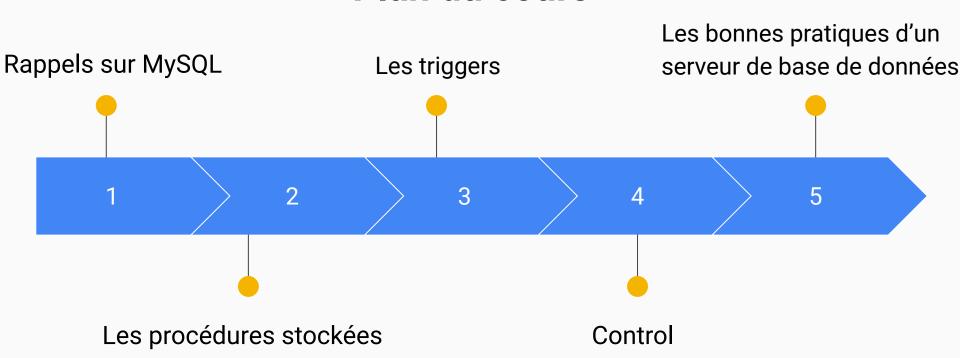
## MySQL perfectionnement

14 février 2022

## Vos attentes?

#### Plan du cours





#### Création d'une base de données

CREATE DATABASE [IF NOT EXISTS] nom\_bd;

**CREATE DATABASE** ma\_premiere\_db

**CHARACTER SET = 'utf8mb4'** 

**COLLATE** = 'utf8mb4\_general\_ci';

### Changer de base de données

**USE** nom\_bd;

### Suppression d'une base de données

DROP DATABASE [IF EXISTS] nom\_bd;

DROP DATABASE IF EXISTS ma\_premiere\_db;

CREATE DATABASE ma\_premiere\_db;

#### Afficher les bases de données

**SHOW DATABASES**;

```
MariaDB [(none)]> SHOW DATABASES;
  Database
  information_schema
  mysql
  performance_schema
  phpmyadmin
  test
5 rows in set (0.259 sec)
```

### Types de données

VARCHAR: chaine de caractére

**TEXT**: variable qui stock de trés longues chaines de caractére (64k caractéres)

**INT:** chiffre entier

**FLOAT:** chiffre decimal

**DATE: date du format AAAA-MM-JJ** 

TIME: temps au format HH:MM:SS

**DATETIME:** date et temps au format AAAA-MM-JJ HH:MM:SS

#### Création d'une table

```
CREATE TABLE nom_table(
          colonne1 type1,
          colonne2 type2,
PRIMARY KEY(colonne1) );
```

**CREATE TABLE Employe**(

num\_secu\_social int,

Nom varchar(255),

Prenom varchar(255),

Addresse varchar(255),

Ville varchar(255)

PRIMARY KEY(num\_secu\_social));

### Clé étrangère

**CREATE TABLE nom\_table(** 

colonne1 type1,

colonne2 type2,

PRIMARY KEY(colonne1),

FOREIGN KEY(colonne2) REFERENCES table2(nom\_cle\_primaire\_table2));

**CREATE TABLE Postes (** 

Code\_poste int NOT NULL,

Salaire float NOT NULL,

**Description VARCHAR(255)** 

**Employe int,** 

PRIMARY KEY (Code\_poste),

FOREIGN KEY (Employe) REFERENCES Employes(num\_secu\_social));

#### **ALTER TABLE Postes**

ADD FOREIGN KEY (Employe) REFERENCES Employes(num\_secu\_social);

#### Insertion de données dans une table

**INSERT INTO nom\_table** 

VALUES (valeur1, valeur2, valeur3, ...);

#### **INSERT INTO Employes**

VALUES (12345678912, Aznavour, Charles, Arcades des Champs-Élysées (75008), Paris);

# Les requêtes SQL SELECT colonne1,colonne4

- [FROM table\_references
- [WHERE where\_condition]]
- [ORDER BY {col\_name | expr | position} [ASC | DESC], ...]
- [LIMIT {[offset,] row\_count | row\_count OFFSET offset } ]);

