



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО  
ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ  
(национальный исследовательский университет)»

---

Институт № 3 «Системы управления, информатика и электроэнергетика»  
Кафедра 311 «Прикладные программные средства и математические методы»

## РАБОТА

Дисциплина: **«Информационные системы»**

Выполнила:

Студентка гр. **МЗО-216Бк-22**

**Хутиева Эрика Арсеновна**

Оценка КР(КП): \_\_\_\_\_

Москва **2023г.**

## **Оглавление.**

Лабораторная №2.....	3
Лабораторная №3.....	22
Лабораторная №4.....	25
Лабораторная №5.....	29
Вывод.....	44
Список литературы.....	45

### Задание 2.1.

1. Выбрать все данные из таблицы customers

Query

Query History

Scratch Pad

1

2

3

4

5

```
SELECT *
FROM customers
```

Data Output

Messages

Notifications

≡+

▼

▼

	customer_id [PK] character	company_name character varying (40)	contact_name character varying (30)	contact_title character varying (30)
1	ALFKI	Alfreds Futterkiste	Maria Anders	Sales Representative
2	ANATR	Ana Trujillo Emparedados y helados	Ana Trujillo	Owner
3	ANTON	Antonio Moreno Taquería	Antonio Moreno	Owner
4	AROUT	Around the Horn	Thomas Hardy	Sales Representative
5	BERGS	Berglunds snabbköp	Christina Berglund	Order Administrator
6	BLAUS	Blauer See Delikatessen	Hanna Moos	Sales Representative
7	BLONP	Blondesddsl père et fils	Frédérique Citeaux	Marketing Manager
8	BOLID	Bólido Comidas preparadas	Martin Sommer	Owner

2. Выбрать все записи из таблицы customers, но только колонки "имя контакта" и "Город"

Query

Query History

↗

Scratch Pad

1

2

3

4

5

SELECT

contact\_name, city

FROM

customers

Data Output

Messages

Notifications

≡+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

	contact_name character varying (30) 🔒	city character varying (15) 🔒
1	Maria Anders	Berlin
2	Ana Trujillo	México D.F.
3	Antonio Moreno	México D.F.
4	Thomas Hardy	London
5	Christina Berglund	Luleå
6	Hanna Moos	Mannheim
7	Frédérique Citeaux	Strasbourg
8	Martín Sommer	Madrid
9	Laurence Leblanc	Marseille

3. Выбрать все записи из таблицы orders, но взять две колонки: идентификатор заказа и колонку, значение в которой мы рассчитываем как разницу между датой отгрузки и датой формирования заказа.

```
1 select order_id, shipped_date- order_date
2 from orders
```

Data Output Messages Notifications



	order_id [PK] smallint	?column? integer
1	10248	12
2	10249	5
3	10250	4
4	10251	7
5	10252	2
6	10253	6
7	10254	12
8	10255	3
9	10256	2

4. Выбрать все уникальные города в которых "зарегистрированы" заказчики

```
1 select DISTINCT city
2 from customers
```

Data Output Messages Notifications

city	
character varying (15)	
Butte	
Paris	
Cowes	
Cork	
Bern	
Brandenburg	
Aachen	
Walla Walla	
Warszawa	

5. Выбрать все уникальные сочетания городов и стран в которых "зарегистрированы" заказчики

```
1 select DISTINCT country,city
2 from customers
```

Data Output Messages Notifications

	country character varying (15) 🔒	city character varying (15) 🔒
1	Germany	Leipzig
2	Denmark	Århus
3	Canada	Vancouver
4	Sweden	Luleå
5	USA	Kirkland
6	Venezuela	San Cristóbal
7	USA	Portland
8	Spain	Sevilla
9	Brazil	Rio de Janeiro

6. Посчитать кол-во заказчиков

```
1 select distinct count(*)
2 from customers
```

Data Output			Messages	Notifications
	count	bigint		
1		91		

7. Посчитать кол-во уникальных стран в которых "зарегистрированы" заказчики

```
1 SELECT COUNT (DISTINCT country)
2 from customers
```

