

# TP : Initiation au SQL

*Instruction générale* : hormis pour les exercices corrigés collectivement, vous ferez valider votre travail par l'enseignant.

Ce TP se poursuit sur DB Browser et a pour but de s'initier au SQL.

## 1 Découverte du SQL

Le SQL est un langage créé en 1974 et normalisé en 1986 ; il est l'acronyme de Structured Query Language. Il s'agit d'un langage servant à exploiter les bases de données relationnelles. Il permet notamment de :

- ajouter, supprimer, rechercher, modifier des données dans une base ;
- créer et modifier l'organisation des données dans la base ;
- débiter et terminer des transactions (suite d'opérations transformant la base de données d'un état à un autre comme un achat, une réservation, etc) ;
- permettre ou d'interdire l'accès à certaines données à certains utilisateurs.

## 2 Commandes pour la gestion des bases de données et des tables

Le tableau ci-dessous recense quelques commandes de bases sur la gestion des bases de données et des tables. Il existe évidemment de nombreuses autres commandes, on pourra notamment les retrouver sur le site du W3 Schools.

Création d'une base	CREATE DATABASE base ;
Suppression d'une base	DROP DATABASE base ;
Création d'une table	CREATE TABLE table (attributs types conditions) ;
Ajout de données dans la table	INSERT INTO table (attribut) VALUES (données);
Suppression d'une table	DROP TABLE table ;
Ajout d'un attribut	ALTER TABLE table ADD attribut type ;
Suppression d'un attribut	ALTER TABLE table DROP COLUMN attribut ;
Modification d'un attribut	ALTER TABLE table ALTER COLUMN attribut type ;
Ajout d'une clé étrangère	ALTER TABLE table ADD FOREIGN KEY (attribut) REFERENCES tableEtrangere(attribut) ;

**Exemple :** le code suivant permet de créer une table morceaux :

```
CREATE TABLE "morceaux" (  
    "id_morceau" INTEGER NOT NULL DEFAULT 1 UNIQUE,  
    "nom_morceau" TEXT NOT NULL,  
    "id_artiste" INTEGER,  
    "annee_morceau" YEAR,  
    FOREIGN KEY("id_artiste") REFERENCES "artistes"("id_artiste"),  
    PRIMARY KEY("id_morceau" AUTOINCREMENT)  
);
```

Il faut avoir au préalable créé la table `artistes` avec l'attribut `id_artiste` puisqu'on y fait référence. Cela pourrait être fait avec le code :

```
CREATE TABLE "artistes" (  
    "id_artiste" INTEGER NOT NULL DEFAULT 1 UNIQUE,  
    "nom_artiste" TEXT NOT NULL UNIQUE,  
    PRIMARY KEY("id_artiste" AUTOINCREMENT)  
);
```

**Remarque :** on pourra observer le type de l'attribut `annee_morceau` : `YEAR`. Il existe de nombreux autres types ayant des spécificités d'utilisation comme `DATE` par exemple. On peut les retrouver ici.

**Exercice 1. [Pokémons, réinitialisation]** Le but de cet exercice est de reconstruire notre base de données `pokémons` à partir de zéro en utilisant cette fois-ci le langage SQL.

1. Ouvrir la base de données `pokémons` dans DB Browser et aller dans l'onglet « Exécuter le SQL ». À présent, un maximum de questions et d'exercices devront être traités sans recourir à l'interface de DB Browser, donc le plus possible en exécutant les scripts SQL que l'on y rentrera.
2. Supprimer toutes les tables contenues dans la base.
3. Recréer les tables `types`, `pokemons` et `attaques` avec leurs attributs et contraintes correspondant au schéma du TP précédent en utilisant le SQL.

### 3 Commandes pour ajouter, supprimer, mettre à jour des données

Le tableau ci-dessous recense quelques commandes de bases sur l'ajout, la suppression et la mise à jour de données dans une table. Il existe évidemment de nombreuses autres commandes, on pourra notamment les retrouver sur le site du W3 Schools.

