

# BDD SQL – Séance 1

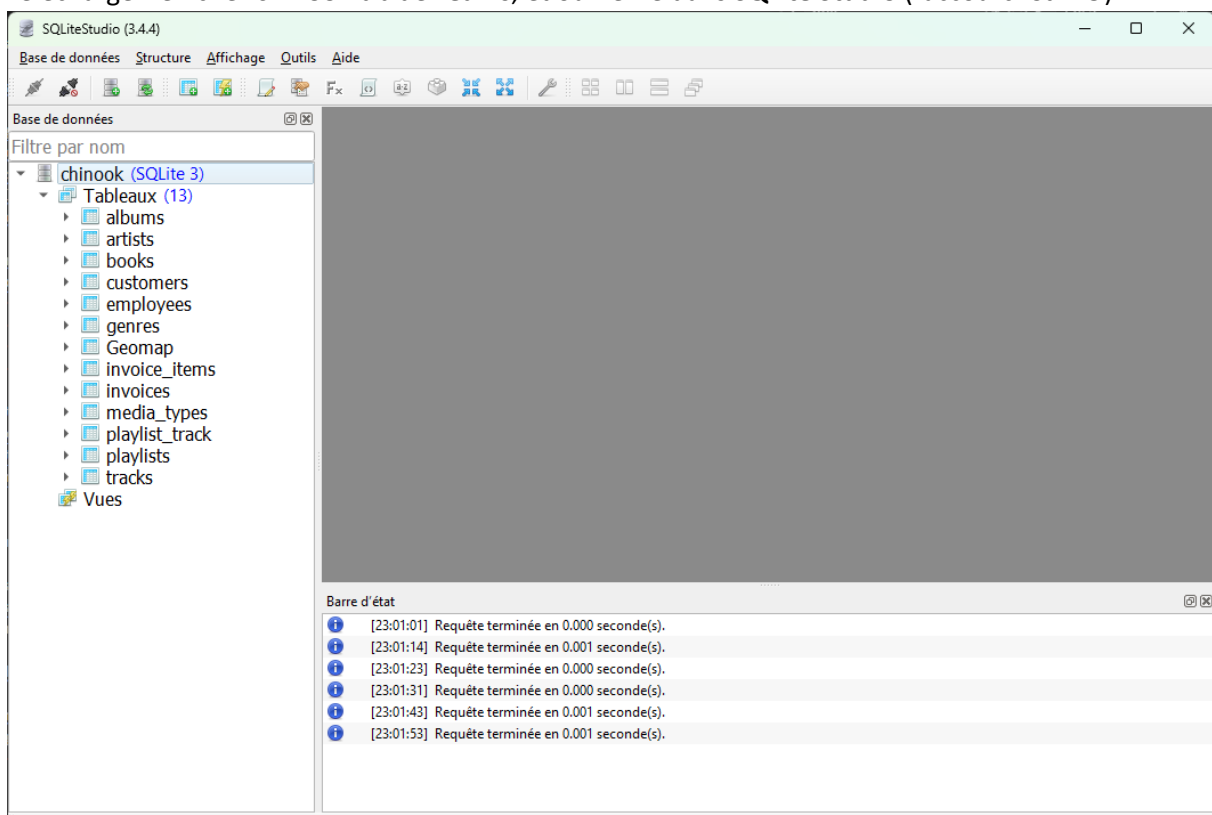
Ce fichier contient plusieurs questions. Ecrivez vos réponses dans **Bloc-notes**, et sauvegardez-le comme **Prénom-NOM-SQL-TD1.txt**, où vous mettez votre nom pour **Prénom-NOM**. Pour chaque question, mettez une ligne contenant uniquement le numéro de ligne suivi d'un point, et mettez votre réponse dans la ligne suivante. Par exemple, les premières lignes du fichier – correspondant aux questions 1 et 2 en bas – seront

```
1.
select * from customers where country = "USA" ;
2.
13
3.
...
```

**Attention :** Si la réponse est de la forme d'une requête "SELECT ...", tapez-la d'abord dans l'éditeur SQL, pour vérifier si ça marche (voir en bas). Puis, copiez-collez la requête dans Bloc-notes.