Data Output			Messages	Notifications
	count	bigint		
1		21		



## Задание 2.2.

### 1. Выбрать все заказы из стран France, Austria, Spain

Query

Query History

Scratch Pad

```
1 SELECT *
2 FROM customers
3 WHERE country in ('France', 'Austria', 'Spain')
4
```

Data Output

Messages

Notifications

	city character varying (15)	region character varying (15)	postal_code character varying (10)	country character varying (15)	phone character varying (24)	fax character varying (24)
1	Strasbourg	[null]	67000	France	88.60.15.31	88.60.15.32
2	Madrid	[null]	28023	Spain	(91) 555 22 82	(91) 555 91 99
3	Marseille	[null]	13008	France	91.24.45.40	91.24.45.41
4	Nantes	[null]	44000	France	40.67.88.88	40.67.89.89
5	Graz	[null]	8010	Austria	7675-3425	7675-3426
6	Madrid	[null]	28034	Spain	(91) 555 94 44	(91) 555 55 93
7	Lille	[null]	59000	France	20.16.10.16	20.16.10.17
8	Nantes	[null]	44000	France	40.32.21.21	40.32.21.20

Total rows: 19 of 19

Query complete 00:00:00.100

Ln 3, Col 15

### 2. Выбрать все заказы, отсортировать по required\_date (по убыванию) и отсортировать по дате отгрузке (по возрастанию)

1 SELECT \*  
2 FROM orders  
3 order by required\_date DESC;  
4  
5

Data OutputMessagesNotifications

	order_id [PK] smallint	customer_id character	employee_id smallint	order_date date	required_date date	shipped_date date	ship_via smallint	freight real	ship_name character varying (40)	shi cha
1	11061	GREAL	4	1998-04-30	1998-06-11	[null]	3	14.01	Great Lakes Food Market	27
2	11059	RICAR	2	1998-04-29	1998-06-10	[null]	2	85.8	Ricardo Adocicados	Av
3	11075	RICSU	8	1998-05-06	1998-06-03	[null]	2	6.19	Richter Supermarkt	Sta
4	11074	SIMOB	7	1998-05-06	1998-06-03	[null]	2	18.44	Simons bistro	Vir
5	11076	BONAP	4	1998-05-06	1998-06-03	[null]	2	38.28	Bon app'	12
6	11077	RATTC	1	1998-05-06	1998-06-03	[null]	2	8.53	Rattlesnake Canyon Grocery	28
7	11072	ERNSH	4	1998-05-05	1998-06-02	[null]	2	258.64	Ernst Handel	Kir
8	11071	LILAS	1	1998-05-05	1998-06-02	[null]	1	0.93	LILA-Supermercado	Ca

```

1 SELECT *
2 FROM orders
3 order by shipped_date ASC;
4
5

```

Data Output Messages Notifications

	order_id [PK] smallint	customer_id character	employee_id smallint	order_date date	required_date date	shipped_date date	ship_via smallint	freight real	ship_name character varying (40)	shi che
1	10249	TOMSP	6	1996-07-05	1996-08-16	1996-07-10	1	11.61	Toms Spezialitäten	Lu
2	10252	SUPRD	4	1996-07-09	1996-08-06	1996-07-11	2	51.3	Suprêmes délices	Bo
3	10250	HANAR	4	1996-07-08	1996-08-05	1996-07-12	2	65.83	Hanari Carnes	Ru
4	10255	RICSU	9	1996-07-12	1996-08-09	1996-07-15	3	148.33	Richter Supermarkt	Sti
5	10251	VICTE	3	1996-07-08	1996-08-05	1996-07-15	1	41.34	Victuailles en stock	2, i
6	10253	HANAR	3	1996-07-10	1996-07-24	1996-07-16	2	58.17	Hanari Carnes	Ru
7	10248	VINET	5	1996-07-04	1996-08-01	1996-07-16	3	32.38	Vins et alcools Chevalier	59
8	10256	WELLI	3	1996-07-15	1996-08-12	1996-07-17	2	13.97	Wellington Importadora	Ru

3. Выбрать минимальное кол-во единиц товара среди тех продуктов, которых в продаже более 30 единиц.

