

# Шпаргалка по DDL-запросам в MySQL

## 1. CREATE - Создание объектов

### Создание базы данных

```
-- Простое создание БД
CREATE DATABASE shop;

-- Создание БД с указанием кодировки
CREATE DATABASE shop
CHARACTER SET utf8mb4
COLLATE utf8mb4_unicode_ci;

-- Создание БД, если она не существует
CREATE DATABASE IF NOT EXISTS shop;
```

### Создание таблицы

```
-- Простая таблица
CREATE TABLE products (
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100) NOT NULL,
    price DECIMAL(10,2)
);

-- Таблица с внешним ключом
CREATE TABLE orders (
    order_id INT PRIMARY KEY AUTO_INCREMENT,
    user_id INT NOT NULL,
    total_amount DECIMAL(10,2),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(id)
);

-- Таблица с составным ключом
CREATE TABLE order_items (
    order_id INT,
    product_id INT,
    quantity INT NOT NULL,
    price DECIMAL(10,2) NOT NULL,
    PRIMARY KEY (order_id, product_id),
    FOREIGN KEY (order_id) REFERENCES orders(order_id),
    FOREIGN KEY (product_id) REFERENCES products(id)
);
```

## 2. ALTER - Изменение структуры

### Изменение таблицы

```
-- Добавление столбца
ALTER TABLE products
ADD COLUMN description TEXT;

-- Добавление столбца с позицией
ALTER TABLE products
ADD COLUMN category_id INT AFTER name;

-- Удаление столбца
ALTER TABLE products
DROP COLUMN description;

-- Изменение типа столбца
ALTER TABLE products
MODIFY COLUMN price DECIMAL(12,2);

-- Переименование столбца
ALTER TABLE products
CHANGE COLUMN name product_name VARCHAR(150);

-- Добавление внешнего ключа
ALTER TABLE products
ADD FOREIGN KEY (category_id) REFERENCES categories(id);

-- Добавление первичного ключа
ALTER TABLE products
ADD PRIMARY KEY (id);
```

### Изменение базы данных

```
-- Изменение кодировки
ALTER DATABASE shop
CHARACTER SET = utf8mb4
COLLATE = utf8mb4_unicode_ci;
```

## 3. DROP - Удаление объектов

```
-- Удаление таблицы
DROP TABLE products;

-- Удаление таблицы, если она существует
DROP TABLE IF EXISTS products;

-- Удаление базы данных
DROP DATABASE shop;

-- Удаление индекса
DROP INDEX idx_name ON products;
```

## 4. TRUNCATE - Очистка таблицы

```
-- Очистка таблицы (удаление всех записей)
TRUNCATE TABLE products;
```

## 5. Управление пользователями

```
-- Создание пользователя
CREATE USER 'username'@'localhost'
IDENTIFIED BY 'password123';

-- Предоставление всех прав на базу данных
GRANT ALL PRIVILEGES ON database_name.*
TO 'username'@'localhost';

-- Предоставление конкретных прав
GRANT SELECT, INSERT, UPDATE ON database_name.*
TO 'username'@'localhost';

-- Применение изменений прав
FLUSH PRIVILEGES;

-- Отзыв всех прав
REVOKE ALL PRIVILEGES ON database_name.*
FROM 'username'@'localhost';

-- Удаление пользователя
DROP USER 'username'@'localhost';
```

## 6. Часто используемые ограничения (CONSTRAINTS)

## NOT NULL - запрет на NULL значения

```
-- При создании таблицы
CREATE TABLE customers (
    id INT AUTO_INCREMENT PRIMARY KEY,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    email VARCHAR(100) NOT NULL
);

-- Добавление ограничения к существующему столбцу
ALTER TABLE customers
MODIFY COLUMN phone VARCHAR(15) NOT NULL;

-- Удаление ограничения
ALTER TABLE customers
MODIFY COLUMN phone VARCHAR(15) NULL;
```

## UNIQUE - уникальные значения

```
-- При создании таблицы
CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(50) UNIQUE,
    email VARCHAR(100) UNIQUE
);

-- Добавление ограничения к существующему столбцу
ALTER TABLE users
ADD CONSTRAINT unique_email UNIQUE (email);

-- Создание составного уникального ограничения
ALTER TABLE employees
ADD CONSTRAINT uk_dept_emp UNIQUE (department_id, employee_id);

-- Удаление ограничения
ALTER TABLE users
DROP INDEX unique_email;
```

