



МИНОБРНАУКИ РОССИИ

**Федеральное государственное бюджетное образовательное учреждение высшего
образования**

«МИРЭА – Российский технологический университет» РТУ

МИРЭА

Колледж программирования и кибербезопасности

Практическая работа №5 учебной дисциплины:

МДК.11.01 Технология разработки и защиты баз данных

Тема: Формирование и выполнение подзапросов

Выполнил студент группы ПКС-31

Лопатин Л.В.

преподаватель

Понеделко Е.В.

Москва

2025

Цель работы: Формирование навыков работы в среде СУБД PostgreSQL по созданию вложенных запросов.

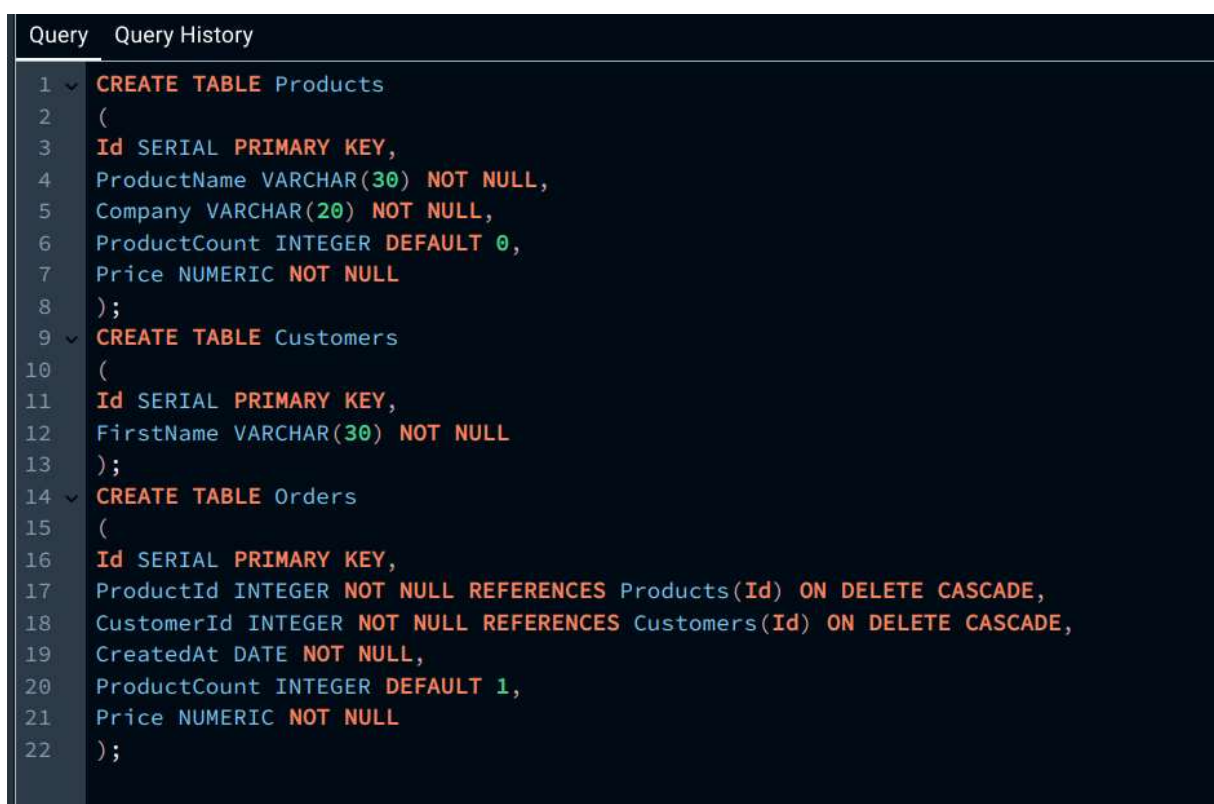
Ход работы:

Часть 1

Усвоил предоставленный материал.

