

Fiche de TP no 1

Considérez la base de données Hotellerie de schéma suivant :

```
hotel(numhotel, nom, ville, etoiles)
chambre(numchambre, numhotel, etage, type, prixnuitht)
client(numclient, nom, prenom)
occupation(numoccup, numclient, numchambre, numhotel, datearrivee, datedepart)
reservation(numresa, numclient, numchambre, numhotel, datearrivee, datedepart)
```

Compréhension du schéma

Question 1. Quel(le)s sont, dans le schéma ci-dessus,

- les tables ?

Solution. hotel, chambre, client, occupation, reservation □

- les attributs ?

Solution.

- numhotel, nom, ville, etoiles
- numchambre, numhotel, etage, type, prixnuitht
- numclient, nom, prenom
- numoccup, numclient, numchambre, numhotel, datearrivee, datedepart
- numresa, numclient, numchambre, numhotel, datearrivee, datedepart



- les clés primaires ?

Solution. numhotel, le couple “numchambre, numhotel”, numclient, numoccup, numresa □

- les clés étrangères ?

Solution. Celles en italique :

- numhotel (dans chambre)
- numclient, numchambre, numhotel (dans occupation)
- numclient, numchambre, numhotel (dans reservation)



Question 2. Dans le schéma ci-dessus, les types de colonnes ne sont pas spécifiés. Quel type de donnée (entier, réel (flottant), chaîne de caractères, date, ...) donneriez vous à chaque colonne (de chaque table) ?

Solution.

- numhotel : entier, nom, ville : chaîne, etoiles : entier
- numchambre : entier, numhotel, etage : entier, type : chaîne, prixnuitht : réel
- numclient : entier, nom, prenom : chaînes
- numoccup, numclient, numchambre, numhotel : entier, datearrivee, datedepart : dates
- numresa, numclient, numchambre, numhotel : entiers, datearrivee, datedepart : dates



Question 3. Quelles sont les tables de cette DB sur lesquels on peut faire un

1. PRODUIT,

Solution. En principe, on peut faire un produit seulement quand les noms de colonnes dans les deux tables sont distingués. Dans notre base, cela s'avère seulement avec les tables chambre et client. □

Mais rappelons, on se permettre toujours de faire un produit après avoir renommé les colonnes, par exemple en préfixant les colonnes avec le nom de la table. □

2. UNION,

Solution. Pour faire cette opération il faut que le schéma des deux tables soit le même. Ici ce n'est jamais le cas, même pour les tables occupation et réservation, dont le nom de la clés étrangères est différente. □

3. NATURAL JOIN ?

Solution. Si les types de données portés par le colonnes nommé de la même façon sont cohérents (comme ici), alors c'est toujours possible de faire cette opération. □

Pour quelle couple de table serait préférable utiliser JOIN à la place de NATURAL JOIN ?

Solution. Pour les tables hotel et client, les deux colonnes appellés de la même façon sont 'nom'. Par ailleurs, elles sont sensé contenir des donnees de nature complément différente (le nom d'un auberge, le nom d'une madame ou d'un monsieur) et donc pas de raison de demander l'égalité des valeur y contenus. Une jointure naturelle sur ces deux tables retournera, avec bcp. de probabilité, aucune tuple. □

Prise en main de DB Browser for SQLite3

Vous allez vous logguer sur les machines CENTRUX ou LUMINUX. Créez un répertoire nommé BD/TP1. Depuis la page du cours sur AMETICE, télé-chargez le fichier BD-Hotel-SQLite.zip dans ce répertoire. Ce fichier archive contient le code sql pour construire et inhabiter la base hotellerie.db).

Question 4. A l'aide d'un terminal, ouvrez DB Browser :

```
> sqlitebrowser
```

Créer (dans le répertoire BD/TP1) une nouvelle base de données qui sera appelée hotellerie.db. Importez dans la base les fichiers sql contenus dans le répertoire BD-Hotel-SQLite.

- Sauvegarder la base de données,
- Naviguer la structure de la base,
- naviguer la données de la base.
- Depuis l'onglet "Executer le code SQL" exécuter la requête

```
SELECT * FROM hotel ;
```

- Quittez DB Browser.

Prise en main de SQLite3

C'est important de comprendre la différence entre **DB Browser for SQLite3** et **SQLite3** : le premier est seulement une (jolie) interface graphique nous permettant de piloter **SQLite3** qui lui est le vrai SGBD. Nous allons nous en convaincre en utilisant **SQLite3** depuis un terminal. Pour ce faire, vous allez effacer le fichier hotellerie.db et le répertoire BD-Hotel-SQLite.

