



# Bases de données

---

## SQL - Calculs et regroupements



# Sommaire

---

- 1. Attributs calculés (calculs horizontaux)
- 2. Fonctions d'agrégats (calculs verticaux)
- 3. Regroupement
- 4. Sélection des agrégats
- 5. Rollup, cube...



# 1. Attributs calculés (calculs horizontaux)

---

- **Informations qui peuvent être calculées**
  - pas stockées dans la BD
  - car sont redondantes
  - mais leur obtention nécessite
    - temps de calcul
    - requêtes qui peuvent être complexes
  - Pour simplifier l'accès
    - Vues et vues nommées
  - Pour réduire le temps de calcul
    - Vues nommées et triggers



# 1. Attributs calculés (calculs horizontaux)

---

- **Attribut calculé**

- *nouvelle* colonne obtenue par calcul
- calcul effectué
  - « sur chaque ligne »
  - à partir des valeurs des autres colonnes, sur la même ligne
- calcul placé dans la clause **SELECT**
  - la colonne calculée doit être affichée
- calcul placé dans : cond. de jointure, **WHERE**, **ORDER BY**
  - colonne pas affichée...
  - ... ajoutée à la table pour les manipulations internes

# 1. Attributs calculés (calculs horizontaux)

## ■ Exemple

- les titres des films et leur durée en secondes
- `SELECT Ftitre, Fduree * 60`  
`FROM film`
- Ajout d'une nouvelle colonne
  - nom de la colonne = formule de calcul
  - calcul effectué « sur chaque ligne » à partir de Fduree

Ftitre	Fduree * 60
Million dollar baby	7920
Eyes wide shut	9540
Le bon, la brute et le truand	10680
Les oiseaux	7200
Psychose	6540
Psycho	8640
Hero	5880
Hana-bi	6180
L'homme qui en savait trop	5040
L'homme qui en savait trop	7200
Les misérables	17400
Les misérables	13620
Les misérables	10800
S'en fout la mort	5460
Les sept samourais	12000
Les sept mercenaires	11160
La chambre verte	5700

# 1. Attributs calculés (calculs horizontaux)

- calcul :  $Fduree * 60$
- calcul effectué « sur chaque ligne » à partir de  $Fduree$

FilmID	Ftitre	FrealisateurID	Fgenre	Fnat	Fduree	Fannee	FRemakeDe
F001	Million dollar baby	A001	Drame	USA	132	2004	
F002	Eyes wide shut	A002	Drame	USA	159	1998	
F004	Le bon, la brute et le truand	A005	Western	ITALIE	178	1968	
F008	Les oiseaux	A003	Thriller	USA	120	1963	
F011	Psychose	A003	Thriller	USA	109	1960	
F012	Psycho	A023	Thriller	USA	144	1998	F011
F024	Hero	A021	Action	CHINE	98	2002	
F025	Hana-bi	A018	Drame	JAPON	103	1997	
F031	L'homme qui en savait trop	A003	Policier	UK	84	1934	
F027	L'homme qui en savait trop	A003	Policier	USA	120	1956	F031

$$132 * 60$$

$$159 * 60$$

$$178 * 60$$

Ftitre	Fduree * 60
Million dollar baby	7920
Eyes wide shut	9540
Le bon, la brute et le truand	10680
Les oiseaux	7200
Psychose	6540
Psycho	8640
Hero	5880
Hana-bi	6180
L'homme qui en savait trop	5040
L'homme qui en savait trop	7200



# 1. Opérateurs, fonctions

---

- **Calculs peuvent être réalisés**
  - sur des nombres
  - sur des chaînes de caractères
  - sur des dates
  - sur la valeur NULL
  - ...

# 1. Opérateurs, fonctions

## ■ Calculs sur les nombres

### ■ opérateurs arithmétiques

■ moins unaire :  $-$  (ex :  $-3$   $-Fduree$ )

■  $+$   $-$   $*$   $/$

■ calcul du reste (modulo) :  $\%$  (ex :  $13 \% 3$  vaut  $1$ )

### ■ parenthèses

### ■ fonctions sur les nombres

■ nom de la fonction en majuscule

■ parenthèses pour les arguments collées au nom de la fonction

■ ex :  $ABS(-3)$

■ certaines absentes de la norme SQL : ajoutées par les SGBD

➔ varient d'un SGBD à l'autre... données ici pour MySQL

13		3
1		4





# 1. Opérateurs, fonctions

## ■ fonctions sur les nombres

### ■ Valeurs relatives

- `ABS (x)`   `SIGN (x)`

### ■ Trigonométriques

- `SIN (x)`   `COS (x)`   `TAN (x)`   `COT (x)`   `ASIN (x)`   `ACOS (x)` ...

