

Projet : (QUESTIONS 3 ET 4)

Réaliser par :

-RHAZLANI Ouissal(INFO4)

-SABRY Kaoutar(INFO4)

3. Requêtes SQL et en algèbre relationnelle :

- a. Afficher la liste des réservations avec le nom du client et la ville de l'hôtel réservé

$R1 \leftarrow \text{Réservation} \bowtie \text{Effectuer}$

$R2 \leftarrow R1 \bowtie \text{Client}$

$R3 \leftarrow R2 \bowtie \text{Concerner}$

$R4 \leftarrow R3 \bowtie \text{Chambre}$

$R5 \leftarrow R4 \bowtie \text{Hotel}$

$\text{Résultat} \leftarrow \pi_{\text{id_Réservation}, \text{Date_arrivée}, \text{Date_départ}, \text{Nom_complet}, \text{Ville}} (R5)$

- b. Afficher les clients qui habitent à Paris :

$\text{Résultat} \leftarrow \sigma_{\text{Ville} = 'Paris'} (\text{Client})$

- c. Calculer le nombre de réservations faites par chaque client :

$R \leftarrow \text{Client} \bowtie \text{Effectuer} \bowtie \text{Réservation}$

$\text{Résultat} \leftarrow \gamma_{\text{Nom_complet}, \text{COUNT}(\text{id_Réservation}) \rightarrow \text{Nb_Reservations}} (R)$

- d. Donner le nombre de chambres pour chaque type de chambre :

$R \leftarrow \text{Chambre} \bowtie \text{Type_Chambre}$

$\text{Résultat} \leftarrow \gamma_{\text{Type}, \text{COUNT}(\text{id_Chambre}) \rightarrow \text{Nb_Chambres}} (R)$

- e. Afficher la liste des chambres qui ne sont pas réservées pour une période

donnée (entre deux dates saisies par l'utilisateur) :

$R1 \leftarrow \sigma_{\text{Date_arrivée} < :date2 \wedge \text{Date_départ} > :date1} (\text{Réservation})$

$R2 \leftarrow \text{Concerner} \bowtie R1$

$\text{Chambres_Réservées} \leftarrow \pi_{\text{id_Chambre}}(R2)$

$\text{Résultat} \leftarrow \pi_{\text{id_Chambre, Etage, Fumeurs}}(\text{Chambre}) - \text{Chambres_Réservées}$

4. Question de réflexion :

Qu'est-ce que SQLite ? Quelle est la différence avec MySQL ?

SQLite est un *système de gestion de base de données relationnelle (SGBDR)* léger, rapide et *embarqué*. Contrairement aux bases comme MySQL ou PostgreSQL, *SQLite ne nécessite pas de serveur* : la base de données est simplement un *fichier unique (.db)* stocké localement.

C'est une solution "*serverless*", utilisée dans :

- les applications mobiles (Android, iOS),
- les petits projets de bureau,
- les outils embarqués (IoT),
- ou même en développement ou test avant de migrer vers un serveur SQL complet.

La différences entre SQLite et MySQL :

Type :

SQLite est un SGBD embarqué qui fonctionne directement à partir d'un fichier local. MySQL, en revanche, est un SGBD client-serveur, ce qui signifie qu'il nécessite un serveur pour fonctionner et gérer les connexions.

Installation :

SQLite ne demande aucune installation de serveur ni configuration complexe. MySQL nécessite l'installation et la configuration d'un serveur MySQL.

Stockage :

SQLite stocke toute la base de données dans un seul fichier .db. MySQL gère les données dans plusieurs fichiers internes à son serveur.

Gestion des utilisateurs :

SQLite ne gère pas les utilisateurs ni les privilèges. MySQL permet une gestion complète des comptes, des rôles et des privilèges.

Accès simultané :

SQLite est limité en termes d'accès concurrent, donc peu adapté aux environnements avec plusieurs utilisateurs en même temps. MySQL est conçu pour supporter de nombreux utilisateurs en parallèle.

Performances :

SQLite est rapide et efficace pour des bases simples et locales.

MySQL est plus adapté aux grandes bases de données et aux requêtes complexes.

Fonctionnalités :

SQLite propose les fonctions de base d'un SGBD, mais reste limité (pas de procédures stockées complexes, peu de triggers).

MySQL est un système très complet, avec transactions, vues, triggers, procédures stockées, etc.

Utilisation typique :

SQLite est parfait pour les applications mobiles, les tests et les petits outils locaux.

MySQL est utilisé dans les sites web dynamiques, les systèmes d'information d'entreprise, et les applications à grande échelle.