Laboratoire des Systèmes Informatiques
Université des sciences et de la Technologie Houari Boumediene
USTHB - Alger

ENTREPOTS et FOUILLE DE DONNEES

Pr. Kamel Boukhalfa
Boukhalk@gmail.com

Présentation de l'Enseignant

- **Pr. Kamel Boukhalfa**
  - Enseignant à l'USTHB
  - Docteur de l'université de Poitiers-France et de l'USTHB
  - Professeur depuis 2017
- **Emails**
  - Email de l'USTHB : kboukhalfa@usthb.dz
  - Emails souvent utilisés
    - boukhalk@gmail.com
    - Kamelboukhalfa@yahoo.fr
- **Site Web personnel**
  - http://boukhalfa.jimdo.com
  - CV
  - Production scientifique
  - Téléchargement de cours

Domaines de Recherches

- **Entrepôts de données**
  - Optimisation de la couche physique, Fragmentation Horizontale, Indexation, Tuning
- **Méta-heuristiques**
  - AG, RS, HC, RT, Optimisation multi-objectifs
- **Data mining**
  - Motifs fréquents, classification, règles d'association, data mining spatial
- **Cloud Computing**
  - DBAAS, Systèmes de stockage hybride,
- **Big Data**
  - Analyse des réseaux sociaux, influence, tri des filles d'actualité

PLAN GLOBAL

- Entrepôts de données
- Optimisation des Entrepôts
- Fouille de données
- Techniques de Fouille de données
- Big Data

Laboratoire des Systèmes Informatiques
Université des sciences et de la Technologie Houari Boumediene
USTHB - Alger

LES ENTREPOTS DE DONNEES
Data Warehouse

Pr. Kamel Boukhalfa
Boukhalk@gmail.com

PLAN

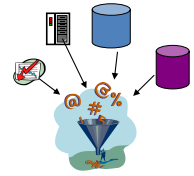
- Entrepôt de données
- Architecture des ED
- Modélisation multidimensionnelle OLAP
- Optimisation des Entrepôts de données

Le contexte

- **Besoin:** prise de décisions stratégiques et tactiques
- **Pourquoi:** besoin de réactivité
- **Qui:** les décideurs (non informaticiens)
- **Comment:** répondre aux demandes d'analyse des données, dégager des informations qualitatives nouvelles

Les données utilisables par les décideurs

- **Données opérationnelles (de production)**
  - Bases de données (Oracle, SQL Server)
  - Fichiers, …
  - Paye, gestion des RH, gestion des commandes…
- **Caractéristiques de ces données:**
  - **Distribuées**: systèmes éparpillés
  - **Hétérogènes**: systèmes et structures de données différents
  - **Détaillées**: organisation des données selon les processus fonctionnels, données surabondantes pour l'analyse
  - **Peu/pas adaptées à l'analyse** : les requêtes lourdes peuvent bloquer le système transactionnel
  - **Volatiles**: pas d'historisation systématique



Le Décisionnel

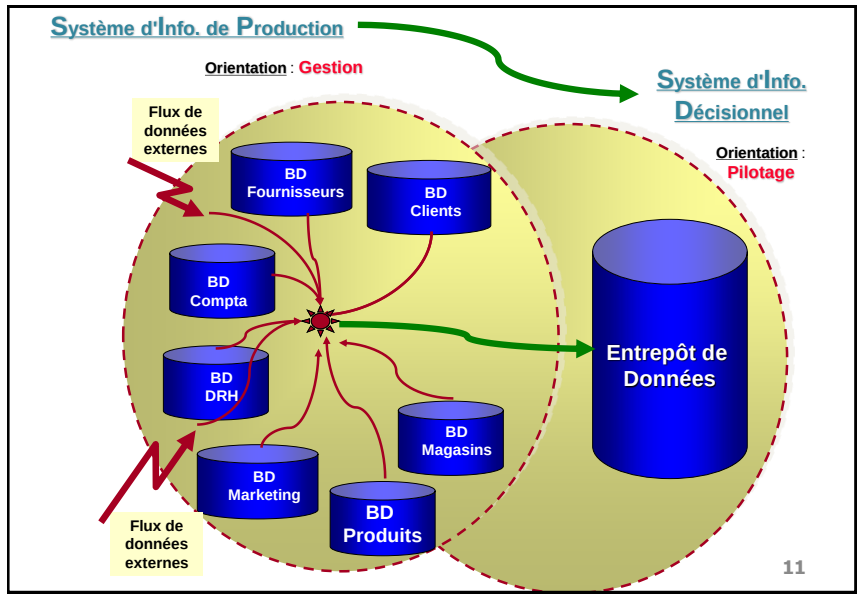
- Les entreprises passent à l'**ère de l'information**
- **Défi** : Transformer une partie de leur système d'information qui avait une vocation de production à un SI décisionnel dont la vocation de pilotage devient majeure.
- Un **système d'information décisionnel** (**S.I.D.**) est un ensemble de données organisé de façon spécifique, approprié à la prise de décision.
- Finalité d'un système décisionnel :
  - **Pilotage de l'entreprise**

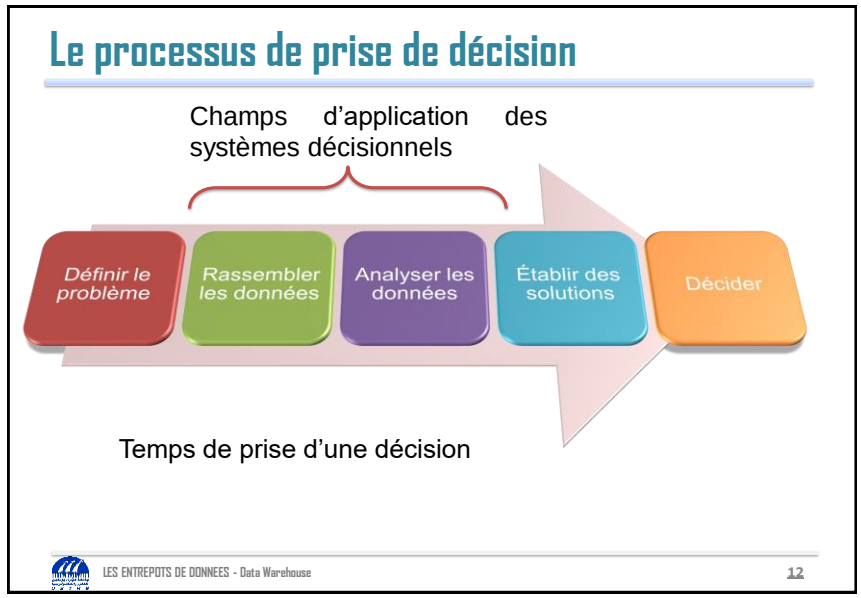
Problématique

- Comment répondre aux demandes des décideurs?
  - En donnant un accès rapide et simple à l'information stratégique
  - En donnant du sens aux données

➡ Mettre en place un système d'information dédié aux applications décisionnelles:

Un Entrepôt de Données (Data Warehouse)


Système d'Info. de Production
Orientation : Gestion
Système d'Info. Décisionnel
Orientation : Pilotage
Flux de données externes
BD Fournisseurs
BD Clients
BD Compta
BD DRH
BD Marketing
BD Produits
BD Magasins
Entrepôt de Données
Flux de données externes
11


Le processus de prise de décision
Champs d'application des systèmes décisionnels
Définir le problème
Rassembler les données
Analyser les données
Établir des solutions
Décider
Temps de prise d'une décision
LES ENTREPOTS DE DONNEES - Data Warehouse
12

Introduction

- Pourquoi le data warehouse ?
  - Améliorer les performances décisionnelles de l'entreprise
- Comment?
  - En répondant aux demandes d'analyse des décideurs
- Exemples
  - **Clientèle:** Qui sont mes clients? Pourquoi sont-ils mes clients?, Comment les conserver ou les faire revenir (préférence d'achat, habitudes, …)? Ces clients sont-ils vraiment intéressants pour moi?
  - **Marketing, actions commerciales**: où placer ce produit dans les rayons? Comment cibler plus précisément le mailing concernant ce produit?
  - **Exemple dans le domaine de sécurité ?**

Introduction

Raisons d'être d'un entrepôt de données

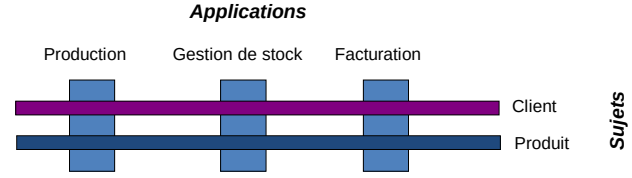
- Rassembler les données de l'entreprise dans un même lieu sans surcharger les BD (systèmes opérationnels)
- Permettre un accès universel à diverses sources de données et assurer la qualité des données
- Extraire, filtrer, et intégrer les informations pertinentes, à l'avance, pour des requêtes ultérieures
- Dégager des connaissances et faire un apprentissage sur l'entreprise, le marché et l'environnement

C'est quoi un entrepôt de données?

- Industrie (Inmon 1992)
  - Collection de données orientées sujets
  - Consolidées dans une base de données unique
  - Non volatiles et historisées variant dans le temps
  - organisées pour le support d'un processus d'aide à la décision
- Recherche (Stanford 1995)
  - Dispositif de stockage d'informations intégrées de sources distribuées, autonomes, hétérogènes

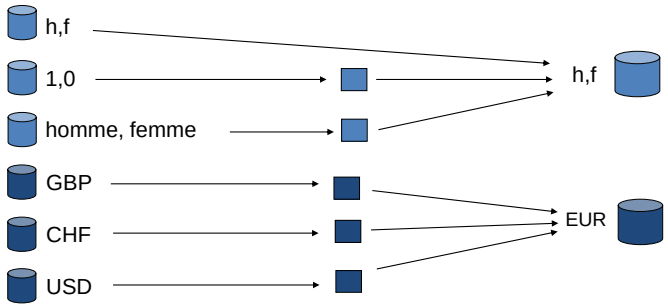
Les 4 caractéristiques des data warehouse

1. Données orientées sujet:
   - Regroupe les informations des différents métiers
   - Ne tiens pas compte de l'organisation fonctionnelle des données

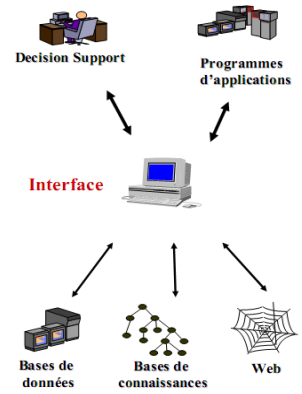


Les 4 caractéristiques des data warehouse

2. Données intégrées:
   - Normalisation des données
   - Définition d'un référentiel unique



Intégration de données



Caractéristiques des sources de données :

- ➔ Hétérogènes (schématique, sémantique),
- ➔ Autonomes
- ➔ Évolutives
- ➔ Réparties

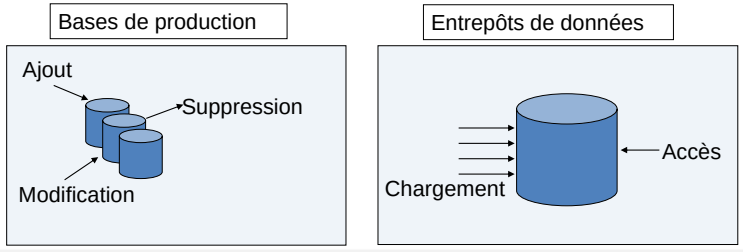
Besoins :

- ➔ Intégration données
- ➔ Gestion de l'évolution des données

Les 4 caractéristiques des data warehouse

3. Données non volatiles

- Traçabilité des informations et des décisions prises
- Copie des données de production



Les 4 caractéristiques des data warehouse

4. Données datées

- Les données persistent dans le temps
- Mise en place d'un référentiel temps

Base de production

Image de la base en Mai 2017

Répertoire

Nom	Ville
Karim	Alger
Salim	Blida

Image de la base en Juillet 2018

Répertoire

Nom	Ville
Karim	Tipaza
Salim	Blida

Entrepôt de données

Calendrier

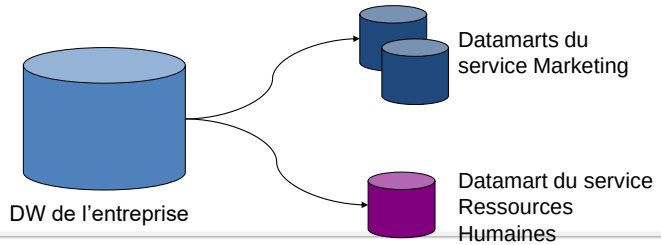
Code	Année	Mois
1	2017	Mai
2	2018	Juillet

Répertoire

Code	Nom	Ville
1	Karim	Alger
1	Salim	Blida
2	Karim	Tipaza

Datamart

- Sous-ensemble d'un entrepôt de données
- Destiné à répondre aux besoins d'un secteur ou d'une fonction particulière de l'entreprise
- Point de vue spécifique selon des critères métiers

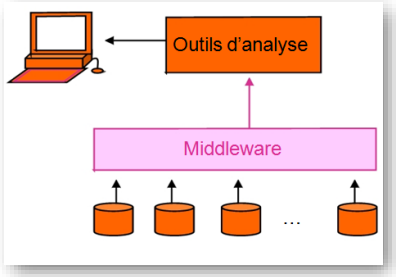


Intérêt des datamart

- Nouvel environnement structuré et formaté en fonction des besoins d'un métier ou d'un usage particulier
- Moins de données que DW
  - Plus facile à comprendre, à manipuler
  - Amélioration des temps de réponse
- Utilisateurs plus ciblés

Architecture d'un entrepôt de données

Approche virtuelle (ou le non-entrepôt)