Query Query History

```

1 select min(units_in_stock) as min_units
2 from products
3 where units_in_stock>30

```

Data Output Messages Notifications

	min_units smallint
1	31

4. Выбрать максимальное кол-во единиц товара среди тех продуктов, которых в продаже более 30 единиц.

Query

Query History

1

`select max(units_in_stock) as min_units`

2

`from products`










3

`where units_in_stock>30`

Data Output

Messages

Notifications













	<div>min_units</div> <div>smallint</div> <div></div>
1	125

5. Найти среднее значение дней уходящих на доставку с даты формирования заказа в USA

Query Query History

```
1 select avg(required_date-order_date) as avg_delivery_days
2 from orders
3 where ship_country= 'USA'
```

Data Output Messages Notifications

        		
	avg_delivery_days numeric	
1	28.5737704918032787	

6. Найти сумму, на которую имеется товаров (кол-во \* цену) причём таких, которые планируется продавать и в будущем (см. на поле discontinued)

Query

Query History

1

select sum(unit\_price \* units\_in\_stock) as bank  
 from products  
 where discontinued <> 1  
 4

Data Output

Messages

Notifications

+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

bank

double precision

🔒

1

68475.74997138977

Задание 3

1. Выбрать все записи заказов, в которых наименование страны отгрузки начинается с 'U'

Query

Query History

1

SELECT \*  
 FROM orders  
 WHERE ship\_country::text LIKE 'U%'  
 4

Data Output

Messages

Notifications

+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

ship\_name

character varying (40)

✎

ship\_address

character varying (60)

✎

ship\_city

character varying (15)

✎

ship\_region

character varying (15)

✎

ship\_postal\_code

character varying (10)

✎

ship\_country

character varying (15)

✎

1

Rattlesnake Canyon Grocery

2817 Milton Dr.

Albuquerque

NM

87110

USA

2

White Clover Markets

1029 - 12th Ave. S.

Seattle

WA

98124

USA

3

Split Rail Beer & Ale

P.O. Box 555

Lander

WY

82520

USA

4

Rattlesnake Canyon Grocery

2817 Milton Dr.

Albuquerque

NM

87110

USA

5

B's Beverages

Fauntleroy Circus

London

[null]

EC2 5NT

UK

6

Rattlesnake Canyon Grocery

2817 Milton Dr.

Albuquerque

NM

87110

USA

7

Old World Delicatessen

2743 Bering St.

Anchorage

AK

99508

USA

8

Lonesome Pine Restaurant

89 Chiaroscuro Rd.

Portland

OR

97219

USA

2. Выбрать записи заказов (включить колонки идентификатора заказа, идентификатора заказчика, веса и страны отгрузки), которые должны быть отгружены в страны имя которых начинается с 'N', отсортировать по весу (по убыванию) и вывести только первые 10 записей.

```
1 SELECT order_id,orders.customer_id,freight, ship_country
2 FROM orders
3 WHERE ship_country::text LIKE 'N%'
4
```

Data Output Messages Notifications



	order_id [PK] smallint	customer_id character	freight real	ship_country character varying (15)
1	10387	SANTG	93.63	Norway
2	10520	SANTG	13.37	Norway
3	10639	SANTG	38.64	Norway
4	10831	SANTG	72.19	Norway
5	10909	SANTG	53.05	Norway
6	11015	SANTG	4.62	Norway

3. Выбрать записи работников (включить колонки имени, фамилии, телефона, региона) в которых регион неизвестен

Query Query History

```
1 select first_name,last_name,birth_date,region
2 from employees
3 where region is null
4
```

Data Output Messages Notifications

	first_name character varying (10) 🔒	last_name character varying (20) 🔒	birth_date date 🔒	region character varying (15) 🔒
1	Steven	Buchanan	1955-03-04	[null]
2	Michael	Suyama	1963-07-02	[null]
3	Robert	King	1960-05-29	[null]
4	Anne	Dodsworth	1966-01-27	[null]

4. Подсчитать кол-во заказчиков регион которых известен

Query

Query History

1

select count(\*)

2

from customers

3

where region is not null

4

Data Output

Messages

Notifications

≡+

▼

▼

	count bigint	
1	31	

5. Подсчитать кол-во поставщиков в каждой из стран и отсортировать результаты группировки по убыванию кол-ва