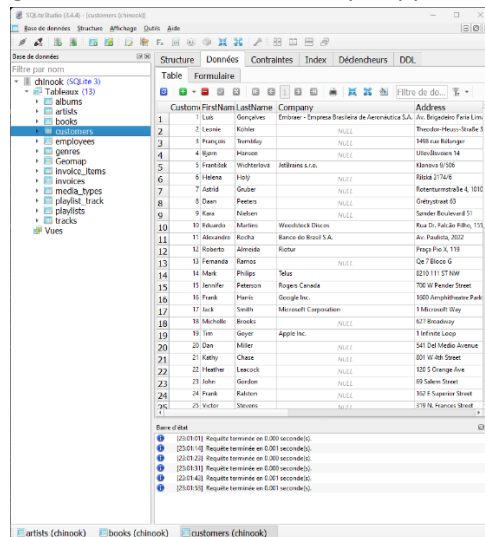
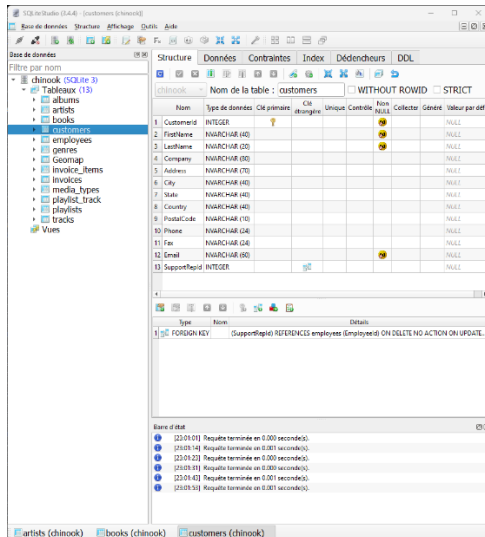
Quand vous aurez finis, déposez le fichier sur Teams. Si vous ne pouvez pas accéder à Teams, envoyez-moi vos réponses par email à [jonas.frey@univ-paris13.fr](mailto:jonas.frey@univ-paris13.fr).

1. S'il n'est pas encore installé sur votre ordinateur, téléchargez et installez le logiciel **SQLite Studio** de <https://sqlitestudio.pl/>
2. Téléchargez le fichier **chinook.db** de **Teams**, et ouvrez le dans **SQLite Studio** (raccourci Ctrl+O).