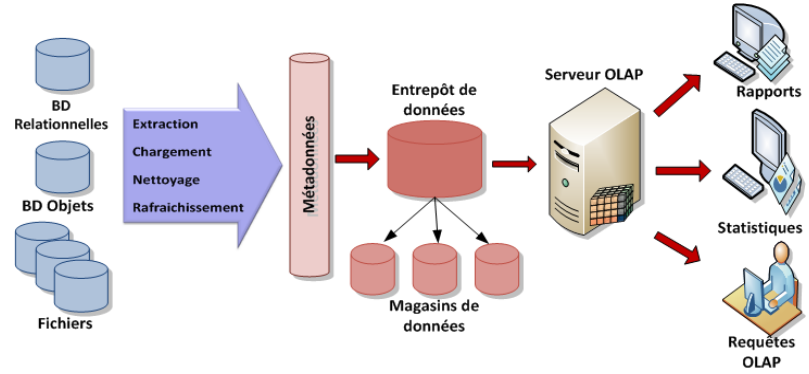


Inconvénients

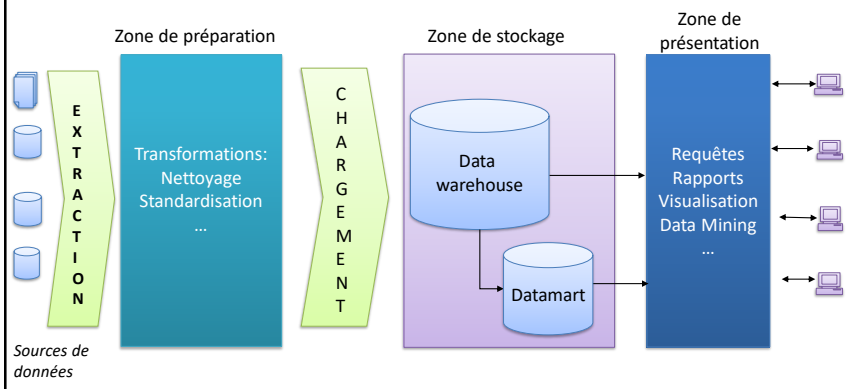
- ❑ pas de réelle intégration des données
- ❑ différentes vues non-réconciliées
- ❑ les requêtes peuvent facilement bloquer les transactions en cours

Architecture d'un entrepôt de données: physique

Approche entrepôt



Architecture générale



Les flux de données

- **Flux entrant**
  - Extraction: multi-source, hétérogène
  - Transformation: filtrer, trier, homogénéiser, nettoyer
  - Chargement: insertion des données dans l'entrepôt
- **Flux sortant**
  - Mise à disposition des données pour les utilisateurs finaux

Les différentes zones de l'architecture

- **Zone de préparation (Staging area)**
  - Zone temporaire de stockage des données extraites
  - Réalisation des transformations avant l'insertion dans le DW:
    - Nettoyage
    - Normalisation…
  - Données souvent détruites après chargement dans le DW
- **Zone de stockage (DW, DM)**
  - On y transfère les données nettoyées
  - Stockage permanent des données
- **Zone de présentation**
  - Donne accès aux données contenues dans le DW
  - Peut contenir des outils d'analyse programmés:
    - Rapports
    - Requêtes…

Processus ETL (Extracting – Transforming – Loading)

☑ Le principe de l'entreposage des données est de rassembler de multiples données sources qui souvent sont hétérogènes en les rendant *homogènes* afin de les analyser.

☑ Ce travail d'homogéinisation nécessite des **règles** précises servant de **dictionnaire** (ou de **référentiel**) et qui seront mémorisées sous forme de **métadonnées** (information sur les données).

☑ Ces règles permettent d'assurer des tâches d'administration et de gestion des donnés entreposées.

ETL

Un système ETL est tout système qui permet :

- d'offrir un environnement de développement, des outils de gestion des opérations et de maintenance.
- de découvrir, analyser et extraire les données à partir de sources hétérogènes;
- de nettoyer et standardiser les données selon les règles d'affaires établies par l'entreprise;
- de charger les données dans un entrepôt de données dans et/ou les propager vers les data-marts.

ETL

L'alimentation d'un ED est un processus qui s'effectue en plusieurs étapes :

- Sélection des données sources
- Extraction des données
- Transformation
- Chargement

Sélection des données sources

- Quelles sont les données de production qu'il faut sélectionner pour alimenter l'ED ?
- Toutes les données sources ne sont forcément pas utiles.
  - Doit-on prendre l'adresse complète ou séparer le code postal ?
- Les données sélectionnées seront réorganisées pour servir à la fabrication **des informations.**
- La **dénormalisation** des données crée des liens entre les données et permet des accès différents

Sélection des données sources (suite)

La sélection des données utiles à partir des BD de production n'est pas simple à faire .

Les données sont :

- **hétérogènes** (**différents SGBD et différentes méthodes d'accès**);
- **diffuses** (**différents environnements matériels et différents réseaux interconnectés ou non**);
- **complexes** (**différents modèles logiques et physiques principalement orientés vers les traitements transactionnels**).
- La définition de la granularité dépend du niveau de raffinement de l'information qu'on veut obtenir.

Sélection des données sources (suite)

Il existe plusieurs niveaux de données :

- Les données sont parfois assemblées avant d'être injectées dans l'ED permettant une vision intégrée et transversale de l'entreprise.
- Cette forme de données constitue **le niveau le plus fin** au niveau de l'ED : ceux sont **les données de détail**. Elles peuvent être agrégées et constituent ainsi un autre niveau de détail.
- Elles seront par la suite **structurées** dans des espaces d'analyse (soit des **cubes de données**, soit des **data marts**).
- Elles seront finalement à **un niveau de présentation**, où elles peuvent avoir plusieurs formes (tableaux, graphiques, tableaux de bord, règles de connaissances...).

Extraction des données

- L'extraction peut se faire à travers un **outil d'alimentation** qui doit travailler de façon **native** avec les SGBD qui gèrent les données sources.
- Ou alors créer des **programmes extracteurs**. L'inconvénient de cette approche est le risque de faire des extractions erronées, incomplètes et qui peuvent biaiser l'ED.
- Il faut gérer les anomalies en les traitant et en gardant une trace

Extraction des données

- L'extraction doit se faire conformément aux règles précises du référentiel.
- Elle ne doit pas non plus perturber les activités de production.
- Il faut faire attention aux données cycliques. Celles qu'on doit calculer à chaque période, pour pouvoir les prendre en considération.
- L'extraction peut se faire en interne selon l'horloge interne ou par un planificateur ou par la détection d'une donnée cible (de l'ED) ; ou en externe par des planificateurs externes.
- Les données extraites doivent être marquées par *" horodatage "* afin qu'elles puissent être pistées.

Transformation

C'est une suite d'opérations qui a pour but de rendre les données cibles **homogènes** et puissent être traitées de façon **cohérente**.

Exemple

Donnés sources	données cibles	Donnés sources	données cibles
Appli 1 : male, femelle	m, f	*Appli 1* : $1 500	179121 DZD
Appli 2 : 1, 0	m, f	*Appli 2* : 160 €	21435 DZD
Appli 3 : Masculin, féminin	m, f	*Appli 3* : 2 000£	311950 DZD

Transformation

- Les données doivent alors filtrées afin d'éliminer les données aberrantes: données sans valeurs, avec des valeurs manquantes.
- Souvent dans les bases de production, certaines données sont sémantiquement fausses.
- Pour avoir une alimentation de qualité, il faut avoir une bonne connaissance des données à entreposer et des règles qui les régissent. Savoir corriger les données pour les doter d'un vrai sens sémantique.

Transformation

- L'ensemble des données sources, après nettoyage ou transformation d'après des règles précises ou par application de programmes, seront restructurées et converties dans un **format cible**.
- Il faut synchroniser les données pour que les valeurs agrégées obtenues soient cohérentes, avant de passer à la phase de chargement.

Data Cleaning

- Valeurs manquantes (nulles)
  - Ignorer le tuple
  - Remplacer par une valeur fixe ou par la moyenne
- Valeurs erronées ou inconsistantes
  - Générées en présence de bruits
  - Détecter par une analyse de voisinage
    - Écart par rapport à la moyenne
    - Factorisation en groupes
  - Remplacer par une valeur fixe ou par la moyenne
- Inspection manuelle de certaines données possible

Chargement

- C'est l'opération qui consiste à charger les données nettoyées et préparées dans le DW.
- C'est une opération qui risque d'être assez longue. Il faut mettre en place des stratégies pour assurer de bonnes conditions à sa réalisation et définir la politique de rafraîchissement.
- C'est une phase plutôt mécanique et la moins complexe.

Chargement

- Le dictionnaire (ou référentiel) de données est constitué de l'ensemble des métadonnées.
- Il renferme des informations sur toutes les données de l'ED.
- Il renferme également des informations sur chaque étape lors de la construction du DW ; sur le passage d'un niveau de données à un autre lors de l'exploitation du DW.
  - Le rôle des métadonnées est de permettre :
    - La définition des données
    - La fabrication des données
    - Le stockage des données
    - L'accès aux données
    - La présentation des données.

Catégories des systèmes d'ETL

Il existe trois catégories d'outils ETL :

1. ***Engine-based*** : les transformations sont exécutées sur un serveur ETL, disposant en général d'un référentiel. Ce genre d'outil dispose d'un moteur de transformation ;
2. ***Database-embedded*** : les transformations sont intégrées dans la BD ;
3. ***Code-generators*** : les transformations sont conçues et un code est généré. Ce code est déployable indépendamment de la base de données.

Catégories des systèmes d'ETL

Avantages des suites ETL :

✓ Développement simple, rapide et moins coûteux. Les coûts de l'outil seront amortis rapidement pour les projets sophistiqués ou de grandes envergures.

✓ Des ressources disposant de connaissances du domaine d'affaire et n'ayant pas de grandes compétences en programmation peuvent développer avec l'outil.

✓ Les outils ETL intègrent des référentiels de gestion des méta-data, tout en permettant de synchroniser les méta-data avec les systèmes sources, les BDs de l'ED et autres outils BI.

✓ Les outils ETL permettent la génération automatique du méta-data à chaque étape du processus ETL et renforcent la mise en place d'une méthodologie commune de gestion de méta-data qui doit être respectée par tous les développeurs.

✓ Les outils ETL disposent de programme intégré qui permet de faciliter la documentation, la création et la gestion de changement. L'outil ETL doit bien gérer les dépendances complexes et les erreurs qui peuvent surgir en cours d'exécution.

Catégories des systèmes d'ETL

Avantages des suites ETL (suite) :

✓ Le référentiel de méta-data des outils ETL peut produire automatiquement des rapports de mise en correspondance des données et d'analyse de dépendance de données

✓ Les outils ETL disposent de connecteurs intégrés pour la plupart des sources de données. Ils permettent aussi d'effectuer des conversions complexes de types de données (selon la source et la destination)

✓ Les outils ETL offrent des mécanismes de cryptage de compression en ligne de données

✓ La plupart des outils ETL offre une très bonne performance même pour une grande quantité de données.

✓ Un outil ETL peut, le cas échéant, gérer des scénarios d'équilibrage de la charge entre les serveurs.

✓Un outil ETL peut être complété ou amélioré en utilisant le scripting ou la programmation.

Catégories des systèmes d'ETL

Avantages des ETL-Maison :

✓Les outils de tests unitaires automatique sont disponibles seulement pour les outils développé maison.

✓Les techniques de programmation orientée objet permettent de rendre consistantes la gestion des erreurs, la validation et la mise à jour des méta-data.

✓Il est possible de gérer manuellement les méta-data dans le code et de créer des interfaces pour la gestion de ces dernières

✓.Disponibilité des programmeurs dans l'entreprise.

✓ Un outil ETL est limité aux capacités du fournisseur.

✓ Un outil ETL est limité à l'outil de scripting propriétaire.

✓Un outil développé maison donne une grande flexibilité et si le besoin se présente. Il est possible de tout faire.

Administration d'un ED

- L'ED est un aspect physique du SI de l'entreprise. Il doit être par conséquent évolutif. Les données doivent donc changer. On doit procéder à d'autres alimentations et donc gérer l'actualisation des données.
- Il existe des outils qui prennent en charge les tâches de rafraîchissement des données.
- Ils procèdent par réplication pour propager les maj effectuées dans les BD sources, dans l'ED.
- Le mécanisme de réplication et une opération de copie de données d'une BD vers une ou plusieurs BD.
- Les réplications sont alors synchrones ou asynchrones.
- Le rafraîchissement des données peut se faire également par des processus de transformation qui exploitent les méta-données.

Administration d'un ED (suite)

- La fonction d'administration porte sur un aspect fonctionnel (*qualité et la pérennité des données*) mais aussi sur un aspect technique (*maintenance, optimisation, sécurisation,...)*
- Elle concerne l'ensemble des tâches du processus d'entreposage de la sélection des données de production à la mise à disposition pour construire les espaces d'analyse.
- L'administrateur de l'ED doit maîtriser la gestion des données (*données, provenance des données, méta-données*).
- Les données agrégées sont aussi une production (*information*) de l'entreprise comme les données de production, doivent être entreposées.

Administration d'un ED (suite)

- La fonction de DBA est très recherchée
- Les DBA sont bien rémunérés (mieux que les développeurs)
- Les compétences demandées chez les DBA :
  - Data warehousing (très recherché)
  - Services de transformation des données (ETL)
  - Environnement de réplication

Rôles et responsabilités

1. Gestionnaire ETL

- Gérer quotidiennement l'équipe ETL.
- Définir les standards et procédures de l'environnement de développement ETL (Règles de nomenclature, Meilleures pratiques…)
- Superviser le développement, les tests et l'assurance qualité

2. Architecte ETL

- Concevoir l'architecture et l'infrastructure de l'environnement ETL.
- Concevoir le mappage logique de données.
- Livrer les routines ETL en production.
- Appréhender les besoins d'affaire.
- Connaître les systèmes source.
- Résoudre les problèmes techniques complexes.

Rôles et responsabilités (suite)

3. Développeur ETL

- Développer les routines ETL.
- Tester les routines ETL.
- S'assurer que les résultats du processus ETL répondent aux besoins d'affaire (Collaboration étroite avec l'architecte ETL)

4. Analyste système

- Rassembler des besoins d'affaire.
- Documenter les besoins d'affaire.
- Travailler en collaboration avec toute l'équipe du DW (Non seulement celle du système ETL).

