**ФЕДЕРАЛЬНОЕ АГЕНТСТВО ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА**

ФедеральноеГосударственное Бюджетное Образовательное Учреждение

Высшего Образования

**«Петербургский Государственный Университет Путей Сообщения**

**Императора Александра I»**

**(ФГБОУ ВО ПГУПС)**

Факультет «Автоматизация и интеллектуальные технологии»

Кафедра «Информационные и вычислительные системы»

**ЛАБОРАТОРНАЯ РАБОТА №7**

по дисциплине

**«Программирование»**

ТЕМА РАБОТЫ

**«Создание таблиц и выполнение основных SQL-запросов. Управление данными (вставка, обновление удаление). Выполнение сложных SQL-запросов. Использование подзапросов и объединение таблиц»**

Выполнил:

|  |  |  |  |
| --- | --- | --- | --- |
| Обучающийся группы ИВБ-417 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись, дата | | М.Д. Карванен\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  И. О. Фамилия |
| Исправить замечания: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |  | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись, дата | к.в.н. доцент В.Е.Петров\_\_\_\_\_\_\_  учёное звание, И.О. Фамилия |
| Защита: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Зачтено/не зачтено | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись, дата | | к.в.н. доцент В.Е.Петров\_\_\_\_\_  учёное звание, И.О. Фамилия |

**Санкт-Петербург**

**2025г.**

**Целями** (целью) выполнения лабораторной работы являются:

**Вариант индивидуального задания**

**Задание:**

1. Теоретическая часть.

Поясните концепцию ключей, которая является ключевым элементом реляционной модели данных. Основные типы ключей.

Раскройте основные принципы реляционной модели данных.

1. Практическая часть.

А) Создайте таблицу студентов и выполните запрос для выбора студентов с оценкой выше 4.

Б) Создайте таблицу товаров и выполните запрос для сортировки по цене.

В) Напишите запрос для объединения данных из таблиц клиентов и заказов.

Г) Создайте таблицу работников и выполните запрос для обновления их должностей.

Д) Напишите запрос для удаления всех заказов, сделанных более года назад.

Е) Создайте таблицу заказов и выполните запрос для подсчета количества заказов каждого клиента.

Ж) Напишите запрос для выборки всех товаров, у которых цена выше среднего.

З) Создайте таблицу с информацией о книгах и выполните запрос для нахождения всех книг конкретного товара

И) Напишите запрос для создания новой таблицы с объединенными данными из нескольких других таблиц.

**Выполнение задания**

1. **Теоретическая часть**

***Поясните концепцию ключей, которая является ключевым элементом реляционной модели данных. Основные типы ключей.***

Ключи являются фундаментальным элементом реляционной модели данных, обеспечивая уникальность записей и устанавливая связи между таблицами. Они играют важную роль в поддержании целостности данных и организации структуры базы данных.

Основные типы ключей:

1. Первичный ключ (Primary Key)
2. Внешний ключ (Foreign Key)
3. Кандидатный ключ (Candidate Key)
4. Составной ключ (Composite Key)

***Раскройте основные принципы реляционной модели данных.***

Реляционная модель данных основана на строгих математических принципах, обеспечивающих логическую целостность и эффективное управление данными.

1. **Практическая часть**

***А)***

**Запрос создания таблицы:**

CREATE TABLE "students" (

"id" INTEGER,

"name" TEXT NOT NULL,

"group\_name" TEXT,

"grade" REAL CHECK(grade BETWEEN 1 AND 5),

PRIMARY KEY("id" AUTOINCREMENT)

)

