

# Analyse des données

CM6 : SQL avancé et OLAP

Mickaël Martin Nevot

V1.0.0



Cette œuvre est mise à disposition selon les termes de la  
[licence Creative Commons Attribution - Pas d'Utilisation Commerciale - Partage à l'Identique  
3.0 non transposé.](https://creativecommons.org/licenses/by-nc-sa/3.0/)

# Analyse des données

- I. Présentation
- II. OLAP
- III. Skyline

# Rappel : SQL

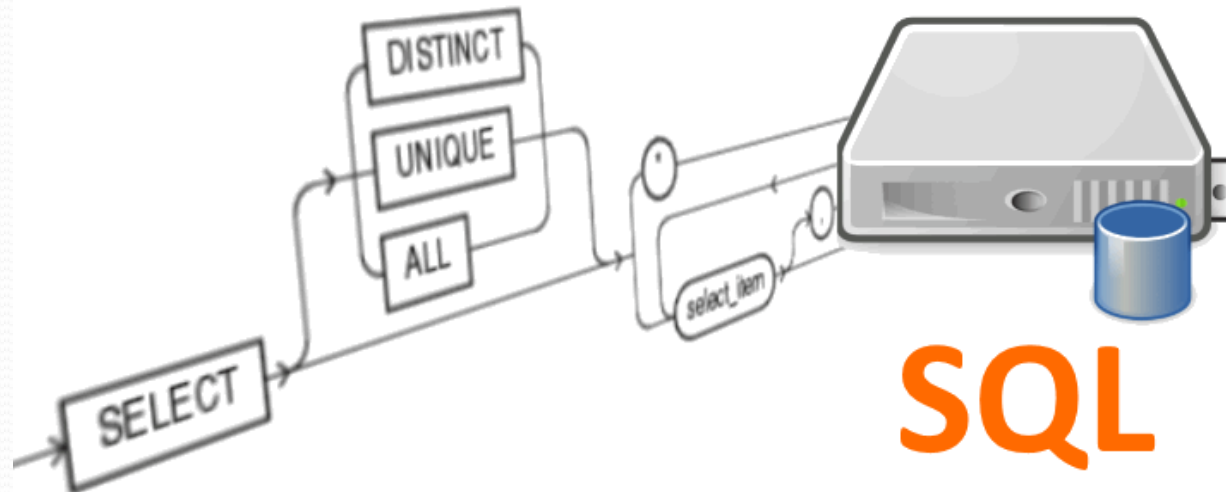
- *Structured query language* : langage de requête structurée
- **Un seul langage général** :
  - Langage de description de données (LDD)
  - **Langage de manipulation de données (LMD)**
  - Langage de description des schémas physiques (LDSP)
  - Contrôle et administration



# SQL

# Rappel : LMD

- **Langage de manipulation de données (LMD)**
- Recherche des données d'une BD
- Mise à jour des données d'une BD



