

Gestion de Masse de Données (GMD)

Cours 4 : Regrouper et interroger les données

II. Transformer les données

② Pour gérer les données manquantes ou bruitées

a) Données manquantes

- La **qualité** des données : impacts sur l'analyse, la fouille, etc.
- Identifier le type des données manquantes
 - données manquantes pour des variables dépendantes VS. indépendantes
 - apparition des données manquantes de façon aléatoire VS. non aléatoire
 - l'absence de valeur d'une variable indépendante est aléatoire
 - c'est rarement le cas en pratique : l'apparition de données manquantes est souvent associée à un phénomène particulier. *Ex: les données sont manquantes en raison de : localisation géographique (ex US State hors des Etats Unis), éducation, âge, le manque d'information, etc.*
 - données manquantes de façon "voulue"
 - non demandé, non applicable (ex : sous contraceptif oral pour les hommes ? enceinte ?)
 - ce sont des cas particuliers de données manquantes de façon non-aléatoire
 - proportion de données manquantes et répartition de ces données
 - quelle proportion ? 1%, 40%
 - quelle répartition ?
 - bcp de données manquantes pour peu de n-uplets ?
Ex : 20 individus sur 5000 dont 40% des données manquent ; ou 2000 avec 8% de données manquantes
 - bcp de données manquantes pour certains attributs ?
Ex : 2 attributs sur 15 avec 80% de données manquantes ; ou 10 avec 2% de données manquantes

II. Transformer les données

② Pour gérer les données manquantes ou bruitées

a) Données manquantes

- Méthodes pour traiter les données manquantes
 - Abandonner des n-uplets
 - réduit le nombre de cas (d'une étude par exemple)
 - l'échantillon final peut ne pas être représentatif ou ne pas être significatif
 - Abandonner un attribut : dans le cas où bcp de ses valeurs sont manquantes
 - évidemment il faut espérer que cet attribut n'est pas trop important !
 - créer une catégorie pour les données manquantes (Attributs nominaux *ONLY*)
 - ex : la variable *gender* peut prendre les valeurs *male*, *female*, *unspecified*
 - Imputer une valeur aux valeurs manquantes
 - ex : remplacer par la moyenne de l'ensemble des valeurs présentes
 - ex : la moyenne d'un sous ensemble des valeurs (ex : homme/femme)
 - ex : remplacer par une valeur attendue (si des règles $x \rightarrow y$ si $x \rightarrow ?$ on peut remplacer ? par y)
 - ex : méthodes plus sophistiquées : les moindres carrés, régression, etc.

Il n'y pas de bonne méthode ... ce sont des méthodes existantes qui sont utilisées...

Exemple

Case #	Y	X1	X2	X3
1	30	2	Missing	12
2	37	2	1	Missing
3	41	3	1	20
4	42	1	Missing	16
5	45	3	2	Missing
6	49	1	2	27
7	51	Missing	1	30
8	55	3	2	33
9	58	Missing	2	19
10	60	2	Missing	24

II. Transformer les données

② Pour gérer les données manquantes ou bruitées

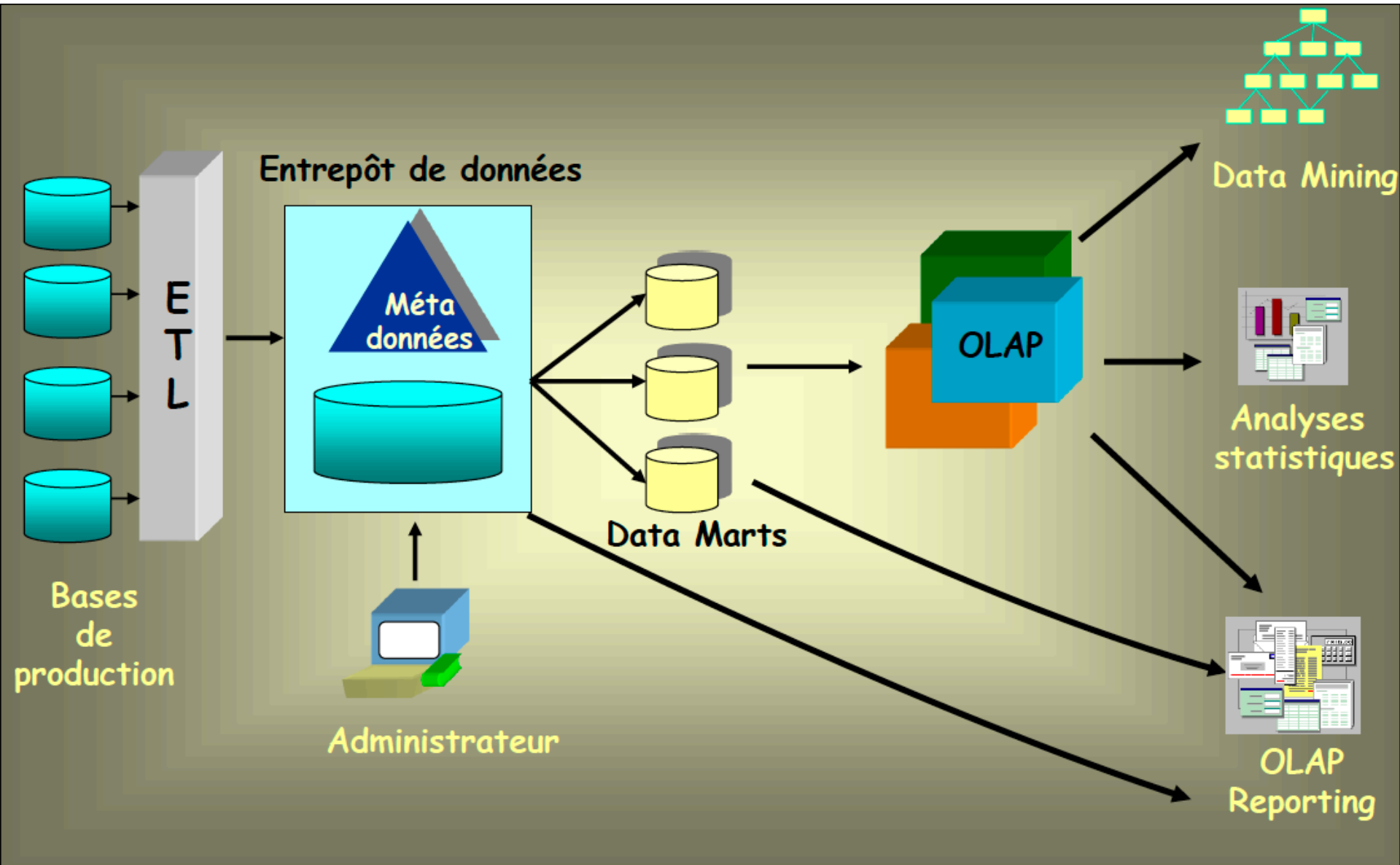
b) Données bruitées

- ... et les données bruitées ?
 - C'est également un problème à considérer surtout si vous avez des données
 - mesurées par un appareil
 - entrées par un humain : le taux d'erreur est à prendre en considération
 - Il existe des outils pour réduire le biais causé par le bruit
 - idées souvent similaires au traitement des données manquantes
 - à **vous** de voir si vous en avez besoin

III. Regrouper et interroger les données

① les Entrepôt de Données (noté ED ou *DW*)

- **Objectif usuel** : aide à la décision – terme "*business intelligence*"
 - décision vs production pour les BD classiques
- **Produits** :
 - Oracle
 - MS SQL Server
- **Architecture générale** (slide suivante)



- extraction
- transformation
- loading
- administration

- conception
- modélisation
- structuration

- analyse
- restitution

III. Regrouper et interroger les données

① les Entrepôt de Données (noté ED ou DW)

- Modélisation des données
 - modèle en étoile
 - 1 table de fait
 - n tables de dimensions
 - granularité figée par les dimensions
 - modèle en flocon
 - modèle en constellation

