Lister le contenu d'une table

Faire défiler une table à l'écran, c'est bien pour douze lignes. Pour le reste, on pose des questions à la base en SQL. Première requête : afficher le contenu d'une table, choisir ses colonnes et renommer ce qu'on affiche.

4TTR 5TTR 6TTR Découverte

L'onglet **Parcourir les données** montre une table comme un tableur. C'est pratique pour jeter un coup d'œil, mais une base de données sert surtout à **répondre à des questions**, y compris quand elle contient des milliers de lignes. Pour lui poser une question, on écrit une requête en SQL.

> 💡 Télécharge `mon-ecole-eleves.db` dans la sidebar et ouvre-la dans DB Browser for SQLite. Elle contient une table `eleve` de douze élèves, avec les colonnes `id`, `nom`, `prenom`, `date_naissance`, `commune`, `email` et `absences`.

🎯 Objectifs

À la fin de cet article, tu seras capable de :

1. Écrire et exécuter une requête dans DB Browser.
2. Afficher toutes les colonnes d'une table avec `SELECT *`.
3. Choisir les colonnes à afficher et leur ordre.
4. Renommer une colonne dans le résultat avec `AS`.
5. Comprendre les messages d'erreur les plus fréquents et repérer l'erreur qui n'en produit pas.

Poser une première question

Ouvre l'onglet **Exécuter le SQL**. La zone du haut sert à écrire, la zone du bas affiche le résultat.

Écris cette ligne :

```sql
SELECT * FROM eleve;
```

Exécute-la avec le bouton ▶ ou avec **Ctrl+Entrée**. Le résultat affiche les douze élèves :

```
id | nom       | prenom  | date_naissance | commune   | email                    | absences
---+-----------+---------+----------------+-----------+--------------------------+---------
1  | Adam      | Lucas   | 2010-03-12     | Jodoigne  | lucas.adam@ecole.be      | 2
2  | Bastin    | Emma    | 2010-07-04     | Wavre     | emma.bastin@ecole.be     | 0
3  | Charlier  | Noah    | 2009-11-23     | Perwez    | NULL                     | 5
4  | Delvaux   | Léa     | 2010-04-15     | Jodoigne  | lea.delvaux@ecole.be     | 1
5  | Englebert | Hugo    | 2010-01-30     | Hannut    | hugo.englebert@ecole.be  | 8
6  | Fontaine  | Chloé   | 2009-09-08     | Jodoigne  | NULL                     | 0
7  | Gérard    | Nathan  | 2010-05-19     | Wavre     | nathan.gerard@ecole.be   | 3
8  | Hubert    | Manon   | 2010-02-27     | Perwez    | manon.hubert@ecole.be    | 0
9  | Istas     | Louis   | 2009-12-11     | Jodoigne  | louis.istas@ecole.be     | 12
10 | Jacques   | Camille | 2010-06-02     | Hannut    | NULL                     | 4
11 | Kevers    | Arthur  | 2009-10-17     | Ramillies | arthur.kevers@ecole.be   | 1
12 | Lambert   | Sarah   | 2010-09-01     | jodoigne  | sarah.lambert@ecole.be   | 0
```

Tu viens d'écrire ta première requête.

> 📖 **NOUVELLE NOTION : LA REQUÊTE**
>
> Une **requête** est une question ou un ordre écrit en SQL et envoyé à la base de données. La base l'exécute et renvoie un résultat, souvent sous forme de tableau.
>
> En anglais : GB ***query***.

Lire la requête mot à mot

Une requête SQL se lit presque comme une phrase en anglais. Traduisons-la morceau par morceau :

Morceau	Traduction	Rôle
`SELECT`	sélectionne	annonce ce qu'on veut voir
`*`	tout	toutes les colonnes
`FROM`	depuis	annonce où chercher
`eleve`		le nom de la table
`;`		la fin de la requête

La phrase complète donne : « **sélectionne toutes les colonnes depuis la table** `eleve` ».

> 📖 **NOUVELLE NOTION : SELECT … FROM**
>
> **SELECT … FROM** est l'instruction SQL qui lit des données. Après `SELECT`, on indique les colonnes à afficher ; après `FROM`, la table où les chercher. Elle ne modifie jamais la table.

En anglais : *select* veut dire « sélectionner » et *from* veut dire « depuis ».

