

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

Факультет «Инфокоммуникационных технологий»

Направление подготовки «09.03.03 Мобильные и сетевые технологии»

О Т Ч Е Т

ЛАБОРАТОРНАЯ РАБОТА №2

**Тема
задания:**

**АНАЛИЗ ДАННЫХ. ПОСТРОЕНИЕ
ИНФОЛОГИЧЕСКОЙ МОДЕЛИ ДАННЫХ БД**

Выполнил:

Студент Ли В.Э K3240
(Фамилия И.О.) номер группы

Проверил:

Преподаватель Говоров А.И.
(Фамилия И.О)

Санкт-Петербург

2020

1 Цель работы:

Необходимо реализовать некоторое количество SQL-запросов.

2 Индивидуальное задание

Создать программную систему, предназначенную для отдела маркетинга рекламного агентства.

Одной из задач, решаемых отделом маркетинга рекламного агентства «Луч», является учет работы с клиентами. Для этого необходимо организовать оперативный учет поступивших и выполненных заявок клиентов (рекламодателей).

Рекламное агентство заключает трудовые соглашения с заказчиками на исполнение определенного вида рекламных услуг. Для оформления заявки рекламодатель должен указать контактное лицо, телефон и электронный адрес для связи. Рекламодатель оформляет заявку на рекламу, пользуясь прайс-листом, в котором указаны цены по наименованию рекламных услуг, предоставляемых агентством «Луч». Здесь же оговариваются исполнители изготовления рекламы (сотрудники агентства), стоимость и объем (количество) работ. Для выполнения работ необходимо знать единицы измерения и материалы. Заказчик должен иметь контактные данные исполнителя

Согласно заявке, выписывается Платежное Поручение Заказчику, которое он обязан оплатить.

После оплаты счета агентство обязуется предоставить рекламные продукты. Заказ считается выполненным, если оплачено Платежное поручение.

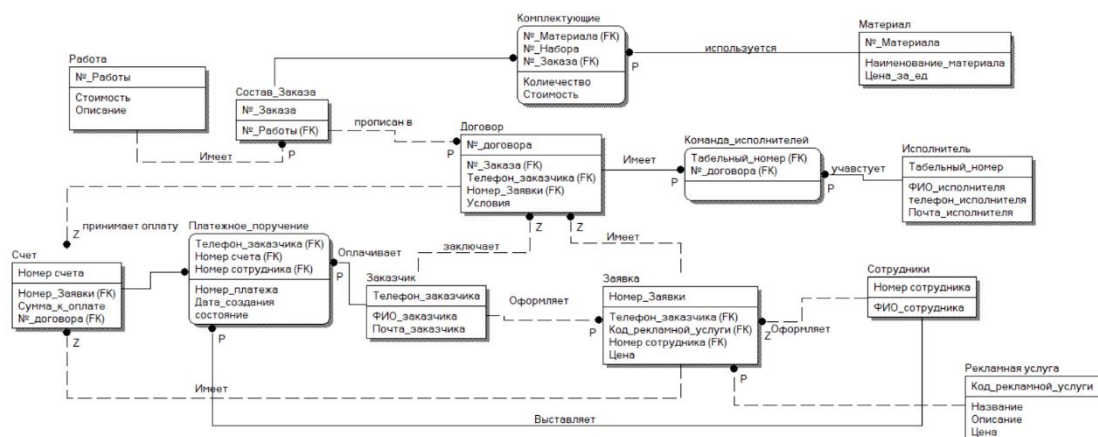
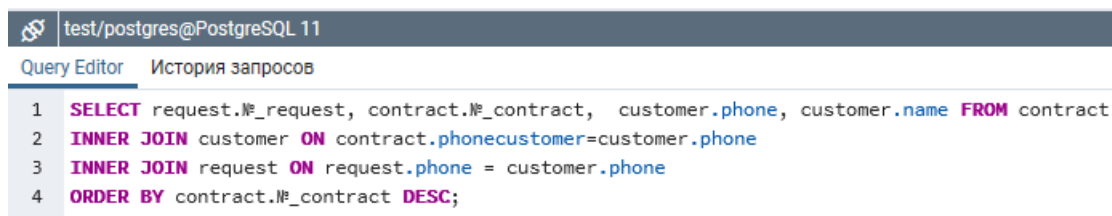


Рисунок 1

3 Список запросов

Запрос первый.