Le résultat d'une requête relationnelle est une relation

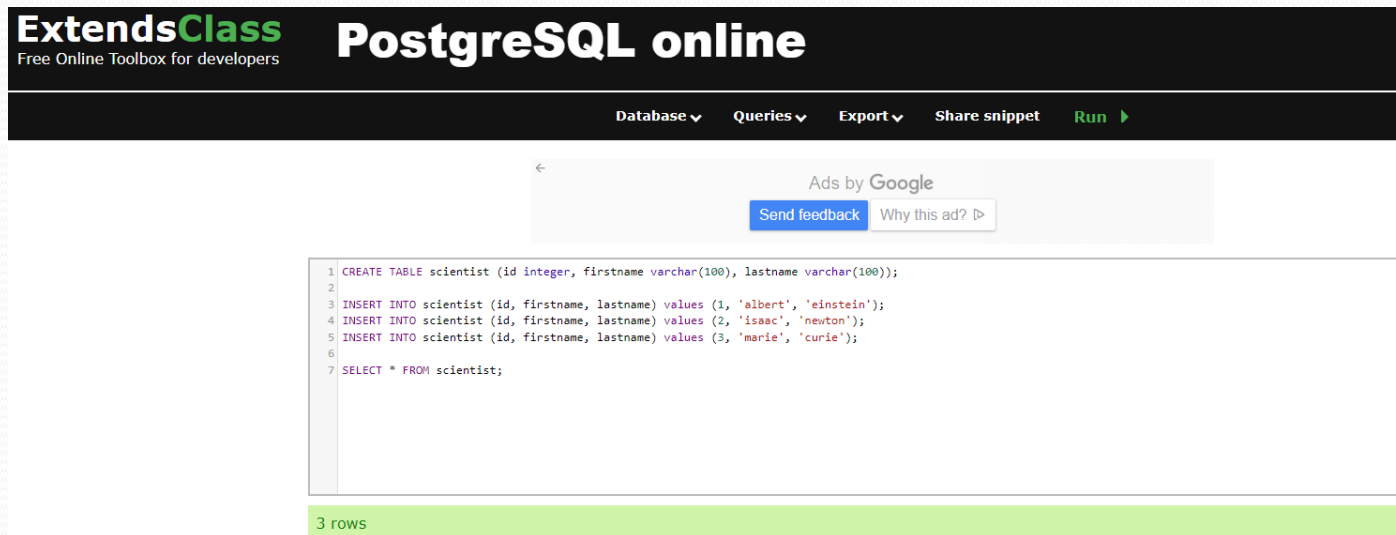
# Pour tester

- Interpréteur en ligne :

<https://extendsclass.com/postgresql-online.html>

- Documentation :

<https://www.postgresql.org/docs/current>



**ExtendsClass**  
Free Online Toolbox for developers

## PostgreSQL online

Database ▾ Queries ▾ Export ▾ Share snippet Run ▶

Ads by Google  
[Send feedback](#) [Why this ad? ▶](#)

```
1 CREATE TABLE scientist (id integer, firstname varchar(100), lastname varchar(100));
2
3 INSERT INTO scientist (id, firstname, lastname) values (1, 'albert', 'einstein');
4 INSERT INTO scientist (id, firstname, lastname) values (2, 'isaac', 'newton');
5 INSERT INTO scientist (id, firstname, lastname) values (3, 'marie', 'curie');
6
7 SELECT * FROM scientist;
```

3 rows

id	firstname	lastname
1	albert	einstein
2	isaac	newton
3	marie	curie

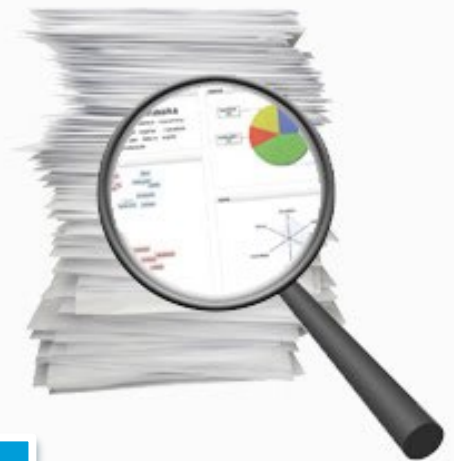
En SQL, les retours à la ligne sont non déterministes

# Rappel : Recherche de données

- Sélection :
  - SELECT : définit les attributs de la relation résultante
  - FROM : spécifie les relations sur lesquelles porte la recherche
  - WHERE : indique des conditions de restriction
  - Les autres clauses seront traitées ultérieurement

## Syntaxe :

```
SELECT [DISTINCT] ...  
FROM ...  
[WHERE ...]  
[GROUP BY ...]  
[HAVING ...]  
[ORDER BY ...]
```



SELECT, FROM, WHERE, etc. sont des clauses

# Rappel : Fonctions d'agrégation

- Agrège les tuples (un seul résultat)
- Uniquement dans clause SELECT (ou HAVING)
- Fonctions :
  - SUM(...) : somme
  - AVG(...) : moyenne algébrique
  - MIN(...), MAX(...) : valeur min., max.
  - COUNT(...) : dénombrement
  - Etc.

