Лабораторная работа №12

База данных:

CREATE TABLE shop (

id INTEGER PRIMARY KEY,

name VARCHAR(255),

balance FLOAT NOT NULL

);

CREATE TABLE product (

id INTEGER PRIMARY KEY,

name VARCHAR(255),

price FLOAT NOT NULL

);

CREATE TABLE warehouse (

shop\_id INTEGER,

product\_id INTEGER,

quantity INTEGER NOT NULL,

PRIMARY KEY (shop\_id, product\_id),

FOREIGN KEY (shop\_id) REFERENCES shop(id) ON DELETE CASCADE,

FOREIGN KEY (product\_id) REFERENCES product(id) ON DELETE CASCADE

);

Тестовые данные:

INSERT INTO shop (id, name, balance) VALUES (1, 'Магазин A', 5000.00);

INSERT INTO shop (id, name, balance) VALUES (2, 'Магазин B', 7000.00);

INSERT INTO product (id, name, price) VALUES (1, 'Товар X', 100.50);

INSERT INTO product (id, name, price) VALUES (2, 'Товар Y', 200.75);

INSERT INTO warehouse (shop\_id, product\_id, quantity) VALUES (1, 1, 50);

INSERT INTO warehouse (shop\_id, product\_id, quantity) VALUES (1, 2, 30);

INSERT INTO warehouse (shop\_id, product\_id, quantity) VALUES (2, 1, 20);

INSERT INTO warehouse (shop\_id, product\_id, quantity) VALUES (2, 2, 40);

Проверка данных:

SELECT s.name AS Магазин, p.name AS Товар, w.quantity AS Количество

FROM warehouse w

JOIN shop s ON w.shop\_id = s.id

JOIN product p ON w.product\_id = p.id;

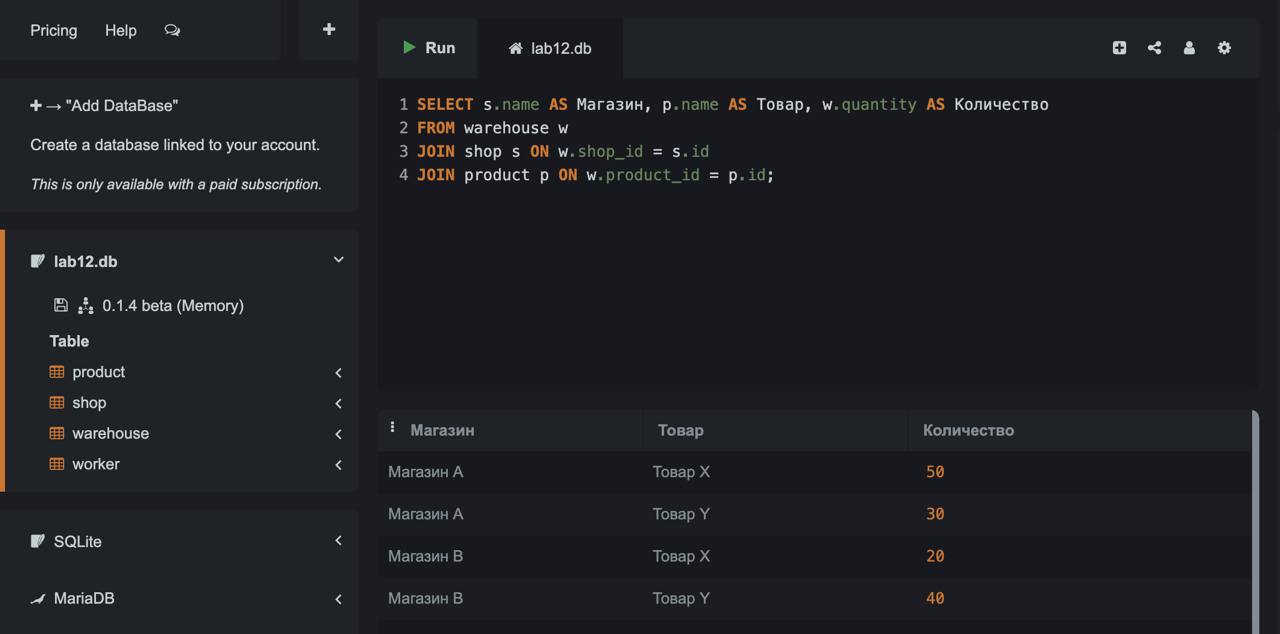


Рисунок таблицы "worker"