Запрашиваю и объединяю номер заявки из таблицы заявок, номер контракта из таблицы контрактов, телефон клиента и ФИО клиента из таблицы клиентов из таблицы контрактов, при этом сортирую по убыванию (рис 2).



```
test/postgres@PostgreSQL 11
Query Editor  История запросов
1 SELECT request.№_request, contract.№_contract, customer.phone, customer.name FROM contract
2 INNER JOIN customer ON contract.phonecustomer=customer.phone
3 INNER JOIN request ON request.phone = customer.phone
4 ORDER BY contract.№_contract DESC;
```

Рис. 2. Запрос первый

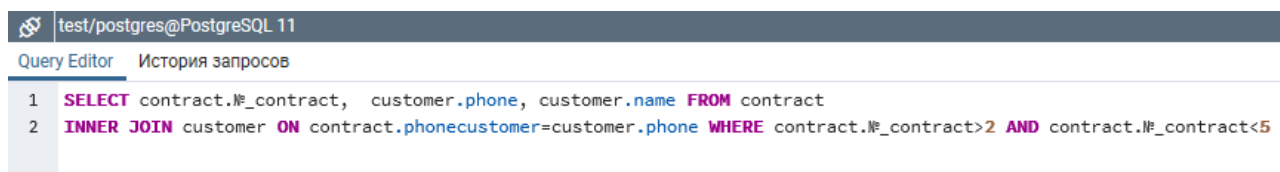
На рисунке 3 представлен результат первого запроса.

	№_request integer	№_contract integer	phone character varying (12)	name character varying (50)
1		5	+72342389374	Игорь Кокорин Игоревич
2		4	+72312884374	Денис Петров Игоревич
3		3	+72342889394	Гриша Ульянов Игоревич
4		2	+72342889324	Василий Ульянов Игоревич
5		1	+72342889374	Дмитрий Ульянов Игоревич

Рис. 3. Результат первого запроса

Запрос второй.

Ограничивая между двумя и пятью номер контракта из балицы контрактов, запрашиваю и объединяю номер контракта из таблицы контрактов, телефон клиента и ФИО клиента из таблицы клиентов (рис 4) .



```
test/postgres@PostgreSQL 11
Query Editor  История запросов
1 SELECT contract.№_contract, customer.phone, customer.name FROM contract
2 INNER JOIN customer ON contract.phonecustomer=customer.phone WHERE contract.№_contract>2 AND contract.№_contract<5
```

Рис. 4. Запрос второй

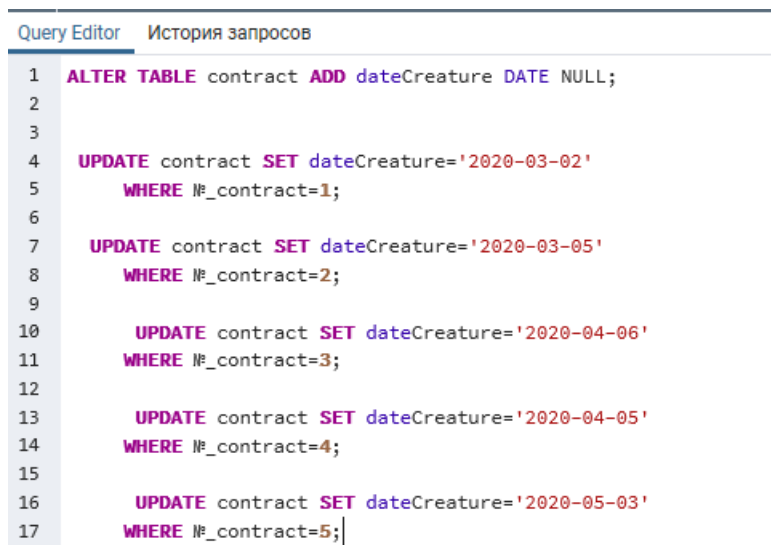
На рисунке 5 представлен результат второго запроса.

