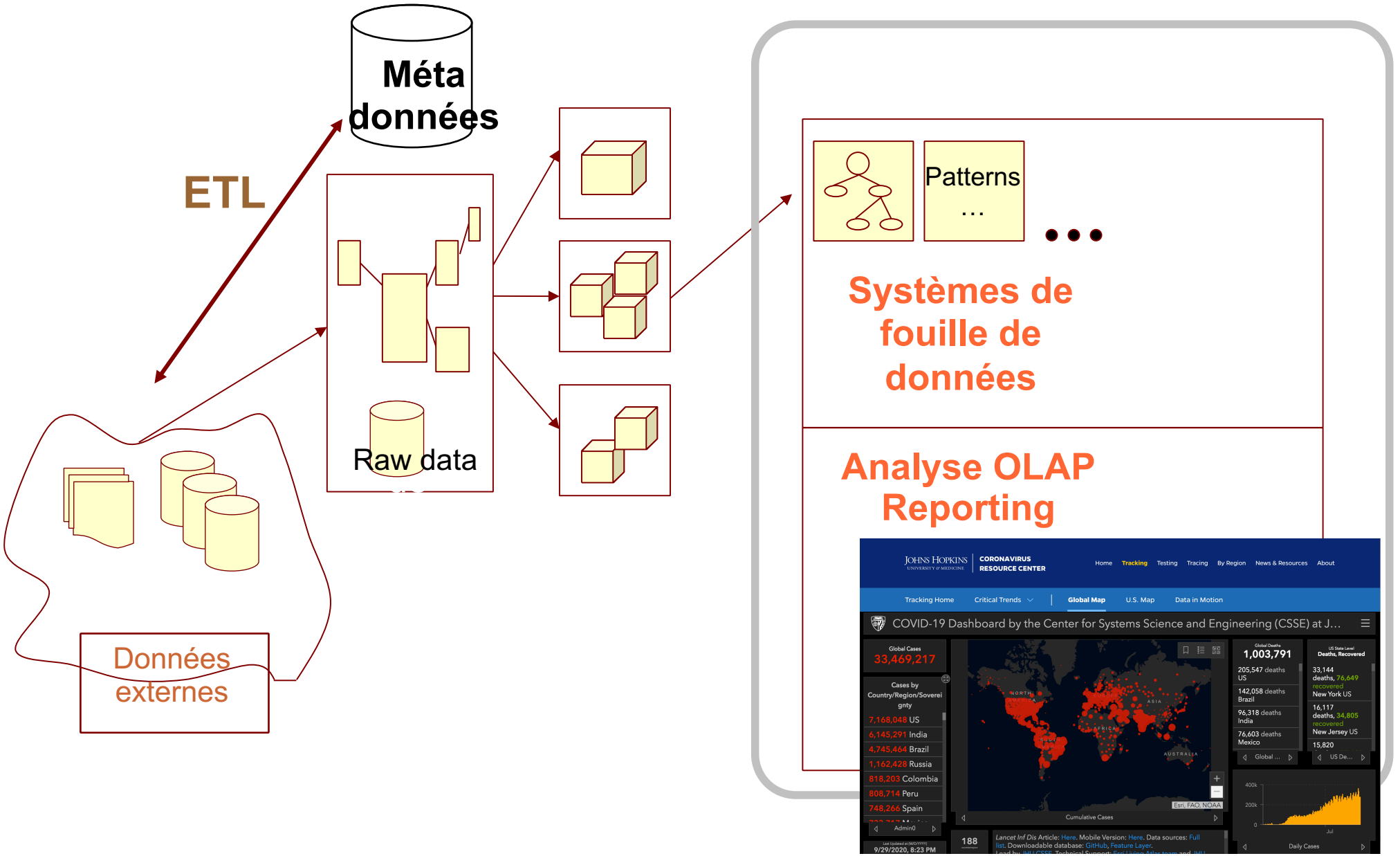
BASES DE DONNES DECISIONNELLES
Entrepôts de données

IG4

Anne LAURENT
UNIVERSITE DE MONTPELLIER
LIRMM – POLYTECH MONTPELLIER
http://www.lirmm.fr/~laurent
Anne.laurent@umontpellier.fr

Bases de données décisionnelles

- Rappels BD opérationnelles…
- Historique BD décisionnelles
- Objectifs et applications
- Modélisation multidimensionnelle
- Challenges


Méta
données
ETL
Raw data
Données
externes
Patterns
...
Systèmes de
fouille de
données
Analyse OLAP
Reporting
JOHNS HOPKINS
CORONAVIRUS
RESOURCE CENTER
Home
Tracking
Testing
Tracing
By Region
News & Resources
About
Tracking Home
Critical Trends
Global Map
U.S. Map
Data in Motion
COVID-19 Dashboard by the Center for Systems Science and Engineering (CSSE) at J...
Global Cases
33,469,217
Cases by
Country/Region/Sovereignty
7,168,048 US
6,145,291 India
4,745,464 Brazil
1,162,428 Russia
818,203 Colombia
808,714 Peru
748,266 Spain
Admin0
Esri, FAO, NOAA
Cumulative Cases
Global Deaths
1,003,791
205,547 deaths
US
142,058 deaths
Brazil
96,318 deaths
India
76,603 deaths
Mexico
Global ...
US State Level
Deaths, Recovered
33,144
deaths, 76,649
recovered
New York US
16,117
deaths, 34,805
recovered
New Jersey US
15,820
US De...
400k
200k
0
Jul
Daily Cases
188
Lancet Inf Dis Article: Here. Mobile Version: Here. Data sources: Full
list. Downloadable database: GitHub, Feature Layer.
9/29/2020, 8:23 PM

Vision multidimensionnelle

- Analyse de *faits* en fonction de *dimensions*
- Analyse des *ventes* selon la *période*, le *lieu*, le type de *produit*
- Gestion de gros volumes de données
- Redondance ?

Entrepôts de données

- Données historisées
- Matérialisées
- Volumineuses
- Sources hétérogènes
- Utilisation décisionnelle

OLAP vs OLTP

OLAP vs. OLTP

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

(**O**n-**L**ine **T**ransaction **P**rocessing)

