

**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ СВЯЗИ И МАССОВЫХ
КОММУНИКАЦИЙ**
Ордена Трудового Красного Знамени
**Федеральное государственное бюджетное образовательное учреждение высшего
образования**
«Московский технический университет связи и информатики»
Кафедра «Математическая кибернетика и информационные технологии»

Отчет по практической работе №12
по дисциплине «Введение в информационные технологии»
на тему: «Основы выборки SQL»

Выполнила: студентка группы

БПИ2401

Алексеева Татьяна Игоревна

Проверил:

Мкртчян Грач Маратович

Москва

2025

1. Цель работы:

Познакомиться с основами выборки в SQL, научиться работать с операторами ORDER BY, GROUP BY и агрегатными функциями.

2. Ход работы:

Создадим таблицы shop, product, warehouse и worker:

```
1 CREATE TABLE shop (  
2 id INTEGER PRIMARY KEY,  
3 name VARCHAR(255) UNIQUE,  
4 balance FLOAT NOT NULL);  
5  
6 CREATE TABLE product (  
7 id INTEGER PRIMARY KEY,  
8 name VARCHAR(255) UNIQUE,  
9 price FLOAT NOT NULL);  
10  
11 CREATE TABLE warehouse (  
12 shop_id INTEGER REFERENCES shop(id),  
13 product_id INTEGER REFERENCES product(id),  
14 quantity INTEGER NOT NULL,  
15 PRIMARY KEY (shop_id, product_id));  
16  
17 CREATE TABLE worker (  
18 worker_id INTEGER PRIMARY KEY,  
19 shop_id INTEGER REFERENCES shop(id),  
20 name VARCHAR(255),  
21 salary INTEGER NOT NULL,  
22 position VARCHAR(255));
```

Заполним таблицы данными:

```
1 INSERT INTO shop (id, name, balance) VALUES (1, 'Магнит', 3342.0);  
2 INSERT INTO shop (id, name, balance) VALUES (2, 'Перекрёсток', 4876.56);  
3 INSERT INTO shop (id, name, balance) VALUES (3, 'Супер Лента', 6900.77);  
4 INSERT INTO shop (id, name, balance) VALUES (4, 'Ашан', 8089.23);  
5  
6 INSERT INTO product VALUES (1, 'Шоколад', 79.99);  
7 INSERT INTO product VALUES (2, 'Печенье', 45.99);  
8 INSERT INTO product VALUES (3, 'Молоко', 89.78);  
9 INSERT INTO product VALUES (4, 'Яйца 10 шт', 129.56);  
10 INSERT INTO product VALUES (5, 'Мясо птицы', 349.90);  
11  
12 INSERT INTO warehouse VALUES (1, 1, 100);  
13 INSERT INTO warehouse VALUES (1, 2, 50);  
14 INSERT INTO warehouse VALUES (1, 3, 30);  
15 INSERT INTO warehouse VALUES (1, 4, 56);  
16 INSERT INTO warehouse VALUES (1, 5, 90);  
17  
18 INSERT INTO warehouse VALUES (2, 1, 99);  
19 INSERT INTO warehouse VALUES (2, 2, 65);  
20 INSERT INTO warehouse VALUES (2, 3, 48);  
21 INSERT INTO warehouse VALUES (2, 4, 34);  
22 INSERT INTO warehouse VALUES (2, 5, 67);  
23  
24 INSERT INTO warehouse VALUES (3, 1, 0);  
25 INSERT INTO warehouse VALUES (3, 2, 12);  
26 INSERT INTO warehouse VALUES (3, 3, 43);  
27 INSERT INTO warehouse VALUES (3, 4, 0);  
28 INSERT INTO warehouse VALUES (3, 5, 33);  
29  
30 INSERT INTO warehouse VALUES (4, 1, 1);  
31 INSERT INTO warehouse VALUES (4, 2, 2);  
32 INSERT INTO warehouse VALUES (4, 3, 3);  
33 INSERT INTO warehouse VALUES (4, 4, 4);  
34 INSERT INTO warehouse VALUES (4, 5, 3);  
  
1 INSERT INTO worker VALUES (1, 1, 'Пётр', 5000, 'Кассир');  
2 INSERT INTO worker VALUES (2, 1, 'Марина', 30000, 'Фасовщик');  
3 INSERT INTO worker VALUES (3, 1, 'Пингвин Маруся', 79000, 'Дегустатор');  
4  
5 INSERT INTO worker VALUES (4, 2, 'Александр', 20020, 'Менеджер');  
6 INSERT INTO worker VALUES (5, 2, 'София', 40000, 'Кассир');  
7 INSERT INTO worker VALUES (6, 2, 'Софья', 40000, 'Кассир');  
8  
9 INSERT INTO worker VALUES (7, 3, 'Гусь Валера', 89900, 'Уборщик');  
10 INSERT INTO worker VALUES (8, 3, 'Жожа', 1000000, 'Доставщик');  
11 INSERT INTO worker VALUES (9, 3, 'Ева', 795461, 'Бухгалтер');  
12  
13 INSERT INTO worker VALUES (10, 4, 'Дарина', 543000, 'Менеджер');  
14 INSERT INTO worker VALUES (11, 4, 'Голубь Паша', 320000, 'Фасовщик');  
15 INSERT INTO worker VALUES (12, 4, 'Мария', 88000, 'Бухгалтер');
```

