

Création de BDD

Si la base n'existe pas

SQL

```
CREATE DATABASE ma_base;
```

Sinon si on veut la recréer

SQL

```
DROP DATABASE ma_base;  
CREATE DATABASE ma_base;
```

Création de tables

Cas de base.

SQL

```
DROP TABLE IF EXISTS MaTable;  
CREATE TABLE MaTable(  
    attribut_1 INT PRIMARY KEY  
    attribut_2 TEXT,  
    attribut_3 VARCHAR(50) -- TEXT limité à 50 caractères  
);
```

Création avec id clé primaire automatique (voir comment insérer des tuples plus loin)*

SQL

```
DROP TABLE IF EXISTS MaTable;  
CREATE TABLE MaTable(  
    attribut_1 INT PRIMARY KEY AUTOINCREMENT,  
    attribut_2 TEXT,  
    attribut_3 VARCHAR(50)  
);
```

Création avec clé étrangère (référence à une clé primaire d'une autre table).

SQL

```
DROP TABLE IF EXISTS MaTable;  
CREATE TABLE MaTable(  
    attribut_1 INT PRIMARY KEY,  
    attribut_2 TEXT FOREIGN KEY REFERENCES autre_table (attribut_de_reference),  
    attribut_3 VARCHAR(50)  
);
```

Insérer des tuples

Dans une table, avec toutes les colonnes renseignées dans le bon ordre

SQL

```
INSERT INTO MaTable VALUES
(val_1_attribut_1, val_1_attribut_2, val_1_attribut_3),
(val_2_attribut_1, val_2_attribut_2, val_2_attribut_3);
```

Dans une table, avec par exemple la clé primaire créée automatiquement*

SQL

```
INSERT INTO MaTable(attribute_2, attribute_3) VALUES
(val_1_attribut_2, val_1_attribut_3),
(val_2_attribut_2, val_2_attribut_3);
```

Requêtes simples

Afficher toute une table

SQL

```
SELECT *
FROM MaTable;
```

Afficher certaines colonnes d'une table

SQL

```
SELECT attribute_2, attribute_3
FROM MaTable;
```

Afficher certaines colonnes d'une table en renommant les colonnes

SQL

```
SELECT attribute_2 AS nouveau_nom, attribute_3
FROM MaTable;
```

Afficher une table avec condition Afficher certaines colonnes d'une table

SQL

```
SELECT *
FROM MaTable
WHERE <condition(s)>;
```

Conditions :

- var_num BETWEEN val1 AND val2
- var_txt LIKE "%motif%" avec % avant et/ou après le motif
- utilisation de <, >, <=, >=, =, AND, OR, NOT, IS NULL, IS NOT NULL

Trier la table produite selon une colonne

SQL

```
SELECT *  
FROM MaTable  
ORDER BY attribut_1 ASC; -- ou DESC
```

Limiter l'affichage de la table produite à 10 lignes

SQL

```
SELECT *  
FROM MaTable  
LIMIT 10;
```

SQL

```
SELECT *  
FROM MaTable  
ORDER BY attribut_1 ASC; -- ou DESC
```

Fonctions d'agrégation

Compter les lignes d'une table suivant une condition

SQL

```
SELECT COUNT(*)  
FROM MaTable  
WHERE <condition(s)>;
```

Faire la somme sur une colonne

SQL

```
SELECT SUM(attribut_1)  
FROM MaTable  
WHERE <condition(s)>;
```

Existent aussi les fonctions **MIN**, **MAX**, **AVG** (min, max et moyenne).

Mise à jour et suppressions

Supprimer des valeurs d'une table suivant une condition

SQL

```
DELETE FROM MaTable  
WHERE <conditions>;
```

Mettre à jour les valeurs d'une table suivant une condition

SQL

```
UPDATE MaTable  
SET attribut_1 = valeur_1
```

```
WHERE condition;
```

Jointures

Sélectionner dans plusieurs tables liées par une référence

SQL

```
SELECT Table_1.attribut_1, Table_2.attribut_3
FROM Table_1
JOIN Table_2 ON Table_1.attribut_1 = Table_2.attribut_de_reference

-- attribut_de_reference est une clé étrangère de Table_2 qui fait référence à la
↔ clé primaire de Table_1
```

ON DELETE CASCADE et **ON UPDATE CASCADE** ne sont pas indispensables, au contraire : elles « affaiblissent » les contraintes logiques de référence.