### ■ Arrondis

- `CEILING (x)`   `FLOOR (x)`   `ROUND (x)`

- `ROUND (x, n)`   `TRUNCATE (x, n)`

*n* = nombre de chiffres après la virgule

### ■ Puissances

- `SQRT (x)`   `POWER (x, y)`   `EXP (x)`   `LOG (x)`

### ■ Divers

- `RAND ( )`   `PI ( )`

- ...

*x* et *y* représentent une valeur constante, une colonne (= sa valeur sur la ligne) ou une expression calculée



# 1. Opérateurs, fonctions

---

- **Calculs sur les chaînes de caractère**
  - certaines dans la norme SQL d'autres ajoutées
  - longueur d'une chaîne
    - `CHARACTER_LENGTH(x)`
  - concaténation
    - `CONCAT(x, y, z, ...)`     ex : `CONCAT(Anom, ' ', Aprenom)`
  - majuscules, minuscules
    - `LOWER(x)`    `UPPER(x)`    (ou `LCASE(x)`    `UCASE(x)` )
  - extraction de sous-chaîne (à partir de prem, longueur nbcar)
    - `SUBSTRING(x, prem, nbcar)`
      - ex : `SUBSTRING(Aprenom, 1, 1)` donne l'initiale du prénom
  - ...



# 1. Opérateurs, fonctions

## ■ Exemples

### ■ fonctions numériques

- `ABS(-35)` vaut : 35
- `ROUND(3.521, 1)` vaut : 3.5
- `SQRT(25)` vaut : 5

### ■ fonctions (MySQL) sur les chaînes

- `CHARACTER_LENGTH('France')` vaut : 6
- `CONCAT('bon', 'jour')` vaut : 'bonjour'
- `UPPER('France')` vaut : 'FRANCE'
- `SUBSTRING('France', 2, 3)` vaut : 'ran'
- `SUBSTRING('France', 3)` vaut : 'ance'
- `SUBSTRING('France', -3, 2)` vaut : 'nc'
- `SUBSTRING('France', -3)` vaut : 'nce'



# 1. Opérateurs, fonctions

---

## ■ Représentation et types des dates

### ■ type `DATE`

- `'2017-09-11'` le 11 septembre 2017

### ■ type `TIME`

- `'13:45:00'` à 13 h 45

### ■ type `TIMESTAMP` ou `DATETIME`

- `'2017-09-11 13:45:00'` le 11 septembre 2017 à 13h45

### ■ avec fraction de secondes : on ajoute (*fsp*)

- *fsp* : précision de la fraction de secondes (1 à 6 digits)
- exemple pour un `TIME(3)` (3 : millièmes de secondes)
  - `'00:02:02.125'` 2 minutes, 2 secondes et 125 millièmes

# 1. Opérateurs, fonctions

Fonctions sur les dates  
spécifiques à MySQL

## ■ Calculs sur les dates

- certaines dans la norme SQL d'autres ajoutées
- obtention de la date courante (celle du système)
  - `CURRENT_DATE()` donne ex : '2017-09-11'
  - `CURRENT_TIME()` donne ex : '13:45:00'
  - `CURRENT_TIMESTAMP()` donne ex : '2017-09-11 13:45:00'
- extraction d'une partie de la date/time
  - `YEAR(d)`      `MONTH(d)`      `DAY(d)`
  - `HOUR(d)`      `MINUTE(d)`      `SECOND(d)`
- ajout ou soustraction à une date
  - `ADDDATE(d, nbj)`      `SUBDATE(d, nbj)`
- jour de la semaine
  - `DAYNAME(d)`
- ...

`d` est une date  
`nbg` un nombre de jours



# 1. Opérateurs, fonctions

## ■ Exemples

### ■ fonctions sur les dates

- `CURRENT_DATE()`      vaut : '2017-09-11'      (le 11/09/17 !)
- `YEAR('2017-09-11')`      vaut : 2017
- `YEAR(CURRENT_DATE())`      vaut : 2017      (le 11/09/17 !)
- `DAY('2017-09-11')`      vaut : 11
- `MINUTE('2017-09-11 13:45:00')`      vaut : 45
- `TIME('2017-09-11 13:45:00')`      vaut : '13:45:00'
  
- `ADDDATE('2017-09-11', 2)`      vaut : '2017-09-13'
- `ADDDATE('2017-12-31', 3)`      vaut : '2018-01-03'
- `DAYNAME('2017-09-11')`      vaut : 'Monday'

■ ...