Uniquement pour des attributs numériques

Aussi appelées fonctions de calcul intégrées

Pas de fonction d'agrégat en paramètre d'une fonction d'agrégat

# Rappel : Partitionnement

- Partitionner les données afin d'effectuer des calculs par ensemble de données groupées :

```
SELECT prenom, COUNT(*)  
FROM Etudiant WHERE sexe = 'M'  
GROUP BY prenom;
```

On obtient autant de partitions que de valeurs distinctes dans l'ensemble d'attributs de la clause GROUP BY

- Partitionnement avec condition de sélection :

```
SELECT prenom, COUNT(*)  
FROM Etudiant WHERE sexe = 'M'  
GROUP BY prenom  
HAVING COUNT (DISTINCT ide) >= 2;
```

Les agrégations s'appliquent à chaque valeur de l'ensemble de la clause SELECT

**Règle d'or** : tous les attributs non agrégés projetés dans un SELECT doivent figurer dans le GROUP BY, et **inversement**



# Rappel : Partitionnement

R	A	B	C
	a1	b1	c1
	a1	b1	c2
	a2	b1	c1
	a2	b2	c1
	a1	b2	c1
	a2	b1	c1

```
SELECT A, B, COUNT(*)
FROM R
GROUP BY A, B;
```

T1	A	B	(*)
	a1	b1	2
	a1	b2	1
	a2	b1	2
	a2	b2	1

```
SELECT C, A, COUNT(*)
FROM R
GROUP BY C, A;
```

T2	C	A	(*)
	c1	a1	2
	c1	a2	3
	c2	a1	1

# OLAP : agrégation étendue

- GROUP BY ne construit **qu'une seule partition** des résultats
- Possibilité d'y adjoindre des opérateurs afin de visualiser **plusieurs partitions** en même temps



**OLAP** (*online analytical processing*) : traitement analytique en ligne (couramment utilisé en informatique décisionnelle) permet l'analyse sur-le-champ d'informations selon plusieurs axes, dans le but d'obtenir des rapports de synthèse

# OLAP : sélection de partitions

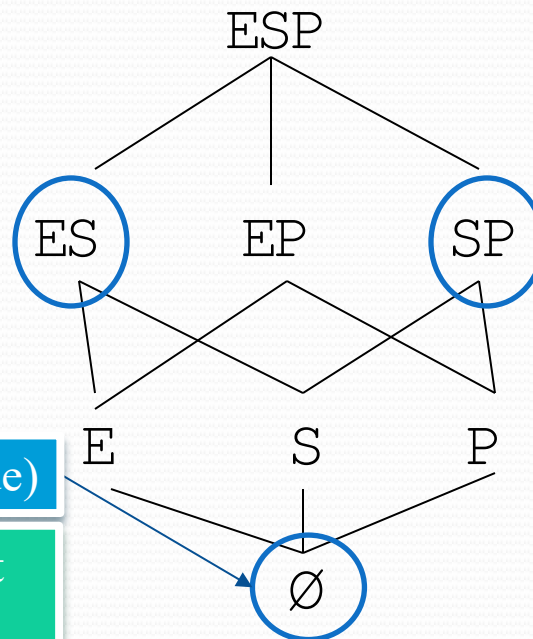
-- Donner les durées totales des stages en entreprise effectués d'une façon générale ; par  
-- étudiant et par société ; et enfin par société et par tuteur.

```
SELECT ide, ids, idp, SUM(duree) AS dureet
```

```
FROM Convention
```

```
GROUP BY GROUPING SETS((), (ide, ids), (ids, idp));
```

ide	ids	idp	dureet
			41
17	8		4
8	21		5
8	8		3
17	34		6
12	13		6
8	34		6
15	13		5
15	8		6
	13	12	11
	8	2	3
	8	4	6
	8	7	4
	21	19	5
	34	53	12



Pas de partition (partition vide)