Query
Query History

```

1 select country, count(*) as suppliers_count
2 from suppliers
3 GROUP BY country
4 order by suppliers_count desc
5

```

Data Output
Messages
Notifications

+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

	country character varying (15)	suppliers_count bigint
1	USA	4
2	France	3
3	Germany	3
4	Italy	2
5	Sweden	2
6	UK	2
7	Australia	2
8	Japan	2
9	Canada	2

Total rows: 16 of 16
Query complete 00:00:00.156

- Подсчитать суммарный вес заказов (в которых известен регион) по странам, затем отфильтровать по суммарному весу (вывести только те записи где суммарный вес больше 2750) и отсортировать по убыванию суммарного веса.

Query
Query History

```

1 select ship_country, sum(freight) as summa
2 from orders
3 where ship_country is not null
4 group by ship_country
5 Having sum(freight)>2750
6 order by sum(freight) desc
7

```

Sc

Data Output
Messages
Notifications

+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

	ship_country character varying (15)	summa real
1	USA	13771.285
2	Germany	11283.278
3	Austria	7391.5005
4	Brazil	4880.19
5	France	4237.8394
6	Sweden	3237.6003
7	UK	2954.27
8	Ireland	2755.24

Total rows: 8 of 8
Query complete 00:00:00.138

- Выбрать все уникальные страны заказчиков и поставщиков и отсортировать страны по возрастанию

Query

Query History

1

SELECT country

2

FROM customers

3

UNION

4

SELECT country

5

FROM suppliers

6

ORDER BY country ASC

Data Output

Messages

Notifications

≡+

▼

▼

	country character varying (15)
1	Argentina
2	Australia
3	Austria
4	Belgium
5	Brazil
6	Canada
7	Denmark
8	Finland

8. Выбрать такие страны, в которых "зарегистрированы" одновременно и заказчики и поставщики и работники.

```
1 SELECT country
2 FROM customers
3 INTERSECT
4 SELECT country
5 FROM suppliers
6 INTERSECT
7 SELECT country
8 FROM employees
```

Data Output		Messages	N
	country		
	character varying (15)		
1	UK		
2	USA		

9. Выбрать такие страны в которых "зарегистрированы" одновременно заказчики и поставщики, но при этом в них не "зарегистрированы" работники.

```
1 SELECT country
2 FROM customers
3 INTERSECT
4 SELECT country
5 FROM suppliers
6 EXCEPT
7 SELECT country
8 FROM employees
```

Data Output Messages Notifications



	country character varying (15) 🔒
1	Spain
2	Italy
3	Sweden
4	Norway
5	France
6	Brazil

## Лабораторная работа №3

### Задание

1. Найти заказчиков и обслуживающих их заказы сотрудников таких, что и заказчики, и сотрудники из города London, а доставка идёт компанией Speedy Express. Вывести компанию заказчика и ФИО сотрудника.

```
select customers.customer_id, employees.last_name, employees.first_name
from customers
INNER JOIN orders on customers.customer_id=orders.customer_id
INNER JOIN employees on employees.employee_id=orders.employee_id
where employees.city='London' and orders.ship_name = 'Vins et alcools Chevalier'
```

Output Messages Notifications

customer_id	last_name	first_name
VINET	Buchanan	Steven
VINET	Suyama	Michael

2.

- Найти активные (см. поле discontinued) продукты из категории Beverages и Seafood, которых в продаже менее 20 единиц. Вывести наименование продуктов, кол-во единиц в продаже, имя контакта поставщика и его телефонный номер.