	№_contract integer	phone character varying (12)	name character varying (50)
1	3	+72342889394	Гриша Ульянов Игоревич
2	4	+72312884374	Денис Петров Игоревич

Рис. 5. Результат второго запроса

Запрос третий.

Для работы с функциями для работы с датами сделаем следующие запросы, представленные на рисунке 6:



```
Query Editor  История запросов
1 ALTER TABLE contract ADD dateCreate DATE NULL;
2
3
4 UPDATE contract SET dateCreate='2020-03-02'
5 WHERE №_contract=1;
6
7 UPDATE contract SET dateCreate='2020-03-05'
8 WHERE №_contract=2;
9
10 UPDATE contract SET dateCreate='2020-04-06'
11 WHERE №_contract=3;
12
13 UPDATE contract SET dateCreate='2020-04-05'
14 WHERE №_contract=4;
15
16 UPDATE contract SET dateCreate='2020-05-03'
17 WHERE №_contract=5;
```

Рис. 6.Добавление дат

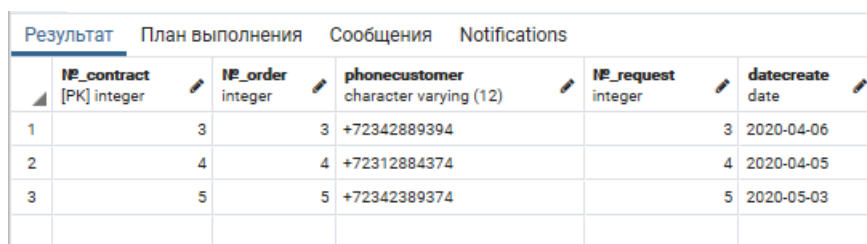
Запросим данные за последние 50 дней из таблицы контрактов. Пример этого представлен на рисунке 7.



```
Query Editor  История запросов
1 SELECT * FROM contract WHERE datecreate> NOW() - INTERVAL '50' DAY;
```

Рис. 7. Запрос третий.

На рисунке 5 представлен результат второго запроса.



Результат		План выполнения	Сообщения	Notifications	
№_contract [PK] integer	№_order integer	phonecustomer character varying (12)	№_request integer	datecreate date	
1	3	+72342889394	3	2020-04-06	
2	4	+72312884374	4	2020-04-05	
3	5	+72342389374	5	2020-05-03	

Рис. 8. Результат 3 запроса

Запрос четвертый.

Запросим ФИО и номер из таблицы заказчиков конкатенируя их с помощью функции CONCAT(), для более удобного чтения, и объединив с номером контрактов из таблицы контрактов (рис 9).

Query Editor История запросов	
1	<code>SELECT №_contract, CONCAT('ФИО: ', name, '; номер: ', phone) AS "информация" FROM customer</code>
2	<code>INNER JOIN contract ON customer.phone = contract.phonecustomer</code>

Рис. 9.запрос четвертый

Результат четвертого запроса представлен на рисунке 10.

Результат План выполнения Сообщения Notifications			
	№_contract [PK] integer	информация text	
1	1	ФИО: Дмитрий Ульянов Игоревич; номер: +72342889374	
2	2	ФИО: Василий Ульянов Игоревич; номер: +72342889324	
3	3	ФИО: Гриша Ульянов Игоревич; номер: +72342889394	
4	4	ФИО: Денис Петров Игоревич; номер: +72312884374	
5	5	ФИО: Инорь Кокорин Игоревич; номер: +72342389374	

Рисунок 10. Результаты четвертого запроса

Запрос пятый.

Запрашиваю номер работы и цену, где цена за работу, указанной в составе заказа, который входит в контракт который был составлен 2020-03-05 числа, является максимальной.