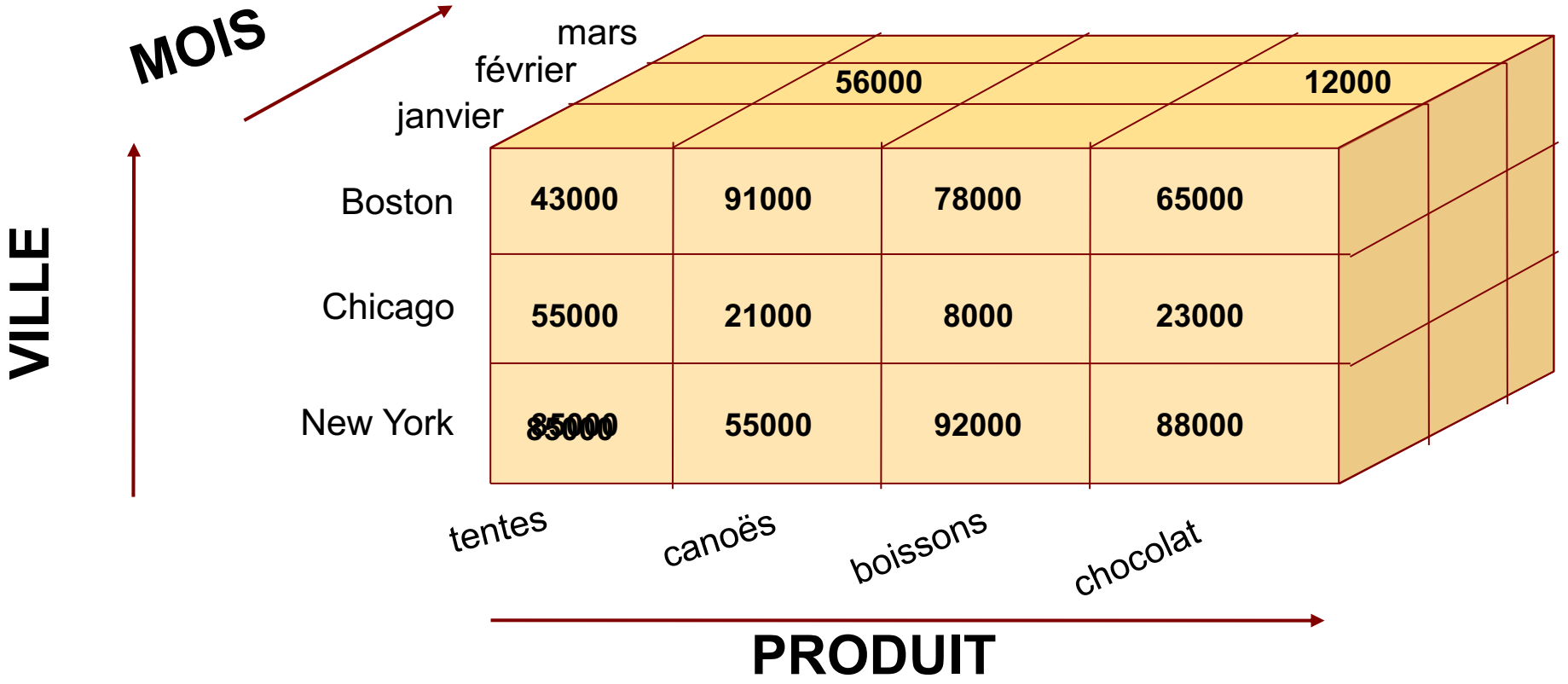
- **Requêtes complexes**
- **Optique décisionnelle**
- **Vision ensembliste (tendances ...)**
- **Destiné aux analystes et décideurs (peu nombreux)**

- **Requêtes simples**
- **Production et Mise à jour des données**
- **Vision au niveau individuel**
- **Destiné aux agents opérationnels (nombreux)**

Exemple

Cube des ventes

C : Produit x Ville x Mois → Ventes



Dimensions

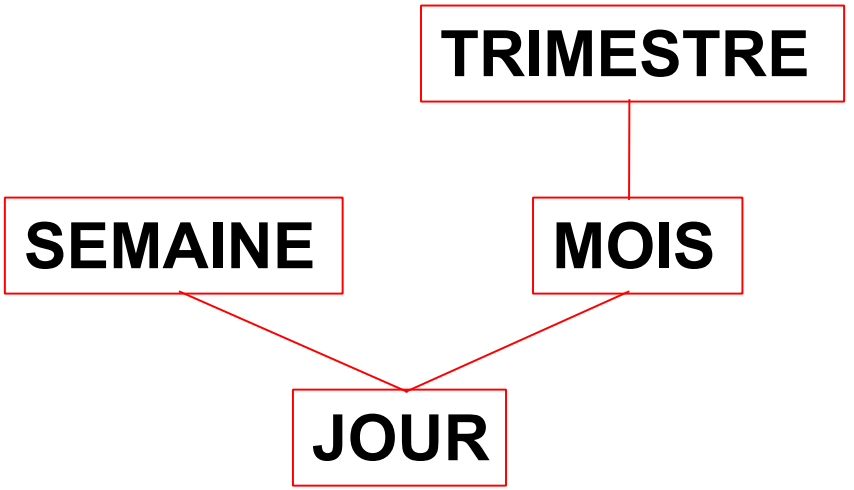
Organisation des informations selon des dimensions *plates*

et des dimensions *hiérarchisées*

hiérarchies ***simples*** (arborescentes)

hiérarchies ***complexes***

Exemple :



ETL – 80% du travail !

- E-xtract
  - Accès aux différentes sources
- T-ransform
  - Nettoyage
  - gestion des inconsistances des données sources
  - formats de données
  - détection des valeurs non valides
- L-oad
  - Chargement dans l'entrepôt

Conception multidimensionnelle

- Tables de faits : entité centrale
  - Objet de l'analyse, taille très importante
- Tables de dimensions : entités périphériques
  - Dimensions de l'analyse, taille peu importante
- Table de faits normalisée (BCNF)
- N-uplets de la tables de faits :
  - Clés étrangères formant une clé primaire
  - Valeurs associées à chaque clé primaire (mesures)
- Associations de type (0,n) – (1,1) connectant les différentes dimensions aux faits

Normalisation des tables

- **Etoile** : tables dimensions non normalisées
- **Flocon** : tables de dimensions normalisées
  - Réduction de la redondance
  - Maintenance simplifiée
  - MAIS navigation coûteuse

