

Présentation SQL avancée : Vues, Procédures et plus encore

 par Safia Khoulaïd



Introduction aux vues : Qu'est-ce qu'une vue ?

Une vue est un objet de base de données qui représente une vue pré-définie d'une ou plusieurs tables. Elle offre une manière simplifiée d'accéder aux données en créant une vue abstraite et personnalisée de la base de données.

Avantages

Simplifier l'accès aux données, améliorer la sécurité en limitant les données visibles, permettre des requêtes complexes avec une syntaxe simplifiée, assurer une maintenance plus facile.

Inconvénients

Performance potentiellement inférieure aux requêtes directes sur les tables, dépendance aux tables sous-jacentes, difficulté à gérer les modifications de données dans les vues.

Création et utilisation de vues simples

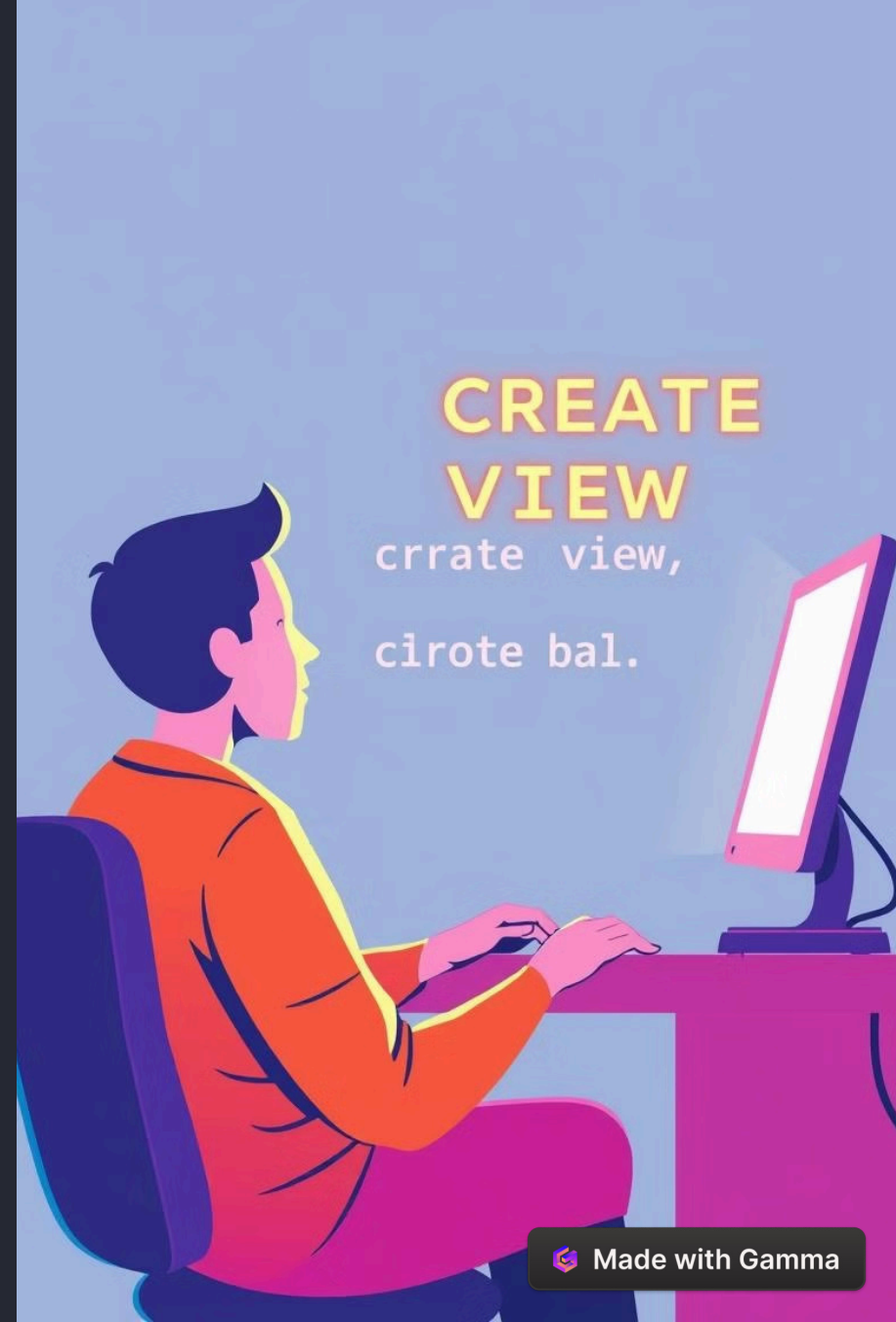
La création d'une vue est simple. La syntaxe consiste à utiliser l'instruction `CREATE VIEW`, suivie du nom de la vue, et de la définition de la requête qui spécifie les données à afficher. L'utilisation d'une vue se fait ensuite par des requêtes standard, comme si la vue était une table normale.