Query Editor История запросов	
1	<code>SELECT №_work, sell</code>
2	<code>FROM work</code>
3	<code>WHERE work.sell =</code>
4	<code>(</code>
5	<code>SELECT MAX(sell)</code>
6	<code>FROM work WHERE №_work =</code>
7	<code>(</code>
8	<code>SELECT №_work</code>
9	<code>FROM orderlist</code>
10	<code>WHERE №_order =</code>
11	<code>(</code>
12	<code>SELECT №_order</code>
13	<code>FROM contract</code>
14	<code>WHERE datecreate = '2020-03-05'</code>
15	<code>)</code>
16	<code>)</code>
17	<code>)</code>

Рис. 11. Запрос пятый

Результат запроса представлен на рисунке 12.

	№_work [PK] integer	sell double precision
1	3	230344

Рис. 13. Результат пятого запроса

Запрос шестой

Запрашиваю №_сотрудника, количество записей сотрудника и среднюю сумму всех заявок с участием сотрудника.

```
test/postgres@PostgreSQL 11
Query Editor  История запросов
1 SELECT №_employees, COUNT(№_employees) AS Qty_№_employees,
2     AVG(request.price) AS Avg_price
3 FROM employees
4 INNER JOIN request ON №_employees=request.id_employee
5 GROUP BY №_employees;
6
```

Рис. 14. Запрос шестой.

Результат 6 запроса представлен на рисунке 14.

	№_employees [PK] integer	qty_№_employees bigint	avg_price double precision
1	4	1	54332
2	2	1	23432
3	3	1	2343
4	5	1	54532
5	1	2	73677

Рис. 15. Результат шестого запроса.

Запрос седьмой.

Запрашиваю, количество заявок, с которыми работал сотрудник с id равным 1.

```
Query Editor  История запросов
1 SELECT count(request.id_employee) FROM request
2 GROUP BY request.id_employee
3 having request.id_employee = 1
```

Рис. 16. Запрос седьмой

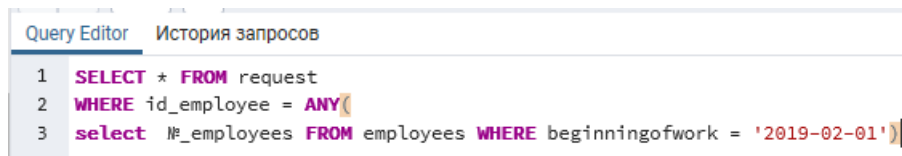
Результат седьмого запроса представлен на рисунке 16.

	count bigint
1	2

Рис. 17. Результат седьмого запроса

Запрос восьмой.

8.1 Запрашиваю все данные о заявках, которые оформляли сотрудники, принятые на работу 2019-02-01 числа.



```
1 SELECT * FROM request
2 WHERE id_employee = ANY(
3 select id_employees FROM employees WHERE beginningofwork = '2019-02-01')
```

Рис. 18. Запрос 8.1

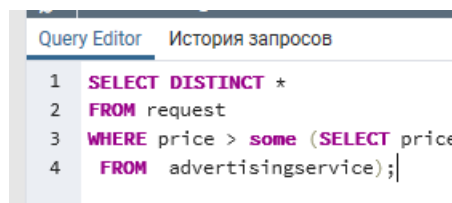
Результат восьмого запроса представлен на рисунке 18.



	phone character varying (12)	id_request [PK] integer	id_advertising integer	id_employee integer	price double precision
1	+72342889374	1	1	1	123433
2	+72342889324	2	2	2	23432
3	+72342889374	6	2	1	23921