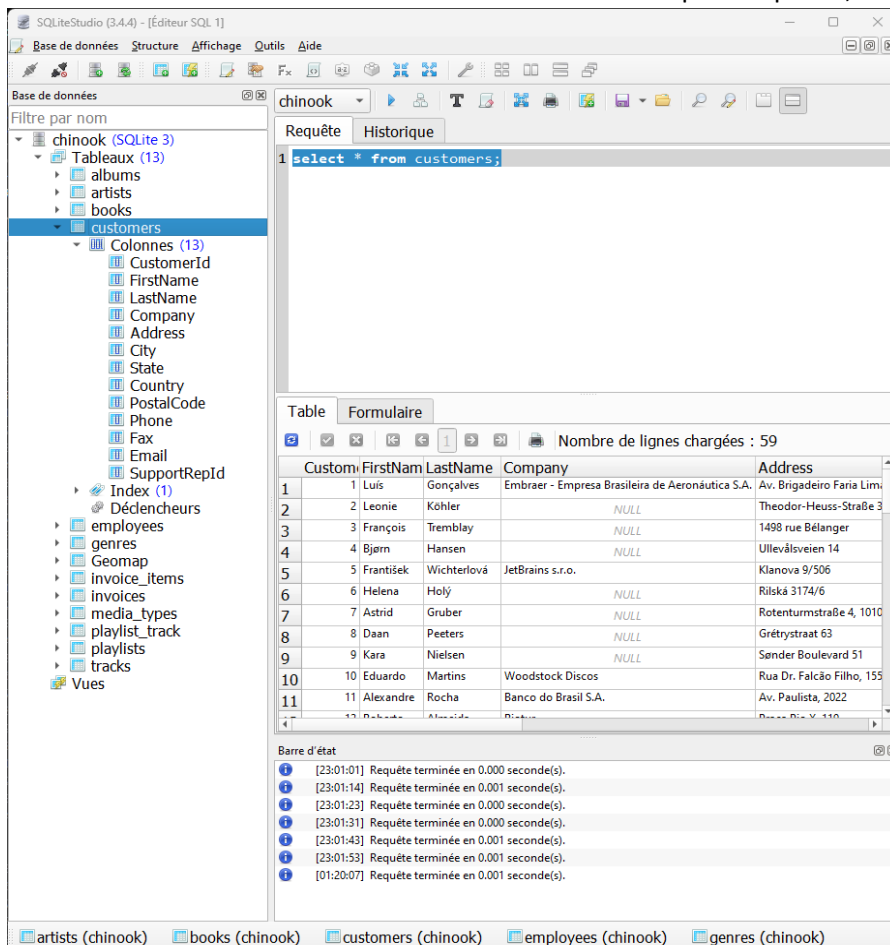


3. Si la barre d'état ou le panneau ne sont pas visibles, cochez la case dans le menu **Affichage**.
4. Pour afficher la liste des tables, cliquez sur le triangle à gauche du mot **chinook**.
5. Pour voir le contenu d'une table, cliquez sur le nom de la table, nous commençons avec le table **customers** (clients).

6. Pour le moment, nous nous intéressons pour les onglets **Structure** et **Données** qui apparaissent à droite :



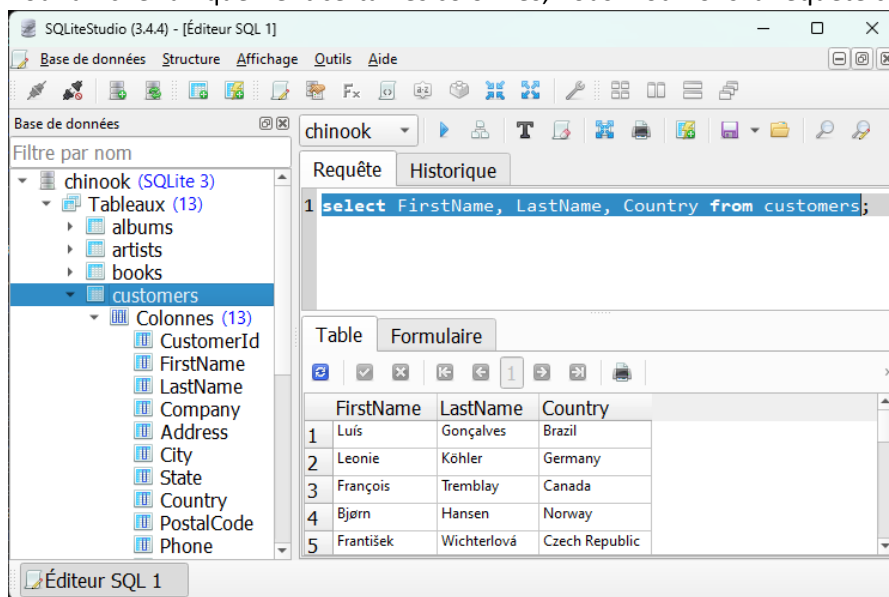
7. L'onglet **Structure** donne une liste des **colonnes** (ou « attributs ») de la table, et l'onglet **Données** contient les données. On peut aussi afficher les noms de colonnes en cliquant sur le triangle à gauche du nom de la table dans le panneau.
8. Pour interroger la base de données, nous devons ouvrir l'**éditeur SQL** (Alt+E ou Outils → Ouvrir l'éditeur SQL).
9. Nous écrivons nos requêtes dans le champ supérieur et les exécutons en cliquant sur le triangle bleu ou avec la touche F9.
10. Écrivez `select * from customers;` dans le champ de requêtes, et tapez F9.



Le contenu de la table s'affichent en bas.