Query Query History

```
1 select product_name, units_in_stock, suppliers.contact_name, suppliers.phone, discontinued
2 from products
3 INNER JOIN categories on products.category_id=categories.category_id
4 INNER JOIN suppliers on products.supplier_id=suppliers.supplier_id
5 where (categories.category_name='Beverages' or categories.category_name='Seafood')
6 and discontinued=1 and products.units_in_stock<20
7
8
```

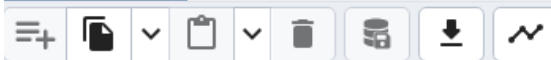
Data Output Messages Notifications

	product_name	units_in_stock	contact_name	phone	discontinued
	character varying (40)	smallint	character varying (30)	character varying (24)	integer
1	Chang	17	Charlotte Cooper	(171) 555-2222	1

3. Найти заказчиков, не сделавших ни одного заказа. Вывести имя заказчика и order\_id.

```
1 select company_name,orders.order_id
2 from customers
3 INNER JOIN orders on orders.customer_id=customers.customer_id
4 INNER JOIN order_details on order_details.order_id=orders.order_id
5 where order_details.discount=0
6
```

Data Output Messages Notifications



	company_name character varying (40)	order_id smallint
1	Vins et alcools Chevalier	10248
2	Vins et alcools Chevalier	10248
3	Vins et alcools Chevalier	10248
4	Toms Spezialitäten	10249
5	Toms Spezialitäten	10249
6	Hanari Carnes	10250

4. Переписать предыдущий запрос, используя симметричный вид джойна (подсказка: речь о LEFT и RIGHT).

```

1  select company_name,orders.order_id
2  from customers
3 left JOIN orders on orders.customer_id=customers.customer_id
4 right JOIN order_details on order_details.order_id=orders.order_id
5 where order_details.discount=0
6

```

Data Output Messages Notifications



	company_name character varying (40)	order_id smallint
1	Vins et alcools Chevalier	10248
2	Vins et alcools Chevalier	10248
3	Vins et alcools Chevalier	10248
4	Toms Spezialitäten	10249
5	Toms Spezialitäten	10249
6	Hanari Carnes	10250
7	Victuailles en stock	10251
8	Suprêmes délices	10252



## Лабораторная работа №4

### Задание

1. Вывести продукты количество которых в продаже меньше самого малого среднего количества продуктов в деталях заказов (группировка по product\_id). Результирующая таблица должна иметь колонки product\_name и units\_in\_stock.

```
6 select product_name, units_in_stock
7 from products
8 where units_in_stock < (select avg(quantity)
9                        from order_details
10                       group by product_id
11                       order by avg(quantity) desc
12                       limit 1)
13
14
```

Data Output Messages Notifications

	product_name character varying (40)	units_in_stock smallint
	Chai	39
!	Chang	17
!	Aniseed Syrup	13
!	Chef Anton's Gumbo Mix	0
!	Uncle Bob's Organic Dried Pea...	15
!	Northwoods Cranberry Sauce	6
!	Mishi Kobe Niku	29
!	Ikura	31

2. Напишите запрос, который выводит общую сумму фрахтов заказов для компаний-заказчиков для заказов, стоимость фрахта которых больше или равна средней величине стоимости фрахта всех заказов, а также дата отгрузки заказа должна находиться во второй половине июля 1996 года. Результирующая таблица должна иметь колонки customer\_id и freight\_sum, строки которой должны быть отсортированы по сумме фрахтов заказов.

Query Query History

```
1 SELECT customer_id, SUM(freight)
2 FROM orders
3 WHERE EXISTS (SELECT SUM(freight)
4               FROM orders
5               WHERE order_date BETWEEN '1996-07-15' AND '1996-07-30'
6               GROUP BY customer_id
7               HAVING sum(freight) >= AVG(freight))
8 group by customer_id
9
```

Data Output Messages Notifications



	customer_id character	sum real
1	TOMSP	125.97
2	LONEP	79.86
3	OLDWO	983.52997
4	WARTH	822.48004
5	MAGAA	469.75
6	QUEEN	1982.7
7	VINFT	58.410004

3. Напишите запрос, который выводит 3 заказа с наибольшей стоимостью, которые были созданы после 1 сентября 1997 года включительно и были доставлены в страны Южной Америки. Общая стоимость рассчитывается как сумма стоимости деталей заказа с учетом дисконта. Результирующая таблица должна иметь колонки customer\_id, ship\_country и order\_price, строки которой должны быть отсортированы по стоимости заказа в обратном порядке.

