiques traditionnels : âge moyen, pourcentage de femmes, salaire moyen

4. Présentation rapide de quelques techniques

ANALYSE DU PANIER DE LA MENAGERE (découverte de règles associatives)

L'analyse du panier de la ménagère est un moyen de trouver les groupes d'articles qui vont ensembles lors d'une transaction. C'est une technique de découverte de connaissances non dirigée (de type analyse de clusters) qui génère des règles et supporte l'analyse des séries temporelles (si les transactions ne sont pas anonymes).

Les règles générées sont simples, faciles à comprendre et assorties d'une probabilité, ce qui en fait un outil agréable et directement exploitable par l'utilisateur métier.

Exemple :

- Le client qui achète de la peinture achète un pinceau
- Le client qui achète un téléviseur achète un magnétoscope sous 5 ans.

LE RAISONNEMENT BASE SUR LA MEMOIRE

Le raisonnement basé sur la mémoire (RBM) est une technique de prédiction et de classification utilisée dans le cadre de la découverte de connaissances dirigée. Elle peut être également utilisée pour l'estimation.

Pour chaque nouvelle instance présentée, le système recherche le(s) voisin(s) le(s) plus proche(s) et procède ainsi à l'affectation ou estimation.

L'avantage du RBM est qu'il est facile à mettre en œuvre, très stable (les nouvelles données n'entraînent pas de refaire fonctionner un système de calcul) et supporte tout type de données.

LA DETECTION AUTOMATIQUE DE CLUSTERS

La détection automatique de clusters est une technique de découverte de connaissances non dirigée (ou apprentissage sans supervision). Elle consiste à regrouper les enregistrements en fonction de leurs similitudes. Chaque groupe représente un cluster.

C'est une excellente technique pour démarrer un projet d'analyse ou de data mining. Les groupes de similitudes permettront de mieux comprendre les données et d'imaginer comment les utiliser au mieux.

L'ANALYSE DES LIENS

L'analyse des liens est une technique de description qui s'inspire et repose sur la théorie des graphes. Elle consiste à relier des entités entre elles (clients, entreprises, ...) par des liens. A chaque lien est affecté un poids, défini par

l'analyse, qui quantifie la force de cette relation.

Cette technique peut être utilisée pour la prédiction ou la classification mais généralement une simple observation du graphe permet de mener à bien l'analyse.

LES ARBRES DE DECISION

Les arbres de décision sont utilisés dans le cadre de la découverte de connaissances dirigée. Ce sont des outils très puissants principalement utilisés pour la classification, la description ou l'estimation. Le principe de fonctionnement est le suivant : pour expliquer une variable, le système recherche le critère le plus déterminant et découpe la population en sous populations possédant la même entité de ce critère. Chaque sous population est ensuite analysée comme la population initiale.

Le modèle rendu est facile à comprendre et les règles trouvées sont très explicites. Ce système est donc très apprécié.

LES RESEAUX DE NEURONES

Les réseaux de neurones représentent la technique de data mining la plus utilisée. Pour certains utilisateurs, elle en est même synonyme. C'est une transposition simplifiée des neurones du cerveau humain.

Dans leur variante la plus courante, les réseaux de neurones apprennent sur une population d'origine puis sont capables d'exprimer des résultats sur des données inconnues. Ils sont utilisés dans la prédiction et la classification dans le cadre de découverte de connaissances dirigée.

Certaines variantes permettent l'exploration des séries temporelles et des analyses non dirigées (réseaux de Kohonen). Le champ d'application est très vaste et l'offre logicielle importante. Cependant, on leur reproche souvent d'être une "boite noire" : il est difficile de savoir comment les résultats sont produits, ce qui rend les explications délicates, même si les résultats sont bons.

LES ALGORITHMES GENETIQUES

Les algorithmes génétiques sont utilisés dans la découverte de connaissances dirigée. Ils permettent de résoudre des problèmes divers, notamment d'optimisation, d'affectation ou de prédiction. Leur fonctionnement s'apparente à celui du génome humain.

Le principe de fonctionnement est le suivant : les données sont converties en chaînes binaires (comme les chaînes d'ADN - acide désoxyribo nucléique-). Celles-ci se combinent par sélection, croisement ou mutation et donnent ainsi une nouvelle chaîne qui est évaluée. En fonction du résultat, les chaînes les plus faibles cèdent leur place aux plus fortes.

Cette technique est particulièrement intéressante pour résoudre des problèmes d'affectation ou des problèmes sur lesquels on peut poser une fonction d'évaluation car elle peut trouver des solutions optimisées parfois inexistantes dans les données d'origine.