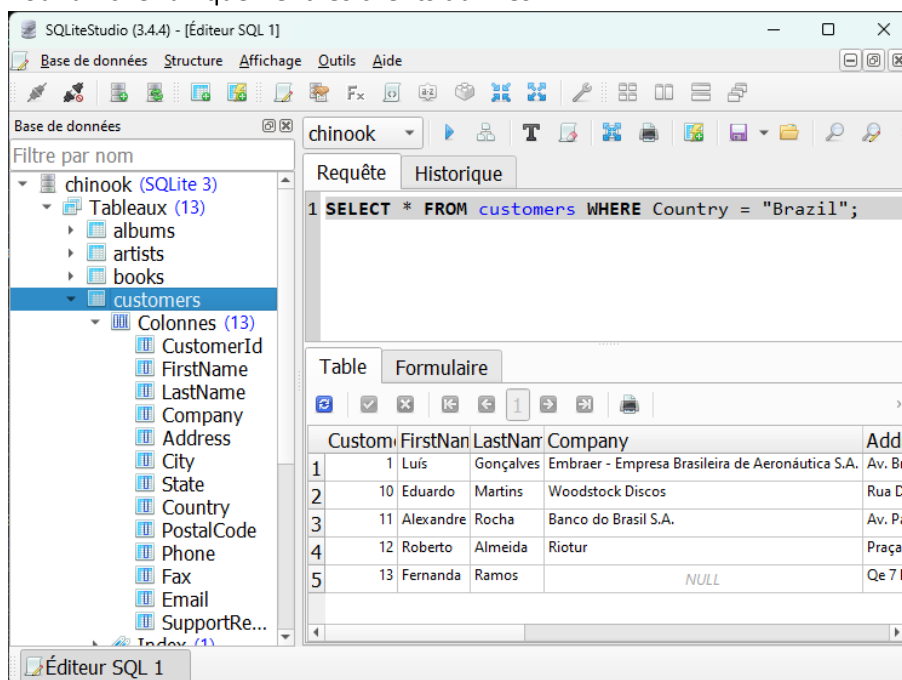
## Sélectionner des colonnes

Pour afficher uniquement certaines colonnes, nous modifions la requête comme suit. Essayez-le !



## Filtres

Pour afficher uniquement les clients du Brésil :



- Exercice 1. Donnez une requête pour afficher seulement les clients des États-Unis.  
 Exercice 2. Combien y a-t-il ?  
 Exercice 3. Donnez une requête pour afficher seulement les clients avec le Prénom « Frank ».

Pour effectuer des requêtes insensibles à la casse, nous pouvons utiliser **LIKE**. Essayez :

- `SELECT * FROM customers WHERE Firstname = "jack";`
- `SELECT * FROM customers WHERE Firstname LIKE "jack";`

## L'opérateur joker "%"

Pour afficher uniquement les clients dont le prénom commence par A, nous pouvons utiliser le **caractère générique** « % ». Essayez :

- `SELECT * FROM customers WHERE Firstname LIKE "a%";`

- Exercice 4. Donnez une requête pour afficher seulement les clients dont le prénom contient un « b » (pas forcément au début).
- Exercice 5. Donnez une requête pour afficher uniquement les entrées de la table **customers** qui ont une adresse email se terminant par **@gmail.com**.
- Exercice 6. Combien y a-t-il ?
- Exercice 7. Donnez une requête pour afficher uniquement les clients dont le prénom contient deux « e » (pas nécessairement consécutifs).
- Exercice 8. Combien y a-t-il ?
- Exercice 9. Quelles lignes de la **table suivante** sont acceptés par la requête `SELECT * FROM clients WHERE LastName LIKE "%a%n";` ?  
Donnez la liste des numéros de ligne (colonne **ID**) dans votre réponse.

Table: clients	
ID	LastName
1	André
2	Naiman
3	Amon
4	Veblen
5	KERAN
6	Makna
7	NAN
8	Karn
9	MEYER
10	An

- Exercice 10. Donnez une requête pour afficher les lignes de la table **albums** dont la colonne **Title** contient le mot "concerts".
- Exercice 11. Combien y a-t-il ?
- Exercice 12. Donnez une requête pour afficher les lignes de la table **albums** dont la colonne **Title** commence par le mot "The".
- Exercice 13. Combien y a-t-il ?