Les résultats obtenus forment un treillis de données

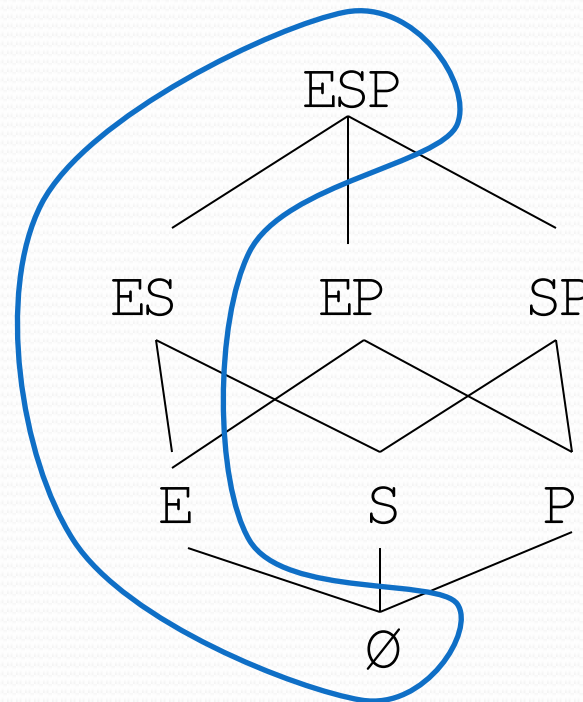
E : ide  
S : ids  
P : idp  
Ø : absence de partition

# OLAP : chemin de partitions

-- Donner les durées totales des stages en entreprise effectués d'une façon générale ; par  
-- étudiant ; par étudiant et par société ; et enfin par étudiant, par société et par tuteur.

```
SELECT ide, ids, idp, SUM(duree) AS dureet
FROM Convention
GROUP BY ROLLUP(ide, ids, idp);
```

ide	ids	idp	dureet
			41
15	13	12	5
8	21	19	5
17	8	7	4
8	8	2	3
15	8	4	6
8	34	53	6
12	13	12	6
17	34	53	6
17	8		4
8	21		5
8	8		3
17	34		6
12	13		6
8	34		6
15	13		5
15	8		6
15			11
17			10
12			6
8			14

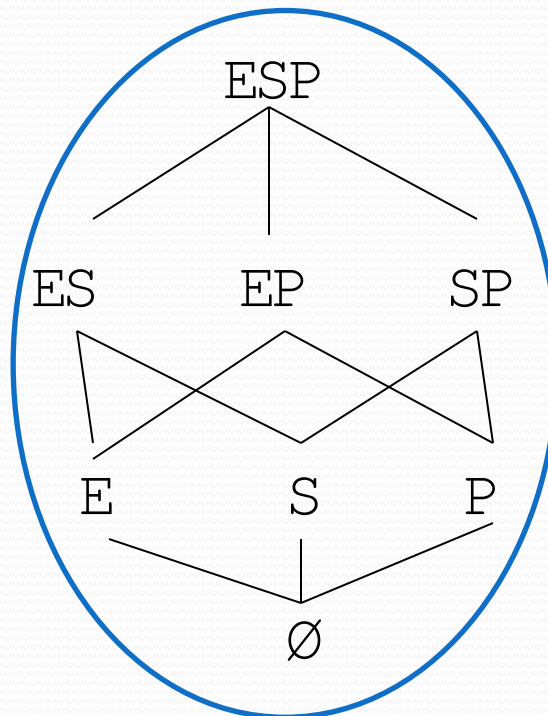


E : ide  
S : ids  
P : idp  
∅ : absence de partition

# OLAP : cube (toutes les partitions)

-- Donner les durées totales des stages en entreprise effectués d'une façon générale ; par  
-- étudiant ; par société ; par tuteur ; par étudiant et par société ; par étudiant et par  
-- tuteur ; par société et par tuteur ; et enfin par étudiant, par société et par tuteur.

```
SELECT ide, ids, idp, SUM(duree) AS dureet  
FROM Convention  
GROUP BY CUBE(ide, ids, idp);
```



