Data Warehouse and Big Data

Master IISC-SID M2
Département Science Informatique
CY Cergy-Paris Université
Tao-Yuan JEN, tao-yuan.jen@cyu.fr

Traditional RDBMS

Relational Table, Constraint, View,
Trigger, Index , etc

ACID Property
- Atomicity
- Consistency
- Isolation
- Durability

OLTP

On-Line Transactional Processing (OLTP)

Operations:

- Simple;
- Interactive;
- Heavily Concurrent;
- Repetitive;
- Structured

Data:

- Limited tuples;
- Detailed;
- Updated

Context of Data Warehouse Emergence

Data Explosion

Progress of Material

From Data to Information

New Needs

Exploration and analysis of historical data

Huge volumes of data (To, Po, ...)

Online analytical processing
(On-Line Analytical Processing OLAP)

Examples of Different Levels of Data (1)

Tera-byte applications :

1 To hard disk, Hitachi 2007

Internet Trafic :
1997 : 100 To /an
2008 : 160 To/sec

Wikipedia : 5,87 To documents (2010)

IBM ordinateur Watson : 16 To RAM

Examples of Different Levels of Data (2)

Peta-byte applications :

Google : 24 Po /day (2009)

Supercomputer Blue Water (2012) :
1,5 Po RAM, 25 Po Hard Disk

Facebook : +100 Po / HDFS (2012)

Microsoft: 150 Po Hotmail to Outlook
6 weeks (2013)

Examples of Different Levels of Data (3)

Exa-byte applications :

64bit computer's RAM limitation

Zetta-byte applications :

2013 WWW's content : 4 Zo

Yotta-byte applications :

BRAIN Project 2013

OLAP

On-Line Analytical Processing (OLAP)