Rôles et responsabilités (suite)

5.Spécialiste qualité de données

- S'assurer de la qualité des données dans l'entrepôt de données en entier.
- S'assurer que les règles d'affaire sont bien implantées par les processus ETL (en collaboration avec l'analyste système et l'architecte ETL)

6.DBA

- Installer, configurer, migrer et maintenir la base de données.
- Traduire le modèle logique de données en modèle physique.

Exploitation de l'entrepôt

- Business Intelligence:
  - Possibilité de visualiser et d'exploiter une masse importante de données complexes
- Trois principaux outils:
  - OLAP :On-Line Analytical Processing
  - Data mining: fouille de données
  - Formulation de requêtes et visualisation des résultats

Architecture d'un entrepôt de données

- Souvent une architecture trois-tiers
  - Serveur d'entrepôt ("Warehouse Database Server")
    - Très souvent un système relationnel (ex. Oracle)
  - Serveur OLAP ("OLAP Server") de type ROLAP, MOLAP, ou HOLAP
  - Clients
    - Outils de requêtes et de production de rapports
    - Outils d'analyse et de prospection de données

Domaines d'applications

- Banque, Assurance
  - Détermination des profils client (prêt, …)
- Commerce
  - Ciblage de clientèle
  - Compagnies de grande production
  - Aménagement des rayons (2 produits en corrélation)
- Compagnies téléphoniques
- Santé

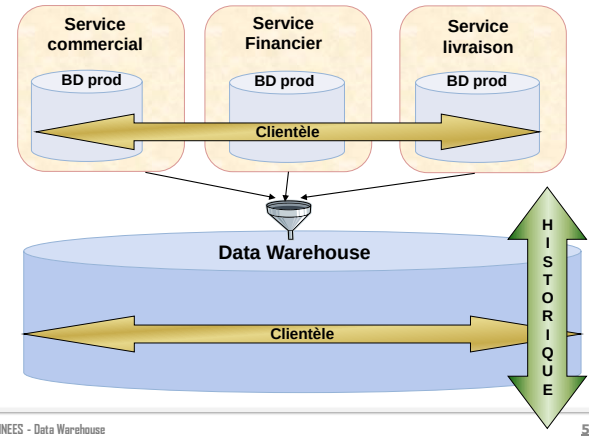
Base de données vs. Entrepôt de donnée

Pourquoi dissocier une BD d'un ED?

- Les objectifs de performances dans les BD ne sont pas les mêmes que ceux dans les EDs :
  - BD : requêtes simples (OLTP), méthodes d'accès et indexation
  - ED : requêtes OLAP souvent complexes!!!
- La nécessité de combiner des données provenant de diverses sources, d'effectuer des agrégations dans un ED et d'offrir des vues multidimensionnelles
- Les données d'un ED sont souvent non volatiles et ont donc une plus longue durée de vie que celles d'une BD

SGBD et DW

OLTP: On-Line Transactional Processing

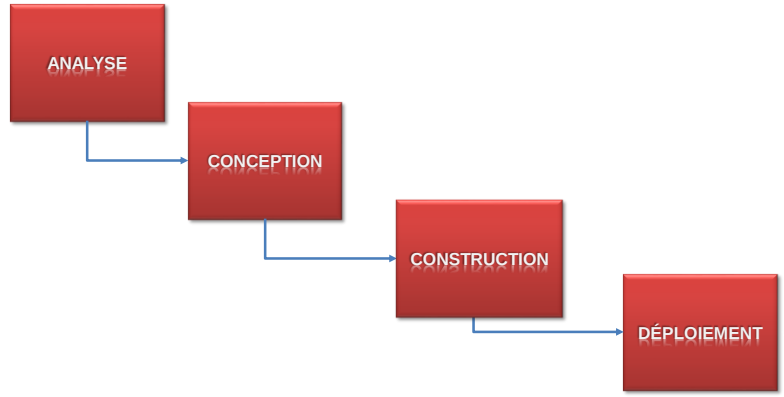


OLAP: On-Line Analitical Processing

OLTP VS DW

OLTP	DW
Orienté transaction	Orienté analyse
Orienté application	Orienté sujet
Données courantes	Données historisées
Données détaillées	Données agrégées
Données évolutives	Données statiques
Utilisateurs nombreux, administrateurs/opérationnels	Utilisateurs peu nombreux, manager
Temps d'exécution: court	Temps d'exécution: long

Cycle de vie de l'entrepôt de données



Cycle de vie

❑ Spécification des besoins

- Rassembler clairement et fidèlement les besoins des utilisateurs (décideurs)
- Clarifier les objectifs spécifiques
  - Comportement de la clientèle, analyse de tendances de prévisions, etc.
- Énumérer les dimensions
- Définir l'architecture du système (modèle de données), l'usage final (rapports, requêtes, outils d'analyse et visualisation)

Cycle de vie

❑ Analyse

- Développer le schéma de l'entrepôt
- Définir les processus nécessaires à la mise en place de l'entrepôt (extraction de données à partir des sources, transformations)

❑ Conception (3 niveaux)

- Conceptuel : mise au point du schéma, définition des méta-données
- Logique : adapté aux particularités du serveur de l'entrepôt (ROLAP, MOLAP, etc.)
- Physique: choix d'index, vues matérialisées, fragmentation

Cycle de vie

❑ Construction

- Développer des programmes d'extraction, d'épuration et de transformation de données

❑ Déploiement

- Fournir une installation initiale incluant la connexion aux données sources, la synchronisation et la réplication de données
- Permettre des extensions futures
- Offrir la formation pour les groupes d'intervenants
- Offrir les divers mécanismes d'administration de l'ED (reprise, sécurité, performances)
- Offrir les outils nécessaires à l'exploitation des données et à la consultation des méta-données

MODELISATION DES ED

Modélisation classique (OLTP)

- **Le modèle relationnel**
  - Table, attributs, tuples, vues, …
  - Normalisation (redondance)
  - Requêtes simples (sélection, projection, jointure, …)
- **Le critère temps**
  - Représentation du passé

Requêtes décisionnelles plus complexes !!

- **Exemples**
- Combien de clients âgés entre 20 et 30 ans et résidant à Alger ont-ils acheté une caméra vidéo au cours des 5 dernières années ?
- Quelle est la répartition des ventes par produit, ville et par mois au cours de la présente année?
- Quelles sont les composantes des machines de production ayant eu le plus grand nombre d'incidents imprévisibles au cours de la période 1992-97 ?

➡ Critère temps est la base de l'analyse décisionnelle

Récapitulatif

	Base de données	Entrepôt de données
Opération	▪ Gestion courante	▪ Support à la décision
Modèle	▪ Entité association	▪ Etoile, flocon de neige
Normalisation	▪ Plus fréquente	▪ Rare
Données	▪ Actuelle, brutes	▪ Historiques, agrégées
Mise à jour	▪ immédiate	▪ Plus différée
Perception	▪ Bidimensionnel	▪ Multidimensionnelle
Opérations	▪ Lecture/écriture	▪ Lecture et rafraichissement
Taille	▪ Des giga-octets	▪ Vers des téra, péta-octets

OLAP

- ❑ Traitement analytique interactif (Codd) typique dans les systèmes informationnels
- ❑ Catégorie de traitements dédiés à l'aide à la décision
- ❑ Analyses diverses (multidimensionnelles)
- ❑ Information : surtout dérivée et sommaire
- ❑ Aide à la prise de décision

Modélisation Multidimensionnelle

❑ Dimension:

- Présente le point de vue selon lequel on veut voir le données décrites par un ensemble d'attributs· Axe de l'analyse
  - Exemple: Commandes, achats, réclamations, produits, clients,...

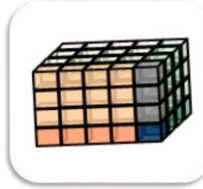
❑ Mesures/faits

- Fonction numérique qui peut être évaluée en tout point du data cube en agrégeant les données correspondant à ce point
- Mesure d'activité (critère d'analyse)
  - Exemple: Chiffre d'affaire, nombre de ventes, gain

Hyper cube OLAP

❑ Objectifs

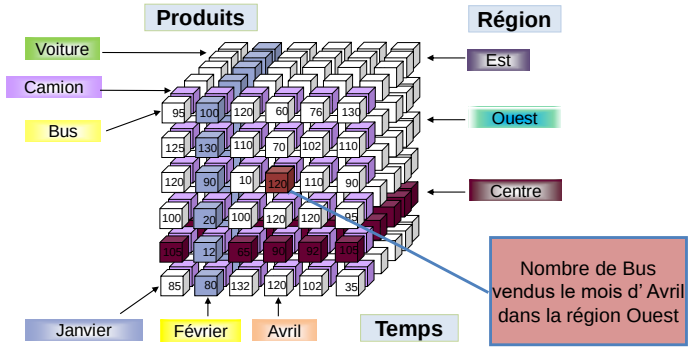
- Obtenir des informations déjà agrégées selon les besoins des utilisateurs
- Représentation de l'information dans un hyper cube à N dimensions



❑ Opérations OLAP

- Fonctionnalités qui servent à faciliter l'analyse multidimensionnelles: opérations réalisables sur l'hyper cube

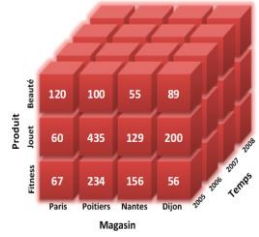
Exemple d'un cube de données



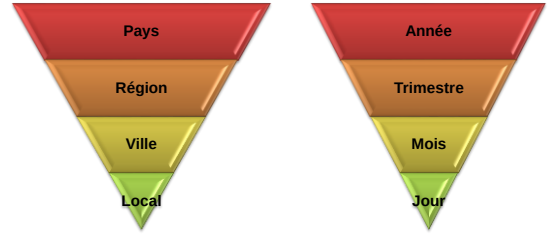
Vue multidimensionnelle des données

❑ Le volume des ventes (mesure) en fonction de : Produit, Temp et Localisation (Dimension)

Dimensions: Produit, Région, Temps



Hiérarchie des dimensions



Comment stocker le cube de données

- ROLAP: Relational On-Line Analytical Processing
  - Les données sont stockées comme des tables relationnelles: une table de faits et des tables de dimension
- MOLAP: Multidimensional On-Line Analytical Processing
  - le cube de données est stocké sous forme d'un tableau multi-dimensionnel.

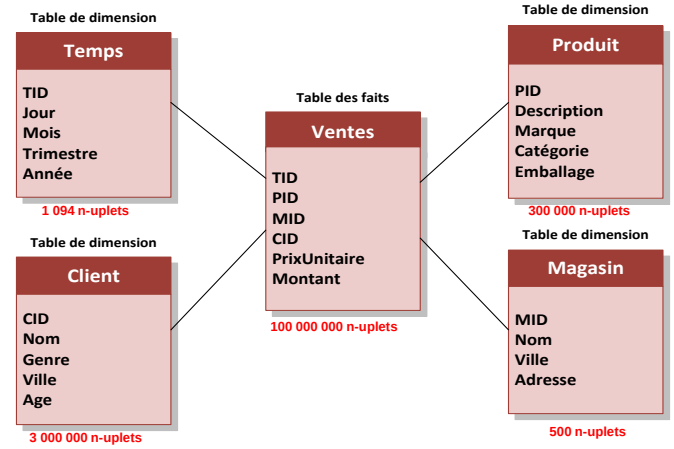
Modèle ROLAP

- Exploiter l'expérience des modèles relationnels (un grand succès!!)
- Il faut des modèles bien adaptés aux ED!
  - Schéma en étoile (star schema)
  - Schéma en flocon de neige (snowflake schema)
  - Schéma en constellation

Modèle en étoile

- Autant de tables de dimension qu'il existe de dimensions
  - **Exemple**
    - Temps, Produit, Client
- Une table de faits contenant la clé de chaque dimension et des mesures
  - **Exemple**
    - Montant en dollars, nombre d'unités vendues

Schéma en étoile



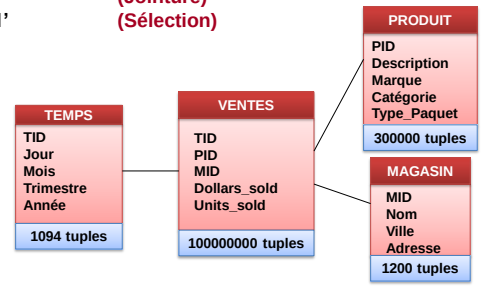
Requête type

Requête de Jointure en étoile :

```
SELECT    P.Marque, sum(dollars_sold),
sum(units_sold)
FROM      Ventes V, PRODUIT P, TEMPS T
WHERE     V.PID = P.PID              (Jointure)
AND       V.TID = T.TID              (Jointure)
AND       T.Trimestre = 'T1'         (Sélection)
GROUP BY  P.Marque
ORDER BY  P.Marque
```

Requêtes de jointure en étoile

- Plusieurs jointures
- Suivies par des sélection



Avantages & Inconvénients

\+ Simple

\+ Le plus utilisé !!!

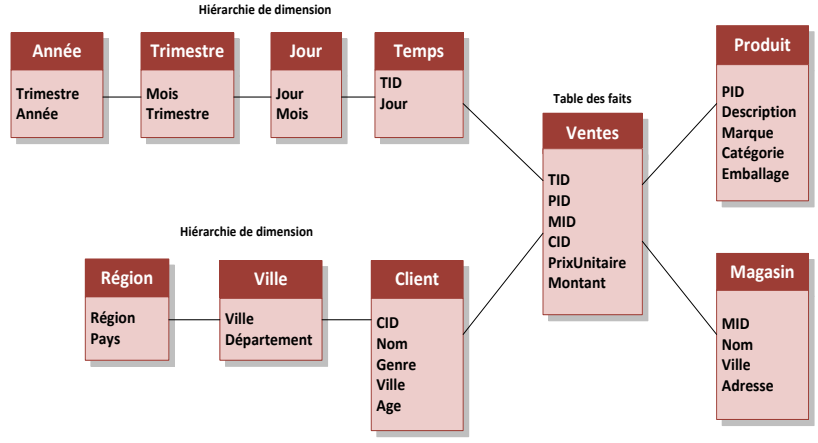
\- Possibilité de redondance car les tables de dimension ne sont pas nécessairement normalisées

\- Taille de dimensions plus grosse

Modèle en flocon de neige

- Variante du modèle en étoile
- Les tables de dimension sont normalisées
- Réduction de la redondance mais exécution parfois plus lente des requêtes (jointure de tables)
- Modèle adopté par Oracle!!

