

1.3

Le langage SQL

NSI TLE - JB DUTHOIT

Nous avons étudié la structure d'une base de données relationnelle, et nous allons maintenant apprendre à :

- Créer une base de données
- créer des attributs
- Ajouter des données
- Modifier des données
- Interroger une base de données afin d'obtenir les informations souhaitées

Pour cela, il nous faut apprendre un langage de requêtes : *SQL* : Structured Query Language.

Nous allons utiliser le logiciel "*DB Browser for SQLite*" : <https://sqlitebrowser.org/>.

Définition 1.4

SQLite est une bibliothèque écrite en langage C qui propose un moteur de base de données relationnelle accessible par le langage SQL

Remarque

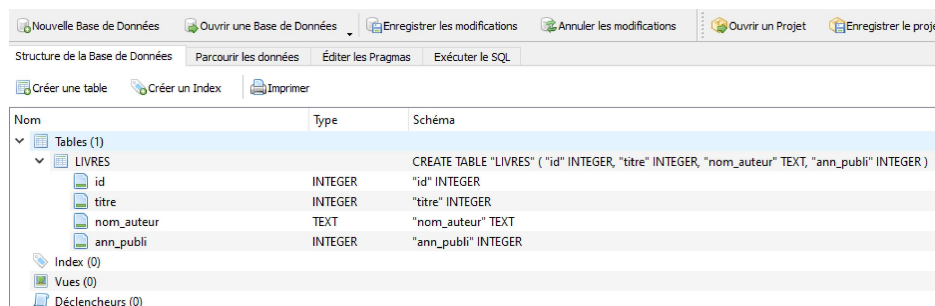
Noter qu'il existe d'autres systèmes de gestion de base de données relationnelle comme MySQL ou PostgreSQL. Dans tous les cas, le langage de requête utilisé est le SQL.

⚠ Attention, on peut parfois noter quelques petites différences de syntaxe entre les différentes SGBD).

1.3.1 Création d'une base de données

1. lancez le logiciel "DB Browser for SQLite"
2. Cliquez sur Nouvelle base de données.
3. Choisissez un nom pour cette base de données
4. Cliquez sur Annuler

La base de données est créée !



1.3.2 Création, suppression et modification d'une table mot-clé CREATE TABLE

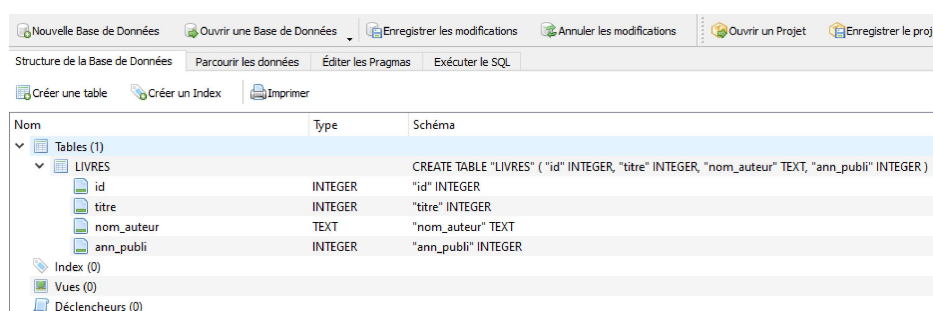
Création d'une table

1. Cliquez sur l'onglet "Exécuter le SQL"
2. Recopiez le texte suivant dans la fenêtre "SQL 1"

```
CREATE TABLE LIVRES
(id INT, titre TEXT, nom_auteur TEXT, ann_publi INT);
```

3. Cliquez ensuite sur "Lecture" (petit triangle ou F5)

Un message confirme la création de la table



Nous avons pour chaque attribut précisé son domaine : id : entier (INT), titre : chaîne de caractères (TEXT), nom__auteur : TEXT : ann__publi : INT