5. Le processus standard d'une étude de data mining

Présentation du CRISP-DM

Le data mining est un processus méthodique : une suite ordonnée d'opérations aboutissant à un résultat. Le CRISP-DM (Cross Industry Standard Process for Data Mining) décrit le data mining processus itératif complet constitué de 6 phases :

1. Compréhension du métier
2. Compréhension des données
3. Préparation des données
4. Modélisation

5. Evaluation de la modélisation
6. Déploiement des résultats obtenus

Compréhension du métier

Cette phase consiste à :

- Énoncer clairement les objectifs globaux du projet et les contraintes de l'entreprise.
- Traduire ces objectifs et ces contraintes en un problème de data mining.
- Préparer une stratégie initiale pour atteindre ces objectifs.

Compréhension des données

Cette phase consiste à :

- Recueillir les données.
- Utiliser l'analyse exploratoire pour se familiariser avec les données, commencer à les comprendre et imaginer ce qu'on pourrait en tirer comme connaissance.
- Évaluer la qualité des données
- Éventuellement, sélectionner des sous-ensembles intéressants.

Préparation des données

Cette phase consiste à :

- Préparer, à partir des données brutes, l'ensemble final des données qui va être utilisé pour toutes les phases suivantes.
- Sélectionner les cas et les variables à analyser.
- Réaliser si nécessaire les transformations de certaines données.
- Réaliser si nécessaire la suppression de certaines données

Cette phase fait suite à la compréhension des données. Celle-ci a mis au jour les corrélations, les valeurs aberrantes, les valeurs manquantes : on peut donc faire la préparation.

Modélisation

Cette phase consiste à :

- Sélectionner les techniques de modélisation appropriées (souvent plusieurs techniques peuvent être utilisées pour le même problème).
- Calibrer les paramètres des techniques de modélisation choisies pour optimiser les résultats
- Éventuellement revoir la préparation des données pour l'adapter aux techniques utilisées.

Evaluation de la modélisation

Cette phase consiste à produire le rapport final :

- Pour chaque technique de modélisation utilisée, évaluer la qualité (la pertinence, la signification) des résultats obtenus.
- Déterminer si les résultats obtenus atteignent les objectifs globaux identifiés pendant la phase de compréhension du métier.
- Décider si on passe à la phase suivante (le déploiement) ou si on souhaite reprendre l'étude en complétant le jeu de données.

Déploiement des résultats obtenus

Cette phase est externe à l'analyse du data mining. Elle concerne le maître d'ouvrage. Cette phase consiste à :

- Prendre les décisions en conséquences des résultats de l'étude de data mining
- Préparer la collecte des informations futures pour permettre de vérifier la pertinence des décisions effectivement mis en oeuvre.

6. Les logiciels de data mining

Il existe de nombreux logiciels de statistiques et de data mining sur PC. Certains sont gratuits, d'autres sont payants. Certains sont mono-utilisateur. D'autres fonctionnent en architecture clients-serveur. Parmi les gros logiciels, on peut citer :

- Clementine de SPSS.
- Clementine est la solution de data mining la plus vendue dans le monde.
- Entreprise Miner de SAS. · Statistica Data Miner de StatSoft
- Insightful Miner de Insightful

  XL Miner (data mining sous excel) · ORACLE, comme d'autres SGBD, fournit des outils de data mining

Parmi les logiciels gratuits, on peut citer :

- TANAGRA, logiciel de data mining gratuit pour l'enseignement et la recherche.
- ORANGE, logiciel libre d'apprentissage et de data mining.

B. Les algorithmes de recherche des règles associatives :

Dans cette partie nous allons tout d'abord définir le terme « règle associative », ensuite présenter différents algorithmes proposés pour la recherche de ces règles.

Définition : Règle associative

Une règle associative est une implication de la forme « si X alors Y » ou plus formellement X implique Y, où X et Y sont des ensembles de produits. La signification intuitive est que si une transaction contient les produits X elle a tendance à contenir les produits Y.

« Règle de type X implique Y où X et Y sont des ensembles d'objets (souvent des produits), traduisant le fait que si les objets X sont présents dans une transaction, alors les objets Y le sont avec une certaine probabilité » .

Les mesures de validité d'une règle :

Une règle est plus ou moins vraie pour un certain pourcentage constituant le support de la règle. Ce dernier est une mesure indiquant le pourcentage de transactions qui vérifient une règle associative. Le support est calculé par la formule suivante : Support (X implique Y) =|X & Y |/|BD|, en désignant par |X|le nombre de transactions comportant l'ensemble X et par |BD|le nombre total de transactions dans la base. Cependant, l'expérience a montré que le support n'est pas suffisant pour évaluer la force d'une règle, par exemple lorsqu'on achète une brosse à dents, on achète très souvent du dentifrice. Mais, on change rarement de brosse à dents ; donc la règle Brosse à dents implique Dentifrice a un faible support.

