

Санкт–Петербургский Национальный Исследовательский Университет
Информационных технологий, Механики и Оптики

Лабораторная работа по дисциплине “Разработка и администрирование баз
данных”
на тему
“Написание SQL-запросов”

Выполнил: студент гр. D41421 Мамонова А.А.

г. Санкт–Петербург
2020 г.

Лабораторная работа № 3

Написание SQL-запросов

Цель работы: овладеть практическими навыками написания SQL запросов к ранее созданной базе данных.

Ход работы:

Вариант 8: создание системы для маркетинга рекламного агентства.

Схема базы данных отображена на рисунке 1. Вариант 8.

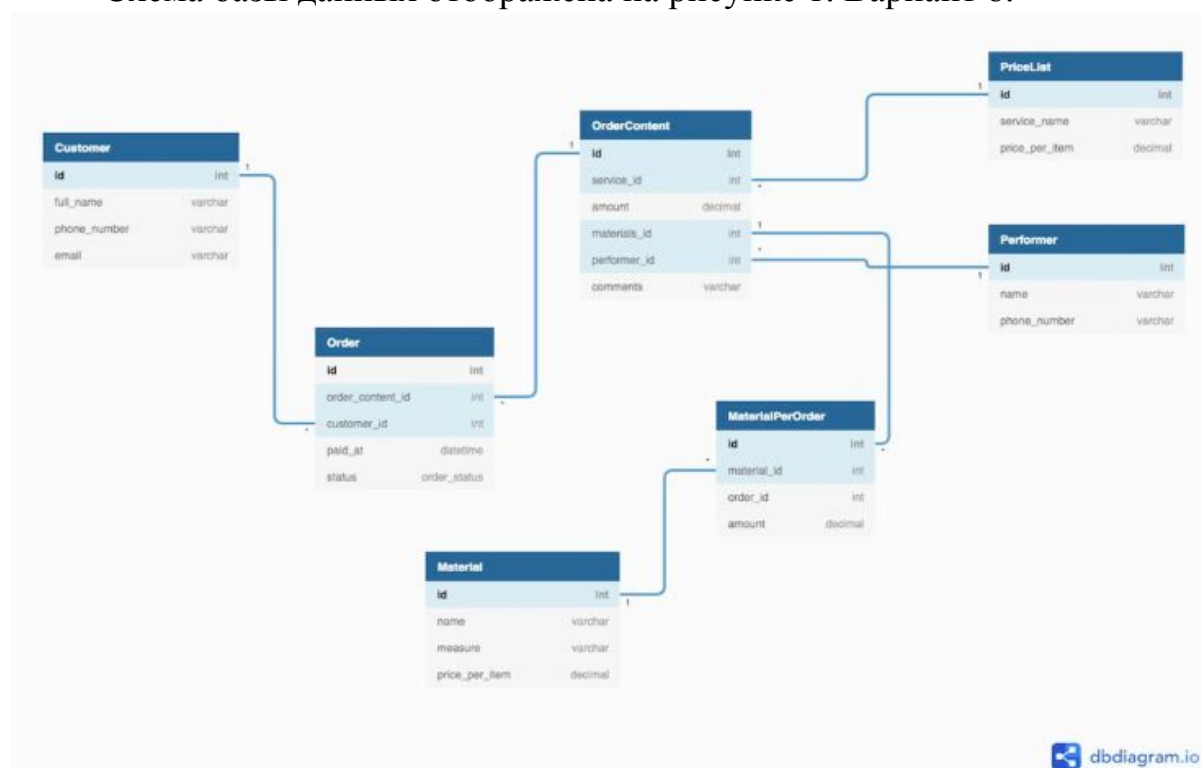


Рисунок 1. Схема БД

Запрос, результат выполнения для просмотра всех клиентов отображен на рисунке 2.

