

Comme souvent en informatique, la pratique permet de mieux comprendre les différentes fonctions; si vous souhaitez mettre en pratique les notions présentées dans ce chapitre, vous pouvez télécharger le logiciel DB Browser for SQLite. Parfois un peu limité (par exemple il ne gère pas tous les formats de données, comme les dates) il a l'avantage d'être très simple d'utilisation.

A. Introduction et vocabulaire

A.1. Généralités

Une base de données est un ensemble structuré permettant d'organiser, de stocker, de mettre à jour et d'interroger des données structurées volumineuses qui peuvent être utilisées par un ou plusieurs programmes ou utilisateurs.

Une base de donnée peut contenir plusieurs tables; chacune de ces tables permettant de stocker les données d'un sujet spécifique.

On peut par exemple disposer d'une base de données contenant une table nommée `Biblio` qui regroupe les livres disponibles dans une librairie :

ISBN	Auteur	Titre	Éditeur	Nbre_pages	Prix
2253007102	Charles Baudelaire	Les Fleurs du Mal	Le livre de poche	374	13
2072898781	Oscar Wilde	Le Portrait de Dorian Gray	Folio	416	4.60
2073042708	John Steinbeck	Des souris et des hommes	Folio	304	10
2073043992	John Steinbeck	Les Raisins de la Colère	Folio	672	11.10
2253082503	Féodor Dostoïevski	Crime et Châtiment	Le livre de poche	704	8.90

une table nommée `Clients` qui regroupe la liste des clients :

N_client	Nom_Client	Prénom_Client
1	Poisson	Denis
2	Pascal	Blaise
...

et une table nommée `Ventes` qui regroupe la liste des livres vendus :

N_Vente	N_Client	Achat	Prix_Achat	Date
1	4	2072898781	4.60	12/02/2023
2	1	2073042708	10	13/04/2024
3	2	2073042708	10	11/05/2024

Étant donnée une table dans une base de donnée :

- un **enregistrement** est une ligne du tableau contenant toutes les informations relatives à un même objet et appelés **attributs** (dans l'exemple, un enregistrement contient par exemple toutes les informations disponibles sur un livre donné),
- un **champ** est une case de la table,
- à chaque **colonne** est associée à un type de données appelé **domaine** (entier, flottants chaîne de caractère, date ou type binaire),

- on appelle **schéma de table** la donnée du nom de la table et de la liste des attributs ainsi que de leurs domaines ; dans l'exemple, le schéma de la table est donc :

```
Biblio (
  ISBN INTEGER,
  Auteur CHAR,
  Titre CHAR,
  Éditeur CHAR,
  Nbre_pages INTEGER,
  Prix FLOAT
)
```

A.2. Clé primaire, clé étrangère

Une base de donnée bien organisée doit permettre d'identifier chaque enregistrement de manière unique (par exemple un auteur peut avoir écrit plusieurs livres, ou plusieurs livres peuvent avoir le même titre) et éventuellement de lier les tables entre elles (un même client peut par exemple avoir acheté plusieurs livres), c'est pourquoi certaines colonnes jouent un rôle particulier.

- On appelle **clé primaire** (ou primary key) un attribut qui permet d'identifier un enregistrement d'une table de manière unique ; par exemple, dans la table `Biblio`, la colonne `ISBN` contient une clé primaire permettant d'identifier chaque livre de manière unique tandis que la colonne `N_Client` est une clé primaire dans la base `Clients`.

Lorsque la clé primaire a uniquement pour vocation de distinguer les enregistrements mais que la valeur en elle-même n'importe pas, il est possible de faire en sorte qu'elle soit saisie automatiquement en ajoutant le mot-clé `AUTOINCREMENT` après le type de valeurs.

- On appelle **clé étrangère** (ou foreign key) un attribut qui permet de lier des enregistrements de deux tables différentes entre eux ; par exemple, dans la table `Ventes`, la colonne `N_Client` est une clé étrangère permettant d'associer une vente donnée à un client (en évitant donc de saisir plusieurs fois les mêmes informations) tandis que la colonne `Achat` est une clé étrangère qui permet, pour un achat donné effectué par un client, de retrouver les caractéristiques du livre acheté.