Pour mieux mesurer la validité d'une règle, la notion de confiance a été introduite. Pour une règle X implique Y, une confiance correspond à la probabilité conditionnelle P(Y/X), qui se calcul à partir du support comme suit :

Confiance (X implique Y) = Support (XY) / Support (X) et on ne retient que les règles satisfaisantes, après un filtrage sur la confiance. Plus formellement : « La confiance d'une règle (Rule Confidence) est une mesure indiquant le pourcentage de transaction qui vérifient la conclusion d'une règle associative parmi celle qui vérifient la prémisse »

Les algorithmes de recherche des règles associatives

Le problème qui se pose maintenant, est comment extraire les règles intéressantes ?

Les algorithmes actuels se basent sur la recherche des règles satisfaisantes aux besoins des utilisateurs et de trouver des règles intéressantes et cela est fait par calcul des supports de tous les ensembles fréquents. Nous allons présenter trois algorithmes :

- Algorithme Apriori.
- Algorithme Apriori TID.
- Algorithme BITMAP.

1. Algorithme Apriori :

L'algorithme Apriori s'applique à des bases de données contenant des transactions. De cet ensemble de transactions, Apriori est capable de sortir des règles liant les transactions entre elles.

L'algorithme procède de manière itérative : il commence par trouver les 1-itemset fréquents, c'est à dire les produits qui sont souvent achetés. A partir de cet ensemble, il peut déterminer les 2-itemset fréquents, c'est à dire les ensembles de deux produits qui sont souvent achetés : car bien sûr pour que deux produits soient souvent achetés ensembles, il faut que chaque produit du 2-itemset soit souvent acheté.

L'algorithme continue jusqu'à ce qu'il ne trouve plus d'ensemble de produits de taille K fréquemment achetés.

L'algorithme peut être donc divisé en trois phases :

1. Phase 1 : Générer un nouvel ensemble candidat (itemset) à partir de l'ensemble exploitable.
2. Phase 2 : Parcourir la base de donnée afin de déterminer le support des candidats et retirer de l'ensemble précédemment généré, ceux qui n'interviennent pas dans les transactions.
3. Phase 3 : Retirer des ensembles candidats, celui de support minimum, de manière à converger.

Exemple

L'exemple comprend un ensemble de transactions identifiées par leurs identifiants (TID). Chaque ensemble d'items appartient à une transaction. L'exemple porte sur les transactions dans un supermarché. Chaque transaction correspond donc à une transaction financière et par conséquent les items sont la liste des courses. Nous choisissons de fixer minimum support (minsup) à 50%.

TID	Items
100	1, 3, 4
200	2, 3, 5
300	1, 2, 3, 5
400	2, 5

La première itération :

Phase1 :

1-Itemset
{1}
{2}
{3}
{4}
{5}

Phase 2 :

1-Itemset	Support
{1}	2
{2}	3
{3}	3
{4}	~~1~~ (va être supprimer car 1<2 Phase3)
{5}	3

Itération 1

La deuxième itération :

Phase 1 :

2-Itemset
{1,2}
{1,3}
{1,5}
{2,3}
{2,5}
{3,5}

Phase 2 :

2-Itemset	Support
~~{1,2}~~	~~1~~ (Phase 3)
{1,3}	2
~~{1,5}~~	~~1~~ (Phase 3)
{2,3}	2
{2,5}	3
{3,5}	2

Itération2

Troisième itération :

Phase 1 :

3-Itemset
{2, 3, 5}

Phase 2 :

2-Itemset	Support
{2, 3, 5}	2

Itération3

L'algorithme s'arrête à ce niveau. Nous constatons finalement, que généralement quand un client achète les produits 2 et 3, il a tendance à acheter le produit 5.

L'algorithme Apriori repose donc sur l'idée principale que le sous ensembles d'un ensemble fréquent doit être fréquent. Ainsi l'astuce de génération d'un espace candidat à partir d'un espace précédent et non à partir des transactions de la base de donnée, qui fait de cet algorithme un algorithme correct de DataMining. Cependant ce n'est pas le meilleur puisqu'une version améliorée existe : Apriori TID.

2. Algorithme Apriori TID

Est une amélioration de l'algorithme Apriori, consiste à éviter les n passes à la base de donnée pour la génération des niéme –ensembles fréquents, pour cela une optimisation possible d'Apriori a permis de générer à un seul accès à la base de donnée tous les ensembles fréquents qui peuvent exister.

Cette optimisation consiste à garder en mémoire dans un premier lieu la liste des identifiants de tout les 1-ensembles fréquents trouver lors de la première (et la seule passe) à la base de donnée. Par la suite la génération des 2-ensembles fréquents est réalisée à partir des 1-ensembles fréquents mais le comptage est fait en utilisant la liste des identifiants d'un ensemble fréquent sans accéder une autre fois à la base de donnée.

Dans le cas général le calcul d'un k-ensemble se fait toujours à partir des deux (k-1)-ensembles contenant un élément de moins mais le comptage se fait simplement par intersection des deux (k-1)-ensembles source.

Exemple

Reprenons le même exemple.

