

# TP base de données SQLite

Pour ce TP : utiliser le compte `prepa`

## 1 Différents choix possibles et choix de SQLite :

La plupart des logiciels de gestion de base de données fonctionnent à l'aide d'un serveur. Ils demandent donc une installation plus importante. Nous nous contenterons ici de SQLite, qui ne demande qu'une installation très rapide et de son interface DB Browser for SQLite.

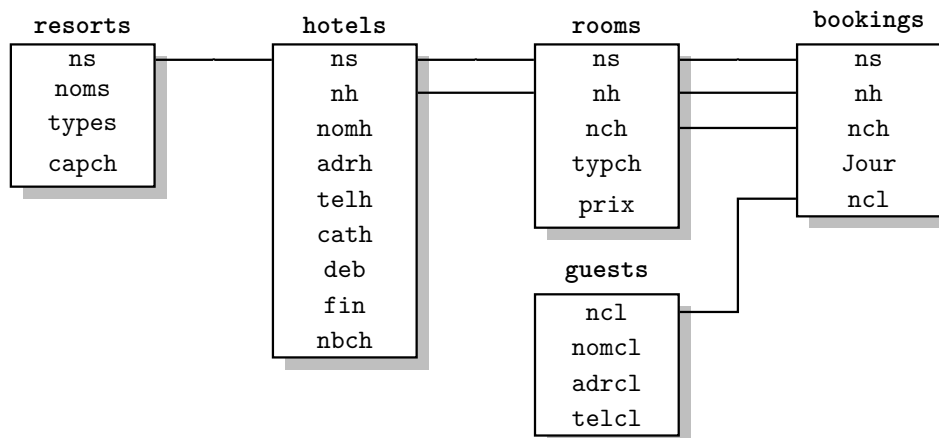
### 1.1 Lecture d'une base de données

## 2 Requêtes sur une base de données

- Récupérer le fichier : `hotel.sqlite` dans `/home/profs/bondil/Public/`
- Avec un clic-droit sélectionner dans le menu **Applications** puis **Développement** puis **DB Browser for SQLite** et ouvrir le fichier `hotel.sqlite` Vous voyez apparaître les 8 tables contenues dans cette base. Trois d'entre elles (eleves, notes, cours) ne nous concerneront pas pour ce TP. Nous allons utiliser les 5 tables qui concernent une agence de Tourisme.

Par commodité, voici les schémas de ces 5 tables de Tourisme, sur papier :

- **resorts** (station en anglais) : (ns, noms, types, capch) : la station de numéro ns a pour nom noms, est de type types (mer ou montagne), et a une capacité de capch chambres au total.
- **hotels** (hôtels en anglais) : (ns, nh, nomh, adr<sub>h</sub>, tel<sub>h</sub>, cath, deb, fin, nbch) : l'hôtel de numéro nh se trouve dans la station de numéro ns, a pour nom nomh, a nbch chambres, admet pour adresse email adr<sub>h</sub>, a pour téléphone tel<sub>h</sub>, est de catégorie cath (nombre d'étoiles) et est ouvert de deb à fin.
- **rooms** : (ns, nh, nch, typch, prix) : la chambre numéro nch de l'hôtel numéro nh est de type typch (S=Simple, D=douche, DWC= douche et WC, SDB=salle de bain) et coûte prix euros.
- **guests** : (ncl, nomcl, adrcl, telcl) : le client numéro ncl a pour nom nomcl, pour adresse email adrcl, avec pour téléphone telcl.
- **bookings** : (ns, nh, nch, Jour, ncl) la chambre numéro nch de l'hôtel numéro nh de la station numéro ns est occupée le jour Jour par le client de numéro ncl.



## 2.1 Requêtes en SQL

Ecrire en langage SQL les requêtes suivantes et les exécuter avec DB Browser, avec l'onglet **Exécuter le SQL**.

*Pour conserver vos requêtes, vous pouvez les écrire dans un petit éditeur de texte, puis les copier coller dans DB Browser.*

**Remarque :** pour les requêtes compliquées, vous pouvez stocker le résultats de requêtes intermédiaires dans de nouvelles tables. Pour cela, faire `CREATE TABLE NomDeLaNouvelleTable AS SELECT . . . . .`