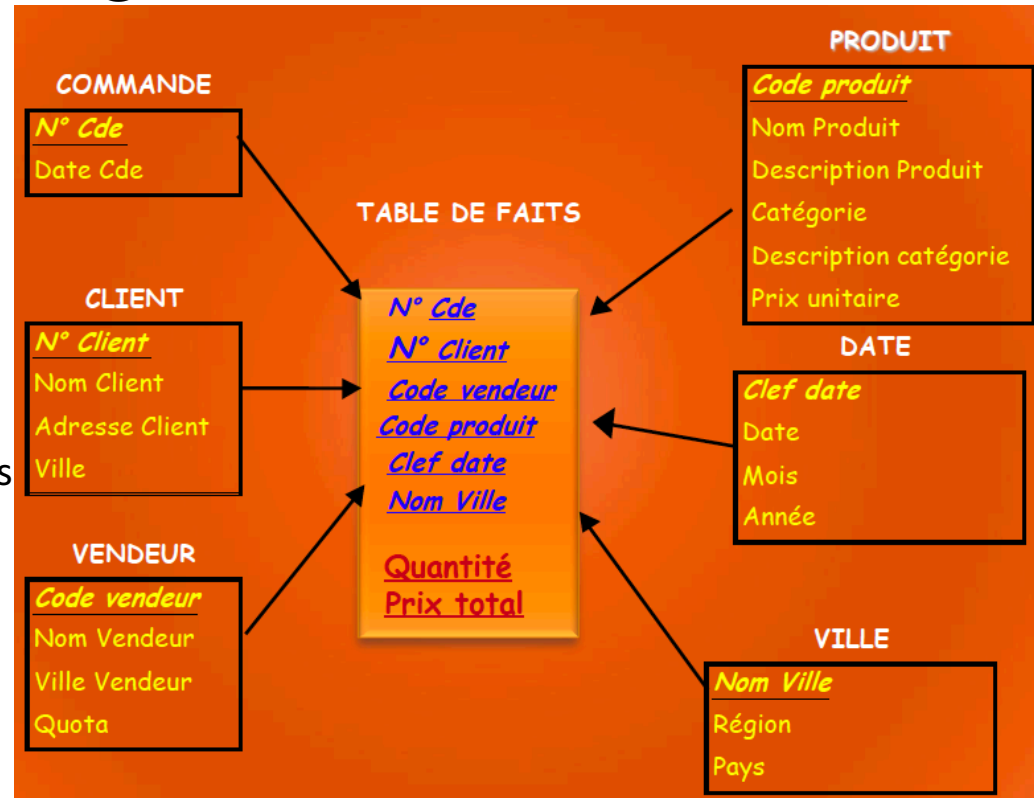


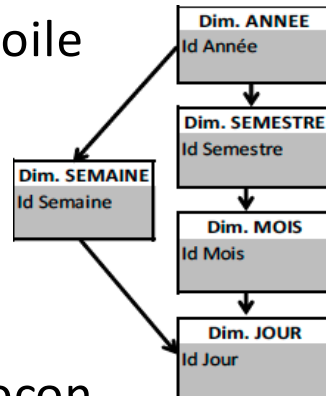
figure de Omar Boussaïd, Univ de Lyon

III. Regrouper et interroger les données

① les Entrepôt de Données (noté ED ou DW)

- Modélisation des données

- modèle en étoile



- modèle en flocon

- certaines dimension sont organisées suivant une hiérarchie
 - permet une analyse par palier (drill down)

- modèle en constellation

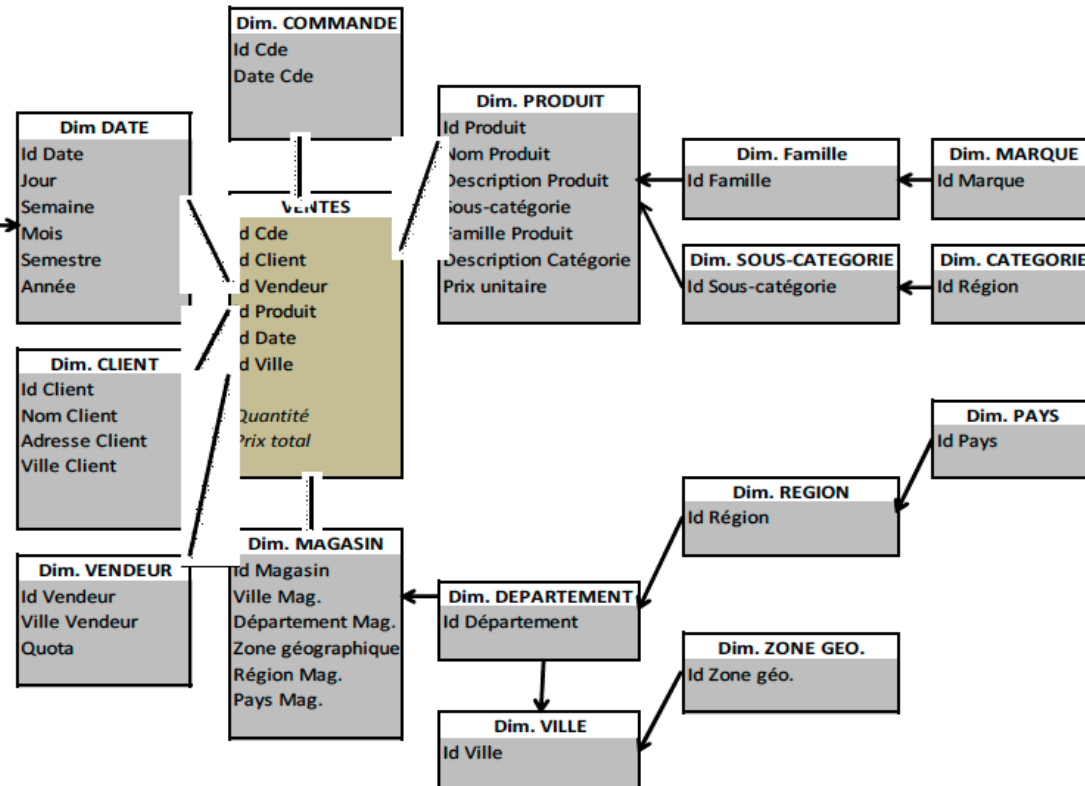


figure de Omar Boussaïd, Univ de Lyon

III. Regrouper et interroger les données

① les Entrepôt de Données (noté ED ou *DW*)

- Modélisation des données
 - modèle en étoile
 - modèle en flocon
 - modèle en constellation
 - fusion de modèles en étoile
 - dimensions partagées

III. Regrouper et interroger les données

① les Entrepôt de Données (noté ED ou *DW*)

- **OLAP On Line Analytical Processing** (vs Transactional Processing in BD)

- Principe : voir les données à travers plusieurs dimensions (on parle de cube OLAP quand $D = 3$)

Produit	Region	Ventes
Clous	Est	50
Clous	Ouest	60
Clous	Centre	100
Vis	Est	40
Vis	Ouest	70
Vis	Centre	80
Boulons	Est	90
Boulons	Ouest	120
Boulons	Centre	140
Nettoyeurs	Est	20
Nettoyeurs	Ouest	10
Nettoyeurs	Centre	30

Représentation des données dans une table relationnelle

	Est	Ouest	Centre
Clous	50	60	100
Vis	40	70	80
Boulons	90	120	140
Nettoyeurs	20	10	30