TID	ITEMS
100	1, 3, 4
200	2, 3, 5
300	1, 2, 3, 5
400	2, 5

$\hat{C}_1$

TID	Set of ItemSets
100	{{1}, {3}, {4}}
200	{{2}, {3}, {5}}
300	{{1}, {2}, {3}, {5}}
400	{{2}, {5}}

L_1

ItemSet	Support
{1}	2
{2}	3
{3}	3
{5}	3

itération 1

C_2

ItemSet
{1,2}
{1,3}
{1,5}
{2,3}
{2,5}
{3,5}

$\hat{C}_2$

TID	Set of ItemSets
100	{{1,3}}
200	{{2,3}, {2,5}, {3,5}}
300	{{1,2}, {1,3}, {1,5}, {2,3}, {2,5}, {3,5}}
400	{{2,5}}

L_2

Itemset	Support
{1,3}	2
{2,3}	2
{2,5}	3
{3,5}	2

itération 2

C_3

Itemset
{2, 3, 5}

$\hat{C}_3$

TID	Set of Itemsets
200	{{2, 3, 5}}
400	{{2, 3, 5}}

L_3

Itemset	Set of Itemsets
{2, 3, 5}	2

itération 3

3. Algorithme Bitmap

Est un algorithme inspiré de l'algorithme Apriori TID. Il se base sur une indexation en Bitmap des différentes transactions de la base à la place des listes des identifiants.

« Un index Bitmap est un tableau de bits de dimension n*p avec n le nombre de lignes de la table indexé et le p le nombre de valeurs possibles pour l'attribut indexé. Le bit (i,j) est à 1 si la ligne i possède la valeur Pj correspondant à la colonne j du tableau »

L'utilité de l'indexation Bitmap s'avère lors du calcul de nombre d'occurrences des composants dans la base qui est obtenu directement à partir de comptage de nombre de bits dans l'intersection des vecteurs du Bitmap associé aux composant.

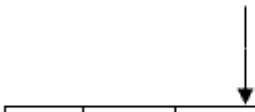
Le but de cet algorithme est de générer un k-ensemble fréquent à partir de deux (k-1) ensemble fréquents en utilisant la table d'index en Bitmap.

Exemple

Reprenons le même exemple

1	P1, P3, P4
2	P2, P3, P5
3	P1, P2, P3, P5
4	P2, P5

Indexation



	P1	P2	P3	P4	P5
1	1	0	1	1	0
2	0	1	1	0	1
3	1	1	1	0	1
4	0	1	0	0	1

Indexation Bitmap

Génération des 1-ensemble fréquent = garder que les produits qui figure dans plus de 2 transactions.

1-ensemble fréquent = {{P1}, {P2}, {P3}, {P5}}

Génération de 2-ensemble fréquent à partir de deux 1-ensemble fréquent

compter (P1, P2). Nouveautidbit = 1010.....0 0

Compte = 2 >= sup donc {P1, P2} forme un 2-ensemble fréquent.

Nous constatons que les algorithmes citées ci-dessus permettent tous d'étudier un certain nombre d'informations ou de données (des enregistrements d'une table....etc.), tel que chacun suit sa propre procédure afin d'atteindre le même objectif, qui consiste à déterminer les relations qui existent entre ces données.

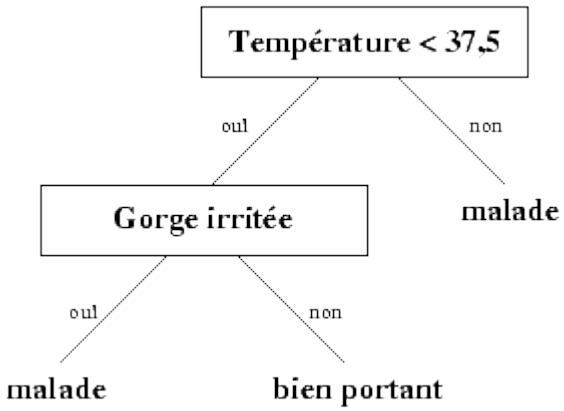
A la fin de cette présentation des différents outils d'analyse, nous pouvons remarquer que les outils d'aide à la décision, OLAP laissent l'initiative à l'utilisateur de choisir les éléments qu'il veut observer ou analyser, par contre dans le Data Mining, c'est le système qui a l'initiative et découvre lui-même les associations entre données, sans que l'utilisateur ait à lui dire de rechercher plutôt dans telle ou telle direction ou à poser des hypothèses.

