

## **Лекция 12**

### **Создание таблиц и управление ими (DDL). Включение ограничений (Constraints)**

#### **Цель**

Изучить язык определения данных (DDL) для создания и управления таблицами, освоить использование ограничений (Constraints) для обеспечения целостности данных.

#### **Основные вопросы**

1. Команды DDL: CREATE, ALTER, DROP;
2. Типы данных в SQL;
3. Ограничения целостности: PRIMARY KEY, FOREIGN KEY, UNIQUE, CHECK, NOT NULL;
4. Создание и изменение таблиц;
5. Управление ограничениями;

#### **Лекция**

#### **Введение в DDL (Data Definition Language)**

DDL - подмножество SQL, используемое для определения и управления структурой базы данных.

Основные команды DDL:

- CREATE - создание объектов базы данных;
- ALTER - изменение объектов базы данных;
- DROP - удаление объектов базы данных;
- TRUNCATE - удаление всех данных из таблицы;

#### **Типы данных в SQL**

#### **Строковые типы данных**

- CHAR(n) - строка фиксированной длины;
- VARCHAR(n) - строка переменной длины;
- TEXT - большой текстовый объект;

### Числовые типы данных

- INT - целое число;
- DECIMAL(p,s) - точное числовое значение;
- FLOAT - приблизительное числовое значение;

### Типы даты и времени

- DATE - дата;
- TIME - время;
- DATETIME - дата и время;

### Другие типы

- BOOLEAN - логический тип;
- BLOB - бинарные данные;

### Создание таблиц

#### Команда CREATE TABLE

Синтаксис:

```
CREATE TABLE table_name (  
    column1 datatype constraint,  
    column2 datatype constraint,  
    ...  
);
```

#### Пример создания простой таблицы:

```
CREATE TABLE Employees (  
    EmployeeID INT PRIMARY KEY,  
    FirstName VARCHAR(50) NOT NULL,  
    LastName VARCHAR(50) NOT NULL,  
    BirthDate DATE,  
    Salary DECIMAL(10,2),  
    DepartmentID INT
```

```
);
```

## Ограничения целостности (Constraints)

Ограничения - правила, обеспечивающие целостность данных.

### Ограничение NOT NULL

Запрещает значение NULL в столбце.

```
CREATE TABLE Products (  
    ProductID INT NOT NULL,  
    ProductName VARCHAR(100) NOT NULL
```

```
);
```

### Ограничение UNIQUE

Гарантирует уникальность значений в столбце.

```
CREATE TABLE Customers (  
    CustomerID INT NOT NULL,  
    Email VARCHAR(100) UNIQUE,  
    Phone VARCHAR(20) UNIQUE
```

```
);
```

### Ограничение PRIMARY KEY

Уникально идентифицирует каждую строку в таблице.

```
CREATE TABLE Orders (  
    OrderID INT PRIMARY KEY,  
    OrderDate DATE NOT NULL,  
    CustomerID INT NOT NULL
```

```
);
```

### Ограничение FOREIGN KEY

Обеспечивает ссылочную целостность между таблицами.

```
sql
```

```
CREATE TABLE Orders (  
    OrderID INT PRIMARY KEY,  
    CustomerID INT NOT NULL,  
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)  
);
```

## Ограничение CHECK

Проверяет значения по заданному условию.

```
CREATE TABLE Employees (  
    EmployeeID INT PRIMARY KEY,  
    Salary DECIMAL(10,2) CHECK (Salary > 0),  
    Age INT CHECK (Age >= 18 AND Age <= 65)  
);
```

## Ограничение DEFAULT

Устанавливает значение по умолчанию.

```
CREATE TABLE Orders (  
    OrderID INT PRIMARY KEY,  
    OrderDate DATE DEFAULT GETDATE(),  
    Status VARCHAR(20) DEFAULT 'Pending'  
);
```