L'attribut "id" va jouer ici le rôle de clé primaire. On peut aussi, par souci de sécurité (afin d'éviter que l'on utilise 2 fois la même valeur pour l'attribut "id"), modifier le l'instruction SQL vue ci-dessus, afin de préciser que l'attribut "id" est bien notre clé primaire :

```
CREATE TABLE LIVRE
(id INT, titre TEXT, nom_auteur TEXT, PRIMARY KEY (id));
```

Modification d'une table

- Pour ajouter un attribut :

```
ALTER TABLE AUTEURS ADD email INT
```

- Pour modifier un attribut :

```
ALTER TABLE AUTEURS MODIFY email TEXT
```

- Pour changer le nom d'un attribut :

```
ALTER TABLE AUTEURS CHANGE email mail
```

- Pour supprimer un attribut :

```
ALTER TABLE AUTEUR DROP mail
```

Suppression d'une table

DROP TABLE PAYS

⚠ A utiliser avec beaucoup d'attention, car après les données sont perdues définitivement.

1.3.3 Ajout de données avec le mot-clé INSERT INTO


1. Cliquez sur "Exécuter le SQL"
2. Entrez le texte :

```
INSERT INTO LIVRES
(id, titre, nom_auteur, ann_publi)
VALUES
(1,"Les fleurs du mal", "Baudelaire", 1857),
(2,"L'étranger", "Camus", 1942),
(3,"Les misérables", "Hugo", 1862),
(4,"Les liaisons dangereuses", "Laclos", 1782),
(5,"Le petit prince", "Saint-Exupéry", 1943),
(6,"Cyrano de Bergerac", "Rostand", 1897),
(7,"Les trois mousquetaires", "Dumas", 1844),
(8,"Le comte de Monte-Cristo", "Dumas", 1844),
(9,"La peste", "Camus", 1947),
(10,"Notre-Dame de Paris", "Hugo", 1831),
(11,"Orgueil et préjugés", "Austen",1813),
(12,"Oliver Twist", "Dickens", 1839),
(13,"Le Portrait de Dorian Gray", "Wilde", 1890),
(14,"L'Ami retrouvé", "Uhlman", 1971),
(15,"Gatsby le Magnifique", "Fitzgerald", 1925);
```

Remarque

- Un message nous indique une nouvelle fois que la requête a été réalisée avec succès.

liste des enregistrements



The screenshot shows a database application interface. At the top, there are buttons for 'Nouvelle Base de Données', 'Ouvrir une Base de Données', and 'Enregistrer'. Below these are tabs for 'Structure de la Base de Données', 'Parcourir les données' (which is selected), and 'Éditer les Pragma'. A dropdown menu shows 'Table : LIVRES'. Below the menu is a table with 15 rows and 4 columns: 'id', 'titre', 'nom_auteur', and 'ann_publi'. Each column has a 'Filtre' button next to it. The table contains the following data:

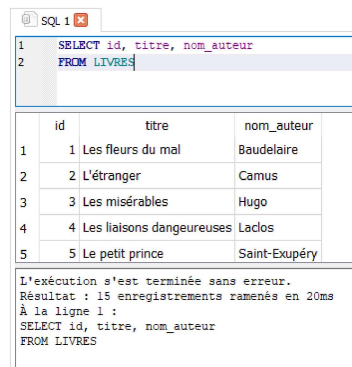
	id	titre	nom_auteur	ann_publi
1	1	Les fleurs du mal	Baudelaire	1857
2	2	L'étranger	Camus	1942
3	3	Les misérables	Hugo	1862
4	4	Les liaisons dangereuses	Laclos	1782
5	5	Le petit prince	Saint-Exupéry	1943
6	6	Cyrano de Bergerac	Rostand	1897
7	7	Les trois mousquetaires	Dumas	1844
8	8	Le comte de Monte-Cristo	Dumas	1844
9	9	La peste	Camus	1947
10	10	Notre-Dame de Paris	Hugo	1831
11	11	Orgueil et préjugés	Austen	1813
12	12	Oliver Twist	Dickens	1839
13	13	Le Portrait de Dorian Gray	Wilde	1890
14	14	L'Ami retrouvé	Uhlman	1971
15	15	Gatsby le Magnifique	Fitzgerald	1925

