Datawarehousing and OLAP

Datawarehousing

Syllabus, materials, notes, etc.

See http://www.info.univ-tours.fr/~marcel/dw.html

today

introduction

definition

data integration

model

introduction

vocabulaire

*D*ecision *S*upport *S*ystems	systèmes d'aide à la décision
*B*usiness *I*ntelligence	décisionnel
*D*ata *W*arehouse	entrepôt de données
*O*n-*L*ine *A*nalytical *P*rocessing	analyse en ligne
*K*nowledge *D*iscovery in *D*atabases	*E*xtraction de *C*onnaissances dans les *D*onnées
*D*ata *M*ining	fouille de données
*C*ustomer *R*elationship *M*anagement	*G*estion de la *R*elation *C*lient

contexte

Lyman and Varian, 2000, www.sims.berkeley.edu/how-much-info

- entre 1 et 2 ExaOctets par année (1 Eo $= 2^{20}$ To)
- 90% électronique
- taux de croissance annuel de 50 %

actualisation en 2003 : 5 Eo en 2002, 92 % électronique

contexte

"we are data rich..."

SGBD traditionnels

- applications commerciales
- importants volumes (Mo/Go)
- fondements mathématiques
- processus transactionnels en ligne
  (*O*n-*L*ine *T*ransactional *P*rocessings)

les processus OLTP

sont

- interactifs
- concurrents
- nombreux
- répétitifs
- structurés
- simples

les processus OLTP

sont

- interactifs
- concurrents
- nombreux
- répétitifs
- structurés
- simples

et concernent

- la mise à jour des données
- un nombre de tuples restreint
- des données détaillées et à jour

les processus OLTP

sont

- interactifs
- concurrents
- nombreux
- répétitifs
- structurés
- simples

et concernent

- la mise à jour des données
- un nombre de tuples restreint
- des données détaillées et à jour

exemple : une chaîne de supermarchés enregistrant ses ventes

nouveaux besoins

"... but information poor"

nourrir les systèmes d'aide à la décision avec un ensemble de BD

- exploration et analyse de données historisées
- énormes volumes de données (To)
- processus analytiques en ligne

(*O*n-*L*ine *A*nalytical *P*rocessing)

les processus OLAP

sont

- interactifs
- concurrents
- peu nombreux
- peu prévisibles
- complexes

les processus OLAP

sont

- interactifs
- concurrents
- peu nombreux
- peu prévisibles
- complexes

et concernent

- l'exploration des données
- un nombre de tuples très important
- des données consolidées et synthétiques

les processus OLAP

sont

- interactifs
- concurrents
- peu nombreux
- peu prévisibles
- complexes

et concernent

- l'exploration des données
- un nombre de tuples très important
- des données consolidées et synthétiques

exemple : une chaîne de supermarchés analysant l'ensemble de ses ventes

OLTP/OLAP

	OLTP	OLAP
orientation	transaction	analyse
utilisateur	DBA	décideur
modèle	E/R	star/snowflake
granularité	détail	résumé
vue	relationnelle	multidimensionnelle
unité de travail	transactions simples	requêtes complexes
accès	lecture/écriture	lecture
nombre de tuple accédés	dizaine	millions
nombre d'utilisateurs	milliers	centaines
unité	Mo/Go	Go/To
métrique	débit de transactions	temps de réponse

nouveaux besoins

traitements difficilement

- formulables

nouveaux besoins

traitements difficilement

- formulables
  - calculs non triviaux
  - utilisateur = décideur, pas technicien !

nouveaux besoins

traitements difficilement

- formulables
  - calculs non triviaux
  - utilisateur = décideur, pas technicien !
- évaluables

nouveaux besoins

traitements difficilement

- formulables
  - calculs non triviaux
  - utilisateur = décideur, pas technicien !
- évaluables
  - énormes volumes
  - beaucoup d'agrégations
  - beaucoup de jointures
  - .....

nouveaux besoins

traitements difficilement

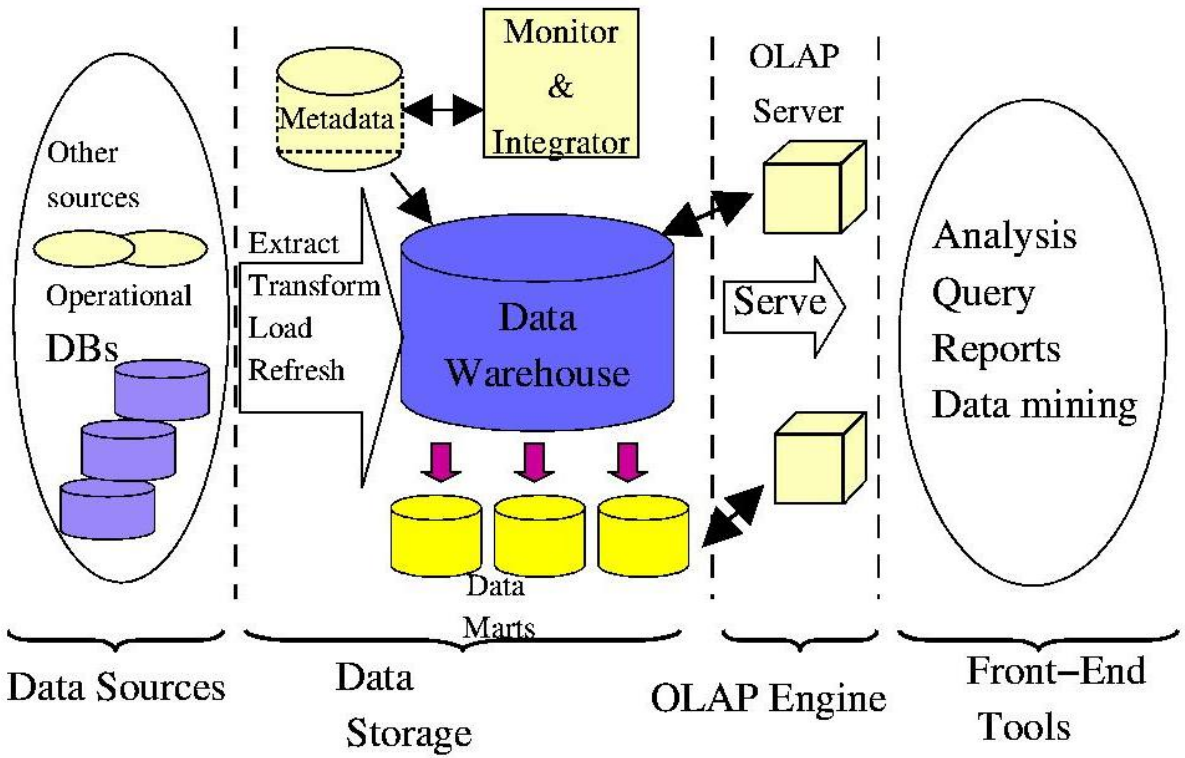
- formulables
  - calculs non triviaux
  - utilisateur = décideur, pas technicien !
- évaluables
  - énormes volumes
  - beaucoup d'agrégations
  - beaucoup de jointures
  - .....