Выполним различные запросы с использованием операторов ORDER BY, GROUP BY и агрегатных функций.

1) Список магазинов, отсортированный по убыванию баланса:

```
1 SELECT name, balance FROM shop
2 ORDER BY balance DESC;
```

name	balance
Ашан	8089.23
Супер Лента	6900.77
Перекрёсток	4876.56
Магнит	3342

2) Товары дороже 100 рублей, отсортированные по цене:

```
1 SELECT name, price FROM product
2 WHERE price > 100
3 ORDER BY price;
```

name	price
Яйца 10 шт	129.56
Мясо птицы	349.9

3) Работники магазина «Магнит» по убыванию зарплаты:

```
1 SELECT w.name, w.salary, w.position FROM worker w
2 JOIN shop s ON w.shop_id = s.id
3 WHERE s.name = 'Магнит'
4 ORDER BY w.salary DESC;
```

name	salary	position
Пингвин Маруся	79000	Дегустатор
Марина	30000	Фасовщик
Пётр	5000	Кассир

4) Товары на складе магазина «Перекрёсток» с количеством > 50:

```
1 SELECT p.name, w.quantity FROM warehouse w
2 JOIN product p ON w.product_id = p.id
3 JOIN shop s ON w.shop_id = s.id
4 WHERE s.name = 'Перекрёсток' AND w.quantity > 50
5 ORDER BY w.quantity DESC;
```

name	quantity
Шоколад	99
Мясо птицы	67
Печенье	65

5) Общая сумма зарплат по должностям:

```
1 SELECT position, SUM(salary) AS total_salary FROM worker
2 GROUP BY position
3 ORDER BY total_salary DESC;
```

position	total_salary
Доставщик	1000000
Бухгалтер	883461
Менеджер	563020
Фасовщик	350000
Уборщик	89900
Кассир	85000
Дегустатор	79000

6) Количество работников и их минимальная зарплата по магазинам:

```
1 SELECT s.name, COUNT(*) AS workers, MIN(w.salary) AS min_salary FROM worker w
2 JOIN shop s ON w.shop_id = s.id
3 GROUP BY s.name
```

name	workers	min_salary
Ашан	3	88000
Магнит	3	5000
Перекрёсток	3	20020
Супер Лента	3	89900

7) Общая стоимость всех товаров на складе для каждого магазина:

```
1 SELECT s.name, SUM(p.price * w.quantity) AS total_inventory_value FROM warehouse w
2 JOIN product p ON w.product_id = p.id
3 JOIN shop s ON w.shop_id = s.id
4 GROUP BY s.name
5 ORDER BY total_inventory_value DESC;
```

name	total_inventory_value
Магнит	51738.259999999995
Перекрёсток	43066.14
Супер Лента	15959.119999999999
Ашан	2009.2499999999998

8) Товары, которых нет в наличии хотя бы в одном магазине:

```
1 SELECT p.name, s.name AS shop_name FROM warehouse w
2 JOIN shop s ON w.shop_id = s.id
3 JOIN product p ON w.product_id = p.id
4 WHERE w.quantity = 0;
```

name	shop_name
Шоколад	Супер Лента
Яйца 10 шт	Супер Лента

9) Должности, где есть работники с зарплатой выше средней:

```
1 SELECT position, name, salary FROM worker
2 WHERE salary > (SELECT AVG(salary) FROM worker)
3 ORDER BY salary DESC;
```

!	position	name	salary
	Доставщик	Жожа	1000000
	Бухгалтер	Ева	795461
	Менеджер	Дарина	543000
	Фасовщик	Голубь Паша	320000

Вывод: в ходе работы я научилась работать с оператором сортировки ORDER BY, оператором группировки GROUP BY и агрегатными функциями (SUM, AVG, COUNT, MIN, MAX).