💡 **MAJUSCULES OU MINUSCULES ?** SQL ne fait pas la différence : `select * from eleve` fonctionne aussi. Mais on écrit toujours les **mots-clés** SQL en MAJUSCULES et les noms de tables et de colonnes en minuscules. La requête devient beaucoup plus facile à lire.

💡 **LE POINT-VIRGULE** marque la fin d'une requête. DB Browser tolère son absence quand il n'y a qu'une seule requête, mais prends dès maintenant l'habitude de le mettre : dès que tu en écriras plusieurs à la suite, il deviendra indispensable.

Choisir les colonnes à afficher

Tu n'as pas toujours besoin de tout. Pour préparer une liste d'appel, le prénom et le nom suffisent. Remplace l'étoile par les colonnes voulues, séparées par des virgules :

```sql
SELECT prenom, nom FROM eleve;
```

```
prenom   | nom
---------+----------
Lucas    | Adam
Emma     | Bastin
Noah     | Charlier
Léa      | Delvaux
Hugo     | Englebert
Chloé    | Fontaine
Nathan   | Gérard
Manon    | Hubert
Louis    | Istas
Camille  | Jacques
Arthur   | Kevers
Sarah    | Lambert
```

Deux choses à remarquer :

- les colonnes s'affichent **dans l'ordre où tu les écris**, pas dans l'ordre de la table ;
- les lignes, elles, sont toujours toutes là : choisir des colonnes ne retire aucune ligne.

🧠 SELECT **RÉPOND À LA QUESTION : « QU'EST-CE QUE JE VEUX VOIR ? »**

ℹ️ Les informaticiens appellent **PROJECTION** le fait de ne garder que certaines colonnes d'une table. Tu croiseras peut-être ce mot dans des livres ou sur internet.

Écrire sur plusieurs lignes

SQL ignore les retours à la ligne et les espaces en trop. Les deux écritures suivantes sont identiques :

```sql
SELECT prenom, nom, commune FROM eleve;
```

```sql
SELECT prenom, nom, commune
FROM eleve;
```

Dès qu'une requête s'allonge, place chaque mot-clé au début d'une nouvelle ligne : on la relit d'un coup d'œil.

Renommer une colonne dans le résultat : AS

Les noms de colonnes sont faits pour la base, pas pour les humains. `date_naissance` n'est pas très joli dans un document. Tu peux afficher la colonne sous un autre nom :

```sql
SELECT nom, date_naissance AS naissance
FROM eleve;
```

```
nom       | naissance
----------+-----------
Adam      | 2010-03-12
Bastin    | 2010-07-04
Charlier  | 2009-11-23
Delvaux   | 2010-04-15
Englebert | 2010-01-30
Fontaine  | 2009-09-08
Gérard    | 2010-05-19
Hubert    | 2010-02-27
Istas     | 2009-12-11
Jacques   | 2010-06-02
Kevers    | 2009-10-17
Lambert   | 2010-09-01
```

📖 **NOUVELLE NOTION : L'ALIAS**

Un **alias** est un nom provisoire donné à une colonne dans le résultat d'une requête, avec le mot-clé `AS`. Il ne change rien dans la table : seul l'affichage est concerné.

En anglais : *alias*, « autre nom » ; *as* veut dire « comme ».

ℹ️ Pour un alias qui contient des espaces ou des accents, entoure-le de guillemets droits : `date_naissance AS "Date de naissance"`.

Quand la base ne comprend pas

Tu vas faire des erreurs, comme tout le monde. SQLite les signale par un message en anglais, court mais précis. Apprends à les lire.

Une table qui n'existe pas.

```
SELECT * FROM eleves;
```

```
no such table: eleves
```

« Pas de table de ce nom : eleves ». La table s'appelle `eleve`, au singulier.

Une colonne qui n'existe pas.

```
SELECT nom, prenon FROM eleve;
```

```
no such column: prenon
```

« Pas de colonne de ce nom ». Une lettre de travers suffit.

Une erreur de syntaxe.

```
SELECT prenom, nom, FROM eleve;
```

```
near "FROM": syntax error
```

« Erreur de syntaxe près de FROM ». La virgule annonçait une colonne de plus, et SQLite est tombé sur `FROM`. Le mot cité dans le message indique **où** SQLite s'est perdu : l'erreur se trouve juste avant.

L'erreur qui ne dit rien

Voici la plus traître. Exécute cette requête, où la virgule a été oubliée :

```
SELECT nom prenom FROM eleve;
```

```
prenom
---------
Adam
Bastin
Charlier
Delvaux
Englebert
Fontaine
Gérard
Hubert
Istas
```

```
12 | Jacques
13 | Kevers
14 | Lambert
```

Aucun message d'erreur. Et pourtant, le résultat est faux : une seule colonne, intitulée `prenom`, qui contient… les noms.

