

3.a

SQL:

SELECT

R.id_Reservation,
C.Nom_complet,
H.Ville AS Ville_Hotel

FROM

Reservation R
JOIN Client C ON R.id_Client = C.id_Client
JOIN Chambre CH ON R.id_Chambre = CH.id_Chambre
JOIN Hotel H ON CH.id_Hotel = H.id_Hotel;

Algèbre:

Resultat PROJECTION_id_Reservation, Nom_complet, Ville

(

Reservation JOIN Client JOIN Chambre JOIN Hotel

)

3.b

SQL:

SELECT *

FROM Client

WHERE Ville = 'Paris';

Algèbre:

Resultat SELECTION_Ville='Paris'(Client)

3.c

SQL:

SELECT

C.id_Client,
C.Nom_complet,
COUNT(R.id_Reservation) AS Nb_Reservations

FROM

Client C
LEFT JOIN Reservation R ON C.id_Client = R.id_Client

GROUP BY

C.id_Client, C.Nom_complet;

Algèbre:

Resultat GROUP BY_id_Client, Nom_complet, COUNT(id_Reservation) AS Nb_Reservations

(
 Client LEFT JOIN Reservation
)

3.d

SQL:

SELECT

 TC.Type,
 COUNT(C.id_Chambre) AS Nb_Chambres

FROM

 Type_Chambre TC
 LEFT JOIN Chambre C ON TC.id_Type = C.id_Type

GROUP BY

 TC.Type;

Algèbre:

Resultat GROUP BY_Type, COUNT(id_Chambre) AS Nb_Chambres

(
 Type_Chambre LEFT JOIN Chambre
)

3.e

SQL:

SELECT *

FROM Chambre C

WHERE C.id_Chambre NOT IN (

 SELECT R.id_Chambre

 FROM Reservation R

 WHERE NOT (

 R.Date_depart < :date_debut OR

 R.Date_arrivee > :date_fin

)
);

Algèbre:

R1 `SELECTIONNOT(Date_depart < :date_debut OR Date_arrivee > :date_fin)(Reservation)`

R2 `PROJECTION_id_Chambre(R1)`

Resultat `Chambre DIFF R2`

Différences entre SQLite et MySQL

Critere	SQLite	MySQL
Type	Fichier local	Serveur de base de donnees
Installation	Aucune, juste un fichier `.db`	Requiert un serveur MySQL
Performances	Tres bien pour les petites applis	Optimise pour les applications complexes
Concurrence	Faible (1 seul ecrivain)	Haute (multi-utilisateurs)
Utilisation	Mobile, desktop, prototypage	Web, entreprise, production