avec la technologie classique

architecture multi-tiers

0	1	2	3
sources externes	entrepôt	OLAP	restitution
BD	SGBDR	SGBRD	tableur
fichiers HTML		SGBDM	data mining
fichiers plats		combinaison des 2	outils statistiques
...			...

architecture multi-tiers



nouvelles technologies

entrepôt de données

récolte, stockage et gestion efficace des gros volumes

nouvelles technologies

entrepôt de données

récolte, stockage et gestion efficace des gros volumes

OLAP

requêtes interactives complexes sur ces volumes

nouvelles technologies

entrepôt de données
: récolte, stockage et gestion efficace des gros volumes

OLAP
: requêtes interactives complexes sur ces volumes

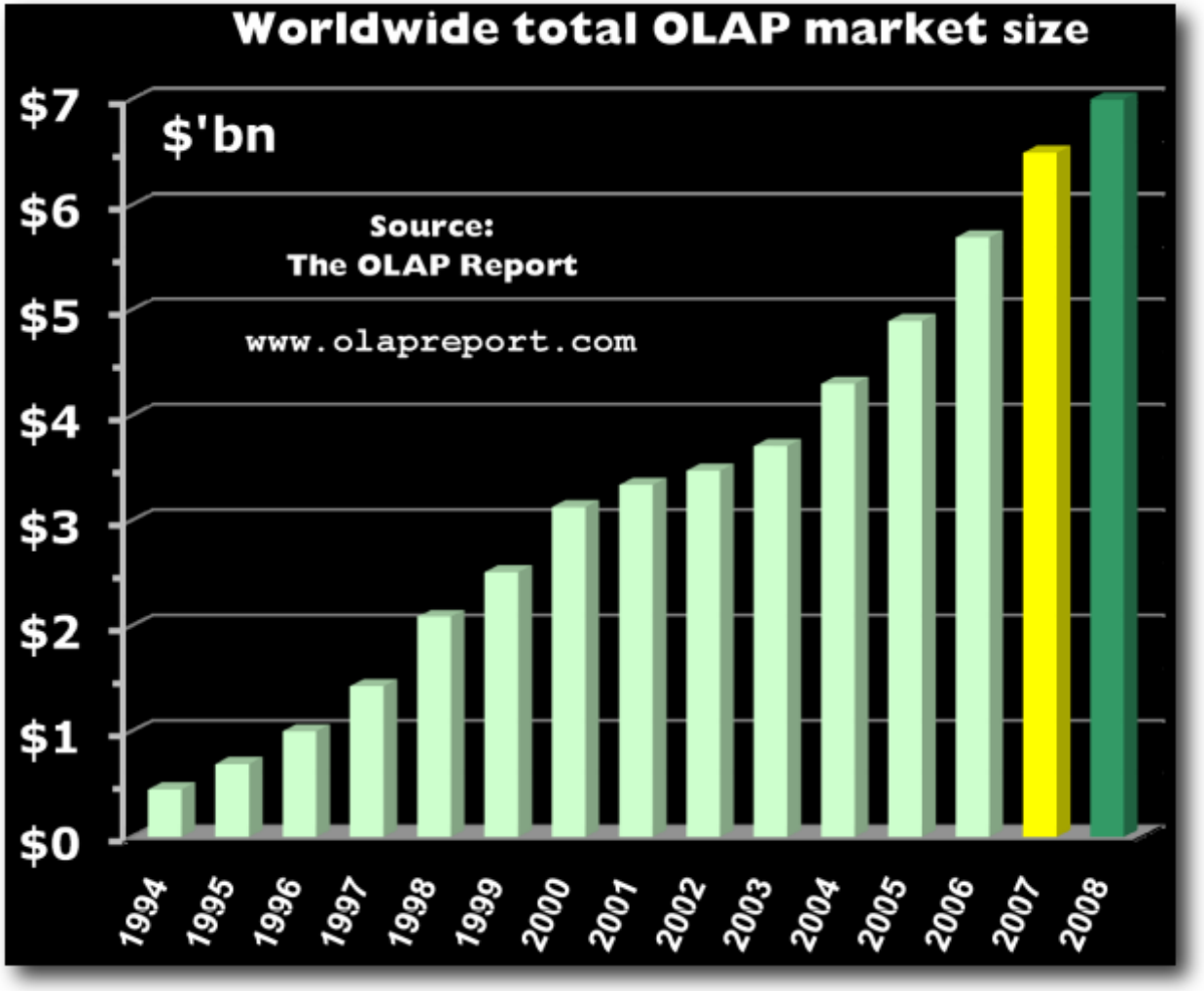
data mining
: extraction automatique de propriétés cachées

