

# **Лабораторная работа №2**

"Работа с данными. Простые запросы на выборку"

Выполнил студент группы Б20-505

**Сорочан Илья**

Московский Инженерно-Физический Институт

Москва 2023

# **1 Введение**

В данной лабораторной работе будут разработаны различные запросы. Запросы будут разобраны в самой лабораторной. Скрипт по созданию тестовой ДБ можно найти в приложении или в репозитории.

## 2 Выполненные SQL запросы

### 2.1 Полная выборка

В качестве примера выведем список доступных категорий товаров:

```
sqlite> select name from category;
```

Ручки

Тетради

Карандаши

### 2.2 Условная выборка

Должности, на которых зарплата больше 2000:

```
sqlite> select name from job where job.salary > 2000;
```

Менеджер

Кассир

### 2.3 Упорядоченная выборка

Выведем должности, но теперь в порядке увеличения зарплат:

```
sqlite> select name from job order by job.salary;
```

Уборщик

Кассир

Менеджер

Наоборот:

```
sqlite> select name from job order by job.salary desc;
```

Менеджер

Кассир

Уборщик

## 2.4 Простые функции и операторы

Посмотрим, на каких должностях больше всего подневная зарплата:

```
select name, job.salary / job.days_per_week from job order by  
job.salary desc;
```

## 2.5 Агрегатные функции

Найдем самые дорогие товары из каждой страны:

```
sqlite> select country,max(price) from product group by country;  
1|999  
2|1299  
3|1499
```

## 2.6 Выбор уникальных записей

Представим ситуацию, что мы хотим посмотреть какие работники отвечали за поставки за все время:

```
sqlite> select distinct employee from supply;  
1  
2  
3
```

Для того, что бы вместо цифр были имена можно склеить две таблицы:

```
sqlite> select distinct name from employee join supply on employee.id  
= supply.employee;  
Джон Стоун
```

Джек Воробей  
Партос

## **2.7 Псевдо-выборка**

```
sqlite> select 2*2;
```

## **2.8 VACUUM**

```
sqlite> VACUUM;
```

### **3 Заключение**

В ходе данной лабораторной работы было выполнено 8 SQL запросов. Все запросы проиллюстрированы и разобраны в самой работе. Код для создания тестовой базы данных можно найти в приложении или на репозитории.

# Приложение А

Код для создания тестовой базы данных:

```
-- CREATE SECTION
```

```
CREATE TABLE vendor (  
    id INTEGER PRIMARY KEY,  
    name TEXT,  
    office TEXT,  
    rating INTEGER  
);
```

```
CREATE TABLE category (  
    id INTEGER PRIMARY KEY,  
    name TEXT  
);
```

```
CREATE TABLE product_type (  
    code INTEGER PRIMARY KEY,  
    vendor INTEGER,  
    category INTEGER,  
    raise INTEGER,  
    FOREIGN KEY(vendor) REFERENCES vendor(id),  
    FOREIGN KEY(category) REFERENCES category(id)  
);
```

```
CREATE TABLE country (  
    id INTEGER PRIMARY KEY,  
    name TEXT  
);
```

```
CREATE TABLE product (  
    id INTEGER PRIMARY KEY,
```

```

    type INTEGER,
    country INTEGER,
    price INTEGER,
    supply INTEGER,
    sold BOOLEAN,
    sale INTEGER,
    FOREIGN KEY(type) REFERENCES product_type(code),
    FOREIGN KEY(country) REFERENCES country(id),
    FOREIGN KEY(supply) REFERENCES supply(id),
    FOREIGN KEY(sale) REFERENCES sale(id)
);

CREATE TABLE supply (
    id INTEGER PRIMARY KEY,
    date DATE,
    done BOOLEAN,
    employee INTEGER,
    price INTEGER,
    FOREIGN KEY(employee) REFERENCES employee(id)
);

CREATE TABLE employee (
    id INTEGER PRIMARY KEY,
    job INTEGER,
    name TEXT,
    active BOOLEAN,
    exp_date TIMESTAMP,
    FOREIGN KEY(job) REFERENCES job(id)
);

CREATE TABLE job (
    id INTEGER PRIMARY KEY,
    name TEXT,
    salary INTEGER,
    days_per_week INTEGER