# 1. Opérateurs, fonctions

---

## ■ Calculs sur la valeur NULL

- première valeur non nulle d'une liste

- `COALESCE(x, y, z, ...)` ou `IFNULL(x,y)` (MySQL)

- `x, y, z ...` peuvent être

- une valeur constante
    - un attribut (= sa valeur sur la ligne)
    - une expression calculée

- renvoie la première valeur non nulle

- ex : `COALESCE(Amort, 'Vivant')`

- affiche l'année de décès si définie, la chaîne 'Vivant' sinon



# 1. Attributs calculés (calculs horizontaux)

---

- **Nombreuses autres fonctions**

- certaines dans la norme SQL
  - sont-elles mise en œuvre dans « notre » SGBDR ?
- d'autres ajoutées par « notre » SGBDR
  - lesquelles ?
  - syntaxe ?

➔ Consulter un (des) manuel(s) de référence

- <http://dev.mysql.com/doc/refman/5.7/en/functions.html>
- <http://sqlpro.developpez.com/cours/sqlaz/fonctions/>
  - mais un très vieux pour MySQL...



# 1. Renommer une colonne

**AS** est facultatif, mais c'est tellement plus clair avec...

## ■ Renommage dans la clause SELECT

- `<attribut> AS <nom sans car spéciaux>`
- `<attribut> AS '<nom avec car spéciaux>'`

- attribut : existant ou calculé
- nouveau nom

- composé comme un nom de colonne → pas besoin d'apostrophes
- contient d'autres caractères → entre apostrophes

## ■ Exemple

- `SELECT Ftitre AS titre, Fduree*60 AS 'durée (sec)' ...`

titre	durée (sec)
Million dollar baby	7920
Eyes wide shut	9540
Le bon, la brute et le truand	10680



# Sommaire

---

- 1. Attributs calculés (calculs horizontaux)
- 2. Fonctions d'agrégats (calculs verticaux)
- 3. Regroupement
- 4. Sélection des agrégats
- 5. Rollup, cube...



## 2. Fonctions d'agrégats (calculs verticaux)

---

- **Fonction d'agrégat**

- fonction pour réaliser des calculs...
- ... sur les valeurs de toutes les lignes d'une colonne (ou plus)  
→ calcul réalisé « verticalement »

- **Calcul**

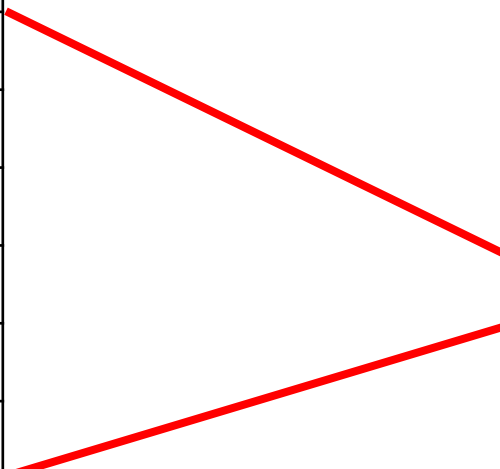
- à partir des **n lignes** d'une table
- donne un résultat en **1 ligne**, avec **1 colonne** par calcul

## 2. Fonctions d'agrégats (calculs verticaux)

### ■ Exemple

- `SELECT SUM(B) , AVG(B+C) , MAX(B)`  
`FROM tab`
- donne un résultat en 1 ligne, et 3 colonnes

tab	A	B	C
	a	1	2
	a	2	2
	b	1	1
	b	1	2
	b	3	2
	c	1	3



SUM(B)	AVG(B+C)	MAX(B)