domaines d'application

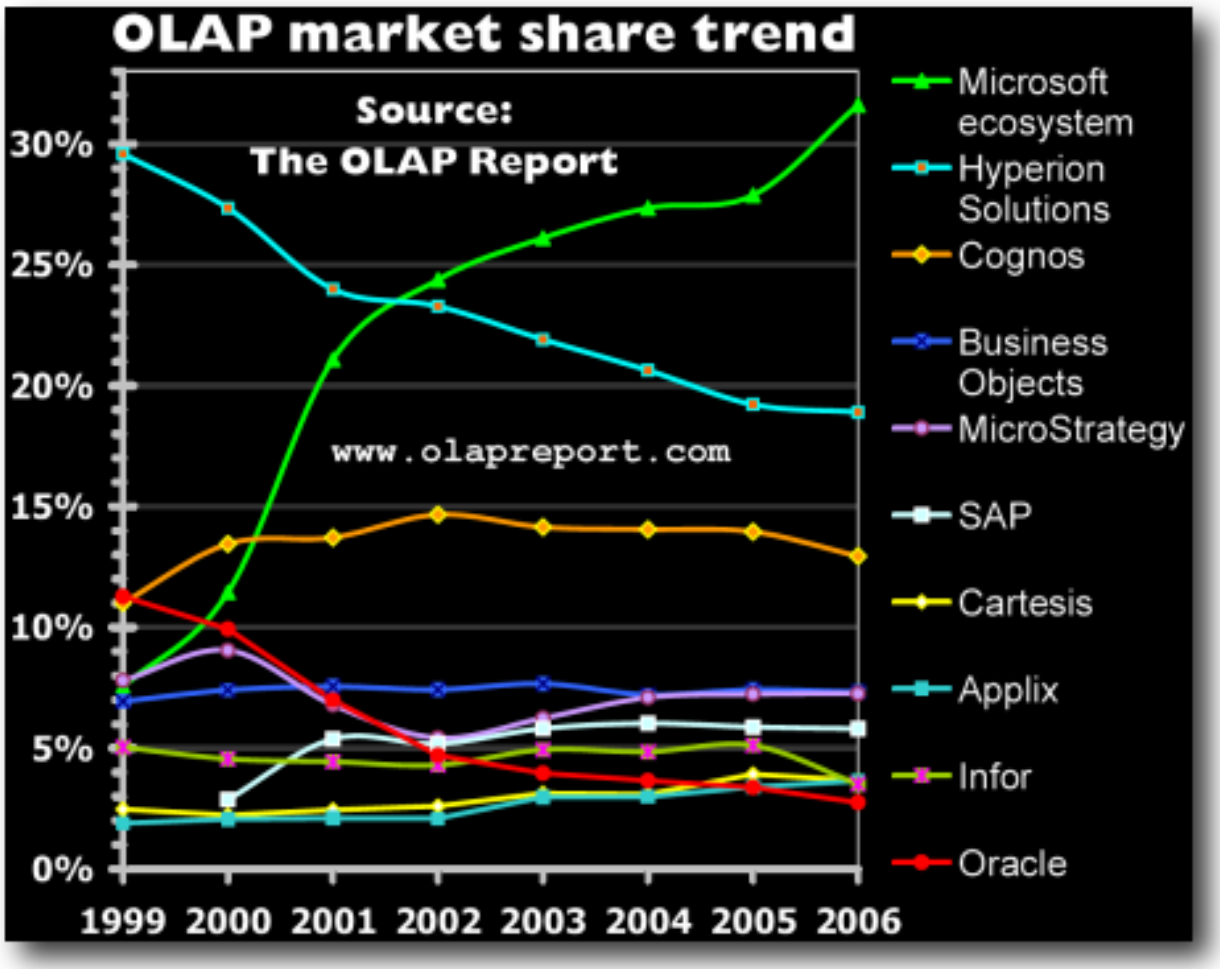
commercial, financier, transport, télécommunications, santé, services, ...

- gestion de la relation client
- gestion de commandes, de stocks
- prévisions de ventes
- définition de profil utilisateur
- analyse de transactions bancaires
- détection de fraudes
- ...

