

# Bases de données

## 1.1 Généralité sur les bases de données

### 1.1.1 Le langage SQL et les bases de données

La **base de données** (BDD) est un système qui enregistre des informations. Ces informations sont toujours classées. On peut donc faire des recherches sur ces données.

Les **SGBD** (**S**ystèmes de **G**estion de **B**ases de **D**onnées) sont les programmes qui se chargent du stockage de vos données.

Les plus connus sont :

- **MySQL** : libre et gratuit, c'est probablement le SGBD le plus connu.
- **PostgreSQL** : libre et gratuit comme MySQL, avec plus de fonctionnalités mais un peu moins connu ;
- **SQLite** : libre et gratuit, très léger et adapté pour des petites bases de donnée ;
- **Oracle** : utilisé par les très grosses entreprises ;

Vous donnez les ordres au SGBD en langage **SQL** (*Structured Query Language*).

Voici un exemple de commande en langage SQL, pour vous donner une idée :

```
SELECT id, auteur, message, datemsg FROM livreor ORDER BY datemsg DESC
```

Pour communiquer avec une base de donnée (créer, modifier, faire une requête) on le fait avec :

- **un logiciel** : sqldbrower pour sqlite, pgadmin pour postgree.
- **php** : phpmyadmin pour mysql ou phppgadmin pour postgree.
- **python** : (import sqlite3) , il y a aussi des modules pour mysql et postgree.

### 1.1.2 Structure d'une base de données

Avec les bases de données, il faut utiliser un vocabulaire précis.

Une table est représentée sous la forme d'un tableau ; par exemple, le tableau suivante vous montre à quoi peut ressembler le contenu d'une table appelée « eleve ».

id	nom	prénom	classe	date_naissance	LV1	LV2
1	AREZ	Michael	TS2	1995	anglais	espagnol
2	ASSONG	Yasmine	TS3	1995	anglais	espagnol

## 1.2 Avec sqlite3

Aller sur mon site <http://megamaths.free.fr/pdf/bases/>, télécharger la base de donnée **isn.sqlite3** et avec le gestionnaire de fichier :

clic droit → ouvrir un terminal ici → **sqlite3 isn.sqlite3**

```
SELECT * FROM sqlite_master;
```

On a ainsi le détail de la base de donnée.

## 1.3 Avec un logiciel *sqlitebrowser*

Ouvrez *sqlitebrowser* : Menu→Développement→SQLite database browser

Ouvrir alors la base *isn.sqlite3*, cliquer alors sur l'onglet *browse data* pour voir la base.

On peut bien sûr modifier les entrées, en ajouter et enlever.

## 1.4 Avec python

Ouvrez **ipython3** (ouvrir une console et taper `ipython3`)

```
import sqlite3 # import du module
con = sqlite3.connect('isn.sqlite3') # ouverture de la base
c = con.cursor() # obtention d'un curseur
c.execute("SELECT * FROM sqlite_master") # la requête
c.fetchall() # fetch signifie récupérer en anglais
# idem mais en une seule ligne
c.execute("SELECT * FROM sqlite_master").fetchall()
c.execute("SELECT * FROM eleves").fetchall() # pour voir toute la table
```

On utilise des bases de données pour effectuer des requêtes et avoir rapidement un résultat.

Par exemple pour afficher tous les élèves de TS3 né en 1995

```
c.execute("SELECT * FROM eleves WHERE classe='TS3'").fetchall()
c.execute("SELECT nom, prenom FROM eleves WHERE classe='TS3'").fetchall()
c.execute("SELECT * FROM eleves WHERE classe='TS3' ").fetchall()
c.execute("SELECT * FROM eleves WHERE classe='TS3' ORDER BY annee_naissance").
    fetchall()
c.execute("SELECT * FROM eleves WHERE classe='TS3' ORDER BY id").fetchall()
c.execute("SELECT * FROM eleves WHERE nom='SAJUDE' AND prenom='Damien' ").fetchall()
()
c.execute("SELECT * FROM eleves WHERE nom LIKE 's%' ").fetchall()
```

On peut bien sûr modifier une base de donnée

```
#pour ajouter
c.execute("INSERT INTO eleves (nom, prenom, classe,annee_naissance) VALUES ('
    Simpson', 'bart', 'TS3', 1995)")
con.commit() # pour valider l'écriture de la base
c.execute("SELECT * FROM eleves").fetchall()
c.execute("SELECT * FROM eleves WHERE prenom='bart'").fetchall()
# pour modifier
c.execute("UPDATE eleves SET prenom='lisa' WHERE nom='Simpson' AND prenom='bart'")
# effacer
c.execute("DELETE FROM eleves WHERE nom='Simpson'") # tous les simpson seront effacés
con.commit()
c.execute("SELECT * FROM eleves").fetchall()
```