## 2. Fonctions d'agrégats (calculs verticaux)

### ■ Exemple

- `SELECT SUM(B) , AVG(B+C) , MAX(B)`  
`FROM tab`
- donne un résultat en 1 ligne, et 3 colonnes

tab	A	B	C	B+C
	a	1	2	3
	a	2	2	4
	b	1	1	2
	b	1	2	3
	b	3	2	5
	c	1	3	4

SUM(B)	AVG(B+C)	MAX(B)
9		

## 2. Fonctions d'agrégats (calculs verticaux)

### ■ Exemple

- `SELECT SUM(B) , AVG(B+C) , MAX(B)`  
`FROM tab`
- donne un résultat en 1 ligne, et 3 colonnes

tab	A	B	C	B+C
	a	1	2	3
	a	2	2	4
	b	1	1	2
	b	1	2	3
	b	3	2	5
	c	1	3	4

SUM(B)	AVG(B+C)	MAX(B)
9	3.5	

## 2. Fonctions d'agrégats (calculs verticaux)

### ■ Exemple

- `SELECT SUM(B) , AVG(B+C) , MAX(B)`  
`FROM tab`
- donne un résultat en 1 ligne, et 3 colonnes

tab	A	B	C	B+C
	a	1	2	3
	a	2	2	4
	b	1	1	2
	b	1	2	3
	b	3	2	5
	c	1	3	4

SUM(B)	AVG(B+C)	MAX(B)
9	3.5	3



## 2. Fonctions d'agrégats (calculs verticaux)

---

- **Fonction d'agrégat + attributs de la table ?**
    - impossible (de façon directe)
    - fonction d'agrégat → 1 ligne
    - Attribut de la table → n lignes
- Impossible d'avoir une table de 1 et n lignes à la fois !



## 2. Fonctions d'agrégats (calculs verticaux)



- Lorsque **SELECT** contient une fonction d'agrégat...  
... il contient *seulement* des fonctions d'agrégat
  - lorsqu'il n'y a pas de regroupement (ou agrégat)  
= calcul vertical sur la table entière
  - sera « un peu » assoupli avec les regroupements (cf. partie 3)
- **Attention à MySQL**
  - ne l'interdit pas...
  - ... mais donne 1 valeur d'attribut au hasard !

## 2. Fonctions d'agrégats (calculs verticaux)

### ■ Exemple

Attention : erreur

■ `SELECT A, MAX(B)`  
`FROM tab`

■ Pour afficher A il faut 6 lignes !

tab	A	B	C		A	MAX(B)
	a	1	2		a	3
	a	2	2		a	
	b	1	1		b	
	b	1	2		b	
	b	3	2		b	
	c	1	3		c	

## 2. Fonctions d'agrégats (calculs verticaux)

### ■ Exemple

Attention : erreur

■ `SELECT A, MAX(B)`  
`FROM tab`

■ mais comme il y a un calcul vertical, le résultat n'a qu'une ligne

tab	A	B	C		A	MAX(B)
	a	1	2		a	3
	a	2	2		a	
	b	1	1		b	
	b	1	2		b	
	b	3	2		b	
	c	1	3		c	

## 2. Fonctions d'agrégats (calculs verticaux)

### ■ Exemple

Attention : erreur

■ `SELECT A, MAX(B)`  
`FROM tab`

- LA valeur pour A conservée est prise au hasard (souvent la 1<sup>re</sup>) n'est pas représentative de la colonne → n'a pas de sens n'est pas celle qui correspond au max dans B !

tab

A	B	C
a	1	2
a	2	2
b	1	1
b	1	2
b	3	2
c	1	3

A	MAX(B)
a	3
<del>a</del>	
<del>b</del>	
<del>b</del>	
<del>b</del>	
<del>c</del>	

Valeur qui n'a plus de sens



## 2. Fonctions d'agrégats

---

### ■ Fonctions à un argument

- |            |  |                         |
|------------|--|-------------------------|
| ■ Moyenne  | <code>AVG (x)</code>                             | nombres                 |
| ■ Somme    | <code>SUM (x)</code>                             | nombres                 |
| ■ Maximum  | <code>MAX (x)</code>                             | nombres, chaînes, dates |
| ■ Minimum  | <code>MIN (x)</code>                             | nombres, chaînes, dates |
| ■ Comptage | <code>COUNT (*)</code><br><code>COUNT (x)</code> | tous types              |

### ■ L'argument `x`

- soit 1 colonne existante
- soit 1 colonne calculée (expression = calcul horizontal)
- valeurs `NULL` pas prises en compte, sauf dans `COUNT (*)`