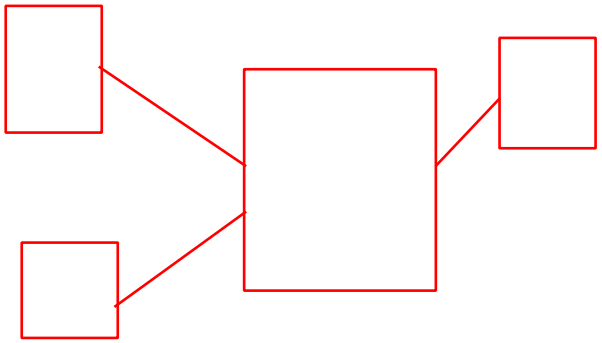


schéma en étoile

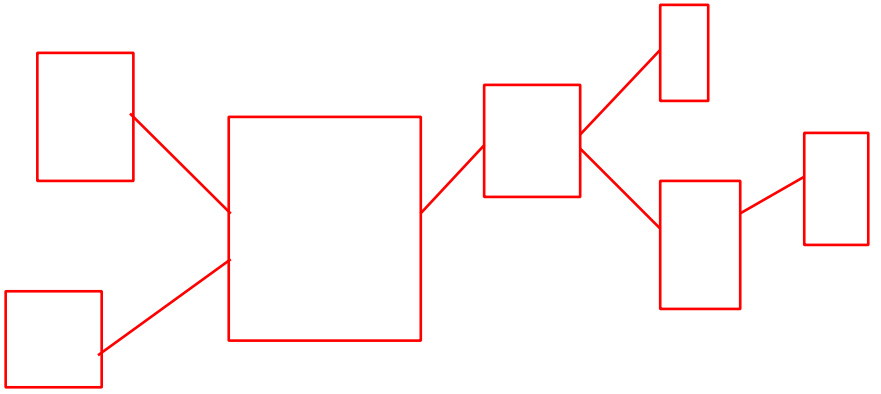
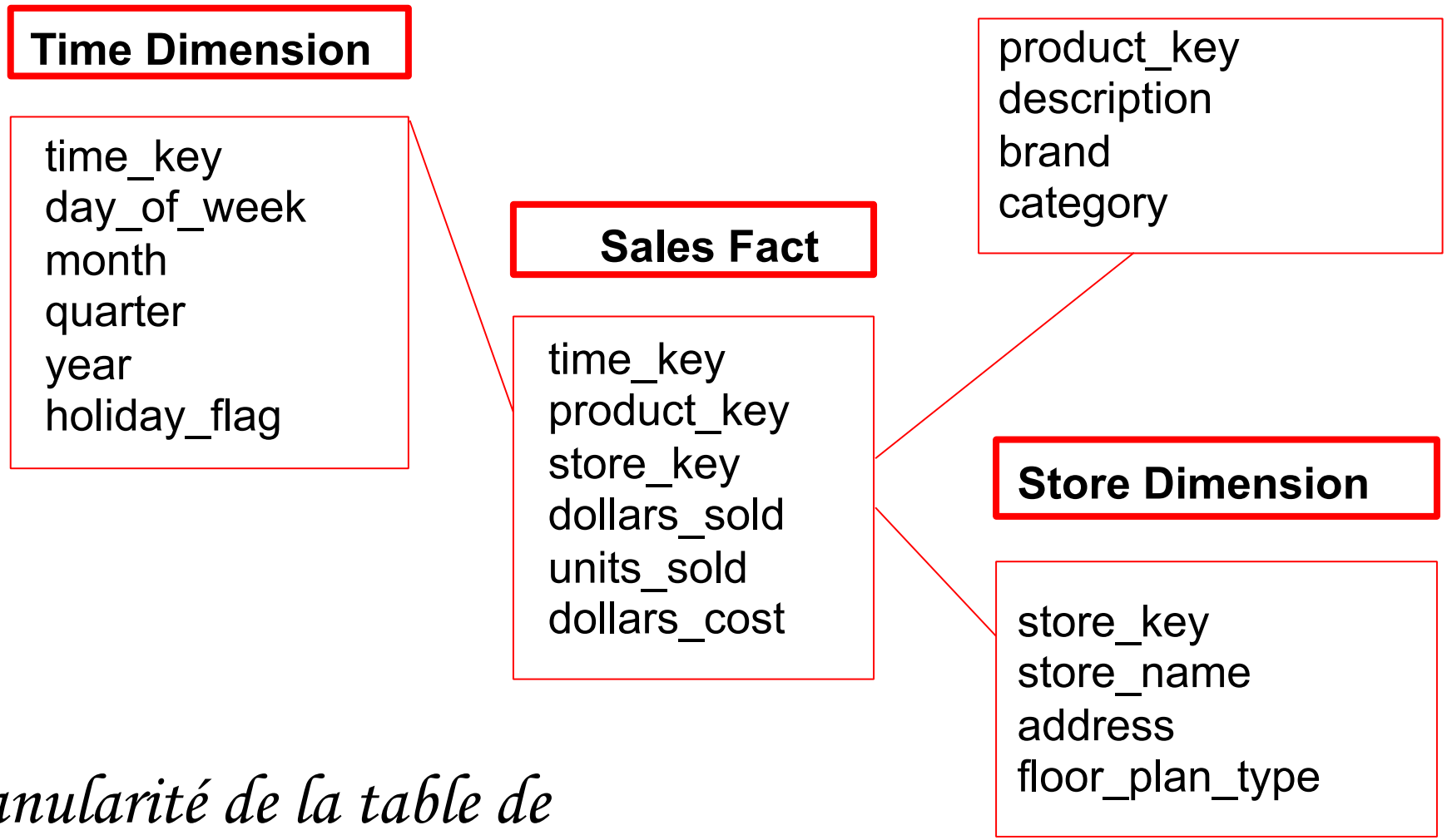
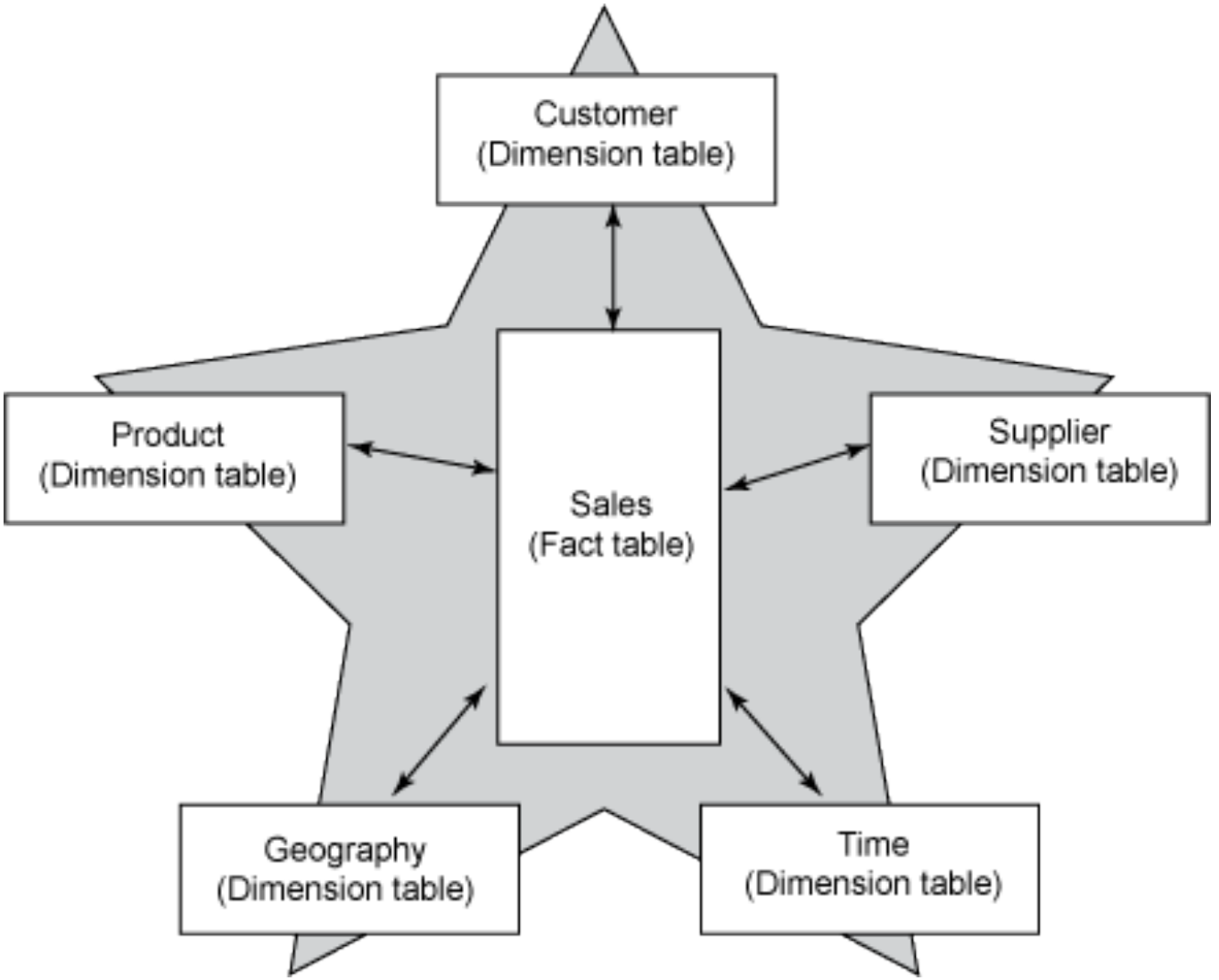


