**Создание базы данных с использованием SQL**

**Создание таблицы в базе данных:**

CREATE TABLE users (

id INT PRIMARY KEY,

name VARCHAR(50),

email VARCHAR(50)

);

**Добавление данных в таблицу:**

INSERT INTO users (id, name, email) VALUES (1, 'John Doe', 'john.doe@example.com');

INSERT INTO users (id, name, email) VALUES (2, 'Jane Smith', 'jane.smith@example.com');

**Выбор данных из таблицы:**

SELECT \* FROM users;

**Обновление данных в таблице:**

UPDATE users SET email = 'john.new@example.com' WHERE id = 1;

**Удаление данных из таблицы:**

DELETE FROM users WHERE id = 2;

**Удаление таблицы:**

DROP TABLE users;

**Примеры действий с базой данных на языке SQL**

1. **Создание новой базы данных**:

CREATE DATABASE mydatabase;

1. **Использование базы данных**:

USE mydatabase;

1. **Создание таблицы с внешним ключом**:

CREATE TABLE orders (

order\_id INT PRIMARY KEY,

user\_id INT,

total\_amount DECIMAL(10, 2),

FOREIGN KEY (user\_id) REFERENCES users(id)

);

1. **Добавление данных в таблицу**:

INSERT INTO orders (order\_id, user\_id, total\_amount) VALUES (1, 1, 100.50);

1. **Выборка данных из нескольких таблиц (JOIN)**:

SELECT users.name, orders.total\_amount

FROM users

JOIN orders ON users.id = orders.user\_id;

1. **Обновление данных в таблице**:

UPDATE users SET email = 'john.doe@example.org' WHERE id = 1;

1. **Удаление данных из таблицы**:

DELETE FROM orders WHERE order\_id = 1;

**Заключение**

Язык SQL является мощным инструментом для работы с базами данных. Создание таблиц, добавление, обновление и удаление данных позволяют эффективно управлять информацией. При использовании SQL важно следить за синтаксисом запросов и обеспечивать безопасность данных, особенно при удалении или обновлении записей.