

**Министерство образования и науки РФ  
Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Норильский государственный индустриальный институт»  
Кафедра Информационных систем и технологий**

Отчет: Лабораторная работа №4

По предмету: Базы данных

Вариант 1

**Выполнил:**

*Шикан А.В.*

**Группы:**

*ИС-21*

**Проверил:**

*Беляев И.С.*

Норильск 2023г

Цель работы: Получение практических навыков работы с СУБД и языком SQL (оператор SELECT).

Задание: разработать запросы к базе данных, созданной и заполненной на предыдущих лабораторных работах, следующих видов:

- a. запрос с условием на числовые данные (>, <, =, between);
- b. запрос с условием на текстовые данные (LIKE, IN);
- c. запрос с вычисляемым полем;
- d. запрос к нескольким таблицам (без явного указания JOIN);
- e. запрос с агрегирующей функцией (AVG, SUM, COUNT, MIN, MAX);
- f. запрос с группировкой (GROUP BY);
- g. запрос с сортировкой (ORDER BY);
- h. запрос с вложенным подзапросом (не менее 3 видов);
- i. запрос с оператором UNION;
- j. запрос с оператором INTERSECT;
- k. запрос с оператором EXCEPT;
- l. запрос с выражением CASE;
- m. запрос с оператором JOIN (пять видов);
- n. иерархический запрос.

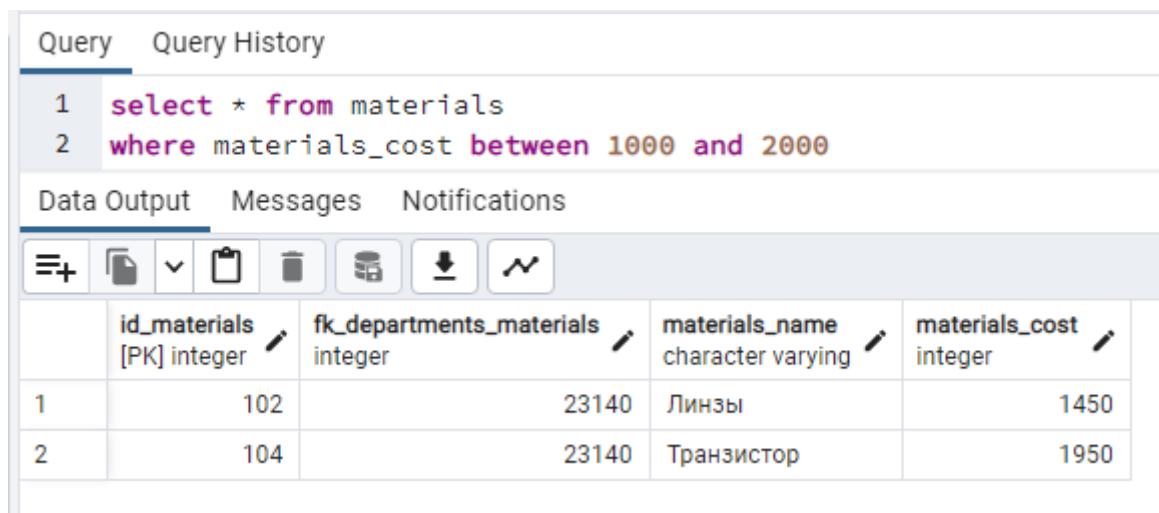
## Скриншоты

а) Запрос с условием на числовые данные *between* выводит данные о материалах стоимостью от 1000 до 2000:

```
select * from materials
```

```
where materials_cost between 1000 and 2000
```

Скриншот № 1(a) – запрос с условием на числовые данные *between*



The screenshot shows a database query interface with tabs for 'Query', 'Query History', 'Data Output', 'Messages', and 'Notifications'. The 'Query' tab is active, displaying a SQL query. Below the query, the 'Data Output' tab shows the results of the query in a table format. The table has five columns: 'id\_materials' (integer, PK), 'fk\_departments\_materials' (integer), 'materials\_name' (character varying), and 'materials\_cost' (integer). The results show two rows of data.

