Лекция 11

Манипулирование данными (DML) DTL (Rollback, Commit, Savepoint)

Цель

Изучить основы манипулирования данными (DML) и управления транзакциями (DTL) в SQL, освоить использование команд COMMIT, ROLLBACK и SAVEPOINT для обеспечения целостности данных и управления транзакциями.

Основные вопросы

1. Понятие транзакции и ее свойства (ACID);
2. Команды манипулирования данными (DML): INSERT, UPDATE, DELETE;
3. Управление транзакциями (DTL): COMMIT, ROLLBACK, SAVEPOINT;
4. Уровни изоляции транзакций;
5. Практические примеры использования транзакций;

Лекция

Введение в манипулирование данными (DML)

Data Manipulation Language (DML) - подмножество SQL, используемое для манипулирования данными в базе данных. Основные команды DML:

- INSERT - добавление новых записей;
- UPDATE - изменение существующих записей;
- DELETE - удаление записей;
- SELECT - выборка данных (часто относят к DML);

Транзакции и их свойства

Транзакция - последовательность операций, выполняемых как одно целое. Транзакции должны соответствовать свойствам ACID:

- Атомарность (Atomicity) - транзакция выполняется полностью или не выполняется вовсе;
- Согласованность (Consistency) - транзакция переводит базу данных из одного согласованного состояния в другое;
- Изолированность (Isolation) - параллельные транзакции не влияют друг на друга;
- Долговечность (Durability) - результаты транзакции сохраняются после ее завершения;

Команды DML

Оператор INSERT

Добавление новых строк в таблицу.

Синтаксис:

```
INSERT INTO table_name (column1, column2, ...)

VALUES (value1, value2, ...);
```

Примеры:

```
-- Вставка одной строки
INSERT INTO Employees (EmployeeID, FirstName, LastName, Department)
VALUES (1, 'Иван', 'Иванов', 'IT');

-- Вставка нескольких строк
INSERT INTO Employees (EmployeeID, FirstName, LastName, Department)
VALUES
    (2, 'Петр', 'Петров', 'HR'),
    (3, 'Мария', 'Сидорова', 'Finance');

-- Вставка из результата SELECT
INSERT INTO FormerEmployees (EmployeeID, FirstName, LastName)
SELECT EmployeeID, FirstName, LastName
FROM Employees

WHERE TerminationDate IS NOT NULL;
```

Оператор UPDATE

Изменение существующих данных в таблице.

Синтаксис:

```
UPDATE table_name
SET column1 = value1, column2 = value2, ...

WHERE condition;
```

Примеры:

```
-- Обновление одной строки

UPDATE Employees
SET Salary = 50000, LastUpdated = GETDATE()
WHERE EmployeeID = 1;

-- Обновление нескольких строк
UPDATE Employees
SET Department = 'IT'
WHERE Department = 'Technology';

-- Обновление с использованием JOIN
UPDATE e
SET e.Salary = e.Salary * 1.1
FROM Employees e
JOIN Departments d ON e.DepartmentID = d.DepartmentID

WHERE d.DepartmentName = 'IT';
```

Оператор DELETE

Удаление строк из таблица.

Синтаксис:

```
DELETE FROM table_name

WHERE condition;
```

Примеры:

```
-- Удаление одной строки
```

```
DELETE FROM Employees
WHERE EmployeeID = 1;

-- Удаление нескольких строк
DELETE FROM Employees
WHERE Department = 'HR' AND TerminationDate IS NOT NULL;

-- Удаление всех строк (опасно!)
DELETE FROM TemporaryData;
```

Управление транзакциями (DTL)

Data Transaction Language (DTL) - команды для управления транзакциями:

- BEGIN TRANSACTION - начало транзакции
- COMMIT - подтверждение транзакции
- ROLLBACK - откат транзакции
- SAVEPOINT - точка сохранения внутри транзакции

Команда BEGIN TRANSACTION

Начинает новую транзакцию.

```
BEGIN TRANSACTION;
-- или
BEGIN TRAN;
```

Команда COMMIT

Подтверждает транзакцию, сохраняя все изменения.

```
COMMIT TRANSACTION;
-- или
COMMIT;
```

Команда ROLLBACK

Откатывает транзакцию, отменяя все изменения.

```
ROLLBACK TRANSACTION;
```

```
-- или

ROLLBACK;
```

Команда SAVEPOINT

Создает точку сохранения внутри транзакции, к которой можно откатиться.

```
SAVE TRANSACTION savepoint_name;
```

Практические примеры использования транзакций

Пример 1: Простая транзакция

```
BEGIN TRANSACTION;

UPDATE Accounts
SET Balance = Balance - 1000
WHERE AccountID = 1;

UPDATE Accounts
SET Balance = Balance + 1000
WHERE AccountID = 2;


COMMIT TRANSACTION;
```

Пример 2: Транзакция с обработкой ошибок

```
BEGIN TRY
   BEGIN TRANSACTION;

   -- Операция 1
   UPDATE Inventory
   SET Quantity = Quantity - 10
   WHERE ProductID = 101;

   -- Операция 2
   INSERT INTO Sales (ProductID, Quantity, SaleDate)
   VALUES (101, 10, GETDATE());

   COMMIT TRANSACTION;
```

```
END TRY
BEGIN CATCH
   ROLLBACK TRANSACTION;

   DECLARE @ErrorMessage NVARCHAR(4000) = ERROR_MESSAGE();
   RAISERROR('Ошибка при выполнении операции: %s', 16, 1,
@ErrorMessage);

END CATCH;
```

Пример 3: Транзакция с точками сохранения

```
BEGIN TRANSACTION;

-- Операция 1
UPDATE Accounts SET Balance = Balance - 500 WHERE AccountID = 1;
SAVE TRANSACTION AfterFirstUpdate;

-- Операция 2
UPDATE Accounts SET Balance = Balance + 500 WHERE AccountID = 2;
SAVE TRANSACTION AfterSecondUpdate;

-- Проверка условий
IF (SELECT Balance FROM Accounts WHERE AccountID = 1) < 0
BEGIN
   ROLLBACK TRANSACTION AfterFirstUpdate;
   PRINT 'Недостаточно средств. Откат до первой операции.';
END
ELSE
BEGIN
   COMMIT TRANSACTION;
   PRINT 'Транзакция успешно завершена.';

END
```

Уровни изоляции транзакций

Уровень изоляции определяет степень изолированности транзакций друг от друга.

Чтение неподтвержденных данных (Read Uncommitted)

- Самый низкий уровень изоляции;
- Возможны "грязные" чтения;

Чтение подтвержденных данных (Read Committed)

- Стандартный уровень для большинства СУБД;
- Исключает "грязные" чтения;

Повторяемое чтение (Repeatable Read)

- Гарантирует, что данные, прочитанные в транзакции, не изменятся;

Сериализуемость (Serializable)

- Самый высокий уровень изоляции;
- Полная изоляция транзакций;

Установка уровня изоляции:

```
SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
```

Контрольные вопросы

1. Что такое транзакция и какими свойствами она обладает?
2. Какие команды относятся к DML и для чего они используются?
3. В чем разница между COMMIT и ROLLBACK?
4. Что такое SAVEPOINT и как он используется?
5. Какие уровни изоляции транзакций вы знаете?
6. Приведите пример использования транзакции с обработкой ошибок.

Литература

1. Дейт К. Дж. Введение в системы баз данных. - Глава 15.
2. Коннолли Т., Бегг К. Базы данных: проектирование, реализация и сопровождение. - Глава 22.
3. Microsoft SQL Server Documentation: Transactions
4. PostgreSQL Documentation: Transaction Management