## 2. Fonctions d'agrégats

- Quelques opérateurs unaires ajoutés dans SQL/2003

- `STDDEV_POP (x)`      écart type d'une population
- `STDDEV_SAMP (x)`      écart type d'un échantillon
- `VAR_POP (x)`      variance d'une population
- `VAR_SAMP (x)`      variance d'un échantillon

- Quelques opérateurs binaires de SQL/2003

- `COVAR_POP (x, y)`      `COVAR_SAMP (x, y)`      covariances
- `CORR (x, y)`      corrélation
- `REGR_SLOPE (x, y)`      `REGR_INTERCEPT (x, y)`
- `REGR_COUNT (x, y)`      `REGR_AVGX (x, y)`
- `REGR_SXY (x, y)`
- ...      différents calculs de régression

## 2. Comptage

- **Cas général : COUNT (\*)**

- compte les lignes de la table (avec les doublons et les NULL)
- **SELECT** COUNT (\*)  
**FROM** <table>  
**WHERE** <condition de sélection>
- opération très rapide
  - la machine *connait* la taille des tables manipulées



- **Principe du comptage en SQL**

1. construire une table avec 1 élément à compter par ligne
  - jointures dans le FROM
  - sélection des lignes à conserver dans le WHERE
2. compter le nombre de lignes de cette table par **COUNT (\*)**



## 2. Comptage

---

### ■ Exemples

- Compter le nombre de films de la BD

```
SELECT COUNT(*)  
FROM film
```

- Compter le nombre de films français de 2h ou plus

```
SELECT COUNT(*)  
FROM film  
WHERE Fnat = 'France' AND Fduree >= 120
```

- Compter le nombre de rôles joués dans des films français

```
SELECT COUNT(*)  
FROM joue JOIN film ON JfilmID = FilmID  
WHERE Fnat = 'France'
```





## 2. Ne pas compter les NULL

---

- **Comptage des valeurs non nulles : COUNT (x)**
  - Se fait par rapport à une colonne x
    - repérer les NULL dans cette colonne pour ne pas les compter
- **Remplace un filtrage des valeurs NULL**
  - → simplification d'écriture
  - sauf cas particulier rare...
  - Opération plus coûteuse que COUNT(\*)
    - nécessite de tester la nullité sur chaque ligne

## 2. Ne pas compter les NULL

COUNT (Amort)

### ■ Exemple

- Compter le nombre d'artistes décédés  
= compter les valeurs non nulles dans Amort (date décès connue)

```
SELECT COUNT (Amort)
FROM artiste
```

- est équivalent à

```
SELECT COUNT (*)
FROM artiste
WHERE Amort IS NOT NULL
```

ArtisteID	Anom	Aprenom	Anat	Anaissance	Amort	Asexe
A197	BACHCHAN	Amitabh	INDE	1942		H
A200	BAKER	Roy Ward	UK	1916		H
A154	BATES	Kathy	USA	1948		F
A094	BAUR	Harry	FRANCE	1880	1943	1
A110	BAYE	Nathalie	FRANCE	1948		F
A225	BERGMAN	Ingrid	SUEDE	1915	1982	2
A032	BERNARD	Raymond	FRANCE	1891	1977	3
A196	BHADURI	Jaya	INDE	1948		F
A098	BLIER	Bernard	FRANCE	1916	1989	4
A129	BONNAIRE	Sandrine	FRANCE	1967		F
A096	BOUQUET	Michel	FRANCE	1925		F
A102	BRIALY	Jean-Claude	FRANCE	1933	2007	5
A105	BRYNNER	Yul	RUSSIE	1920	1985	6
A201	BURTON	Tim	USA	1958		H
A151	CAMERON	James	CANADA	1954		H
A077	CHEUNG	Maggie	CHINE	1964		F
A204	CLOSE	Glenn	USA	1947		F
A064	CRUISE	Tom	USA	1962		H
A091	DAY	Doris	USA	1924		F
A114	DAY-LEWIS	Daniel	UK	1957		H

etc

## 2. Ne pas compter les NULL

### ■ Erreur fréquente

- pour compter les titres de films

- `SELECT COUNT (Ftitre) ...`

- pour compter les rôles

- `SELECT COUNT (Jrole) ...`

- pour compter les membres

- `SELECT COUNT (Mnom) ...`

Mauvais raisonnement !

### ■ On compte en réalité...

- le nombre de lignes de la table construite...

- ... sans les NULL dans la colonne spécifiée

- avec la sémantique de la table construite...

- ... qui n'est pas forcément celle de la colonne !