Sans la virgule, SQLite a compris `nom AS prenom` : le mot `AS` est facultatif, et `prenom` est devenu l'alias de la colonne `nom`.

> ⚠️ **UNE REQUÊTE QUI S'EXÉCUTE N'EST PAS FORCÉMENT UNE REQUÊTE JUSTE.** Après chaque exécution, vérifie que le résultat répond vraiment à la question : bon nombre de colonnes, bons titres, valeurs qui ont du sens.

La grille ou la requête ?

	Parcourir les données	Requête `SELECT`
Pour quoi faire ?	jeter un coup d'œil, corriger une case	répondre à une question précise
Peut-on la réutiliser ?	non, il faut refaire les clics	oui, on la garde dans un fichier
Un programme peut-il l'utiliser ?	non	oui

Une requête s'enregistre, se partage et se réexécute à l'identique. C'est exactement ce que fera un programme qui utilise la base.

🧪 Exercices

Exercice 1 — Premières requêtes ★☆☆☆☆

Écris et exécute une requête pour afficher :

1. toutes les colonnes de la table `eleve` ;
2. uniquement les prénoms ;
3. le nom et la commune de chaque élève ;
4. l'adresse e-mail, puis le prénom, puis le nom, dans cet ordre ;
5. le nom et le nombre d'absences, avec la colonne `absences` affichée sous le nom `demi_jours`.

Exercice 2 — Prédire avant d'exécuter ★★☆☆☆

Pour chaque requête, **écris d'abord sur papier** combien de colonnes et combien de lignes le résultat contiendra, et le titre de chaque colonne. Exécute ensuite pour vérifier.

```sql
-- a)
SELECT id FROM eleve;

-- b)
SELECT commune, commune FROM eleve;

-- c)
SELECT prenom AS nom, nom AS prenom FROM eleve;

-- d)
SELECT email adresse FROM eleve;
```

Explique ce qui se passe dans la requête d).

Exercice 3 — Réparer des requêtes ★★☆☆☆

Chaque requête contient une erreur. Exécute-la, recopie le message, explique l'erreur et corrige-la.

```sql
-- a)
SELECT * eleve;

-- b)
SELEC nom FROM eleve;

-- c)
SELECT nom, prenom, email, FROM eleve;

-- d)
SELECT nom, date_de_naissance FROM eleve;
```

Exercice 4 — La virgule oubliée ★★★☆☆

1. Exécute `SELECT prenom nom, commune FROM eleve;`.
2. Combien de colonnes obtiens-tu, et quels sont leurs titres ?
3. Explique pourquoi SQLite n'a signalé aucune erreur.
4. Un collègue utilise ce résultat pour imprimer des étiquettes « Nom : … ». Qu'est-il imprimé sur l'étiquette de Lucas Adam ?

Exercice 5 — Préparer des documents ★★★☆☆

Le secrétariat a besoin de trois listes. Pour chacune, écris la requête, avec des titres de colonnes lisibles grâce aux alias.

1. Une liste d'appel : prénom et nom.
2. Une liste pour envoyer un courrier électronique : prénom, nom et adresse e-mail.
3. Un relevé pour l'éducateur : nom, prénom et nombre de demi-jours d'absence.

Pour la liste 2, qu'est-ce qui pose problème dans le résultat ? Note ta réponse : ce problème se règle avec un outil que tu n'as pas encore vu.

✍️ À retenir

- Une **requête** est une question écrite en SQL ; on l'exécute dans l'onglet **Exécuter le SQL** avec Ctrl+Entrée.
- `SELECT … FROM …` lit des données sans jamais modifier la table.
- `*` affiche toutes les colonnes ; sinon, on liste les colonnes voulues, séparées par des virgules, dans l'ordre d'affichage souhaité.
- `AS` donne un **alias** à une colonne, uniquement dans le résultat.
- Mots-clés en MAJUSCULES, noms de tables et de colonnes en minuscules, point-virgule à la fin.
- Le message d'erreur cite l'endroit où SQLite s'est perdu ; mais une requête sans erreur peut donner un résultat faux.

Suite

Les mêmes requêtes peuvent être envoyées par un programme plutôt que tapées à la main. L'article Lire une table depuis Python te montre comment.