**Запрос на заполнение данных:**

INSERT INTO students (name, group\_name, grade) VALUES

('Иванов Иван', 'Группа 101', 4.5),

('Петрова Мария', 'Группа 102', 3.8),

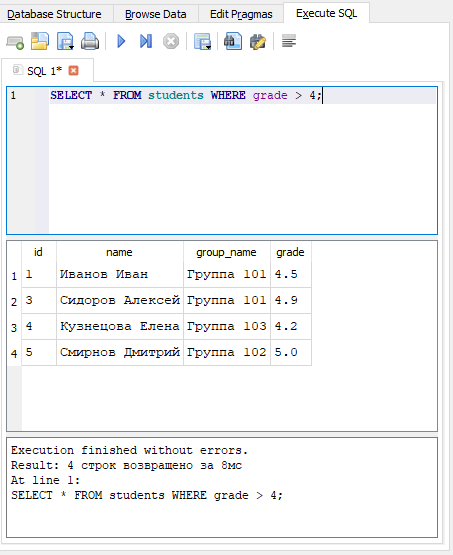
('Сидоров Алексей', 'Группа 101', 4.9),

('Кузнецова Елена', 'Группа 103', 4.2),

('Смирнов Дмитрий', 'Группа 102', 5.0);

**Запрос на выборку студентов с оценкой выше 4:**

SELECT \* FROM students WHERE grade > 4;

****

***Б)***

**Создание таблицы товаров:**

CREATE TABLE "products" (

"id" INTEGER,

"name" TEXT NOT NULL,

"price" REAL NOT NULL,

"category" TEXT,

PRIMARY KEY("id" AUTOINCREMENT)

);

**Запрос на заполнение таблицы данными:**

INSERT INTO products (name, price, category) VALUES

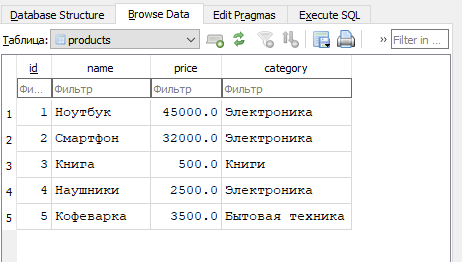
('Ноутбук', 45000, 'Электроника'),

('Смартфон', 32000, 'Электроника'),

('Книга', 500, 'Книги'),

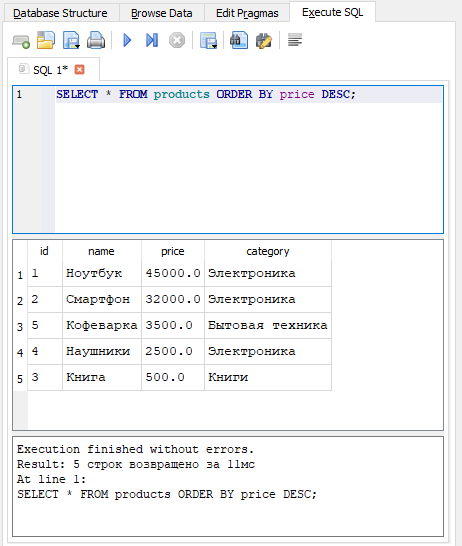
('Наушники', 2500, 'Электроника'),

('Кофеварка', 3500, 'Бытовая техника');



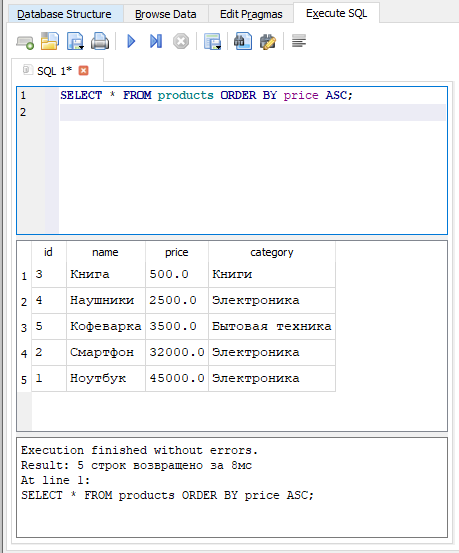
**Запрос на сортировку по цене (по возрастанию):**

SELECT \* FROM products ORDER BY price DESC;



**Запрос на сортировку по цене (по убыванию):**

SELECT \* FROM products ORDER BY price ASC;