Operations:
- Complex;
- Interactive;
- Lightly concurrent;
- Unpredictable

Data:
- Very large number of tuples;
- Consolidated/synthetic;
- Historical

New Technologies

Data Warehouse
- Collection of Data

OLAP
- Data to Information

Data Mining
- Information to Knowledge

Why Data Warehouse

Heterogeneous Data Sources

Performance Degradation

Lock-based control for concurrent usage

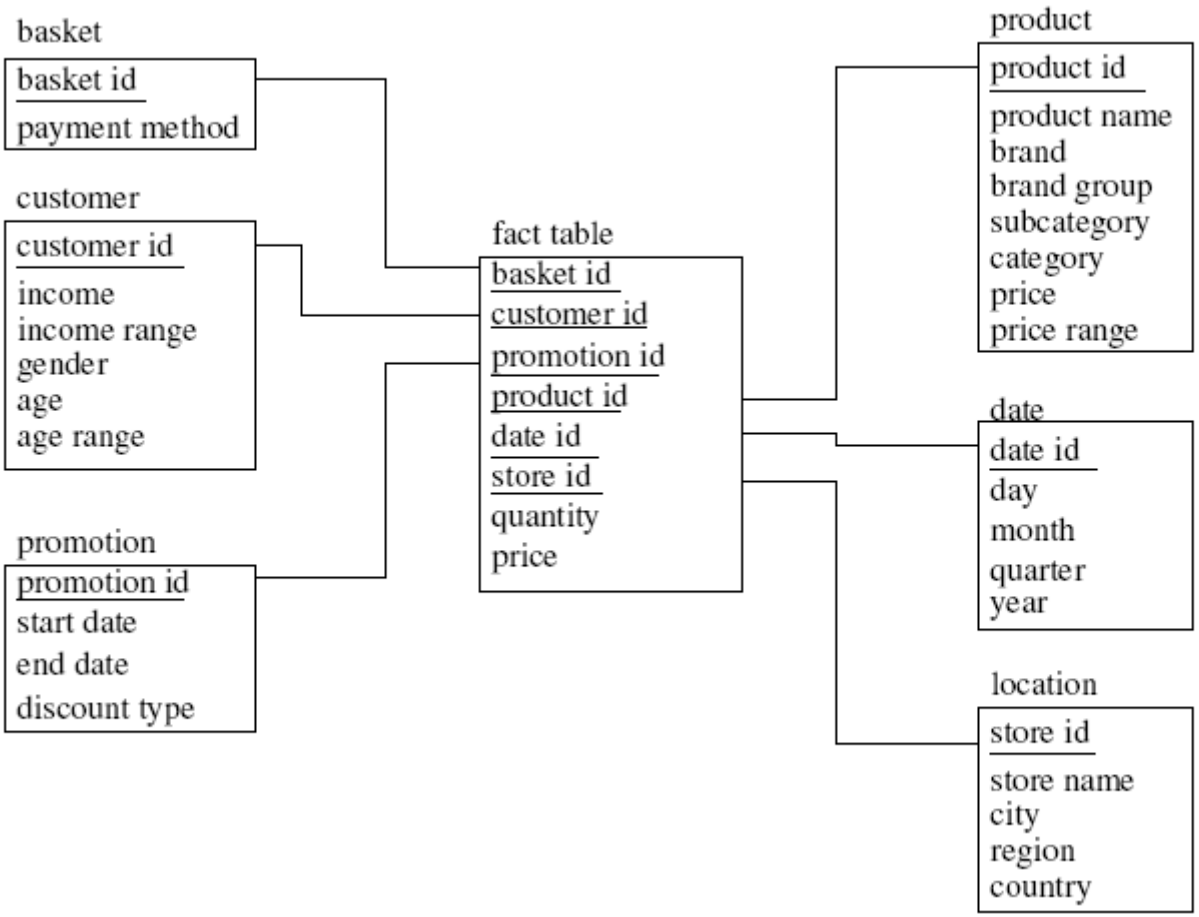
Modelization in Data Warehouse

Star schema

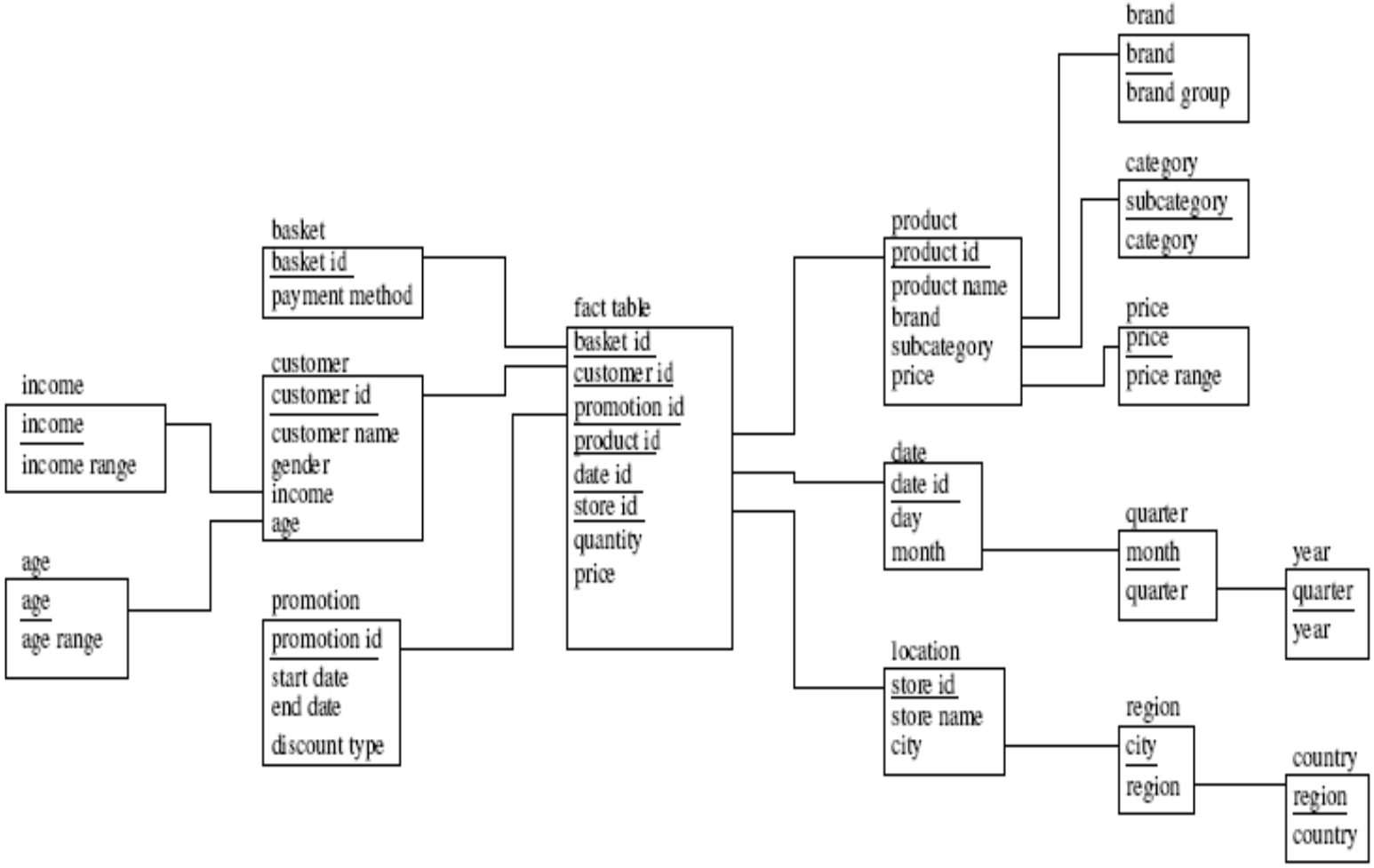
Snowflake schema

Fact constellation