AdvertAgency/vladiknaska@PostgreSQL					
Query Editor Query History					
1 <code>select * from "AdvertAgencySchema"."Customer";</code>					
Data Output Explain Messages Notifications					
	id [PK] integer	full_name character (256)	phone_number character (30)	email character (256)	
1	0	Ivanov Petr	424242	ivanov@ya.ru	
2	1	Petrov Ivan	88005553535	petrov@ya.ru	
3	2	Mamonova Anastasiy...	900	mam@gmail.com	
4	3	Popova Katya	1234	popova@mail.ru	
5	4	Filimonova Katya	5555	fili@ya.ru	

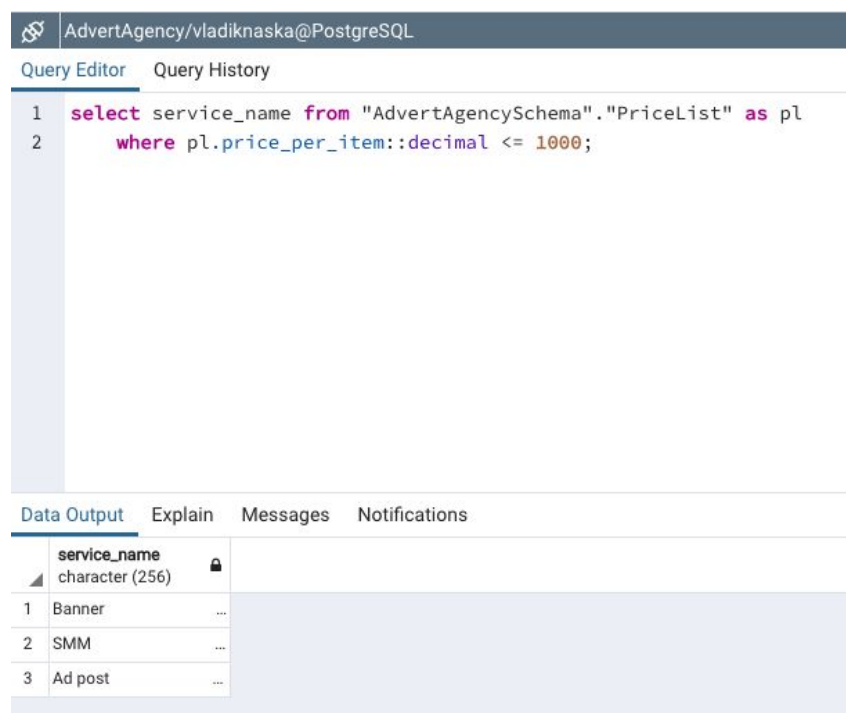
Рисунок 2. Просмотр всех клиентов

Запрос и результат его выполнения для просмотра всего списка услуг агенства представлены на рисунке 3.

AdvertAgency/vladiknaska@PostgreSQL			
Query Editor Query History			
1 <code>select * from "AdvertAgencySchema"."PriceList";</code>			
Data Output Explain Messages Notifications			
	id [PK] integer	service_name character (256)	price_per_item money
1	0	Banner	\$1,000.00
2	1	Promotion	\$4,200.00
3	2	Ad movie	\$9,000.00
4	3	SMM	\$1,000.00
5	4	Ad post	\$500.00

Рисунок 3. Таблица PriceList

Запрос и результат выполнения для нахождения услуг, цена которых меньше 1000 долларов представлены на рисунке 4.