CREATE TABLE worker (

worker\_id INTEGER PRIMARY KEY,

shop\_id INTEGER REFERENCES product (id),

name VARCHAR(255),

salary INTEGER NOT NULL,

position VARCHAR(255)

);

INSERT INTO worker (worker\_id, shop\_id, name, salary, position)

VALUES

(1, 1, 'Иван Иванов', 50000, 'Менеджер'),

(2, 2, 'Петр Петров', 45000, 'Продавец'),

(3, 1, 'Анна Смирнова', 60000, 'Администратор');

A screenshot of a computer

Description automatically generated

**Сортировка данных (ORDER BY)**

Запрос для сортировки сотрудников по зарплате (по убыванию):

SELECT \* FROM worker ORDER BY salary DESC;



Запрос для сортировки сотрудников сначала по должности, затем по имени (по алфавиту):

SELECT \* FROM worker ORDER BY position ASC, name ASC;

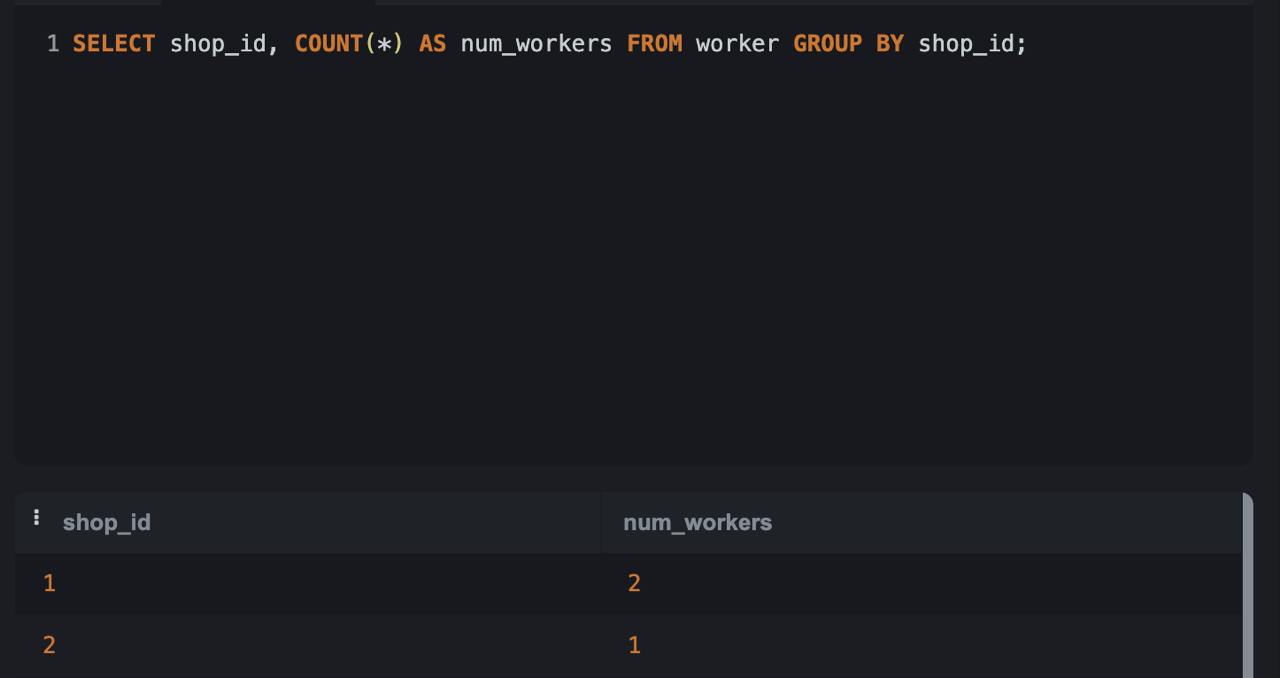