Exemple d'un modèle en flocon de neige



Définition d'un schéma en étoile avec DMQL

define cube ventes_star [Temps, Produit, Client]

Montant_vente = sum(somme),

moyenne_vente = avg(somme),

unités_vendues = count(*)

Mesures

Dimensions

define dimension Temps as (TID, Année, mois, jour)

define dimension Produit as (PID, nom_item, marque, taille, poids)

define dimension Client as (Cid, Nom, Sexe, Age, Ville)

Définition d'un schéma snowflake avec DMQL

define cube ventes_snowflake [temps, item, branche, lieu]

Montant_vente = sum(somme),

moyenne_vente = avg(somme),

unités_vendues = count(*)

Mesures

Dimensions

define dimension item as (Id_item, nom_item, marque, type,

fournisseur(Id_fournisseur, type_fournisseur))

define dimension branche as (Id_branche, nom_branche, type_branche)

define dimension temps as (Id_temps, jour, jour_semaine, mois, trimestre, année)

define dimension lieu as (Id_lieu, rue, ville(Id_ville, département, pays))

Hiérarchies

MOLAP: Représentation Tableaux

Produit	Temps	Trimestre 1			Trimestre 2			Trimestre 3			Trimestre 4			Total		
	Ville	P	N	Total	P	N	Total	P	N	Total	P	N	Total	P	N	Total
TV LCD		12	34	46	22	36	58	24	37	61	33	55	88	91	162	253
Lecteur DVD		29	66	95	44	50	94	56	55	111	44	39	83	173	210	383
Caméoscope		55	34	89	69	27	96	31	26	57	68	70	138	223	157	380
Total		96	134	230	135	113	248	111	118	229	145	114	309	487	529	1016

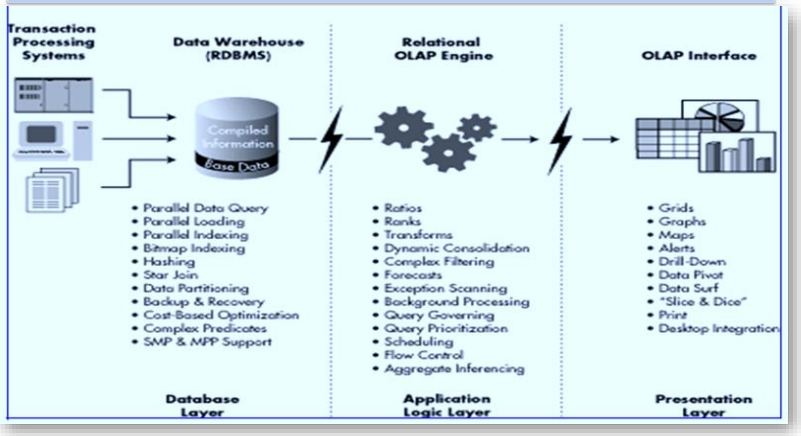
Répartition des ventes par produit, temps et ville

P : Poitiers, N : Nantes

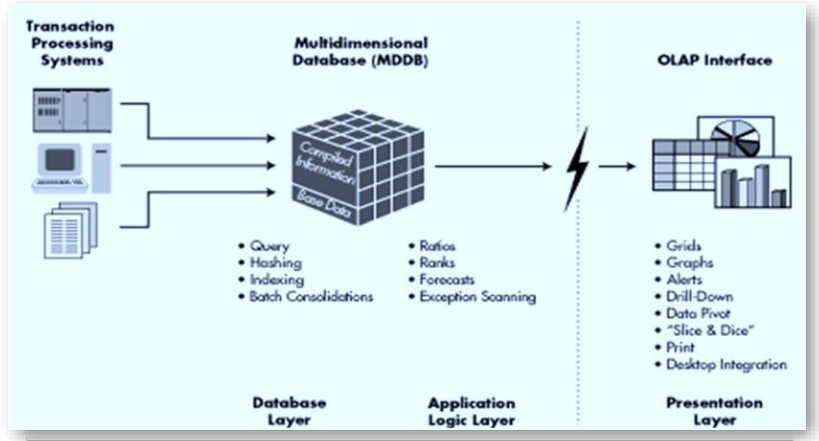
ROLAP vs. MOLAP

	Avantages	Inconvénients
ROLAP	▪ Standard bien établi ▪ Efficace pour le transactionnel ▪ Capacité d'expansion (téra-octet)	▪ Absence de vue conceptuelle ▪ SQL peut être inadéquat pour la formation de tableaux croisés
MOLAP	▪ Implémentation souvent plus performante que ROLAP	▪ Inadéquat pour le transactionnel ▪ Capacité d'expansion limitée (dizaine de géga-octet) ▪ Absence de standard

Serveurs ROLAP (Microstrategy)



Serveurs MOLAP

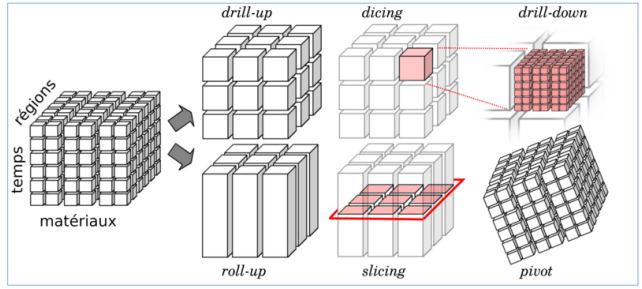


Hybrid OLAP

- HOLAP combine ROLAP et MOLAP
- Les données sont partitionnées en deux sous-ensembles
  - Données non fréquentes sont stockées comme tables relationnelles
  - Données fréquentes sont stockées comme tableaux multidimensionnels
- Données brutes dans ROLAP
- Données agrégées dans MOLAP
- Cette séparation est transparente pour l'utilisateur.

Opérations de base OLAP

- Roll-up (ou drill-up)
- Drill-down
- Slice & dice
- Pivot



Optimisation des Entrepôts de Données

Optimisation des requêtes

❑ **Caractéristiques des entrepôts**

- Grand volume de données (TB)
- Requêtes décisionnelles complexes
- Peu de mises à jour (ajout seulement!)

❑ **Exigence des décideurs**

- Temps de réponse raisonnable
- Respect des délais

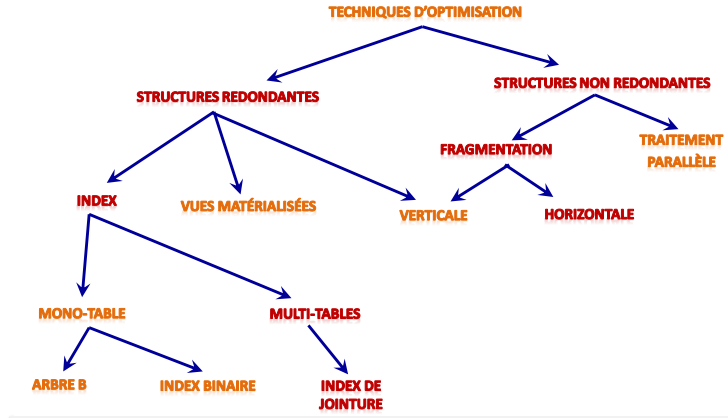
Objectif Exécution rapide des requêtes ∀ la complexité des requêtes ou des données

➡ **Nécessité de techniques d'optimisation**

Techniques utilisées

- **Vues matérialisées**
  - Une vue est une requête nommée
  - Une vue matérialisée : les données résultant de sa requête sont stockées et maintenues
- **Index**
  - Structures permettant d'associer à une clé d'un n-uplet l'adresse relative de cet n-uplet
- **Traitement parallèle**
- **Groupement**
- **Fragmentation**

Classification



Vues matérialisées

- **Exigence de ressources:**
  - Espace disque
  - Coût de maintenance (rafraîchissement des données)
  - Coût de calcul

➡ Impossibilité de matérialiser toutes les vues

- **Problème de Sélection des Vues (PSV):**

  - Sélectionner un ensemble de vues afin de maximiser ou minimiser une fonction objectif sous une contrainte

  - Fonction « objectif »:

    Optimiser le coût d'exécution des requêtes ou Optimiser le coût de maintenance

Exemple

```
CREATE MATERIALIZED VIEW VUE1
ENABLE QUERY REWRITE
AS
SELECT  *
FROM Vente, Client, Article
 WHERE Vente.noClient = Client.noClient AND
Vente.noArticle = Article.noArticle
```

Maintenance des vues matérialisées

- Vues matérialisées sont calculées à partir des sources (tables)
- Mise à jour des sources implique la mise à jour des vues
- Deux méthodes de maintenance
  - Statique
  - Incrémentale

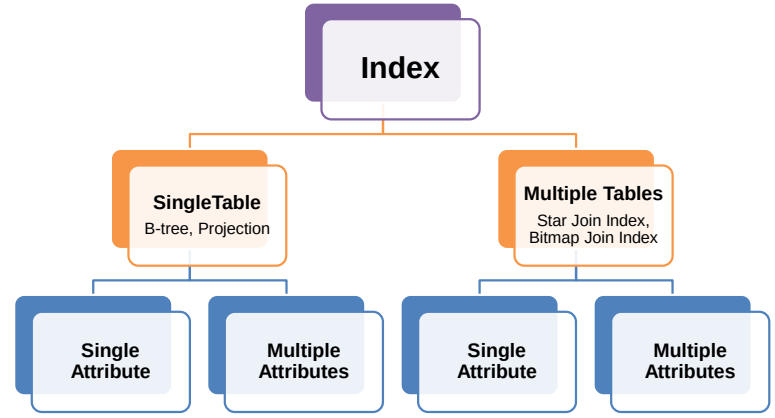
Réécriture des requêtes

- Processus de réécriture:
  - Après le processus de sélection des vues, toutes les requêtes définies sur l'entrepôt doivent être réécrites en fonction des vues
- Sélectionner la meilleure réécriture pour une requête est une tâche difficile.
- Processus supporté dans la plupart des SGBD multidimensionnels (Oracle)

Index avancés

- Index populaires dans les entrepôts:
  - Index simple: sur une seule table ou une seule vue (arbre B, index binaire, index de projection)
  - Index de jointure: sur plusieurs tables dans un schéma en étoile
    - index de jointure en étoile
    - Index de jointure binaire
- Problème de sélection d'index
- Algorithmes de sélection d'index

Classification des Index



Techniques d'indexation

❑Index bitmap (index binaire)

❑ Populaire dans les produits OLAP

- Microsoft SQL server, Sysbase IQ, Oracle
- Un vecteur de bits pour chaque valeur d'attribut

❑ Opération bit à bit pour l'exécution de requêtes

- Comparaison
- Jointure
- Agrégation
- Plus compact que les B+ arbres (compression)

Principe de l'index bitmap

- Soit un attribut A, ayant prenant n valeurs possibles [v_1, …,v_n] (domaine)
- Création d'un index bitmap sur l'attribut A:
  - On crée n tableaux de bit, un pour chaque valeur v_i
  - Le tableau contient un bit pour chaque tuple t
  - Le bit d'un tuple t est à 1 si: t.A = v_i, à 0 sinon

Table Employé

ROWID(RID)	N°E	Nom	Fonction
00055 :000 :0023	1	Karim	Fraiseur
00234 :020 :8922	2	Hichem	Soudeur
19000 :328 :6200	3	Salim	Tourneur
21088 :120 :1002	4	Ali	Sableur

Index binaire sur l'attribut Fonction

ROWID(RID)	Soudeur	Fraiseur	Sableur	Tourneur
00055 :000 :0023	0	1	0	0
00234 :020 :8922	1	0	0	0
19000 :328 :6200	0	0	0	1
21088 :120 :1002	0	0	1	0

Index de jointure

- Pré-calculent une opération de jointure
- Utilisés avec les schémas en étoile
- Maintient des relations entre
  - Une clé étrangère (identifiant des tables de dimensions)
  - Les clés primaires de la table des faits

Index de jointure

IdLigneVente	idLigneClient	idLigneArticle
V1	C1	A1
V2	C2	A2
V3	C3	A1
…		

Table Client

IdLigne	noClient	nomVille
C1	1	Poitiers
C2	2	Nantes
C3	3	Poitiers
C4	4	Paris

Table Ventes

IdLigne	noClient	noArticle	DateVente	Montant
V1	1	10	10/01/2008	100
V2	2	20	10/01/2008	200
V3	3	10	10/01/2008	500
…				

Table Article

IdLigne	noArticle	catégorie
A1	10	Beauté
A2	20	Beauté
A3	40	Beauté
A4	50	Jouet
A5	60	Jouet

Index de jointure Bitmap : Exemple

Client

RID^C	CID	Nom	Ville
6	616	Gilles	Poitiers
5	515	Yves	Paris
4	414	Patrick	Nantes
3	313	Didier	Nantes
2	212	Eric	Poitiers
1	111	Pascal	Poitiers

Produit

RID^P	PID	Nom	Catégorie
6	106	Sonoflore	Beauté
5	105	Clarins	Beauté
4	104	WebCam	Multimédia
3	103	Barbie	Jouet
2	102	Manure	Jardinage
1	101	SlimForm	Fitness

Temps

RID^T	TID	Mois	Année
6	11	Janvier	2003
5	22	Février	2003
4	33	Mars	2003
3	44	Avril	2003
2	55	Mai	2003
1	66	Juin	2003