Рис. 19. Результат запроса 8.1

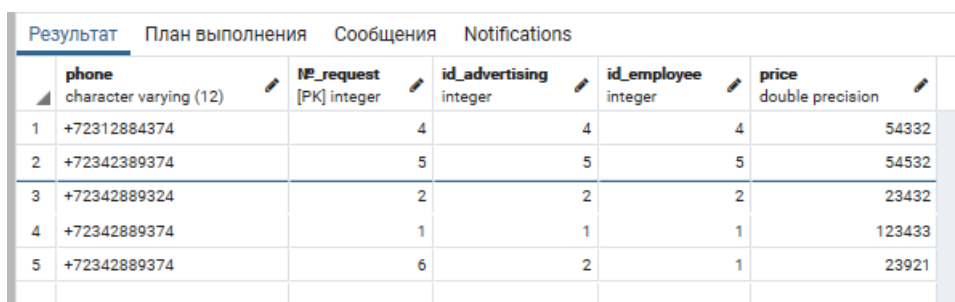
8.2 Запрашиваю все данные о заявках, цена которых превосходит какую-нибудь рекламную цену.



```
1 SELECT DISTINCT *
2 FROM request
3 WHERE price > some (SELECT price
4 FROM advertisingservice);
```

Рис. 20. Запрос 8.2.

Результат запроса 8.2 представлен на рисунке 20.



	phone character varying (12)	id_request [PK] integer	id_advertising integer	id_employee integer	price double precision
1	+72312884374	4	4	4	54332
2	+72342389374	5	5	5	54532
3	+72342889324	2	2	2	23432
4	+72342889374	1	1	1	123433
5	+72342889374	6	2	1	23921

Рисунок 21. Результат запроса 8.2.

8.3 запрашиваю все данные о заявках, цена которых превосходит цену всех реклам.

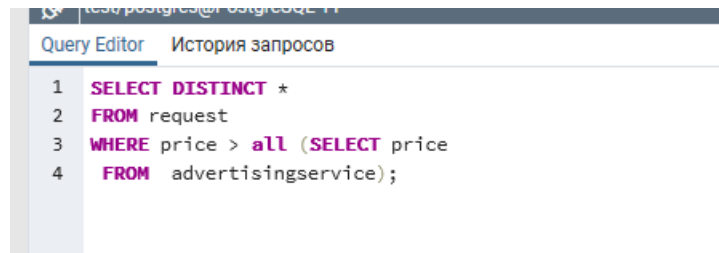


Рис. 21. Запрос 8.3.

Результат запроса 8.3. представлен на рисунке 22.

Результат					
	phone	№_request	id_advertising	id_employee	price
	character varying (12)	[PK] integer	integer	integer	double precision
1	+72342889374	1	1	1	123433

Рис. 22. Результат запроса 8.3.

Запрос девятый.

Запрашиваю всех сотрудников, которые не оформляли ни одной заявки.

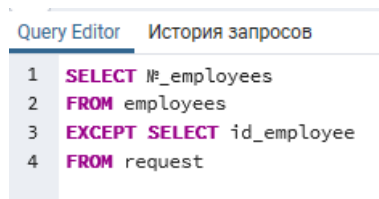


Рис. 23. Запрос девятый

Результат девятого запроса представлен на рисунке 24.

Результат		План вы
	№_employees	
	integer	
1		6
2		7

Рисунок 24

Запрос десятый.

Запрашиваю номер контракта и дату создания контракта из таблицы контрактов и имя заказчика, так, что номер заказчика и номер заказчика в контракте совпадают.

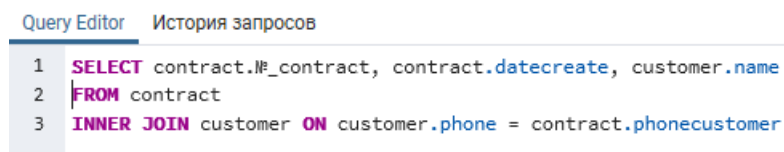


Рис. 25..

Результат десятого запроса представлен на рисунке 26.

Результат				План выполнения	Сообщения	Notifications
	IP_contract integer		datecreate date		name character varying (50)	
1		1	2020-03-02		Дмитрий Ульянов Игоревич	
2		2	2020-03-05		Василий Ульянов Игоревич	
3		3	2020-04-06		Гриша Ульянов Игоревич	
4		4	2020-04-05		Денис Петров Игоревич	
5		5	2020-05-03		Инорь Кокорин Игоревич	

Рис. 26. Результат десятого запроса.

Вывод: В ходе лабораторной работы были получены навыки запросов PostgreSQL.