AdvertAgency/vladiknaska@PostgreSQL

Query Editor Query History

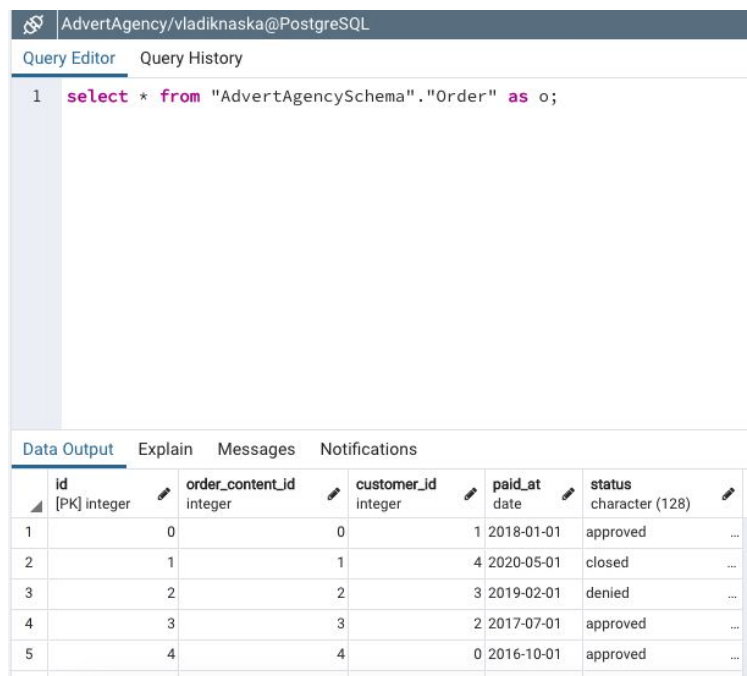
```
1 select service_name from "AdvertAgencySchema"."PriceList" as pl
2 where pl.price_per_item::decimal <= 1000;
```

Data Output Explain Messages Notifications

	service_name character (256)	
1	Banner	...
2	SMM	...
3	Ad post	...

Рисунок 4. Недорогие услуги

Запрос и результат его выполнения для вывода всей таблицы заказов представлены на рисунке 5.



AdvertAgency/vladiknaska@PostgreSQL

Query Editor Query History

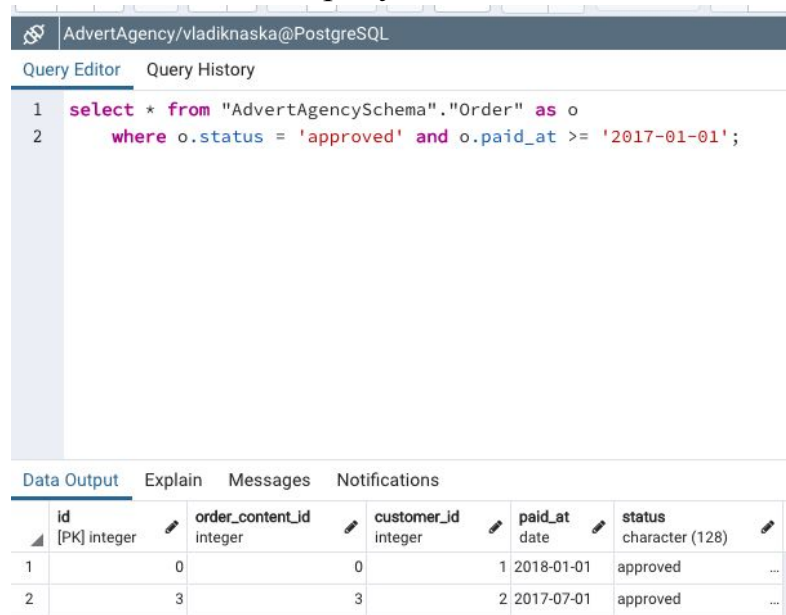
```
1 select * from "AdvertAgencySchema"."Order" as o;
```

Data Output Explain Messages Notifications

	id [PK] integer	order_content_id integer	customer_id integer	paid_at date	status character (128)	
1	0	0	1	2018-01-01	approved	...
2	1	1	4	2020-05-01	closed	...
3	2	2	3	2019-02-01	denied	...
4	3	3	2	2017-07-01	approved	...
5	4	4	0	2016-10-01	approved	...

Рисунок 5. Таблицы Orders

Запрос и результат его выполнения для вывода подтвержденных заказов с 2017 года показаны на рисунке 6.