Ventes

RID^S	CID	PID	TID	Montant
1	616	106	11	25
2	616	106	66	28
3	616	104	33	50
4	515	104	11	10
5	414	105	66	14
6	212	106	55	14
7	111	101	44	20
8	111	101	33	27
9	212	101	11	100
10	313	102	11	200
11	414	102	11	102
12	414	102	55	103
13	515	102	66	100
14	515	103	55	17
15	212	103	44	45
16	111	105	66	44
17	212	104	66	40
18	515	104	22	20
19	616	104	22	20
20	616	104	55	20
21	212	105	11	10
22	212	105	44	10
23	212	105	55	18
24	212	106	11	18
25	313	105	66	19
26	313	105	22	17
27	313	106	11	15

	Ville		
RID	P	Pr	N
1	1	0	0
2	1	0	0
3	1	0	0
4	0	1	0
5	0	0	1
6	1	0	0
7	1	0	0
8	1	0	0
9	1	0	0
10	0	0	1
11	0	0	1
12	0	0	1
13	0	1	0
14	0	1	0
15	1	0	0
16	1	0	0
17	1	0	0
18	0	1	0
19	1	0	0
20	1	0	0
21	1	0	0
22	1	0	0
23	1	0	0
24	1	0	0
25	0	0	1
26	0	0	1
27	0	0	1

```
SELECT    count(*)
FROM      Ventes V, Client C
WHERE     V.CID = C.CID
AND       C.Ville = 'Poitiers'
```

```
CREATE BITMAP INDEX
Ventes_Pictaviens_bjix ON
Ventes(Client.Ville) FROM
Ventes, Client WHERE
Ventes.CID = Client.CID
```

Deux opérations :

1. Lire dans le bitmap la colonne P
2. Compter le nombre de 1.

➔ Pas de lecture dans la table Ventes

Caractéristiques des Bitmaps

- Les opérations de comparaison, jointure et d'agrégation sont réduites à une arithmétique sur les bits, d'où un traitement plus efficace
- La réponse à une requête multidimensionnelle va consister à faire l'intersection des vecteurs de bits des diverses dimensions
- Réduction significative en espace et nombre d'accès en mémoire secondaire

Intérêt des index de jointure binaire

```
CREATE BITMAP INDEX EMPDEPT_IDX ON
 EMP1(DEPT1.DEPTNO)
FROM EMP1, DEPT1
WHERE EMP1.DEPTNO = DEPT1.DEPTNO
```

```
SELECT /*+ INDEX(EMP1 EMPDEPT_IDX) */ COUNT(*)
FROM EMP1, DEPT1
WHERE EMP1.DEPTNO = DEPT1.DEPTNO;
```

```
COUNT(*)
----------
        14
Elapsed: 00:00:00.67
```

Bitmap Join Index Example 2

Deux tables EMP5/EMP6 ayant 2 million d'instances chacune, empno est la clé étrangère

```
create bitmap index emp5_j6 on
emp6(emp5.empno) from emp5, emp6
where emp5.empno=emp6.empno;
```

Index created.

Elapsed: 00:02:29.91

Bitmap Join Index Example 2

```
select count(*)
from emp5, emp6
where emp5.empno=emp6.empno
```

COUNT(*)

2005007

Elapsed: 00:01:07.18

Bitmap Join Index Example 2

Plan d'exécution

```
0       SELECT STATEMENT Optimizer=CHOOSE
1   0   SORT (AGGREGATE)
2   1     NESTED LOOPS
3   2       TABLE ACCESS (FULL) OF 'EMP6'
4   2       INDEX (RANGE SCAN) OF 'EMP5I_EMPNO' (NON-UNIQUE)
```


Bitmap Join Index Example 2

FORCER L'UTILISATION DE L'INDEX

```
select /*+ index(emp6 emp5_j6) */ count(*)
from emp5, emp6
where emp5.empno=emp6.empno
```

```
COUNT(*)
----------
2005007
```

Elapsed: 00:00:00.87! Comme avec les tables de petites tailles!

Bitmap Join Index - 10,000 plus rapide

Plan d'exécution

```
0      SELECT STATEMENT Optimizer=CHOOSE
1   0    SORT (AGGREGATE)
2   1      BITMAP CONVERSION (COUNT)
3   2        BITMAP INDEX (FULL SCAN) OF 'EMP5_J6'
```


Selection of BJI : Formalization

INPUT

- Set of dimension tables D={D_1, D_2, ..., D_d}
- Fact Table F
- Workload of frequently queries Q
- **Storage Space S**

OUTPUT

- Set S_BJI of BJI

OBJECTIVES

- Reduce execution cost of Q
- Size (S_BJI) ≤ S

Selection of BJI : Complexity

- A={A_1, A_2, …, A_K} is a set of indexable attributes
- A potential BJI is defined on a subset of A
- N the number of potential BJI

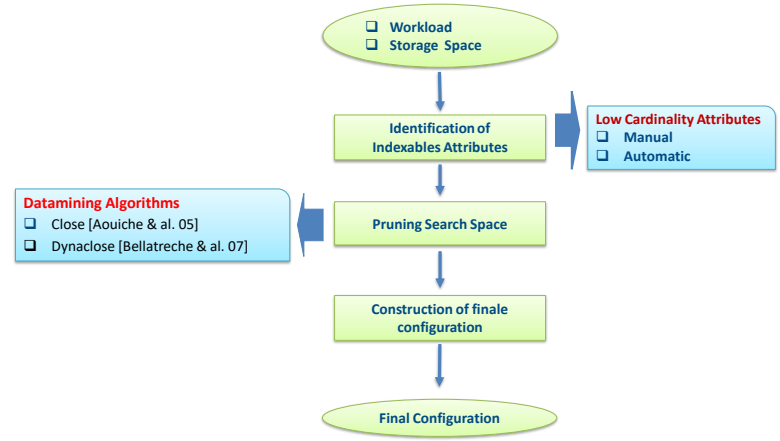
Select One BJI

$$N = \binom{k}{1} + \binom{k}{2} + \ldots\ldots + \binom{k}{k} = 2^k - 1$$

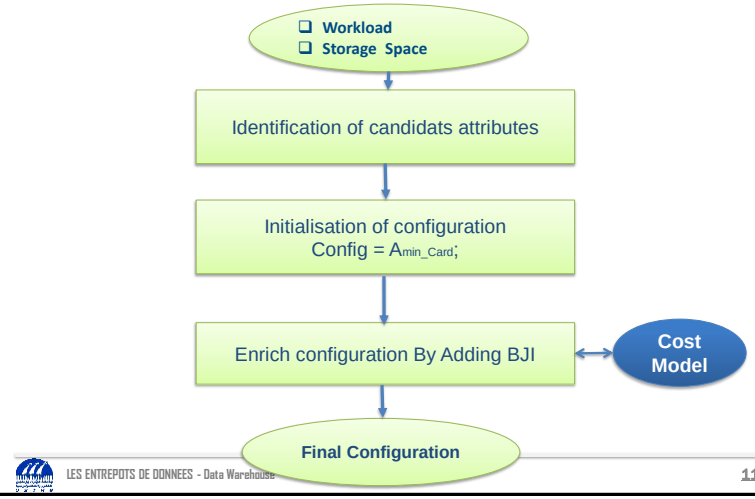
Select More Than One BJI

$$N = \binom{2^k-1}{1} + \binom{2^k-1}{2} + \ldots\ldots + \binom{2^k-1}{2^k-1} = 2^{2^k-1}$$

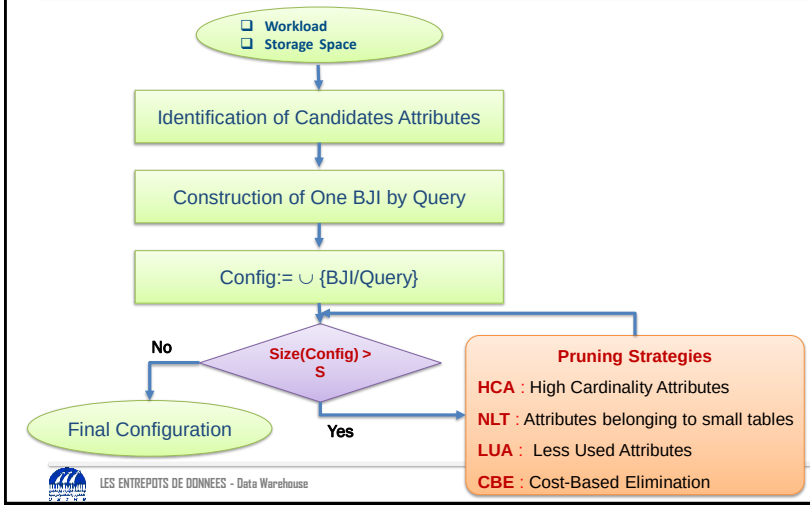
BJI: Existant



Our approach : Single attribute BJI



Our approach : Multiple Attributes BJI



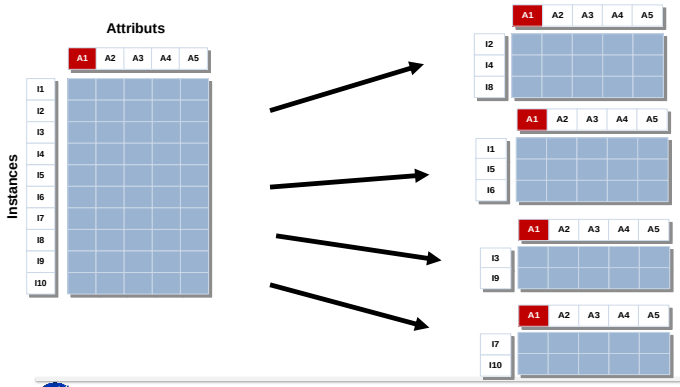
Fragmentation
Ou
Partitionnement

La fragmentation de données

- **Définition**
  - ❑ **Décomposer les objets de la BD (relation, index, vues) en un ensemble de morceaux appelés Partitions.**
- **Fragmentation horizontale**
  - ❑ **La table est fragmentée par rapport à ses instances en un ensemble de lignes.**
- **Fragmentation verticale**
  - ❑ **La table est fragmentée selon ses attributs en un ensemble de colonnes.**
- **Fragmentation mixte**
  - ❑ **La table est fragmentée horizontalement et verticalement.**

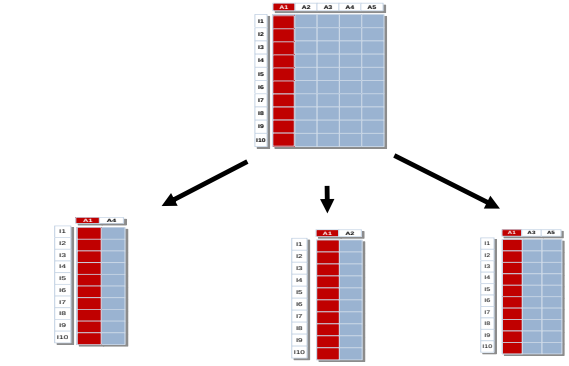
Types de fragmentation

- **Fragmentation horizontale**
  - Décomposer les objets en un ensemble de lignes (instances)



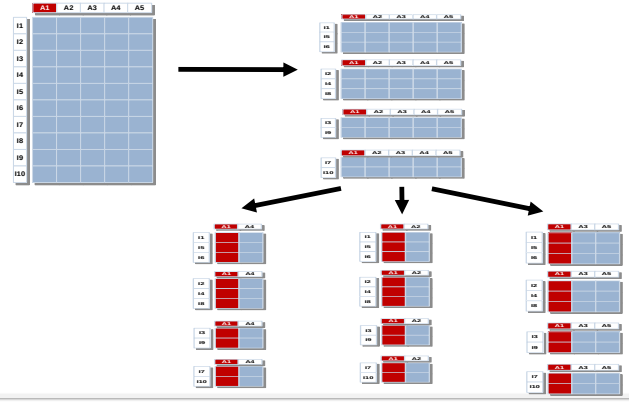
Types de fragmentation

- **Fragmentation verticale**
  - Décomposer les objets en un ensemble de colonnes (attributs).



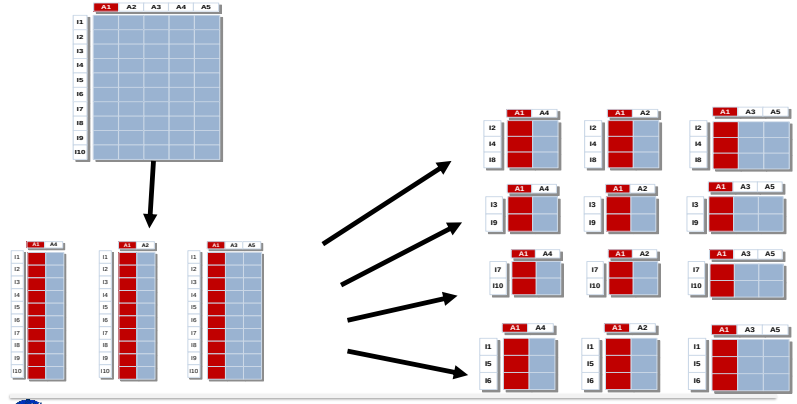
Types de fragmentation

- **Fragmentation mixte**
  - ❑ Horizontale suivie d'une verticale.



Types de fragmentation

- **Fragmentation mixte**
  - ❑ Verticale suivie d'une horizontale



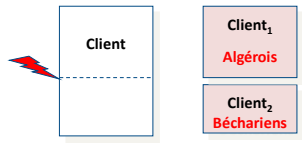
Fragmentation Horizontale Primaire et dérivée (I)

Fragmentation horizontale primaire (FHP)

- Fragmenter une table en utilisant les prédicats de sélection définis sur cette table

  Prédicat : Attribut θ Valeur, θ ∈{=,<, ≤, >, ≥,} et valeur ∈Domaine(Attribut).

