

Introduction à SQL : Requêtes simples, jointures et agrégations

Découvrez comment interroger efficacement vos bases de données relationnelles, des requêtes de base aux analyses avancées avec agrégations.

Qu'est-ce qu'une requête SQL simple ?

Les fondamentaux

SQL (Structured Query Language) est le langage universel pour interroger et manipuler des bases de données relationnelles. C'est l'outil essentiel pour extraire l'information dont vous avez besoin.

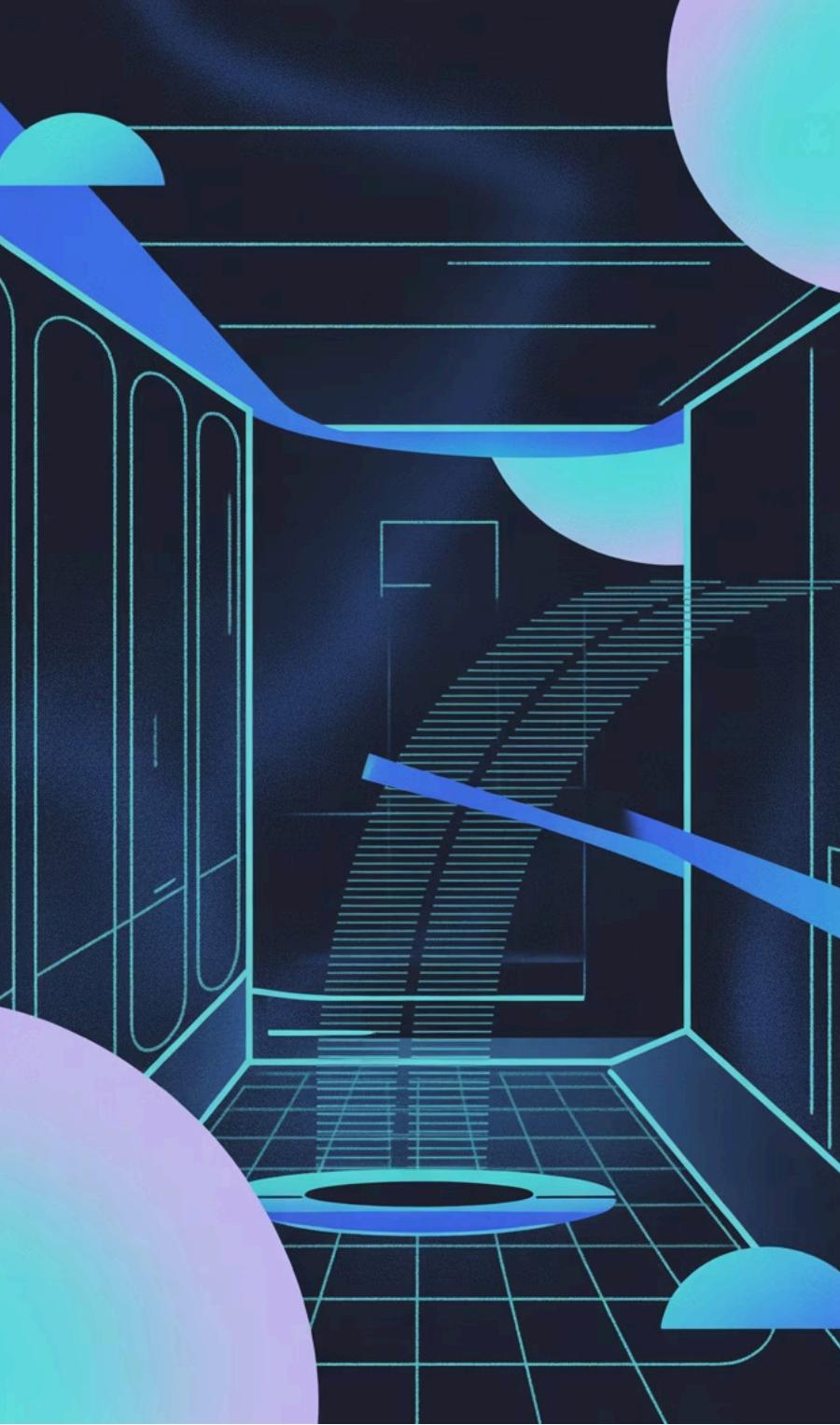
Une requête simple permet de récupérer des données brutes sans transformation ni filtrage complexe.