C. Arbres de décision

Pour certains domaines d'application, il est essentiel de produire des procédures de classification compréhensibles par l'utilisateur. C'est en particulier le cas pour l'aide au diagnostic médical où le médecin doit pouvoir interpréter les raisons du diagnostic. Les arbres de décision répondent à cette contrainte car ils représentent graphiquement un ensemble de règles et sont aisément interprétables. Pour les arbres de grande taille, la procédure globale peut être difficile à appréhender, cependant, la classification d'un élément particulier est toujours compréhensible. Les algorithmes d'apprentissage par arbres de décision sont efficaces, disponibles dans la plupart des environnements de fouille de données

Exemple

La population est constituée d'un ensemble de patients. Il y a deux classes : malade et bien portant. Les descriptions sont faites avec les deux attributs : Température qui est un attribut à valeurs décimales et gorge irritée qui est un attribut logique. On considère l'arbre de décision de la figure ci dessous



Arbre de décision

Un arbre de décision est un arbre au sens informatique du terme. On rappelle que les noeuds d'un arbre sont repérés par des positions qui sont des mots sur {1,...,p}*, où p est l'arité maximale des noeuds. Si on note le mot vide par ε, les positions pour l'arbre de la figure ci dessus sont :

ε étiquetée par le test Température<37,5

1 étiquetée par le test Gorge irritée,

2 étiquetée par la feuille malade,

11 étiquetée par la feuille malade,

12 étiquetée par la feuille bien portant.

Les noeuds internes sont appelés noeuds de décision. Un tel noeud est étiqueté par un test qui peut être appliqué à toute description d'un individu de la population. En général, chaque test examine la valeur d'un unique attribut de l'espace des descriptions. Les réponses possibles au test correspondent aux labels des arcs issus de ce noeud. Dans le cas de noeuds de décision binaires, les labels des arcs sont omis et, par convention, l'arc gauche correspond à une réponse positive au test. Les feuilles sont étiquetées par une classe appelée classe par défaut.

Un arbre de décision est la représentation graphique d'une procédure de classification. En effet, à toute description complète est associée une seule feuille de l'arbre de décision. Cette association est définie en commençant à la racine de l'arbre et en descendant dans l'arbre selon les réponses aux tests qui étiquettent les noeuds internes. La classe associée est alors la classe par défaut associée à la feuille qui correspond à la description. La procédure de classification obtenue a une traduction immédiate en terme de règles de décision. Les systèmes de règles obtenus sont particuliers car l'ordre dans lequel on examine les attributs est fixé et les règles de décision sont mutuellement exclusives.

Exemple

Soit l'arbre de décision de la figure précédente. Un patient ayant une température de 39 et ayant la gorge non irritée sera classé comme malade par cet arbre. La traduction de cet arbre en règles de décision est :

SI Température<37,5 ET gorge irritée ALORS malade

SI Température<37,5 ET NON(gorge irritée) ALORS bien portant

SI NON(Température<37,5) ALORS malade

Les algorithmes d'apprentissage par arbres de décision, sont des algorithmes qui, prenant en entrée un échantillon S, construisent un arbre de décision. Nous allons, tout d'abord, introduire quelques notations. Étant donnés un échantillon S, un ensemble de classes {1,...,c} et un arbre de décision t, à chaque position p de t correspond un sous-ensemble de l'échantillon qui est l'ensemble des exemples qui satisfont les tests de la racine jusqu'à cette position. Par conséquent, on peut définir, pour toute position p de t, les quantités suivantes :

N(p) est le cardinal de l'ensemble des exemples associé à p,

N(k/p) est le cardinal de l'ensemble des exemples associé à p qui sont de classe k,

P(k/p) = N(k/p)/N(p) la proportion d'éléments de classe k à la position p.

On considère l'arbre de décision de la figure précédente. De plus, on dispose d'un échantillon de 200 patients. On sait que 100 sont malades et 100 sont bien portants, la répartition entre les deux classes M (pour malade) et S (pour bien portant) est donnée par :

	gorge irritée	gorge non irritée
température< 37,5	(6 S, 37 M)	(91 S, 1 M)
température>= 37,5	(2 S, 21 M)	(1 S, 41 M)

On a alors : N(11)=43 ; N(S/11)=6 ; N(M/11)=37 ; P(S/11)=6/43 et P(M/11)=37/43.

Exemple

Nous allons considérer l'exemple très simple suivant pour introduire les algorithmes d'apprentissage par arbres de décision. Une banque dispose des informations suivantes sur un ensemble de clients:

Client	M	A	R	E	I
1	moyen	moyen	village	oui	oui
2	élevé	moyen	grande ville	non	non
3	faible	âgé	grande ville	non	non
4	faible	moyen	grande ville	oui	oui
5	moyen	jeune	ville	oui	oui
6	élevé	âgé	ville	oui	non
7	moyen	âgé	ville	oui	non
8	faible	moyen	village	non	non

L'attribut ternaire M décrit la moyenne des montants sur le compte client. Le second attribut ternaire A donne la tranche d'âge du client. Le troisième attribut ternaire R décrit la localité de résidence du client. Le dernier attribut binaire E a la valeur oui si le client a un niveau d'études supérieures. La classe associée à chacun de ces clients correspond au contenu de la colonne I. La classe oui correspond à un client qui effectue une consultation de ses comptes bancaires en utilisant Internet.