Il y a une variante de l'**opérateur joker** '%' : le symbol '\_' (*tiret bas*, en anglais *underscore*) représente **précisément un** caractère. Par exemple, la requête

```
select * from customers where FirstName LIKE "_u%"
```

affiche toutes les lignes de la table **customers** dont le deuxième caractère de **FirstName** est un 'u'.

- Exercice 14. Donnez une requête pour afficher toutes les lignes de la table **customers** dont le troisième caractère de **FirstName** est un 'a'.
- Exercice 15. En utilisant '\_', donnez une requête pour afficher toutes les lignes de la table **customers** dont **FirstName** a précisément quatre caractères.

Une autre façon de donner des conditions sur la longueur d'une chaîne de caractères est en utilisant **LENGTH**. Par exemple,

- `SELECT * FROM customers WHERE LENGTH(FirstName)=7`

affiche toutes les entrées de **customers** dont **FirstName** a la longueur 7.

Exercice 16. Écrivez une requête sur la table **Customers** qui affiche tous les clients dont l'adresse contient exactement 15 caractères.

Exercice 17. Combien y a-t-il ?

## Opérateurs booléens

Pour comparer des valeurs on peut utiliser les opérateurs

=	égal
LIKE	égal à minuscule/majuscule près
!=	différent
<	inférieur à
<=	inférieur ou égal
>	supérieur à
>=	supérieur ou égal

Par exemple, pour afficher seulement les factures dont le prix « total » vaut moins de \$10.00, nous utilisons

- `SELECT * FROM invoices WHERE total < 10;`

Exercice 18. Donnez une requête pour afficher uniquement les factures dont le prix total vaut au moins \$20.00 (utilisez l'opérateur >= pour « supérieur ou égal »)

Exercice 19. Combien y a-t-il ?

Exercice 20. Donnez une requête pour afficher seulement les lignes de la table **invoices** dont le **Total** est différent de **0.99**

Exercice 21. Combien y a-t-il ?

Exercice 22. Donnez une requête pour afficher les lignes la table **customers** dont **FirstName** a plus de quatre caractères.

On peut combiner des conditions booléennes avec **AND** ("et") et **OR** ("ou") :

- `SELECT * FROM invoices WHERE total < 10 AND BillingCity = "Stuttgart";`
- `SELECT * FROM invoices WHERE total < 1 OR CustomerID = 3;`

Exercice 23. Donnez une requête pour afficher seulement les lignes de la table **invoices** dont le total est soit **0.99** soit **1.98**

Exercice 24. Combien y a-t-il ?

Exercice 25. Donnez une requête pour afficher les entrées de la table **invoices** dont **BillingCountry** est égal à **Germany** et **TOTAL** est différent de **1.98**.

Pour tester si une valeur est dans une liste, on peut utiliser **IN**. Par exemple, la requête

- `SELECT *  
FROM customers  
WHERE LastName IN ("Peeters", "Nielsen", "Martins");`

affiche les lignes du tableau **customers** dont la valeur de **LastName** est soit "Peeters", soit "Nielsen", soit "Martins".

Cette requête est trop longue pour une ligne, et on l'affiche sur plusieurs lignes. En fait, il y a une façon standard pour afficher des requêtes compliquées, qu'on peut produire dans l'éditeur SQL avec le raccourci **CTRL+T**.

Exercice 26. Donnez une requête équivalente à celle de numéro 23, en utilisant **IN**.

Pour *nier* une condition on utilise **NOT**. Par exemple, les deux requêtes équivalentes suivantes rendent les lignes du tableau **customers** dont la valeur de **LastName** est ni Peeters, ni Nielsen, et ni Martins.

- ```
SELECT *  
  FROM customers  
 WHERE LastName NOT IN ("Peeters", "Nielsen", "Martins");
```
- ```
SELECT *  
  FROM customers  
 WHERE NOT (LastName = "Peeters" OR  
           LastName = "Nielsen" OR  
           LastName = "Martins");
```

Exercice 27. (*Difficile*) Écrivez une requête équivalente aux deux exemples précédents en utilisant **!=** et **AND**. Formatez la requête dans l'éditeur SQL en utilisant le raccourci **Ctrl+T**, puis copiez-la dans votre fichier de réponses.