Exemple basique

```
SELECT * FROM clients;
```

Résultat : Affiche toutes les colonnes et toutes les lignes de la table "clients".

L'astérisque (*) signifie "toutes les colonnes". C'est le point de départ de toute exploration de données.



Filtrer les données avec **WHERE**



La clause WHERE

Permet d'affiner vos résultats en appliquant des conditions précises sur les données.



Syntaxe

```
SELECT * FROM clients  
WHERE ville = 'Paris';
```



Résultat

Seuls les clients parisiens apparaissent dans les résultats. Le filtrage s'effectue avant l'affichage.

Vous pouvez combiner plusieurs conditions avec AND, OR, et utiliser des opérateurs comme >, <, LIKE pour des recherches encore plus précises.

Pourquoi utiliser les jointures SQL ?

01

Données réparties

Dans une base relationnelle, l'information est organisée en plusieurs tables pour éviter la redondance et optimiser le stockage.

02

Relations par clés

Les tables sont liées entre elles par des clés étrangères. Par exemple, la table "commandes" contient un client_id qui référence la table "clients".

03

Combiner pour enrichir

Les jointures permettent de fusionner ces tables pour obtenir une vue complète et cohérente des données.



Exemple de jointure INNER JOIN

L'INNER JOIN est la jointure la plus courante : elle ne conserve que les correspondances exactes entre les deux tables.

1

Objectif

Afficher les commandes avec le nom complet du client qui les a passées.

2

Syntaxe SQL

```
SELECT commandes.id,  
       clients.nom,  
       commandes.montant  
  FROM commandes  
 INNER JOIN clients  
    ON commandes.client_id =  
       clients.id;
```

3

Résultat

Un tableau combiné montrant uniquement les commandes pour lesquelles un client correspondant existe dans la base.

Autres types de jointures

LEFT JOIN

Affiche tous les clients, même ceux qui n'ont passé aucune commande. Les valeurs manquantes apparaissent comme NULL.

Le choix de la jointure dépend de votre besoin analytique : voulez-vous voir toutes les données ou uniquement les correspondances ?

RIGHT JOIN

Affiche toutes les commandes, même sans client correspondant (cas rare dans une base bien structurée).

FULL JOIN

Combine LEFT et RIGHT JOIN pour afficher toutes les lignes des deux tables. Attention : non supporté par tous les systèmes.

Introduction à GROUP BY : regrouper les données

Principe fondamental

GROUP BY permet de regrouper les lignes qui partagent une valeur commune dans une ou plusieurs colonnes.

Cette clause est **indispensable** lorsqu'on utilise des fonctions d'agrégation comme SUM, AVG, COUNT, MAX ou MIN.

Cas d'usage typique

- Total des ventes par client
- Nombre de commandes par région
- Revenus moyens par catégorie de produit



Exemple concret avec GROUP BY et SUM



Table "achats"



Colonnes : client, tarif



Requête SQL



```
SELECT client,  
       SUM(tarif) AS total_achats  
  FROM achats  
 GROUP BY client;
```

