Table des matières

Exercices avec les bases relationnelles 3
Introduction 3
Création de tables 3
Exercice 1 4
Créez la table Livre0 4
Insérez des données 4
Exécutez des requêtes SELECT 5
Questions à propos du modèle relationnel 6
Exercice 2 7
Créez les tables 7
Insérez des lignes dans les tables 8
Requêtes avec jointures 9
Exercice 3 9

Exercices avec les bases relationnelles

Introduction

Ces exercices introduisent les notions suivantes :

- les formes simples de requêtes SELECT.
- clé primaire ;
- clé étrangère ;
- première forme normale (1FN) ;

Pour vous aider dans les exercices, consultez les tutoriaux SQL du site developpez.com. Et notamment :

- Le site de SQLite :
  - sa documentation
  - le langage SQL adapté à SQLite
- des définitions
- Jointures
- La normalisation (1FN, 2FN,..
- L'ensemble des tutoriel sur SQL et les SGBBDR

Des logiciels pour créer un dictionnaire des données, dessiner des MCD et générer automatiquement les modèles logiques, physiques et les ordres SQL de création des tables. Ces deux logiciels demandent que Java (machine virtuelle) soit installée sur votre poste :

- AnalyseSI : https://launchpad.net/analysesi/+download
- JMerise : La page d'accueil, La page du logiciel avec les consignes d'installation

Voici un lien vers un document PDF intéressant qui remet en perspective tout le vocabulaire abordé Lundi :

- du niveau conceptuel (à partir de la page 10) ;
- au niveau physique (page 2 à 8) ;
- en passant par le modèle relationnel, les dépendances fonctionnelles (page 34) et toute la théorie mathématique sous-jacente au niveau des langages de requêtes :
  - l'algèbre relationnelle (à partir de la page 41) ;
  - les formes normales (à partir de la page 41) et vous présente des exemples de requêtes SELECT.
- Ce lien (et les autres fichiers de l'auteur) vous présente un résumé très succint et pratique du modèle relationnel, des dépendances fonctionnelles et des formes normales.
- Ces deux derniers documents vous présentent des requêtes utilisant uniquement l'algèbre relationnelle et des notions sur UML et la conception des programmes informatiques et des bases de données en mettant en oeuvre le "paradigme objet".

Création de tables

Exercice 1

Voici le MCD correspondant :

Créez la table Livre0

`livre0(titre, auteur, editeur, annee_parution, isbn, prix)`

Colonne	Type
titre	varchar(100)
auteur	varchar(300)
editeur	varchar(100)
annee_parution	INTEGER (INT en SQLite)
prix	DOUBLE (REAL en SQLite)
isbn	varchar(20)

Vous pouvez recréer une base ou utiliser la base existante. <spoiler|Solution>

```
F:\>cd __2012_2013\_PREPAS\jeudi\sqlite3\
F:\__2012_2013\_PREPAS\jeudi\sqlite3>sqlite3 base_livres.sqlite3
SQLite version 3.7.17 2013-05-20 00:56:22
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
sqlite> .tables
sqlite>
```

```
CREATE TABLE livre0(titre VARCHAR(100) NOT NULL, isbn VARCHAR(100), auteur
VARCHAR(300), annee_parution INT, prix DOUBLE, editeur  VARCHAR(100));
```

```
sqlite> .tables
livre0
sqlite>
```

</spoiler>

Insérez des données

Utilisez un script SQL avec une vingtaine d'ouvrages d'éditeurs et d'auteurs différents mais aussi avec "doublons" ou "triplés" sur les dates, éditeurs, auteurs.

Quelques données: Design Patterns: Elements of Reusable Object-Oriented Software; Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides;1994;Addison-Wesley Professional;ISBN10 : 0201634988 UML 2 en action , De l'analyse des besoins à la conception ;Pascal Roques, Franck Vallée;2007;Eyrolles;ISBN10 : 2212121040 UML en action , De l'analyse des besoins à la conception;Pascal Roques, Franck Vallée;2000;Eyrolles;ISBN-10 : 2212091273 UML pour les

décideurs;Franck Vallée;2005;Eyrolles;ISBN-10 : 2212116217 <spoiler|Solution : >

```
INSERT INTO livre0 VALUES('UML 2 en action ,De l''analyse des besoins à la conception', '2212121040','Pascal Roques, Franck Vallée', 2007, 42.95, 'Eyrolles');
```

Lorsqu'on désire ne renseigner que quelques colonnes, on indique leurs noms à la suite du nom de la table

```
INSERT INTO livre0(titre,isbn) VALUES('Design Patterns: Elements of Reusable Object-Oriented Software','0201634988');
```

</spoiler>

Ajoutez un livre ou deux livres dont vous ne connaissez que le titre et l'auteur. <spoiler|Solution : >

```
INSERT INTO livre0(titre,auteur) VALUES('Le Seigneur des Anneaux','Tolkien');
INSERT INTO livre0(titre,auteur) VALUES('Cryptonomicon','Neal Stephenson');
```

</spoiler>

Exécutez des requêtes SELECT

Projection

N'afficher que quelques colonnes d'une table : titre et auteur par exemple. <spoiler|Solution : >

```
SELECT titre, auteur FROM livre0;
```

</spoiler>

Sélection (restriction)

N'afficher que certaines lignes d'une requête (table ou jointure) : liste des titres de 2007. <spoiler|Solution >

- Afficher toutes les colonnes (symbole *) d'une sélection

  ```
  SELECT * FROM livre0 WHERE annee_parution=2007;
  ```

- Même requête de sélection avec une projection

  ```
  SELECT titre, auteur, editeur FROM livre0 WHERE annee_parution=2007;
  ```

</spoiler> Affichez les livres dont l'éditeur ou l'année de parution ou le prix n'ont pas été renseignés. <spoiler|Solution >

```
SELECT * FROM livre0 WHERE editeur IS NULL;
Le Seigneur des Anneaux||Tolkien|||SELECT titre, auteur FROM livre0 WHERE
annee_parution IS NULL;

SELECT titre, auteur FROM livre0 WHERE prix IS NULL;
```

</spoiler>

Opérateurs logiques

AND, OR, NOT, LIKE, ...

Exemples de requêtes :

- liste de tous les livres de l'éditeur `Eyrolles`, parus en 2007 ;
- les livres dont le titre contient `Python` ;
- les livres de l'éditeur Eyrolles dont le titre contient Python,
- nom des auteurs qui ont publié en 2007.
- liste des livres dont l'éditeur ou l'année de parution ou le prix n'ont pas été renseignés.

<spoiler|Solution >

```
SELECT titre, auteur FROM livre0 WHERE editeur IS NULL OR annee_parution IS
NULL OR prix IS NULL;
```

</spoiler>

- ...

Opérations sur les regroupements

- Afficher le nombre total de livres ;
- Afficher le nombre de livres par éditeur ;
- Afficher le prix moyen d'un livre par éditeur.

Questions à propos du modèle relationnel

Cette table est elle en 1ère Forme Normale, dans les différents cas évoqués ci-dessous :

1. si dans votre sujet, on vous dit que, dans le champ auteur, il ne peut y avoir qu'un seul nom ?
2. sans précisions particulières sur le sujet traité ?
3. si dans votre sujet, on vous dit que, dans le champ auteur, il peut y avoir plusieurs noms d'auteur ?

Solutions : que devez vous faire dans chacun de ces 3 cas ?

Exercice 2

Voici le MCD correspondant :

Il s'agit, ici de vous familiariser avec les notions de clés primaires et étrangères et de jointure entre tables dans les requêtes SQL.

Créez les tables

- Créez la table *editeur* : `editeur(num_editeur, nom_editeur)`

Attention, il s'agit de la colonne : num_editeur.

- Créez la table *livre* : `livre(num, titre, auteur, fk_num_editeur#, annee_parution, prix)`

num signifie que la colonne est clé primaire.
`fk_num_editeur#` : Le caractère dièse (#) signifie que fk_num_editeur (le numéro éditeur) est une clé étrangère.

Vous chercherez dans la documentation de SQLite3 comment déclarer une clé étrangère : http://www.sqlite.org/foreignkeys.html et trouverez ici : des liens vers de la documentation SQLite.
Remarque : SQLite ne respecte pas entièrement le santard SQL92 : http://stackoverflow.com/questions/1884818/how-do-i-add-a-foreign-key-to-an-existing-sqlite-3-6-21-table

Colonne	Type	Commentaires
num_editeur	type compteur auto-incrémenté	
num	type compteur auto-incrémenté	
titre	varchar(100) NOT NULL	valeur obligatoire
nom_editeur	varchar(100) NOT NULL	valeur obligatoire
auteur	varchar(100)	
fk_num_editeur	entier	INTEGER ou INT (en Sqlite3)
annee_parution	entier	INTEGER ou INT (en Sqlite3)
prix	réel	DOUBLE

<spoiler |Solution>

```
F:\>cd __2012_2013\_PREPAS\jeudi\sqlite3\
F:\__2012_2013\_PREPAS\jeudi\sqlite3>sqlite3 base_livres.sqlite3
SQLite version 3.7.17 2013-05-20 00:56:22
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
sqlite> .tables
editeur  livre
sqlite> .schema
```

Affichage de

```
CREATE TABLE editeur( num_editeur INTEGER PRIMARY KEY ,  nom_editeur
VARCHAR(100) NOT NULL );
CREATE TABLE livre(num_livre INTEGER PRIMARY KEY ,  titre VARCHAR(100) NOT
NULL, isbn VARCHAR(100) NOT NULL, auteur VARCHAR(300), annee_parution INT,
prix DOUBLE, fk_num_editeur INTEGER, FOREIGN KEY (fk_num_editeur) REFERENCES
editeur(num_editeur));
sqlite>
```

</spoiler>

Insérez des lignes dans les tables

Avec un script SQL

Vous pouvez copier et modifier votre script de l'exercice précédent.
Attention : veillez à respecter la cohérence entre les tables. Vous pouvez, aussi, faire un INSERT à partir d'un SELECT sur la table livre0. INSERT .... INTO table1 .. SELECT ..... FROM table2 WHERE .....; \\Attention : veillez à respecter la cohérence entre les tables. Voir ici : http://www.sqlite.org/lang_insert.html Remarque pour les clés étrangères : pour SQLIte, il est impératif de préciser en début de session : sqlite>PRAGMA foreign_keys = ON; Sinon les contraintes de clés étrangères ne seront pas vérifiées. <spoiler | Solution>

```
SELECT * INTO livre FROM livre0;

ou

CREATE TABLE livre AS SELECT * FROM livre0
```

</spoiler>

<spoiler | Un exemple plus complexe >

```
sqlite> SELECT * FROM livre;
1|Python 2|222222|BBBBB|2007|22.225|2
2|Python et PyQT|444444|DDDDD|2007|22.22|2
3|Python 3|333333|CCCCC|2010|33.33|1
sqlite> SELECT * FROM livre0;
UML 2 en action ,De l'analyse des besoins à la conception|2212121040|Pascal
Roques, Franck
 Vallée|2007|42.95|Eyrolles
Le Seigneur des Anneaux||Tolkien|||
sqlite> INSERT INTO livre0(titre,auteur) VALUES('Cryptonomicon','Neal
Stephenson');
sqlite> insert into livre0 select titre, isbn, auteur, annee_parution, prix,
nom_editeur f
rom livre INNER JOIN editeur ON editeur.num_editeur = livre.fk_num_editeur;
sqlite> select * from livre0;
```

```
UML 2 en action ,De l'analyse des besoins à la conception|2212121040|Pascal
Roques, Franck
 Vallée|2007|42.95|Eyrolles
Le Seigneur des Anneaux||Tolkien|||
Cryptonomicon||Neal Stephenson|||
Python 2|222222|BBBBB|2007|22.225|Dunod
Python et PyQT|444444|DDDDD|2007|22.22|Dunod
Python 3|333333|CCCCC|2010|33.33|Eyrolles
sqlite>
```

</spoiler>

En mode interactif

Utilisez directement l'ordre INSERT en "ligne de commandes". Evitez les accents pour pouvoir réutiliser votre base sous Windows et en Python.

Essayez de :

- provoquer des erreurs de cohérence entre les tables ;
- respecter l'intégrité de la base en insérant des données valides.

Requêtes avec jointures

Effectuez les mêmes requêtes que dans l'exercice 1, mais en prenant en compte "l'aspect jointure". Conseil : ajoutez un éditeur qui n'a pas encore de livre stocké dans votre bibliothèque et essayez des jointures externes : gauche, droite et complète.

Requêtes : Affichez le nombre de livres par éditeur : afficher le nom de l'éditeur et le nombre de lignes. Affichez le prix moyen d'un livre par éditeur : afficher le nom de l'éditeur et le prix moyen.

Exercice 3

Créez une table `auteur` et liez la à la table `livre`.

Posez les contraintes d'intégrité (clés primaires et étrangères).

Insérez des données qui respectent et/ou ne respectent pas les contraintes d'intégrité.

Attention : le MCD fait apparaître une association (un lien) entre ces deux tables, mais les cardinalités sont (1,N) de chaque côté.

Vous pouvez maintenant passer à l'Écriture de requêtes sur une base existante.

From:
https://wikisio.lyceejeanbart.fr/ - **wikiSio**

Permanent link:
https://wikisio.lyceejeanbart.fr/doku.php?id=ouvert_a_tous:prepas:bdd:bdd_relationnelle&rev=1440865027

Last update: **2022/12/03 07:45**