The screenshot shows a PostgreSQL Query Editor window with the following SQL query:

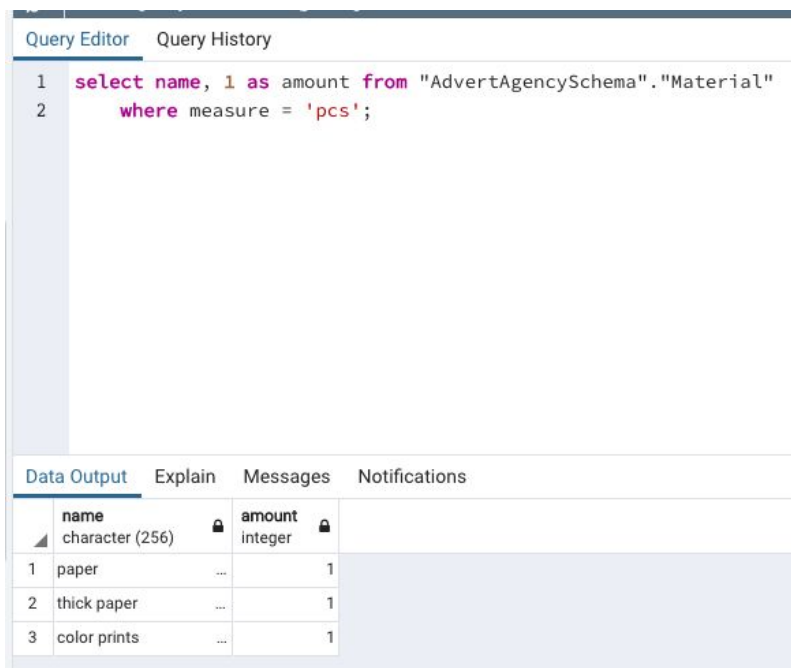
```
1 select * from "AdvertAgencySchema"."Order" as o
2 where o.status = 'approved' and o.paid_at >= '2017-01-01';
```

The results are displayed in a table with the following columns: id [PK] integer, order_content_id integer, customer_id integer, paid_at date, and status character (128). The results show two rows:

	id [PK] integer	order_content_id integer	customer_id integer	paid_at date	status character (128)
1	0	0	1	2018-01-01	approved
2	3	3	2	2017-07-01	approved

Рисунок 6. Подтвержденные заказы с 2017 года

Запрос и результат выполнения для нахождения материалов, которые измеряются в штуках и создание колонки количество со значением 1, рисунок 7.



The screenshot shows a PostgreSQL Query Editor window with the following SQL query:

```
1 select name, 1 as amount from "AdvertAgencySchema"."Material"
2 where measure = 'pcs';
```

The results are displayed in a table with the following columns: name character (256) and amount integer. The results show three rows:

	name character (256)	amount integer
1	paper	1
2	thick paper	1
3	color prints	1

Рисунок 7. Материалы, которые измеряются в шт

Просмотр всех клиентов, рисунок 8.

AdvertAgency/vladknaska@PostgreSQL				
Query Editor Query History				
1 select * from "AdvertAgencySchema"."Customer";				
Data Output Explain Messages Notifications				
	id [PK] integer	full_name character (256)	phone_number character (30)	email character (256)
1		0 Ivanov Petr	424242	ivanov@ya.ru
2		1 Petrov Ivan	88005553535	petrov@ya.ru
3		2 Mamonova Anastasiy...	900	mam@gmail.com
4		3 Popova Katya	1234	popova@mail.ru
5		4 Filimonova Katya	5555	filii@ya.ru


Рисунок 8. Все клиенты

Просмотр трех последних клиентов, отсортированных по алфавиту в обратном порядке, рисунок 9.

Query Editor Query History				
1 select full_name from "AdvertAgencySchema"."Customer"				
2 order by full_name desc				
3 limit 3;				
Data Output Explain Messages Notifications				
	full_name character (256)			
1	Popova Katya			
2	Petrov Ivan			
3	Mamonova Anastasiy...			