Exemple

```
CREATE VIEW Vue_Employes AS  
SELECT Nom, Prenom FROM  
Employes;
```

Utilisation

```
SELECT * FROM Vue_Employes;
```



Vues complexes avec jointures et fonctions

Les vues peuvent être conçues pour des requêtes complexes incluant des jointures, des fonctions d'agrégation et des conditions de filtrage. Cela permet de créer des vues qui présentent des données regroupées, calculées ou filtrées selon des besoins spécifiques.



Jointures

Joindre plusieurs tables pour créer une vue intégrant des données de différentes sources.



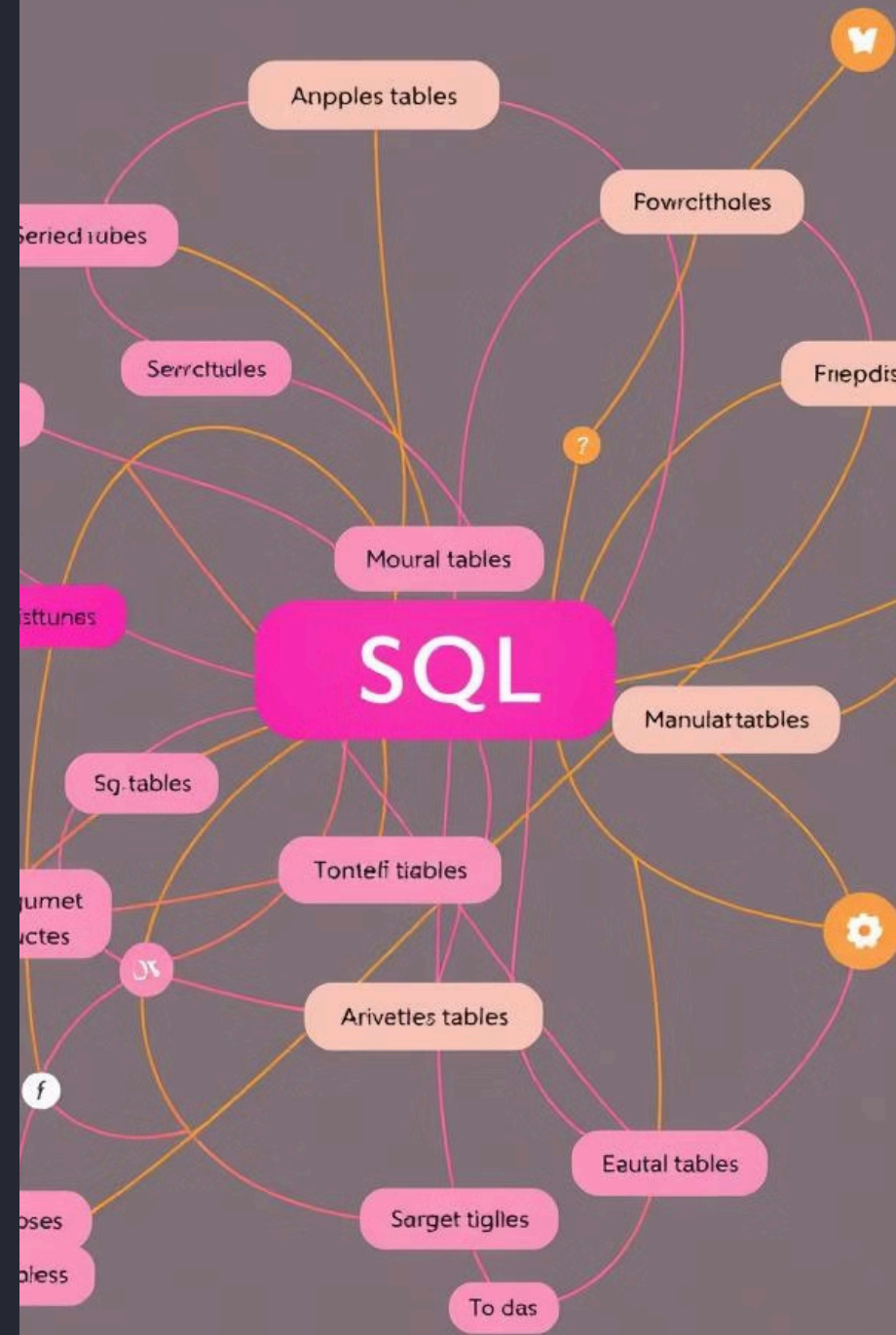
Filtrage

Utiliser des conditions de filtrage pour sélectionner uniquement les données souhaitées dans la vue.



Fonctions

Appliquer des fonctions d'agrégation, de calcul ou de transformation aux données pour personnaliser la vue.



Avantages des vues : Sécurité et performances

Les vues offrent des avantages considérables en matière de sécurité et de performances. Elles permettent de restreindre l'accès aux données en créant des vues avec un champ de vision limité, améliorant la sécurité et l'intégrité des données.



Procédures stockées : Définition et syntaxe

Une procédure stockée est un ensemble d'instructions SQL pré-compilées et stockées dans la base de données. Elle peut être appelée et exécutée à la demande, simplifiant l'accès aux données et automatisant des tâches complexes.

1

Définition

Création d'une procédure stockée avec l'instruction `CREATE PROCEDURE`, suivie du nom de la procédure et du corps de code SQL.

2

Syntaxe

Syntaxe standardisée pour la création et l'exécution de procédures stockées, offrant une approche structurée et cohérente.

3

Exécution

Appel et exécution de la procédure stockée à la demande, simplifiant l'interaction avec la base de données.

Paramètres d'entrée et de sortie des procédures

Les procédures stockées peuvent accepter des paramètres d'entrée et retourner des valeurs en sortie. Ces paramètres permettent de personnaliser le comportement de la procédure et de transmettre des informations spécifiques à l'exécution.

1

Entrée

Paramètres d'entrée pour personnaliser le comportement de la procédure et transmettre des valeurs spécifiques.

2

Sortie

Retour de valeurs en sortie par la procédure, permettant de transmettre des résultats ou des informations spécifiques.

