

# Découverte des bases de données

**Thèmes :** prise en main d'un logiciel de gestion de bases de données, premières requêtes SQL.

Les bases de données sont le plus souvent installées sur un serveur distant. Pour nos TP une base de données sera représentée sous forme de fichier SQLite. Nous utiliserons le logiciel *DB Browser for SQLite* pour lire, créer et modifier une base de données. Le logiciel permet également l'exécution de requêtes SQL.

## 1 Interrogation d'une base de données existante

À l'aide du bouton *Ouvrir une base données*, ouvrir la base de données `pizza.sqlite`. Cette base de données correspond à l'exemple présenté dans le document de cours. L'onglet *Structure de la Base de Données* permet de voir la liste des tables. L'onglet *Parcourir les données* permet de voir le contenu des tables.

Pour toutes les questions suivantes, proposer une requête SQL. On testera les résultats de la requête dans l'onglet *Exécuter le SQL*. L'onglet se divise en deux parties : une pour saisir les requêtes SQL, l'autre pour observer les résultats sous forme de tables. On utilisera le bouton *Exécuter la ligne courante* pour lancer l'évaluation d'une requête.

- a) Obtenir la liste des prénoms des clients.
- b) Obtenir la liste des pizzas qui coûtent moins de 10 euros.
- c) Obtenir la liste des pizzas dont le prix est entre 10 euros et 12 euros inclus.
- d) Obtenir le prix de la pizza végétarienne.
- e) Obtenir la liste de tous les noms de famille (client ou livreur).
- f) Obtenir la liste des scooters qui ont parcouru plus de 10 000 km.
- g) Obtenir la liste des commandes qui n'ont pas été livrées (heure de livraison NULL).
- h) Obtenir la liste des commandes passées après 17 h.
- i) Obtenir la liste des noms de pizzas commandées par le client de code `idClient 1`.
- j) Obtenir la liste des noms et prénoms des livreurs ayant livré une pizza chez le client de code `idClient 1`.

## 2 Création, remplissage et interrogation d'une base de données

Nous voulons créer une base de données des notes d'élèves. Chaque élève a un numéro étudiant, un prénom et un nom. Les devoirs ont un titre, une matière et un numéro d'identification unique. Il est alors possible de représenter cette base de données avec 3 tables, le schéma relationnel étant le suivant :

```
Eleves(idEleve INTEGER, Prenom TEXT, Nom TEXT)
Devoirs(idDevoir INTEGER, Titre TEXT, Matiere TEXT)
Resultats(idEleve INTEGER, idDevoir INTEGER, note INTEGER, rang INTEGER)
```

### 2.1 Création et remplissage

- a) Dans le schéma de base de données proposé, indiquer les attributs qui correspondent à des clefs primaires. Identifier les clefs étrangères.
- b) À l'aide des boutons *Nouvelle base de données*, *Créer table*, construire cette base de données. (Le programme affichera alors automatiquement les commandes SQL permettant de créer les tables.)
- c) Dans l'onglet *Parcourir les données*, à l'aide du bouton *Nouvel enregistrement*, alimenter les tables à l'aide des informations ci-après. Vous choisirez vous-même les identifiants des élèves.

## Notes au DS 3 (informatique) : programmation impérative

Jack	Sparrow	4
Peter	Pan	2
Sherlock	Holmes	16
Thomas	Anderson	20
Ellen	Ripley	17

## Notes au DS 7 (physique) : énergie mécanique

Jack	Sparrow	20
Peter	Pan	18
Sherlock	Holmes	10
Thomas	Anderson	4
Ellen	Ripley	15

## 2.2 Interrogation de la base

On utilisera maintenant l'onglet Exécuter le SQL pour interroger cette base de données.

- Afficher la liste des prénoms des élèves.
- Afficher la liste des matières traitées.
- Afficher la liste de toutes les notes obtenues. À l'aide de la commande `SELECT DISTINCT`, obtenir de même la liste des notes obtenues, sans répétition.
- Afficher la liste des matières où au moins un élève a eu moins de 3.
- Afficher la liste des noms des élèves ayant eu moins de 10 au devoir numéro 3.
- Afficher les prénoms des élèves qui se sont classés premiers à l'un au moins des devoirs.
- Afficher les numéros étudiants des élèves s'étant classés dans les 3 premiers aux deux devoirs.
- Afficher la listes des élèves qui étaient dans les 3 premiers au DS3 mais pas au DS7.

## 3 Réseau STAS

- Charger la base de données `lignes_STAS.sqlite` et étudier sa structure.
- Afficher la liste des lignes disponibles (sans répétitions).
- Afficher la liste des villes desservies par le réseau (toujours sans répétitions).
- Afficher tous les noms d'arrêts de la commune La Fouillouse.
- Combiner les deux tables en une seule contenant toutes les informations de façon cohérente.
- Afficher le plan de la ligne 10 (arrêts desservis et communes).
- Afficher l'ensemble des lignes desservant l'arrêt Grand Gonnet.
- Afficher le nom de l'arrêt de départ de chaque ligne.
  - Afficher les lignes partant de Saint-Étienne.
  - Afficher l'ensemble des lignes qu'on peut prendre pour aller de Grand Gonnet à Terrasse.
- Afficher les arrêts communs des lignes 2 et 3.
  - Afficher les arrêts desservis par la ligne 2 mais pas par la ligne 1 ni la ligne 3.
- Afficher l'ensemble des arrêts accessibles sans correspondance depuis le Lycée Fauriel (indication : on peut utiliser un `WHERE valeur IN table`).
- Afficher l'ensemble des arrêts accessibles avec 1 correspondance depuis le lycée.
- Le maire décide de changer le nom de la commune de Saint-Héand en Saint-Héant. Expliquer les modifications à effectuer. La structure de la base de données est-elle bien conçue ? Réfléchir à un nouveau schéma de base de données minimisant la redondance d'information.