

TCL

Transaction Control Language

Introduction aux requêtes TCL

Les commandes TCL (Transaction Control Language) en SQL sont utilisées pour gérer les transactions dans une base de données. Les transactions sont des séquences d'opérations SQL qui sont exécutées comme une seule unité.

Les commandes TCL permettent de contrôler le début, la fin et le résultat de ces transactions.

Les commandes TCL les plus couramment utilisées sont BEGIN, COMMIT et ROLLBACK.

Exemple d'une transaction

Voici un exemple de transaction

```
-- Démarre une nouvelle transaction
```

```
BEGIN;
```

```
-- Effectue des opérations SQL (par exemple, INSERT, UPDATE, DELETE)
```

```
INSERT INTO clients (nom, prenom) VALUES ('Doe', 'John');
```

```
UPDATE commandes SET total = 100 WHERE client_id = 1;
```

```
-- Valide la transaction
```

```
COMMIT;
```

Mots clés

BEGIN: cette commande démarre une nouvelle transaction.

Toutes les opérations SQL exécutées après BEGIN font partie de la même transaction jusqu'à ce qu'une commande COMMIT ou ROLLBACK soit utilisée pour la terminer.

COMMIT: valide la transaction enregistrant toutes les modifications apportées à la base de données depuis le dernier BEGIN. Les modifications deviennent permanentes et sont visibles par d'autres utilisateurs de la base de données.

ROLLBACK: annule la transaction en abandonnant toutes les modifications apportées à la base de données depuis le dernier BEGIN.

La base de données est restaurée à l'état précédent la transaction.

Gestion des erreurs

Si une erreur se produit lors de l'exécution de la transaction ou si vous souhaitez annuler la transaction, vous pouvez utiliser la commande ROLLBACK à la place de COMMIT.

La commande ROLLBACK annule toutes les modifications apportées depuis le début de la transaction.

Les transactions sont essentielles pour garantir la cohérence et la fiabilité des données dans une base de données relationnelle.

Utilisation des points de sauvegarde

La clause SAVEPOINT définit un nouveau point de sauvegarde à l'intérieur de la transaction en cours.

Voici un exemple

```
BEGIN;  
  INSERT INTO table1 VALUES (1);  
  SAVEPOINT sp1;  
  INSERT INTO table1 VALUES (2);  
  ROLLBACK TO SAVEPOINT sp1;  
  INSERT INTO table1 VALUES (3);  
COMMIT;
```

La transaction ci-dessus insère 1 et 3, mais pas 2.

Utilisation des points de sauvegarde

Vous pouvez utiliser qu'un seul point de sauvegarde

```
BEGIN;
INSERT INTO table1 VALUES (1);
SAVEPOINT my_savepoint;
INSERT INTO table1 VALUES (2);
SAVEPOINT my_savepoint;
INSERT INTO table1 VALUES (3);
-- rollback jusqu'au deuxième point de sauvegarde
ROLLBACK TO SAVEPOINT my_savepoint;
SELECT * FROM table1;           -- affiche les lignes 1 et 2
-- libère le deuxième point de sauvegarde
RELEASE SAVEPOINT my_savepoint;
-- annule jusqu'au premier point de sauvegarde
ROLLBACK TO SAVEPOINT my_savepoint;
SELECT * FROM table1;           -- affiche seulement la ligne 1
COMMIT;
```