## Изменение таблиц

### Команда ALTER TABLE

#### Добавление столбца:

```
ALTER TABLE Employees
```

```
ADD Email VARCHAR(100);
```

#### Удаление столбца:

```
ALTER TABLE Employees
```

```
DROP COLUMN Email;
```

### Изменение столбца:

```
ALTER TABLE Employees
```

```
ALTER COLUMN FirstName VARCHAR(100);
```

### Добавление ограничения:

```
ALTER TABLE Employees
```

```
ADD CONSTRAINT CHK_Salary CHECK (Salary > 0);
```

### Удаление ограничения:

```
ALTER TABLE Employees
```

```
DROP CONSTRAINT CHK_Salary;
```

### Удаление таблиц

#### Команда DROP TABLE

Удаляет таблицу и все ее данные.

```
DROP TABLE Employees;
```

#### Команда TRUNCATE TABLE

Удаляет все данные из таблицы, но сохраняет ее структуру.

```
TRUNCATE TABLE TemporaryData;
```

### Практические примеры

#### Пример 1: Создание сложной таблицы с ограничениями

```
CREATE TABLE Employees (  
    EmployeeID INT IDENTITY(1,1) PRIMARY KEY,  
    FirstName VARCHAR(50) NOT NULL,  
    LastName VARCHAR(50) NOT NULL,  
    Email VARCHAR(100) UNIQUE NOT NULL,
```

```

BirthDate DATE NOT NULL,
HireDate DATE NOT NULL,
Salary DECIMAL(10,2) NOT NULL CHECK (Salary > 0),
DepartmentID INT NOT NULL,
ManagerID INT,
CONSTRAINT CHK_BirthDate CHECK (BirthDate <= DATEADD(YEAR,
-18, GETDATE())),
CONSTRAINT CHK_HireDate CHECK (HireDate >= BirthDate),
CONSTRAINT FK_Department FOREIGN KEY (DepartmentID)
REFERENCES Departments(DepartmentID),
CONSTRAINT FK_Manager FOREIGN KEY (ManagerID) REFERENCES
Employees(EmployeeID)
);

```

## Пример 2: Создание таблицы с составным первичным ключом

```

CREATE TABLE OrderDetails (
    OrderID INT NOT NULL,
    ProductID INT NOT NULL,
    Quantity INT NOT NULL CHECK (Quantity > 0),
    UnitPrice DECIMAL(10,2) NOT NULL CHECK (UnitPrice > 0),
    PRIMARY KEY (OrderID, ProductID),
    FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
);

```

## Пример 3: Изменение существующей таблицы

*-- Добавление нового столбца*

```

ALTER TABLE Employees
ADD PhoneNumber VARCHAR(15);

```

*-- Добавление ограничения на новый столбец*

```

ALTER TABLE Employees
ADD CONSTRAINT CHK_PhoneNumber
CHECK (PhoneNumber LIKE '+7([0-9][0-9][0-9])[0-9][0-9][0-9]-[0-9][0-9]-[0-9]
[0-9]');

```

*-- Создание индекса на столбце*

```
CREATE INDEX IX_Employees_LastName  
ON Employees (LastName);
```

*-- Удаление индекса*

```
DROP INDEX IX_Employees_LastName ON Employees;
```

### **Контрольные вопросы**

1. Какие команды относятся к DDL и для чего они используются?
2. Что такое ограничения целостности и какие типы ограничений вы знаете?
3. В чем разница между PRIMARY KEY и UNIQUE ограничениями?
4. Как создать внешний ключ между двумя таблицами?
5. Каким образом можно изменить структуру существующей таблицы?
6. Приведите пример создания таблицы с различными типами ограничений.

### **Литература**

1. Дейт К. Дж. Введение в системы баз данных. - Глава 4.
2. Коннолли Т., Бегг К. Базы данных: проектирование, реализация и сопровождение. - Глава 5.
3. Microsoft SQL Server Documentation: CREATE TABLE
4. PostgreSQL Documentation: Data Definition