QueryQuery History

```

1 SELECT customer_id, ship_country, (order_details.unit_price * order_details.quantity * (1 - order_details.discount)) as order_price
2 FROM orders
3 INNER JOIN order_details ON orders.order_id = order_details.order_id
4 WHERE EXISTS (SELECT (unit_price * quantity * (1 - discount)) as order_price
5               FROM order_details
6               ORDER BY order_price DESC)
7               AND required_date > '1997-09-01'
8               AND ship_country IN ('Argentina', 'Spain', 'Venezuela', 'Brazil')
9
10
11

```

Data OutputMessagesNotifications

	customer_id character	ship_country character varying (15)	order_price double precision
1	RICAR	Brazil	380
2	RICAR	Brazil	179.99999932944775
3	GODOS	Spain	2475.800018310547
4	GODOS	Spain	299.25
5	QUEEN	Brazil	210
6	QUEEN	Brazil	385.93749969732016
7	QUEEN	Brazil	2165.9999983012676
8	LINOD	Venezuela	190
9	LINOD	Venezuela	442.0499839782715

Total rows: 245 of 245Query complete 00:00:00.123

Ножницы

Снимок экрана сохранен

Нажмите здесь, чтобы загрузить изображение

- Вывести все товары (уникальные названия продуктов), которых заказано ровно 10 единиц (конечно же, это можно решить и без подзапроса).

Query    Query History

```
1 select product_name
2 from products
3 where units_on_order=10
4
```

Data Output    Messages    Notifications



	product_name character varying (40)
1	Ipoh Coffee
2	Gnocchi di nonna Alice
3	Scottish Longbreads
4	Outback Lager

## Лабораторная работа №5

1. Создать таблицу teacher с полями teacher\_id serial, first\_name varchar, last\_name varchar, birthday date, phone varchar, title varchar

The screenshot shows a database management interface with a 'Query' tab selected. The query editor contains the following SQL code:

```
1 CREATE TABLE teacher
2 (
3     teacher_id serial,
4     first_name varchar,
5     last_name varchar,
6     birthday date,
7     phone varchar,
8     title varchar
9
10 );
11 select *
12 from teacher
13
```

Below the query editor, the 'Data Output' tab is active, displaying the table structure for 'teacher':

	teacher_id	first_name	last_name	birthday	phone	title
	integer	character varying	character varying	date	character varying	character varying

2. Добавить в таблицу после создания колонку middle\_name varchar

Query

Query History

Scratch Pad

```

1 ALTER TABLE teacher
2 ADD COLUMN middle_name varchar;
3
4 select *
5 from teacher
6

```

Data Output

Messages

Notifications

+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

teacher_id	first_name	last_name	birthday	phone	title	middle_name
integer	character varying	character varying	date	character varying	character varying	character varying

### 3. Удалить колонку middle\_name

Query

Query History

Scratch Pad

```

1 ALTER TABLE teacher
2 DROP COLUMN middle_name ;
3
4 select *
5 from teacher
6

```

Data Output

Messages

Notifications

+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

teacher_id	first_name	last_name	birthday	phone	title
integer	character varying	character varying	date	character varying	character varying

### 4. Переименовать колонку birthday в birth\_date

Query

Query History

Scratch Pad

```

1 ALTER TABLE teacher
2 rename COLUMN birthday to birth_date ;
3
4 select *
5 from teacher
6

```

Data Output

Messages

Notifications

teacher_id	first_name	last_name	birth_date	phone	title
integer	character varying	character varying	date	character varying	character varying

## 5. Изменить тип данных колонки phone на varchar(32)

```

1 ALTER TABLE teacher
2 ALTER COLUMN phone SET DATA TYPE varchar(32);
3
4 select *
5 from teacher
6

```

Data Output

Messages

Notifications

teacher_id	first_name	last_name	birth_date	phone	title
integer	character varying	character varying	date	character varying (32)	character varying

## 6. Создать таблицу exam с полями exam\_id serial, exam\_name varchar(256), exam\_date date

Query

Query History

1

2

3

4

5

6

7

8

9

10

create TABLE exam

(

exam\_id serial,

exam\_name varchar(256),

exam\_date date

);

select \*

from exam

Data Output

Messages

Notifications

≡+

▼

▼

exam\_id

integer

exam\_name

character varying (256)

exam\_date

date

7. Вставить три любых записи с автогенерацией идентификатора



Query

Query History

1

2

3

4

5

6

7

8

9

10

11

INSERT INTO exam (exam\_name)