Représentation des données dans un tableau multidimensionnel

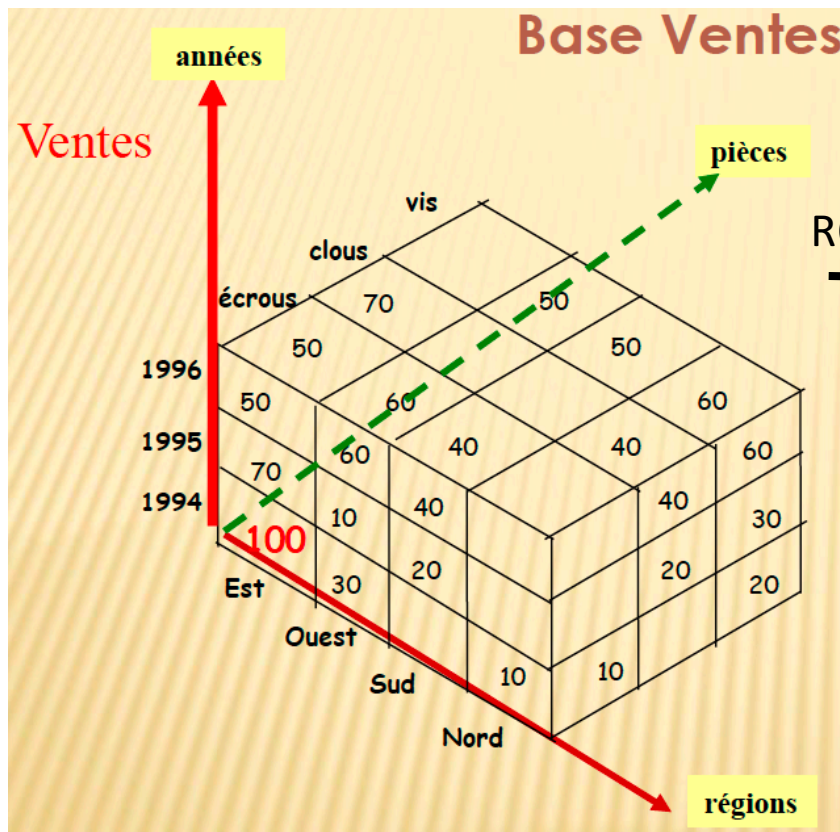
figure de Omar Boussaïd, Univ de Lyon

III. Regrouper et interroger les données

① les Entrepôt de Données (noté ED ou DW)

• OLAP

- Opérations principales :
 - agrégation de données (ROLL-UP),
 - forage de données (DRILL-DOWN)



ROLL UP

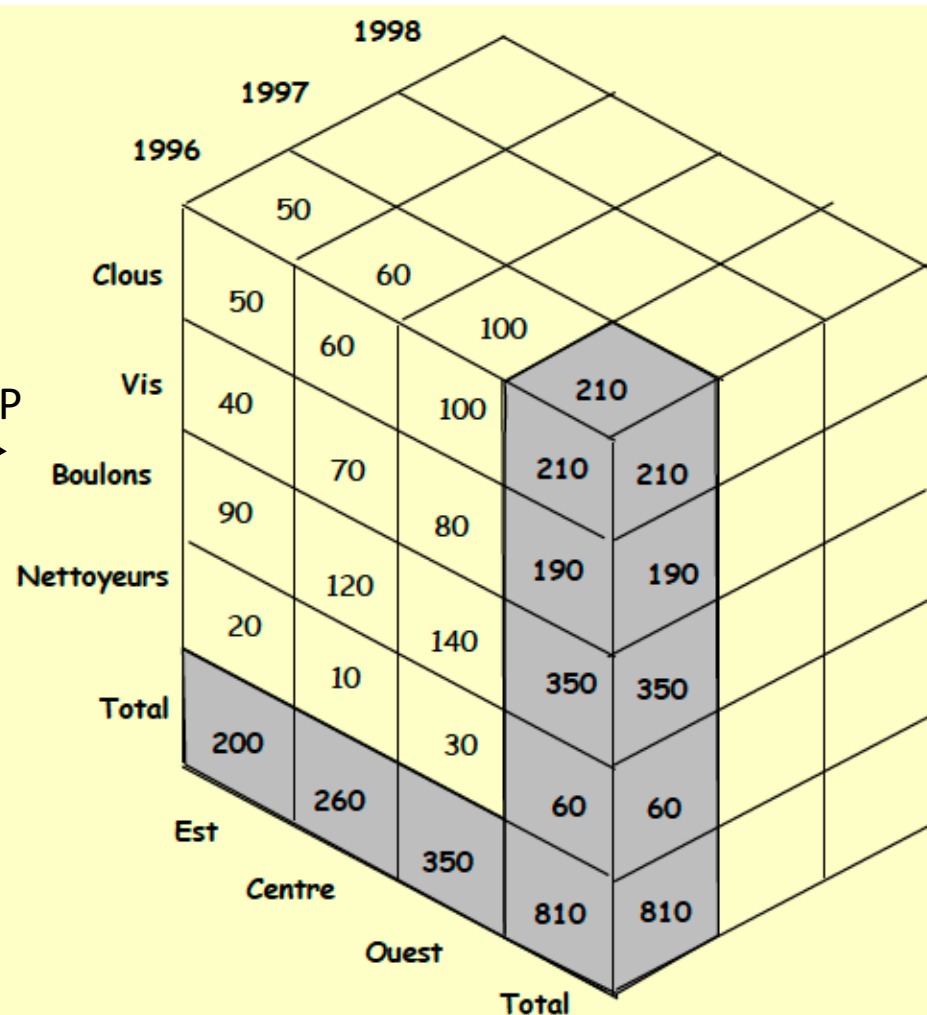


figure de Omar Boussaïd, Univ de Lyon

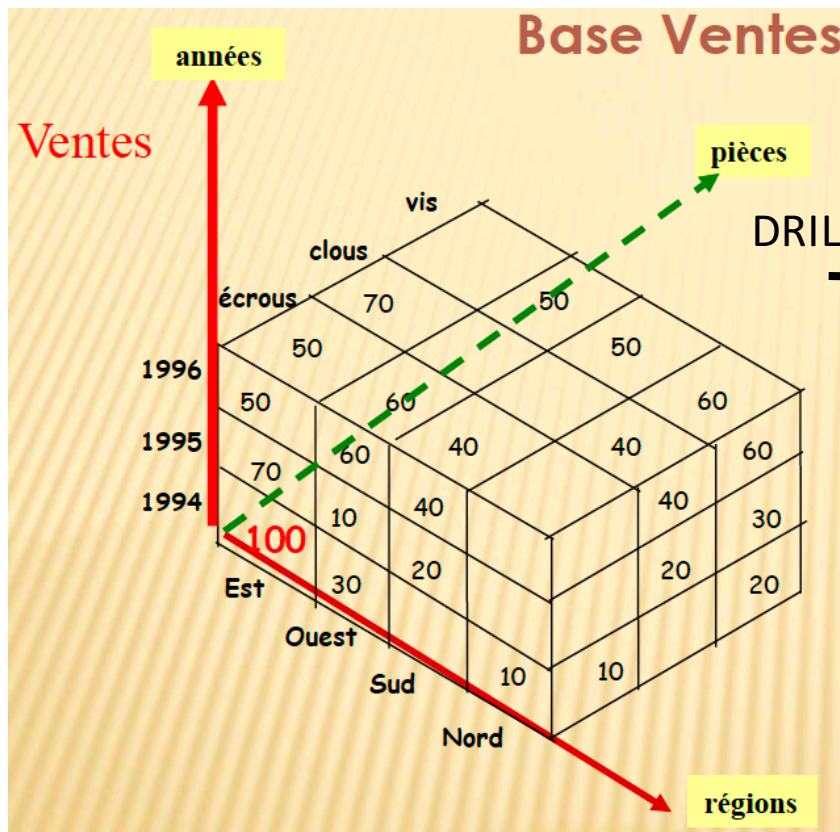
III. Regrouper et interroger les données

① les Entrepôt de Données (noté ED ou DW)

- **OLAP On Line Analytical Processing** (vs Transactional Processing in BD)

- Opérations principales :