## 2. Ne pas compter les NULL

---

- Exemple

- Que compte ?

```
SELECT COUNT(Fgenre)
FROM film JOIN note ON FilmID = NfilmID
           JOIN membre ON NmembreID = MembreID
WHERE Msexe = 'F' AND Fannee BETWEEN 1980 AND 1989
```



## 2. Ne pas compter les NULL

### ■ Exemple

- Que compte ?

```
SELECT COUNT(Fgenre)
FROM film JOIN note ON FilmID = NfilmID
        JOIN membre ON NmembreID = MembreID
WHERE Msexe = 'F' AND Fannee BETWEEN 1980 AND 1989
```

- Le nombre de notes mises par les femmes du club aux films des années 80 ! → rien à voir avec les genres...

- Il aurait fallu écrire (même résultat / moins de calculs)

```
SELECT COUNT(*)
FROM film JOIN note ON FilmID = NfilmID
        JOIN membre ON NmembreID = MembreID
WHERE Msexe = 'F' AND Fannee BETWEEN 1980 AND 1989
```



## 2. Ne pas compter les NULL

---

- **Moralité**

- S'il n'y a pas d'histoire de « NULL à ne pas compter »...

... utiliser `COUNT(*)` !

... sur une table qui contient 1 ligne par élément à compter



## 2. Calculs sur les valeurs distinctes

---

### ■ Calculs verticaux

- par défaut, utilisent toutes les valeurs de la colonne
- peu se faire sur les *valeurs distinctes*

### ■ Syntaxe

- `fonc(ALL x)` par défaut ; abrégé en `fonc(x)`
- `fonc(DISTINCT x)` sur les valeurs distinctes
- pas d'intérêt pour `MIN` et `MAX` (même résultat)
- utile avec `COUNT` (compter 1 seule fois chaque valeur)
- change la valeur de `SUM`, `AVG` (attention à la sémantique)



## 2. Calculs sur les valeurs distinctes

---

- **Exemple : compter les valeurs distinctes**
  - Combien de nationalités différentes pour les artistes ?
  - `SELECT COUNT(DISTINCT Anat)`  
`FROM artiste`
  - compte une seule fois chaque valeur de la colonne Anat
  - Revient à compter le nombre de lignes du résultat de
    - `SELECT DISTINCT Anat`  
`FROM artiste`
    - ... avec un `COUNT(*)` et une sous-requête (voir prochain cours)



## 2. Calculs sur les valeurs distinctes

COUNT (DISTINCT Anat)

### ■ Exemple : compter les valeurs distinctes

- Combien de nationalités différentes pour les artistes ?
- **SELECT** COUNT (DISTINCT Anat)  
**FROM** artiste

Attention, résultat *différent* de :  
**SELECT** DISTINCT COUNT (Anat)  
**FROM** artiste

ArtisteID	Anom	Aprenom	Anat	Anaissance	Amort	Asexe
A197	BACHCHAN	Amitabh	INDE	1942		H
A200	BAKER	Roy Ward	UK	1916		H
A154	BATES	Kathy	USA	1948		F
A094	BAUR	Harry	FRANCE	1880	1943	H
A110	BAYE	Nathalie	FRANCE	1948		F
A225	BERGMAN	Ingrid	SUEDE	1915	1982	F
A032	BERNARD	Raymond	FRANCE	1891	1977	H
A196	BHADURI	Jaya	INDE	1948		F
A098	BLIER	Bernard	FRANCE	1916	1989	H
A129	BONNAIRE	Sandrine	FRANCE	1967		F
A096	BOUQUET	Michel	FRANCE	1925		H
A102	BRIALY	Jean-Claude	FRANCE	1933	2007	H
A105	BRYNNER	Yul	RUSSIE	1920	1985	H
A201	BURTON	Tim	USA	1958		H
A151	CAMERON	James	CANADA	1954		H
A077	CHEUNG	Maggie	CHINE	1964		F
A204	CLOSE	Glenn	USA	1947		F
A064	CRUISE	Tom	USA	1962		H
A091	DAY	Doris	USA	1924		F
A114	DAY-LEWIS	Daniel	UK	1957		H

etc



# Questions ?

---



# Sommaire

---

- 1. Attributs calculés (calculs horizontaux)
- 2. Fonctions d'agrégats (calculs verticaux)
- 3. Regroupement
- 4. Sélection des agrégats
- 5. Rollup, cube...