## 1.5 Exercices

### Exercice 1 Les communes en France *communes.sqlite3*

Télécharger la base **communes.sqlite3**

Cette table porte sur les communes en france *ref : (IGN)*

id	code_comm	insee_com	nom_comm	statut	x_chf_lieu	y_chf_lieu	x_centroid	y_centroid
1	'271'	'29271'	'SAINT-VOUGAY'	'Commune simple'	1743	68564	1737	68559
z_moyen	superficie	population	code_cant	code_arr	code_dept	nom_dept	code_reg	nom_region
88	1483	0.9	'31'	'3'	'29'	'FINISTERE'	'53'	'BRETAGNE'

À l'aide de python et sqlite3 répondez aux questions suivantes :

1. Quelles sont les caractéristiques de la communes qui a pour id 586 ?  
.....
2. La liste des régions  
.....
3. Obtenez de même la liste des valeurs de statut.  
.....
4. Donner le nom des villes des sous prefectures de la région Picardie.  
.....
5. Combien y a t il de communes en france ?  
.....
6. Combien y a t il de Sous-préfecture dans ma base de donnée ?  
.....
7. Combien y a t il de communes dans le département de l'isère ?  
.....
8. Combien y a t il de communes dans le département de l'isère entre 10 000 habitants et 20 000 habitants ?  
.....
9. Les 3 communes de l'isère qui ont la plus grande superficie.  
.....
10. Citer les 10 communes les plus peuplés dans le département de l'isère.  
.....
11. Les 10 les moins peuplés.  
.....
12. Le nombre de communes en france dont le nom commence par "**Gren**"  
.....
13. Le nom des communes du département de l'isère dont le nom commence par "**Gre**"  
.....

## 1.6 Memento

### 1.6.1 Obtention des données

```
SELECT <liste des noms de colonnes> FROM <liste des noms de tables>
SELECT * FROM tab1                               /*(toutes les colonnes)*/
SELECT col1,col3 FROM tab1                        /*(une partie des colonnes)*/
SELECT DISTINCT col1 FROM tab1                   /*(élimine les doublons)*/
SELECT nom AS "nom personne" FROM tab1          /*(renommage des colonnes)*/
SELECT nom "nom personne" FROM tab1
```

### 1.6.2 Expression des restrictions

```
SELECT * FROM tab1 WHERE <condition(s)>
SELECT * FROM tab1 WHERE ville IN ('Brest ','Rennes ','Paris ')
SELECT * FROM tab1 WHERE age NOT BETWEEN 15 AND 20
SELECT * FROM tab1 WHERE travail IS NULL
SELECT * FROM tab1 WHERE travail IS NOT NULL
SELECT * FROM tab1 WHERE ville LIKE 'Gre%' /* commence par Gre */
SELECT * FROM tab1 WHERE ville LIKE 'reno%' /* contient reno */
```

```
> >= < <= = <>          /*(comparateur arithmétiques)*/
AND OR NOT                /*(comparateur logique)*/
%                          /*(n'importe quelle séquence de car.)*/
_ (barre basse)           /*(n'importe quel caractère)*/
```

### 1.6.3 Tri et présentation des résultats

```
SELECT * FROM tab1 ORDER BY col1                 /*(tri ascendant par défaut)*/
SELECT * FROM tab1 ORDER BY col5,col7           /*(tri par col5 puis tri par col7)*/
SELECT * FROM tab1 ORDER BY age ASC, sexe DESC /*(tri ascendant ou descendant)*/
SELECT * FROM tab1 ORDER BY age LIMIT 0,10      /* Les 10 premiers résultats*/
SELECT * FROM tab1 ORDER BY age LIMIT 10,5     /* Les 5 suivants*/
```

### 1.6.4 Expression des jointures

```
SELECT * FROM tab1,tab2 /*(jointure sans qualification =produit cartésien)*/
SELECT * FROM tab1,tab2 WHERE tab1.col1=tab2.col2 /*(jointure avec égalité =
équijointure)*/
SELECT * FROM tab1 t1,tab2 t2,tab3 t3 WHERE t1.col1=t2.col2 AND t2.col2=t3.col3
/*(jointures en cascades)*/
```

### 1.6.5 Expression des manipulations de données

```
SELECT designation,prix*1.196 AS "P. TTC" FROM tab1
YEAR, MONTH, DATE /*(date)*/
SUBSTRING, UPPER, LOWER, CHARACTER_LENGTH /* (manipulation de chaînes de car.)*/
```

## 1.6.6 Regroupements

```
SELECT * FROM tab1 GROUP BY col1
```

## 1.6.7 Les fonctions statistiques

```
AVG                /*(moyenne)*/  
COUNT            /*(nombre d'éléments)*/  
MAX               /*(maximum)*/  
MIN              /* (minimum)*/  
SUM              /*(somme)*/  
SELECT COUNT(*)FROM tab1  
SELECT SUM(col1) FROM tab2
```