schéma en flocon

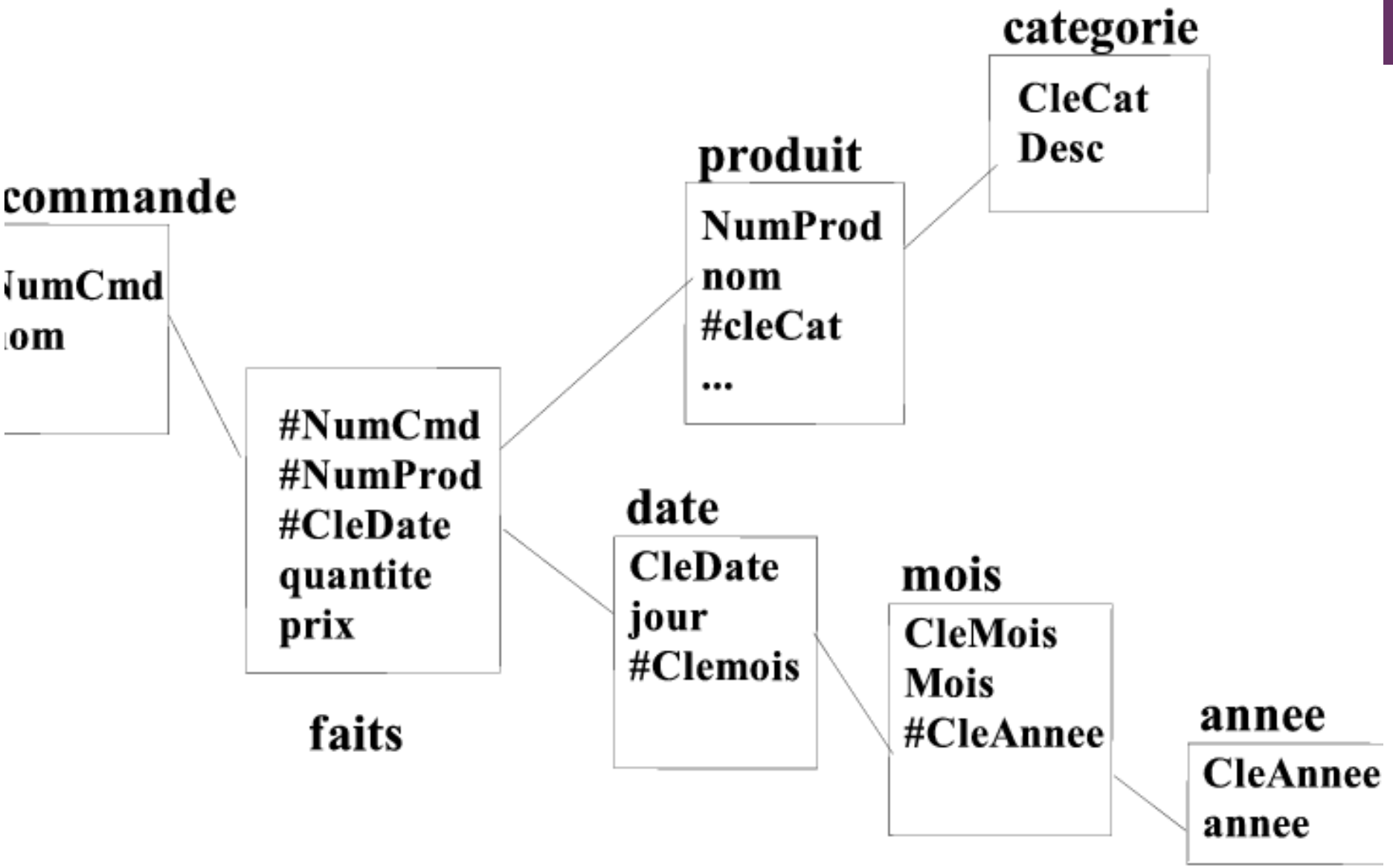
Exemple : étoile



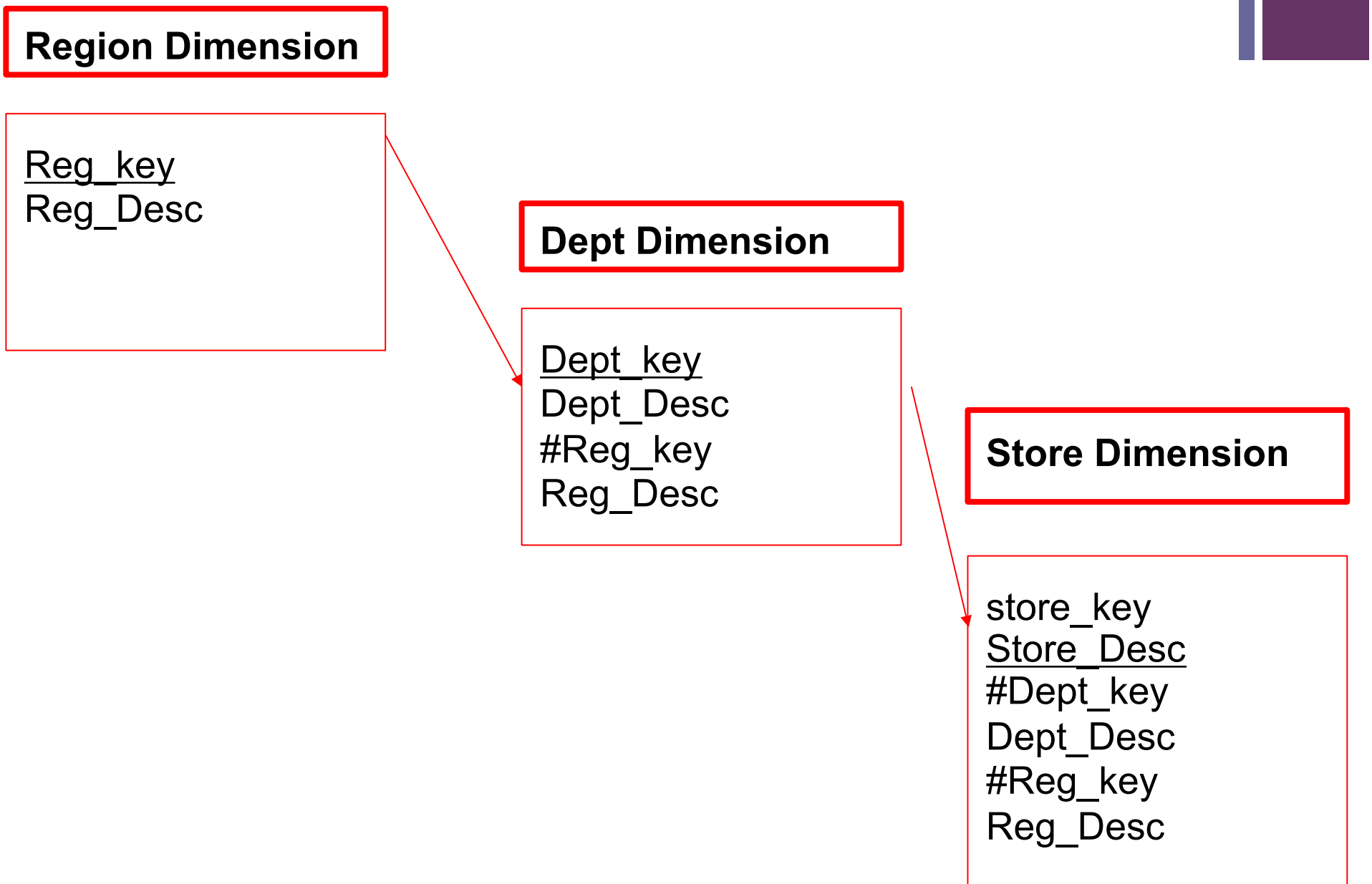
granularité de la table de faits ?