Часть 2

1. Создал БД «sales», создал необходимые таблицы в соответствии с предметной областью (см. рисунок 1).



```
Query  Query History
1  CREATE TABLE Products
2  (
3  Id SERIAL PRIMARY KEY,
4  ProductName VARCHAR(30) NOT NULL,
5  Company VARCHAR(20) NOT NULL,
6  ProductCount INTEGER DEFAULT 0,
7  Price NUMERIC NOT NULL
8  );
9  CREATE TABLE Customers
10 (
11 Id SERIAL PRIMARY KEY,
12 FirstName VARCHAR(30) NOT NULL
13 );
14 CREATE TABLE Orders
15 (
16 Id SERIAL PRIMARY KEY,
17 ProductId INTEGER NOT NULL REFERENCES Products(Id) ON DELETE CASCADE,
18 CustomerId INTEGER NOT NULL REFERENCES Customers(Id) ON DELETE CASCADE,
19 CreatedAt DATE NOT NULL,
20 ProductCount INTEGER DEFAULT 1,
21 Price NUMERIC NOT NULL
22 );
```

Рис. 1 Создание таблиц

2. Добавил в таблицы наборы данных (см. рисунок 2).

```
Query  Query History
1  INSERT INTO Products(ProductName, Company, ProductCount, Price)
2  VALUES ('iPhone X', 'Apple', 2, 66000),
3  ('iPhone 8', 'Apple', 2, 51000),
4  ('iPhone 7', 'Apple', 5, 42000),
5  ('Galaxy S9', 'Samsung', 2, 56000),
6  ('Galaxy S8 Plus', 'Samsung', 1, 46000),
7  ('Nokia 9', 'HDM Global', 2, 26000),
8  ('Desire 12', 'HTC', 6, 38000);
9  INSERT INTO Customers(FirstName)
10 VALUES ('Tom'), ('Bob'), ('Sam');
11 INSERT INTO Orders(ProductId, CustomerId, CreatedAt, ProductCount, Price)
12 VALUES
13 (
14 (SELECT Id FROM Products WHERE ProductName='Galaxy S9'),
15 (SELECT Id FROM Customers WHERE FirstName='Tom'),
16 '2017-07-11',
17 2,
18 (SELECT Price FROM Products WHERE ProductName='Galaxy S9')
19 ),
20 (
21 (SELECT Id FROM Products WHERE ProductName='iPhone 8'),
22 (SELECT Id FROM Customers WHERE FirstName='Tom'),
23 '2017-07-13',
24 1,
25 (SELECT Price FROM Products WHERE ProductName='iPhone 8')
26 ),
27 (
28 (SELECT Id FROM Products WHERE ProductName='iPhone 8'),
29 (SELECT Id FROM Customers WHERE FirstName='Bob'),
30 '2017-07-11',
31 1,
```

Data Output Messages Notifications

Рис. 2 Добавление данных в таблицы

3. Нашел товары из таблицы Products, которые имеют минимальную цену (см. рисунок 3).

Query

Query History

1

SELECT *

2

FROM Products

3

WHERE Price = (SELECT MIN(Price) FROM Products);

Data Output

Messages

Notifications

≡+

▼

▼

SQL

	id [PK] integer	productname character varying (30)	company character varying (20)	productcount integer	price numeric
1	6	Nokia 9	HDM Global	2	26000

Рис. 3 Товар с минимальной ценой

4. Нашел товары с ценой выше средней (см. рисунок 4).

</

Рис. 4 Товары с ценой выше средней

5. Нашел все заказы из таблицы Orders, добавив к ним информацию о товаре (см. рисунок 5).