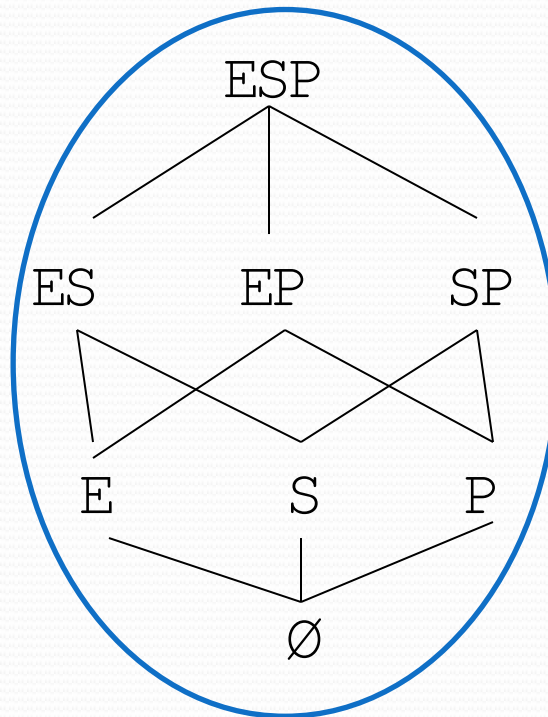
45 résultats

E : ide  
S : ids  
P : idp  
∅ : absence de partition

# OLAP : cube iceberg

-- Donner les durées totales des stages de sept mois ou plus en entreprise effectués d'une  
 -- façon générale ; par étudiant ; par société ; par tuteur ; par étudiant et par société ;  
 -- par étudiant et par tuteur ; par société et par tuteur ; et enfin par étudiant, par  
 -- société et par tuteur.

```
SELECT ide, ids, idp, SUM(duree) AS dureet
FROM Convention
GROUP BY CUBE(ide, ids, idp)
HAVING SUM(duree) >= 7;
```

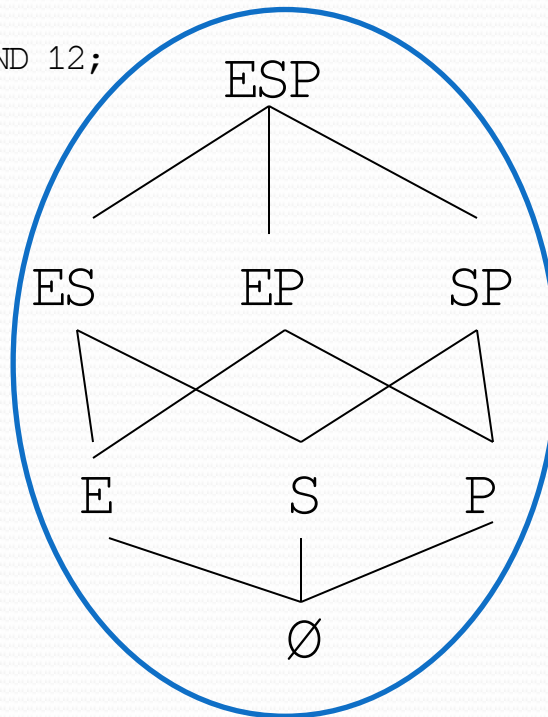


ide	ids	idp	dureet
			41
15			11
17			10
8			14
	13	12	11
	34	53	12
	34		12
	13		11
	8		13
		53	12
		12	11

# OLAP : cube intervallaire

-- Donner les durées totales des stages de sept à douze mois en entreprise effectués d'une  
 -- façon générale ; par étudiant ; par société ; par tuteur ; par étudiant et par société ;  
 -- par étudiant et par tuteur ; par société et par tuteur ; et enfin par étudiant, par  
 -- société et par tuteur.

```
SELECT ide, ids, idp, SUM(duree) AS dureet
FROM Convention
GROUP BY CUBE(ide, ids, idp)
HAVING SUM(duree) BETWEEN 7 AND 12;
```



ide	ids	idp	dureet
15			11
17			10
	13	12	11
	34	53	12
	34		12
	13		11
		53	12
		12	11