Customer
(Dimension table)
Product
(Dimension table)
Sales
(Fact table)
Supplier
(Dimension table)
Geography
(Dimension table)
Time
(Dimension table)

Exemple : flocon



Dimensional Normal Form



Calcul de l'hypercube

- Quel cube construire ?
- Comment le construire ?
  - Requêtes de type *group by*

SELECT mois, produit, ville, count(*)

FROM Ventes

GROUP BY mois, produit, ville

  - Treillis des cuboides

Stockage du cube

- **ROLAP (Relational OLAP) :** les données sont stockées dans une base de données relationnelle. Le cube n'est pas matérialisé du tout sauf au moment de la phase de requête
- **MOLAP (Multidimensional OLAP) :** tout le cube est matérialisé physiquement
- **HOLAP (Hybrid OLAP) :** seule une partie du cube est matérialisée sous forme multidimensionnelle. Les autres données sont laissées dans la base relationnelle et extraites de manière dynamique au moment des requêtes

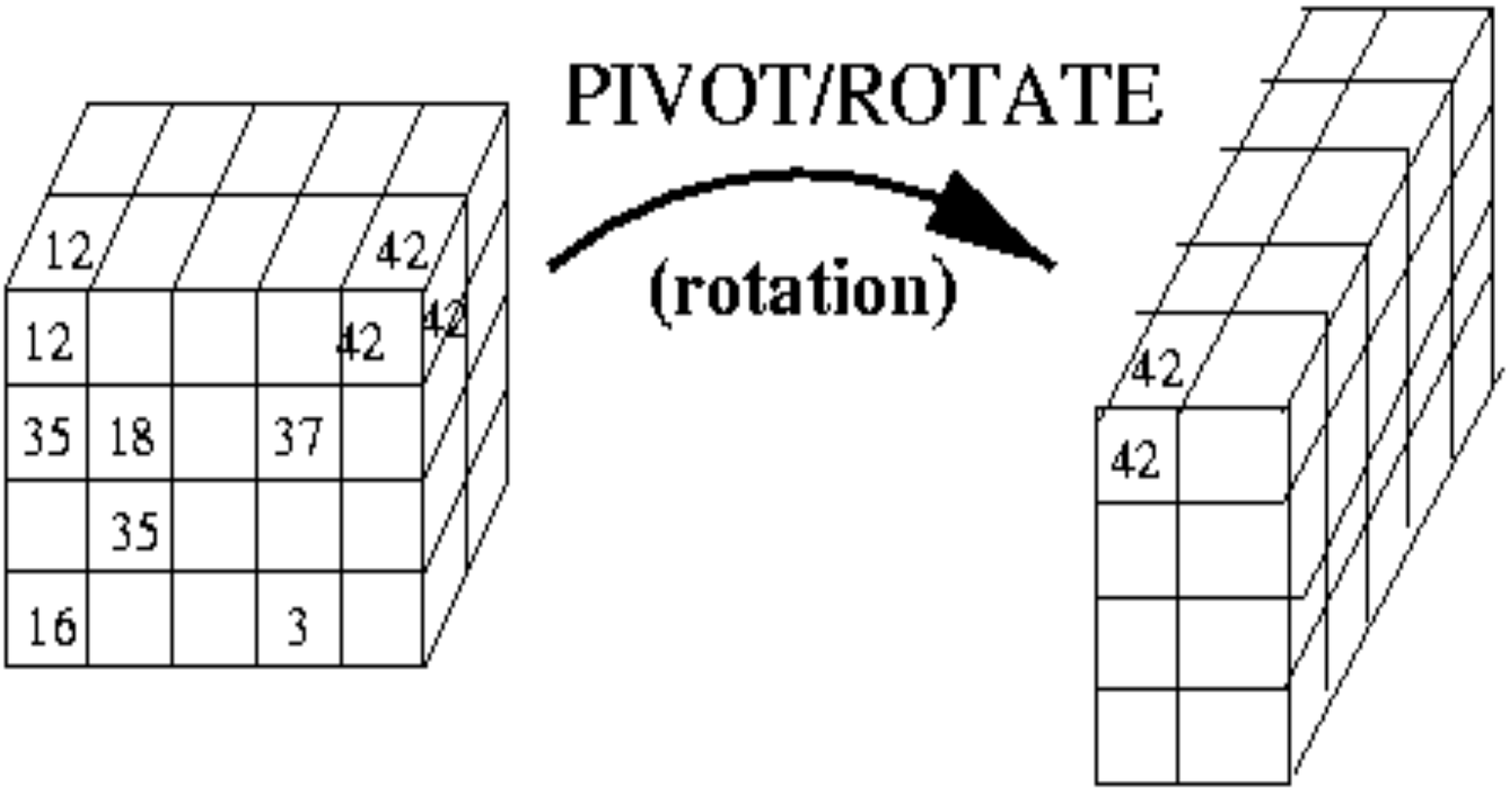
Les cubes sont très clairsemés (sparsity)