The screenshot shows a PostgreSQL query editor interface. At the top, the connection is set to 'sales/postgres@main'. Below the connection bar is a toolbar with various icons for file operations, query execution, and settings. The main area displays a SQL query:

```
1 SELECT CreatedAt,  
2 Price,  
3 (SELECT ProductName FROM Products  
4 WHERE Products.Id = Orders.ProductId) AS Product  
5 FROM Orders;
```

Below the query editor, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table with the results of the query. The table has four columns: 'createdat' (date), 'price' (numeric), and 'product' (character varying (30)). The results are as follows:

	createdat date	price numeric	product character varying (30)
1	2017-07-11	56000	Galaxy S9
2	2017-07-13	51000	iPhone 8
3	2017-07-11	51000	iPhone 8

Рис. 5 Товары с дополнительной информацией

6. Выбрал из таблицы Products те товары, стоимость которых выше средней цены товаров для данного производителя (см. рисунок 6).

The screenshot shows a SQL query editor with the following query:

```

1 SELECT ProductName,
2    Company,
3    Price,
4    (SELECT AVG(Price) FROM Products AS SubProds
5     WHERE SubProds.Company=Prods.Company) AS AvgPrice
6 FROM Products AS Prods
7 WHERE Price >
8    (SELECT AVG(Price) FROM Products AS SubProds
9     WHERE SubProds.Company=Prods.Company);

```

Below the query editor, the 'Data Output' tab is active, showing a table with the following data:

	productname character varying (30)	company character varying (20)	price numeric	avgprice numeric
1	iPhone X	Apple	66000	53000.000000000000
2	Galaxy S9	Samsung	56000	51000.000000000000

Рис. 6 Товары, стоимость которых выше средней цены производителя

7. Проанализировал запросы и сделал выводы.

Часть 3

1. Создал запрос в соответствии с потребностью пользователя: Получить список студентов, получивших оценку только 5 (см. рисунок 7)