Ajout de données	INSERT INTO table VALUES (valeur_1,valeur_2,...) ;
Mise à jour de données	UPDATE table SET attribut_1=valeur_1,... WHERE condition ;
Suppression de données	DELETE FROM table WHERE condition ;

#### Remarques :

- Il est possible d'ajouter des données seulement pour certains attributs en les spécifiant à l'aide de la syntaxe suivante : `INSERT INTO table (attribut_1,attribut_2,...) VALUES (valeur_1,valeur_2,...)`.
- Pour supprimer tous les enregistrements sans supprimer la table, il suffit d'utiliser `DELETE FROM table` sans préciser de conditions.

**Exemple :** le script suivant permet d'insérer les données de plusieurs artistes simultanément dans la table `artistes` en séparant les différents enregistrements à l'aide de virgules. On notera aussi qu'il n'y a pas besoin de s'occuper de l'attribut `id_artiste` qui est généré automatiquement grâce aux contraintes `DEFAULT 1` et `AUTO INCREMENT`.

```
INSERT INTO artistes(nom_artiste)
VALUES ("Pink Floyd"),
("Led Zeppelin"),
("Greta Van Fleet"),
("The Doors"),
("The Who"),
("Queen"),
("Scorpions");
```

Une fois les artistes ajoutés, on peut intégrer des morceaux à la table `morceaux` de la même façon.

```
INSERT INTO morceaux(nom_morceau,id_artiste,annee_morceau)
VALUES ("Money",1,1973),
("Stairway to Heaven",2,1971),
("Highway Tune",3,2017),
("The End",4,1967),
("Behind Blue Eyes",5,1971),
("Bohemian Rhapsody",6,1975),
("Still Loving You",7,1984);
```

**Exercice 2. [Alimentation de la base pokédex 1]**

1. Ajouter deux enregistrements dans la table `types` : `Plante` et `Poison`.
2. Ajouter un enregistrement dans la table `pokemons` correspondant à `Bulbizarre`. Ajouter quelques-uns des pokémons suivants d'un seul coup.
3. Que pensez-vous de l'idée d'entrer à la main toutes les données des trois tables ?

**Exercice 3. [Alimentation de la base pokédex 2]**

1. Télécharger les fichiers CSV présents dans ce dossier.
2. Ouvrir les fichiers `moves.csv`, `pokemon.csv`, `types.csv` puis identifier leurs attributs. Quelles seraient les clés primaires, étrangères et les contraintes ? Dessiner le schéma de cette base.
3. Supprimer encore une fois les tables de la base de données.
4. Importer les fichiers CSV directement dans la base gr « Fichier » puis en faisant « Importer » et enfin « Table depuis un fichier CSV ».
5. Modifier les tables afin de tenir compte des contraintes du schéma.

## 4 Requêtes d'affichage

### 4.1 Syntaxe générale

Une requête d'affichage (ou sélection) s'effectue grâce à la syntaxe suivante :

```
SELECT attributs
FROM table
WHERE conditions
+ options (tri, nombre d'éléments, etc)
```

**Remarques :**

- Si on souhaite sélectionner tous les attributs, on écrit `SELECT *`.
- Il est possible de ne pas mettre de `WHERE` et de conditions, on obtient alors une restriction de la table aux attributs sélectionnés.

**Exemple :** Le script suivant permet d'obtenir le nom et le prénom à partir de son identifiant.

```
SELECT nom_morceau, annee_morceau
FROM morceaux
WHERE id_morceau=2
```

On obtient alors réponse « `Stairway to Heaven 1971` » sous la forme d'un tableau.

**Exercice 4.**

1. Déterminer à l'aide d'une requête le nom du pokémon numéro 563.
2. Déterminer la liste des pokémons de type feu.

## 4.2 Conditions

- Il est possible de cumuler plusieurs conditions avec des opérations booléennes comme AND, OR, NOT et IN (liste).
- On peut chercher une chaîne de caractère dans un texte à l'aide de la commande LIKE et du symbole % remplaçant les caractères de la chaîne.