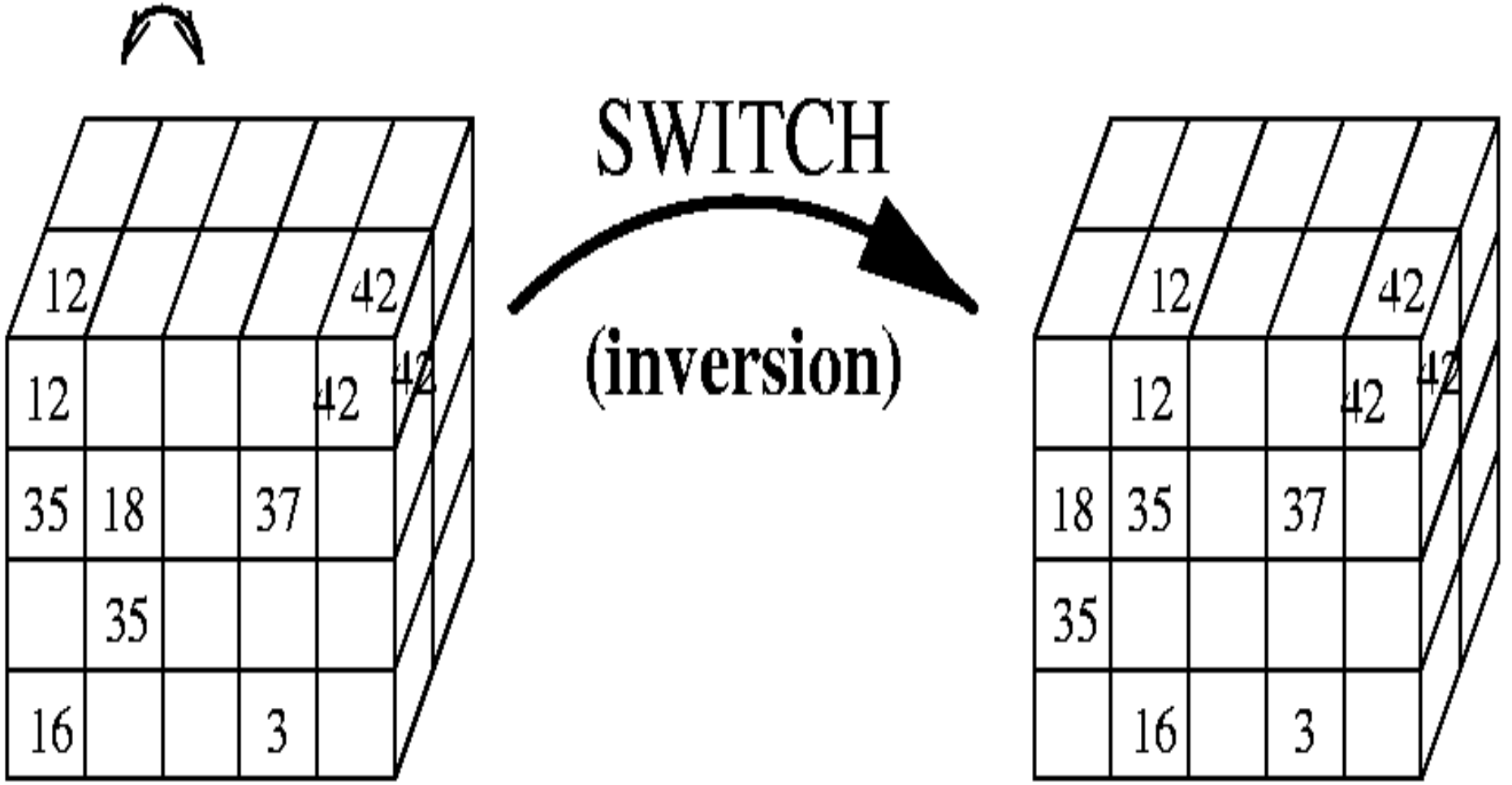
Mise à jour des données

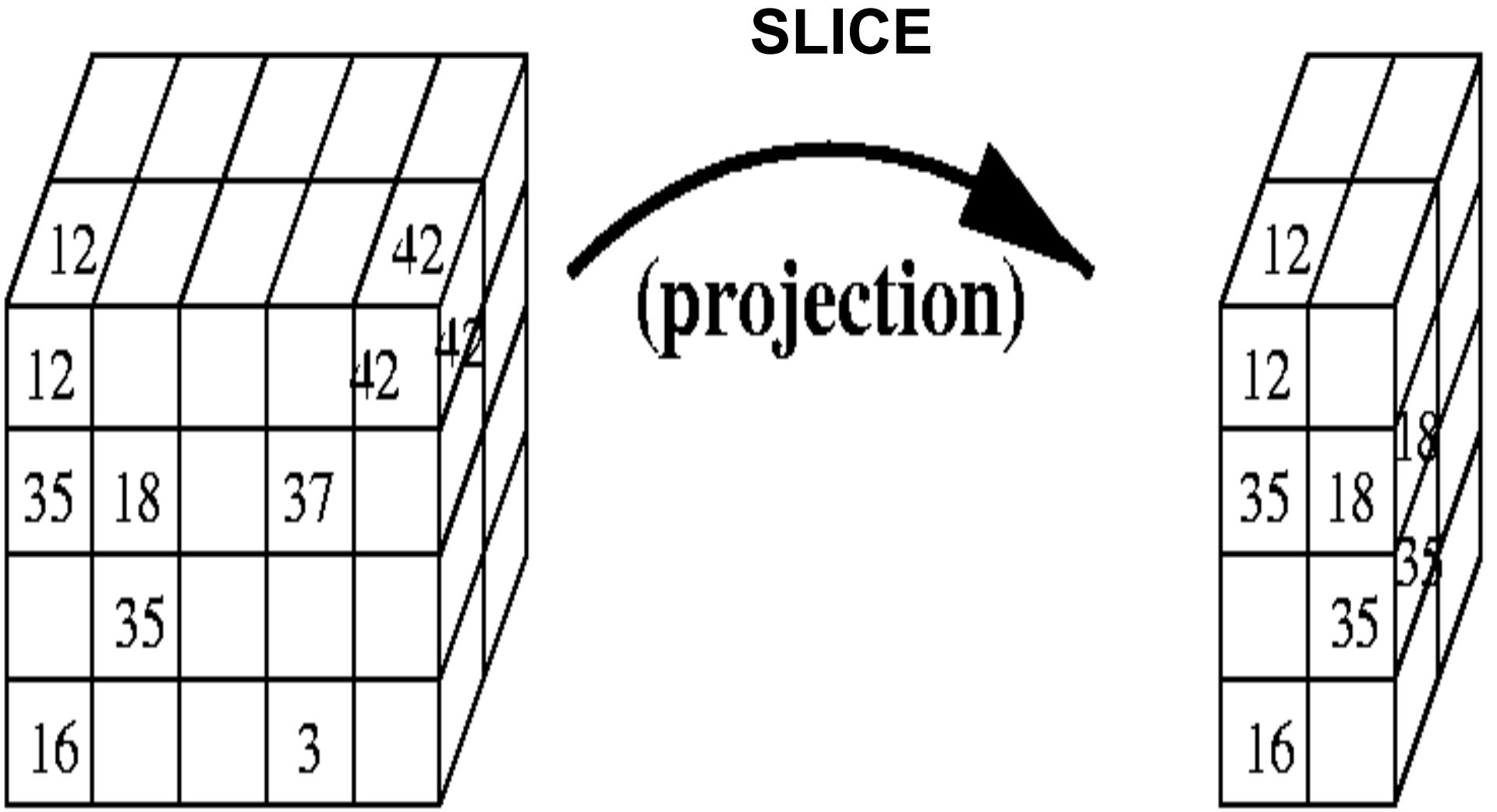
- Incrémentale
- Recalcul total du cube

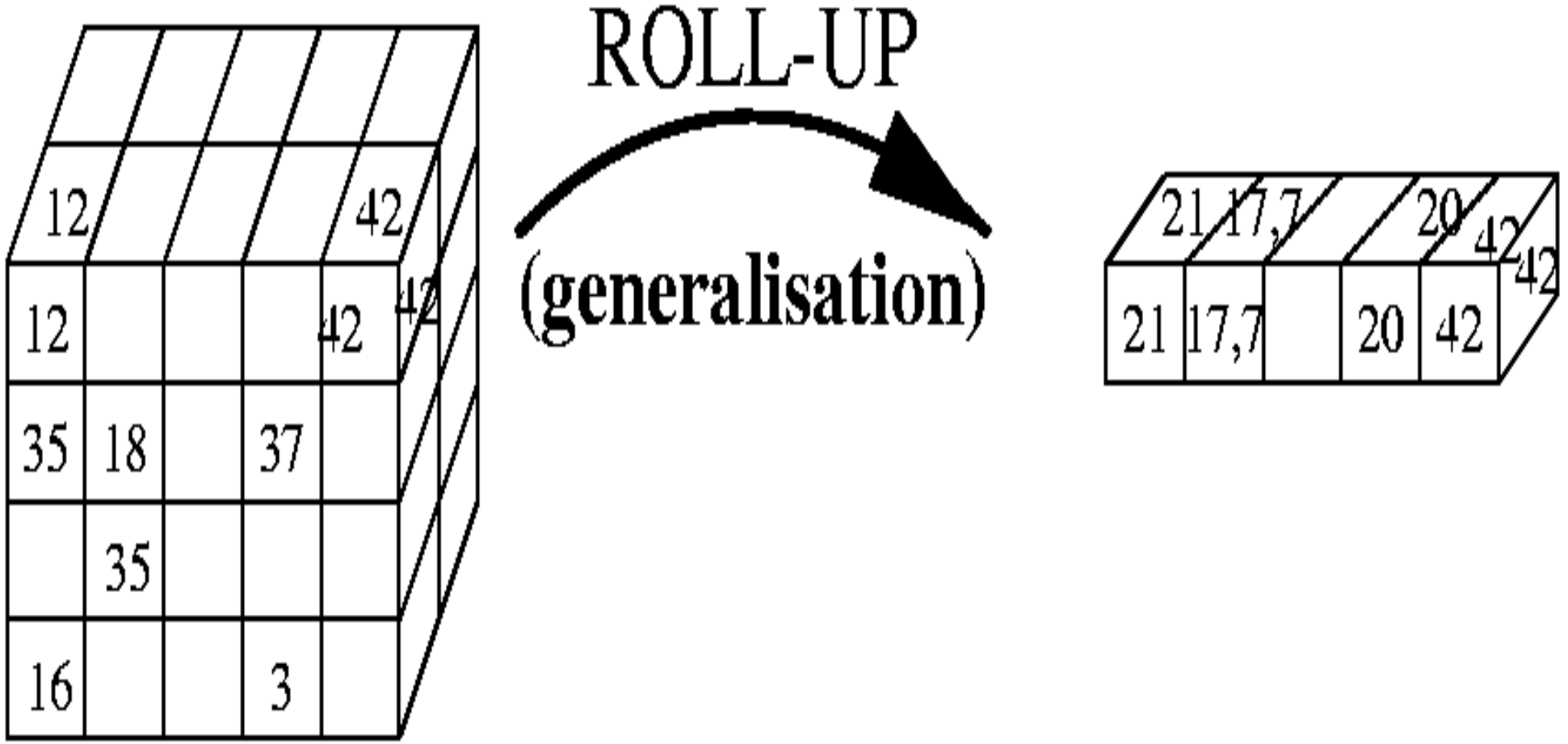
Opérations OLAP

- Visualisation :
  - Rotation
  - Inversion des valeurs de dimensions (switch)
- Modification des données :
  - Sélection sur les cellules (slice)
  - Sélection sur les tranches (dice)
  - Généralisation (roll-up)
  - Spécialisation (drill-down)