- agrégation de données (ROLL-UP),
- forage de données (DRILL-DOWN)



	vis	bordeaux	dijon	grenoble	lille	lyon	marseille	montpellier	nantes	paris	poitiers
vis	20			10		20	30	20	50	10	
clous		10	30	30				40			10
écrous	30	20		30	40	20		10			10
1996	30	20		30	40	20		10	10		10
1995	10	10		60	20					20	
1994	10	20	10	10	70	10		10			

figure de Omar Boussaïd, Univ de Lyon

III. Regrouper et interroger les données

① les Entrepôt de Données (noté ED ou *DW*)

- **OLAP On Line Analytical Processing** (vs Transactional Processing in BD)

- Opérations principales :

- agrégation de données (ROLL-UP),
- forage de données (DRILL-DOWN)
- et sélection, projection, rotation, *switch*, *pushing*, *nesting*

Ventes		1996	1995	1997
é c r o u s	Est	50	70	100
	Ouest	60	10	30
	Nord			10
	Sud	40	20	
V i s	Est		10	10
	Ouest	50	50	50
	Nord	60	30	20
	Sud	50	60	60
C l o u s	Est	70	50	40
	Ouest		10	40
	Nord	40	20	
	Sud		10	

figure de Omar Boussaïd, Univ de Lyon

III. Regrouper et interroger les données

② les systèmes médiateurs

- **Similaire** à l'approche entrepôt,
 - mais l'entrepôt n'est pas matérialisé
 - le schéma global est matérialisé
- **Réécriture de requêtes** suivant les mappings – exemple :

Opodo_Flight(AF84, 14/04/2011, 10h40, 1, 412) Orbitz_Flight(AF10, "14/04/2011 – 19h40", "regular")

Opodo_Flight(AF83, 13/05/2011, 15h35, 1, 412) Orbitz_Flight(AF84, "14/04/2011 – 10h40", "regular
+vegetarian")

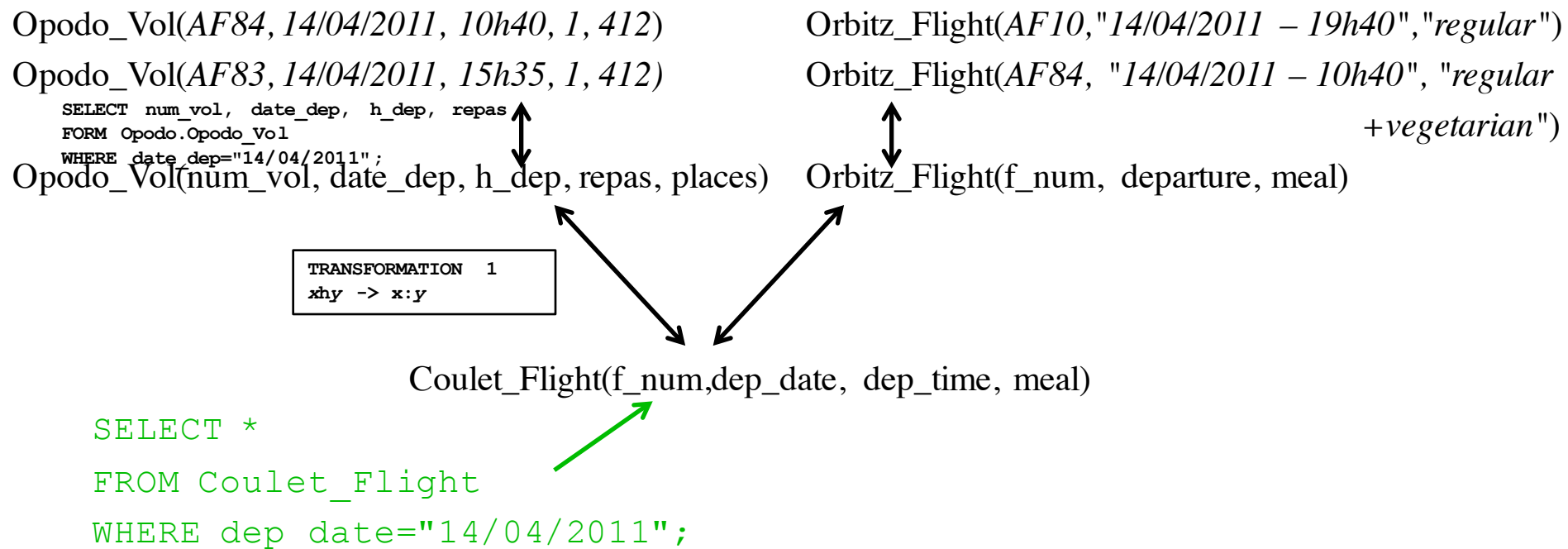
Opodo_Vol(num_vol, date_dep, h_dep, repas, places) Orbitz_Flight(f_num, departure, meal)

Coulet_Flight(f_num, dep_date, dep_time, meal)

III. Regrouper et interroger les données

② les systèmes médiateurs

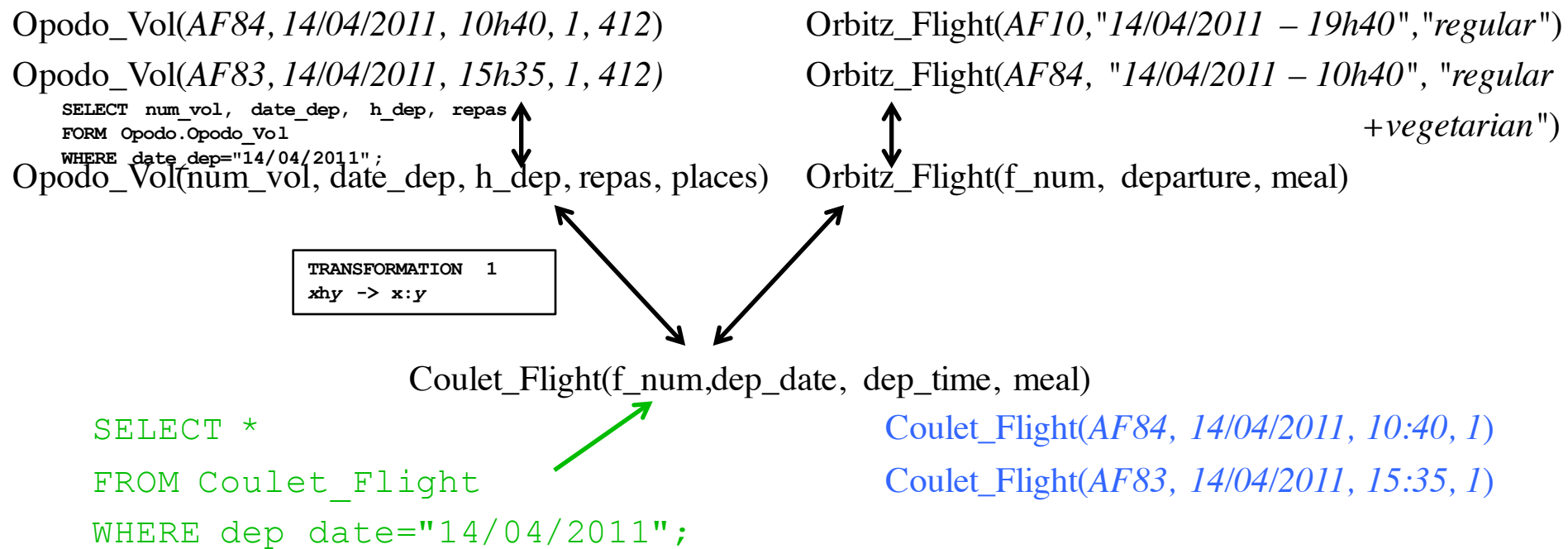
- **Similaire** à l'approche entrepôt
 - mais l'entrepôt n'est pas matérialisé (phase ETL *"on the fly"*)
 - le schéma global est matérialisé
- Réécriture de requête et transformation selon les mappings – ex:



III. Regrouper et interroger les données

② les systèmes médiateurs

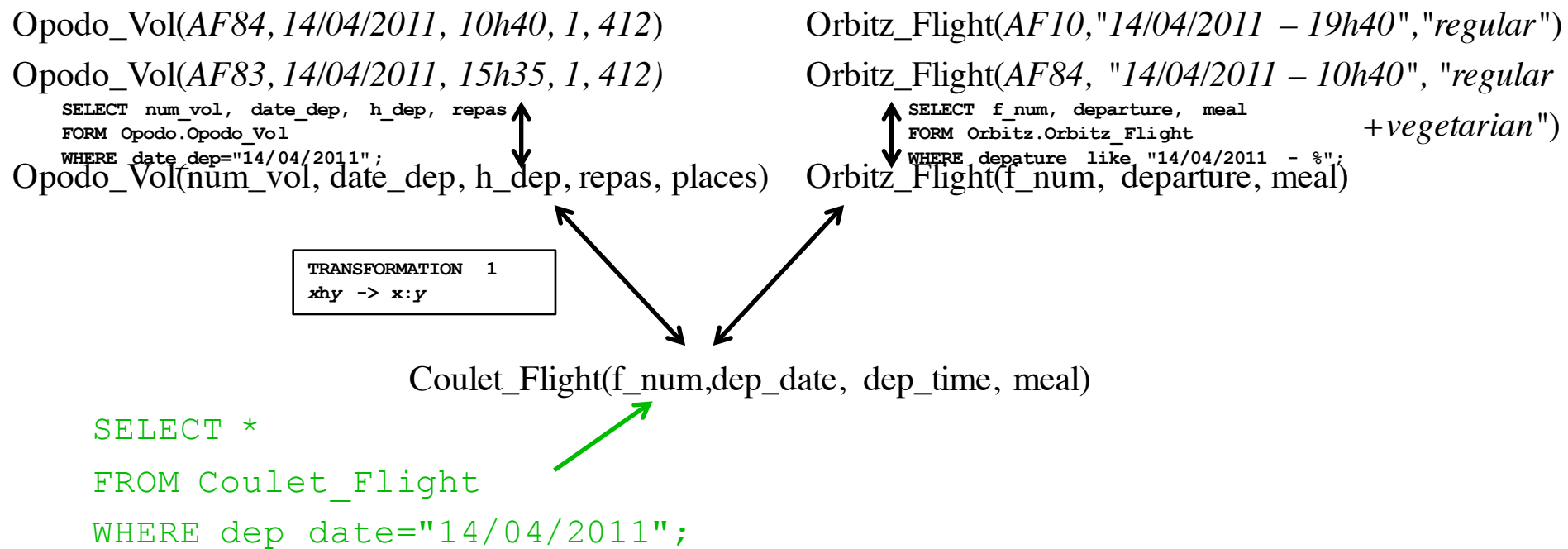
- **Similaire** à l'approche entrepôt
 - mais l'entrepôt n'est pas matérialisé (phase ETL *"on the fly"*)
 - le schéma global est matérialisé
- Réécriture de requête et transformation selon les mappings – ex:



III. Regrouper et interroger les données

② les systèmes médiateurs

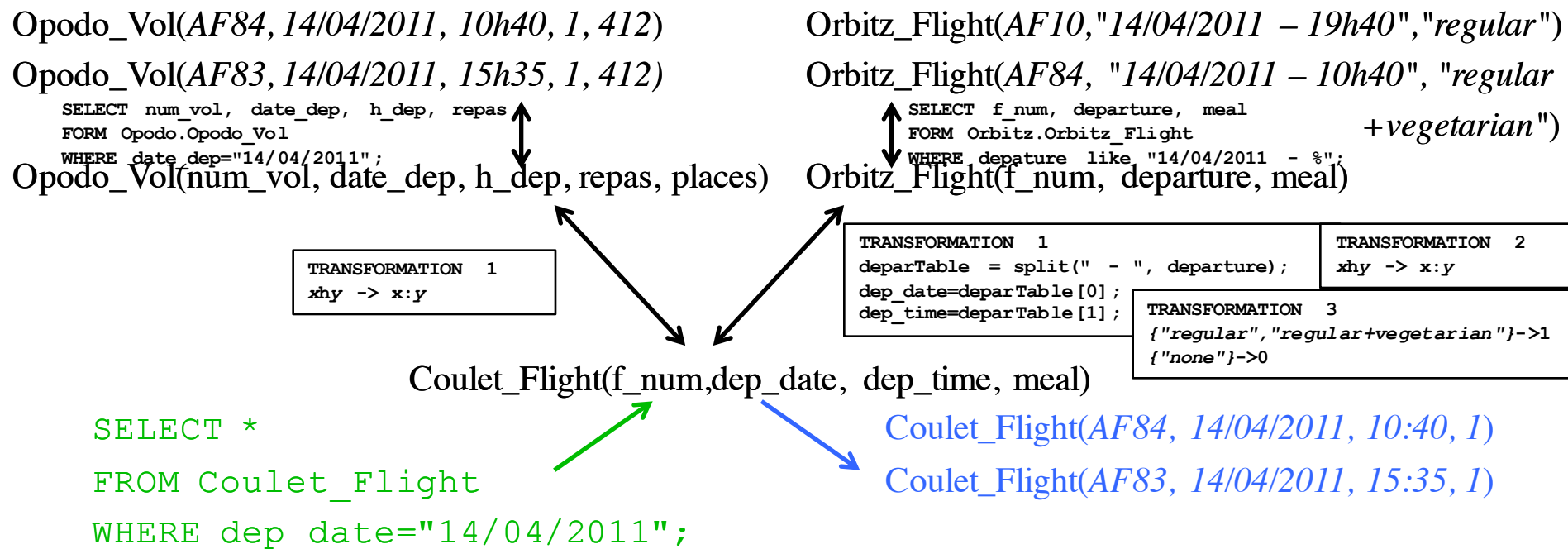
- **Similaire** à l'approche entrepôt
 - mais l'entrepôt n'est pas matérialisé (phase ETL *"on the fly"*)
 - le schéma global est matérialisé
- Réécriture de requête et transformation selon les mappings – ex:



III. Regrouper et interroger les données

② les systèmes médiateurs

- **Similaire** à l'approche entrepôt
 - mais l'entrepôt n'est pas matérialisé (phase ETL *"on the fly"*)
 - le schéma global est matérialisé
- Réécriture de requête et transformation selon les mappings – ex:



III. Regrouper et interroger les données

② les systèmes médiateurs

- **Similaire** à l'approche entrepôt

- mais l'entrepôt n'est pas matérialisé (phase ETL "on the fly")
- le schéma global est matérialisé

- Réécriture de requête et transformation selon les mappings – ex:

Opodo_Vol(AF84, 14/04/2011, 10h40, 1, 412)

Opodo_Vol(AF83, 14/04/2011, 15h35, 1, 412)

```
SELECT num_vol, date_dep, h_dep, repas
FROM Opodo.Opodo_Vol
WHERE date_dep="14/04/2011";
```

Opodo_Vol(num_vol, date_dep, h_dep, repas, places)

Orbitz_Flight(AF10, "14/04/2011 – 19h40", "regular")

Orbitz_Flight(AF84, "14/04/2011 – 10h40", "regular
+vegetarian")

Orbitz_Flight(f_num, departure, meal)

```
SELECT f_num, departure, meal
FROM Orbitz.Orbitz_Flight
WHERE departure like "14/04/2011 - %";
```

TRANSFORMATION 1
xhy -> x:y

TRANSFORMATION 1
deparTable = split(" - ", departure);
dep_date=deparTable[0];
dep_time=deparTable[1];