***В)***

**Запрос на создание таблиц с клиентами и заказами, и их заполнение данными**CREATE TABLE "customers" (

"id" INTEGER,

"name" TEXT NOT NULL,

"email" TEXT,

"phone" TEXT,

PRIMARY KEY("id" AUTOINCREMENT)

);

CREATE TABLE orders (

id INTEGER PRIMARY KEY AUTOINCREMENT,

customer\_id INTEGER,

order\_date TEXT,

amount REAL,

FOREIGN KEY (customer\_id) REFERENCES customers(id)

)

INSERT INTO customers (name, email, phone) VALUES

('ООО "Ромашка"', 'romashka@mail.com', '+79151234567'),

('ИП Сидоров', 'sidorov@mail.com', '+79059876543'),

('ЗАО "Вектор"', 'vector@mail.com', '+74951234567');

INSERT INTO orders (customer\_id, order\_date, amount) VALUES

(1, '2023-05-15', 12000),

(2, '2023-06-20', 8500),

(1, '2023-07-10', 23500),

(3, '2023-08-05', 15000);

**Запрос на объединение таблиц через INNER JOIN:**

SELECT

c.name AS customer\_name,

c.email,

o.order\_date,

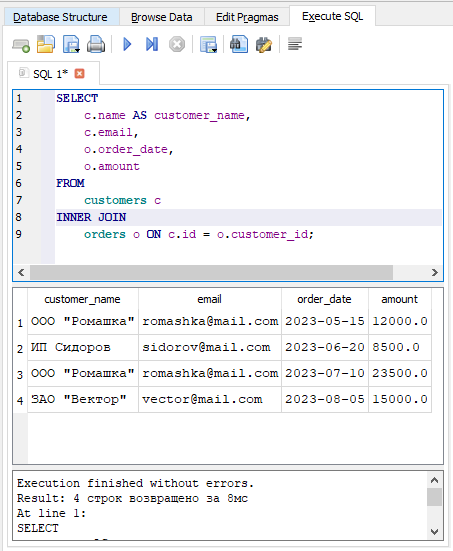
o.amount

FROM

customers c

INNER JOIN

orders o ON c.id = o.customer\_id;



***Г)***

**Создание таблицы работников и заполнение её данными:**

CREATE TABLE "employees" (

"id" INTEGER,

"name" TEXT NOT NULL,

"position" TEXT,

"department" TEXT,

"salary" REAL,

PRIMARY KEY("id" AUTOINCREMENT)

);

INSERT INTO employees (name, position, department, salary) VALUES

('Иванов И.И.', 'Менеджер', 'Продажи', 50000),

('Петрова П.П.', 'Разработчик', 'IT', 80000),

('Сидоров С.С.', 'Аналитик', 'Маркетинг', 60000),

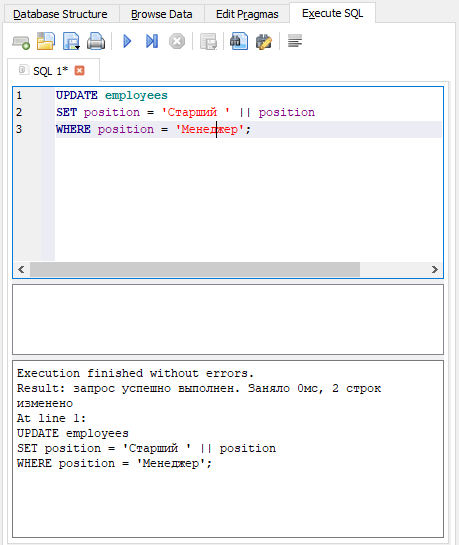
('Кузнецова Е.В.', 'Менеджер', 'Продажи', 55000);

**Запрос на обновление данных (Менеджеров повышаем до старших менеджеров):**

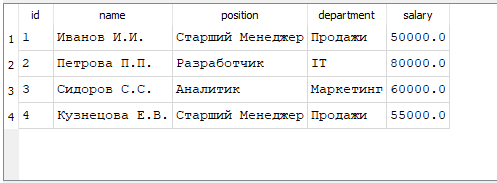
UPDATE employees

SET position = 'Старший ' || position

WHERE position = 'Менеджер';