PIVOT/ROTATE
(rotation)
12
42
12
42
42
35
18
37
35
16
3
42
42


SWITCH
(inversion)
12
42
12
42
42
35
18
37
35
16
3
12
42
12
42
42
18
35
37
35
16
3


SLICE
(projection)
12
42
42
12
42
35
18
37
35
18
16
3
16


12
42
12
42
42
35
18
37
35
16
3
ROLL-UP
(generalisation)
21
17,7
20
42
42
21
17,7
20
42

Outils commerciaux

- Moteurs / Reporting

- IBM, Microsoft, Oracle
- Business Objects,

Tableau Software,

Zendesk,...

Gartner

Become

Gartner Magic Quadrant & Critical Capabilities

Gartner Magic Quadrant research methodology provides a graphical competitive positioning of four types of technology providers in fast-growing markets: Leaders, Visionaries, Niche Players and Challengers As companion research, Gartner Critical Capabilities notes provide deeper insight into the capability and suitability of providers' IT products and services based on specific or customized use cases.

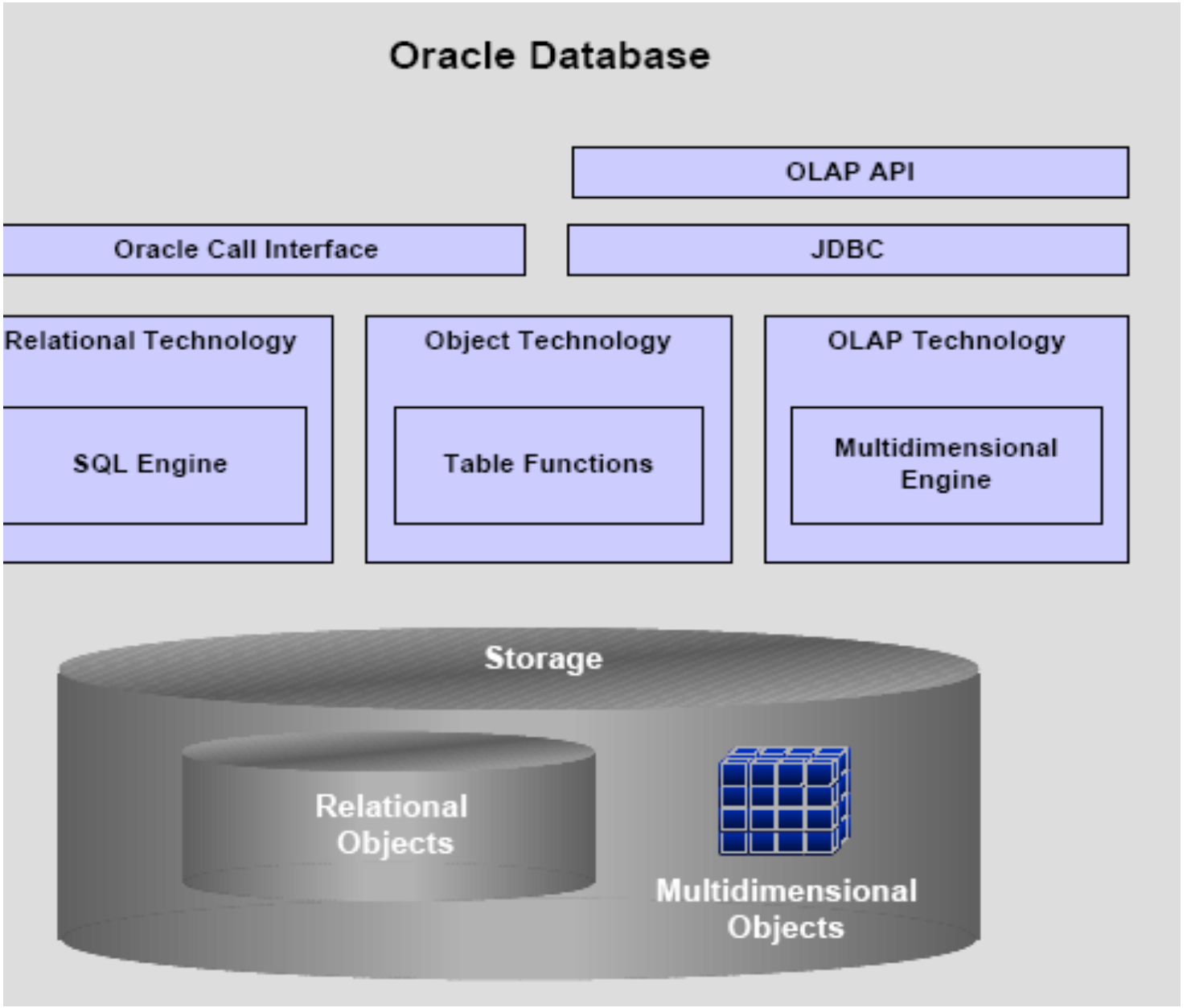
Figure 1. Magic Quadrant for Analytics and Business Intelligence Platforms

Source: Gartner (February 2020)

Oracle et OLAP

- OLAP OPTION
- Analytic Workspace et OLAP DML (TAD)
- Requêtes de partitionnement
- Fonctions d'analyse
- Outils ETL
- Outils d'analyse

MOLAP – ROLAP - HOLAP



Création de BD

- Par exemple Oracle DB Configuration Assistant
- Configurations prédéfinies
  - DW
  - General Purpose
  - Transaction Processing
- Et options (Oracle DW, Oracle OLAP, …)
- Index
  - B*tree
  - Bitmap Index
  - Bitmap Join Index

Partitionnement de données

- Pour gérer de gros volumes de données en les répartissant en mémoire (important si traitements //)
- Différentes stratégies :
  - Range partitioning
  - Hash partitioning
  - List paritioning
  - Composite partitioning

Calcul du cube : group by

```
SQL> select ville, etat, count(*)
from abonne, emprunt, exemplaire
where abonne.num_ab = emprunt.num_ab
      and exemplaire.numero = emprunt.num_ex
group by ville, etat;

VILLE                          ETAT           COUNT(*)
------------------------------ -------------- ----------
BEZIER                          BON                    4
BEZIER                         ABIME                   2
MONTPELLIER                     BON                   18
MONTPELLIER                    ABIME                   2
```

Calcul du cube : group by cube

```
SQL> select ville, etat, count(*)
from abonne, emprunt, exemplaire
where abonne.num_ab = emprunt.num_ab
      and exemplaire.numero = emprunt.num_ex
group by cube(ville, etat);

VILLE                     ETAT          COUNT(*)
------------------------- ------------- ---------
                                              26
                          BON                 22
                          ABIME                4
BEZIER                                         6
BEZIER                    BON                  4
BEZIER                    ABIME                2
MONTPELLIER                                   20
MONTPELLIER               BON                 18
MONTPELLIER               ABIME                2

9 rows selected.
```