**SELECT Description, Code\_poste** 

**FROM Postes** 

WHERE salaire > 2000.00

**LIMIT 10 OFFSET 5**;

### Fonctions fréquemment utilisées

```
les opérateurs (+,-,*,/,<,>,= ,=>,<=)
```

BETWEEN(): BETWEEN '2021-01-01' AND '2021-01-31';

IN(): SELECT val1 FROM tbl1 WHERE val1 IN (1,2,a');

LIKE(): SELECT 'This is a string' LIKE '%ing';

### Fonctions fréquemment utilisées

**DISTINCT(): considère les valeurs distinctes seulement** 

**COUNT(): SELECT COUNT(DISTINCT (colonne1)) FROM table1;** 

SUM(): SELECT SUM(valeur\_vente) FROM ventes;

MIN() et MAX(): SELECT MIN(note) FROM etudiants;

AVG():SELECT AVG(age) FROM habitant\_paris;

## GROUP BY SELECT

- [FROM table\_references
- [WHERE where\_condition]
- [GROUP BY {col\_name | expr | position} [ASC | DESC]]
- [HAVING where\_condition]
- [ORDER BY {col\_name | expr | position} [ASC | DESC], ...]
  - [LIMIT {[offset,] row\_count | row\_count OFFSET offset } ]

**SELECT COUNT**(id\_client), Pays

**FROM Clients** 

**GROUP BY Pays** 

**ORDER BY COUNT(id\_client) DESC;** 

**SELECT COUNT**(id\_client), Pays

**FROM Clients** 

**GROUP BY Pays** 

**HAVING COUNT**(id\_client) > 5

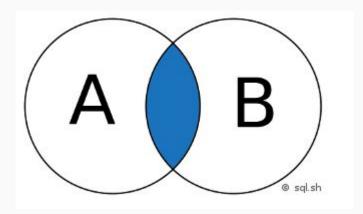
**ORDER BY COUNT(id\_client) DESC;** 

## Les jointures

https://sql.sh/cours/jointures

- jointure interne
- jointure externe complète
- jointure externe droite ou gauche