******

***Результат:***

******

***Д)***

***Добавим в раннее созданную таблицу «orders» данные:***

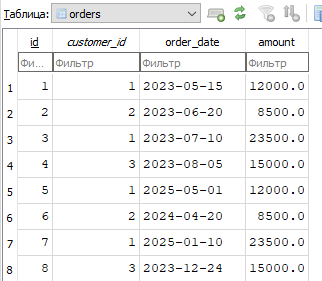
INSERT INTO orders (customer\_id, order\_date, amount) VALUES

(1, '2025-05-01', 12000),

(2, '2024-04-20', 8500),

(1, '2025-01-10', 23500),

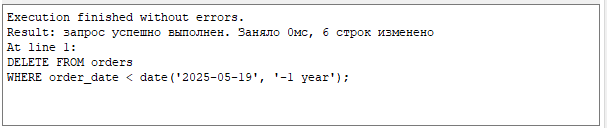
(3, '2023-12-24', 15000);

******

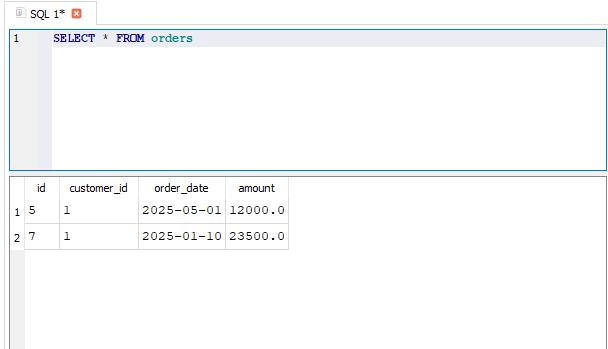
**Удалим заказы, сделанные более 1 года назад:**

DELETE FROM orders

WHERE order\_date < date('2025-05-19', '-1 year');



***Результат:***

******

***Е)***

**Используем раннее созданные таблицы «customers» и «orders»**

**Заполним их данными:**

INSERT INTO orders (customer\_id, order\_date, amount) VALUES

(1, '2023-05-15', 12000),

(2, '2023-06-20', 8500),

(1, '2023-07-10', 23500),

(3, '2023-08-05', 15000);

**Запрос на подсчет заказов для каждого клиента:**

SELECT

c.id,

c.name AS customer\_name,

COUNT(o.id) AS orders\_count,

SUM(o.amount) AS total\_amount

FROM

customers c

LEFT JOIN

orders o ON c.id = o.customer\_id

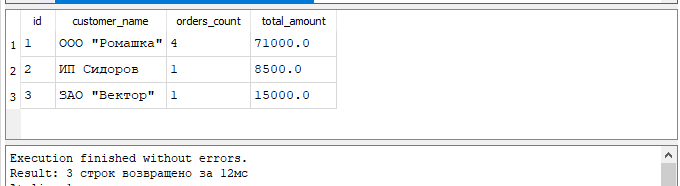
GROUP BY

c.id, c.name

ORDER BY

orders\_count DESC;

**Результат:**

******

***Ж)***

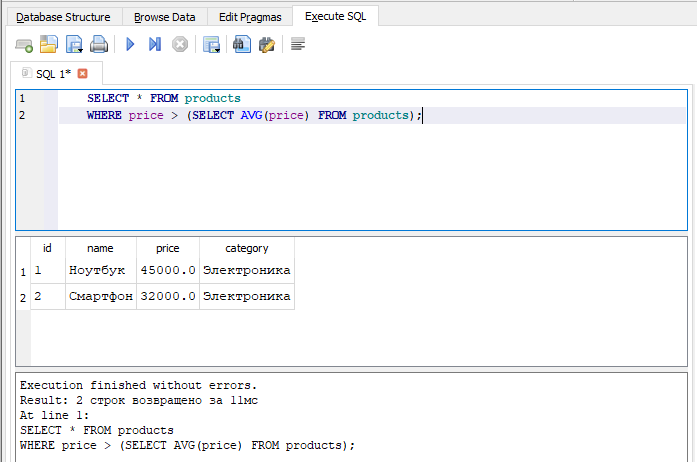
***Используем раннее созданную таблицу «products»***

