

Type de document	TP	Classe	Tle	Durée	2h	Date	13/03/2025
Thème et contenu(s)	Bases de données						
Capacités attendues	Faire un MCD, un MLD, écrire des requêtes simples avec une seule jointure entre 2 tables						
Prérequis	Cours BDD						
Description	TP en ligne sur sqlite.com						

I) Le MCD (2 min)

Une entreprise désire stocker ses clients et les commandes qui ont été passées. Un client a un id, un nom, un prénom, une ville. Une commande a un id, une date, un montant. Modéliser ce fonctionnement au moyen d'un MCD.

II) Le MLD (3 min)

Maintenant que nous avons notre MCD, nous devons créer le MLD, c'est-à-dire le schéma relationnel correspondant. C'est à vous !

III) Le MPD (10 min)

On va maintenant créer nos tables, c'est à vous ! Ensuite il vous faut aller l'implémenter sur sqliteonline.com

IV) Insertion de « dummy data » (5 min)

Dans un premier temps vous pouvez vous entraîner à insérer des données (i.e. des enregistrements) dans la base de données. Vous pouvez le faire à la volée en séparant vos différentes requêtes par un point-virgule. Rappel : soit vous gérez vous-mêmes les id, soit vous laissez le SGBD s'en occuper (c'est ce que nous faisons ici). 2 manières de faire si le SGBD gère les id :

- `INSERT INTO table (att1,...attn) VALUES (val1,...,valn)` : ici on spécifie les attributs que l'on veut insérer (sauf la clef) et on donne leurs valeurs, attention à ce qu'il y ait autant d'attributs que de valeurs
- `INSERT INTO table VALUES (NULL, val1,...,valn)` : ici on veut insérer une valeur pour chaque attribut donc pour gagner du temps on ne les cite pas. En revanche comme l'id fait partie des attributs et qu'on veut le renseigner, on va le mettre à NULL afin que MySQL lui assigne un numéro (n'oubliez pas qu'il est auto-increment !)

En avant ! D'abord la première méthode, ensuite la seconde.

```
INSERT INTO client (nom, prenom, ville) VALUES
('PUZZA', 'Maguy', 'Ajaccio'),
('ZUMORTU', 'Amhed', 'Ajaccio'),
('PORTA', 'Sarah', 'Ajaccio');
```

```
INSERT INTO client VALUES
(NULL, 'KOLIK', 'Al', 'Bastia'),
(NULL, 'TAZEGA', 'Simon', 'Bastia'),
(NULL, 'TALU', 'Jean', 'Corte'),
(NULL, 'RIEN', 'Jean-Sé', 'Bastelicaccia');
```

1. On vérifie que tout s'est bien passé en faisant : `SELECT * FROM client` ;

On fait pareil pour les commandes :

```
INSERT INTO commande VALUES
(NULL, '2020-11-04', 258.3, 1),
(NULL, '2020-11-04', 6894.25, 1),
(NULL, '2020-10-05', 45323.0, 5),
(NULL, '2020-12-20', 2583.9, 1),
(NULL, '2020-12-05', 456.0, 2),
(NULL, '2020-12-08', 789.6, 2),
(NULL, '2020-05-23', 1500.0, 1),
(NULL, '2020-11-04', 1287.12, NULL),
(NULL, '2020-01-03', 1586.2, 6);
```

2. Que signifie le NULL en dernier attribut de l'avant-dernière ligne ?

Notez que vous devez d'abord créer des clients pour ensuite créer des commandes sous peine d'avoir des id clients qui n'existent pas (encore) ! En pratique si vous voulez créer une commande alors que le client n'existe pas dans la base, vous mettrez dans commande sa clef à NULL !

Vous devrez ensuite renseigner la bonne clef grâce à une requête UPDATE (facile, mais hors-programme).