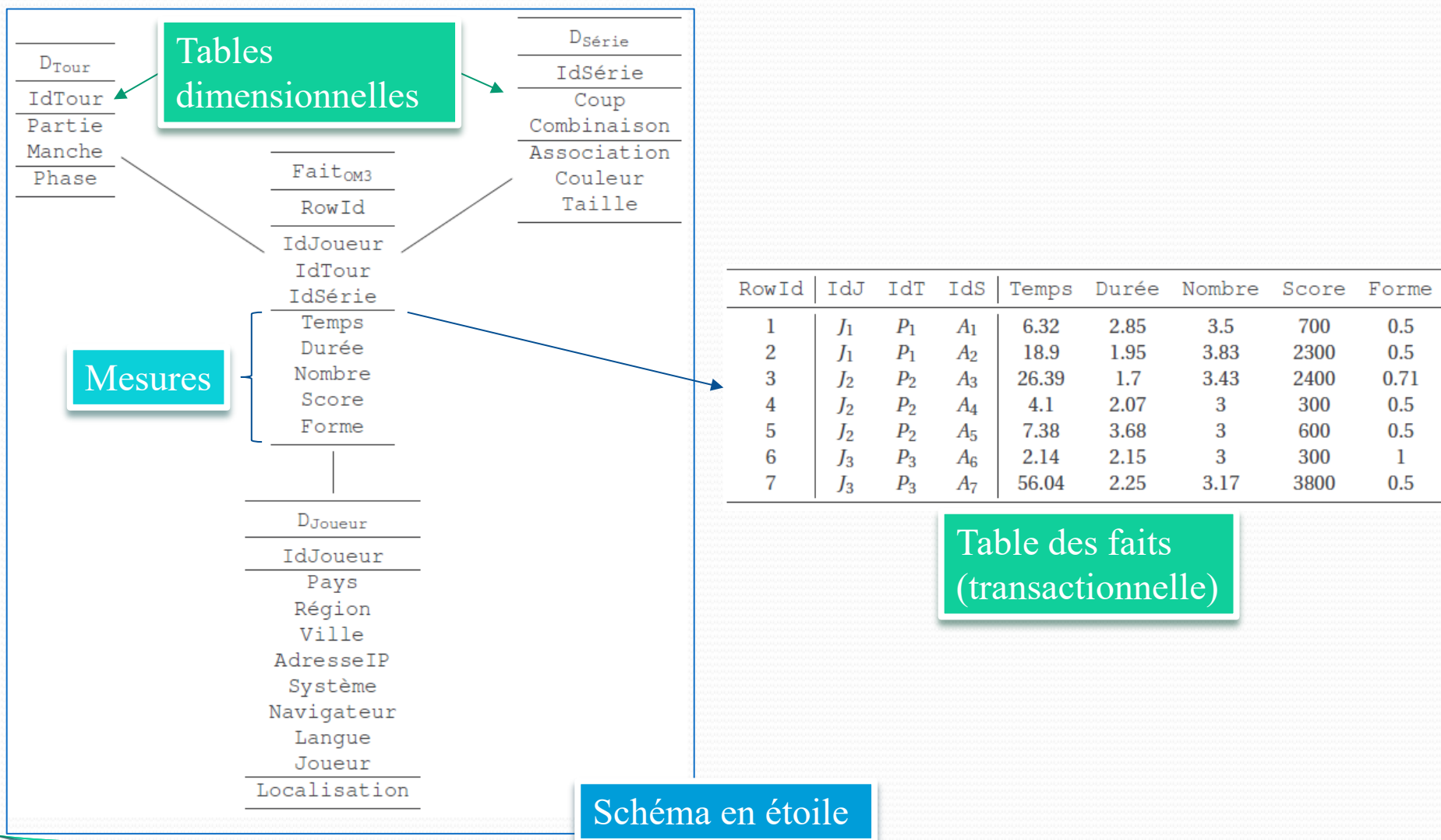
# Analyse multidimensionnelle

- Cube de données (hypercube OLAP) :
  - Union des résultats des requêtes agrégatives de partitionnement (avec GROUP BY) sur toutes les combinaisons possibles des attributs dimensions
  - Un (pré-)calcul permet des réponses rapides à toutes requêtes utiles mais crée aussi beaucoup d'opérations, et donc d'espace de stockage





# Entrepôt de données



# Table des faits récapitulative

Avec agrégation de données, ou mesure

RowId	Type	Propriété	Équipement	Quantité <sup>a</sup>
1	<i>Féticheur</i>	<i>Vitesse<sup>b</sup></i>	<i>Bastion<sup>c</sup></i>	400
2	<i>Féticheur</i>	<i>Vitesse</i>	<i>Marche<sup>d</sup></i>	100
3	<i>Sorcière</i>	<i>Chance<sup>e</sup></i>	<i>Marche</i>	100
4	<i>Sorcière</i>	<i>Vitesse</i>	<i>Bastion</i>	300
5	<i>Croisé</i>	<i>Chance</i>	<i>Bastion</i>	200

a. Nombre arrondi de personnages dans le classement pour l'Europe.

b. Vitesse d'attaque.

c. Le Bastion de la volonté.