**Rappel du cours sur la syntaxe SQL :** `SELECT . . . FROM . . . WHERE`

- (R0) Liste de noms de tous les hotels, avec leur numéros de station.  
**Vérif. :** On doit obtenir 78 réponses. Le nombre de réponses est marqué en bas de la fenêtre.
- (R1) En faisant suivre la requête précédente de la commande `ORDER BY nomh`, on s'aperçoit que des hotels situés dans des stations différentes portent le même nom : essayez ! On se demande alors combien de *noms* distincts d'hôtels, il y a dans la base. Pour cela, une commande est utile : `SELECT DISTINCT`.  
**Vérif.** On ne trouve plus que 22 réponses, ce qui montre que l'imagination des hôteliers est ici limitée !
- (R2) Liste des numéros et noms des stations de montagne avec leur capacité en chambres.  
**Vérif. :** On doit obtenir six réponses.
- (R3) Liste des noms des hôtels trois étoiles et quatre étoiles, avec leur numéros de station.  
**Vérif. :** On doit obtenir 31 réponses.
- (R4) Liste des noms des hôtels avec leur catégorie et le nom de la station où ils se trouvent, se trouvant dans une station balnéaire.  
*Indication :* Attention, il y a deux tables à croiser.... car le caractère balnéaire d'une station ne se voit pas dans la table hôtel.  
**Vérif. :** On doit obtenir 37 réponses.
- (R5) Liste des noms des stations au bord de la mer ayant au moins un hôtel 4 étoiles.  
**Vérif. :** Vous obtiendrez trois réponses. Si vous ne faites pas de `SELECT DISTINCT`, vous obtiendrez quatre réponse : qu'est ce que cela signifie ?
- (R6) Liste des noms et adresses email des clients dont l'adresse email est l'adresse d'un hôtel de la base (des hoteliers quoi!)  
**Vérif. :** 2 réponses avec `SELECT DISTINCT`, et quatre réponses avec `SELECT` simple, expliquer pourquoi !
- (R7) Liste des noms et adresses des clients ayant réservé à la montagne.  
**Remarque :** Ici deux jointures. Quand elle est possible dans les deux sens, la jointure est associative.  
**Vérif. :** 889 réponses.
- (R8) Nom de l'hotel ayant le plus grand nombre de chambre, avec son numéro de station : attention, on veut que ce nom soit une valeur de retour de SQL, il ne s'agit pas de faire simplement `ORDER BY nbch` pour lire le résultat dans la liste.  
**Vérif. :** Sea Hotel, 1.

- (R9) Liste des numéros des hôtels avec leur numéro de station ayant *au moins* une chambre dont le prix est supérieur strictement à 40 euros.  
 Liste des *vrais noms* des hôtels avec leur numéro de station ayant *au moins* une chambre dont le prix est supérieur strictement à 40 euros  
 En déduire la liste des noms des hôtels avec leur numéro de station ayant *toutes leurs chambres* de prix inférieur ou égal à 40 euros.  
**Vérif.** 31 réponses pour la dernière question.
- (R10) Prix de la chambre la moins chère située dans un hôtel trois étoiles d'une station balnéaire ?  
**Vérif.** 65 euros.
- (R11) Liste des numéros de chambres d'hôtels avec leur noms d'hôtels et leur numéro de station, ayant un prix inférieur au prix moyen de toutes les chambres d'hôtels. **Vérif.** : 1726 réponses.  
 Puis la liste des hôtels (nomh, ns), avec leur *nombre de chambres* vérifiant la condition de prix précédente.  
 (On utilisera la fonction COUNT)
- On peut se demander ce qu'on met en argument à COUNT. Ici, il paraît naturel de mettre COUNT(room.nch) mais en fait, ici, on pourrait faire compter n'importe quel attribut ce serait pareil ! Notamment, on peut utiliser COUNT(\*).  
 Cependant, il y a une différence entre COUNT(\*) et COUNT(nomAttribut) : le second ne comptera que les lignes où l'attribut nomAttribut est non vide.  
 Voir aussi la requête (R13) ci-dessous où les jointures peuvent aussi faire oublier des choses à compter...
- Vérif.** : 47 réponses.
- (R12) Obtenir la liste des numéros d'hôtels avec leur numéro de station des hôtels 4 étoiles n'ayant que des chambres avec salles de bain.  
**Vérif.** 11 réponses.
- (R13) (i) Listes des hôtels avec leur nom, adresse et leur nombre de réservations dans l'année affiché dans une colonne appelée NbResa, ordonné par nombre de réservation décroissante.  
**Vérif.** on devrait trouver 78 réponses, car on sait bien qu'il y a 78 hôtels mais :  
**Attention :** un hôtel n'a pas de réservation, donc n'apparaît pas dans la table bookings. Donc la jointure avec bookings ne fera apparaître que 77 hôtels. Donc :  
 (ii) Ecrire une requête qui permet de détecter l'hôtel (disons avec ns, nh) qui n'est pas dans bookings.
- (R14) Nom et adresse de l'hôtel de la station « Chamonix » ayant eu le plus de réservations dans l'année.  
*Indication* On peut bien sûr utiliser des requêtes imbriquées, mais cela en fait pas mal, les tester au fur et à mesure. Une autre méthode peut-être plus agréable (requête moins lourde) est d'utiliser un HAVING.  
**Vérification** On trouvera l'Hôtel de la Gentiane.
- (R15) Nom de clients ayant réservé au moins deux jours consécutifs une même chambre dans le même hôtel de la même station.  
**Vérif.** 949 réponses.  
*Indication* – L'idée est de faire une *auto-jointure* c'est-à-dire une jointure de la table bookings avec elle-même. En pratique, la requête commencera par :  
 SELECT DISTINCT g.nomcl FROM guests g, bookings b1, bookings b2
- (R16) Jour de l'année où l'hôtel « Bon séjour » de la station « Chamonix » a eu le plus de réservations.  
**Vérification** On trouvera deux jours : le jour 99 et le jour 100.
- (R17) Noms des clients ayant réservé dans tous les hôtels 2 étoiles de la station « Chamonix ».  
**Vérifications** 5 réponses.