1.3.4 Les requêtes

Liste des enregistrements avec le mot-clé SELECT

```
SELECT id,titre, nom_auteur
FROM LIVRES
```

liste des enregistrements



SQL 1

```
1 SELECT id, titre, nom_auteur
2 FROM LIVRES
```

	id	titre	nom_auteur
1	1	Les fleurs du mal	Baudelaire
2	2	L'étranger	Camus
3	3	Les misérables	Hugo
4	4	Les liaisons dangereuses	Laclos
5	5	Le petit prince	Saint-Exupéry

L'exécution s'est terminée sans erreur.
 Résultat : 5 enregistrements ramenés en 20ms
 À la ligne 1 :
 SELECT id, titre, nom_auteur
 FROM LIVRES

Remarque

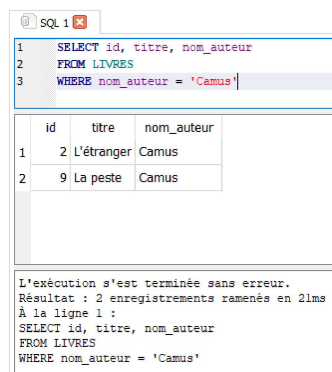
Il est possible de saisir :

```
SELECT *
FROM LIVRES
```

pour obtenir tous les attributs !

le mot-clé WHERE

liste des enregistrements



SQL 1

```
1 SELECT id, titre, nom_auteur
2 FROM LIVRES
3 WHERE nom_auteur = 'Camus'
```

	id	titre	nom_auteur
1	2	L'étranger	Camus
2	9	La peste	Camus

L'exécution s'est terminée sans erreur.
 Résultat : 2 enregistrements ramenés en 21ms
 À la ligne 1 :
 SELECT id, titre, nom_auteur
 FROM LIVRES
 WHERE nom_auteur = 'Camus'

Remarque

Il est possible de combiner les conditions à l'aide d'un OR ou d'un AND

Savoir-Faire 1.2

Écrire une requête qui permet d'obtenir la liste des livres qui ont été écrits après 1950 ou par Camus.

	id	titre	nom_auteur	ann_publi
1	2	L'étranger	Camus	1942
2	9	La peste	Camus	1947
3	14	L'Ami retrouvé	Uhlman	1971

le mot-clé ORDER BY

Il est aussi possible de rajouter la clause ORDER BY afin d'obtenir les résultats classés dans un ordre précis.

```
SELECT titre
FROM LIVRES
WHERE ann_publi >= 1900 ORDER BY nom_auteur
```

Savoir-Faire 1.3

Écrire une requête qui permet d'obtenir la liste "titre, nom__ auteur", liste rangée par ordre chronologique de publication, pour les livres écrits après 1900.

Remarque

🔍 Il est possible d'obtenir un classement en sens inverse à l'aide de la clause DESC

```
SELECT titre
FROM LIVRES
WHERE ann_publi >= 1900 ORDER BY nom_auteur DESC
```

mot-clé DISTINCT

Si on effectue :

```
SELECT nom_auteur
FROM LIVRES
```

On se retrouve avec des noms d'auteur en double.
Il suffit alors d'utiliser DISTINCT :