A screenshot of a computer

Description automatically generated

**Группировка данных (GROUP BY)**

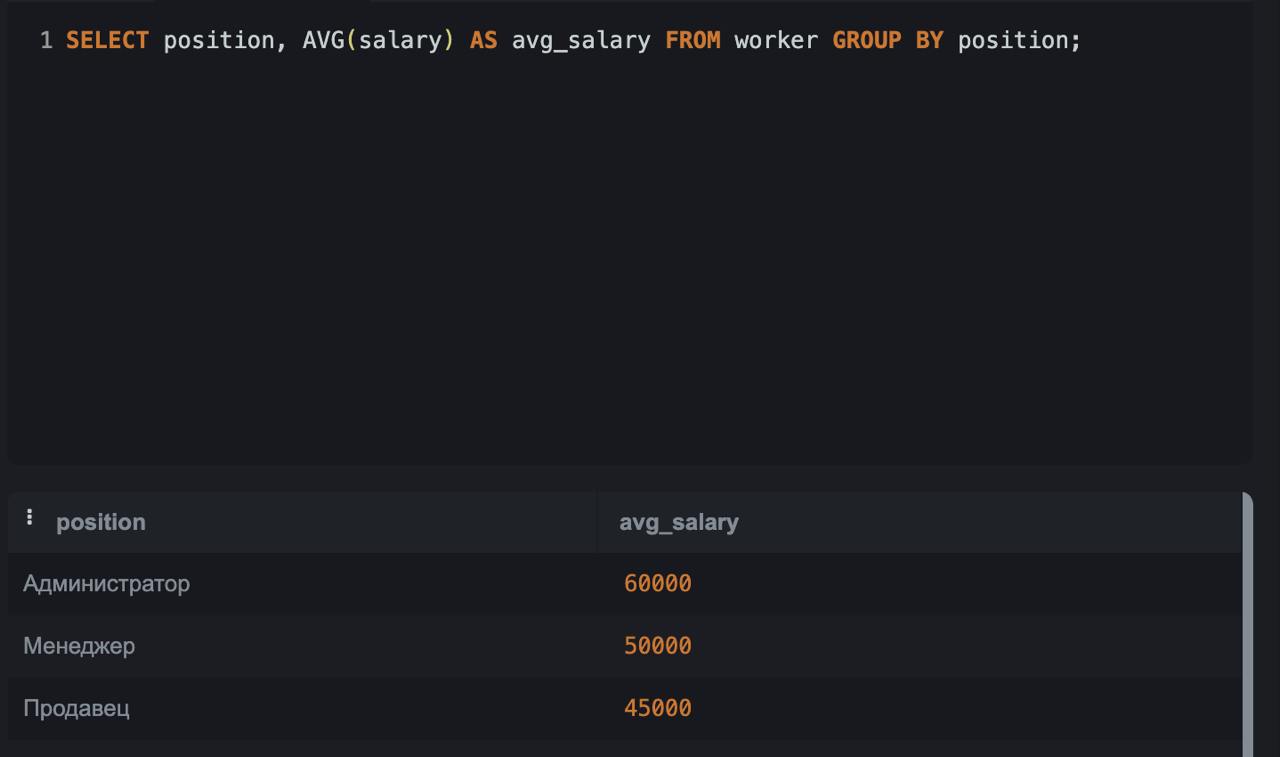
Запрос для подсчета количества сотрудников в каждом магазине:

SELECT shop\_id, COUNT(\*) AS num\_workers FROM worker GROUP BY shop\_id;



Запрос для определения средней зарплаты по каждой должности:

SELECT position, AVG(salary) AS avg\_salary FROM worker GROUP BY position;



**Использование агрегатных функций**

Запрос для нахождения максимальной и минимальной зарплаты среди сотрудников:

SELECT MAX(salary) AS max\_salary, MIN(salary) AS min\_salary FROM worker;

A screen shot of a computer

Description automatically generated

Запрос для вычисления общей суммы зарплат всех сотрудников:

SELECT SUM(salary) AS total\_salary FROM worker;