TRANSFORMATION 2
xhy -> x:y

TRANSFORMATION 3
{"regular", "regular+vegetarian"}->1
{"none"}->0

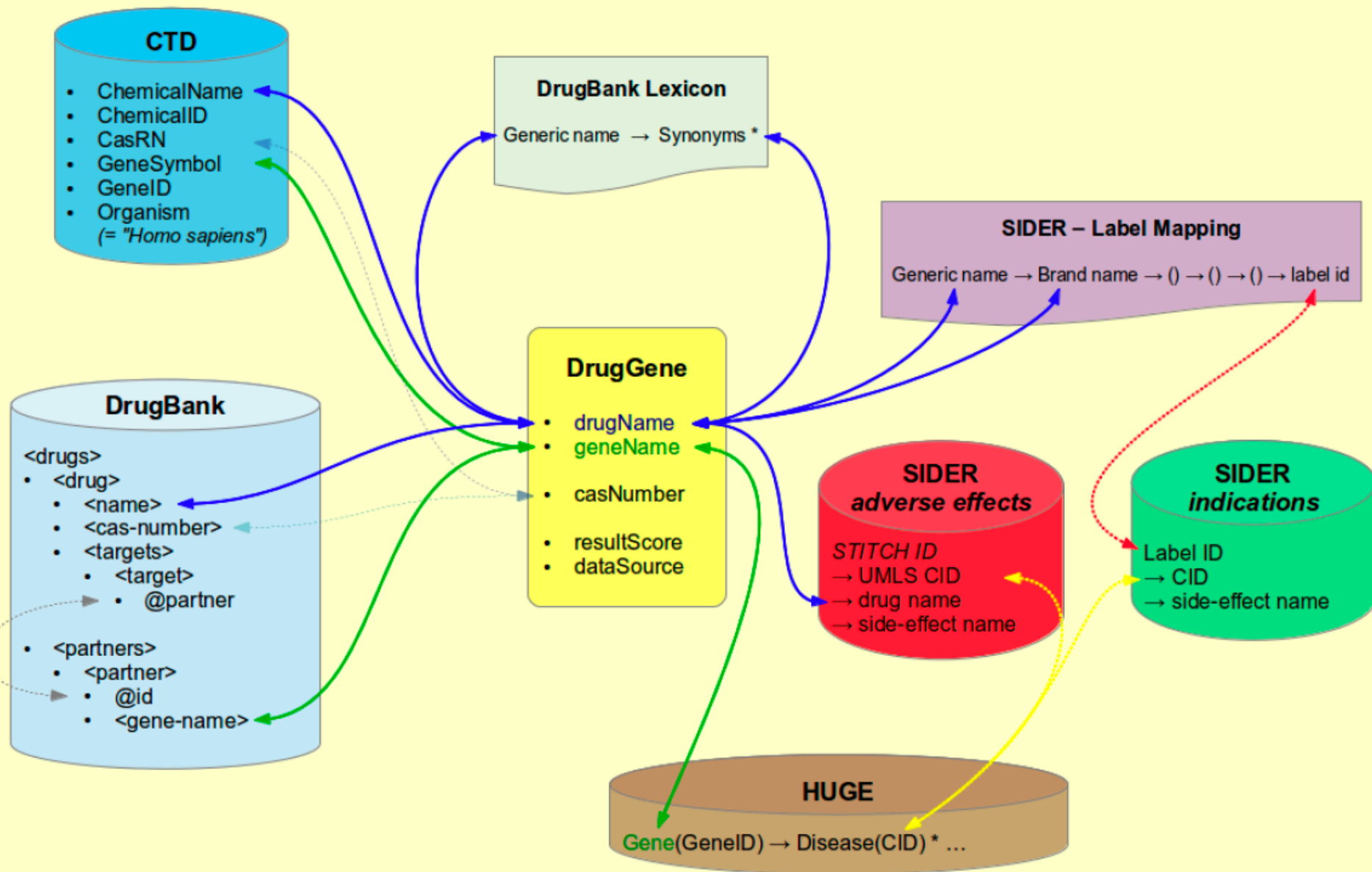
Coulet_Flight(f_num, dep_date, dep_time, meal)

```
SELECT *
FROM Coulet_Flight
WHERE dep_date="14/04/2011";
```

Coulet_Flight(AF84, 14/04/2011, 10:40, 1)

Coulet_Flight(AF83, 14/04/2011, 15:35, 1)

Coulet_Flight(AF10, 14/04/2011, 19:40, 1)



"si on avait trop de ressources, on ne pourrait pas les utiliser efficacement"

FAUX

"on a pas assez de ressources..."
"c'est trop cher à acheter, entretenir, administrer..."

FAUX

"si on avait trop de ressources, on ne pourrait pas les utiliser efficacement"

FAUX

"on a pas assez de ressources..."

"c'est trop cher a acheter, entretenir, administrer..."

FAUX

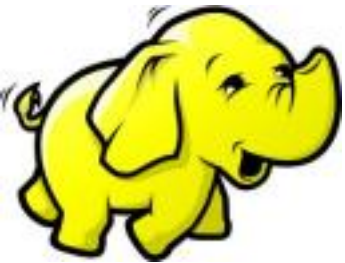
Le cloud computing c'est facile, pas cher et ça *SCALE* -- ie ça passe à l'échelle

VRAI

III. Regrouper et interroger les données

③ *le cloud computing*

"bla, on ne pourrait pas les utiliser"



C'est faux : Hadoop (un autre projet Apache)

- MODELE DE DISTRIBUTION = MapReduce
- UNE ARCHI = HDFS : Hadoop Distributed File System
- d'autres

La technologie du 21^{ème} siècle

III. Regrouper et interroger les données

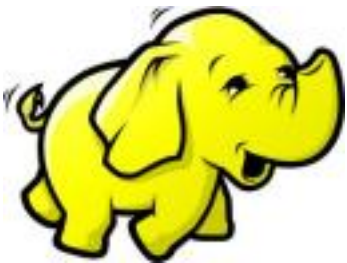
③ *le cloud computing*

- Pourquoi MapReduce/cloud computing ?

Parce que GMD

Parce que

- Google : nouvel index
- Facebook : statistiques internes
- Yahoo! : Yahoo! Search
- Last.fm : statistique hebdo
- etc.



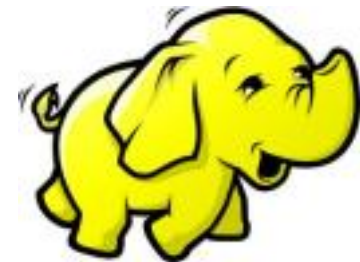
donc Hadoop (processus *MapReduce* et archi *HDFS*)

III. Regrouper et interroger les données

③ le *cloud computing*

a) MapReduce

- **MapReduce est un paradigme de programmation**
- dédié au traitement de gros volume de données
- qui se décompose en 2 étapes
 - Map : étape de traitement des données qui renvoie un résultat sous la forme de paire *{clé, valeur}*
 - Reduce : étape de fusion des résultats unitaires par clé pour former le résultat final
- Algo :
 - Les données en entrée sont découpées en petites unités de données
 - Les unités sont alors traitées en parallèle par la fonction Map
 - Le résultat des traitement unitaires par la fonction Map est trié par clé pour former des unités de résultats passées à la fonction Reduce
 - La fonction Reduce regroupe les différents résultats unitaires pour constituer un résultat final et unique



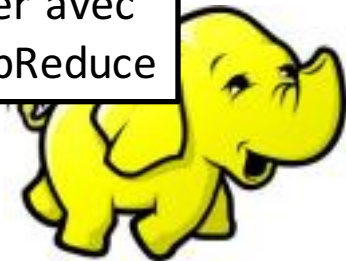
III. Regrouper et interroger les données

③ le *cloud computing*

a) MapReduce

- **MapReduce est un paradigme de programmation**
- dédié au traitement de gros volume de données
- MapReduce se décompose en 2 étapes
 - Map : étape de traitement des données qui renvoie un résultat sous la forme de paire *{clé, valeur}*
 - Reduce : étape de fusion des résultats unitaires par clé pour former le résultat final
- Algo :
 - Les données en entrée sont découpées en petites unités de données
 - Les unités sont alors traitées en parallèle par la fonction Map
 - Le résultat des traitements unitaires par la fonction Map est trié par clé pour former des unités de résultats passées à la fonction Reduce
 - La fonction Reduce regroupe les différents résultats unitaires pour constituer un résultat final et unique