3. Essayez de DROP la table client : que se passe-t-il ? Pourquoi ?

Faites maintenant DROP de la table commande puis de la table client, et importez sur sqliteonline.com le fichier jointures_sqlite.sql que vous trouverez sur mon site 😊

V) Requêtage simple (15 min)

4. Quels clients (leur id) n'ont pas passé de commande ?
5. Quelle est le montant de la commande la plus chère ?
6. Quel est le prix moyen d'une commande ?
7. Quels sont les prénoms des clients habitant Ajaccio ?
8. Afficher les montants et les dates (triées par ordre décroissant donc de la plus récente à la plus ancienne) des commandes passées par le client 1 ?
9. Quelle est la somme des montants de toutes les commandes ?

VI) Requêtage avec jointures (35 min)

10. Quelle est le nom du client dont la commande a pour montant 1500€ ?
11. Quelle est la ville du client qui a commandé le 01/03/2020 ?
12. Quels sont les id des commandes passées par Maguy ?
13. Combien a-t-elle passé de commandes ?
14. Quels sont les prénoms et les noms des clients qui n'ont jamais commandé ?
15. Combien d'argent a dépensé Amhed ?
16. À quelle(s) date(s) a commandé Jean ?

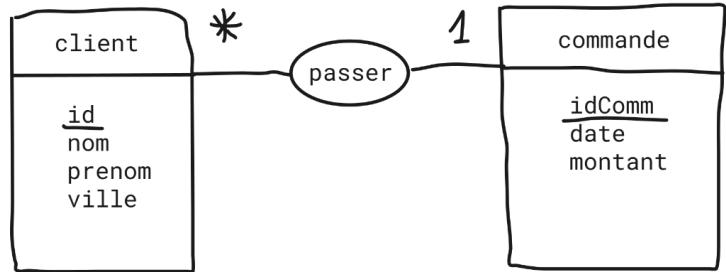
VII) Requêtage avancé (sympa mais hors programme)

17. Afficher l'id, le prénom, le nom et la somme totale des commandes de chaque client ayant commandé

Indication : *On ne peut pas s'en sortir si on n'utilise pas la clause GROUP BY*

Le MCD : correction

On identifie facilement le fait qu'il y a 2 tables en jeu : Client et Commande. Un client peut passer plusieurs commandes donc la cardinalité ici est * (ou aucune, mais 1...* aurait été juste aussi, cela symboliserait le fait qu'il ne peut créer de compte que s'il passe une commande !), une commande n'est passée que par un et un seul client. Veillez à bien indiquer les clefs primaires !



Le MLD : correction

On a une cardinalité « many-to-one » entre nos 2 tables, ce qui provoque une migration de clef vers la table qui porte le « many » (ici, client) vers la table qui porte le « one » (ici, commande).

```
client(id, nom, prenom, ville)
commande(idComm, date, montant, #id)
```

Concernant le domaine des attributs (leur type), bien que cela ne soit pas demandé, on pourrait avoir :