Рисунок 9. Три клиента

Количество элементов в таблицы Performer, рисунок 10.

 AdvertAgency/vladiknaska@PostgreSQL

Query Editor

Query History

1

```
select count(*) from "AdvertAgencySchema"."Performer";
```

Data Output

Explain

Messages

Notifications



	count bigint	
1	5	

Рисунок 10. Количество исполнителей

Применение агрегатной функции sum и условия where, рисунок 11.


AdvertAgency/vladiknaska@PostgreSQL

Query Editor
Query History

```

1  select sum(content.amount * price.price_per_item)
2  from "AdvertAgencySchema"."OrderContent" as content,
3  "AdvertAgencySchema"."PriceList" as price
4  where content.service_id=price.id;

```

Data Output
Explain
Messages
Notifications


	sum money	
1	\$27,600.00	

Рисунок 11. Подсчет заработанных денег

Просмотре контента таблицы MaterialPerOrder, рисунок 12.

AdvertAgency/vladiknaska@PostgreSQL				
Query Editor Query History				
1 select * from "AdvertAgencySchema"."MaterialPerOrder";				
Data Output Explain Messages Notifications				
	id [PK] integer	material_id integer	amount double precision	
1		0	1	2
2		1	1	10
3		2	2	100
4		3	3	1
5		4	4	1000

Рисунок 12. Таблицы MaterialPerOrder

Использование группировки и условия having, рисунок 13.

AdvertAgency/vladiknaska@PostgreSQL				
Query Editor Query History				
1 select material_id, sum (amount) from "AdvertAgencySchema"."MaterialPerOrder"				
2 group by material_id				
3 having sum (amount) > 10;				
Data Output Explain Messages Notifications				
	material_id integer	sum double precision		
1	4	1000		
2	2	100		
3	1	12		

Рисунок 13. id материалов, количество коротых в сумме больше 10

Использование предиката any, рисунок 14.

AdvertAgency/vladiknaska@PostgreSQL	
Query Editor	Query History
<pre> 1 select service_name from "AdvertAgencySchema"."PriceList" 2 where price_per_item::decimal > 3 any(select price_per_item::decimal from "AdvertAgencySchema"."Material"); </pre>	
Data Output	Explain Messages Notifications
<div>service_name</div> <div>character (256)</div>	
1 Banner ...	
2 Promotion ...	
3 Ad movie ...	
4 SMM ...	
5 Ad post ...	

Рисунок 14. Услуги с ценой выше, чем любая цена за материал

Использование объединения таблицы, рисунок 15.

AdvertAgency/vladiknaska@PostgreSQL	
Query Editor	Query History
<pre> 1 select m_order.amount, material.name, material.price_per_item 2 from "AdvertAgencySchema"."MaterialPerOrder" as m_order 3 inner join "AdvertAgencySchema"."Material" as material 4 on m_order.material_id = material.id; </pre>	
Data Output	Explain Messages Notifications
<div>amount</div> <div>double precision</div>	<div>name</div> <div>character (256)</div>
price_per_item	money
1	2 thick paper ... \$10.00
2	10 thick paper ... \$10.00
3	100 hd camera ... \$15,000.00
4	1 magazine paper ... \$100.00
5	1000 color prints ... \$500.00

Рисунок 15. Объединение Material и MaterialPerItem

Использование нескольких объединений, рисунок 16.

AdvertAgency/vladiknaska@PostgreSQL

Query Editor

Query History

```
1 select p.full_name, pl.service_name
2 from "AdvertAgencySchema"."OrderContent" as o
3 inner join "AdvertAgencySchema"."Performer" as p
4 on o.performer_id = p.id
5 inner join "AdvertAgencySchema"."PriceList" as pl
6 on o.service_id = pl.id;
```

Data Output

Explain

Messages

Notifications

	<div>full_name</div> <div>character (256)</div>		<div>service_name</div> <div>character (256)</div>	
1	Vasin Oleg	...	Banner	...
2	Tiktok Petr	...	Promotion	...
3	Vasin Oleg	...	Ad movie	...
4	Tretyak Alina	...	SMM	...
5	Holod Liza	...	Ad post	...

