

Cours : Transactions avec MariaDB et DBeaver

 Propriétaire	 Marine
 Étiquettes	

Objectif

Apprendre à utiliser les transactions en SQL avec MariaDB, en s'appuyant sur DBeaver comme interface. Ce cours vous permettra de comprendre les principes des transactions, comment les gérer, et comment les appliquer dans vos bases de données

Introduction aux transactions

Une transaction est une série d'opérations SQL (insertion, mise à jour, suppression, etc.) qui sont exécutées comme une unité logique. Les transactions permettent de garantir que toutes les opérations sont effectuées avec succès ou qu'aucune d'entre elles ne l'est.

Les transactions suivent les principes ACID :

1. **Atomicité** : Toutes les opérations d'une transaction sont réalisées ou aucune ne l'est.
2. **Cohérence** : Une transaction valide maintient la base de données dans un état cohérent.
3. **Isolation** : Les transactions ne doivent pas interférer entre elles.
4. **Durabilité** : Une fois validée, une transaction est définitivement enregistrée.

Commandes principales

Voici les commandes SQL pour gérer les transactions dans MariaDB :

1. **START TRANSACTION** : Démarre une nouvelle transaction.

2. **COMMIT** : Valide et enregistre les modifications effectuées par la transaction.
3. **ROLLBACK** : Annule toutes les modifications effectuées depuis le début de la transaction.
4. **SAVEPOINT** : Définit un point de restauration intermédiaire dans une transaction.
5. **RELEASE SAVEPOINT** : Supprime un point de restauration.
6. **ROLLBACK TO SAVEPOINT** : Annule les opérations jusqu'à un point de restauration.

Prérequis

- **MariaDB** : Assurez-vous que votre serveur MariaDB est configuré pour accepter les transactions.
- **DBeaver** : Téléchargez et configurez [DBeaver](#) comme interface pour interagir avec MariaDB.

Configurer DBeaver pour MariaDB

1. Ouvrez DBeaver.
2. Cliquez sur **Fichier > Nouvelle connexion**.
3. Sélectionnez **MariaDB** dans la liste des bases de données.
4. Entrez les informations de connexion (hôte, port, utilisateur, mot de passe).
5. Testez la connexion et cliquez sur **Terminer**.

Exemples pratiques

1. Transaction de base

Voici un exemple pour illustrer l'utilisation des transactions dans MariaDB :

Scénario :

Vous gérez une base de données pour une bibliothèque, et vous devez enregistrer l'emprunt de livres tout en mettant à jour le stock.

Script SQL :

```

-- Démarrer une transaction
START TRANSACTION;

-- Insérer un emprunt
INSERT INTO emprunts (noLecteur, refLivre, dateEmprunt, eta
t)
VALUES (101, 'L003', CURDATE(), 'EN COURS');

-- Mettre à jour le stock de livres
UPDATE livres
SET stock = stock - 1
WHERE refLivre = 'L003';

-- Valider les changements
COMMIT;

```

Explication :

1. La transaction commence avec `START TRANSACTION`.
2. On insère un emprunt dans la table `emprunts`.
3. On met à jour le stock dans la table `livres`.
4. Si tout est correct, on valide les changements avec `COMMIT`.

2. Annuler une transaction

Scénario :

Lors d'un emprunt, une erreur est détectée (par exemple, le stock est insuffisant). Vous souhaitez annuler toutes les modifications.

Script SQL :

```

-- Démarrer une transaction
START TRANSACTION;

-- Tenter d'insérer un emprunt

```

```
INSERT INTO emprunts (noLecteur, refLivre, dateEmprunt, etat)
VALUES (102, 'L005', CURDATE(), 'EN COURS');

-- Tenter de mettre à jour le stock
UPDATE livres
SET stock = stock - 1
WHERE refLivre = 'L005';

-- Vérifier le stock restant
SELECT stock FROM livres WHERE refLivre = 'L005';

-- Annuler les modifications si une erreur est détectée
ROLLBACK;
```

Bonnes pratiques

1. Testez les préconditions avant de valider :

- Par exemple, vérifiez si le stock est suffisant avant d'insérer un emprunt.

2. Utilisez les transactions uniquement si nécessaire :

- Si toutes les opérations ne sont pas liées, évitez de les regrouper dans une transaction.

3. Utilisez `ROLLBACK` pour tester vos scripts :

- Lors de la phase de développement, utilisez `ROLLBACK` pour éviter de modifier la base de données réelle.