VALUES

('exam1'),

('exam2'),

('exam3');

SELECT \* FROM exam;

Data Output

Messages

Notifications

+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈










	exam_id integer	exam_name character varying (256)	exam_date date
1	1	exam1	[null]
2	2	exam2	[null]
3	3	exam3	[null]

8. Посредством полной выборки убедиться, что данные были вставлены нормально и идентификаторы были сгенерированы с инкрементом
9. Удалить все данные из таблицы со сбросом идентификатор в исходное состояние

Query Query History

```
1 INSERT INTO exam (exam_name)
2 VALUES
3 ('exam1'),
4 ('exam2'),
5 ('exam3');
6 TRUNCATE TABLE exam RESTART IDENTITY;
7 SELECT * FROM exam;
8
9
10
11
```

Data Output Messages Notifications

								
	exam_id integer		exam_name character varying (256)		exam_date date			

1. Создать таблицу exam с полями:










- идентификатора экзамена - автоинкрементируемый, уникальный, запрещает NULL;- наименования экзамена- даты экзамена

```

1
2 CREATE TABLE exam (
3     exam_id SERIAL UNIQUE NOT NULL,
4     exam_name VARCHAR(255) NOT NULL,
5     exam_date DATE
6 );
7 select *
8 from exam

```

Data Output Messages Notifications

								
	exam_id integer		exam_name character varying (255)		exam_date date			










## 1. Удалить ограничение уникальности с поля идентификатора

```

1
2 ALTER TABLE exam
3 DROP CONSTRAINT exam_exam_id_key;
4 select *
5 from exam

```

Data Output Messages Notifications

								
	exam_id integer		exam_name character varying (255)		exam_date date			

### 3. Добавить ограничение первичного ключа на поле идентификатора

Query Query History

1

ALTER TABLE exam

2

ADD CONSTRAINT exam\_pk PRIMARY KEY (exam\_id);










3

select \*

4

from exam

Data Output Messages Notifications



exam_id	exam_name	exam_date
[PK] integer	character varying (255)	date

### 4. Создать таблицу person с полями










- идентификатора личности (простой int, первичный ключ)- имя- фамилия

Query Query History

1

CREATE TABLE person (  
2     person\_id INT PRIMARY KEY,  
3     first\_name VARCHAR(255),  
4     last\_name VARCHAR(255)  
5 );  
6 select \*  
7 from person

Data Output Messages Notifications



person_id [PK] integer	first_name character varying (255)	last_name character varying (255)
---------------------------	---------------------------------------	--------------------------------------

5. Создать таблицу паспорта с полями:

- идентификатора паспорта (простой int, первичный ключ)- серийный номер (простой int, запрещает NULL)- регистрация- ссылка на идентификатор личности (внешний ключ)

Query

Query History

Scratch Pad

```

1 CREATE TABLE passports (
2     passport_id INT PRIMARY KEY,
3     serial_number INT NOT NULL,
4     registration VARCHAR(255),
5     person_id INT,
6     FOREIGN KEY (person_id) REFERENCES person (pers
7 );
8 select *
9 from passports

```

Data Output

Messages

Notifications

≡+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

passport_id [PK] integer	serial_number integer	registration character varying (255)	person_id integer
-----------------------------	--------------------------	---	----------------------

6. Добавить колонку веса в таблицу book (создавали ранее) с ограничением, проверяющим вес (больше 0 но меньше 100)

```

1
2 ALTER TABLE book
3 ADD COLUMN weight DECIMAL CHECK (weight > 0 AND weight < 100);
4 select *
5 from book

```

Data Output Messages Notifications

	book_id [PK] integer	title text	isbn character varying (32)	year text	weight numeric
1	1	The Diary Young girl	015475345	[null]	[null]
2	2	Pride	0874532856	[null]	[null]
3	3	The Cider House Rules	7854753821	[null]	[null]
4	4	The Catcher in the Rye	0853853634	[null]	[null]
5	5	War and piece	0154753643	[null]	[null]

