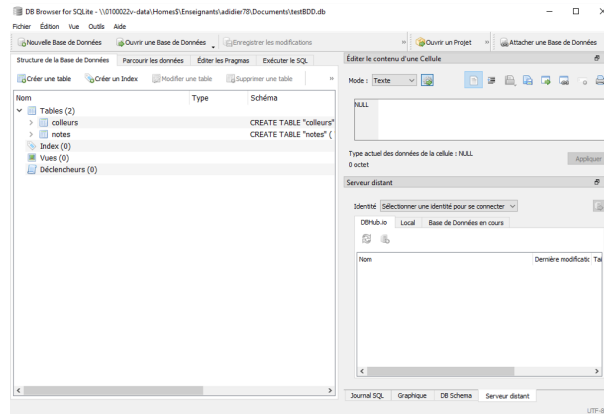


## TP classique : exemples de requêtes SQL

La semaine dernière, nous avons vu quelques exemples de requêtes dans le langage SQL : projections, sélections, jointure, fonctions d'agrégation... il est important de multiplier les exemples pour se familiariser avec ce langage qui manque parfois de souplesse.

Dans tout ce TP, on travaille **avec le logiciel SQLite** dont on pourra trouver une version sur l'ordinateur ou... en ligne tout simplement sur <https://sqliteonline.com/>:



Une fois la base de données ouvertes, on peut naturellement, à l'aide des onglets proposés, **parcourir les données** ou **exécuter des requêtes dans le langage SQL**.

### Exercice 1 (gestion d'une base de données des intégrations).

Dans cet exercice, on considère la base de données relatives à vos intégrations aux concours...

Nom	Type
▼ Tables (2)	
▼ écoles	
Nom	TEXT
Frais	INTEGER
Site	TEXT
%5/2	INTEGER
%filles	INTEGER
▼ élèves	
Nom	TEXT
Entrée	INTEGER
Naissance	INTEGER
Ecole	TEXT
Concours	TEXT

Celle-ci contient donc deux tables nommées **écoles** et **élèves**, avec des attributs différents.

- Ouvrir la base de données *BDD\_EX1.db*.
- Dans l'onglet **Exécuter le SQL**, écrire une requête pour afficher toutes les données de la table **écoles**, puis de la même façon celles de la table **élèves**.
- Dans la table **élèves**, pourquoi l'attribut **Concours** ne peut pas être une clef primaire ?
- Ecrire une requête qui renvoie le nom des élèves et l'école intégrée lorsque ces derniers ont obtenu le concours CCINP.
- Ecrire une requête qui renvoie le nombre d'élèves présents dans la table **élèves**.
- Ecrire une requête qui renvoie le nombre d'élèves qui a intégré chaque concours, ainsi que le concours associé.
- Ecrire une requête qui renvoie le nom des élèves, l'école et les frais d'inscription associés. On précisera la clef étrangère qui vous a permis de faire la jointure de ces données.
- Ecrire une requête qui renvoie le nom des élèves et l'année d'arrivée des élèves qui ont choisi de poursuivre en fac de maths.
- Ecrire une requête qui renvoie la moyenne des frais d'inscriptions.
- Ecrire une requête qui renvoie la liste des écoles, et leur site internet rangées par ordre croissant du pourcentage de filles présentes en école.
- Ecrire une requête qui renvoie le nom des élèves, l'école et le concours lorsque ces derniers ont intégré une école dont le pourcentage de 5/2 dépasse 20%.

**Exercice 2** (gestion d'une base de données relative à l'anniversaire de Jeannette).

[ ]

Jeannette a eu la gentillesse d'inviter quelques amis pour son anniversaire : le nom de ses amis... les boissons consommées... et même l'activité pêche qui leur propose, Jeannette a tout noté dans une base de données : c'est fou la campagne, avec toutes ces activités dans les champs, la boue et au bord de l'eau !

Nom	Type
▼ Tables (3)	
▼ boissons	
Boisson	TEXT
Titre	INTEGER
Prix	REAL
Quantité	INTEGER
▼ invités	
Nom	TEXT
Tel	TEXT
Célibataire	TEXT
Sexe	TEXT
Age	INTEGER
▼ pêche	
Nom	TEXT
Taille	INTEGER
Temps	INTEGER
Type	TEXT

Celle-ci contient donc trois tables nommées **invités**, **boissons** et **pêche**, avec des attributs différents.

1. Ouvrir la base de données *BDD\_EX2.db*.
2. Dans l'onglet **Exécuter le SQL**, écrire une requête pour afficher les noms et numéros de téléphone de ses amis.
3. Ecrire une requête qui renvoie le nom de ses copines célibataires.
4. Ecrire une requête qui renvoie le nom et le numéro de tous ses amis les plus âgés.
5. Ecrire une requête qui renvoie le nombre de boissons différentes.
6. Ecrire une requête qui renvoie le nombre de bouteilles d'alcool que Jeannette a prévues.
7. Ecrire une requête qui renvoie le nom, le sexe et la taille des poissons que chacun a pêchés. On précisera la **clef étrangère** qui vous a permis de faire la jointure de ces données.
8. Ecrire une requête qui renvoie le nom et le numéro de téléphone de celles et ceux qui n'ont pas pêchés.
9. Ecrire une requête qui renvoie le nombre de poissons en fonction de l'espèce.
10. Ecrire une requête qui renvoie la taille moyenne des poissons pêchés.
11. Ecrire une requête qui renvoie le nom, le sexe, l'âge des personnes qui ont réussi à attraper un poisson en moins de 8 minutes.
12. Ecrire une requête qui renvoie le meilleur pêcheur.

**Exercice 3** (gestion d'une base de données relative à la soirée beaux-gosses).

[ ]

Odin a décidé d'emmener ses potes faire la tournée des bars... certains sont plus riches que d'autres et pourront rentrer dans certains bars, mais pas tous, trop réputés, trop chers.

En tout cas, pas d'inquiétude : Zoé qui était célibataire dans l'EX2, va peut-être rencontrer l'amour lors d'une folle soirée avec deux guests !

C'est beau tout ça, mais d'abord soyons SELECT (*rires...* ok, blague de merde).

Celle-ci contient donc trois tables nommées **copains**, **bars** et **soirée**, avec des attributs différents.

1. Ouvrir la base de données *BDD\_EX3.db*.
2. Dans l'onglet **Exécuter le SQL**, écrire une requête qui renvoie le nombre de copains qui part faire la tournée des bars dans la voiture que Marwan leur a prêtée pour la soirée (la zarmouhmobile)... pas sûr que cela rentre.
3. Ecrire une requête qui renvoie la liste des copains et leurs surnoms du plus jeune au plus âgé.
4. Ecrire une requête qui renvoie le nom des copains riches et leur argent de poche, c'est à dire celles et ceux qui ont un argent de poche  $\geq 50$ .
5. Ecrire une requête qui renvoie le nombre de personnes riches regroupées suivant la valeur de leur argent de poche, ainsi que le montant associé.
6. Ecrire une requête qui renvoie les noms des connaissances de Odin qui ne sont pas des copains, mais qu'il a pu retrouver dans la soirée.
7. Ecrire une requête qui renvoie le surnom des copains de Odin qui ont suffisamment d'argent pour rentrer au ranch.
8. Ecrire une requête qui renvoie le nom des bars dans lesquels Yanis peut rentrer, ainsi que le prix des dépenses généralement constatées dans ce bar.
9. Ecrire une requête qui renvoie le nom et le prix moyen des bars rangés par ordre décroissant d'attractivité (5: super bar... 0: bof).
10. Ecrire une requête qui renvoie le nombre moyen de personnes rencontrées pour les copains de Odin lors de la soirée.
11. Ecrire une requête qui renvoie les surnoms des copains de Odin et l'heure du dodo pour celles et ceux qui se sont couchés avant 5 heures du matin.
12. Ecrire une requête qui renvoie l'heure et le nombre de personnes regroupées suivant l'heure à laquelle elles se sont couchées.

Et vous, vous vous en souvenez de cette soirée ?!

**Correction des exercices****EX1**

- 1.
2. `SELECT * FROM écoles ; SELECT * FROM élèves`
3. Il y a plusieurs lignes avec la même valeur. On ne peut donc pas identifier les relations à partir de cet attribut.
4. `SELECT Nom,Ecole FROM élèves WHERE Concours='CCINP'`
5. `SELECT COUNT(Nom) FROM élèves`
6. `SELECT Concours, COUNT(Concours) FROM élèves GROUP BY Concours`
7. On va réaliser une jointure sur le nom des écoles.  
`SELECT élèves.Nom,élèves.Ecole,écoles.Frais FROM élèves JOIN écoles ON élèves.Ecole=écoles.Nom`
8. `SELECT Nom,Entrée FROM élèves WHERE Ecole='UFR Maths'`
9. `SELECT AVG(Frais) FROM écoles`
10. `SELECT Nom,Site FROM écoles ORDER BY Filles ASC`
11. `SELECT élèves.Nom,élèves.Concours,écoles.Nom FROM élèves JOIN écoles ON élèves.Ecole=écoles.Nom WHERE Cinqdem>20`

**EX2**

- 1.
2. `SELECT Nom,Tel FROM invités`
3. `SELECT Nom FROM invités WHERE Célibataire='oui' AND Sexe='F'`
4. `SELECT Nom,Tel FROM invités WHERE Age=(SELECT MAX(Age) FROM invités)`
5. `SELECT COUNT(DISTINCT Boisson) FROM boissons`
6. `SELECT SUM(Quantité) FROM boissons`
7. On réalise la jointure sur le nom des invités et des pêcheurs.  
`SELECT invités.Nom,Sexe,Taille FROM invités JOIN pêche ON invités.Nom=pêche.Nom`
8. `SELECT Nom,Tel FROM invités WHERE Nom NOT IN (SELECT Nom FROM pêche)`
9. `SELECT Type,COUNT(*) FROM pêche GROUP BY Type`
10. `SELECT AVG(Taille) FROM pêche`
11. `SELECT invités.Nom,Sexe,Age FROM invités JOIN pêche ON invités.Nom=pêche.Nom WHERE Temps<8`
12. On peut convenir que le meilleur pêcheur est celui qui attrape un poisson rapidement... et ainsi, avec cette convention:  
`SELECT Nom FROM pêche WHERE Temps=(SELECT MIN(Temps) FROM pêche)`

**EX3**

- 1.
2. `SELECT COUNT(nom) FROM copains`
3. `SELECT nom,surnom FROM copains ORDER BY age ASC`
4. `SELECT nom,argentdepoeche FROM copains WHERE argentdepoeche>=50`
5. `SELECT COUNT(nom),argentdepoeche FROM copains GROUP BY argentdepoeche HAVING argentdepoeche>=50`
6. `SELECT nom FROM soirée WHERE nom NOT IN (SELECT nom FROM copains)`
7. `SELECT surnom FROM copains WHERE argentdepoeche>=(SELECT prix FROM bars WHERE nomdubar='ranch')`
8. `SELECT nomdubar,prix FROM bars WHERE (SELECT argentdepoeche FROM copains WHERE nom='Yanis')>=prix`
9. `SELECT nomdubar,prix FROM bars ORDER BY attractivité DESC`
10. `SELECT AVG(fillesougars) FROM soirée`
11. `SELECT surnom,dodo FROM copains JOIN soirée ON copains.nom=soirée.nom WHERE dodo<5`
12. `SELECT dodo AS heuredudodo,COUNT(nom) AS nombre FROM soirée GROUP BY dodo`