## 1.6.8 Sous-requêtes SQL

```
SELECT * FROM tab1 WHERE prix > (SELECT MIN(prix) FROM tab2)  
SELECT * FROM tab1 WHERE nom NOT IN (SELECT nom FROM tab2)  
SELECT * FROM tab1 WHERE prix > ALL (SELECT prix FROM tab2)  
SELECT * FROM tab1 WHERE (sup. à ttes les valeurs)  
SELECT * FROM tab1 WHERE prix > ANY (SELECT prix FROM tab2)  
SELECT * FROM tab1 WHERE (sup. à au moins 1)
```

## 1.6.9 Opérateurs ensemblistes

```
UNION                /*S'intercalent entre deux sélections*/  
INTERSECT  
EXCEPT
```

## 1.6.10 Insertions

```
INSERT INTO tab1 VALUES ('abc',5,7)  /*(toutes les valeurs doivent être renseignées)*/  
INSERT INTO tab1(col1,col7) VALUES ('abc',7)  /*(on ne renseigne que les colonnes indiquées, les cols non précisées sont mises à NULL)*/  
INSERT INTO tab1 SELECT * FROM tab2
```

## 1.6.11 Mises à jour

```
UPDATE tab1 SET col7='abc' WHERE col1=1
```

## 1.6.12 Suppressions

```
DELETE FROM tab1 WHERE col1=1  
DELETE FROM tab1 WHERE col1 IN (SELECT no FROM tab2)  
DELETE FROM tab1                /*(vide complètement la table)*/
```

Ref : [http://public.enst-bretagne.fr/~ptanguy/mementos/memento\\_SQL.pdf](http://public.enst-bretagne.fr/~ptanguy/mementos/memento_SQL.pdf)

## 1.7 La base de donnée communes.sqlite3

- **Une clé primaire : id**

*Type : INTEGER NOT NULL PRIMARY KEY*

- **Code géographique de la commune : CODE\_COMM**

*Type : chaîne de caractères.*

Il s'agit du code géographique permettant d'identifier la commune dans son département d'appartenance (texte de trois caractères).

- **Numéro INSEE de la commune : INSEE\_COM**

*Type : chaîne de caractères.*

Il s'agit d'un numéro de 5 caractères : le code du département suivi du code géographique de la commune. Pour les DOM, le 3ème caractère est commun au n° de département et au n° de commune.

- **Nom : NOM\_COMM**

*Type : chaîne de caractères.*

Nom de la commune (source INSEE). C'est un texte en majuscules non accentuées d'au plus 50 caractères.

- **Statut administratif : STATUT**

*Type : énuméré.*

Valeurs possibles :

- Capitale d'État
- Préfecture de région
- Préfecture de département
- Sous préfecture
- Chef-lieu de canton
- Commune simple

- **Abscisse du chef-lieu : X\_CHF\_LIEU**

*Type : entier*

Abscisse du chef-lieu de la commune (en hectomètres).

- **Ordonnée du chef-lieu : Y\_CHF\_LIEU**

*Type : entier*

Ordonnée du chef-lieu de la commune (en hectomètres).

- **Abscisse du centroïde : X\_CENTROID**

*Type : entier*

Abscisse du centroïde de la commune (en hectomètres).

- **Ordonnée du centroïde : Y\_CENTROID**

*Type : entier*

Ordonnée du centroïde de la commune (en hectomètres).

- **Altitude moyenne : Z\_MOYEN**

*Type : entier*

Altitude moyenne de la commune (en mètres).

- **Superficie : SUPERFICIE**

*Type : entier*

Superficie de la commune en hectares. C'est la somme des surfaces des faces BD CARTO-composant la commune (avant allégement géométrique et suppression des îles et enclaves).

- **Population : POPULATION**

*Type : décimal*

Chiffre de population sans doubles comptes au dernier recensement, en milliers d'habitants, à une décimale. Pour Mayotte, ce chiffre provient du recensement de 1997.

- **Code géographique du canton : CODE\_CANT**

*Type : chaîne de caractères*

Code géographique du canton auquel appartient la commune (texte de deux caractères).

- **Code géographique de l'arrondissement : CODE\_ARR**

*Type : chaîne de caractères*

Code géographique de l'arrondissement auquel appartient la commune (texte d'un caractère). Ce champ est vide pour les communes de Mayotte : il n'y a pas d'arrondissement dans cette collectivité départementale

- **Code géographique du département : CODE\_DEPT**

*Type : chaîne de caractères*

Code géographique du département auquel appartient la commune (texte de deux caractères).

- **Nom du département : NOM\_DEPT**

*Type : chaîne de caractères*

Nom du département auquel appartient la commune. C'est un texte en majuscules non accentuées d'au plus 30 caractères.

- **Code géographique de la région : CODE\_REG**

*Type : chaîne de caractères*

Code géographique de la région à laquelle appartient la commune (texte de deux caractères).

- **Nom de la région : NOM\_REGION**

*Type : chaîne de caractères*

Nom de la région à laquelle appartient la commune. C'est un texte en majuscules non accentuées d'au plus 30 caractères.