- Exemple: Client (Client_id, Nom, Ville)
  - $Client_1$: $\sigma_{Ville='Alger'}(Client)$
  - $Client_2$: $\sigma_{Ville='Béchar'}(Client)$



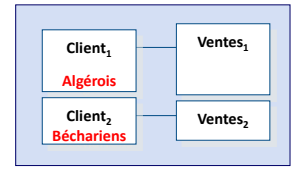
Impact de la FHP sur les requêtes

- **Optimisation des sélections**

Fragmentation Horizontale Primaire et dérivée (II)

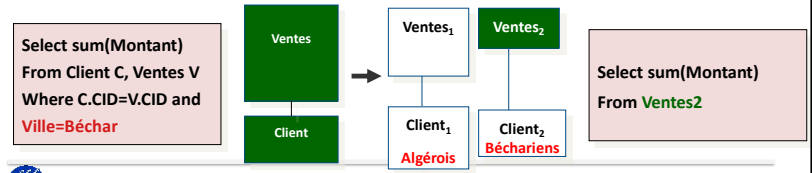
Fragmentation horizontale dérivée (FHD)

- Fragmenter une table (S) selon des attributs d'une autre table (T) : (existence de lien entre S et T)
- Ventes(Client_id, Produit_id, Date, Montant)
  - $Ventes_1$=Ventes ⋉$Client_1$
  - $Ventes_2$=Ventes ⋉ $Client_2$



Impact de la FHD sur les requêtes

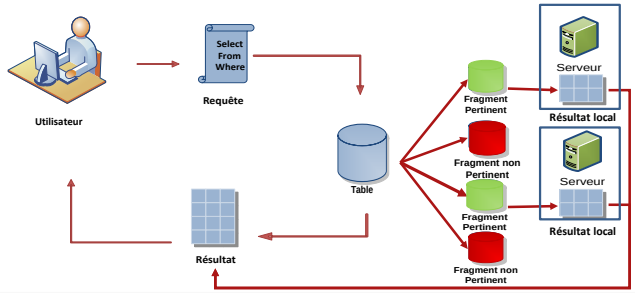
- **Optimisation de la jointure entre S et T**



Intérêt de la fragmentation horizontale(I)

- Améliorer la performance

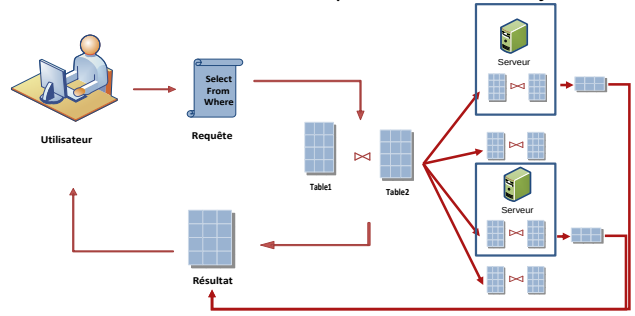
1. Partition Elimination (Pruning)
   - Elimination des partitions non pertinentes
   - Possibilité d'exécution parallèle des sous requêtes



Intérêt de la fragmentation horizontale(II)

- Amélioration de la performance

2. Partition Wise Joins
   - ❑ Elimination des jointures non pertinentes
   - ❑ Possibilité d'exécution parallèle des sous-jointures



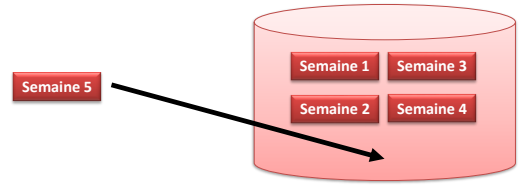
Intérêt de la fragmentation horizontale(III)

- Améliorer la manageabilité
  - Fragmentation horizontale préserve le schéma logique
    - Toutes les opérations se font au niveau partition
    - Possibilité de manipulation individuelle ou collective des partitions
  - Manipuler une partition à la fois.
    - Possibilité de définir des index locaux aux partitions
    - Existences de plusieurs Fonctions de manipulation des partitions

Fonction	Signification
ADD PARTITION	Ajouter une partition à une table déjà fragmentée
COLESCE PARTITION	Redistribuer les tuples d'une partition dans les autres partitions
DROP PARTITION	Supprimer une partition ainsi que son contenu
EXCHANGE PARTITION	Convertir une table non fragmentée en une partition d'une autre table et inversement
MERGE PARTITION	Fusionner deux partitions dans une seule
SPLIT PARTITION	Eclater une partition en deux partitions
TRUNCATE PARTITION	Vider une partition sans la supprimer

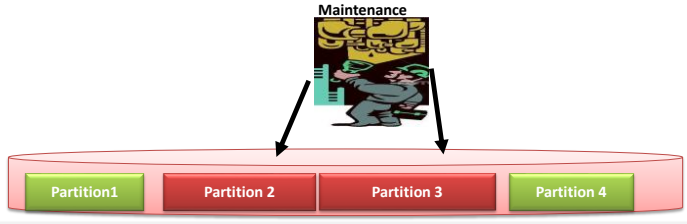
Exemple d'ajout de données

- Alimentation hebdomadaire d'une table
  - L'administrateur partitionne sa table par semaine
  - L'alimentation de l'entrepôt consiste alors à ajouter une partition à la table
  - Aucune autre partition ne sera touchée.



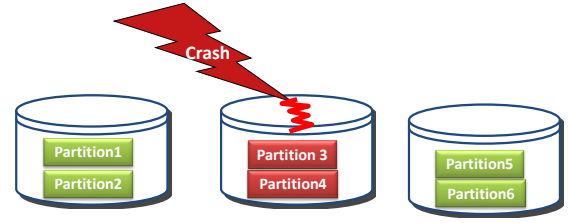
Intérêt de la fragmentation horizontale(IV)

- Améliorer la maintenance et la disponibilité
  - La manipulation au niveau partition permet
    - Cibler la maintenance sur certaines partitions
    - Minimiser le temps de maintenance
  - La possibilité de stocker les partitions sur des emplacements différents permet
    - L'effet des pannes de disques est limité
    - Partitions saines toujours disponibles



Améliorer la maintenance et la disponibilité

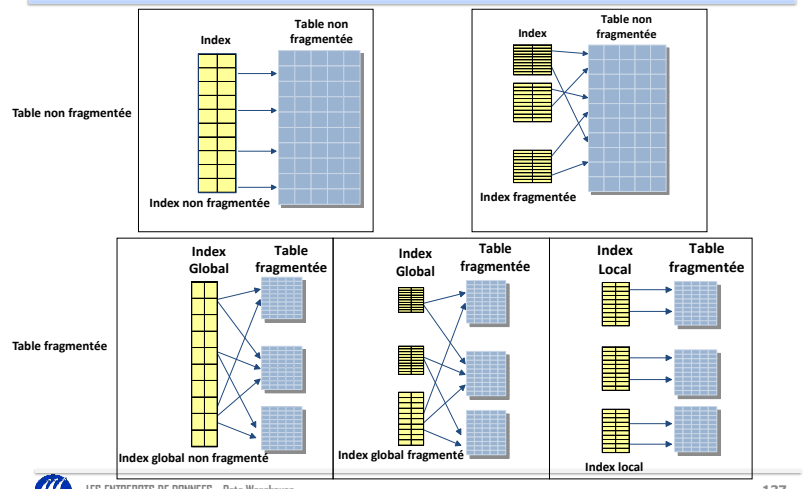
- Améliorer la maintenance et la disponibilité
  - La possibilité de stocker les partitions sur des emplacements différents permet
    - L'effet des pannes de disques est limité
    - Partitions saines toujours disponibles



Fragmentation horizontale et index

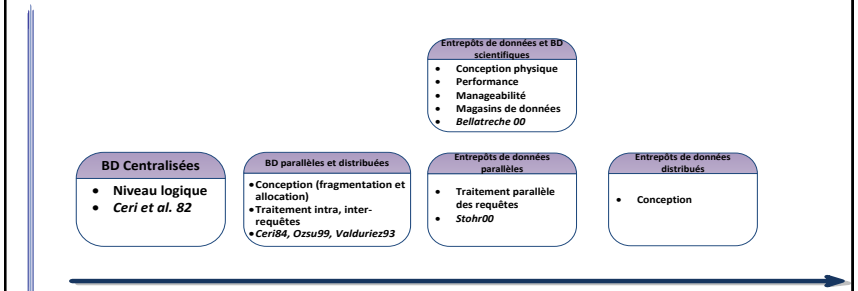
- Un index peut être défini sur une table fragmentée ou non
- Un index défini sur une table fragmentée peut être fragmenté ou non.
- **Index global**
  - Peut être non fragmenté défini sur une table fragmentée
  - Peut être fragmenté où chaque partition de l'index référence plusieurs partitions de la table
- **Index local**
  - Equi-partitionné avec la table qu'il référence
  - Chaque partition de l'index référence une et une seule partition de la table

Fragmentation et index



Evolution de la fragmentation horizontale

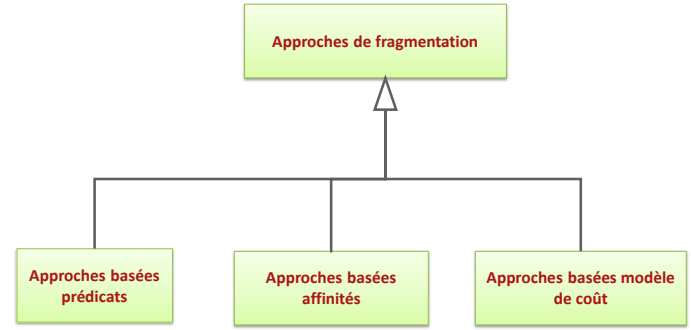
- Evolution Académique



2(?

Evolution Académique (I)

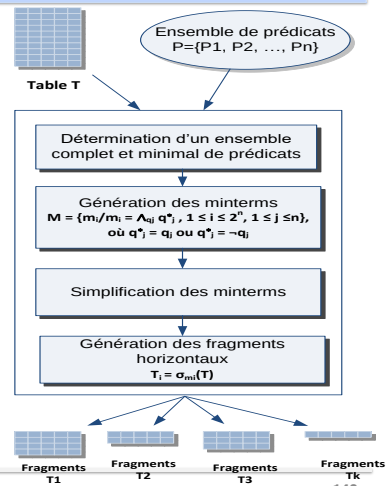
- Approches de fragmentation



Evolution Académique (II)

Approches basés prédicats

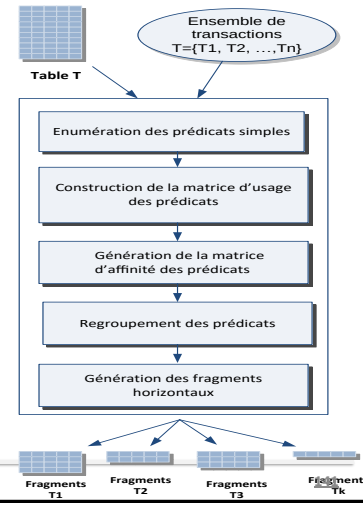
- **Avantages**
  - Simple
- **Inconvénients**
  - Complexité : 2^n minterms générés
  - Aucune métrique pour estimer la qualité du schéma obtenu



Evolution Académique (III)

Approches basées Affinités

- Adaptation des travaux de Navathé sur la fragmentation verticale
- Regrouper les prédicats souvent utilisés ensemble pour former des fragments horizontaux
- **Avantages**
  - Complexité réduite
- **Inconvénients**
  - Aucune métrique pour estimer la qualité du schéma obtenu

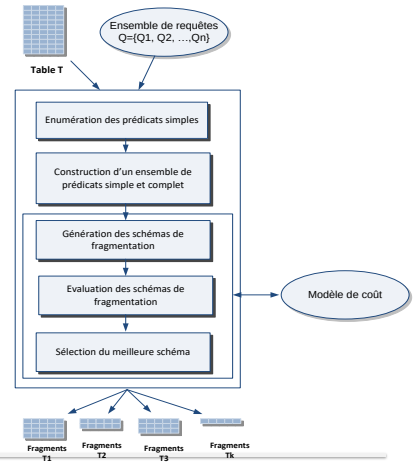


Evolution Académique (IV)

Approches basées modèle de coût

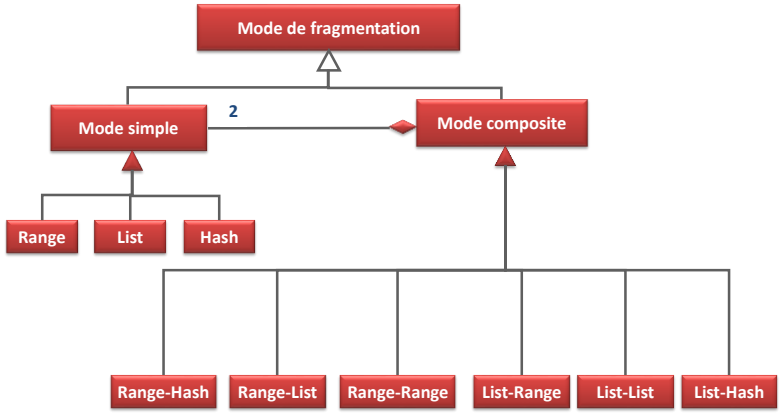
- Chaque schéma de fragmentation est évalué en utilisant un modèle de coût
- Le modèle de coût permet d'estimer le coût d'exécution des requêtes les plus fréquentes sur le schéma fragmenté

- **Inconvénient**
  - Le nombre de fragments générés peut être très important

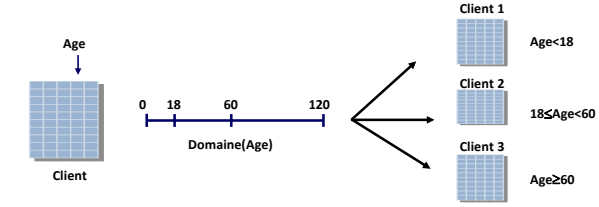


Evolution industrielle

Modes de fragmentation

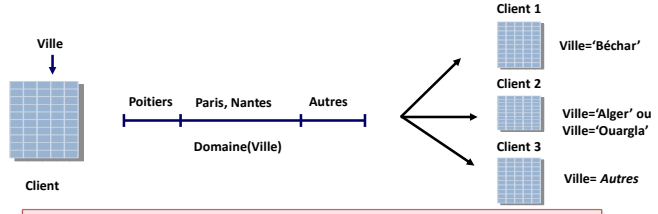


Fragmentation par intervalle « Range »