```



```
);
```

```
CREATE TABLE sale (  
    id INTEGER PRIMARY KEY,  
    date TIMESTAMP,  
    employee INTEGER,  
    customer INTEGER,  
    bonus INTEGER,  
    bonus_used INTEGER,  
    FOREIGN KEY(employee) REFERENCES employee(id),  
    FOREIGN KEY(customer) REFERENCES customer(id)  
);
```

```
CREATE TABLE customer (  
    id INTEGER PRIMARY KEY,  
    phone INTEGER,  
    name TEXT,  
    reg_date TIMESTAMP  
);
```

```
-- INSERT SECTION
```

```
INSERT INTO vendor (id, name, office, rating) VALUES (1, 'Hatber',  
'Россия, г. Москва, ул. X1, дом Y1', 8);  
INSERT INTO vendor (id, name, office, rating) VALUES (2, 'Самсон',  
'Россия, г. Воронеж, ул. X2, дом Y2', 7);  
INSERT INTO vendor (id, name, office, rating) VALUES (3, 'Апплика',  
'Россия, г. Москва, ул. X3, дом Y3', 6);
```

```
INSERT INTO category (id, name) VALUES (1, 'Ручки');  
INSERT INTO category (id, name) VALUES (2, 'Тетради');  
INSERT INTO category (id, name) VALUES (3, 'Карандаши');
```

```
INSERT INTO product_type (code, vendor, category, raise) VALUES  
(1001, 1, 1, 15);
```

```
INSERT INTO product_type (code, vendor, category, raise) VALUES  
(1002, 2, 1, 10);
```

```
INSERT INTO product_type (code, vendor, category, raise) VALUES  
(1003, 1, 2, 20);
```

```
INSERT INTO country (id, name) VALUES (1, 'Германия');
```

```
INSERT INTO country (id, name) VALUES (2, 'Россия');
```

```
INSERT INTO country (id, name) VALUES (3, 'Китай');
```

```
INSERT INTO product (id, type, country, price, supply, sold,  
sale) VALUES (1, 1001, 1, 999, 1, false, null);
```

```
INSERT INTO product (id, type, country, price, supply, sold,  
sale) VALUES (2, 1002, 2, 899, 2, true, 1);
```

```
INSERT INTO product (id, type, country, price, supply, sold,  
sale) VALUES (3, 1003, 3, 1499, 3, false, null);
```

```
INSERT INTO product (id, type, country, price, supply, sold,  
sale) VALUES (4, 1004, 2, 1299, 2, true, 2);
```

```
INSERT INTO product (id, type, country, price, supply, sold,  
sale) VALUES (5, 1005, 3, 129, 3, true, 3);
```

```
INSERT INTO supply (id, date, done, employee, price) VALUES (1,  
'2023-06-01', true, 1, 850);
```

```
INSERT INTO supply (id, date, done, employee, price) VALUES (2,  
'2023-06-15', true, 2, 750);
```

```
INSERT INTO supply (id, date, done, employee, price) VALUES (3,  
'2023-06-30', false, 3, 1400);
```

```
INSERT INTO employee (id, job, name, active, exp_date) VALUES  
(1, 1, 'Джон Стоун', true, '2023-12-31');
```

```
INSERT INTO employee (id, job, name, active, exp_date) VALUES  
(2, 2, 'Джек Воробей', true, '2024-06-30');
```

```
INSERT INTO employee (id, job, name, active, exp_date) VALUES  
(3, 3, 'Партос', false, '2023-09-30');
```

```
INSERT INTO job (id, name, salary, days_per_week) VALUES (1,  
'Менеджер', 5000, 4);  
INSERT INTO job (id, name, salary, days_per_week) VALUES (2,  
'Кассир', 2500, 5);  
INSERT INTO job (id, name, salary, days_per_week) VALUES (3,  
'Уборщик', 500, 7);
```

```
INSERT INTO customer (id, phone, name, reg_date) VALUES (1, 1111111111,  
'Вася Пупкин', '2022-01-15');  
INSERT INTO customer (id, phone, name, reg_date) VALUES (2, 1111211111,  
'Джек Ричер', '2022-02-28');  
INSERT INTO customer (id, phone, name, reg_date) VALUES (3, 1111131111,  
'Том Сойер', '2022-06-01');
```

```
INSERT INTO sale (id, date, employee, customer, bonus, bonus_used)  
VALUES (1, '2023-06-20 10:30:00', 1, 1, 50, 0);  
INSERT INTO sale (id, date, employee, customer, bonus, bonus_used)  
VALUES (2, '2023-06-22 14:15:00', 2, 2, 25, 0);  
INSERT INTO sale (id, date, employee, customer, bonus, bonus_used)  
VALUES (3, '2023-06-23 11:00:00', 3, 3, 75, 75);
```