Рисунок 16. Объединение нескольких таблицы

Использование подзапросов, рисунок 17.

AdvertAgency/vladknaska@PostgreSQL

Query Editor

Query History

```
1 select * from "AdvertAgencySchema"."Customer" as c
2 where c.id in
3 (select customer_id from "AdvertAgencySchema"."Order"
4  where paid_at > '2019-01-01');
```

Data Output

Explain

Messages

Notifications

	id [PK] integer	full_name character (256)	phone_number character (30)	email character (256)
1	3	Popova Katya	1234	popova@mail.ru
2	4	Filimonova Katya	5555	fili@ya.ru

Рисунок 17. Использование подзапроса

Использование строковой функции и вычисление агрегатной функции, рисунок 18.



The screenshot shows a PostgreSQL Query Editor window with the following SQL query:

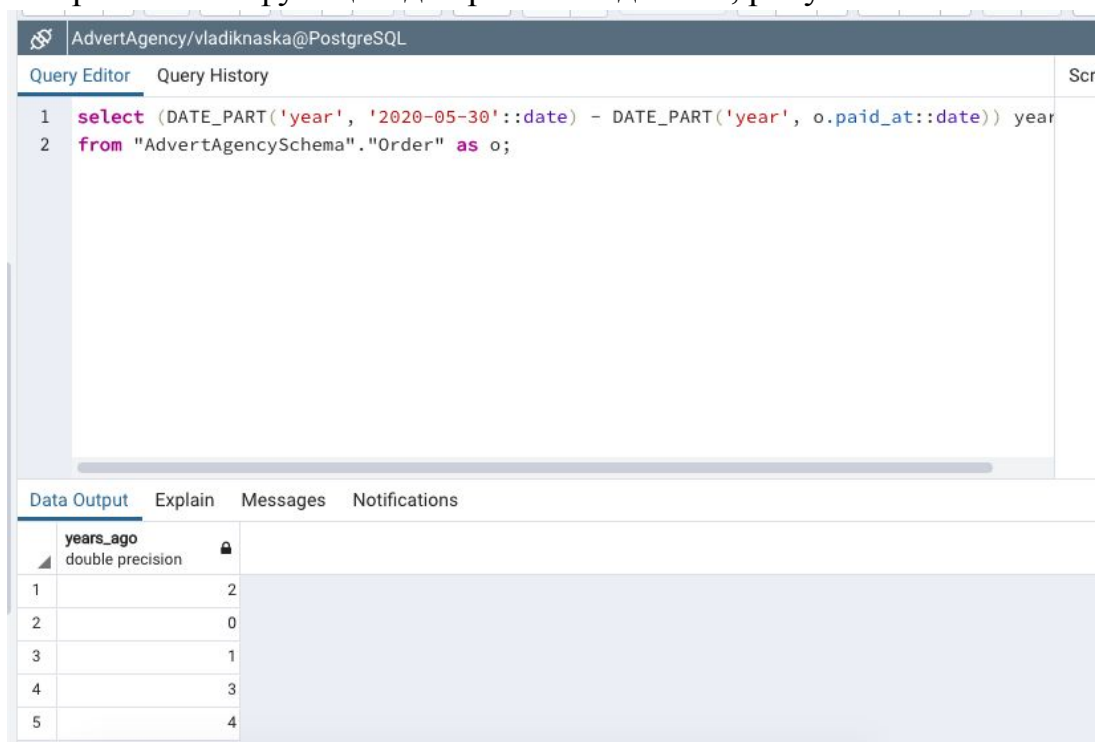
```
1 select upper(status), count(*) as Frequency
2 from "AdvertAgencySchema"."Order"
3 group by status;
```

The "Data Output" tab is active, displaying the results of the query in a table with two columns: "upper" (text) and "frequency" (bigint).

	upper text	frequency bigint
1	CLOSED	1
2	APPROVED	3
3	DENIED	1

Рисунок 18. Частота статусов

Применение функции для работы с датами, рисунок 19.