Une clé étrangère doit être déclarée à la création de la table en spécifiant à quelle clé primaire elle fait référence ; par exemple le schéma de la table `Ventes` sera le suivant :

```
Ventes (
  N_Vente INTEGER,
  N_Client INTEGER,
  Achat INTEGER,
  Date DATE
  FOREIGN KEY(Achat) REFERENCES Biblio(ISBN)
)
```

B. Création et modification d'une base de données

Il y a plusieurs façons de créer une base de données (utiliser un logiciel dédié, ou utiliser un langage de programmation par exemple). Les instructions étudiées ici sont dans le langage SQL, qui est un langage standard pour manipuler des données.

Quand on manipule des tables, il n'y a pas de règles quant à l'utilisation de minuscules ou de majuscules pour les mots-clés car le programme ne les différencie pas, mais l'usage est d'utiliser des majuscules pour identifier les mots-clés.

B.1. Création d'une base de donnée

Pour créer une table dans une base de donnée, on saisit le mot-clé

```
CREATE TABLE
```

suivi du schéma de la table.

Par exemple, les tables Biblio et Ventes ont été définies avec les instructions suivantes :

```
CREATE TABLE Biblio (
  ISBN INTEGER PRIMARY KEY,
  Auteur CHAR,
  Titre CHAR,
  Editeur CHAR,
  Nbre_pages INTEGER,
  Prix FLOAT,
  PRIMARY KEY (ISBN)
)

CREATE TABLE Ventes (
  N_Vente INTEGER,
  N_Client INTEGER,
  Achat INTEGER,
  Date DATE,
  PRIMARY KEY N_Vente AUTOINCREMENT,
  FOREIGN KEY (Achat) REFERENCES Biblio(ISBN),
)
```

B.2. Ajout d'enregistrements à une table

Pour ajouter un enregistrement à une table, le mot clé est

```
INSERT INTO
```

suivi du nom de la table et de la liste des attributs. Par exemple, si l'on exécute les instructions

```
INSERT INTO Biblio
VALUES (2253194611, 'Pierre Lemaitre', 'Au revoir là-haut', 'Le Livre de Poche', 624, 9.9)
```

la table Biblio devient :

ISBN	Auteur	Titre	Éditeur	Nbre_pages	Prix
2253007102	Charles Baudelaire	Les Fleurs du Mal	Le livre de poche	374	13
2072898781	Oscar Wilde	Le Portrait de Dorian Gray	Folio	416	4.60
2073042708	John Steinbeck	Des souris et des hommes	Folio	304	10
2073043992	John Steinbeck	Les Raisins de la Colère	Folio	672	11.10
2253082503	Fédor Dostoïevski	Crime et Châtiment	Le livre de poche	704	8.90
2253194611	Pierre Lemaitre	Au revoir là-haut	Le Livre de Poche	624	9.9

B.3. Suppression d'enregistrements dans une table

Pour supprimer un ou plusieurs enregistrements d'une table, le mot clé est

```
DELETE FROM
```

suivi du nom de la table et de la liste des conditions pour identifier les enregistrements à supprimer. Par exemple, si l'on exécute les instructions

```
DELETE FROM Biblio
WHERE Auteur="Oscar Wilde"
```

la table Biblio devient :

ISBN	Auteur	Titre	Éditeur	Nbre_pages	Prix
2253007102	Charles Baudelaire	Les Fleurs du Mal	Le livre de poche	374	13
2073042708	John Steinbeck	Des souris et des hommes	Folio	304	10
2073043992	John Steinbeck	Les Raisins de la Colère	Folio	672	11.10
2253082503	Fédor Dostoïevski	Crime et Châtiment	Le livre de poche	704	8.90
2253194611	Pierre Lemaitre	Au revoir là-haut	Le Livre de Poche	624	9.9

On retiendra également que :