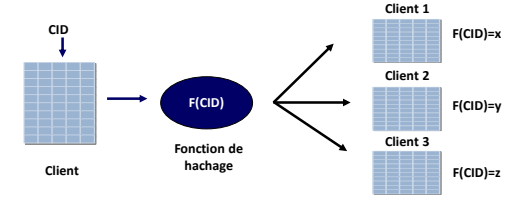

```
CREATE TABLE Client
(Client_id  NUMBER(5),
Nom  Varchar2(20),
Ville Varchar2(20),
Age Number(3),
Genre Varchar2(1))
PARTITION BY RANGE (Age)
 (
PARTITION Client_Moins_18 VALUES LESS THAN (18) TABLESPACE TBSMoins27,
PARTITION Client_18_59 VALUES LESS THAN (60) TABLESPACE TBS27-59,
PARTITION Client_60_Et_Plus VALUES LESS THAN (MAXVALUES) TABLESPACE TBSPlus60
);
```

Fragmentation par liste « List »



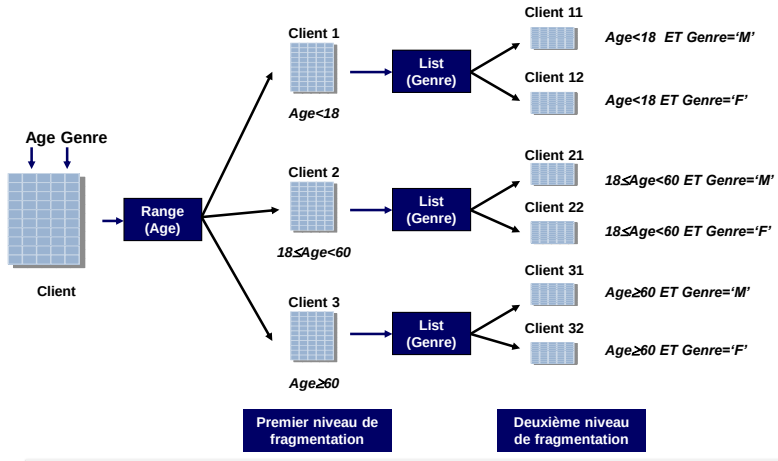

```
CREATE TABLE Client
(Client_id  NUMBER(5),
Nom  Varchar2(20),
Ville Varchar2(20),
Age Number(3),
Genre Varchar2(1))
PARTITION BY List (Ville)
 (
PARTITION Client_Béchar VALUES ('Béchar') TABLESPACE TBSBECHAR,
PARTITION Client_Alg_Oua VALUES ('Alger','Ouargla') TABLESPACE TBSALGOUA,
PARTITION Client_Autres VALUES (DEFAULT) TABLESPACE TBSAUTRES
);
```


Fragmentation par hachage « Hash »




```
CREATE TABLE Client
(Client_id  NUMBER(5),
Nom  Varchar2(20),
Ville Varchar2(20),
Age Number(3),
Genre Varchar2(1))
PARTITION BY Hash (CID)
 (
PARTITIONS 3
STORE IN (TBS1, TBS2, TBS3)
);
```

Modes de fragmentation composite

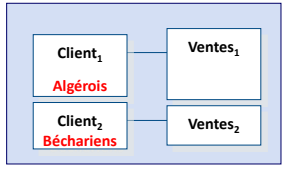


Modes de fragmentation composite

```
CREATE TABLE Client
(Client_id  NUMBER(5),
Nom  Varchar2(20),
Ville Varchar2(20),
Age Number(3),
Genre Varchar2(1))
PARTITION BY RANGE (Age)
SUBPARTITION BY LIST (Genre)
SUBPARTITION TEMPLATE(
SUBPARTITION  Client1 VALUES ('M') TABLESPACE TBSMasculin ,
SUBPARTITION Client2 VALUES ('F') TABLESPACE TBSFéminin )
(
PARTITION Client_Moins_18 VALUES LESS THAN (18),
PARTITION Client_18_59 VALUES LESS THAN (60),
PARTITION Client_60_Et_Plus VALUES LESS THAN (MAXVALUES)
);
```

Fragmentation dérivée par le mode Référence

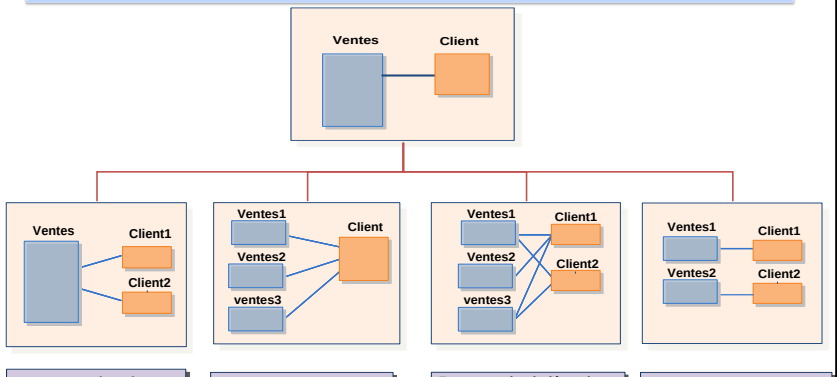
- Fragmenter une table selon le schéma de fragmentation d'une autre table en utilisant le lien par clé étrangère.




```
CREATE TABLE Ventes
(Client_id  NUMBER(5),
Time_id  NUMBER(5)),
Montant NUMBER(20),
 CONSTRAINT order_items_fk
 FOREIGN KEY(Client_id) REFERENCES Client(Client_id)
 )
PARTITION BY REFERENCE(order_items_fk);
```


Sélection d'un schéma de fragmentation horizontale pour un entrepôt de données relationnel

Scénarii de fragmentation d'un ED relationnel

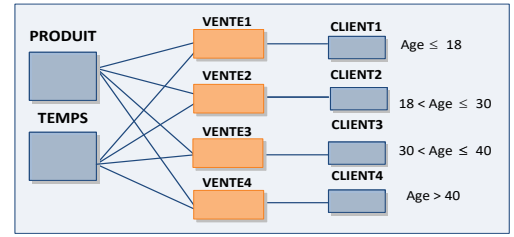


Fragmentation des TD	Fragmentation de la TF	Fragmentation indépendante de ma TF et des TD	Fragmentation des TD
+ Optimisation des sélection - Pas d'optimisation de la jointure	- Pas d'optimisation des jointures - Pas d'optimisation des sélections	+ Optimisation des sélections - Pas d'optimisation des jointures	+ Optimisation des sélection + Optimisation des jointures - explosion du nombre de fragments

Notre démarche de fragmentation

1. Fragmenter (virtuellement/physiquement) des tables de dimension en utilisant la fragmentation primaire
2. Fragmenter la tables des faits (en utilisant les schémas de fragmentation des tables de dimension)

- **Exemple**



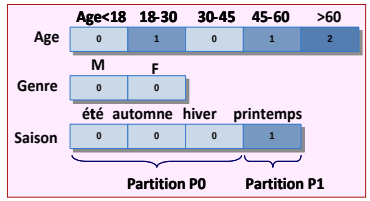
- **Explosion du nombre de fragments**

$$N = \prod_{i=1}^{k} Mi$$

Mi : nombre de fragments de la table de dimension Di
k : nombre de tables de dimension fragmentées

Représentation d'un schéma de fragmentation

- Décomposition des domaines des attributs de fragmentation en sous domaines
- Codage d'un schéma de fragmentation



Exemple

❑Trois attributs de dimension:
Age, Genre, Saison

Les tables sont fragmentées comme suit

- **Table Client en 3 fragments sur Age (Genre n'est pas utilisé)**
  - –Client1 : Age <18 **OU** 30≤Age<45
  - –Client2 : 18≤Age<30 **OU** 45≤Age<60
  - –Client 3 : Age≥60
- **Table temps en 2 fragments sur Saison**
  - –Temps 1 : Saison=été **OU** automne **OU** hiver
  - –Temps2 : Saison=printemps

⇒ Table Ventes fragmentée en :
3x2=6 fragments

Fragmentation Dirigée par le Nombre de Fragments: Formalisation

- **Entrées**
  - Entrepôt de données composé de
    - Un ensemble D de tables de dimension $D=\{D_1, D_2, \ldots, D_d\}$
    - Une table des faits F
  - Charge de requêtes les plus fréquentes $Q=\{Q_1, Q_2, \ldots, Q_m\}$
  - W : seuil (fixé par l'administrateur)
- **Sorties** :
  - Ensemble $D' \subseteq D$ des tables de dimension fragmentées
  - Ensemble de N fragments de faits $F_1, F_2, \ldots, F_N$
- **Objectifs** :
  - Réduire le temps d 'exécution de Q
  - **$N \leq W$**

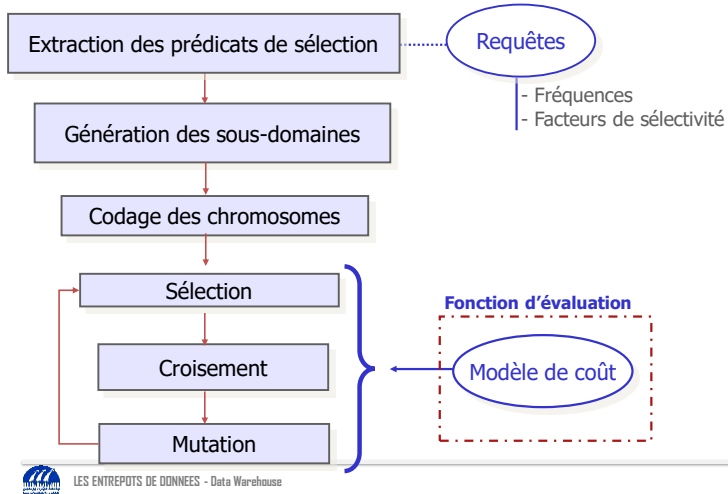
NP-complétude du problème de fragmentation horizontale

- Problème de fragmentation horizontale à un seul domaine (PFHSD)
  - Une seule table de dimension
  - Un seul attribut A composé de n sous domaines
  - Nombre de schémas possibles : Nombre de Bell B_n
  - Pour n grand alors $B_n \cong n^n$.
- Réduction à partir du problème 3-Partition
  - 3-Partition NP-Complet
  - PFHSD NP-Complet
- Le problème de fragmentation est plus compliqué
  - Plusieurs tables de dimension
  - Plusieurs attributs par table de dimension
  - Nombre de schémas de fragmentation $\prod_{i=1}^{k} B_{ni}$

Algorithmes de sélection

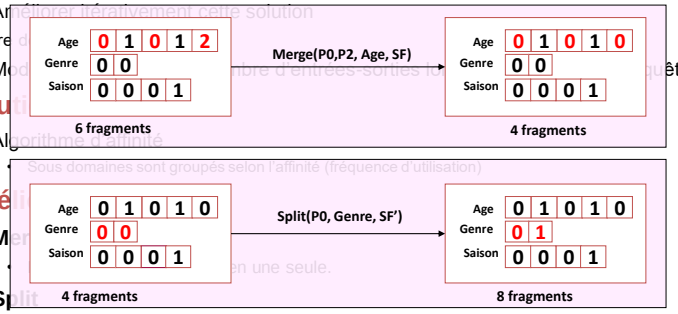
- Algorithme Génétique
- Algorithme de Hill Climbing
- Algorithme de Recuit Simule

Algorithme de sélection : Algorithme Génétique



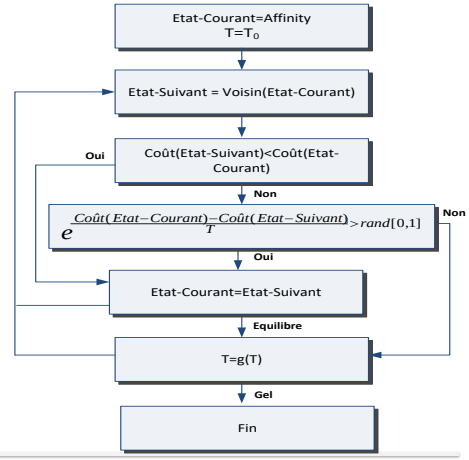
Algorithme de Sélection: Hill Climbing

- **Principe**
  1. Générer une solution initiale
  2. Ar[illegible] cette solution
- Mesure[illegible]
  - Mo[illegible]êtes.
- **Solu[illegible]**
  - Al[illegible]
- **Amé[illegible]**
  - M[illegible]
  - Sp[illegible]
    - Eclater une partition de sous domaines en deux partitions



Algorithme de sélection : Recuit Simulé

- Même principe que HC
- Acceptation de mauvaises solutions
- Eviter les optimums locaux.