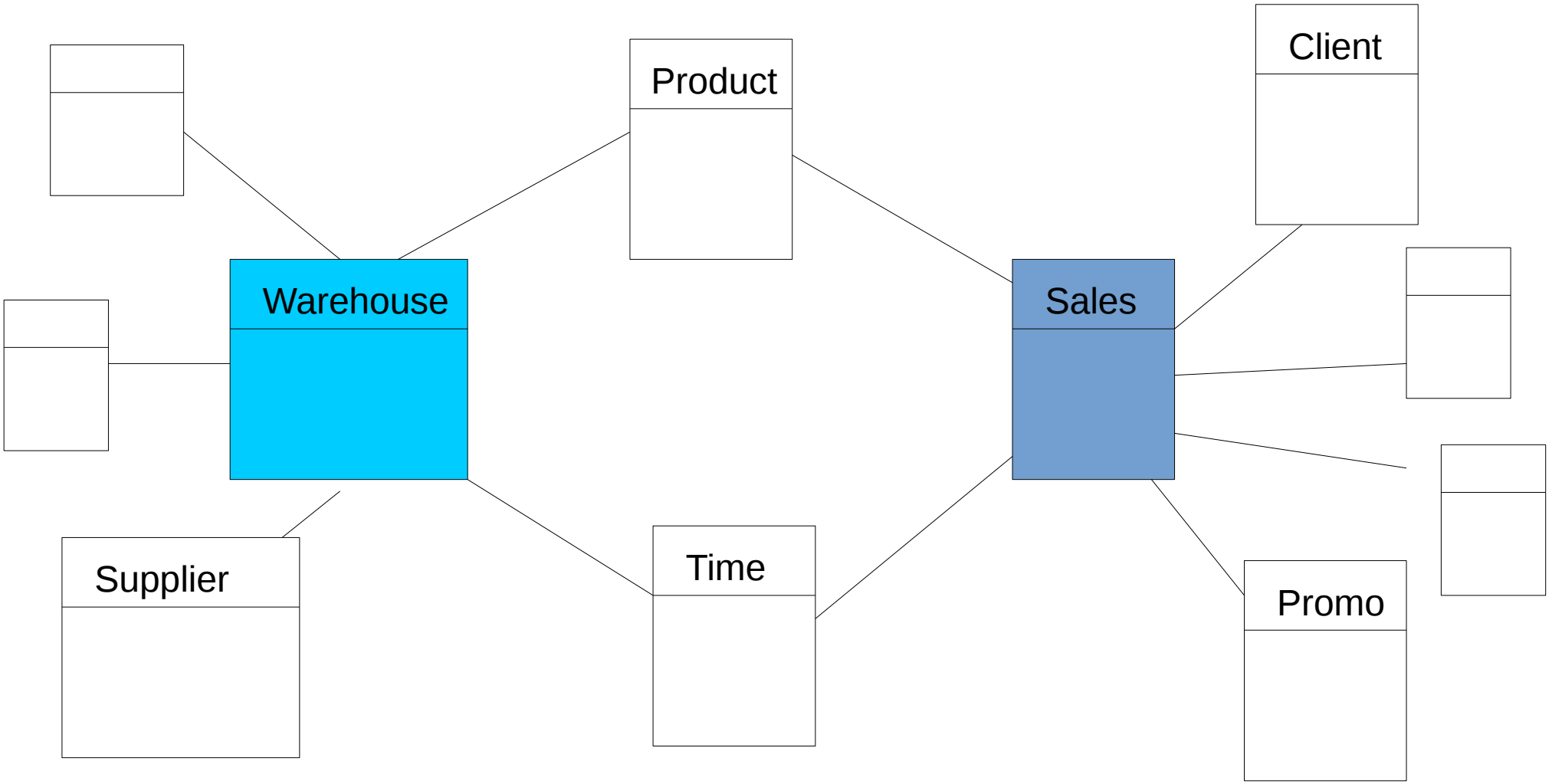
Star schema in Data Warehouse



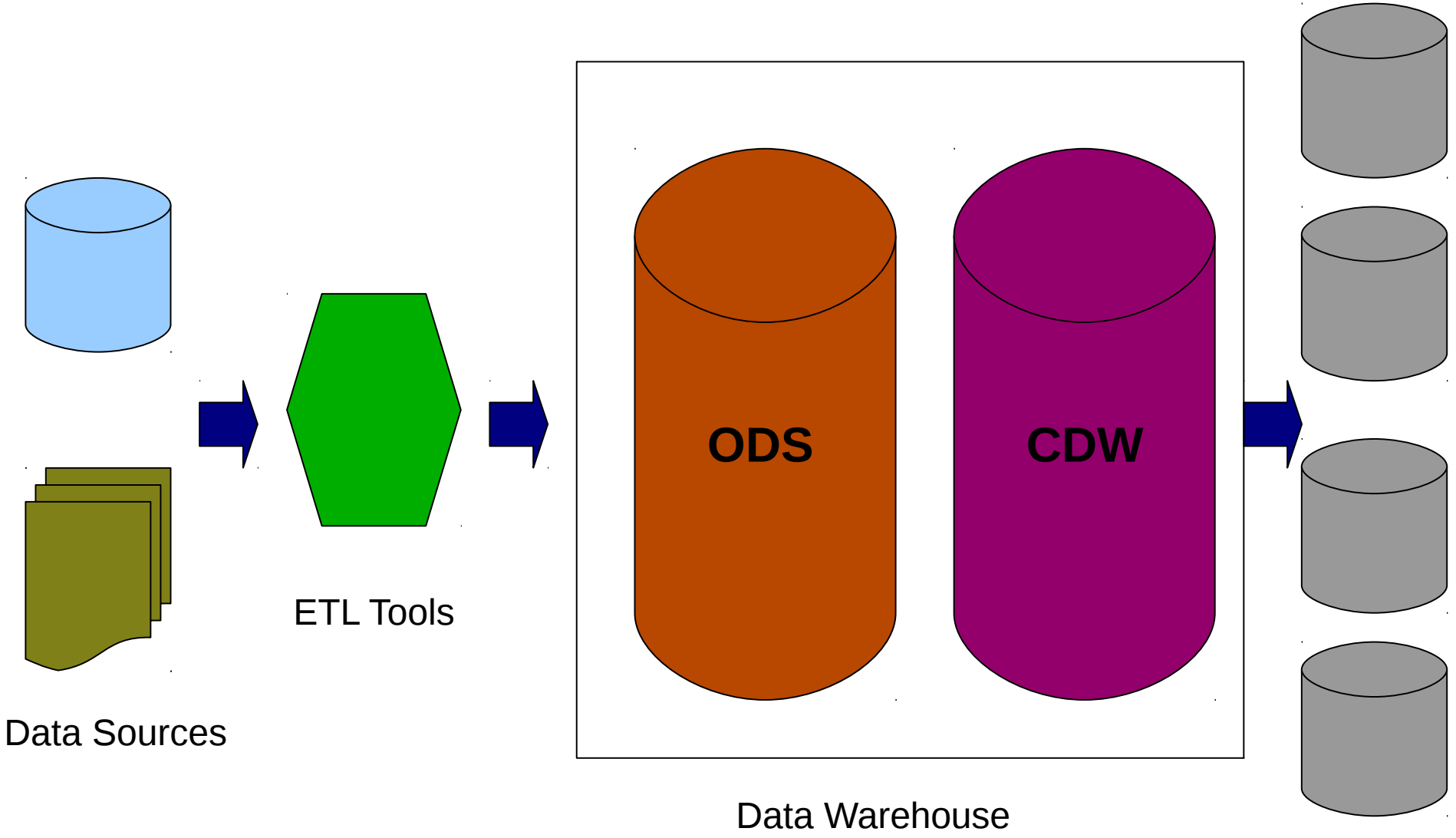
Snowflake schema in Data Warehouse



Fact constellation in Data Warehouse



General Structure of DW



Storage Architecture in Data Warehouse

ODS (Operational Data Store) :
Integrated data

CDW (Cooperated Data Warehouse):
Aggregation materialized views

Steps for Building DW

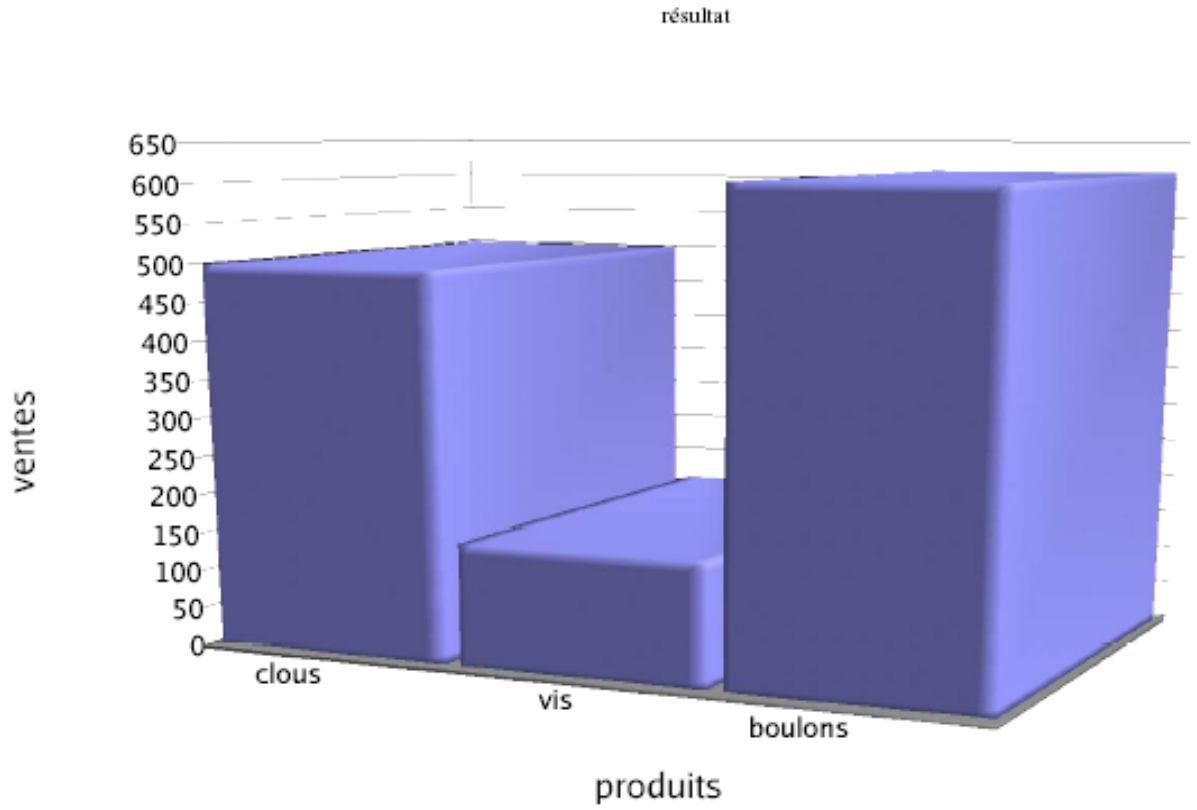
Data Preparation

Data Integration

Data Aggregation

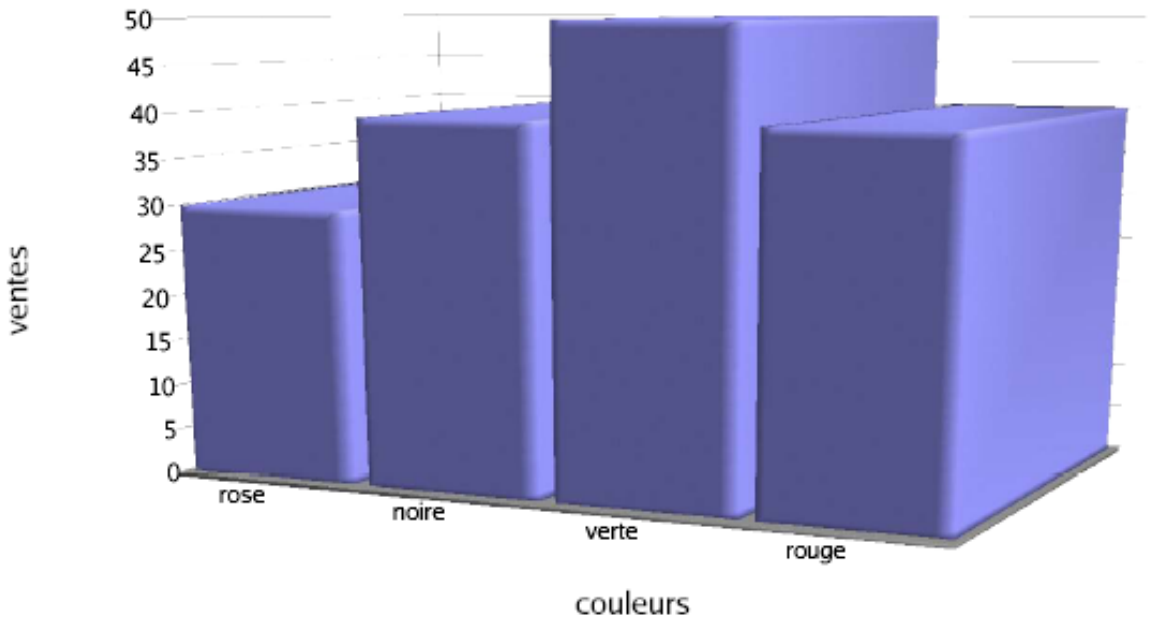
Data Mart Personalization

Scenario of an OLAP Example (1)