- L'instruction DELETE FROM suivie du nom de la table mais sans aucune autre précision efface toutes les données de la table (mais ne supprime pas la table de la base de données);
- La condition peut utiliser les opérateurs de comparaison usuels : =, <, >, <=, >=

B.4. Modification d'enregistrements dans une table

Pour modifier un ou plusieurs enregistrements d'une table, le mot clé est

UPDATE

suivi du nom de la table et de la liste des champs à modifier et des conditions éventuelles pour identifier les enregistrements à modifier. Par exemple, si l'on exécute les instructions

```
UPDATE Biblio
SET Prix=Prix+1
```

tous les prix sont augmentés de 1 et la table Biblio devient :

ISBN	Auteur	Titre	Éditeur	Nbre_pages	Prix
2253007102	Charles Baudelaire	Les Fleurs du Mal	Le livre de poche	374	14
2073042708	John Steinbeck	Des souris et des hommes	Folio	304	11
2073043992	John Steinbeck	Les Raisins de la Colère	Folio	672	12.10
2253082503	Fédor Dostoïevski	Crime et Châtiment	Le livre de poche	704	9.90
2253194611	Pierre Lemaitre	Au revoir là-haut	Le Livre de Poche	624	10.9

Si l'on exécute les instructions

```
UPDATE Biblio
SET Prix=Prix+1
WHERE Auteur="Pierre Lemaitre"
```

seuls le prix des livres dont l'auteur est Pierre Lemaitre changent et la table Biblio devient :

ISBN	Auteur	Titre	Éditeur	Nbre_pages	Prix
2253007102	Charles Baudelaire	Les Fleurs du Mal	Le livre de poche	374	14
2073042708	John Steinbeck	Des souris et des hommes	Folio	304	11
2073043992	John Steinbeck	Les Raisins de la Colère	Folio	672	12.10
2253082503	Fédor Dostoïevski	Crime et Châtiment	Le livre de poche	704	9.90
2253194611	Pierre Lemaitre	Au revoir là-haut	Le Livre de Poche	624	10.9

C. Sélectionner les données disponibles dans une ou plusieurs tables

Pour sélectionner (et afficher) une partie des données seulement, le mot-clé est :

SELECT

C.1. Sélection basique d'une ou plusieurs colonnes

Si l'on exécute l'instruction

```
SELECT nom_champ1, nom_champ2 FROM nom_table
```

on obtient la liste de toutes les valeurs dans les colonnes nom_champ1 et nom_champ2 de la table nom_table. Par exemple, si l'on exécute la commande

```
SELECT Titre,Auteur FROM Biblio
```

la table Biblio est inchangée mais on obtient la liste suivante :

Titre	Auteur
Les Fleurs du Mal	Charles Baudelaire
Des souris et des hommes	John Steinbeck
Les Raisins de la Colère	John Steinbeck
Crime et Châtiment	Fédor Dostoïevski
Au revoir là-haut	Pierre Lemaitre

C.2. Sélection de toutes les données du tableau

Pour sélectionner toute une table d'une base de données, on peut utiliser la commande

```
SELECT * FROM nom_table
```

C.3. Sélection de certains enregistrements uniquement

Pour sélectionner toute une table d'une base de données, on peut utiliser la commande

```
SELECT FROM nom_table WHERE condition
```

Par exemple, si on exécute les instructions suivantes :

```
SELECT *
FROM Biblio
WHERE Auteur='Charles Baudelaire' OR Prix>10
```

tous les enregistrements de la table `Biblio` dont l'auteur est Charles Baudelaire ou dont le prix est strictement supérieur à 10 seront renvoyés, et on obtient ainsi la liste suivante :

ISBN	Auteur	Titre	Éditeur	Nbre_pages	Prix
2253007102	Charles Baudelaire	Les Fleurs du Mal	Le livre de poche	374	13
2073043992	John Steinbeck	Les Raisins de la Colère	Folio	672	11.10

C.4. Jointure de tables

Lorsque deux tables ont un attribut en commun, il est possible de regrouper les données des deux tables en effectuant une jointure, utilisant le mot clé

```
INNER JOIN
```