7. Убедиться в том, что ограничение на вес работает (попробуйте вставить невалидное значение)

8. Создать таблицу student с полями:


- идентификатора (автоинкремент)- полное имя- курс (по умолчанию 1)


```


1
2 CREATE TABLE student (
3     student_id SERIAL PRIMARY KEY,
4     full_name VARCHAR(255),
5     course INT DEFAULT 1
6 );
7 select *
8 from student


```


Data Output Messages Notifications























	<b>student_id</b> [PK] integer	<b>full_name</b> character varying (255)	<b>course</b> integer	
--	-----------------------------------	---	--------------------------	--

9. Вставить запись в таблицу студентов и убедиться, что ограничение на вставку значения по умолчанию работает

```

1 INSERT INTO student (full_name)
2 VALUES ('John Doe');
3 SELECT *
4 FROM student;

```

Data Output Messages Notifications

	<div>student_id</div> <div>[PK] integer</div>	<div>full_name</div> <div>character varying (255)</div>	<div>course</div> <div>integer</div>
1	1	John Doe	1



## 10. Удалить ограничение "по умолчанию" из таблицы студентов

Query

Query History

1

ALTER TABLE student

2

ALTER COLUMN course DROP DEFAULT;

3

SELECT \*

4

FROM student

Data Output

Messages

Notifications

	student_id [PK] integer	full_name character varying (255)	course integer
1	1	John Doe	1

## 11. Подключиться к БД northwind и добавить ограничение на поле unit\_price таблицы products (цена должна быть больше 0)

Query Query History

```
1 ALTER TABLE products
2 ADD CONSTRAINT unit_price_check CHECK (unit_price > 0);
3 SELECT unit_price
4 FROM products
5 |
```

Data Output Messages Notifications



	unit_price real	
1	18	
2	19	
3	10	
4	22	
5	21.35	
6	25	
7	30	
8	40	
9	27	

12. "Навесить" автоинкрементируемый счётчик на поле product\_id таблицы products (БД northwind). Счётчик должен начинаться с числа следующего за максимальным значением по этому столбцу.

```

1
2
3 CREATE SEQUENCE products_product_id_seq
4 START WITH 78;
5 select *
6 FROM products|

```

Data Output Messages Notifications

	product_id [PK] smallint	product_name character varying (40)	supplier_id smallint	category_id smallint	quantity_per_unit character varying (20)
1	1	Chai	8	1	10 boxes x 30 bags
2	2	Chang	1	1	24 - 12 oz bottles
3	3	Aniseed Syrup	1	2	12 - 550 ml bottles
4	4	Chef Anton's Cajun Seasoning	2	2	48 - 6 oz jars
5	5	Chef Anton's Gumbo Mix	2	2	36 boxes
6	6	Grandma's Boysenberry Spread	3	2	12 - 8 oz jars
7	7	Uncle Bob's Organic Dried Pears	3	7	12 - 1 lb pkgs.
8	8	Northwoods Cranberry Sauce	3	2	12 - 12 oz jars
9	9	Mishi Koko Miso	4	6	18 - 500 g pkgs

13. Произвести вставку в products (не вставляя идентификатор явно) и убедиться, что автоинкремент работает. Вставку сделать так, чтобы в результате команды вернулось значение, сгенерированное в качестве идентификатора.

**Вывод:** Postgre SQL — это мощная и надежная система управления базами данных, которая предлагает широкий набор возможностей для организации и управления данными. Она обладает гибкой архитектурой, позволяющей эффективно работать с различными типами данных, такими как числа, строки, даты, временные интервалы и многое другое. При работе с Postgre SQL мы изучили основные операции и команды для создания таблиц, добавления, изменения и удаления данных, а также выборки данных из базы. Postgre SQL предоставляет ряд расширений и функций, позволяющих расширить его возможности и упростить работу с данными. В процессе работы с Postgre SQL мы также рассмотрели принципы оптимизации запросов и создания индексов для улучшения производительности базы данных.

## **Список литературы**

PostgreSQL Documentation. (<https://www.postgresql.org/docs/>) - официальная документация PostgreSQL, содержит подробное описание возможностей и функций системы.