market news

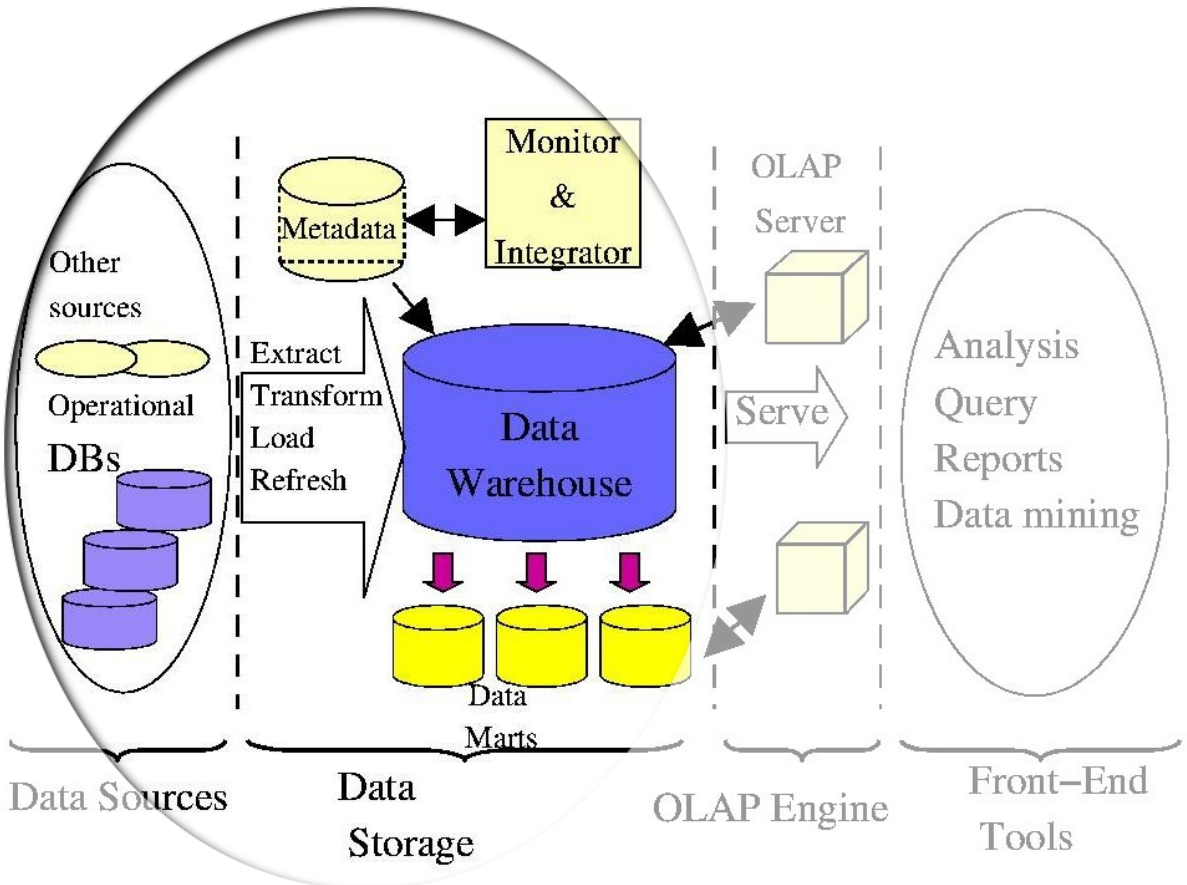

Worldwide total OLAP market size
$'bn
Source:
The OLAP Report
www.olapreport.com
$7
$6
$5
$4
$3
$2
$1
$0
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008

market news


OLAP market share trend
Source:
The OLAP Report
www.olapreport.com
30%
25%
20%
15%
10%
5%
0%
1999 2000 2001 2002 2003 2004 2005 2006
Microsoft ecosystem
Hyperion Solutions
Cognos
Business Objects
MicroStrategy
SAP
Cartesis
Applix
Infor
Oracle

Data Warehousing

Data Warehousing



data warehousing

1. definition
2. data integration
3. modeling

Definition

contexte

premières solutions commerciales (début 90)

- peu flexibles (ad hoc)
- peu pratiques / limitées
- sans fondement mathématique

enjeux

- économiques
  - stratégie commerciale
- scientifiques
  - nombreux problèmes de recherche

contexte

secteur récent et à la mode

- besoins provenant de l'industrie
- la recherche est très active depuis 95
- IBM, Microsoft, Oracle, ... adaptent leurs produits

les concensus sont parfois difficiles à obtenir
les standards commencent juste à émerger

définition 1

industrie (Inmon 1992)

collection de données

- *orientées sur un sujet*
- *intégrées de différentes sources*
- *non volatiles*
- *historisées*

définition 2

recherche (Stanford 1995)

dispositif de stockage d'informations intégrées de sources distribuées, autonomes, hétérogènes

données sources

- BD
  - relationnelles
  - objets
  - réseau
  - ...
- fichiers (flat files)
- documents HTML, XML
- bases de connaissances
- .....

données sources

- BD
  - relationnelles
  - objets
  - réseau
  - ...
- fichiers (flat files)
- documents HTML, XML
- bases de connaissances
- .....