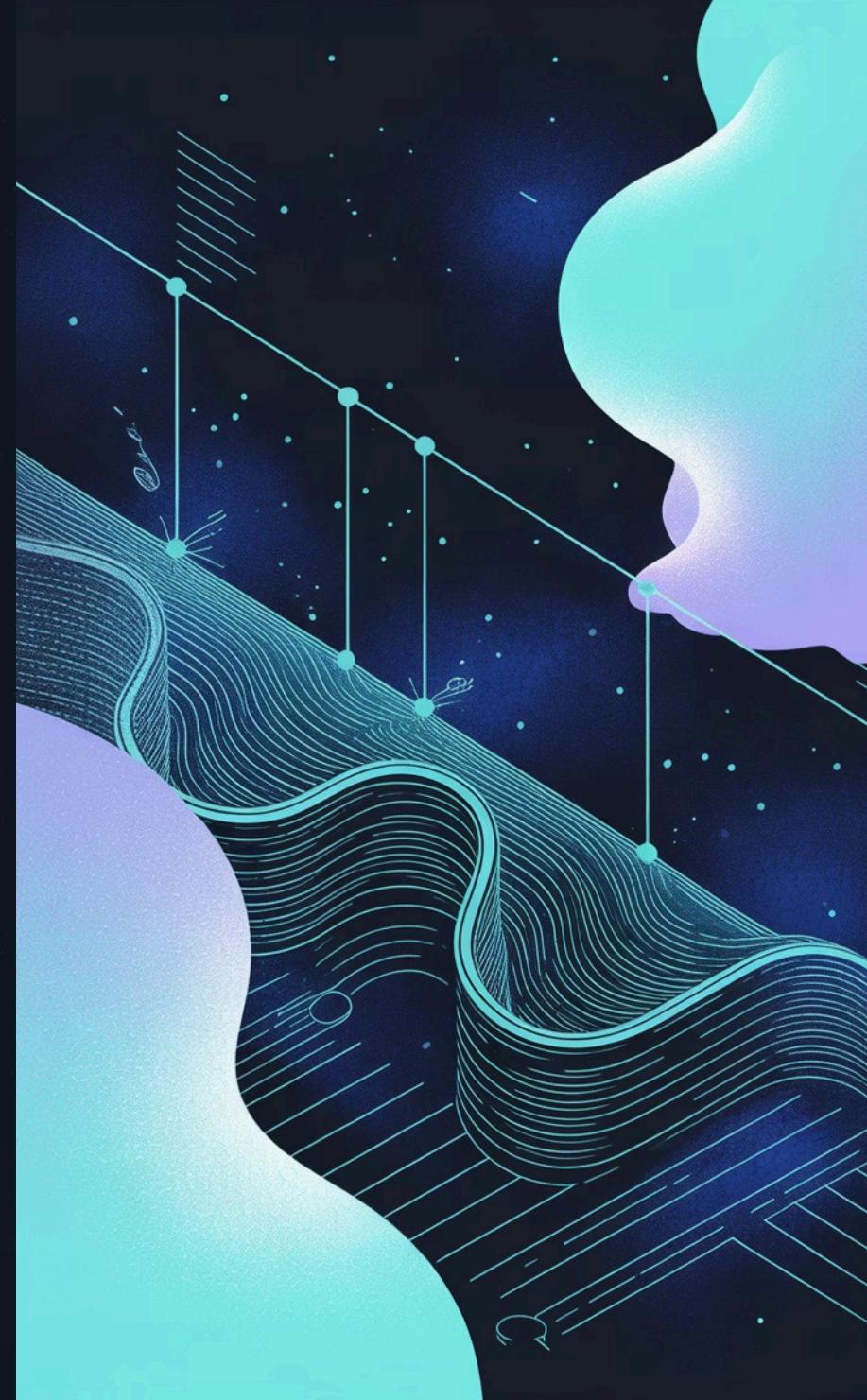


Résultat



Une seule ligne par client avec le montant total de tous ses achats.

Évite les doublons et synthétise l'information.



Fonctions d'agrégation courantes avec GROUP BY

COUNT()

Compte le nombre d'enregistrements par groupe.

Exemple : Nombre de commandes par client

AVG()

Calcule la moyenne des valeurs numériques.

Exemple : Panier moyen par client

MAX() / MIN()

Identifie les valeurs extrêmes dans chaque groupe.

Exemple : Achat le plus cher par client

Exemple : panier moyen par client

```
SELECT client,  
       AVG(tarif) AS panier_moyen  
  FROM achats  
 GROUP BY client;
```

Cette requête calcule le montant moyen dépensé par achat pour chaque client, révélant ainsi leurs habitudes de consommation.

Maîtriser SQL pour exploiter les données efficacement



Requêtes simples

Extraire des données brutes avec SELECT et WHERE pour filtrer selon vos critères.



Jointures

Combiner plusieurs tables pour enrichir vos analyses et obtenir une vue d'ensemble cohérente.



GROUP BY et agrégations

Synthétiser les données pour comprendre les tendances, calculer des moyennes et identifier les patterns.

Pour aller plus loin

- Pratiquez sur vos propres données
- Testez différentes combinaisons de clauses
- Explorez des tutoriels interactifs en ligne
- Utilisez des bases de données d'exemple
- Rejoignez des communautés SQL
- Documentez vos requêtes pour référence future