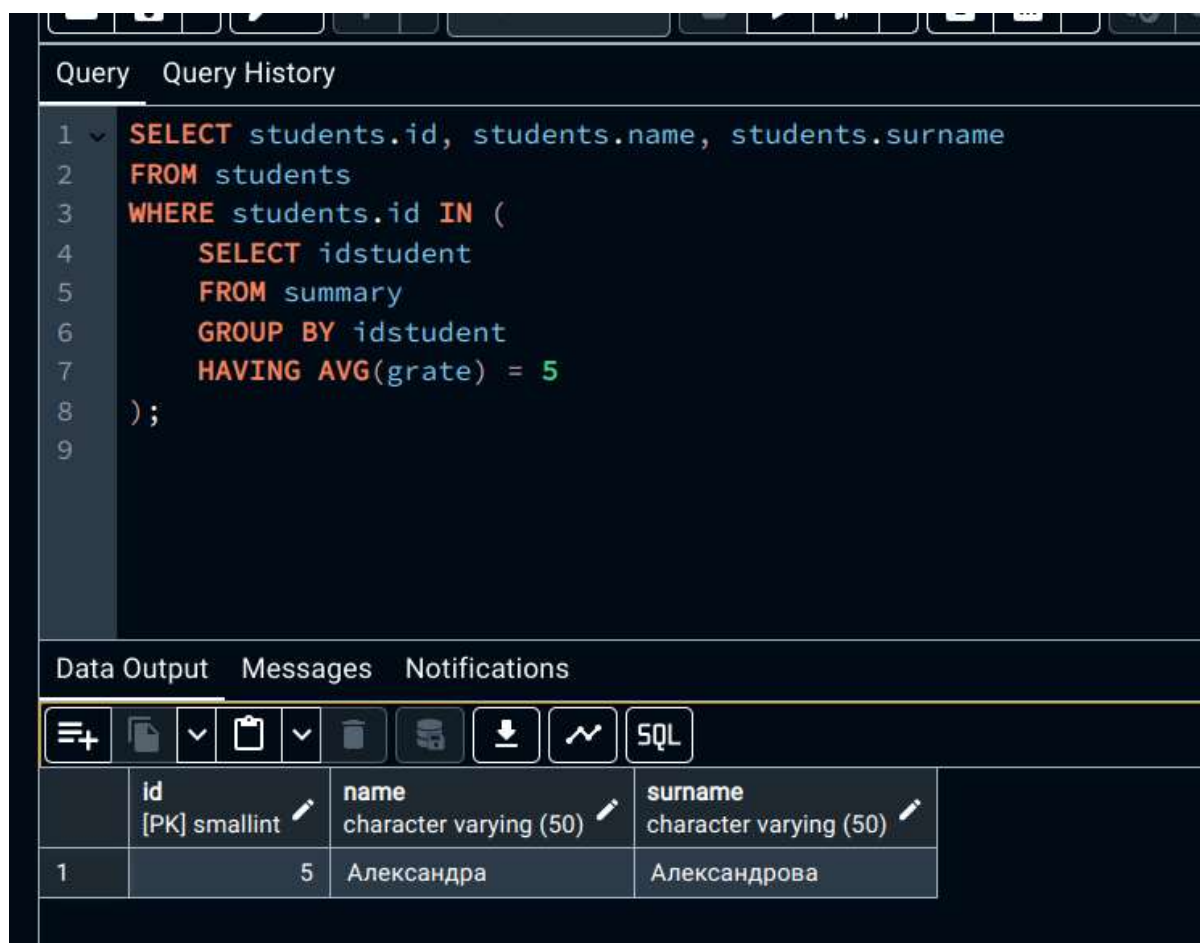


Рис. 7 Запрос 1

2. Создал запрос в соответствии с потребностью пользователя: Получить список преподавателей с формой отчётности: экзамен (см. рисунок 8)

The screenshot shows a database query editor with a dark theme. The top toolbar includes icons for file operations, query execution, and settings. The 'Query' tab is active, displaying the following SQL code:

```

1 SELECT teachers.id, teachers.name, teachers.surname
2 FROM teachers
3 WHERE teachers.id IN (
4     SELECT idteacher FROM public.plan WHERE examform = 'Экзамен'
5 );

```

Below the query editor, the 'Data Output' tab is active, showing a table with 3 rows and 4 columns. The columns are 'id' (smallint, PK), 'name' (character varying (50)), 'surname' (character varying (50)), and an unlabeled column. The data is as follows:

	id [PK] smallint	name character varying (50)	surname character varying (50)
1	1	Виктор	Семенов
2	2	Виктория	Симонова
3	3	Анна	Ромашкина

Рис. 8 Запрос 2

3. Создал запрос в соответствии с потребностью пользователя: Получить список студентов, получивших оценку 5 (см. рисунок 9)

The screenshot shows a database query editor with a dark theme. The top toolbar includes icons for file operations, query execution, and settings. The 'Query' tab is active, displaying the following SQL code:

```

1 SELECT id, name, surname
2 FROM students
3 WHERE id IN (
4     SELECT idstudent FROM summary WHERE grate = 5
5 );
6

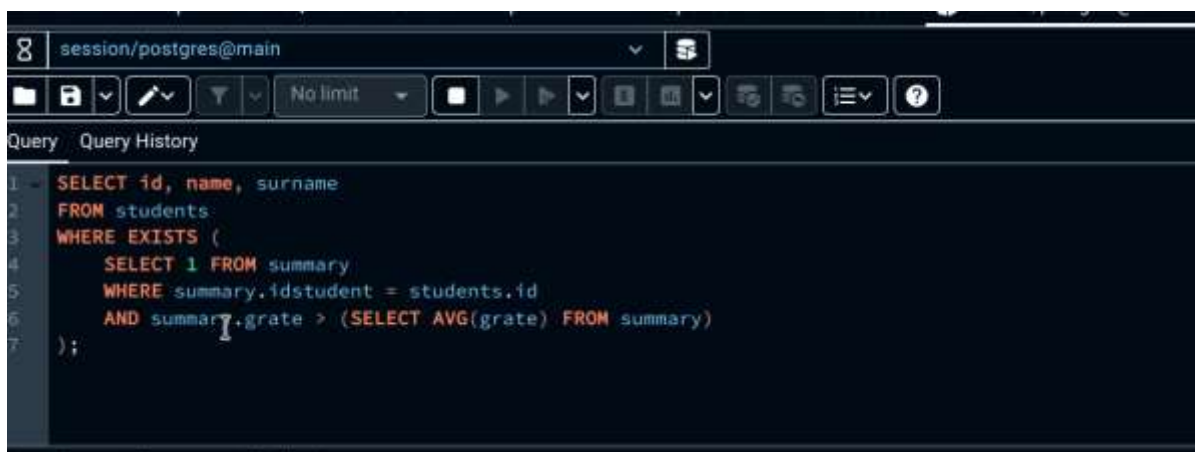
```