On souhaite trouver un arbre de décision qui soit capable de dire si un client effectue des consultations de ses comptes par Internet en connaissant les valeurs des attributs M (montant), A (âge), R (résidence) et E (études) pour ce client.

A partir de ce tableau, il s'agit donc de construire un arbre de décision qui classifie

les clients. Les algorithmes construisent les arbres de façon descendante. Lorsqu'un test est choisi, on divise l'ensemble d'apprentissage pour chacune des branches et on réapplique récursivement l'algorithme. Sur notre exemple, on initialise avec l'arbre vide. L'échantillon contient 8 éléments, 3 sont de classe oui et 5 de classe non. Donc, à la racine de l'arbre qui n'est étiqueté par aucun test, l'échantillon peut être caractérisé par le couple (3,5).

On se pose alors la question de savoir si ce noeud est terminal, c'est-à-dire encore s'il est nécessaire de rechercher un test qui discrimine de façon intéressante l'échantillon. Par exemple, on attribuerait une feuille si nous étions dans le cas (0,8), c'est-à-dire si aucun client n'utilise Internet. Pour notre cas supposons que nous devions choisir un test. Nous aurions quatre choix possibles qui sont décrits comme suit :

(3,5) → M (1,2) (2,1) (0,2)

(3,5) → A (1,0) (2,2) (0,3)

(3,5) → R (1,1) (1,2) (1,2)

(3,5) → E (3,2) (0,3)

Laquelle des quatre possibilités faut-il choisir ? Si on regarde le test sur le type de résidence R, on remarque que ce test ne permet une discrimination sur aucune des branches, on peut donc se dire que le choix de ce test ne fait rien gagner, il sera donc à rejeter. Par contre, pour le test sur l'âge A, on remarque que sur la première branche, tous les éléments correspondants de l'échantillon sont de classe oui et que sur la troisième branche, tous les éléments sont de classe non. Ce test peut donc être considéré comme << intéressant >>. Ce raisonnement informel doit être automatisé.

Par conséquent, il nous faut introduire des quantités qui permettent de comparer les différents choix possibles. Dans ce but, on définit des fonctions qui permettent de mesurer le degré de mélange des exemples entre les différentes classes. Une telle fonction doit vérifier la propriété suivante : elle doit prendre son minimum lorsque tous les exemples sont dans une même classe (le noeud est pur) et son maximum lorsque les exemples sont équirépartis. Par exemple, si on dispose de 8 éléments et de deux classes, une telle fonction devra prendre son minimum pour les couples (0,8) et (8,0) et son maximum pour le couple (4,4). Il existe différentes fonctions qui satisfont ces propriétés, nous en citerons deux : la fonction de Gini et la fonction entropie. Soit S un échantillon, soit p une position, en reprenant les notations définies précédemment, ces fonctions sont définies par :

$$Entropie(p) = -\Sigma_{k=1}^{c} P(k/p) \times \log(P(k/p))$$

$$\begin{aligned} Gini(p) &= 1 - \Sigma_{k=1}^{c} P(k/p)^2 \\ &= 2 \Sigma_{k<k'} P(k/p)P(k'/p) \end{aligned}$$

Formules

Considérons le cas de deux classes et appelons x la proportion d'éléments de classe 1 en position p. On a donc Entropie(p)= - x log x - (1-x) log (1-x). Cette fonction de x prend ses valeurs dans l'intervalle [0,1], a son minimum pour x=0 et x=1 qui vaut 0 et a son maximum pour x=1/2 qui vaut 1. La fonction de Gini est définie par Gini(p) = 2x(1-x). Cette fonction de x prend ses valeurs dans l'intervalle [0,1/2], a son minimum pour x=0 et x=1 qui vaut 0 et a son maximum pour x=1/2 qui vaut 1/2. Ces deux fonctions sont symétriques par rapport à x=1/2. Pour notre exemple courant, considérons, par exemple, l'arbre construit à l'aide de l'attribut E, nous avons :

$$Entropie(\varepsilon) = -3/8 \log 3/8 - 5/8 \log 5/8 \simeq 0{,}954$$
$$Entropie(1) = -3/5 \log 3/5 - 2/5 \log 2/5 \simeq 0.970$$
$$Entropie(2) = -0/3 \log 0/3 - 3/3 \log 3/3 = 0$$
$$Gini(\varepsilon) = 2 \times 3/8 \times 5/8 \simeq 0{,}469$$
$$Gini(1) = 2 \times 3/5 \times 2/5 = 0.480$$
$$Gini(2) = 2 \times 0/3 \times 3/3 = 0$$

