



# Requêtes SQL et Algèbre Relationnelle

Réalisé par :

Salah Eddine CHELKHA

Ikram MOUNTASSIR

Année universitaire 2024-2025

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

SQL :

```
SELECT r.Id_Reservation, c.Nom_complet, h.Ville
FROM Reservation r
JOIN Client c ON r.Id_Client = c.Id_Client
JOIN Chambre_Reservation cr ON r.Id_Reservation = cr.Id_Reservation
JOIN Chambre ch ON cr.Id_Chambre = ch.Id_Chambre
JOIN Hotel h ON ch.Id_Hotel = h.Id_Hotel;
```

Algèbre relationnelle :

$$\pi_{\text{Id\_Reservation}, \text{Nom\_complet}, \text{Ville}} (((((\text{Reservation} \bowtie \text{Client}) \bowtie \text{Chambre\_Reservation}) \bowtie \text{Chambre}) \bowtie \text{Hotel}))$$

## b. Clients habitant à Paris

SQL :

```
SELECT * FROM Client WHERE Ville = 'Paris';
```

Algèbre relationnelle :

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

## c. Nombre de réservations faites par chaque client

SQL :

```
SELECT c.Id_Client, c.Nom_complet, COUNT(r.Id_Reservation) AS
    Nombre_Reservations
FROM Client c
LEFT JOIN Reservation r ON c.Id_Client = r.Id_Client
GROUP BY c.Id_Client, c.Nom_complet;
```

Algèbre relationnelle :

$$\gamma_{\text{Id\_Client}, \text{Nom\_complet}; \text{COUNT}(\text{Id\_Reservation}) \rightarrow \text{Nombre\_Reservations}}(\text{Client} \bowtie \text{Reservation})$$

## d. Nombre de chambres pour chaque type de chambre

SQL :

```
SELECT tc.Type, COUNT(c.Id_Chambre) AS Nombre_Chambres
FROM Type_Chambre tc
LEFT JOIN Chambre c ON tc.Id_Type = c.Id_Type
GROUP BY tc.Type;
```

Algèbre relationnelle :

$$\gamma_{\text{Type}; \text{COUNT}(\text{Id\_Chambre}) \rightarrow \text{Nombre\_Chambres}}(\text{Type\_Chambre} \bowtie \text{Chambre})$$

## e. Chambres non réservées entre deux dates données

SQL :

```
SELECT ch.*
FROM Chambre ch
WHERE ch.Id_Chambre NOT IN (
    SELECT cr.Id_Chambre
    FROM Chambre_Reservation cr
    JOIN Reservation r ON cr.Id_Reservation = r.Id_Reservation
    WHERE (r.Date_arrivee <= '2025-06-20' AND r.Date_depart >= '
        2025-06-15')
);
```

Algèbre relationnelle :

$$\text{Occupées} \leftarrow \pi_{\text{Id\_Chambre}} \left( \sigma_{\text{Date\_arrivee} \leq '2025-06-20' \wedge \text{Date\_depart} \geq '2025-06-15'} (\text{Reservation} \bowtie \text{Chambre\_Reservation}) \right)$$

$$\text{Résultat} \leftarrow \text{Chambre} - \text{Occupées}$$

## Qu'est-ce que SQLite ?

SQLite est un moteur de base de données SQL open-source, autonome, sans serveur et sans configuration. Il est appelé " lite" car il est léger et ne nécessite qu'une configuration minimale. SQLite est intégré à l'application elle-même, ce qui signifie qu'il ne nécessite pas de processus serveur distinct.

## Comparaison entre SQLite et MySQL

Critère	SQLite	MySQL
Architecture	Embarqué, sans serveur, autonome	Système client-serveur, nécessite un serveur MySQL
Configuration	Aucune configuration requise	Installation et configuration nécessaires
Stockage	Base de données dans un fichier unique	Fichiers multiples (tables, journaux, etc.)
Vitesse	Rapide pour les lectures mono-utilisateur	Optimisé pour accès concurrent multi-utilisateur
Concurrence	Une seule opération d'écriture à la fois	Lectures et écritures simultanées supportées
Taille de base de données	Petites à moyennes (jusqu'à 281 To)	Très grandes bases (plusieurs téraoctets, selon le matériel)
Types de données	Typage dynamique (affinité de type)	Typage strict avec types spécifiques
Transactions	Prise en charge ACID complète	Prise en charge ACID avec fonctionnalités avancées
Gestion des utilisateurs	Basique, mono-utilisateur	Multi-utilisateurs avec rôles et permissions
Procédures stockées	Non prises en charge	Prises en charge
Réplication	Non supportée en natif	Réplication maître-esclave, clustering
Recherche plein texte	Recherche de base	Recherche en texte intégral
Chiffrement	Optionnel via extensions	Chiffrement SSL/TLS intégré
Authentification des utilisateurs	Pas de gestion native	Authentification robuste intégrée
Sécurité réseau	Accès local uniquement	Sécurité réseau via SSL/TLS
Installation	Très simple, sans configuration	Installation plus complexe
Administration	Facile, peu de gestion requise	Nécessite une gestion continue, surtout en production
Courbe d'apprentissage	Facile pour les débutants	Moyenne, nécessite des compétences en gestion de bases de données serveur