Calcul du cube : group by rollup

```
SQL> select ville, etat, count(*)
from abonne, emprunt, exemplaire
where abonne.num_ab = emprunt.num_ab
      and exemplaire.numero = emprunt.num_ex
group by rollup (ville, etat);

VILLE                          ETAT           COUNT(*)
------------------------------ -------------- ----------
BEZIER                          BON                    4
BEZIER                          ABIME                  2
BEZIER                                                 6
MONTPELLIER                     BON                   18
MONTPELLIER                     ABIME                  2
MONTPELLIER                                           20
                                                      26

7 rows selected.
```

Calcul du cube : group by rollup avec fonction DECODE

```
SQL> select decode(grouping (ville),1,'Toutes les villes', ville ) ville,
  2     decode(grouping (etat),1,'Tous etats confondus', etat ) etat,
  3     count(*)
  4  from abonne, emprunt, exemplaire
  5  where abonne.num_ab = emprunt.num_ab
  6        and exemplaire.numero = emprunt.num_ex
  7  group by rollup (ville, etat);
```

VILLE	ETAT	COUNT(*)
BEZIER	BON	4
BEZIER	ABIME	2
BEZIER	Tous etats confondus	6
MONTPELLIER	BON	18
MONTPELLIER	ABIME	2
MONTPELLIER	Tous etats confondus	20
Toutes les villes	Tous etats confondus	26

Calcul du cube : group by grouping sets

```
SQL> select ville, etat, count(*)
from abonne, emprunt, exemplaire
where abonne.num_ab = emprunt.num_ab
      and exemplaire.numero = emprunt.num_ex
group by grouping sets ((ville, etat), ()) ;

VILLE                          ETAT           COUNT(*)
------------------------------ -------------- ----------
BEZIER                          BON                    4
BEZIER                          ABIME                  2
MONTPELLIER                     BON                   18
MONTPELLIER                     ABIME                  2
                                                      26
```

Fonctions d'analyse : rank

```
SQL> select *
from (select ville,etat,count(*),
        rank() over (order by count(*) desc) as rnk
        from emprunt, abonne, exemplaire
        where emprunt.num_ab = abonne.num_ab and
exemplaire.numero=emprunt.num_ex
        group by ville, etat) ;
```

VILLE	ETAT	COUNT(*)	RNK
MONTPELLIER	BON	18	1
BEZIER	BON	4	2
BEZIER	ABIME	2	3
MONTPELLIER	ABIME	2	3

Fonctions d'analyse : TOP-N

```
SQL> select *
from (select ville,etat,count(*),
        rank() over (order by count(*) desc) as rnk
        from emprunt, abonne, exemplaire
        where emprunt.num_ab = abonne.num_ab and
exemplaire.numero=emprunt.num_ex
        group by ville, etat)
where rnk <= 2 ;

VILLE                          ETAT          COUNT(*)  RNK
------------------------------ ------------- --------- ----------
MONTPELLIER                     BON                  18          1
BEZIER                          BON                   4          2
```

Fonctions d'analyse : BOTTOM-N

```
SQL> select *
from (select ville,etat,count(*),
        rank() over (order by count(*)) as rnk
        from emprunt, abonne, exemplaire
        where emprunt.num_ab = abonne.num_ab and
exemplaire.numero=emprunt.num_ex
        group by ville, etat)
where rnk <= 2 ;

VILLE                          ETAT           COUNT(*)       RNK
------------------------------ -------------- ---------- ----------
BEZIER                         ABIME                   2          1
MONTPELLIER                    ABIME                   2          1
```

Fonctions d'analyse : ratio_to_repor

```
SQL> select abonne.num_ab, ville, count(*),
        ratio_to_report(count(*)) over (partition by ville) as
ratio
from emprunt, abonne
where emprunt.num_ab = abonne.num_ab
group by ville, abonne.num_ab ;
```

NUM_AB	VILLE	COUNT(*)	RATIO
911007	BEZIER	6	1
901001	MONTPELLIER	4	,2
902043	MONTPELLIER	4	,2
902075	MONTPELLIER	2	,1
911021	MONTPELLIER	1	,05
911023	MONTPELLIER	6	,3
921102	MONTPELLIER	3	,15

Vues matérialisées : création

```
SQL> CREATE MATERIALIZED VIEW OLAPV_EMPRUNTS
  2  REFRESH START WITH SYSDATE NEXT SYSDATE+1
  3  ENABLE QUERY REWRITE
  4  AS
  5  SELECT VILLE, ETAT, COUNT(*)
  6  FROM EMPRUNT, EXEMPLAIRE, ABONNE
  7  WHERE EMPRUNT.NUM_AB=ABONNE.NUM_AB AND
EXEMPLAIRE.NUMERO=EMPRUNT.NUM_EX
  8  GROUP BY VILLE, ETAT;

Materialized view created.
```

Vues matérialisées : interrogation

```
SQL> select * from olapv_emprunts;

VILLE                          ETAT            COUNT(*)
------------------------------ --------------- ----------
BEZIER                          BON                     4
BEZIER                         ABIME                    2
MONTPELLIER                     BON                    18
MONTPELLIER                     ABIME                   2

SQL> insert into emprunt
values(911007,1010,SYSDATE,NULL,NULL,NULL);

1 row created.

SQL> select * from olapv_emprunts;

VILLE                          ETAT            COUNT(*)
------------------------------ --------------- ----------
BEZIER                          BON                     4
BEZIER                         ABIME                    2
MONTPELLIER                     BON                    18
MONTPELLIER                     ABIME                   2
```

Vues matérialisées : rafraîchissement

```
SQL> begin
        dbms_mview.refresh('olapv_emprunts');
end;
/

PL/SQL procedure successfully completed.

SQL>  select * from olapv_emprunts;

VILLE                          ETAT           COUNT(*)
------------------------------ -------------- ----------
BEZIER                          BON                    5
BEZIER                         ABIME                   2
MONTPELLIER                     BON                   18
MONTPELLIER                    ABIME                   2
```

Dimensions : clés

- On ajoute souvent une clé de subtsitution (pour remplacer la combinaison d'attributs clé initiale). Surrogate key – natural/business key
  - Exemple code_produit -> id_produit
  - Type numérique -> accélère
  - La clé d'affaire peut changer -> indépendance, historique possible des changements

Dimensions : domaines évolutifs

- Dimensions à évolution lente / SCD (slowly Changing Dimension)
  - Changement de noms
  - Changement de statut (livre devient abimé, mariage, …)

  - 3 solutions
    - Écrasement de la valeur
    - Versionnement
    - Valeur d'origine/valeur courante
    - Attention, parfois : 2 valeurs co-existent (2 prduits restent en rayon avec des noms différents)

Dimensions : domaines évolutifs

- Dimensions à évolution rapide/ RCD (Rapid Changing Dimension)
  - Changement d'année d'étude

  - Grande dimension : clients
  - -> création de Mini-dimension contenant les attributs sur lesquels porte l'évolution

Estimer le volume de l'entrepôt

- Prendre en compte
  - Table de faits
  - Dimensions significatives
  - Agrégats
  - Index
  - Saisonnalité (ventes)
  - …

Conclusion

- Nécessité d'outils dédiés aux bases de données décisionnelles
- Navigation OLAP / fouille de données / reporting / tableaux de bord
- il existe de nombreuses (autres) méthodes
- et pas de meilleure
- méthode à choisir selon
  - les données (continues ? manquantes ? volumineuses ? denses ? ...)
  - la tâche
  - le temps de calcul dont on dispose
- autres types de données

Bibliographie

- Ralph Kimball, *Entrepôts de Données*, Vuibert, 2002.
- Jiawei Han, Micheline Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publishers, 2000.
- http://www.oracle.com