généralement
modifiées
quotidiennement

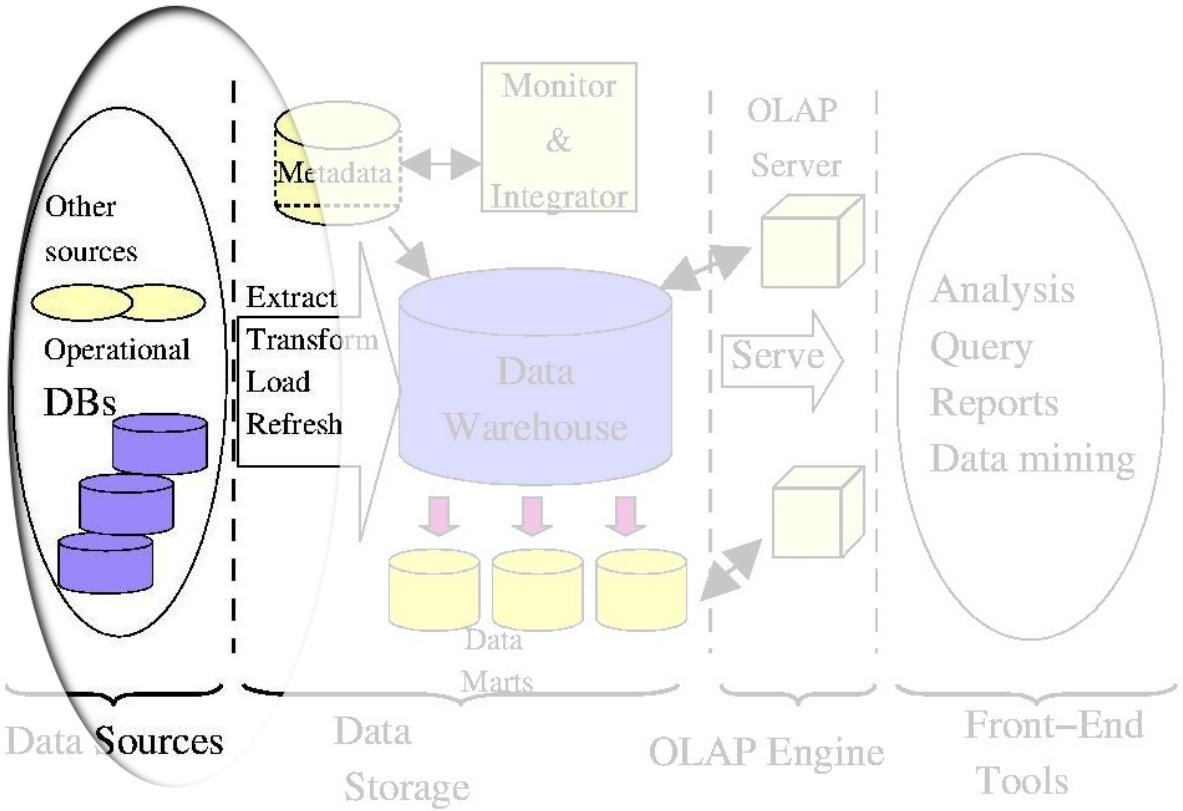
hétérogénéité des données

Goglin, 1998

source d'information	environnement
gestion commerciale	progiciel sybase/unix
gestion marketing	progiciel SQL server/NT
gestion financière, paye	mainframe DB2/IBM
suivi de production	oracle/NT
contrôle qualité	oracle/NT
gestion du temps	progiciel oracle/unix
gestion des stocks	progiciel oracle/HP
fichier mailings	fichier ASCII
références nationales	document excel

data integration

data integration



problématique : intégration de données

approche paresseuse (lazzy, query-driven)

- requête dirigée dynamiquement vers la cible
- avantages :
  - des données à jour
  - pas de duplication de données
- inconvénients : la requête
  - nécessite un traitement supplémentaire
  - est en compétition avec les requêtes locales

n'avait pas rencontré de succès dans l'industrie...

problématique : intégration de données

approche paresseuse (lazzy, query-driven)

- requête dirigée dynamiquement vers la cible
- avantages :
  - des données à jour
  - pas de duplication de données
- inconvénients : la requête
  - nécessite un traitement supplémentaire
  - est en compétition avec les requêtes locales

n'avait pas rencontré de succès dans l'industrie...
jusqu'à l'apparition des EII (Enterprise Information Integration)
... à suivre

problématique : intégration de données

intégration par avance (eager, update-driven)

- requête évaluée sur le résultat de l'intégration
- avantages :
  - indépendance vis à vis des sources
  - performances de l'évaluation de requêtes
- inconvénient :
  - duplication de données
  - nécessité d'un rafraîchissement

a rencontré du succès dans l'industrie

data integration

- database federations (outdated)
- multibase architecture (query driven)
- mediation (query driven)
- data warehouse (update driven)

problème des sources hétérogènes

chaîne de concessionnaires

concession 1	véhicules(série, modèle, couleur, autoradio, ...)
	ex : véhicules(“1234” , “206 xrp” , “rouge” , “abs.” , ...)
concession 2	automobiles(num_série, modèle, couleur)
	options(num_série, option)
	ex : automobiles(1234, “206” , “r”)
	ex : automobiles(1233, “206” , “r”)
	ex : options(1234, “abs.”)
...	

sources hétérogènes

pour un même concept

- schémas différents
- noms d'attribut différents
- types de données différents
- valeurs différentes
- sémantiques différentes

bases de données fédérées

- chaque BD est reliée aux BD avec lesquelles elle doit communiquer
- demande dynamique d'information
- traductions locales
- $n(n-1)$ connexions si n BD veulent communiquer
- intéressant quand la communication entre les bases est limitée

architecture multibase

- collection de BDs
- pas de schéma global
- un langage multibase
  - définition de schémas multibases
  - définition des dépendances entre les bases
  - définition des requêtes

exemple

BD bnp

br(num_br, nom_br, ville, rue, tel)

compte(num_compte, num_cl, solde, num_br)

client (num_cl, nom_cl, tel_, type_cl, adresse_cl)

spe-acc (num_compte, num_br, num_cl, solde, devise)

exemple

BD sg

branche (nbranche, nombranche, localité, rue)

cpt(numcpt, nbranche, numc, solde)

client (numc, nomc, adressec)

exemple

BD cic

branche(<u>num_br</u>, nom_br, ville, rue, tel)

compte(<u>num_cpt</u>, <u>num_br</u>, <u>num_cl</u>, solde, date_ouverture)

client(<u>num_cl</u>, nom_cl, tel_cl, type_cl, adresse_cl)

exemple

CREATE MULTIDATABASE Banques (bnp cic sg);

```
USE      bnp, cic
SELECT   bnp.br.nom_br, cic.branche.nom_br
FROM     bnp.br, cic.branche
WHERE    bnp.br.ville = cic.branche.ville;
```

```
USE      bnp, cic
SELECT   *
FROM     br%
WHERE    ville = 'Paris';
```

exemple

```
USE     Banques
LET     x BE ville localité
LET     y BE sg bnp
SELECT  A.*
FROM    y.b% A, cic.b% B
WHERE   B.x = 'Paris'
AND     B.rue = A.rue
AND     A.x = 'Paris';
```

médiation

- schéma global (mediator)
- pas de stockage de données (définition de vues)
- traduction dynamique des requêtes
- suppose une interface avec chaque source (wrapper)

exemple

autosMed(num_série, modèle, couleur, autoradio, concession)

```
SELECT   num_série, modèle
FROM     autosMed
WHERE    couleur = "rouge"
```

wrapper 1 (pour concession 1) :

```
SELECT   série, modèle
FROM     véhicules
WHERE    couleur = "rouge"
```

wrapper 2 (pour concession 2) :

```
SELECT   num_série, modèle
FROM     automobiles
WHERE    couleur = "r"
```

conception des wrappers

collection de requêtes paramétrées (*template*)

$Q_1[p_1^1, p_1^2, \ldots] \rightarrow Q'_1[p_1^1, p_1^2, \ldots]$
$Q_2[p_2^1, p_2^2, \ldots] \rightarrow Q'_2[p_2^1, p_2^2, \ldots]$
...