Formules appliquées à l'exemple

On dispose ainsi de fonctions permettant de mesurer le degré de mélange des classes pour tout échantillon et donc pour toute position de l'arbre en construction. Appelons i la fonction choisie. Il reste à définir une fonction permettant de choisir le test qui doit étiqueter le noeud courant. Rappelons que, sur notre exemple, à la racine de l'arbre, il nous faut choisir entre les quatre tests correspondants aux quatre attributs disponibles. Dans ce but, on introduit une fonction gain par :

$$Gain(p,t) = i(p) - \sum_{j=1}^{n} P_j \times i(pj)$$

Gain

où p désigne une position, test un test d'arité n et Pj est la proportion d'éléments de S à la position p qui vont en position pj (qui satisfont la jème branche du test test). Si on considère comme fonction i la fonction entropie, le terme i(p) représente l'entropie actuelle du noeud p, le deuxième terme de la différence représente l'entropie espérée en introduisant le test test qui est égale à la somme pondérée des entropies des nouveaux noeuds créés. On souhaite obtenir des entropies les plus faibles possibles car, d'après les propriétés de la fonction entropie, si l'entropie est faible, la plupart des éléments se trouvent dans une même classe. On cherche donc à obtenir le gain maximum. Sur notre exemple, nous obtenons :

$$Gain(\varepsilon,M) = Entropie(\varepsilon) - (3/8\ Entropie(1) + 3/8\ Entropie(2) + 2/8\ Entropie(3)) = Entropie(\varepsilon) - 0{,}620$$
$$Gain(\varepsilon,A) = Entropie(\varepsilon) - (1/8\ Entropie(1) + 4/8\ Entropie(2) + 3/8\ Entropie(3)) = Entropie(\varepsilon) - 0{,}500$$
$$Gain(\varepsilon,R) = Entropie(\varepsilon) - (2/8\ Entropie(1) + 3/8\ Entropie(2) + 3/8\ Entropie(3)) = Entropie(\varepsilon) - 0{,}870$$
$$Gain(\varepsilon,E) = Entropie(\varepsilon) - (5/8\ Entropie(1) + 3/8\ Entropie(2)) = Entropie(\varepsilon) - 0{,}607$$

Gains dans l'exemple

Le gain maximal ou encore l'entropie espérée minimale est obtenue pour le choix du test A. On remarque que le choix du test R est très mauvais, ce qui correspond bien à l'intuition. Dans ce paragraphe, nous avons introduit la problématique et quelques éléments fondamentaux utilisés par les algorithmes d'apprentissage par arbre de décision. Parmi les algorithmes connus dans ce domaine nous citons CART et ID3.

D. Détection de clusters

La détection automatique de clusters est une méthode de découverte de connaissances non dirigée. Cette technique est très puissantes, elle peut être utilisé dans de très nombreux secteurs d'activités. En effet, celle-ci consiste à procéder à une classification du type regroupement par similitude. A la suite de ces différents regroupements, on nomme chacun des groupes cluster.

Une utilisation classique consiste à clusteriser une population puis, après étude de chaque cluster, faire une offre commerciale tout à fait adaptée à la population

il existe deux grandes méthodes:

- la méthode des K-moyennes
- les méthodes par agglomération

Pour utiliser une des méthodes de détection de clusters, il faut prévoir une fonction de distance qui mesure l'écart entre deux enregistrements. Nous allons expliciter ci-après les deux méthodes proposées pour la détection automatique de cluster.

Méthode des K-moyennes

Cette méthodes des K-moyennes permet de diviser une population donnée avec ces caractéristiques en K groupes appelés clusters. Le nombres K de clusters est déterminé par l'utilisateurs selon ses attentes. Principe de fonctionnement : Après avoir déterminé un nombre K de clusters on positionne les K premiers points (appelés graines) au hasard (on utilise en général les K premiers enregistrements). Chaque enregistrement est affecté à la graine dont il est le plus proche (en utilisant la fonction de distance). A la fin de la première affectation, la valeur moyenne de chaque cluster est calculée et la graine prend cette nouvelle valeur. Le processus est répété jusqu'à stabilisation des clusters.

Méthode par agglomération

Cette technique de clusterisation par agglomération permet de créer un arbre. Cet arbre se différencie des arbres de décisions du fait qu'on applique le cheminement contraire, c'est à dire que nous commençons des feuilles afin d'arriver à la racine.

Principe de fonctionnement : On mesure les distances de tous les éléments entre eux, puis on regroupe ceux qui sont les plus proches. On calcule le centroïde de chaque groupe et on recommence jusqu'à ce que tous les éléments soient reliés

Fonction de distance

Il s'agit d'une mesure de la distance d'un nouvel item avec ceux contenus ans la base de connaissance. De nombreuses formules sont tolérées. Voici les critères qu'elle doit respecter pour être accepter en tant que fonction de distance :

- La fonction de distance doit être positive
- La distance d'un point à lui-même est nulle
- Commutativité de la fonction distance
- Raccourcir la distance de A à B en passant par un point C n'est pas possible

Pour les données numériques, les trois fonctions de distance les plus courantes

entre une valeur A et une valeur B sont :

- La valeur absolue de la différence : |A-B|
- Le carré de la différence : |(A – B) ²|
- La valeur absolue normalisée : | A – B | / (différence maximale)

Pour les autres types de données, c'est à l'utilisateur de définir sa propre fonction de distance. Par exemple, pour comparer le sexe d'un individu, on pourra affecter la valeur 1 s'ils sont de sexe différent ou la valeur 0 s'ils sont identiques. Pour une catégorie socioprofessionnelle, il suffit de créer une métrique. Pour des communes, pourquoi ne pas prendre la distance entre elles ou affecter une codification en fonction du type (urbaine, périurbaine, rurale) ou de la région. Il est toujours préférable de faire une codification dont le résultat se situera entre 0 et 1.