#### Jointures internes



**SELECT\*** 

FROM A

**INNER** JOIN B ON A.key = B.key

ou bien

**SELECT \*** 

FROM A, B

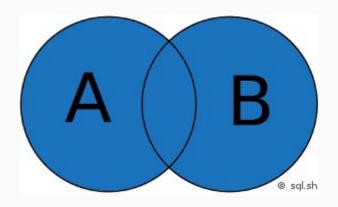
WHERE A.key = B.key

**SELECT** Commandes.num\_commande, Clients.Nom\_client

**FROM Commandes** 

**INNER JOIN Clients ON Commandes.id\_client= Clients.id\_client;** 

#### Jointures externe compléte

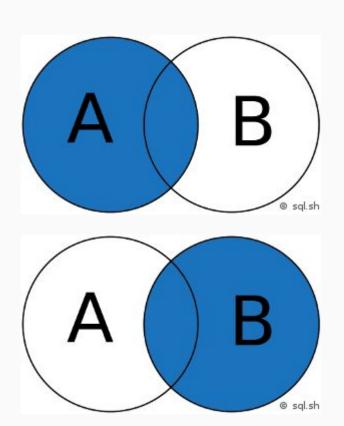


**SELECT\*** 

FROM A

**FULL** JOIN B ON A.key = B.key

#### jointure externe droite ou gauche



**SELECT\*** 

FROM A

**RIGHT** JOIN B ON A.key = B.key

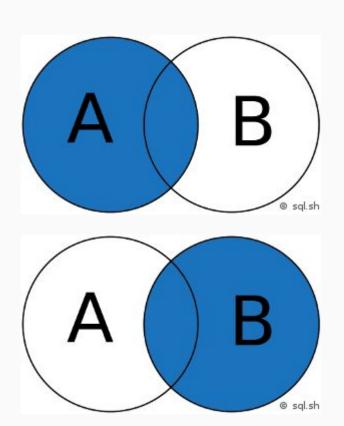
et

**SELECT\*** 

FROM A

**LEFT** JOIN B ON A.key = B.key

#### jointure externe droite ou gauche



**SELECT\*** 

FROM A

**RIGHT** JOIN B ON A.key = B.key

et

**SELECT\*** 

FROM A

**LEFT** JOIN B ON A.key = B.key

**SELECT** Commandes.num\_commande, Clients.Nom\_client

**FROM Commandes** 

**LEFT JOIN Clients** 

**ON** Commandes.id\_client= Clients.id\_client

**ORDER BY Clients.Nom\_client;** 

télécharger les fichiers de base de données et écrire les requêtes suivantes:

- 1. Obtenir la liste des 10 villes les plus peuplées en 2012
- 2. Obtenir la liste des 50 villes ayant la plus faible superficie
- 3. Obtenir la liste des départements d'outres-mer, c'est-à-dire ceux dont le numéro de département commencent par "97"
- 4. Obtenir le nom des 10 villes les plus peuplées en 2012, ainsi que le nom du département associé
- 5. Obtenir la liste du nom de chaque département, associé à son code et du nombre de commune au sein de ces département, en triant afin d'obtenir en priorité les départements qui possèdent le plus de communes
- 6. Obtenir la liste des 10 plus grands départements, en terme de superficie

- 7. Compter le nombre de villes dont le nom commence par "Saint"
- 8. Obtenir la liste des villes qui ont un nom existants plusieurs fois, et trier afin d'obtenir en premier celles dont le nom est le plus souvent utilisé par plusieurs communes
- 9. Obtenir en une seule requête SQL la liste des villes dont la superficie est supérieur à la superficie moyenne
- 10. Obtenir la liste des départements qui possèdent plus de 2 millions d'habitants
- 11. Remplacez les tirets par un espace vide, pour toutes les villes commençant par "SAINT-" (dans la colonne qui contient les noms en majuscule)

Créer les tables suivantes en utilisant la ligne de commandes:

Table Avion (NA : numéro avion de type entier (clé primaire),

Nom: nom avion de type texte (12),

Capacite : capacité avion de type entier,

Localite : ville de localité de l'avion de type texte (10))

Créer les tables suivantes en utilisant la ligne de commandes:

Table Pilote (NP : numéro pilote de type entier,
Nom : nom du pilote de type texte (25),
Adresse : adresse du pilote de type texte (40)

Créer les tables suivantes en utilisant la ligne de commandes:

Table Vol (NV: numéro de vol de type texte (6),

NP: numéro de pilote de type entier,

NA: numéro avion de type entier,

VD : ville de départ de type texte (10),

VA: ville d'arrivée de type texte (10),

HD: heure de départ de type entier,

HA : heure d'arrivée de type entier

)

Ecrire les requêtes suivantes:

1) Insérer les avions suivants dans la table Avion :

(100, AIRBUS, 300, RABAT), (101,B737,250,CASA), (102, B737,220,RABAT)

- 2) Afficher tous les avions
- 3) Afficher tous les avions par ordre croissant sur le nom
- 4) Afficher les noms et les capacités des avions
- 5) Afficher les localités des avions sans redondance

Ecrire les requêtes suivantes:

- 6) Afficher les avions dans la localité est Rabat ou Casa
- 7) Modifier la capacité de l'avion numéro 101, la nouvelle capacité est 220
- 8) Supprimer les avions dans la capacité est inférieure à 200
- 9) Afficher la capacité maximale, minimale, moyenne des avions
- 10) Afficher les données des avions dont la capacité est la plus basse

Ecrire les requêtes suivantes:

- 11) Afficher les données des avions dont la capacité est supérieure à la capacité moyenne
- 12) Afficher le nom et l'adresse des pilotes assurant les vols IT100 et IT104
- 13) Afficher les numéros des pilotes qui sont en service
- 14) Afficher les numéros des pilotes qui ne sont pas en service
- 15) Afficher les noms des pilotes qui conduisent un AIRBUS