The screenshot shows a PostgreSQL Query Editor window with the following SQL query:

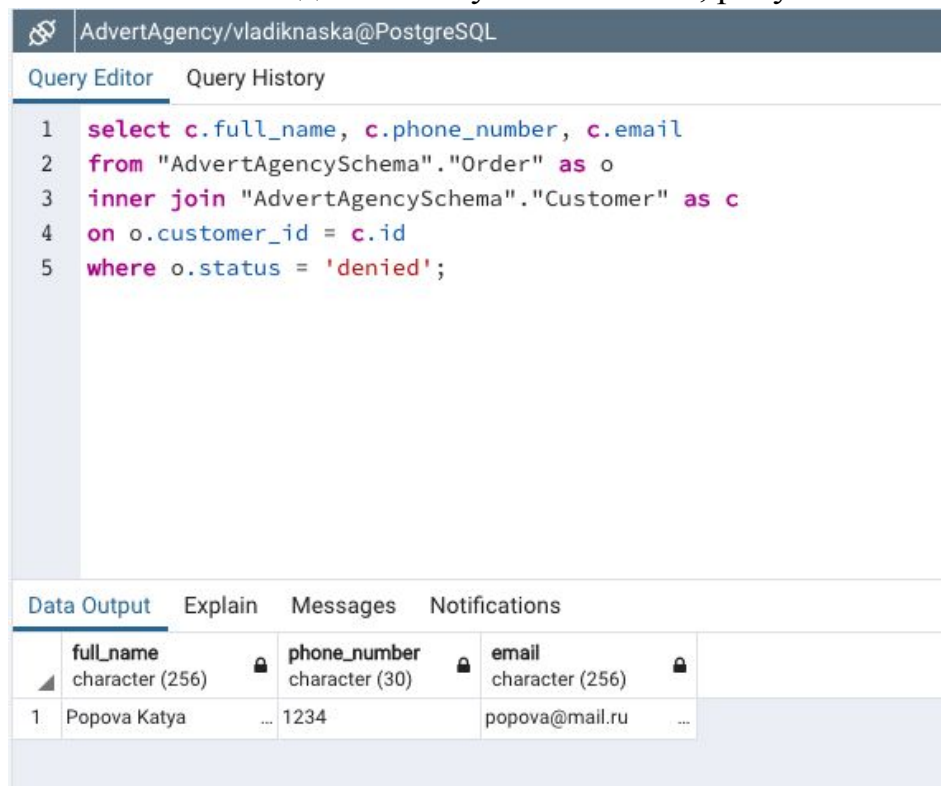
```
1 select (DATE_PART('year', '2020-05-30'::date) - DATE_PART('year', o.paid_at::date)) year
2 from "AdvertAgencySchema"."Order" as o;
```

The "Data Output" tab is active, displaying the results of the query in a table with two columns: "years_ago" (double precision) and an unlabeled column (bigint).

	years_ago double precision	
1		2
2		0
3		1
4		3
5		4

Рисунок 19. Сколько лет назад были сделаны заказы

Использование объединения и условия where, рисунок 20.