On peut boucler sur cet algo, ou l'enchaîner avec d'autres MapReduce



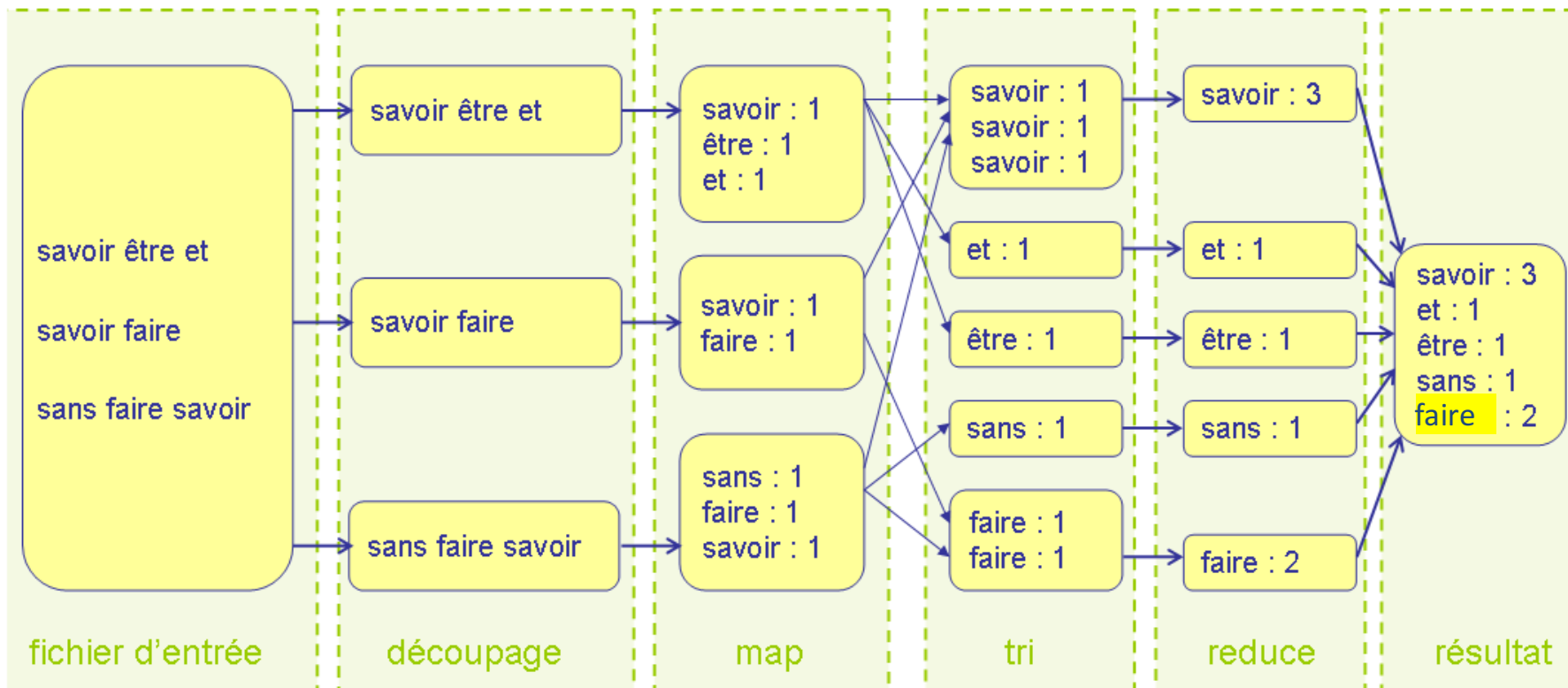
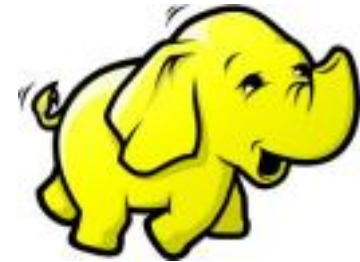
III. Regrouper et interroger les données

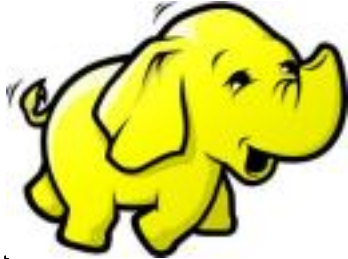
③ le *cloud computing*

- Exemple : compter le nombre de mots dans différents documents, e.g., trois phrases

```
1.  savoir être et
2.  savoir faire
3.  sans faire savoir
```

a) MapReduce





III. Regrouper et interroger les données

③ le *cloud computing*

a) MapReduce

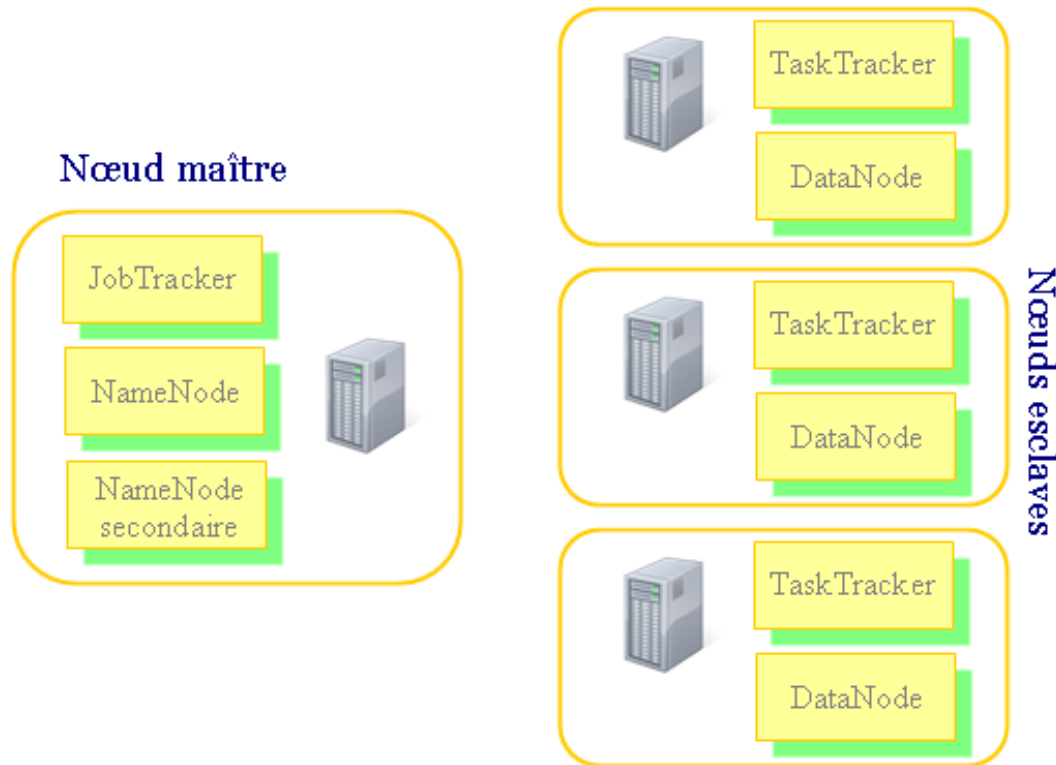
- Pour lancer un process MapReduce avec Hadoop
 - Configuration du job (définie par l'utilisateur)
 - Découpage du jeu de données et distribution sur le cluster (**géré par Hadoop**)
 - Démarrage de chaque tâche `Map` avec son propre jeu de données issu du découpage (**géré par Hadoop**)
 - Exécution en parallèle de chaque fonction `Map` (implémenté par l'utilisateur)
 - Les sorties des fonctions `Map` sont triées par clé pour former de nouvelles unités de données (**géré par Hadoop**)
 - Démarrage des tâches `Reduce` avec son propre jeu de données issu du tri (**géré par Hadoop**)
 - Exécution en parallèle de chaque fonction `Reduce` (implémenté par l'utilisateur)
 - Assemblage du résultat des opérations `Reduce` et stockage (**géré par Hadoop**)