Exemple

Pour illustrer cette technique nous avons choisi d'utiliser un exemple. Pour en faciliter la compréhension celui-ci sera très simple.

Enoncé : classifier des individus selon leurs âges. Soit une liste aléatoire d'individus dont les âges sont les suivants :

27 - 51 - 52 - 33 - 45 - 22 - 28 - 44 - 40 - 38 - 20 - 57

1. Méthode des K-moyennes

Nous fixons ici K à 3. Comme expliqué auparavant, les trois premières graines prennent donc les trois premières valeurs. Nous allons maintenant calculer la distance : distance = différence / (amplitude maximum) L'amplitude maximum correspond à l'écart maximum entre deux valeurs de l'âge des individu. Ici l'amplitude maximum vaut 37 et correspond l'écart entre 20 et 57. La distance est calculé entre chaque graines (27, 51, 52) et tous les autres points de la liste. Cela va permettre d'affecter chaque points aux différentes graines. Nous obtenons le tableau suivant :

	27	51	52	33	45	22	28	44	40	38	20	57
Graine 27	0.00	0.65	0.68	0.16	0.49	0.14	0.03	0.46	0.35	0.30	0.19	0.81
Graine 51	0.65	0.00	0.03	0.49	0.16	0.78	0.62	0.19	0.30	0.35	0.84	0.16
Graine 52	0.68	0.03	0.00	0.51	0.19	0.81	0.65	0.22	0.32	0.38	0.86	0.14
Minimum	0	0	0	0.16	0.16	0.14	0.03	0.19	0.3	0.3	0.19	0.14
Affectation	1	2	3	1	2	1	1	2	2	1	1	3

calcul des distances entre chaque graine et chaque point

Il apparaît clairement après cette première affectation les associations suivantes :

Graine 1 (27) : 27 - 33 - 22 - 28 - 38 - 20

Graine 2 (51) : 51 - 45 - 44 - 40

Graine 3 (52) : 52 - 57

Pour obtenir les nouveaux centroïdes nous prenons la moyenne arithmétique de chaque groupes ou clusters. Nous avons pour cet exemple les valeurs suivantes :

graine 1 : 28

graine 2 : 45

graine 3 : 54.5

Nous réitérons le processus précédent à ces nouvelles valeurs, ce qui nous permet d'obtenir le tableau suivant :

	27	51	52	33	45	22	28	44	40	38	20	57
Graine 28	0.03	0.62	0.65	0.14	0.46	0.16	0	0.43	0.32	0.27	0.22	0.78
Graine 45	0.49	0.16	0.19	0.32	0	0.62	0.46	0.03	0.14	0.19	0.68	0.32
Graine 54.5	0.74	0.09	0.07	0.58	0.26	0.88	0.72	0.28	0.39	0.45	0.93	0.07
Minimum	0.03	0.09	0.07	0.14	0	0.16	0	0.03	0.14	0.19	0.22	0.07
Affectation	1	3	3	1	2	1	1	2	2	2	1	3

calcul des distances entre chaque point et les nouvelles graines (centroïdes)

Il apparaît clairement après cette deuxième affectation les associations suivantes :

Graine 1 (28) : 27 - 33 - 22 - 28 - 20 Moyenne = 26

Graine 2 (45) : 45 - 44 - 40 - 38 Moyenne = 41.75

Graine 3 (54.5) : 51 - 52 - 57 Moyenne = 53.33

En réitérant une troisième fois le processus, nous voyons qu'il ne modifie plus les affectations. Les clusters sont donc finalisés :

Cluster 1: 27 - 33 - 22 - 28 - 20 Jeunes majeurs - Centroïde = 26

Cluster 2: 45 - 44 - 40 - 38 Quadragénaires - Centroïde = 41.75

Cluster 3: 51 - 52 - 57 Quinquagénaires - Centroïde = 53.33

2. AVANTAGES ET INCONVÉNIENTS

Les avantages de cette technique sont :

- résultats clairs
- technique facile à mettre en œuvre
- La méthode des K-moyennes ne consomme pas d'énorme de ressources
- application facile

Les inconvénients de cette technique sont :

- difficulté de trouver une bonne fonction de distance
- difficulté d'expliquer certains clusters