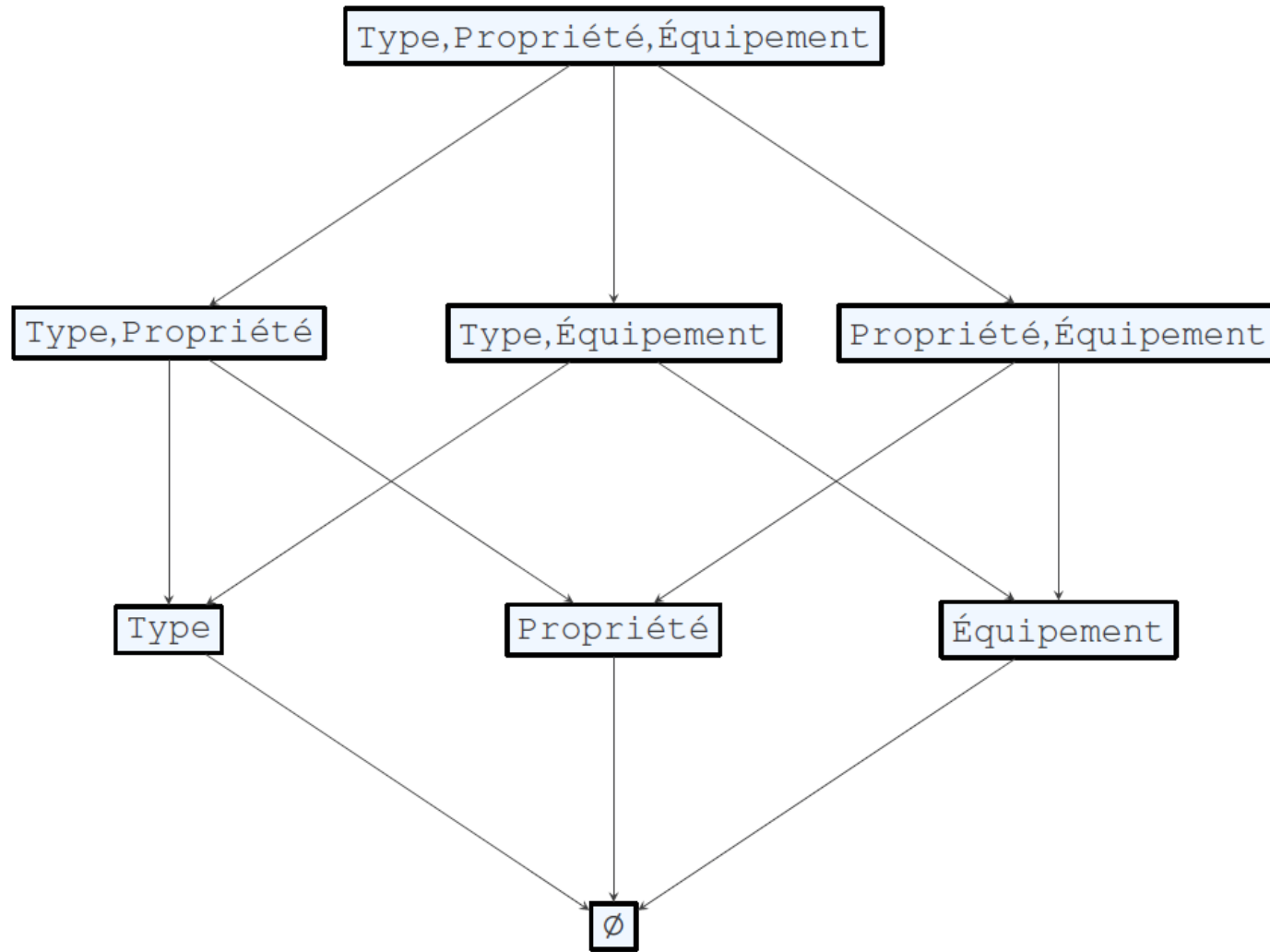
d. La Marche sans fin.

e. Chance de coup critique.

