

Projet Bases de données

IAP S4

Tayibi Ouiame

23 33 45 2

N147049607

Partie 1: Création et interrogation de la base de données

3. Ecrire les requêtes en algèbre relationnelle qui permettent de:

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

Résultat $\leftarrow \pi$ Id_Reservation, Nom_complet, Ville (

(((Reservation \bowtie Id_Client = Client.Id_Client)

\bowtie Reservation.Id_Reservation = Concerner.Id_Reservation)

\bowtie Concerner.Id_Chambre = Chambre.Id_Chambre)

\bowtie Chambre.Id_Hotel = Hotel.Id_Hotel)

)

b. Afficher les clients qui habitent à Paris

Résultat $\leftarrow \sigma$ Ville = 'Paris' (Client)

c. Calculer le nombre de réservations faites par chaque client

Résultat \leftarrow

γ Nom_complet, COUNT(Id_Reservation) \rightarrow Nombre_Reservations (

 Reservation \bowtie Reservation.Id_Client = Client.Id_Client

)

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

Résultat $\leftarrow \gamma$ Type, COUNT(Id_Chambre) \rightarrow Nombre_Chambres (

 Chambre \bowtie Chambre.Id_Type = Type_Chambre.Id_Type

)

e. Afficher la liste des chambres qui ne sont pas réservées pour une période donnée (entre deux dates saisies par l'utilisateur)

ChambresReservees $\leftarrow \pi$ Id_Chambre (

 σ Date_arrivee \leq :date_fin \wedge Date_depart \geq

 :date_debut (

 Reservation \bowtie Concerner ON Id_Reservation

)

)

 Résultat $\leftarrow \pi^* (\text{Chambre}) - \text{ChambresReservees}$

4. Qu'est ce que SQLite, quelle différence avec MySQL?

SQLite est un système de gestion de base de données relationnelle (SGBDR) léger, intégré directement dans l'application qui l'utilise.

Contrairement aux serveurs de bases de données classiques, SQLite ne nécessite pas d'installation ni de configuration : les données sont stockées dans un simple fichier local.

Différences principales entre SQLite et MySQL

Aspect	SQLite	MySQL
Type de système	SGBD embarqué (pas de serveur)	SGBD client-serveur (nécessite un serveur dédié)
Installation	Aucune : fonctionne directement, intégré à l'application	Nécessite l'installation d'un serveur et configuration
Stockage	Fichier local unique (.db)	Données réparties dans plusieurs fichiers sur le serveur
Multi-utilisateurs	Non recommandé (concurrence limitée)	Oui, conçu pour gérer plusieurs connexions simultanées
Performances	Optimisé pour de petites bases et un seul utilisateur	Optimisé pour les applications complexes, utilisateurs multiples
Fonctionnalités	Moins complet	Fonctionnalités riches
Sécurité	Très basique	Gestion complète des droits, utilisateurs, rôles

Utilisation typique	Applications mobiles, tests locaux, petits outils	Applications web, systèmes d'entreprise, services cloud
Support réseau	Non	Oui

SQLite : simple, rapide, local ,idéal pour des projets individuels, tests,
Prototypage

MySQL : puissant, robuste, évolutif , recommandé pour les systèmes en
production avec plusieurs utilisateurs