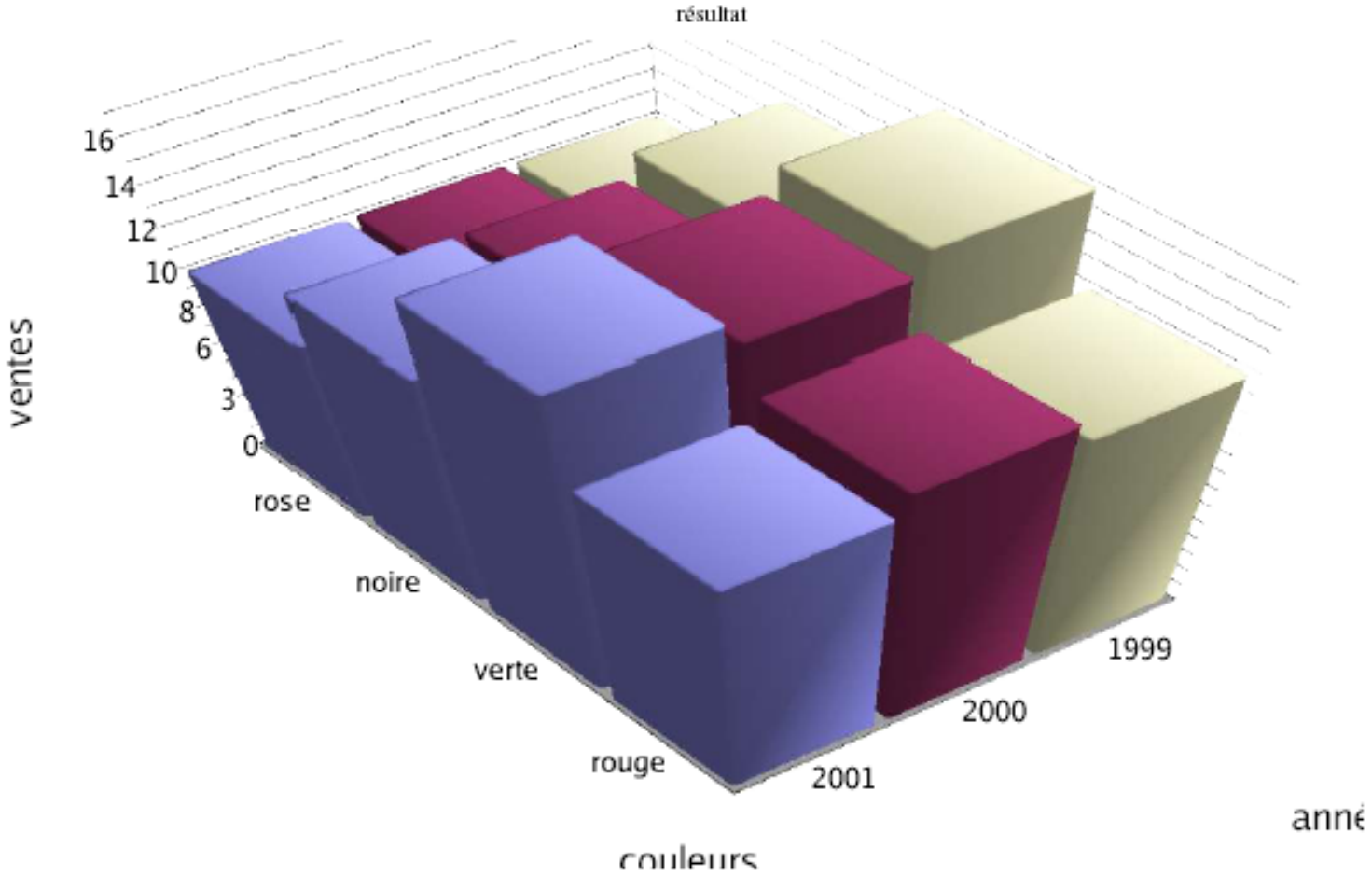


Comparison among sales of nails, screws and blots

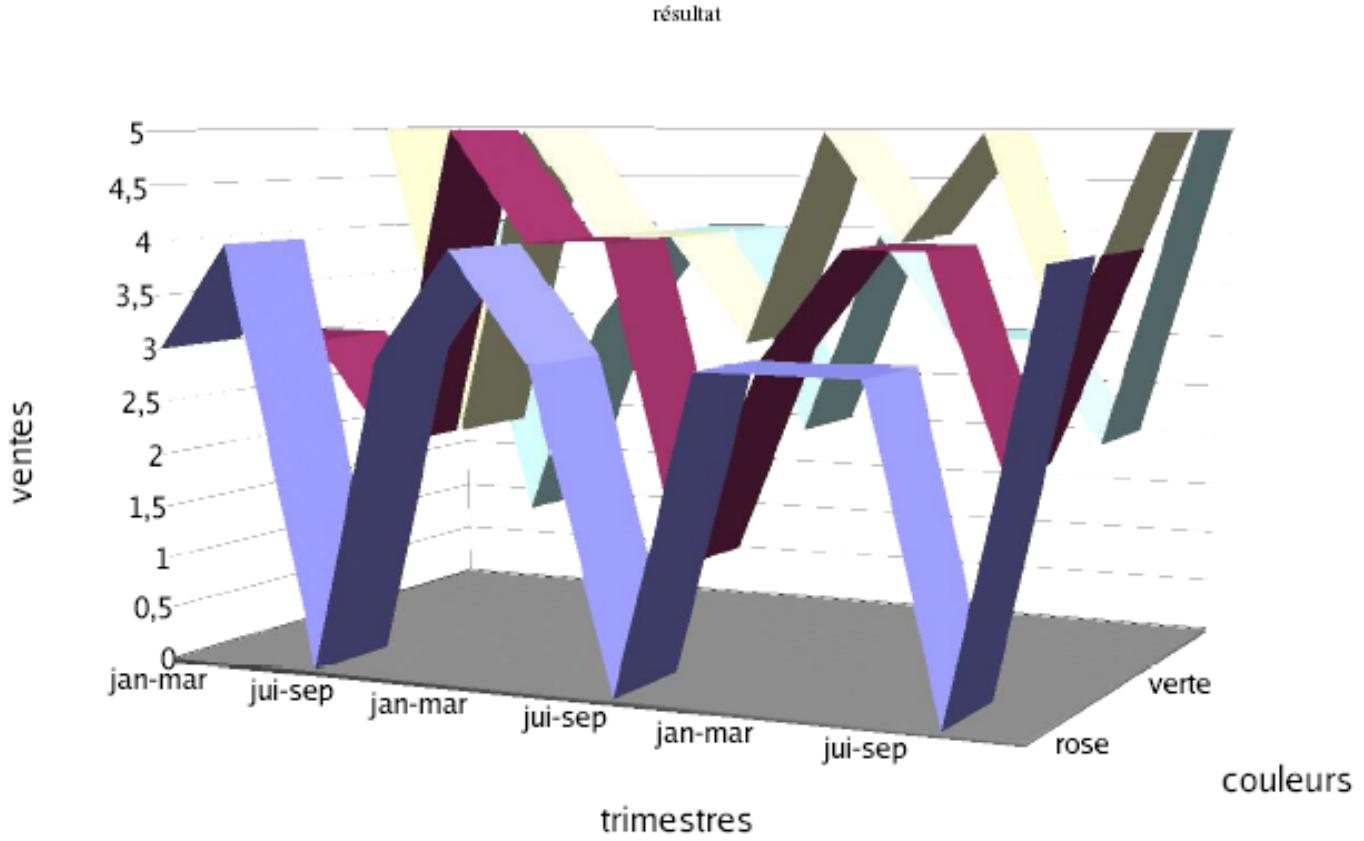
Scenario of an OLAP Example (2)