paramètres p_i^j fournis par le médiator
réécriture de la requête du mediator en requête source après substitution des paramètres

exemple

template T_1 pour le wrapper du concessionnaire 1

```
SELECT   *
FROM     autosMed
WHERE    couleur = $c
→
SELECT   série, modèle, couleur, autoradio, "concessionnaire 1"
FROM     véhicules
WHERE    couleur = $c
```

exemple

modèlesRouges(num_série, modèle, autoradio)

template T_2 pour le wrapper du concessionnaire 1

```
SELECT   COUNT(num_série)
FROM     modèlesRouges
```

$\longrightarrow$

```
SELECT   COUNT(série)
FROM     véhicules
WHERE    couleur = "rouge"
```

fonctionnement

un wrapper = une table de templates + un driver

- réception d'une requête du mediator
- recherche d'un template correspondant dans la table
- instanciation d'une requête source
- envoi de la requête source à la source
- réception du résultat
- envoi du résultat au mediator

optimisation des wrappers

éviter la multiplication des templates

- utiliser des templates retournant un sur-ensemble de la réponse
- filtrer les résultats

suppose tester la contenance de requête !

exemple

```
SELECT   *
FROM     autosMed
WHERE    couleur = "rouge" AND modèle = "206"
```

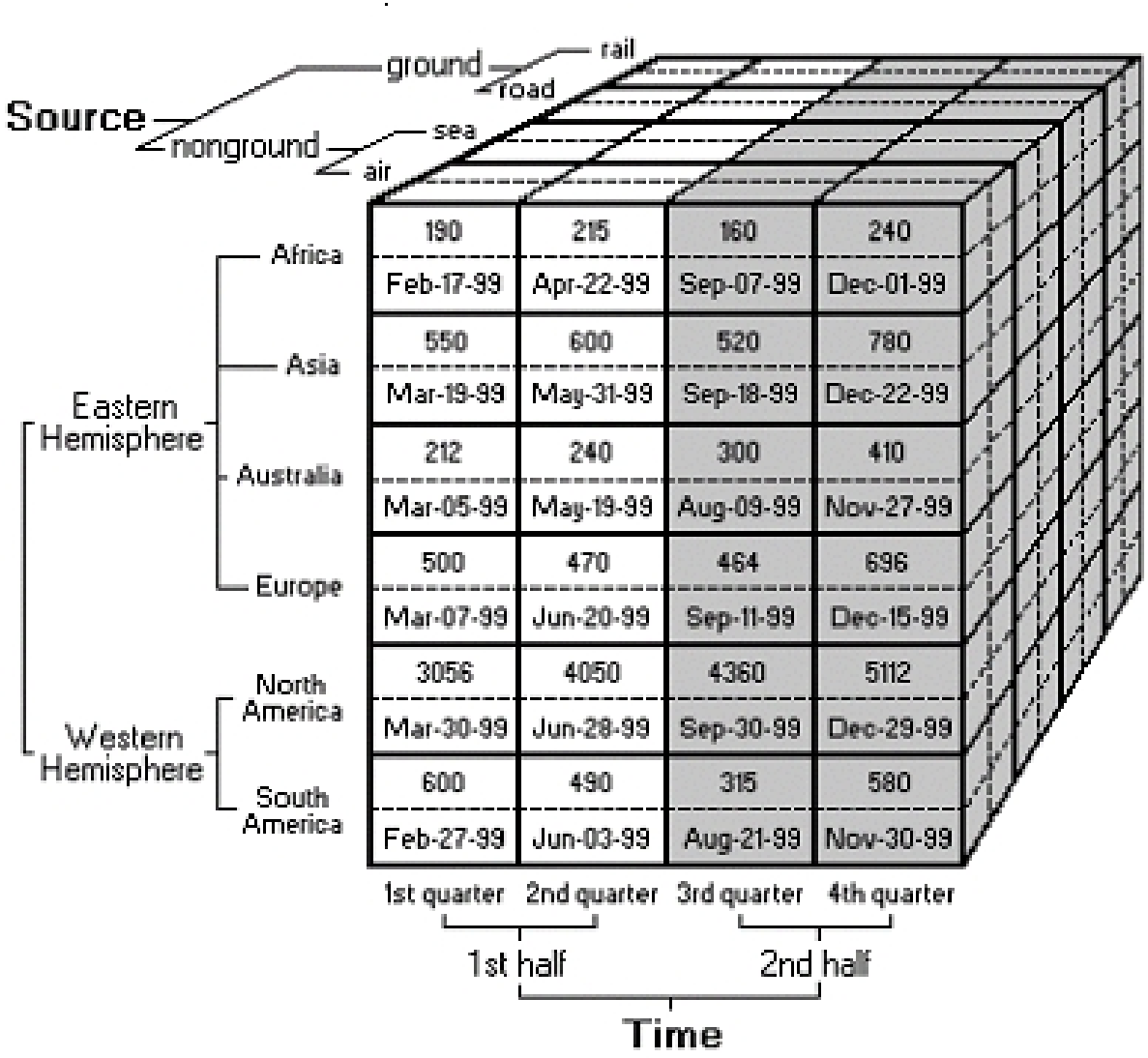
1. utiliser le template T_1 avec \$c = "rouge"
2. stocker le résultat dans une relation temporaire
   tempAutos(série, modèle, couleur, autoradio, concessionnaire)
3. sélectionner les modèle "206 xrp" avec la requête
   ```
   SELECT   *
   FROM     tempAutos
   WHERE    modèle = "206 xrp"
   ```

entreposage de données

- données sources combinées dans un schéma global
- données stockées dans une BD de l'entrepôt
- mise à jour sur les sources exclusivement
- rafraichissement périodique

modeling

cube de données



niveau conceptuel

schéma de BD relationnelle reflétant la vue de l'analyste :

