Лабораторная работа №3

"Сложные запросы на выборку. Соединения"

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 Подзапросы

Должности, на которых зарплата выше среднего:

```
sqlite> select job.name from job where job.salary > (select avg(job.salar
from job);
Менеджер
```

Товары, цена которых выше среднего:

```
sqlite> select product.id from product where product.price >
(select avg(product.price) from product);
1
3
4
```

2.2 Соединение

Должности, на которых зарплата больше 2000:

```
sqlite> select employee.name, job.name from employee, job where
employee.job = job.id;
Джон Стоун|Менеджер
Джек Воробей|Кассир
Партос|Уборщик
```

Покупатель для последних 10 покупок:

```
sqlite> select sale.id, customer.name from sale, customer where
sale.customer = customer.id order by sale.date limit 10;
1|Вася Пупкин
2|Джек Ричер
3|Том Сойер
```

2.3 Теоретико-множественные операции

Посмотрим, на каких должностях можно заработать много либо работать мало:

```
sqlite> select name from job where salary > 2000 union select
name from job where days_per_week < 5;
Кассир
Менеджер
```

Иногда продавцы могут покупать товары на свое имя. Посмотрим есть ли такие:

```
sqlite> select name from employee intersect select name from
customer;
```

2.4 Обычные и рекурсивные табличные выражения

В прошлой лабораторной мы выводили имена сотрудников, отвечавших за доставку хоть раз. В этот раз выведем не только имена, но и всю информацию о них:

```
sqlite> with supply_emp as (select distinct employee from supply)
select * from employee where id in supply_emp;
1|1|Джон Стоун|1|2023-12-31
2|2|Джек Воробей|1|2024-06-30
3|3|Партос|0|2023-09-30
```

2.5 Аналитические функции

Посмотрим как меняются цены за доставку:

```
sqlite> select id, (price - lag(price, 1, null) over (order by
date)) as change from supply;
```

```
1|
2|-100
3|650
```

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, 11111311111,
'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);
```