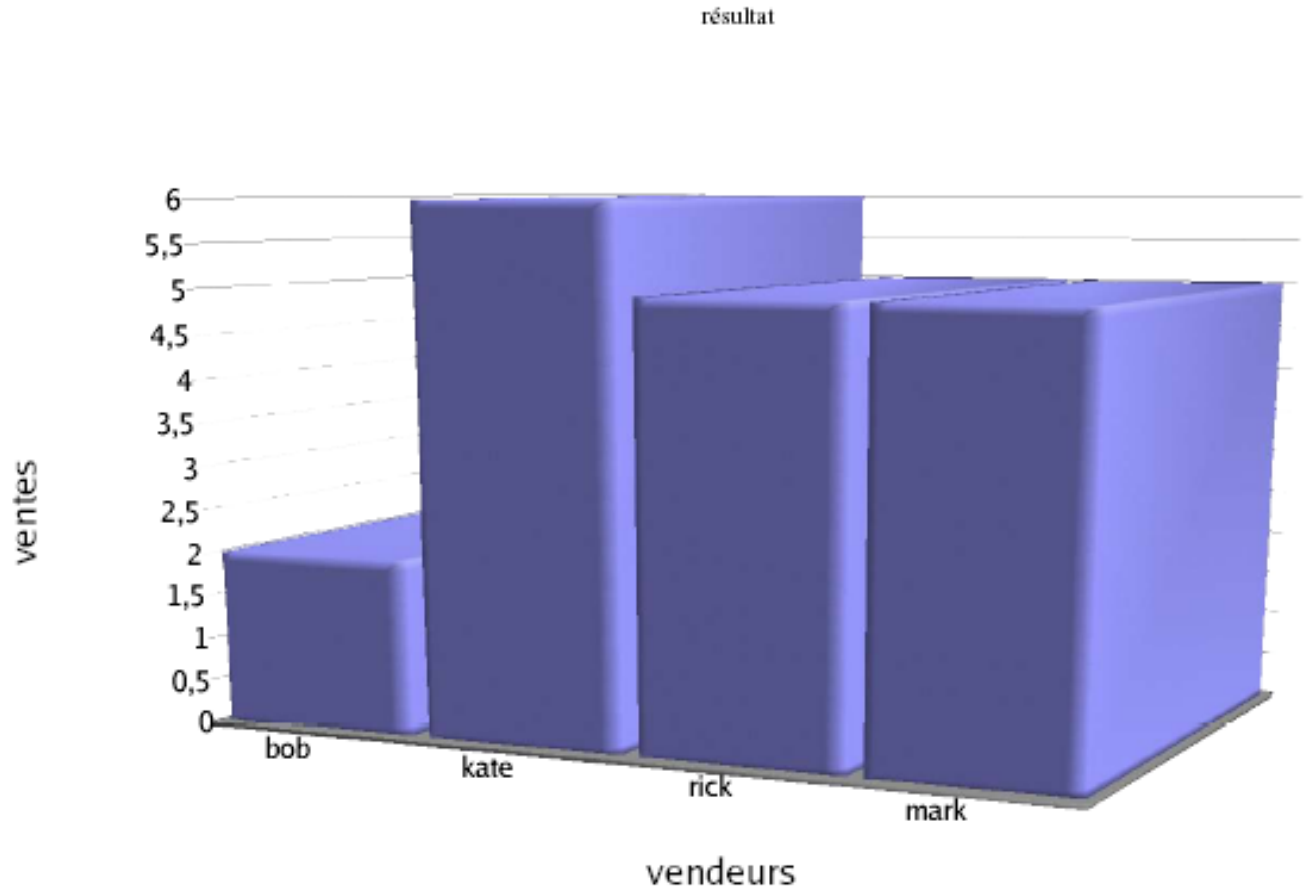
Scenario of an OLAP Example (3)



Scenario of an OLAP Example (4)



Scenario of an OLAP Example (5)



Structured Query Language (SQL)

Select [Distinct] <List of attributes>

From <List of tables>

Where <List of conditions>

Group by <List of attributes...>

Having <List of conditions for groups>

Order by <List of attributes> [asc|desc]

Evaluation order of queries

From: decide working table(s)

Where : filter the data with conditions

Group By: grouping data with attribute values
Having: filter groups

Select: decide attributes to be presented
Distinct: eliminate redundancies

Order By: arrangement

Query Evaluation Example (1)

Marins(M_id: int, M_nom : string, grade : int, age : float)

M_id	M_nom	Grade	Age
22		7	45,0
31		8	55,5
32		8	25,5
85		3	25,5
95		3	63,5
58		10	35,0
29		1	33,0
64		7	35,0
71		10	16

```
Select      M.grade, min(M.age) As minage
From        Marins M
Where       M.age >=18
Group by    M.grade
Having      count(M_id) > 1
Order by    M.grade;
```

Query Evaluation Example (2)

Grade	Age
7	45,0
8	55,5
8	25,5
3	25,5
3	63,5
10	35,0
1	33,0
7	35,0

```
Select      M.grade, min(M.age) As minage
From        Marins M
Where       M.age >=18
Group by    M.grade
Having      count(M_id) > 1
Order by    M.grade;
```

Query Evaluation Example (3)

Grade	Age
7	45
7	35
8	55,5
8	25,5
3	25,5
3	63,5
10	35
1	33

```
Select      M.grade, min(M.age) As minage
From        Marins M
Where       M.age >=18
Group by    M.grade
Having      count(M_id) > 1
Order by    M.grade;
```

Query Evaluation Example (4)

Grade	Age
7	45
7	35
8	55,5
8	25,5
3	25,5
3	63,5

```
Select      M.grade, min(M.age) As minage
From        Marins M
Where       M.age >=18
Group by    M.grade
Having      count(M_id) > 1
Order by    M.grade;
```

Query Evaluation Example (5)

grade	minage
7	35
8	25,5
3	25,5

```
Select    M.grade, min(M.age) As minage
From      Marins M
Where     M.age >=18
Group by  M.grade
Having    count(M_id) > 1
Order by  M.grade;
```

Query Evaluation Example (6)

grade	minage
3	25,5
7	35,0
8	25,5

```
Select      M.grade, min(M.age) As minage
From        Marins M
Where       M.age >=18
Group by    M.grade
Having      count(M_id) > 1
Order by    M.grade;
```