### Exemples :

```
SELECT nom_morceau
FROM morceaux
WHERE id_artiste IN (2,3)
```

renvoie tous les morceaux de la table de Led Zeppelin et Greta Van Fleet : « Stairway to Heaven » et « Highway Tune ».

```
SELECT nom_morceau
FROM morceaux
WHERE id_artiste NOT IN (2,3)
```

renvoie tous les morceaux de la table à l'exception de ceux de Led Zeppelin et Greta Van Fleet.

```
SELECT nom_morceau
FROM morceaux
WHERE nom_morceau LIKE "%way%"
```

renvoie tous les morceaux contenant la chaîne de caractères « way », on obtient « Stairway to Heaven » et « Highway Tune ». Ici, on considère les chaînes de caractères en ayant avant et après « way » tandis que dans l'exemple ci-dessous, on ne considère que les chaînes commençant par « B » car sans % devant le caractère.

```
SELECT nom_morceau
FROM morceaux
WHERE nom_morceau LIKE "B%"
```

Dans ce dernier cas, on obtient tous les morceaux dont le nom commence par B : « Behind Blue Eyes » et « Bohemian Rhapsody ».

### Exercice 5.

1. Rechercher tous les pokémons de type roche ayant une vitesse supérieure à 30.
2. Rechercher tous les pokémons ayant une défense entre 40 et 60.
3. Rechercher tous les pokémons de types ténèbre ou fantôme.
4. Rechercher tous les pokémons de types eau, plante et feu.
5. Rechercher tous les pokémons qui ne sont pas des trois types ci-dessus.
6. Rechercher tous les pokémons dont le nom se finit par « ar ».
7. Rechercher tous les pokémons dont le nom contient un « y ».

### 4.3 Options

- La commande `ORDER BY attribut` permet d'afficher les réponses de la requête par ordre croissant selon l'attribut choisi.
- Pour obtenir un tri décroissant, il suffit d'ajouter `DESC` à la commande comme suit : `ORDER BY attribut DESC`.
- On peut limiter le nombre de résultats affichés grâce à la commande `LIMIT nombre`.
- On peut compter les résultats d'une recherche à l'aide de la commande `SELECT COUNT`, le reste de la syntaxe étant identique à ce qui a été vu plus haut.
- On peut éliminer les résultats redondants d'une recherche à l'aide de la commande `SELECT DISTINCT`, seuls les résultats différents les uns des autres seront conservés.

#### Exemples :

```
SELECT *
FROM morceaux
ORDER BY annee_morceau
```

permet d'obtenir tous les attributs de la table `morceaux` par ordre croissant des années.

```
SELECT *
FROM morceaux
ORDER BY annee_morceau DESC
LIMIT 5
```

permet d'obtenir tous les attributs de la table `morceaux` par ordre décroissant des années en se limitant aux cinq premiers résultats. On a donc les cinq morceaux les plus récents de la table.

#### Exercice 6.

1. Afficher les pokémons de types électriques par ordre croissant en fonction de leur attaque.
2. Afficher les pokémons légendaires par ordre décroissant de valeur d'attaque spéciale.
3. Faire les palmarès des dix pokémons ayant les plus hautes valeurs de points de vie, d'attaque, de défense, de vitesse, d'attaque et de défense spéciale.
4. Combien de pokémons électriques existe-t-il ?
5. Quels sont les types des pokémons entre les numéros 349 à 403 ?

## 5 Ressources supplémentaires

- Vidéo introductive sur le SQL
- Vidéo sur les requêtes d'affichage
- Cours du W3 Schools sur le SQL
- Cours du site `sql.sh`
- Mémo du site `sql.sh`
- Plus de fichiers CSV sur les pokémons sont disponibles sur ce Github.