Par exemple, si on exécute les instructions suivantes :

```
SELECT Auteur, Titre, N_Client, Date
FROM Biblio
INNER JOIN Ventes ON Biblio.ISBN=Ventes.Achat
```

toutes les données de la table `Biblio` correspondant à une donnée de la table `Vente` (en résumé, tous les livres ayant été vendus au moins une fois) seront listées et les colonnes seront agrégées, et on obtient ainsi la liste suivante :

Auteur	Titre	N_Client	Date
John Steinbeck	Des souris et des hommes	1	13/04/2024
John Steinbeck	Des souris et des hommes	2	11/05/2024

D. Les commandes non exigibles des candidats

Les commandes qui suivent sont supposées vues par les étudiants mais ne sont pas exigibles aux concours, c'est pourquoi elles ne sont que peu détaillées ici (voir les exercices pour plus de détail).

D.1. Les commandes DISTINCT et ORDER BY

- La commande `ORDER BY` permet de trier les données d'un tableau ; par exemple les instructions `SELECT * FROM Biblio ORDER BY Auteur` renvoie la liste de tous les enregistrements (si on veut les trier dans l'ordre décroissant, on, ajoute `DESC` à la fin de l'instruction).
- La commande `DISTINCT` permet de sélectionner des données en évitant les doublons ; par exemple la commande `SELECT DISTINCT Auteur FROM Biblio` renverra la liste de tous les auteurs de la table `Biblio`, sans répétition.

D.2. Les fonctions d'agrégation

- MIN permet de renvoyer la valeur minimale d'une colonne; par exemple `SELECT MIN(Prix), Auteur FROM Biblio` renverra le prix minimal d'un livre ainsi que l'auteur correspondant dans la table Biblio.
- MAX permet de renvoyer la valeur maximale d'une colonne.
- SUM permet de renvoyer la somme des données d'une colonne.
- AVG permet de renvoyer la valeur moyenne d'une colonne.
- COUNT permet de compter des enregistrements; par exemple l'instruction `SELECT COUNT(Auteur) FROM Biblio` renvoie le nombre total de valeurs (non forcément distinctes) dans la colonne Auteur, tandis que `SELECT COUNT(DISTINCT Auteur) FROM Biblio` renvoie le nombre distincts de valeurs dans cette colonne.

D.3. Les opérations ensemblistes

- UNION permet d'obtenir la concaténation des résultats de deux requêtes; pour qu'elle fonctionne, il est néanmoins nécessaire que chacune des requêtes à concaténer retourne le même nombre de colonnes, avec les mêmes types de données et dans le même ordre; la syntaxe est la suivante :

```
SELECT colonnes
FROM table_1
UNION
SELECT
colonnes
FROM table_2
```

- INTERSECT renvoie les enregistrements communs à deux requêtes, ce qui peut être utile par exemple pour trouver les données similaires dans deux tables distinctes.
- EXCEPT, utilisé entre deux instructions, permet de renvoyer les enregistrements de la première instruction sans inclure les résultats de la seconde requête.



Sommaire

Bases de données	1
A. Introduction et vocabulaire	1
A.1. Généralités	1
A.2. Clé primaire, clé étrangère	2
B. Création et modification d'une base de données	2
B.1. Création d'une base de donnée	2
B.2. Ajout d'enregistrements à une table	3
B.3. Suppression d'enregistrements dans une table	3
B.4. Modification d'enregistrements dans une table	4
C. Sélectionner les données disponibles dans une ou plusieurs tables	4
C.1. Sélection basique d'une ou plusieurs colonnes	4
C.2. Sélection de toutes les données du tableau	5
C.3. Sélection de certains enregistrements uniquement	5
C.4. Jointure de tables	5
D. Les commandes non exigibles des candidats	5
D.1. Les commandes DISTINCT et ORDER BY	5
D.2. Les fonctions d'agrégation	6
D.3. Les opérations ensemblistes	6