Relational Schema in the Example

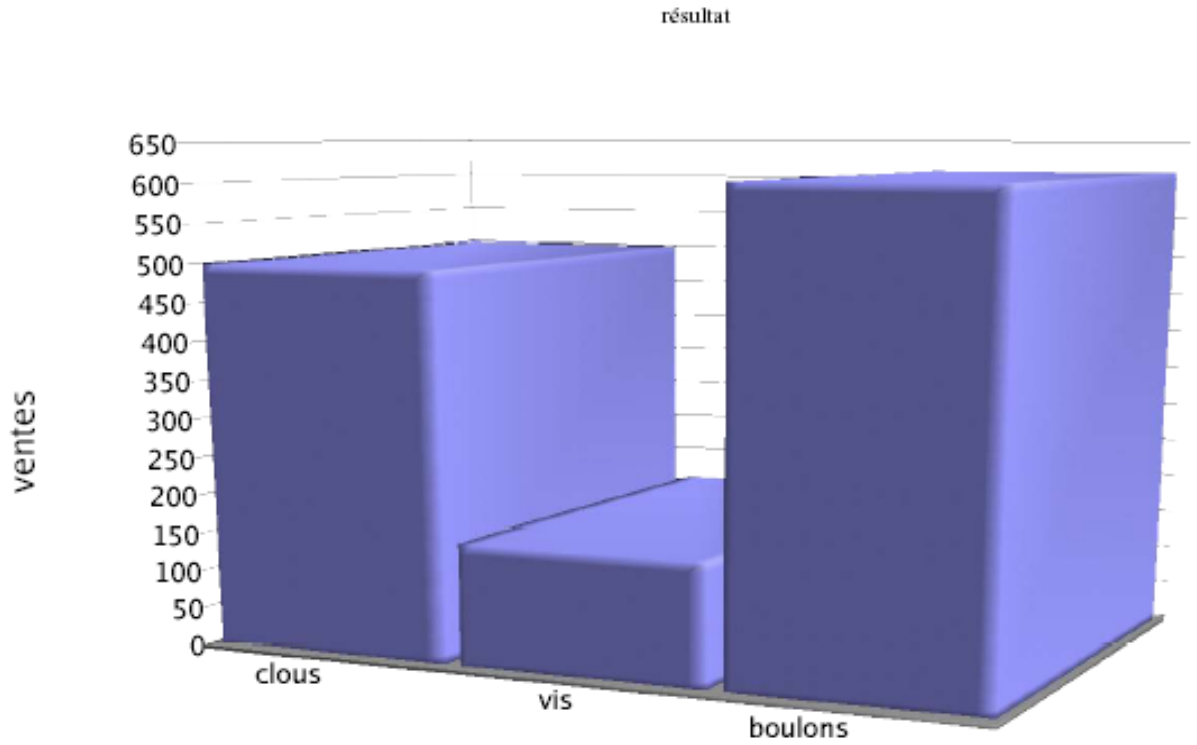
Ventes(codeProduit, date, vendeur, montant)

Produits(codeProduit, modele, couleur)

Vendeurs(nom, ville, departement, etat, pays)

Temps(jour, semaine, mois, trimestre, annee)

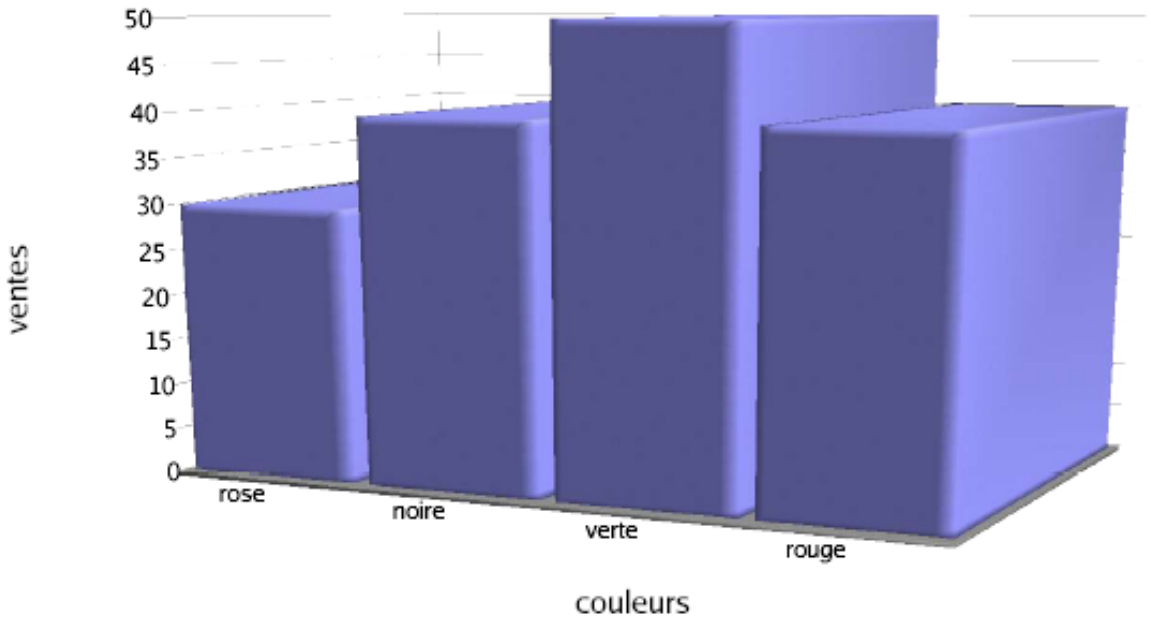
Query for the Scenario (1)



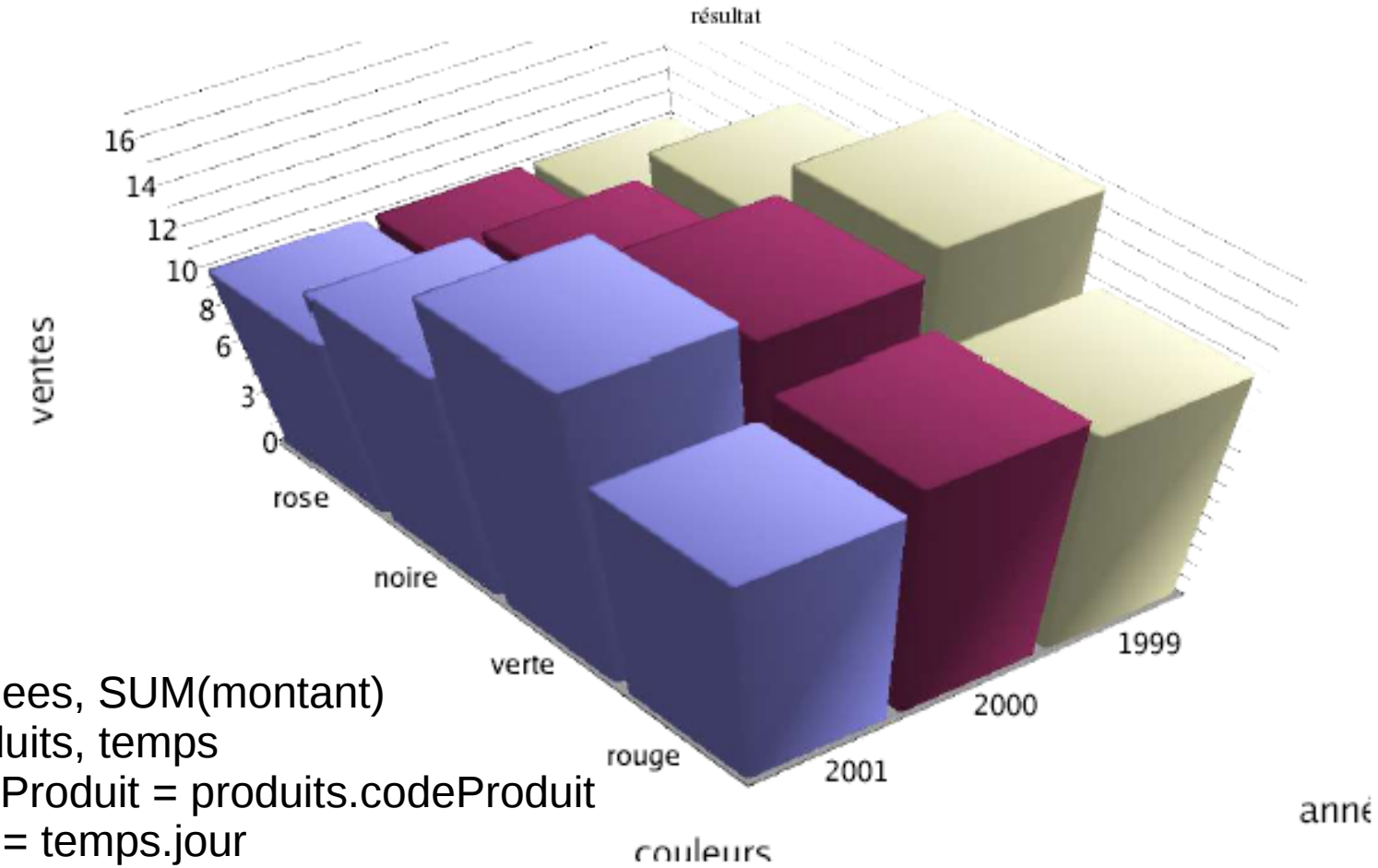

```
SELECT      SUM(montant)
FROM        Ventes, Produits
WHERE       ventes.codeProduit = produits.codeProduit
GROUP BY    Modele;
```

Query for the Scenario (2)

```
SELECT      couleur, SUM(montant)
FROM        ventes, produits
WHERE       ventes.codeProduit = produits.codeProduit
  AND       modele = 'vis'
GROUP BY    couleur ;
```



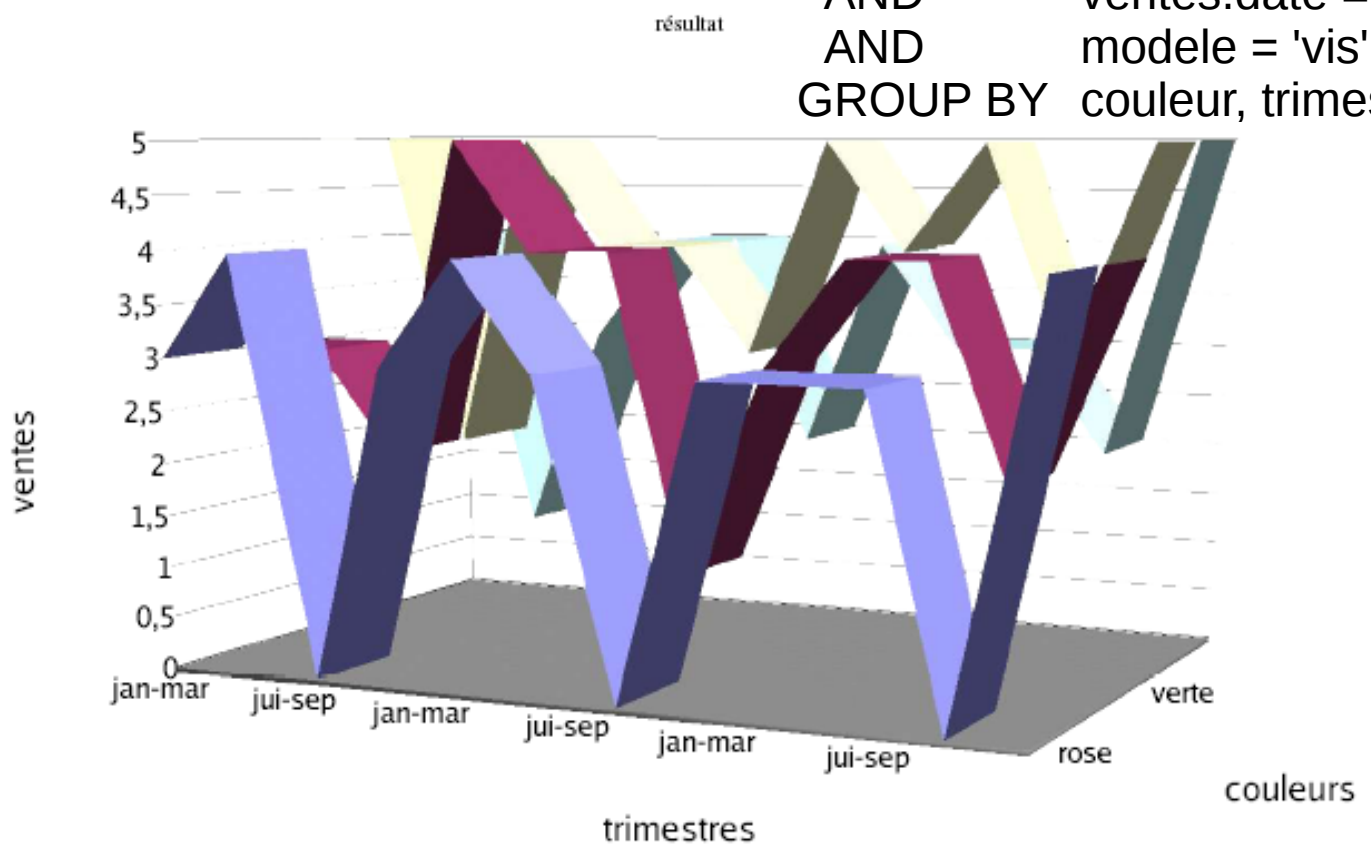
Query for the Scenario (3)




```
SELECT      couleur, annees, SUM(montant)
FROM        ventes, produits, temps
WHERE       ventes.codeProduit = produits.codeProduit
  AND       ventes.date = temps.jour
  AND       modele = 'vis'
GROUP BY    couleur, années ;
```

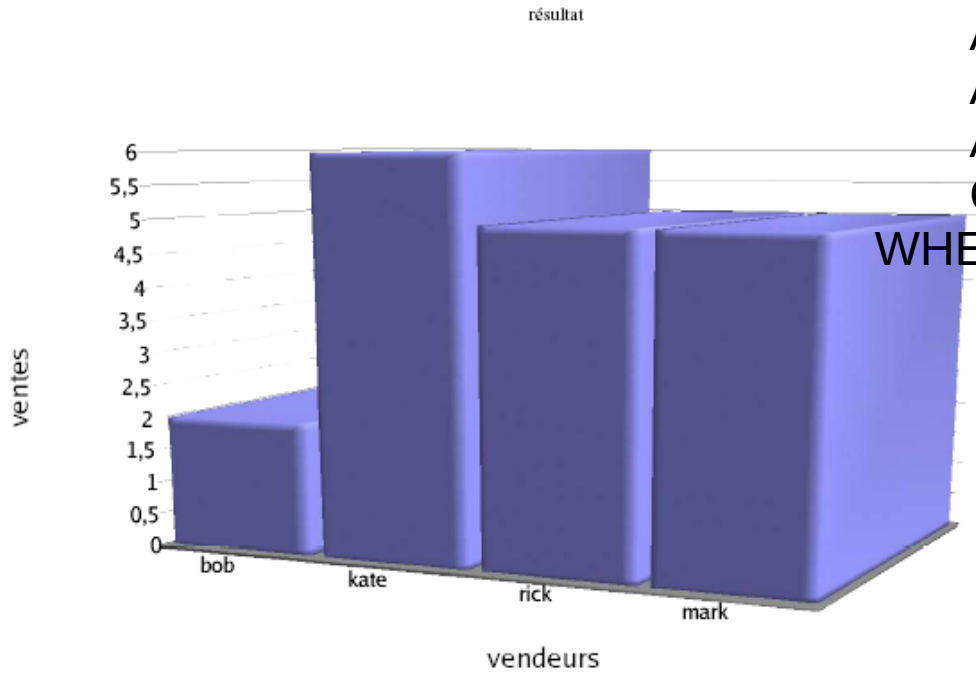
Query for the Scenario (4)

```
SELECT      couleur, trimestre, SUM(montant)
FROM        ventes, produits, temps
WHERE       ventes.codeProduit = produits.codeProduit
  AND       ventes.date = temps.jour
  AND       modele = 'vis'
GROUP BY    couleur, trimestre ;
```



Query for the Scenario (5)

```
SELECT          vendeur, somme
FROM(
    SELECT      vendeur, trimestre,
                SUM(montant) as somme
    FROM        ventes, produits, temps
    WHERE       ventes.codeProduit =
                    produits.codeProduit
    AND         ventes.date = temps.jour
    AND         ventes.vendeur = vendeurs.nom
    AND         modèle = 'vis'
    GROUP BY    trimestre, vendeur)
WHERE           trimestre = “jui-sep”;
```



HIVE - DW for Big Data

Brief History

Google : Map Reduce

Apache : Hadoop

Facebook : Hive

Apache : Hive

Emerged needs

Data Volume

Facebook data :

15 TB/day(2007) → 700 TB/day (2012)

Data Type Variation

semi-structured data , unstructured data, Images, stream

Data Types in HIVE

Basic Data Types:
Int, float, string, …

Complex Data Types : (Non 1NF)
Structure, Array, Map

Compound Data Types

Data Storage in HIVE

Table

Partition (for Group by - Aggregate)

Bucket (for data sampling)

Meta-data

Stored in MySQL severs

Table Creation with HiveQL

```
Create Table page_view(viewTime Int,userId BigInt,
    page_url String, referrer_url String,
    ip_String)

Comment 'Page View Table'

Partitioned by (dt String, country String)
Clustered by (userId) Into 32 Buckets

Row Format Delimited
    Fields Terminated by '\t'
    Line Terminated by '\n'

Stored As SequenceFile;
```

Data Load in Hive

Goal : an infrastructure to data analyses

- **No** *Delete*, *Update*, *Insert into ...Values*

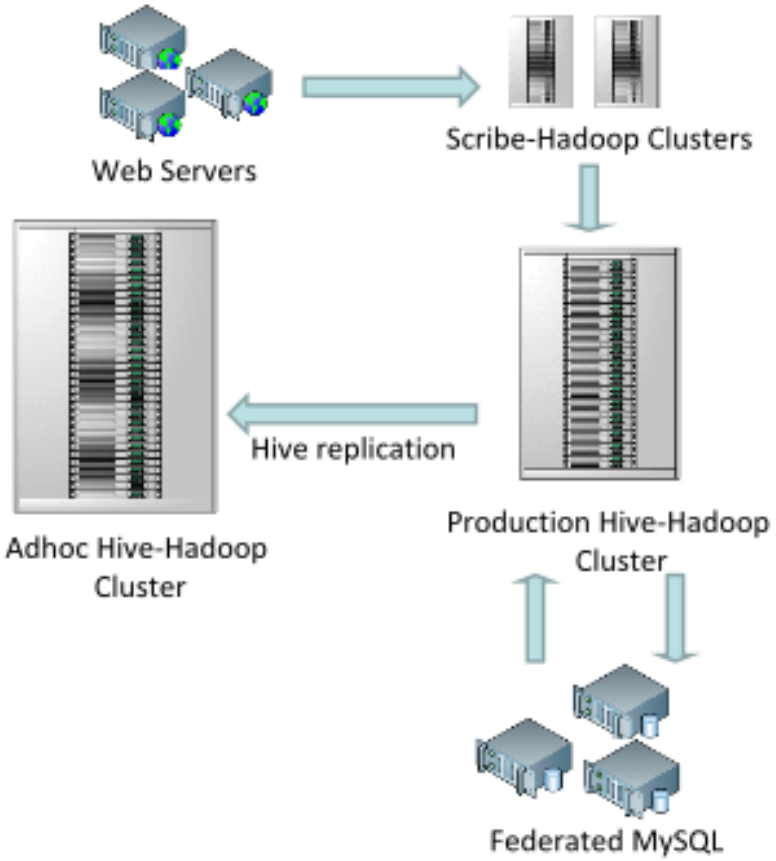
- Data load with directed ways
  Load, Insert into Select

Data Load in HiveQL

```
Load dat Local Inpath 'cite75_99.txt'
    Overwrite into tableTable cite_count
Select cited, count(citing)
From cite
Group by cited;
```

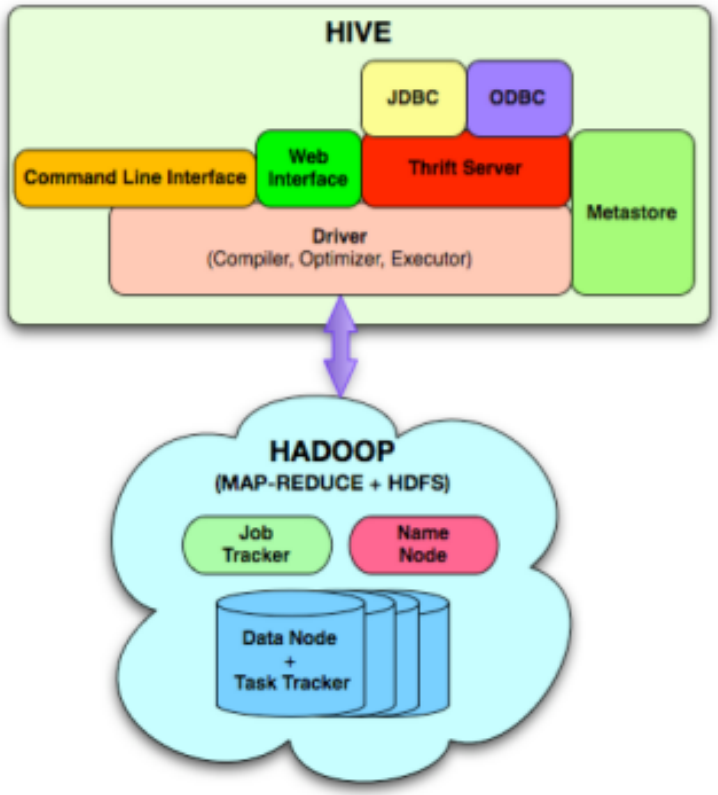
```
Insert Overwrite Table cite_count
Select cited, count(citing)
From cite
Group by cited;
```

Data Flow Architecture



Hive-Hadoop Data Flow Architecture

Data Flow Architecture



Index Example in HIVE (3)

Bitmap Index

Rid	Rouge	Vert	Bleu
r1	0	0	1
r2	0	1	0
r3	0	1	0
r4	0	0	1
r5	1	0	0
r6	0	1	0

Rid	Ville1	Ville2	Ville3
r1	1	0	0
r2	0	1	0
r3	1	0	0
r4	0	0	1
r5	0	1	0
r6	0	1	0

(Vert, Ville2) = {r2, r6}

Index Example in HIVE (1)

Bitmap Index

Rid	Couleur
r1	Bleu
r2	Vert
r3	Vert
r4	Bleu
r5	Rouge
r6	Vert

Rid	Rouge	Vert	Bleu
r1	0	0	1
r2	0	1	0
r3	0	1	0
r4	0	0	1
r5	1	0	0
r6	0	1	0

Index Example in HIVE (2)

Bitmap Index

Rid	Ville
r1	Ville1
r2	Ville2
r3	Ville1
r4	Ville3
r5	Ville2
r6	Ville2

Rid	Ville1	Ville2	Ville3
r1	1	0	0
r2	0	1	0
r3	1	0	0
r4	0	0	1
r5	0	1	0
r6	0	1	0

Data Flux in HIVE (1)

Create Table tab (col1 Int, col2 Int);

Select tab (col1, col2)
Using '/bin/cat' As (newA Int, newB double)

> 1 2.0
> 3 4.0

Data Flux in HIVE (1)

Create Table tab (col1 Int, col2 Int);

Operation

Original Data Type

Select tab (col1, col2)
Using '/bin/cat' As (newA Int, newB double)

> 1 2.0
> 3 4.0

Int → Double

Data Flux in HIVE (2)

```
From(
    Map doctext Using 'python wc_mapper.py'
        As (word, cnt)
    From docs
    Cluster by word ) a
Reduc word, cnt Using 'python wc_reduce.py' ;
```

HIVE vs Traditional DW (1)

Advantages

- Performance
  Daily Analyses one day → several hours

- Extensibility
  Distributed Computing

HIVE vs Traditional DW (2)

Disadvantages

- Analytical operators
  Grouping Sets, Cube, Rollup,
  ancestor, child , etc

- Visualization Tools