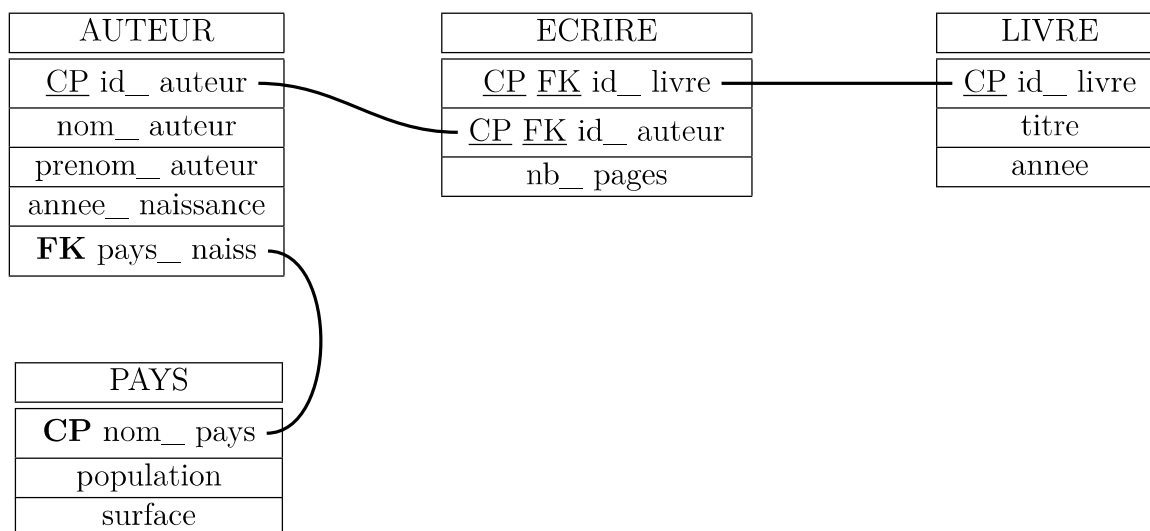
```
SELECT DISTINCT nom_auteur
FROM LIVRES
```

1.3.5 Lier plusieurs tables d'une même base de données

Création des tables avec les différentes clés

Créer une nouvelle base de données nommée "Exemple1".

Créer les tables suivantes :



On considère les livres suivants afin de compléter notre base de données :

"Les fleurs du mal", "L'étranger", "Les misérables", "Les liaisons dangereuses", "Le petit prince", "Cyrano de Bergerac", "Les trois mousquetaires", "Le comte de Monte-Cristo", "La peste", "Orgueil et préjugés", "Oliver Twist", "Le Portrait de Dorian Gray", "L'Ami retrouvé", "Gatsby le Magnifique", "Le Cercle littéraire des amateurs d'épluchures de patates".

Exercice 1.5

Compléter les instruction SQL quivantes :

```
INSERT INTO LIVRE
VALUES
```

```
(1,"Les fleurs du mal", 1857),
(2,"L'étranger", 1942),...
```

...

```
INSERT INTO AUTEUR
```

```
(id_auteur, nom_auteur, prenom_auteur, annee_naiss, pays_naiss)
```

```
VALUES
```

```
(1,'Baudelaire', 'Charles',1821, 'France'), (2,'Camus', 'Albert', 1913, 'Algérie'),
```

...

```
INSERT INTO ECRIRE
```

```
(id_livre,id_auteur, nb_page)
```

```
VALUES
```

```
(1,1,100),
```

```
(2,2,100),
```

...

(on indiquera de façon arbitraire un nb non nul de pages)

```
INSERT INTO PAYS
```

```
(nom_pays, population, superficie)
```

```
VALUES
```

```
('France', 67 , 643),
```

```
('Irlande', 5, 70),
```

On exprimera la population en million d'habitants et la superficie en milliers de km^2

Les jointures

Observez bien cet exemple qui permet de joindre les différentes tables et d'afficher le nom, prénom et année de naissance des auteurs nés en France

```

SELECT nom_auteur, prenom_auteur, annee_naiss
FROM LIVRES
INNER JOIN ECRIRE ON ECRIRE.id_livre = LIVRES.id_livre
INNER JOIN AUTEUR ON ECRIRE.id_auteur = AUTEUR.id_auteur
INNER JOIN PAYS ON PAYS.nom_pays = AUTEUR.pays_naiss
WHERE pays_naiss = "France"