## DEFAULT - значение по умолчанию

```

-- При создании таблицы
CREATE TABLE products (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    price DECIMAL(10,2) NOT NULL,
    status ENUM('active', 'inactive') DEFAULT 'active',
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

-- Добавление значения по умолчанию
ALTER TABLE products
ALTER COLUMN in_stock SET DEFAULT TRUE;

-- Удаление значения по умолчанию
ALTER TABLE products
ALTER COLUMN in_stock DROP DEFAULT;

```

## CHECK - проверка условия (MySQL 8.0+)

```

-- При создании таблицы
CREATE TABLE products (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    price DECIMAL(10,2) CHECK (price > 0),
    quantity INT CHECK (quantity >= 0)
);

-- Добавление ограничения к существующему столбцу
ALTER TABLE products
ADD CONSTRAINT chk_price CHECK (price > 0);

-- Удаление ограничения
ALTER TABLE products
DROP CHECK chk_price;

```

## PRIMARY KEY - первичный ключ

```
-- При создании таблицы
CREATE TABLE categories (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(50) NOT NULL
);

-- Составной первичный ключ
CREATE TABLE order_items (
    order_id INT,
    product_id INT,
    quantity INT NOT NULL,
    PRIMARY KEY (order_id, product_id)
);

-- Добавление первичного ключа к существующей таблице
ALTER TABLE suppliers
ADD PRIMARY KEY (supplier_id);

-- Удаление первичного ключа
ALTER TABLE suppliers
DROP PRIMARY KEY;
```

## FOREIGN KEY - внешний ключ

```

-- При создании таблицы
CREATE TABLE orders (
    id INT AUTO_INCREMENT PRIMARY KEY,
    customer_id INT NOT NULL,
    order_date DATE NOT NULL,
    FOREIGN KEY (customer_id) REFERENCES customers(id)
);

-- Добавление внешнего ключа с опциями ON DELETE и ON UPDATE
CREATE TABLE order_items (
    id INT AUTO_INCREMENT PRIMARY KEY,
    order_id INT,
    product_id INT,
    quantity INT NOT NULL,
    FOREIGN KEY (order_id) REFERENCES orders(id) ON DELETE CASCADE,
    FOREIGN KEY (product_id) REFERENCES products(id) ON UPDATE RESTRICT
);

-- Добавление внешнего ключа к существующей таблице
ALTER TABLE orders
ADD CONSTRAINT fk_customer
FOREIGN KEY (customer_id) REFERENCES customers(id);

-- Удаление внешнего ключа
ALTER TABLE orders
DROP FOREIGN KEY fk_customer;

```

## AUTO\_INCREMENT - автоинкремент

```

-- При создании таблицы
CREATE TABLE products (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL
);

-- Изменение начального значения автоинкремента
ALTER TABLE products AUTO_INCREMENT = 1000;

-- Проверка текущего значения автоинкремента
SELECT AUTO_INCREMENT
FROM information_schema.TABLES
WHERE TABLE_SCHEMA = 'your_database'
AND TABLE_NAME = 'products';

```

## ENUM - перечисление допустимых значений

```

-- При создании таблицы
CREATE TABLE tickets (
    id INT AUTO_INCREMENT PRIMARY KEY,
    title VARCHAR(100) NOT NULL,
    priority ENUM('low', 'medium', 'high', 'critical') DEFAULT 'medium',
    status ENUM('open', 'in_progress', 'resolved', 'closed') NOT NULL DEFAULT
'open'
);

-- Добавление ENUM к существующей таблице
ALTER TABLE products
MODIFY COLUMN size ENUM('S', 'M', 'L', 'XL') NOT NULL;

-- Изменение ENUM добавлением новых значений
ALTER TABLE tickets
MODIFY COLUMN priority ENUM('low', 'medium', 'high', 'critical', 'blocker')
NOT NULL;

```

## Полезные заметки:

1. После выполнения DDL-команд изменения нельзя отменить через ROLLBACK
2. Перед удалением таблицы убедитесь, что сделали резервную копию данных
3. При создании таблиц рекомендуется всегда указывать кодировку
4. Используйте IF EXISTS/IF NOT EXISTS для предотвращения ошибок
5. При работе с внешними ключами сначала создавайте родительские таблицы
6. Индексы ускоряют поиск, но замедляют вставку и обновление данных
7. Для внешних ключей доступны опции:
  - ON DELETE: CASCADE, SET NULL, RESTRICT, NO ACTION, SET DEFAULT
  - ON UPDATE: CASCADE, SET NULL, RESTRICT, NO ACTION, SET DEFAULT

## Часто используемые типы данных:

- INT: целые числа
- DECIMAL(M,N): десятичные числа
- VARCHAR(N): строки переменной длины
- TEXT: длинные текстовые данные
- DATE: дата
- TIMESTAMP: дата и время



- BOOLEAN: логический тип
- ENUM: список допустимых значений