***Запрос на выборку товаров с ценой выше среднего:***

SELECT \* FROM products

WHERE price > (SELECT AVG(price) FROM products);

**Результат:**



***З)***

***Запрос на создание таблицы «books»***

CREATE TABLE books (

id INTEGER PRIMARY KEY AUTOINCREMENT,

title TEXT NOT NULL,

author TEXT NOT NULL,

year INTEGER,

price REAL,

genre TEXT

)

**Заполнение таблицы данными:**

INSERT INTO books (title, author, year, price, genre) VALUES

('Мастер и Маргарита', 'Булгаков М.А.', 1967, 450, 'Роман'),

('Преступление и наказание', 'Достоевский Ф.М.', 1866, 380, 'Роман'),

('Война и мир', 'Толстой Л.Н.', 1869, 600, 'Роман'),

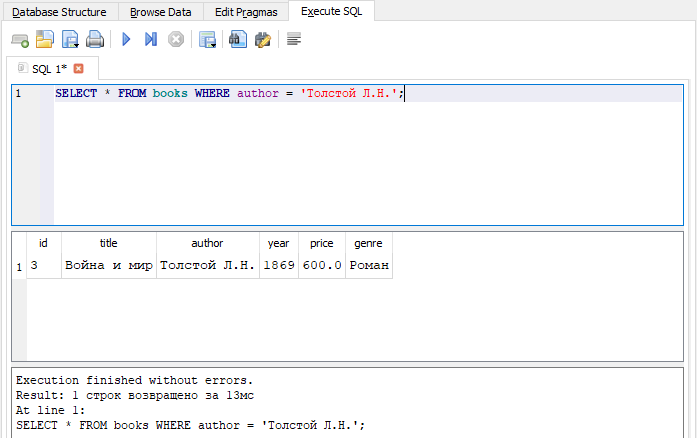
('Мёртвые души', 'Гоголь Н.В.', 1842, 400, 'Поэма'),

('Герой нашего времени', 'Лермонтов М.Ю.', 1840, 350, 'Роман');

**Запрос на нахождения всех книг принадлежащим конкретному автору (Л.Н. Толстой):**

SELECT \* FROM books WHERE author = 'Толстой Л.Н.';

**Результат:**



***И)***

**Запрос на создание новой таблицы «customer\_orders\_summary» с объединенными данными из таблиц «customers» и «orders»:**

CREATE TABLE customer\_orders\_summary AS

SELECT

c.id AS customer\_id,

c.name AS customer\_name,

c.email,

COUNT(o.id) AS orders\_count,

SUM(o.amount) AS total\_amount,

MAX(o.order\_date) AS last\_order\_date

FROM

customers c

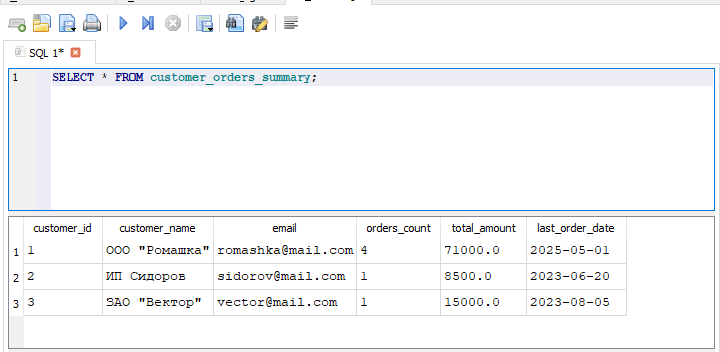
LEFT JOIN

orders o ON c.id = o.customer\_id

GROUP BY

c.id, c.name, c.email;

**Результат:**

****

**Выводы**

В ходе выполнения лабораторной работы: