

Projet Interrogation de Bases de Données

L3 MIAGE - 2014-2015

Date de rendu : le fichier contenant vos requêtes SQL doit être déposé sous Arche avant le 12/01/2015, 23h00.

A propos des groupes : le projet est à réaliser en binôme. Un seul trinôme sera accepté, dans le cas où il y aurait un nombre impair d'étudiants.

Fichier à rendre : le fichier que vous déposerez contiendra en entête un commentaire (ligne précédée de "--") contenant le nom des personnes de votre binôme. Pour chaque question, vous indiquerez en commentaire le numéro de la question, puis vous donnerez la/les requêtes répondant à la question. Exemple :

```
-- Dupont François
-- Albert Denis

PRAGMA foreign_keys = ON;

-- Question 1
SELECT * FROM ...

-- Question 2
...
```

Remarques :

- Les requêtes contenues dans votre fichier ne doivent en aucun cas générer de messages d'erreur
- Même si sous SQLite 3, le typage des attributs est dynamique, nous vous demandons de typer les attributs de vos tables
- Pensez à indiquer à SQLite de prendre en compte les clés étrangères (PRAGMA foreign_keys = ON;)
- Les jointures seront faites **obligatoirement** avec un JOIN
- Testez vos requêtes

1. Des boutiques de vente en ligne

Nous considérons une application en ligne permettant à ses abonnés de comparer les prix de produits vendus dans différentes boutiques. La base de données de notre application contient les relations suivantes :

Boutique (idB, nomBoutique, mail, numero, rue, ville, CP, idGerant)

où *idGerant* est une clef étrangère vers la relation *Personne* (le gérant de la boutique)

Personne (idP, nom, prenom, mail, login, motDePasse)

Marque (idM, nomMarque)

Produit (idProd, libelle, idMarque, description)

où *idMarque* est une clef étrangère vers la relation *Marque*

PrixProduit (idProduit, idBoutique, prix)

où *idProduit* est une clef étrangère vers la relation *Produit*

où *idBoutique* est une clef étrangère vers la relation *Boutique*

Promotion (idBoutique, reduction)

où *idBoutique* est une clef étrangère vers la relation *Boutique*

Chaque boutique possède un gérant. Les gérants et les abonnés sont stockés dans la table *Personne*. Nous remarquons qu'un produit possède une marque (*idMarque*) et peut être vendu dans plusieurs boutiques et à des prix différents (relation *PrixProduit*). De plus, les gérants peuvent créer des promotions : réduction accordée sur les prix des produits vendus dans une boutique (en pourcentage). Dans ce cas, le prix d'un produit vendu dans une boutique est égal au prix indiqué par la relation *PrixProduit*, moins la réduction indiquée dans la relation *Promotion* (bien entendu si une réduction est accordée par la boutique). Pour finir, nous considérons qu'un produit (*idProd*=*x*) est vendu dans une boutique (*idB*=*y*) s'il existe un réel *p* tel que le n-uplet (*x*, *y*, *p*) appartient à la relation *PrixProduit*.

2. Questions

1. Écrire le code SQL permettant de générer notre base de données :

- Concernant la relation *Personne* :
 - Le nom, prénom, login et mot de passe d'une personne ne doivent pas être NULL
 - Le login de chaque personne doit être unique
- Concernant la relation *Boutique* :

La suppression d'un gérant doit avoir pour conséquence la mise à NULL de la valeur de l'attribut *idGerant* pour les n-uplets correspondant à sa ou ses boutiques.
- Concernant la relation *Produit* :

Lors de la suppression d'une marque, l'attribut *idMarque* des produits de cette marque sont mis à NULL.
- Concernant la relation *PrixProduit* :
 - Les prix sont des réels
 - La suppression d'un produit ou d'une boutique doit entraîner la suppression des n-uplets de cette relation en lien avec le produit ou la boutique supprimé.

- Concernant la relation Promotion :
 - La réduction accordée sur un produit par une boutique est un entier.
 - La suppression d'une boutique doit aussi entraîner la suppression des n-uplets de cette relation en lien avec la boutique supprimée.

2. Insérez des données dans vos tables
3. Quels sont les noms des boutiques (sans doublons) ?
4. Quel est l'adresse mail de la personne dont le nom est "Dupont" et le prénom est "François" ?
5. Combien y a-t-il de produits en tout ?
6. Pour chaque produit, indiquez le libellé ainsi que le nom de sa marque.
7. Insérez un produit dont l'attribut idMarque est NULL.
Nous souhaiterions que la requête précédente fasse aussi apparaître le produit que vous venez d'insérer. Proposez une solution simple (sans utiliser la clause WHERE).
8. Quels sont les libellés et marque des produit vendus dans la boutique nommée « Chez Lulu » ? Vous trierez les résultats par ordre alphabétique.
9. Quels sont les gérants ?
10. Créez une vue contenant la liste des gérants, que vous nommerez *gérant*.
11. En utilisant cette vue, pour chaque gérant, indiquez leur nom, prénom et nombre de boutiques possédées.
12. Même question, en n'affichant uniquement les gérants ayant entre 2 et 3 boutiques.
13. Un utilisateur effectue une recherche pour un produit dont le libellé est "fourchette". Pour chaque boutique vendant ce produit, indiquez le nom de la boutique, le prix de vente du produit "fourchette", la réduction accordée par la boutique, ainsi que le prix en tenant compte de la promotion, dans le cas où il y en aurait une.
La dernière colonne sera renommée "prixApresReduction".
14. Donnez la liste des personnes qui ne sont pas gérantes. Pour cela, vous utiliserez un opérateur ensembliste .
15. Même question, mais cette-fois, vous utiliserez une sous-requête.
16. Donnez la liste de toutes les adresses mail (boutique et personne confondues)
17. Quelle est la (ou les) boutique offrant la plus grande réduction ?
18. Afficher les boutiques vendant au moins un des articles vendus dans l'une des boutiques du gérant dont le prénom est "John" et le nom est "Smith".
Concernant cette question, vous justifierez en commentaire votre réponse.
19. Affichez les marques pour lesquelles il existe au moins une boutique vendant tous les produits de celle-ci.
Concernant cette question, vous justifierez en commentaire votre réponse.