Question 5. Des-archivez le fichier BD-Hotel-SQLite.zip et déplacez vous dans le répertoire BD-Hotel-SQLite :

```
> unzip BD-Hotel-SQLite.zip  
> cd BD-Hotel-SQLite
```

Listez le contenu de ce répertoire. Vous pouvez ouvrir les fichiers dans le répertoire (avec l'éditeur de texte **geany** ou votre éditeur favori) pour en découvrir le contenu.

Question 6. Lancez **sqlite3** de façon à créer la base **hotellerie.db** :

```
> sqlite3 hotellerie.db
```

Exécutez la commande **.help** pour afficher l'aide de **sqlite3**. Lisez attentivement la description des commandes suivantes qui s'avéreront être utiles : **.read**, **.header**, **.mode**, **.exit**.

Question 7. Lisez les fichiers (toujours depuis **sqlite3**, avec la commande **.read**) pour construire les tables et y ajouter les données. Dans quel fichier, à votre avis, est contenue la description de la structure de la base de données ? Dans quel fichier sont contenues les tuples ? Dans quel ordre faut-il lire ces fichiers : faut-il d'abord lire la description de la structure de la base de données, ou bien faut-il ajouter les tuples à la base de données ?

Question 8. Utilisez la requête

```
SELECT * FROM hotel ;
```

pour afficher le contenu de la table **hotel**. Attention à ne pas oublier le ; à la fin. De façon simillaire, affichez le contenu des autres tables (**client**, **chambre**, **occupation**, **reservation**).

Question 9. Sélectionnez un mode d'affichage des tables qui plus vous convient. Pour ce faire, demandez (en utilisant la commande **.help .mode**) quels sont les modes d'affichage disponibles.

Question 10. Sortez de **sqlite3**. Jetez un œil au fichier de sauvegarde. Quelle est sa taille ?

Question 11. Relancez **sqlite3** avec la base **hotellerie.db**. Vérifiez que tout a bien fonctionné : les tables sont toujours là ? les données aussi ?

Interrogation des données

De maintenant, vous pouvez utiliser **DB Browser** ou **SQLite3**, comme vous le souhaitez. Sauf si vous avez déjà un doctorat en informatique, nous suggérons d'utiliser **DB Browser for SQLite3**, beaucoup plus convivial.

Question 12. Considérez les expressions de l'algèbre relationnelle ci dessous. Pour chaque expression :

- Écrivez une requête SQL équivalente.
- Exécutez, avec **sqlite3**, cette requête sur la base de données **hotellerie.bd**.
- Proposez une phrase équivalente dans la langue française.

1.

Solution.

```
SELECT DISTINCT etoiles FROM hotel ;
```

```
SELECT DISTINCT etoiles FROM hotel ORDER BY etoiles ;
```

Combien d'étoiles un hotel peut avoir ?

2. PROJECT(Chambre, type)

Solution.

```
SELECT DISTINCT type FROM Chambre ;
```

```
SELECT DISTINCT type FROM Chambre ORDER BY type ;
```

Quels sont les types de chambre ?

3. PROJECT(RESTRICT(hotel, etoiles <= 2), nom, ville)

Solution.

```
SELECT nom, ville FROM hotel WHERE etoiles <= 2 ;
```

Les hotels avec au plus 2 étoiles

4. PROJECT(
 RESTRICT(
 NATURAL JOIN(client, occupation)
 datearrivee >= '2014-04-01')
 nom, prenom)

Solution.

```
SELECT nom, prenom FROM client NATURAL JOIN occupation WHERE datearrivee >= '2014-04-01' ;
```

Les clients ayant séjourné quelque part à partir du 1 avril 2014

5. PROJECT
 RESTRICT(
 JOIN(client, NATURAL JOIN(occupation, hotel),
 client.numclient = occupation.numclient)
 ville = 'Lyon')
 client.nom, client.prenom)

Solution.

```
SELECT
-- ligne suivante ajoutée
ROW_NUMBER() OVER(ORDER BY client.nom) as row_number,
--
client.nom, client.prenom
FROM
(client NATURAL JOIN occupation) JOIN hotel USING (numhotel)
WHERE ville = 'Lyon' ;
```

Les clients ayant séjourné à Nice



Question 13. Exécutez la requête suivante (depuis sqlite3 ou depuis **DB Browser for SQLite3**) sur la base de donnée **hotellerie.db** :

```
SELECT * FROM
client NATURAL JOIN hotel NATURAL JOIN occupation
```

- Combien d'enregistrements sont trouvés ?
- Expliquez pourquoi.

Comment modifier cette requête pour trouver dans quelle ville chaque client a séjourné ?

Question 14. Écrivez et puis exécutez les requêtes SQL correspondantes aux questions suivantes. Avant rédiger les requêtes SQL, listez, pour chaque question, les opérations de l'algèbre relationnelle qu'il faut utiliser.