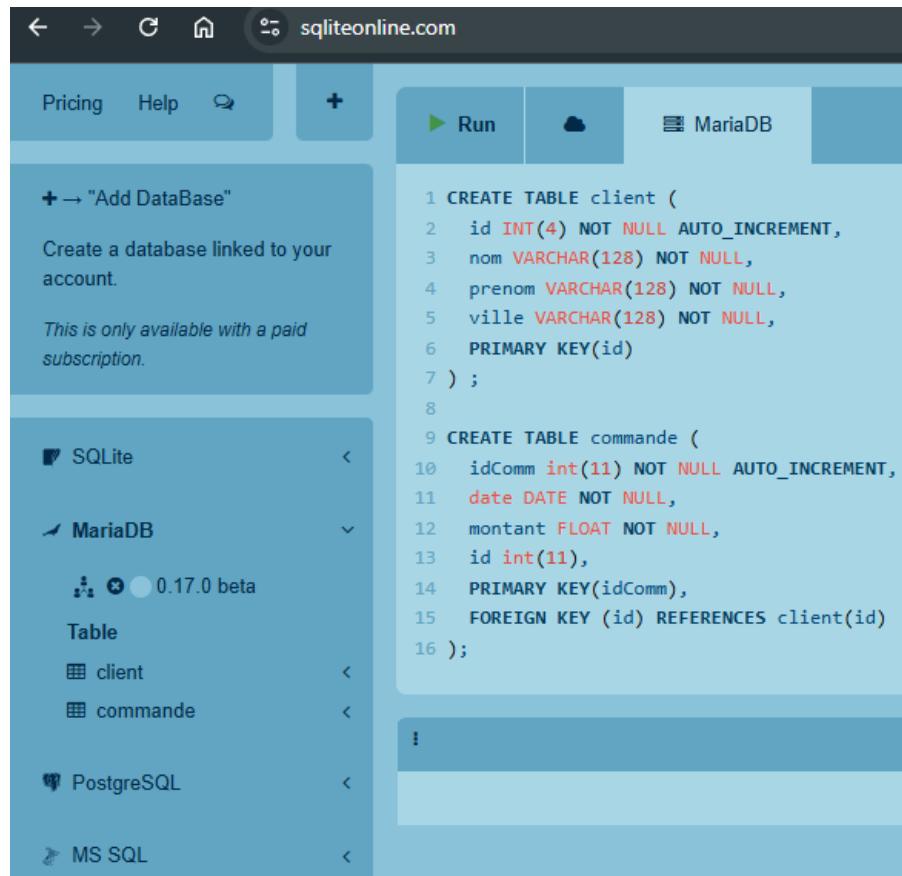
id, idComm : INT
nom, prenom, ville : VARCHAR(128)
date : DATE
montant : FLOAT

Le MPD : correction

```
CREATE TABLE client (
    id int(4) NOT NULL AUTO_INCREMENT,
    nom varchar(128) NOT NULL,
    prenom varchar(128) NOT NULL,
    ville varchar(128) NOT NULL,
    PRIMARY KEY(id)
) ;
```

```
CREATE TABLE commande (
    idComm int(11) NOT NULL AUTO_INCREMENT,
    datecommande date NOT NULL,
    id int(4) NOT NULL,
    PRIMARY KEY (idComm),
    FOREIGN KEY (id) REFERENCES client(id)
) ;
```

On va sur sqliteonline.com, on clique sur MariaDB, si la table demo est présente on clique droit et on fait DROP, ensuite on entre notre requête de création (les points-virgule permettent de combiner des requêtes) et on vérifie que les tables ont bien été créées. En cas d'erreur on peut les DROP et recommencer. Pour le moment, elles sont vides !



The screenshot shows the sqliteonline.com web interface. On the left, a sidebar lists databases: SQLite (selected), MariaDB (0.17.0 beta), PostgreSQL, and MS SQL. The main area displays SQL code for creating two tables: 'client' and 'commande'. The 'client' table has columns: id (INT(4) NOT NULL AUTO_INCREMENT), nom (VARCHAR(128) NOT NULL), prenom (VARCHAR(128) NOT NULL), and ville (VARCHAR(128) NOT NULL), with 'id' as the PRIMARY KEY. The 'commande' table has columns: idComm (int(11) NOT NULL AUTO_INCREMENT), date (DATE NOT NULL), montant (FLOAT NOT NULL), and id (int(11)), with 'idComm' as the PRIMARY KEY and a FOREIGN KEY reference to 'client(id)'. A 'Run' button is visible at the top right.

```
1 CREATE TABLE client (
2   id INT(4) NOT NULL AUTO_INCREMENT,
3   nom VARCHAR(128) NOT NULL,
4   prenom VARCHAR(128) NOT NULL,
5   ville VARCHAR(128) NOT NULL,
6   PRIMARY KEY(id)
7 );
8
9 CREATE TABLE commande (
10   idComm int(11) NOT NULL AUTO_INCREMENT,
11   date DATE NOT NULL,
12   montant FLOAT NOT NULL,
13   id int(11),
14   PRIMARY KEY(idComm),
15   FOREIGN KEY (id) REFERENCES client(id)
16 );
```

Insertion de « dummy data » : correction

Pas grand-chose à dire : il suffit de recopier et/ou d'utiliser mon fichier !

Requête simple

Requête avec jointures