Below the query editor, the 'Data Output' tab is active, showing a table with 3 rows and 4 columns. The columns are 'id' (smallint, PK), 'name' (character varying (50)), 'surname' (character varying (50)), and an unlabeled column. The data is as follows:

	id [PK] smallint	name character varying (50)	surname character varying (50)
1	1	Виктор	Викторов
2	3	Семен	Семенов
3	5	Александра	Александрова

Рис. 9 Запрос 3

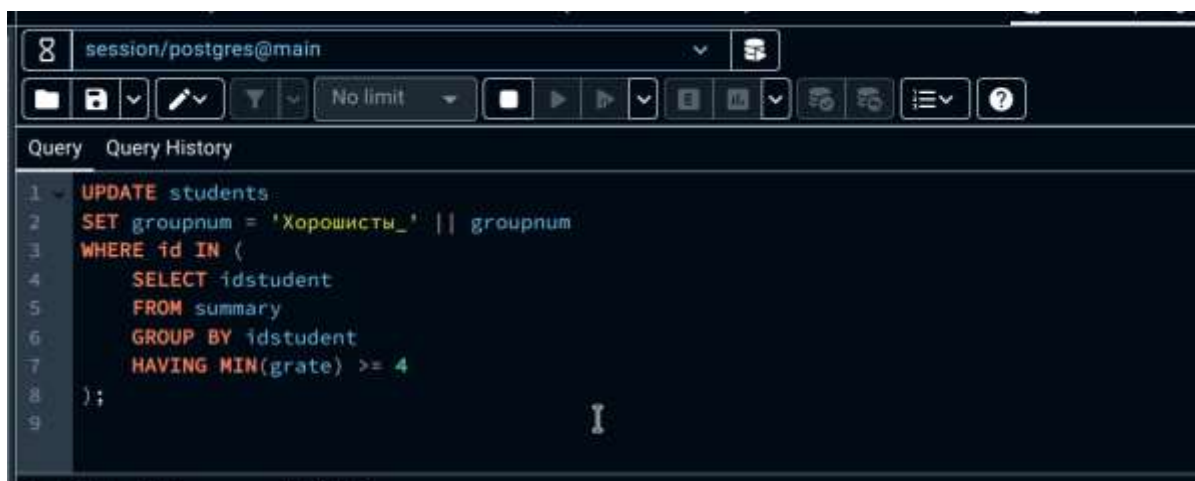
4. Создал запрос в соответствии с потребностью пользователя: Получить список студентов получивших оценку выше средней (см. рисунок 10)



```
1 SELECT id, name, surname
2 FROM students
3 WHERE EXISTS (
4     SELECT 1 FROM summary
5     WHERE summary.idstudent = students.id
6     AND summary.grate > (SELECT AVG(grate) FROM summary)
7 );
```

Рис. 10 Запрос 4

5. Создал запрос в соответствии с потребностью пользователя: Пометить группы студентов, оценка которых выше 4 (см. рисунок 11)



```
1 UPDATE students
2 SET groupnum = 'Хорошисты_' || groupnum
3 WHERE id IN (
4     SELECT idstudent
5     FROM summary
6     GROUP BY idstudent
7     HAVING MIN(grate) >= 4
8 );
```

Рис. 11 Запрос 5

Вывод: в ходе выполнения практической работы сформировали навыки работы в среде СУБД PostgreSQL по созданию вложенных запросов.