- multidimensionnelle
- hiérarchisée

- schéma en étoile (star schema)
- schéma en flocon (snowflake schema)
- constellation de faits (fact constellation)

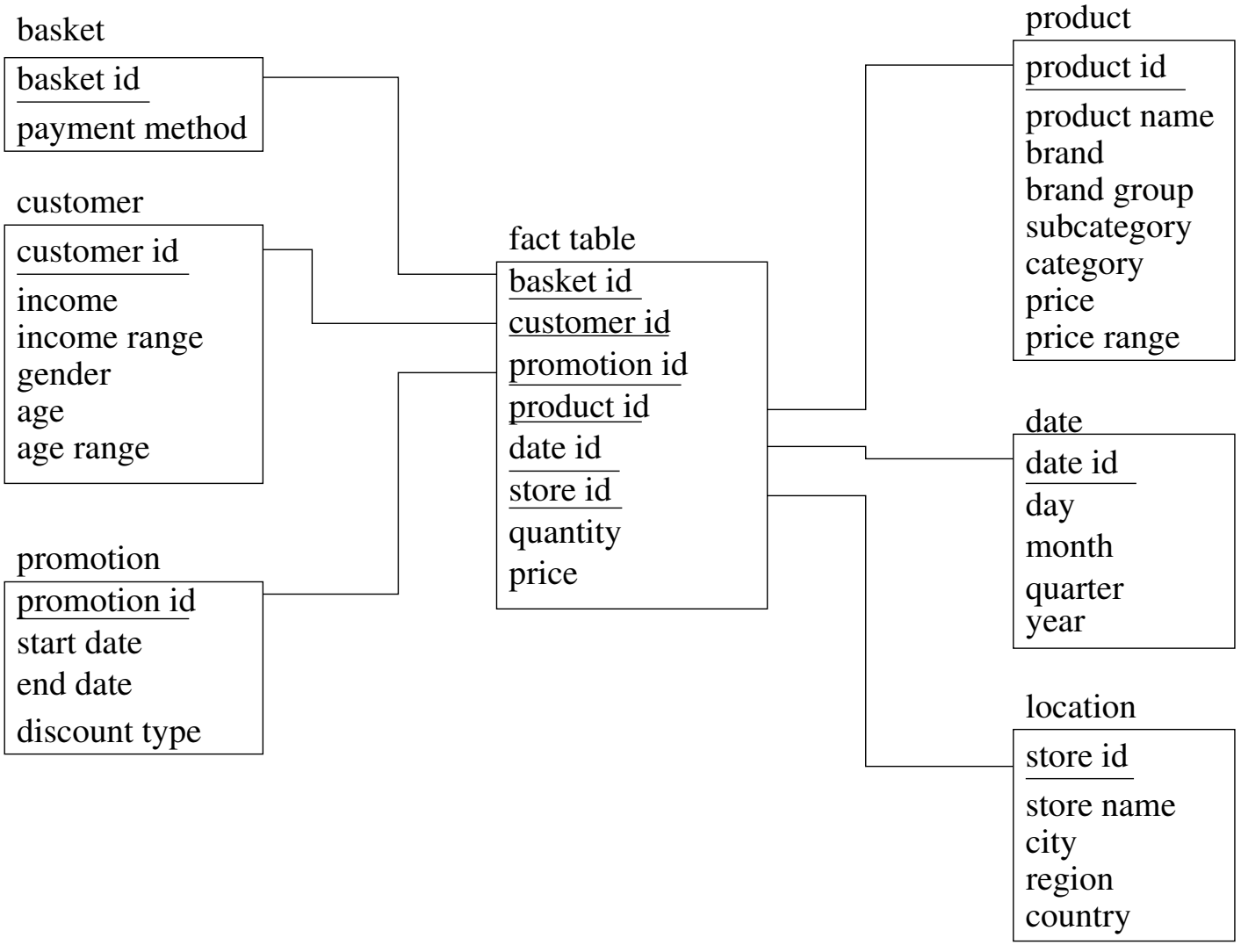
le schéma en étoile est souvent utilisé pour l'implantation physique

star schema

structure simple utilisant le modèle entité-relation

- une entité centrale (table des faits)
  - objets de l'analyse
  - taille très importante
  - beaucoup de champs
- des entités périphériques (tables de dimensions)
  - critères de l'analyse
  - taille peu importante
  - peu de champs

star schema


basket
basket id
payment method
customer
customer id
income
income range
gender
age
age range
promotion
promotion id
start date
end date
discount type
fact table
basket id
customer id
promotion id
product id
date id
store id
quantity
price
product
product id
product name
brand
brand group
subcategory
category
price
price range
date
date id
day
month
quarter
year
location
store id
store name
city
region
country

star schema

un fait :

il a été acheté 3 exemplaires à 1 euro

- du produit pid_3
- par le client cid_1
- à la date did_3
- dans le magasin mid_2
- dans le chariot cid_8
- correspondant à la promotion $prid_1$

star schema

un élément de la dimension *location* :

- store id mid_2
- store name *rondpoint*
- city *blois*
- region *centre*
- country *france*

star schema

attributs de la table des faits

- des clés étrangères formant une clé primaire
- des *mesures* associées à chaque clé primaire

association de type (0,n) - (1,1) connectant les différentes dimensions aux faits

normalisation

- table des faits en Boyce-Codd Normal Form
- tables de dimensions non normalisées

normalisation

- table des faits en Boyce-Codd Normal Form
- tables de dimensions non normalisées

une relation r est en BCNF si
$\forall x \rightarrow y$ DF définie sur r, x contient une clé de r

normalisation

- table des faits en Boyce-Codd Normal Form
- tables de dimensions non normalisées

une relation r est en BCNF si
$\forall x \longrightarrow y$ DF définie sur r, x contient une clé de r

chaque attribut non clé dépend fonctionnellement de la seule clé de la relation

les tables de dimensions

- représentent une ou plusieurs hiérarchies
- contiennent des données redondantes

les tables de dimensions

- représentent une ou plusieurs hiérarchies
- contiennent des données redondantes

faut-il les normaliser ?

les tables de dimensions

- représentent une ou plusieurs hiérarchies
- contiennent des données redondantes

faut-il les normaliser ?