3

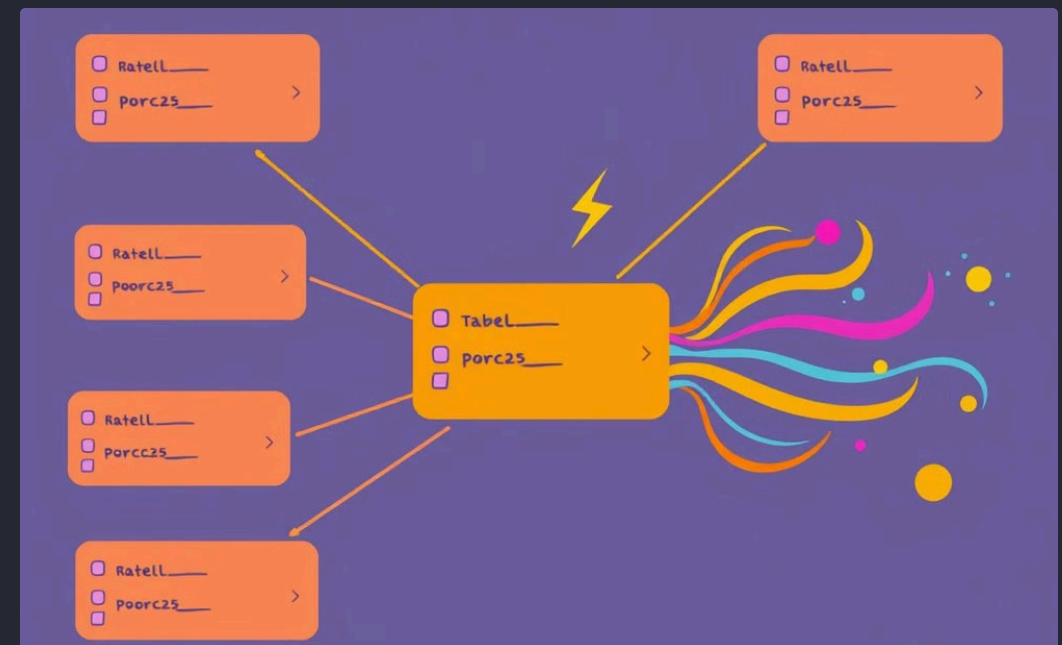
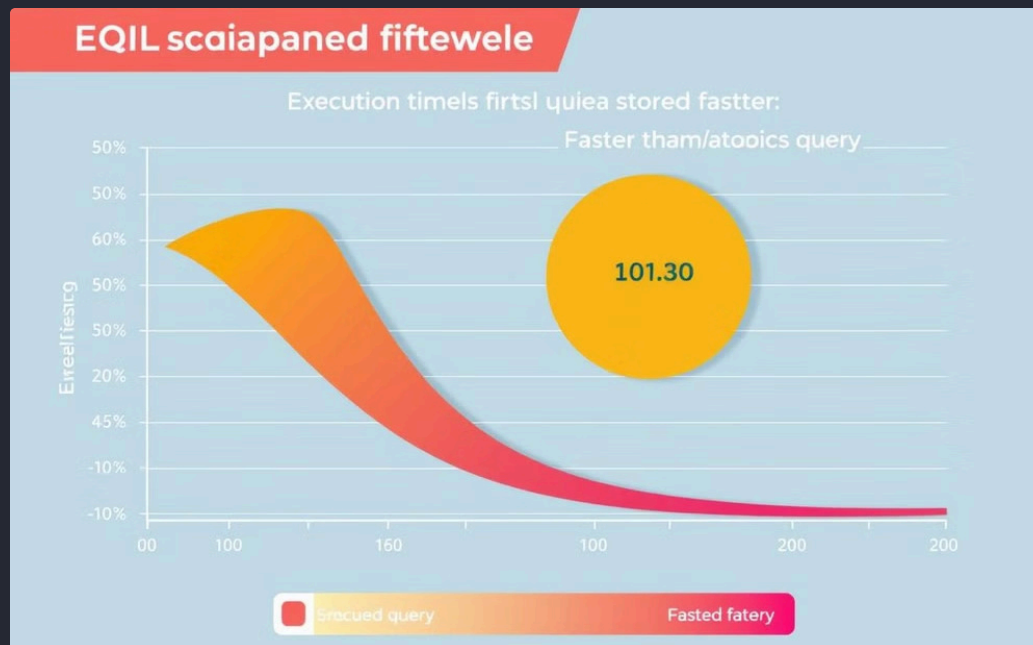
Combinaison

Utilisation de paramètres d'entrée et de sortie pour créer des procédures flexibles et interactives.

```
STodes:: Faropecta/auratores.  
SOates: Today teach(acogreton octaty)  
CDate: fu19950 pethute the farendate  
scessurf bosay cilse.  
SOates: 182-05000 201101/ccfooratures.
```

Exemples pratiques de procédures stockées

Les procédures stockées sont utilisées pour automatiser des tâches répétitives, exécuter des opérations complexes, améliorer la sécurité et la modularité du code SQL.