Validation sous Oracle(I)

- Problèmes rencontrés
  - FHP sur n (n>2) attributs n'est pas supportée.
  - FHD sur (m >1) tables de dimension n'est pas supportée.
- Solutions
  1. Méthode d'implémentation de la FHP (ajout d'une nouvelle propriété)
  2. Méthode d'implémentation de la FHD (vues matérialisée)

  ⇨ Nécessité de réécrire les requêtes sur les schémas fragmentés
  - Identifier les fragments valides pour chaque requête
  - Réécrire la requête sur ces fragments

Validation Sous Oracle(II)

- Implémentation de la FHP sur plusieurs attributs
  - Fragmentation de la table Client sur Age, Sexe, Ville
  - Age : Age<26, 26≤Age≤60, Age>60 → **Range**
  - Genre : M, F → **Range-List**
  - Ville : Poitiers, Paris, Nantes → **?**

Client

RID^C	CID	Nom	Age	Genre	Ville	Col_C
6	616	Gilles	15	M	Poitiers	1
5	515	Yves	25	F	Paris	4
4	414	Patrick	33	M	Nantes	9
3	313	Didier	50	M	Nantes	9
2	212	Eric	40	F	Poitiers	10
1	111	Pascal	20	M	Poitiers	1

Clients masculins ayant moins de 26 ans habitant Poitiers

RID^C	CID	Nom	Age	Genre	Ville	Col_C
6	616	Gilles	15	M	Poitiers	1
1	111	Pascal	20	M	Poitiers	1

RID^C	CID	Nom	Age	Genre	Ville	Col_C
5	515	Yves	25	F	Paris	4

RID^C	CID	Nom	Age	Genre	Ville	Col_C
4	414	Patrick	33	M	Nantes	9
3	313	Didier	50	M	Nantes	9

RID^C	CID	Nom	Age	Genre	Ville	Col_C
2	212	Eric	40	F	Poitiers	10

Validation sous Oracle(III)

- Implémentation de la FHD en utilisant plus d'une table de dimension

Ventes

RID^S	CID	PID	TID	Montant	Col_F
1	616	106	11	25	1-1
2	616	106	66	28	1-2
3	616	104	33	50	1-1
4	515	104	11	10	4-1
5	212	101	33	27	9-2
6	212	101	66	19	10-2
7	212	106	11	25	1-2
8	111	101	33	27	1-1
9	111	101	66	19	10-1
10	111	106	11	25	9-1
11	515	101	33	27	9-1
12	515	101	66	19	9-2
13	515	106	11	25	4-2
14	414	101	33	27	4-2
15	414	101	66	19	10-2
16	414	106	11	25	1-2
17	313	101	33	27	10-2
18	313	101	66	19	4-1
19	313	106	11	25	1-1
20	313	104	33	50	1-2
21	313	104	66	20	10-1
22	313	104	11	50	10-2
23	313	104	22	20	10-1
24	313	104	66	20	9-2
25	313	105	66	19	9-1
26	313	105	22	17	9-1
27	313	106	11	15	9-2

RID^C	CID	Non
6	616	Gille
5	515	Yves
4	414	Patri
3	313	Didie
2	212	Eric
1	111	Pasc

RID^T	TID	M
6	11	Ja
5	22	Fé
4	33	M
3	44	A
2	55	M
1	66	J

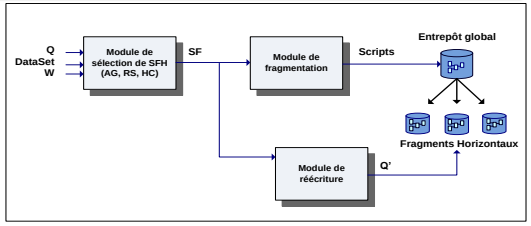
```
CREATE MATERIALIZED VIEW V
AS
SELECT v.CID, v.PID, v.TID, Montant, Col_C||'-'||Col_T as Col_F
FROM Ventes v, Client c, Temps t
WHERE v.CID= c.CID
AND v.TID =  t.TID
```

Ventes effectuées durant le premier trimestre par des clients masculins ayant moins de 26 ans et habitant Poitiers

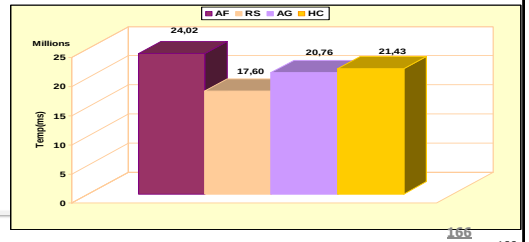
RID^S	CID	PID	TID	Montant	Col_F
1	616	106	11	25	1-1
3	616	104	33	50	1-1
8	111	101	33	27	1-1
19	616	104	22	20	1-1

Validation Sous Oracle (III)

- Architecture



- Résultats



SÉLECTION MULTIPLE DE TECHNIQUES D'OPTIMISATION

Classification des techniques

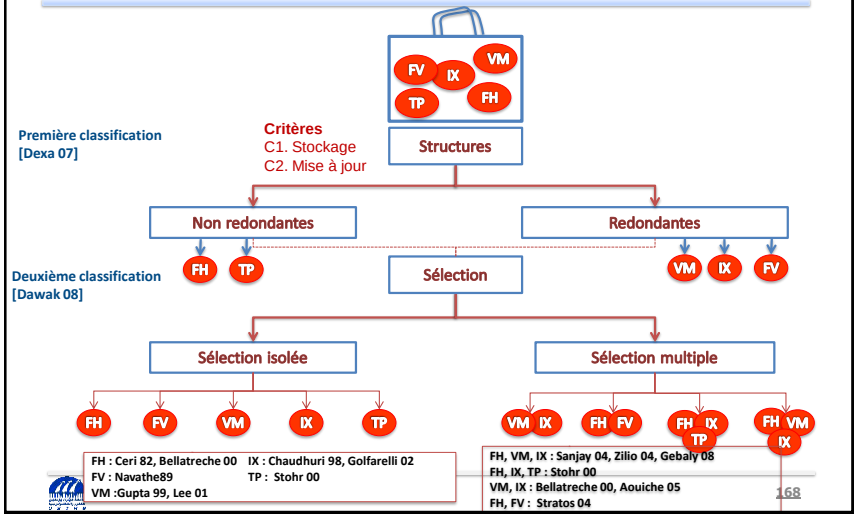
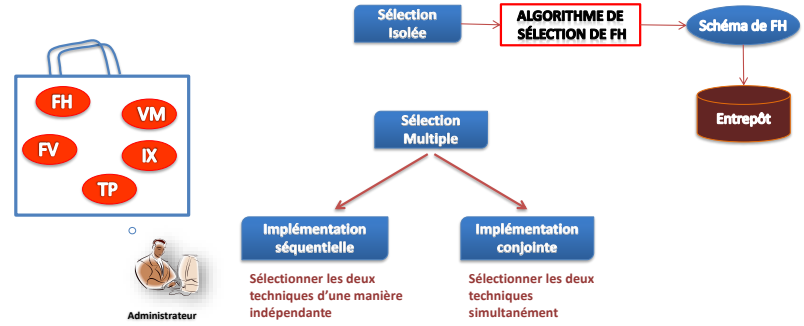


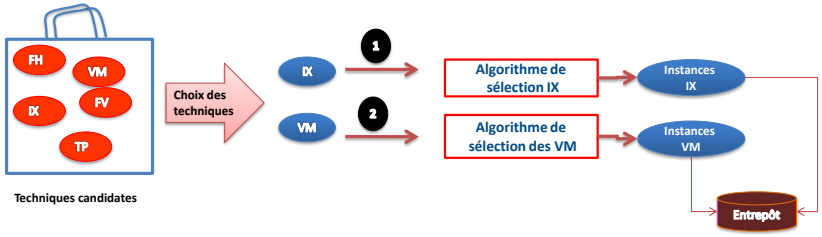
Illustration des sélection isolée et multiple

- Sélection isolée : choisir une seule technique
- Sélection multiple : choisir plusieurs techniques



Implémentation séquentielle

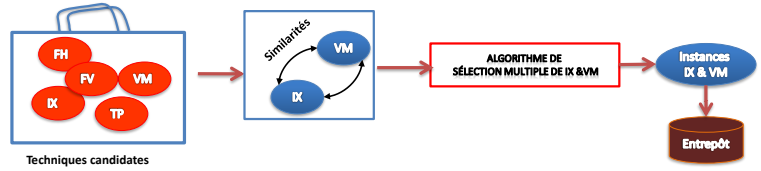
- Plusieurs sélections isolées



- **Inconvénient**
  - Absence de prise en compte des similarités et dépendances entre les techniques d'optimisation
  - Une vue peut être indexée ➔ Sélectionner des vues ensuite des index pourrait être un scénario pertinent (ordre de sélection).

Implémentation conjointe

- ❑ Considération d'un seul espace de recherche
- ❑ Utilisation d'un algorithme global
  - ➔ Adaptée pour les techniques ayant de fortes similarités (Vues Matérialisées, Index)

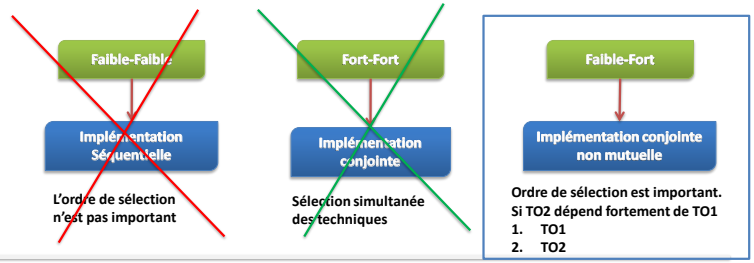


- ❑ Combinaison de plusieurs problèmes NP-Complets
  - ➔ Complexité élevée

Réduction de la complexité de l'implémentation conjointe

Deux types de dépendance [IBM'04]

- Dépendance forte
  - TO1 dépend fortement de TO2 : tout changement dans la sélection de TO2 ➔ changement dans la sélection de TO1.
    - **Exemple :** Si une vue V est sélectionnée, de nouveaux index sur V peuvent être sélectionnés.
- Dépendance faible
  - TO1 dépend faiblement de TO2 : changement dans la sélection de TO2 n'affecte pas la sélection de TO1.



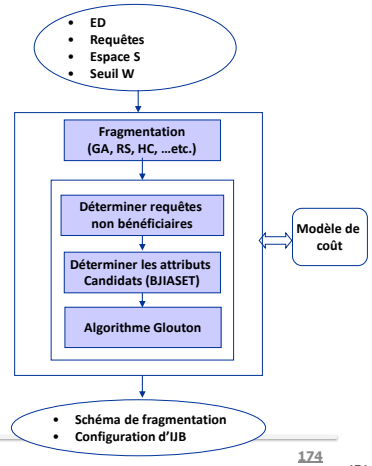
Sélection combinée : HP&BJI

- **La combinaison repose sur les similarités entre techniques d'optimisation**
- Approches de combinaison
  - Séquentielle : chaque technique est sélectionnée indépendamment de l'autre
    - **Exemple :** fragmentation horizontale et verticale
    - **Inconvénient :** Ne prends pas en compte les interactions entre les techniques
  - Combinée : les techniques sont sélectionnées simultanément
    - Un seul espace de recherche est considéré.
    - **Exemple :** la sélection des vues et des index.
    - **Avantage :** elle prend en compte les interactions inter-techniques
    - **Inconvénient :** la complexité du problème augmente.
  - Notre approche de combinaison
    - Utiliser la fragmentation horizontale pour élaguer (réduire) l'espace de recherche des index de jointure bitmap.

Exploitation de la similarité pour tuner

1. Sélectionner un schéma de fragmentation
2. Déterminer les requêtes non bénéficiaires
   - ❑ Utiliser le taux λ fixé par l'administrateur
   - ❑ si($\frac{Coût(Q,FS)}{Coût(Q,\phi)} \leq \lambda$) alors Q est bénéficiaire

   Coût(Q,ϕ) , **Coût(Q,FS)** : coût avant et après fragmentation
3. Déterminer les attributs candidats (BJIASET)
   - ❑ Attributs candidats = Attributs de faibles cardinalités dans les requêtes non bénéficiaires et non utilisés pour fragmenter l'entrepôt
4. Sélectionner un ensemble de BJIs avec l'algorithme glouton



Exemple

```
SELECT   count(*)
FROM  Ventes V, Client C, Produit P, Temps T
WHERE  V.CID = C.CID AND V.TID=T.TID AND V.PID=P.PID
AND    C.Ville  =  'Poitiers'  AND  T.Mois='Juin'  AND
P.Catégorie='Beauté'
```

❑Fragmentation sur Ville et Mois

❑Fragment valide pour la requête est Ventes_PJ : Ventes effectuées le mois de juin par des clients de Poitiers

$Client_1 = \sigma_{ville=Poitiers}(Client)$

RID^C	CID	Nom	Ville
6	616	Gilles	Poitiers
2	212	Eric	Poitiers
1	111	Pascal	Poitiers

$Temps_6 = \sigma_{mois=juin}(Temps)$

RID^T	TID	Mois	Année
1	66	Juin	2003

Produit

RID^P	PID	Nom	Catégorie
6	106	Sonoflore	Beauté
5	105	Clarins	Beauté
4	104	WebCam	Multimédia
3	103	Barbie	Jouet
2	102	Manure	Jardinage
1	101	SlimForm	Fitness

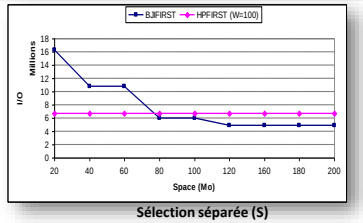
Ventes_PJ

RID^S	CID	PID	TID	Montant
2	616	106	66	28
16	111	105	66	44
17	212	104	66	40

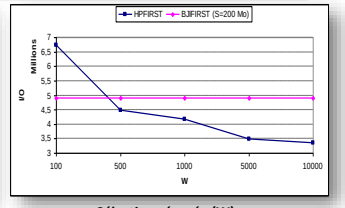
CAT_BJI

	Catégorie				
	B	M	J	Jr	F
2	1	0	0	0	0
16	1	0	0	0	0
17	0	1	0	0	0

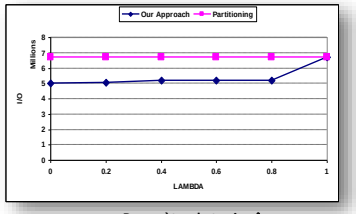
Validation



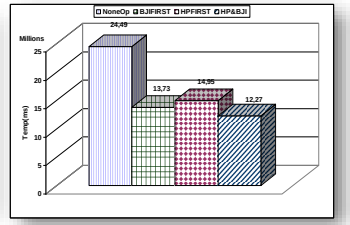
Sélection séparée (S)



Sélection séparée (W)



Paramètre de tuning λ



Validation Oracle 10g

LES ENTREPOTS DE DONNES
Data Warehouse

Kamel Boukhalfa
Boukhalk@gmail.com