## Sélection de colonnes et répétitions

Pour afficher seulement certaines colonnes d'une table, on liste les noms des colonnes à la place de l'étoile dans la requête. Par exemple, la requête

- ```
SELECT BillingCity, BillingCountry FROM invoices;
```

affiche uniquement les colonnes **BillingCity** et **BillingCountry** de la table **invoices**. Cette liste a des répétitions, parce qu'il peut y avoir différentes factures avec la même ville. Pour supprimer les répétitions, on utilise **DISTINCT** : la requête

- ```
SELECT DISTINCT BillingCity, BillingCountry FROM invoices;
```

supprime les répétitions.

Exercice 28. Combien de villes différentes y a-t-il dans la table des factures ?

Exercice 29. Donnez une requête pour afficher seulement les colonnes **FirstName** et **Country** de la table **customers**, en supprimant les répétitions.

Exercice 30. Donnez une requête listant sans répétitions les valeurs de la colonne **BillingCity** de la table **invoices** dont le valeur **BillingCountry** correspondant est "USA".

Exercice 31. Combien y a-t-il ?

## Trier les résultats

On peut trier les résultats de manière alphabétique ou numérique en utilisant **ORDER BY ... ASC/DESC**. Par exemple, la requête

- ```
SELECT DISTINCT BillingCity,  
               BillingCountry  
  FROM invoices  
 ORDER BY BillingCountry ASC;
```

affiche le résultat de l'exemple précédent, trié de manière alphabétique par **BillingCountry**. Si on remplace **ASC** par **DESC**, on obtient l'ordre inverse.

Exercice 32. Donnez une requête qui affiche le résultat de l'exercice 30 de manière alphabétique.

Exercice 33. Copiez-collez la liste des villes d'exercice 32.

## Calculs

On peut afficher des valeurs qui sont calculés à partir de valeurs numériques dans un tableau. Par exemple, la requête

- ```
SELECT Milliseconds / 1000 AS Seconds FROM tracks;
```

affiche la valeur de **Milliseconds / 1000** -- arrondie à l'entier inférieur -- dans une colonne nommée **Seconds**. Pour afficher la valeur non-arrondie on utilise

- `SELECT Milliseconds / 1000.00 AS Seconds FROM tracks;`

et pour afficher la valeur à deux décimales près, on utilise

- `SELECT ROUND(Milliseconds / 1000.0, 2) AS Seconds FROM tracks;`

où à la place de 2 on peut mettre le numéro de décimales désiré.

- Exercice 34. Donnez une requête qui affiche la colonne **Name** du tableau **tracks**, ainsi que l'expression **Bytes \* 1000.0 / Milliseconds** arrondi à deux décimales près, dans une colonne nommée **ByteRate**, trié en ordre croissant.
- Exercice 35. Quel track a la plus petite **ByteRate** ?

## Joins (s'il reste du temps)

Parfois il faut faire des requêtes sur plusieurs tables en même temps. Par exemple, imaginons qu'on veut savoir les noms des clients qui ont fait des ordres de valeur **TOTAL** 0.99. Le problème est que la table **invoices** contient pas de noms de clients, seulement une colonne **customerid**. La solution est de combiner les tables **invoices** et **customers** en leur "joignant" à travers l'attribut **customerid**, avec la requête suivante :

- ```
SELECT DISTINCT FirstName,
                  LastName
FROM invoices
INNER JOIN
customers ON invoices.customerid = customers.customerid
WHERE TOTAL = 0.99;
```

- Exercice 36. Écrivez une requête avec **INNER JOIN** pour afficher les valeurs **TOTAL** de toutes les commandes de "Frank Harris", en ordre décroissant.
- Exercice 37. Copiez-collez la liste de numéros affichée par la requête précédente.