

Лекция 12 - DDL и DML операторы

DDL (Data Definition Language) и DML (Data Manipulation Language) являются двумя категориями операторов в SQL, каждая из которых служит для разных целей при работе с базами данных.

DDL (Data Definition Language)

DDL включает в себя операторы, которые определяют структуру базы данных и управляют объектами базы данных (таблицы, индексы, схемы и т.д.).

Основные операторы DDL:

CREATE – Создает новые объекты в базе данных, такие как таблицы, индексы, функции, триггеры, и схемы.

Например:

```
CREATE TABLE students (  
    student_id INT PRIMARY KEY,  
    name VARCHAR(100),  
    age INT  
);
```

ALTER – изменение структуры существующих таблиц, например, добавление, удаление или изменение атрибутов. Пример:

```
ALTER TABLE students ADD email VARCHAR(255);
```

DROP – удаление таблиц, баз данных, и других объектов. Например:

```
DROP TABLE students;
```

TRUNCATE – удаление всех строк в таблице, не удаляя саму таблицу. Это быстрее, чем удаление строк оператором DELETE:

```
TRUNCATE TABLE students;
```

DML (Data Manipulation Language)

DML включает операторы, которые используются для манипуляции данными внутри таблиц (добавление, изменение, удаление данных).

Основные операторы DML:

SELECT – извлечение данных из базы данных. Можно использовать различные условия, сортировки, группирования и функции. Пример:

```
SELECT name, age FROM students WHERE age > 18 ORDER BY name;
```

INSERT – добавление новых данных в таблицу. Пример:

```
INSERT INTO students (student_id, name, age) VALUES (1, 'Alice', 20);
```

UPDATE – обновление существующих данных. Пример:

```
UPDATE students SET age = 21 WHERE student_id = 1;
```

Обновление нескольких полей:

```
UPDATE students SET age = 20, email = 'new.email@example.com' WHERE student_id = 1;
```

DELETE – удаление данных из таблицы. Пример:

```
DELETE FROM students WHERE student_id = 1;
```

Удаление с условием:

```
DELETE FROM students WHERE age < 18;
```

Важные моменты

- DDL операторы изменяют структуру базы данных и влияют на метаданные. Эти изменения часто необратимы (например, DROP TABLE).
- DML операторы влияют на данные, но не на структуру таблиц. Их действия можно отменить с помощью транзакций.
- Использование DDL и DML операторов требует понимания текущей структуры базы данных и планирования изменений, чтобы не потерять данные и не снизить производительность системы.

Задача 1: Основы DDL

Цель: Создать базу данных для учебного заведения.

Задание:

1. Создайте базу данных School.
2. В базе данных School, создайте таблицу Students со следующими полями:
3. StudentID (целочисленный тип, первичный ключ, автоинкрементный)
4. FirstName (строковый тип, максимум 50 символов, не может быть NULL)
5. LastName (строковый тип, максимум 50 символов, не может быть NULL)
6. BirthDate (дата, может быть NULL)
7. Email (строковый тип, уникальный)
8. Создайте таблицу Courses с полями:
9. CourseID (целочисленный тип, первичный ключ, автоинкрементный)
10. CourseName (строковый тип, максимум 100 символов)
11. TeacherName (строковый тип, максимум 100 символов)
12. Добавьте в таблицу Students столбец PhoneNumber (строковый тип, может быть NULL).

Задача 2: Операции DML

Цель: Манипулировать данными в таблицах Students и Courses.

Задание:

1. Вставьте в таблицу Students минимум 5 записей.
2. Вставьте в таблицу Courses минимум 3 записи.
3. Напишите запрос для выбора всех студентов, чья фамилия начинается на 'S'.
4. Обновите данные, установив PhoneNumber для одного из студентов.
5. Удалите студента, у которого нет указанного Email.

Задача 3: Продвинутый SQL

Цель: Использовать сложные запросы, включая соединения и подзапросы.

Задание:

1. Создайте таблицу Enrollments, которая включает поля:
2. EnrollmentID (целочисленный тип, первичный ключ, автоинкрементный)
3. StudentID (целочисленный тип, внешний ключ к Students)
4. CourseID (целочисленный тип, внешний ключ к Courses)
5. EnrollmentDate (дата)
6. Вставьте данные в таблицу Enrollments, связывая студентов с курсами.
7. Напишите запрос для выбора всех курсов, на которые записан определенный студент.
8. Напишите запрос, который выбирает всех студентов, не записанных ни на один курс.
9. Создайте представление V_StudentCourses, которое показывает FirstName, LastName и CourseName каждого студента.