```

	nom_auteur	prenom_auteur	annee_naiss
1	Baudelaire	Charles	1821
2	Hugo	Victor	1802
3	Laclos	Choderlos de	1741
4	Saint-Exupéry	Antoine de	1900
5	Rostand	Edmond	1868
6	Dumas	Alexandre	1802
7	Dumas	Alexandre	1802
8	Hugo	Victor	1802

Remarque

On peut spécifier une contrainte de suppression et de mise à jour sur la clé étrangère avec respectivement `ON DELETE CASCADE` et `ON UPDATE CASCADE`. Cela signifie que si on supprime des occurrences dans la relation vers laquelle "pointe" la clé étrangère, on supprime aussi les lignes en référence dans la relation où se trouve la clé étrangère. Si on modifie la valeur de l'attribut "pointé" par la clé étrangère, on modifie su coup la ligne correspondante dans la relation où se trouve la clé étrangère.

1.3.6 Les fonctions d'agrégation

Les fonctions d'agrégation dans le langage SQL permettent d'effectuer des opérations statistiques sur une colonne. Les principales fonctions sont :

sum : pour calculer la somme d'un attribut

min : pour récupérer la valeur minimale

max : pour récupérer la valeur maximale

avg : pour calculer la moyenne

count : pour compter le nombre d'enregistrements

1.3.7 Requête de mise à jour

INSERT INTO

INSERT INTO permet d'ajouter une ou plusieurs données.

DELETE FROM

DELETE permet de supprimer des lignes dans une relation. On peut utiliser la clause WHERE pour sélectionner les ligne à supprimer :

```
DELETE FROM CLIENTS
WHERE id\_client <= 3
```

⚠ Il est préférable de sauvegarder la base de données avant de supprimer des lignes. De cette façon, un retour en arrière sera toujours possible!

⚠ Si la clause WHERE n'est pas présente, toutes les lignes seront effacées!!

Modification de données avec UPDATE

La commande UPDATE permet d'effectuer des modifications sur des lignes existantes. Exemple :

```
UPDATE CLIENTS SET solde = 11724
WHERE prenom = 'John'
```

Il est possible également de réaliser ceci :

```
UPDATE CLIENTS SET nom = UPPER(nom), prix = prix * 1.1
```

1.4**Un autre exemple !**

NSI TLE - JB DUTHOIT

On considère la petite base de donnée suivante, constituée de 4 relations :

Relation **STATIONS**

nom_station	capacité	lieu	region	tarif
Tanger	350	maroc	Afrique	1200
La Bourboule	250	Auvergne	Europe	700
Victoria	200	Seychelles	Océan Indien	1500
Courchevel	400	Alpes	Europe	2200

Relation **ACTIVITES**

id_activite	nom_station	libelle	prix
1	La Bourboule	Pêche	50
2	La Bourboule	Randonnée	0
3	Tanger	Plongée	120
4	Tanger	Excursion	0
5	Victoria	Plongée	130
6	Courchevel	Ski	1200

Relation **CLIENTS**

id__client	nom	prenom	ville	region	solde
1	Bolle	Pierre	Paris	Europe	9825
2	Jonhson	Britney	New York	Amérique	6721
3	Smith	John	Londres	Europe	12436

Relation **SEJOURS**

id__sejour	id__client	station	arrivee	nb__places
1	1	Courchevel	17/02/2019	2
2	3	Tanger	17/11/2018	5
3	2	Courchevel	25/12/2019	4
4	3	La Bourboule	20/08/2019	6
5	3	Victoria	13/04/2016	3
6	2	La Bourboule	27/07/2018	5
7	1	Victoria	05/09/2018	3

Exercice 1.6

Créez les tables dans DB browser for SQLite. Ne pas oublier les clefs primaires et les clefs étrangères.***