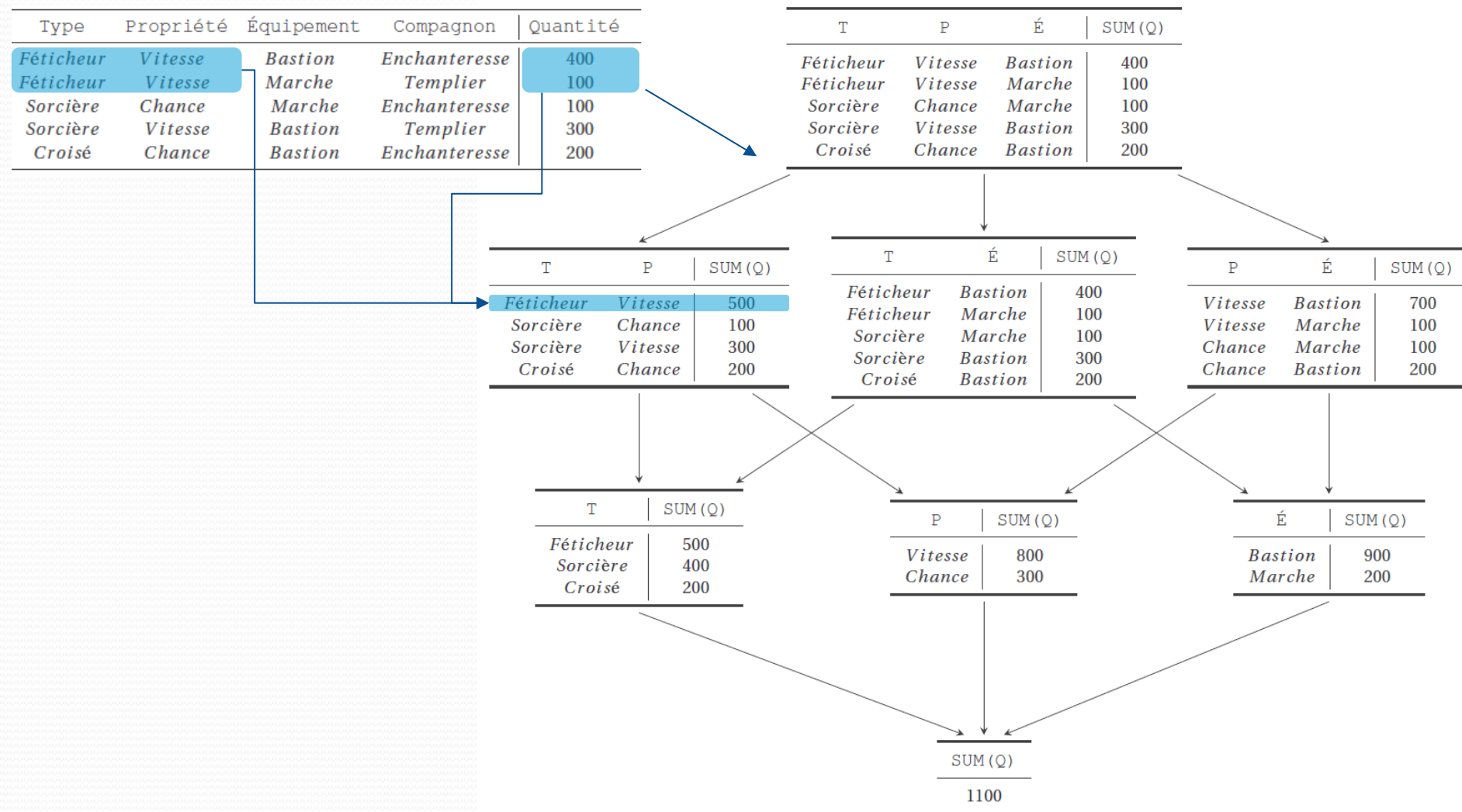
Jeu vidéo Diablo III : Reaper of Souls



# Exemple de cuboïdes



# Exemple de cuboïdes détaillés



# Crédits

## Auteur

Mickaël Martin Nevot

[mmartin.nevot@gmail.com](mailto:mmartin.nevot@gmail.com)

- Laurent Carmignac



Carte de visite électronique

## Relecteurs

Cours en ligne sur : [www.mickael-martin-nevot.com](http://www.mickael-martin-nevot.com)