Performance

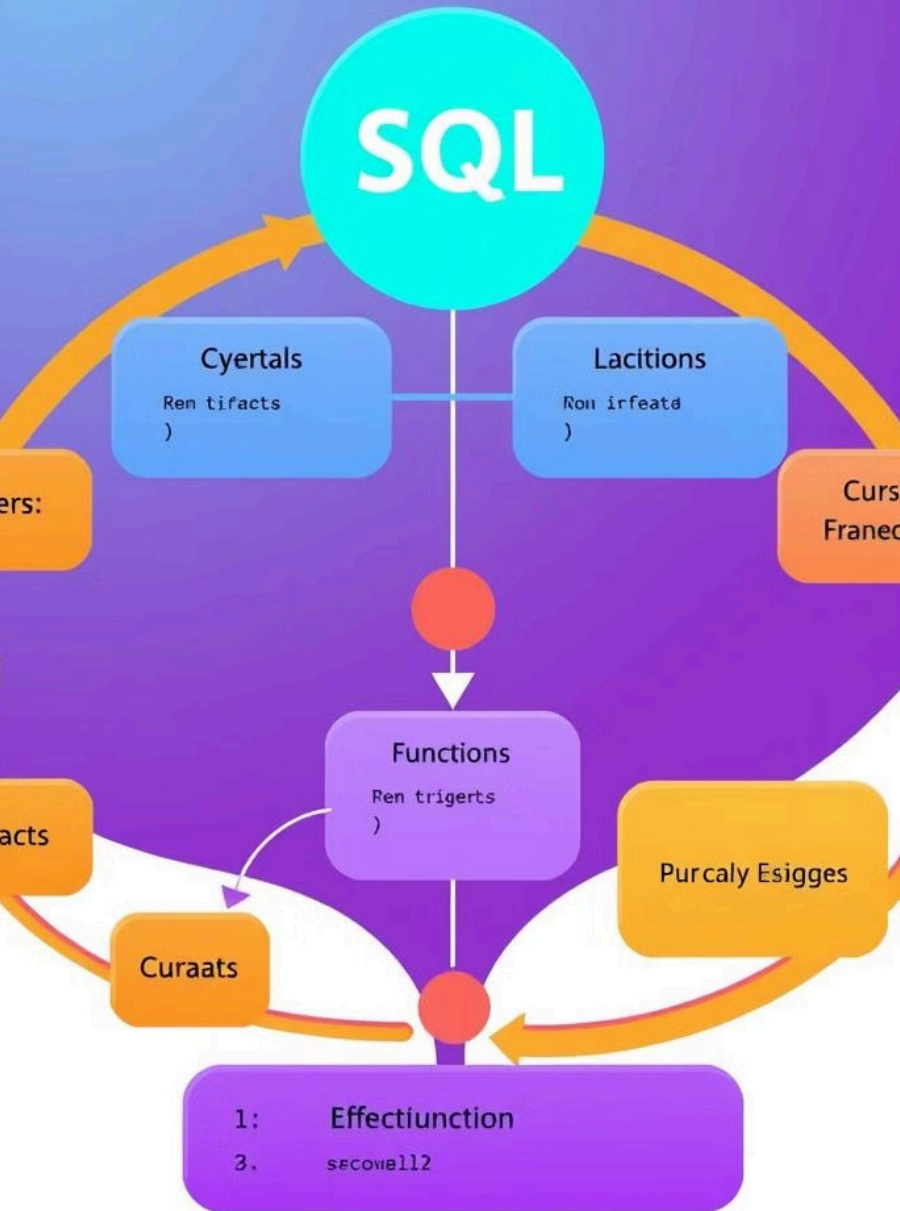
Optimisation des performances en exécutant des instructions SQL pré-compilées et en limitant le trafic réseau.

Sécurité

Amélioration de la sécurité en centralisant les accès aux données et en limitant les privilèges des utilisateurs.

Autres objets SQL avancés : Triggers, curseurs, etc.

SQL offre une variété d'objets avancés pour la gestion des données et l'automatisation des processus. Ces objets, tels que les triggers, les curseurs et les fonctions, permettent de réaliser des opérations complexes et de personnaliser le comportement de la base de données.



1

Triggers : Exécution automatique d'instructions SQL en réponse à des événements spécifiques dans la base de données.

2

Cursors : Permettent de traiter les données ligne par ligne, offrant une flexibilité pour les opérations complexes.

3

Fonctions : Unités de code SQL réutilisables qui effectuent des opérations spécifiques et retournent des valeurs.

Démonstrations et études de cas pratiques

Pour illustrer les concepts abordés, des démonstrations et des études de cas pratiques seront présentées. Ces exemples concrets permettront de comprendre les applications concrètes des vues, des procédures et des autres objets SQL avancés.