Exercice 1.7

Afficher le nom de la station et le lieu de toutes les stations d'Europe :

	nom_station	lieu
1	Courchevel	Alpes
2	La Bourboule	Auvergne

Exercice 1.8

Affichez toutes les activités , sans doublon :

	libelle
1	Pêche
2	Randonnée
3	Plongée
4	Excursion
5	ski

Exercice 1.9

Afficher la liste des stations suivant l'ordre croissant des tarifs respectifs :

	nom_station
1	La Bourboule
2	Tanger
3	Victoria
4	Courchevel

Exercice 1.10

Afficher les noms de clients qui commencent par un "B".***

Remarque

On peut utiliser l'opérateur LIKE dans la clause WHERE :

LIKE '% A' : Recherche de toutes les valeurs de l'attribut qui se termine par le caractère "A"

LIKE 'A %' : Recherche de toutes les valeurs de l'attribut qui commencent par le caractère "A"

LIKE '% a %' : Recherche de toutes les valeurs de l'attribut qui contiennent par le caractère "a"

Exercice 1.11

Afficher les tuples (nom du client, station visité) , rangés par ordre alphabétique des noms de clients :

	nom	station
1	Bolle	Courchevel
2	Bolle	Victoria
3	Jonhson	Courchevel
4	Jonhson	La Bourboule
5	Smith	Tanger
6	Smith	La Bourboule
7	Smith	Victoria

Exercice 1.12

Afficher le tableau suivant :

	nom_station	lieu	region	libelle	prix
1	Tanger	Maroc	Afrique	Excursion	0
2	La Bourboule	Auvergne	Europe	Randonnée	0
3	La Bourboule	Auvergne	Europe	Pêche	50
4	Courchevel	Alpes	Europe	ski	120
5	Tanger	Maroc	Afrique	Plongée	120
6	Victoria	Seychelles	Océan indien	Plongée	130

Exercice 1.13

Lister les régions où ont séjourné les clients :

	nom	region
1	Bolle	Europe
2	Bolle	Océan indien
3	Jonhson	Europe
4	Smith	Afrique
5	Smith	Europe
6	Smith	Océan indien

Essayez de réaliser le tableau suivant, obtenu en concaténant le nom et le prénom du client :

	client	region
1	Pierre BOLLE	Europe
2	Pierre BOLLE	Océan indien
3	Brithney JONHSON	Europe
4	John SMITH	Afrique
5	John SMITH	Europe
6	John SMITH	Océan indien

Exercice 1.14

Afficher le minimum, le maximum et la moyenne des tarifs :

	min	max	moyenne
1	700	2200	1400.0

Exercice 1.15

Afficher le nombre de place que Monsieur Smith a réservé :

	sum(nb_place)	
1		14

Exercice 1.16

1. Donner le nom des clients qui sont allés à La Bourboule
2. Donner les stations ayant strictement plus de 200 places
3. Donner le nom des stations qui proposent la plongée
4. Donner le prix moyen d'une activité à Tanger
5. Combien de séjours ont eu lieu à Victoria ?
6. Donner les stations visitées par des européens.
7. Donner l'expression SQL permettant d'afficher la liste des stations suivi du lieu (en majuscules) entre parenthèses, et du tarif HT et TTC comme le montre l'exemple suivant (On supposera que le prix affiché est HT et on prendra une TVA à 20%) :

	Stations	tarif HT	Tarif TTC
1	Tanger (MAROC)	1200	1440.0
2	La Bourboule (AUVERGNE)	700	840.0
3	Victoria (SEYCHELLES)	1500	1800.0
4	Courchevel (ALPES)	2200	2640.0

8. Donner l'expression SQL permettant d'ajouter la cliente venant de Toronto (Canada) suivante : mme Karibou Juliette avec un solde de 7213 euros. Cette cliente a séjourné (3 places) à La Bourboule le 10/07/2019.
9. Donner l'expression SQL permettant de mettre à jour la capacité de la station Courchevel à 450 places ainsi que le nouveau tarif de 2300.
10. Donner l'expression SQL permettant de supprimer tout ce qui concerne Mme Karibou.

Remarque

il faut pour cela avoir bien spécifié ON DELETE CASCADE et ON UPDATE CASCADE pour chaque clé étrangère créée.