- la table des faits constitue l'essentiel du stockage
- pas/peu de mises à jour des dimensions
- la perte d'espace n'est donc pas significative

snowflake schema

évolution du star schema

- normalisation des tables de dimensions

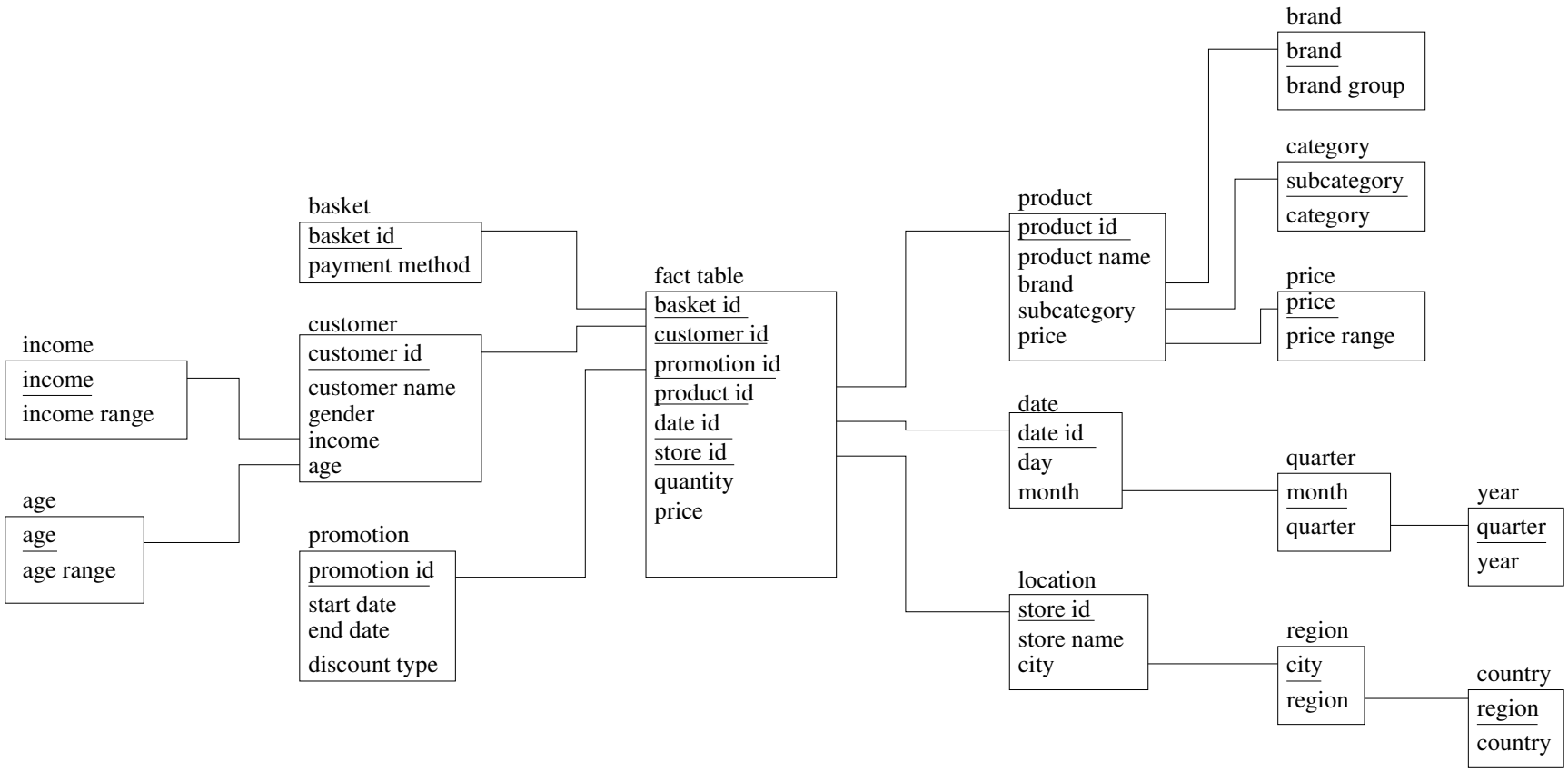
structure hiérarchique des dimensions

un *niveau* inférieur identifie un *niveau* supérieur

snowflake schema

- avantage
  - maintenance des tables de dimensions simplifiée
  - réduction de la redondance
- inconvénient
  - navigation coûteuse

snowflake schema



fact constellation schema

généralisation du star schema

- plusieurs tables des faits
- partage de tables de dimensions

en général

- fact constellation schema pour l'entrepôt
- une étoile de la constellation pour un magasin de données (data mart)

pré-agrégations

agrégation des faits selon une ou plusieurs dimensions

2 moyens de les représenter :

1. une table des faits séparée/dédiée avec les tables pour les dimensions correspondantes
2. dans la même table des faits, en codant les niveaux hiérarchiques dans les tables de dimensions

exemple

cas 1
faits1(idProduit,idVille,idJour,5)
faits2(idProduit,idVille,idMois,60)
avec une table *jour* et une table *mois*

cas 2
faits(idProduit,idVille,idDate1,5)
faits(idProduit,idVille,idDate2,5)

avec une table *date* contenant

date(idDate1, 22, 01, 2000)
date(idDate2, ALL, 01, 2000)

conclusion

So far: heterogeneous data sources $\longrightarrow$ a cube

Next: how to go from the sources to the cube