III. Regrouper et interroger les données

③ le *cloud computing*

a) HDFS

- Typologie d'un cluster Hadoop : maître - esclaves
 - HDFS : NameNode – DataNode
 - Le contrôle de Job : JobTracker - TaskTracker

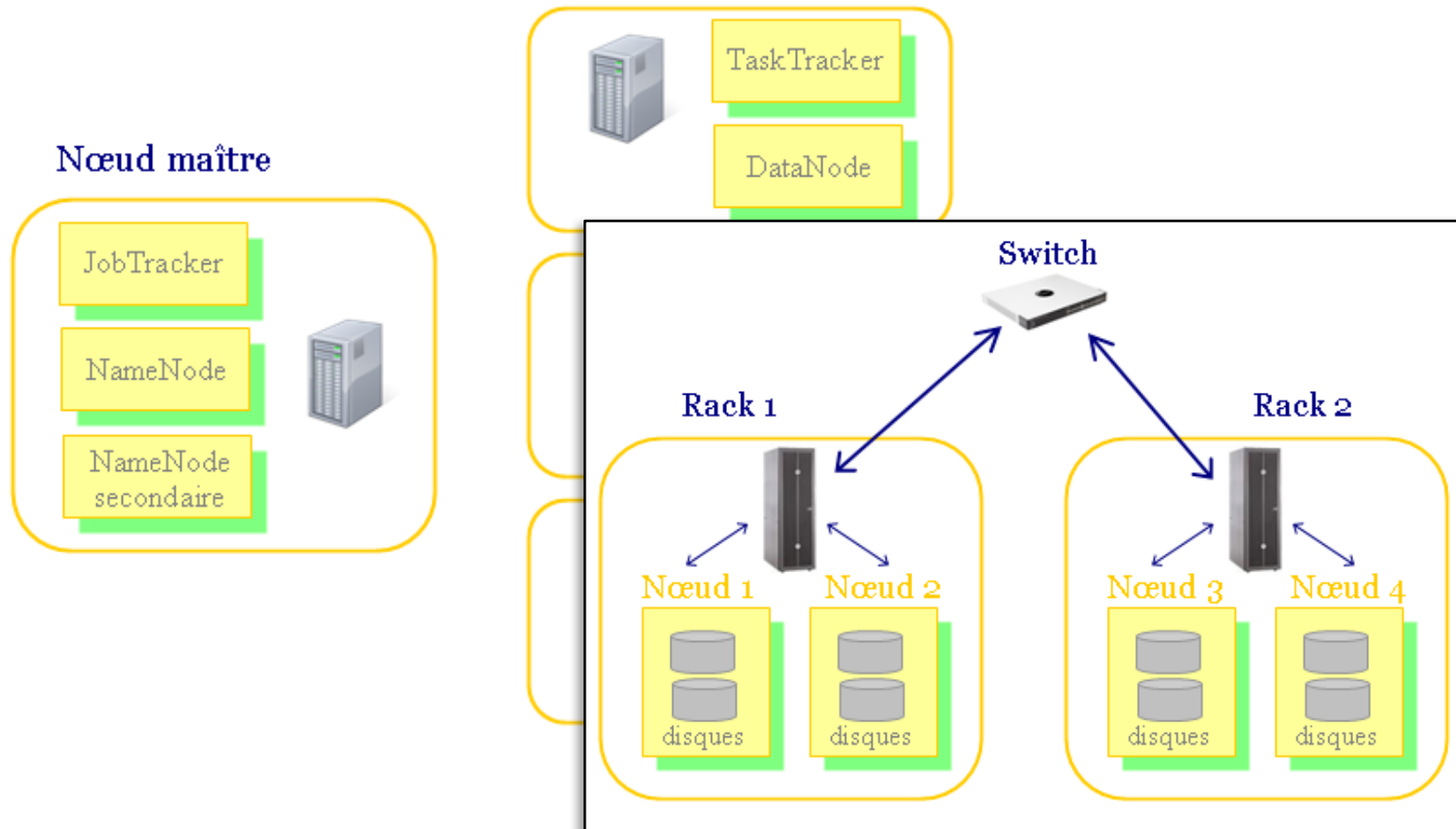


III. Regrouper et interroger les données

③ le *cloud computing*

a) HDFS

- Typologie d'un cluster Hadoop : maître - esclaves
 - HDFS : NameNode – DataNode
 - Le contrôle de Job : JobTracker - TaskTracker



III. Regrouper et interroger les données

③ le *cloud computing*

"bla trop cher bla bla"

- C'est faux : Amazon Elastic Compute Cloud (EC2)



[Sign in to the AWS Management Console](#) | [Create an AWS Account](#) | [English](#)

Search: AWS Product Information

AWS

Products

Developers

Community

Support

Account

Products & Services

Amazon Elastic Compute Cloud (Amazon EC2)

Amazon EC2 Details

- **EC2 Overview**
- [EC2 FAQs](#)
- [EC2 Pricing](#)
- [Amazon EC2 SLA](#)
- [EC2 Instance Types](#)
- [EC2 Instance Purchasing Options](#)
- [Reserved Instances](#)
- [Spot Instances](#)
- [Windows Instances](#)

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale computing easier for developers.

Amazon EC2's simple web service interface allows you to obtain and configure capacity with minimal friction. It provides you with complete control of your computing resources and lets you run on Amazon's proven computing environment. Amazon EC2 reduces the time required to obtain and boot new server instances to minutes, allowing you to quickly scale capacity, both up and down, as your computing requirements change. Amazon EC2 changes the economics of computing by allowing you to pay only for capacity that you actually use. Amazon EC2 provides developers the tools to build failure resilient applications and isolate themselves from common failure scenarios.

[Sign Up For Amazon EC2](#)

This page contains the following categories of information. Click to jump down:

On-Demand Instances

On-Demand Instances let you pay for compute capacity by the hour with no long-term commitments. This frees you from the costs and complexities of planning, purchasing, and maintaining hardware and transforms what are commonly large fixed costs into much smaller variable costs.

The pricing below includes the cost to run private and public AMIs on the specified operating system ("Windows Usage" prices apply to both Windows Server® 2003 and 2008). Amazon also provides you with additional instances for [Amazon EC2 running Microsoft](#), [Amazon EC2 running SUSE Linux Enterprise Server](#) and [Amazon EC2 running IBM](#) that are priced differently.

Region: EU – Ireland		
Standard On-Demand Instances	Linux/UNIX Usage	Windows Usage
Small (Default)	\$0.095 per hour	\$0.12 per hour
Large	\$0.38 per hour	\$0.48 per hour
Extra Large	\$0.76 per hour	\$0.96 per hour
Micro On-Demand Instances		
Micro	\$0.025 per hour	\$0.035 per hour
High-Memory On-Demand Instances		
Extra Large	\$0.57 per hour	\$0.62 per hour
Double Extra Large	\$1.14 per hour	\$1.24 per hour
Quadruple Extra Large	\$2.28 per hour	\$2.48 per hour
High-CPU On-Demand Instances		
Medium	\$0.19 per hour	\$0.29 per hour
Extra Large	\$0.76 per hour	\$1.16 per hour
Cluster Compute Instances		
Quadruple Extra Large	N/A*	N/A*
Cluster GPU Instances		
Quadruple Extra Large	N/A*	N/A*
* Cluster Compute and Cluster GPU Instances are currently only available in the US – N. Virginia Region.		

Amazon Elastic Block Store

Region:

Amazon EBS Volumes

- \$0.11 per GB-month of provisioned storage
- \$0.11 per 1 million I/O requests

Amazon EBS Snapshots to Amazon S3

- \$0.15 per GB-month of data stored
- \$0.01 per 1,000 PUT requests (when saving a snapshot)
- \$0.01 per 10,000 GET requests (when loading a snapshot)

et il n'y a pas que Amazon : Microsoft Azure, Google App Engine, etc.