The screenshot shows a PostgreSQL Query Editor window. The title bar indicates the user is 'AdvertAgency/vladiknaska@PostgreSQL'. The 'Query Editor' tab is active, displaying the following SQL query:

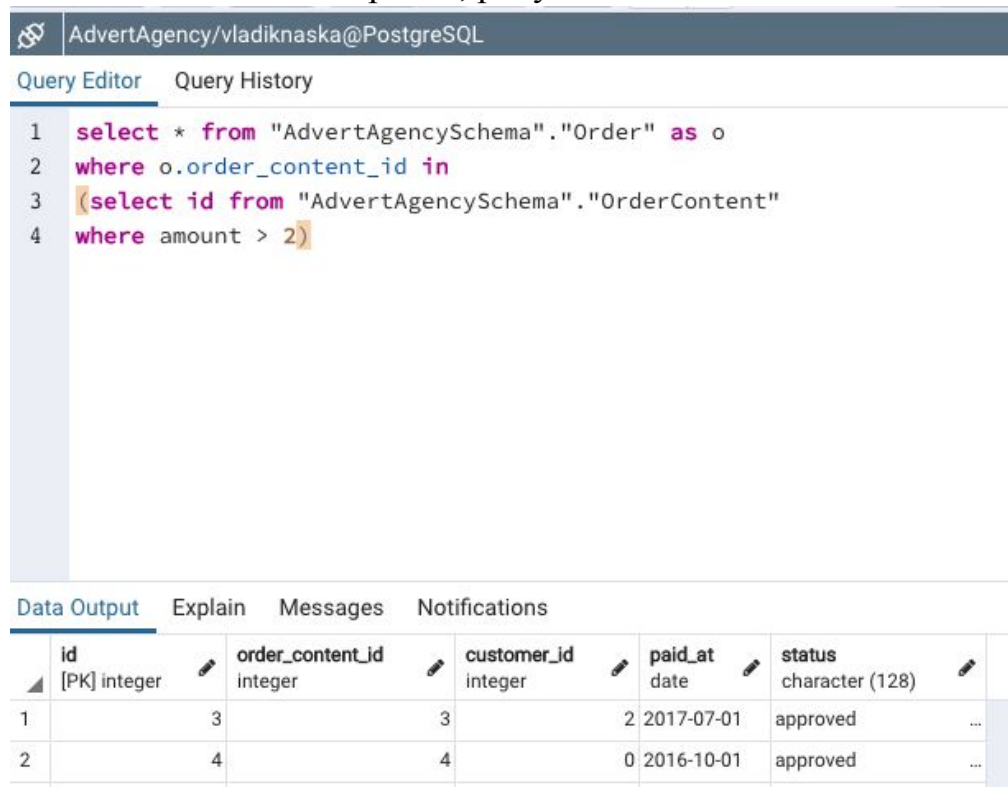
```
1 select c.full_name, c.phone_number, c.email
2 from "AdvertAgencySchema"."Order" as o
3 inner join "AdvertAgencySchema"."Customer" as c
4 on o.customer_id = c.id
5 where o.status = 'denied';
```

Below the query editor, the 'Data Output' tab is active, showing the results of the query in a table format. The table has three columns: 'full_name', 'phone_number', and 'email'. The data is as follows:

	full_name character (256)	phone_number character (30)	email character (256)
1	Popova Katya	1234	popova@mail.ru

Рисунок 20. Информация о клиентах, чьи заказы были отменены

Использование подзапросов, рисунок 21.



The screenshot shows a PostgreSQL Query Editor window. The title bar indicates the user is 'AdvertAgency/vladiknaska@PostgreSQL'. The 'Query Editor' tab is active, displaying the following SQL query:

```
1 select * from "AdvertAgencySchema"."Order" as o
2 where o.order_content_id in
3 (select id from "AdvertAgencySchema"."OrderContent"
4 where amount > 2)
```

Below the query editor, the 'Data Output' tab is active, showing the results of the query in a table format. The table has six columns: 'id', 'order_content_id', 'customer_id', 'paid_at', and 'status'. The data is as follows:

	id [PK] integer	order_content_id integer	customer_id integer	paid_at date	status character (128)
1	3	3	2	2017-07-01	approved
2	4	4	0	2016-10-01	approved

Рисунок 21. Заказы с больше чем двумя компонентами

Вывод: в ходе выполнения данной лабораторной работы получены практические навыки составления SQL-запросов.