Conseil : créez et un fichier **tp1.sql** et écrivez dans ce fichier (avec geany ou un autre éditeur de texte) les requêtes ; puis copier-collez les dans le terminal où **sqlite3** est ouvert pour tester (sous linux : copier = surligner, coller = clique bouton du milieu sur la souris). Pour exécuter les requêtes, vous pouvez utiliser **sqlite3** ou **DB Browser for SQLite3**.

1. Listez les noms d'hôtels et leur ville (12 enregistrements trouvés)

Solution.

```
select nom, ville from hotel ;
```



2. Listez les noms d'hôtels ayant 3 étoiles ou plus (2 enregistrements trouvés)

Solution.

```
select nom from hotel where etoiles > 3 ;
```



3. Listez les hôtels sur lesquels portent une réservation

Solution.

```
select nom from hotel natural join reservation ;
select distinct nom from hotel natural join reservation ;
select nom from hotel, reservation where hotel.numhotel = r.numhotel ;
select nom from hotel join reservation on (hotel.numhotel = reservation.numhotel) ;
select nom from hotel as h join reservation as r on (h.numhotel = r.numhotel) ;
select nom from hotel h join reservation r on (h.numhotel = r.numhotel) ;
select nom from hotel join reservation using (numhotel) ;
```



(30/9 enregistrements trouvés, selon que l'on utilise DISTINCT)

4. Listez les clients ayant déjà occupé (ou en cours d'occupation d') une chambre.

Solution.

```
select nom, prenom from client natural join occupation ;
select distinct nom, prenom from client natural join occupation ;
```



(248/163 enregistrements trouvés, selon que l'on utilise DISTINCT)

5. Quels sont tous ceux qui ont été, ou sont encore, clients de l'Hôtel des voyageurs à Nice ?

Solution.

```
SELECT client.nom, prenom
FROM
client NATURAL JOIN reservation JOIN hotel USING(numhotel)
WHERE ville='Nice' AND hotel.nom = 'Hotel des voyageurs'
UNION
SELECT client.nom, prenom
FROM client NATURAL JOIN occupation JOIN hotel USING(numhotel)
WHERE ville='Nice' AND hotel.nom = 'Hotel des voyageurs'
```

```

WITH allOccRes AS
  (SELECT numclient, numhotel FROM occupation
  UNION
  SELECT numclient, numhotel FROM reservation)
  SELECT client.nom, client.prenom
  FROM
  (client NATURAL JOIN allOccRes)
  JOIN hotel USING(numhotel)
  WHERE
  hotel.nom = 'Hotel des voyageurs'
  AND
  hotel.ville = 'Nice' ;

```

□

6. Pour chaque client, lister le nom et la ville des hôtels dans lesquels il a séjourné, ou séjourne encore, avec les dates d'arrivée et de départ correspondant à chaque séjour.

Solution.

```

SELECT client.nom, client.prenom, hotel.nom, ville, datearrivee, datedepart
FROM
client, hotel, occupation
where
client.numclient = occupation.numclient
AND
hotel.numhotel = occupation.numhotel;

SELECT client.nom, prenom, hotel.nom, ville, datearrivee, datedepart
FROM
client NATURAL JOIN occupation JOIN hotel USING(numhotel)

```

□

7. Listez les numéros de chambres disponibles dans l'hôtel terminus à Bordeaux, pour les nuits du 12 avril 2014 au 17 avril 2014 (on pourra faire l'hypothèse que la date courante est le 7 avril 2014, de façon à considérer seulement les réservations).

Solution.

```

SELECT numchambre FROM
(hotel NATURAL JOIN chambre) LEFT JOIN reservation
USING(numchambre, numhotel)
WHERE
nom = 'Hotel terminus' AND ville = 'Bordeaux'
AND
('2014-04-17' <= datearrivee
OR
datedepart <= '2014-04-12'
OR
datedepart IS NULL);

```

□

8. Combien de chambres triples possède l'Hôtel des ambassadeurs à Grenoble ?

Solution.

```
SELECT count(numchambre)
FROM chambre NATURAL JOIN hotel
WHERE
type = 'triple'
AND nom = 'Hotel des ambassadeurs'
AND ville = 'Grenoble'
;
```

□

(1 enregistrements trouvés)

9. Pour chaque hôtel, indiquer le nombre total de chambres. (il faudra utiliser les fonctions d'agrégats, COUNT avec GROUP BY en SQL).

Solution.

```
SELECT nom, ville, COUNT(numchambre)
FROM hotel natural join Chambre
GROUP BY numhotel
```

□

(12 enregistrements trouvés)