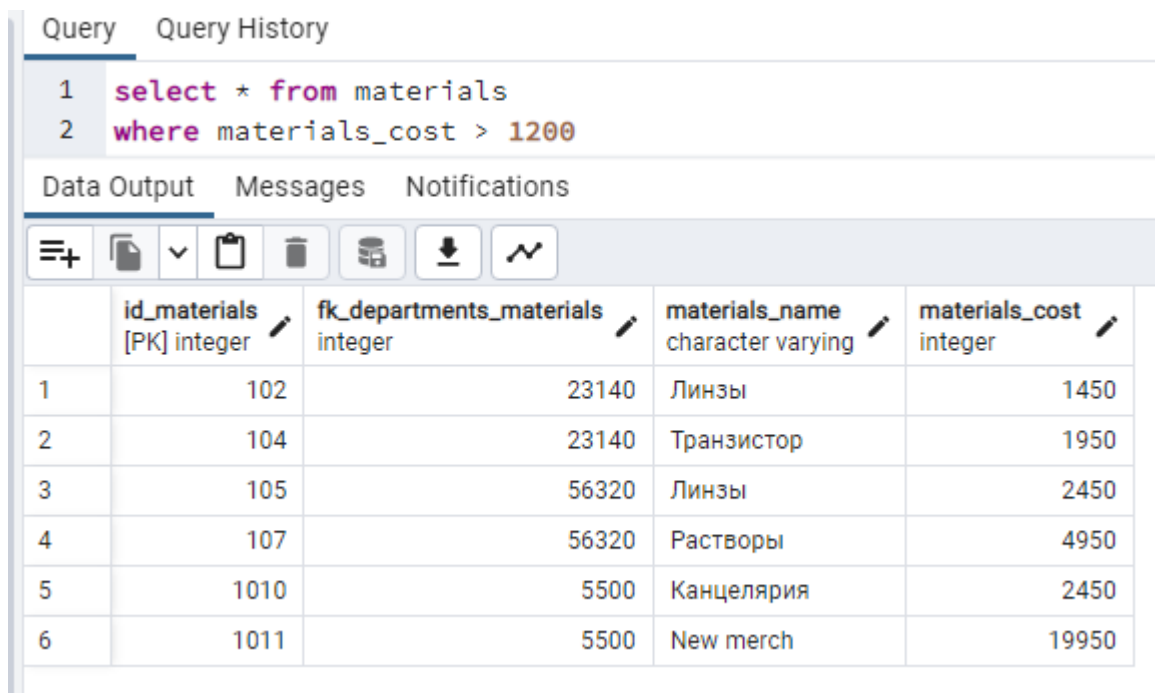
	id_materials [PK] integer	fk_departments_materials integer	materials_name character varying	materials_cost integer
1	102	23140	Линзы	1450
2	104	23140	Транзистор	1950

Запрос с условием на числовые данные “>” выводит данные о материалах стоимостью больше 1200:

```
select * from materials
```

```
where materials_cost > 1200
```

Скриншот № 2 (a) – запрос с условием на числовые данные “>”



The screenshot shows a database query interface with tabs for 'Query', 'Query History', 'Data Output', 'Messages', and 'Notifications'. The 'Query' tab is active, displaying a SQL query. Below the query, the 'Data Output' tab shows the results of the query in a table format. The table has five columns: 'id\_materials' (integer, PK), 'fk\_departments\_materials' (integer), 'materials\_name' (character varying), and 'materials\_cost' (integer). The results show six rows of data.

	id_materials [PK] integer	fk_departments_materials integer	materials_name character varying	materials_cost integer
1	102	23140	Линзы	1450
2	104	23140	Транзистор	1950
3	105	56320	Линзы	2450
4	107	56320	Растворы	4950
5	1010	5500	Канцелярия	2450
6	1011	5500	New merch	19950

- b) Запрос с условием на текстовые данные (LIKE, IN) выводит данные о материалах, которые начинаются на “Ли”:

```
select * from materials
```

```
where materials_name like 'Ли%'
```

Скриншот № 3 (b) – запрос с условием на текстовые данные *like*

The screenshot shows a database query interface. At the top, there are tabs for 'Query' and 'Query History'. The 'Query' tab is active, displaying a SQL query:

```
1 select * from materials
2 where materials_name like 'Ли%'
```

Below the query, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table of results. Above the table is a toolbar with icons for various actions like expand, copy, paste, delete, and download.

	id_materials [PK] integer	fk_departments_materials integer	materials_name character varying	materials_cost integer
1	102	23140	Линзы	1450
2	105	56320	Линзы	2450

- с) запрос с вычисляемым полем выводит данные об общей сумме материалов со скидкой в 2700:

```
select sum(materials_cost-2700) from materials
```

Скриншот № 4 (с) – запрос с вычисляемым полем

The screenshot shows a database query interface. At the top, there are tabs for 'Query' and 'Query History'. The 'Query' tab is active, displaying a SQL query:

```
1 select sum(materials_cost-2700) from materials
2 |
```

Below the query, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table of results. Above the table is a toolbar with icons for various actions like expand, copy, paste, delete, and download.

	sum bigint
1	10000

- d) Запрос к нескольким таблицам (без явного указания JOIN) выводит данные о сотрудниках и их должностях вместе в одной таблице:

```
select * from staff, post
```

## Скриншот № 5 (d) – запрос к нескольким таблицам

Query Query History

```
1 select * from staff, post
2
```

Data Output Messages Notifications

	id_staff integer	f_name character varying	s_name character varying	b_day date	id_post integer	fk_staff_post integer	fk_departments_post integer	post_name character varying	post_salary integer
1	1	Алексей	Шикан	2003-07-06	1	1	5500	Главный специалист	100000
2	2	Антон	Посметюк	2000-10-07	1	1	5500	Главный специалист	100000
3	3	Александр	Цимбалюк	2003-06-18	1	1	5500	Главный специалист	100000
4	4	Юлия	Лахмостова	2001-12-01	1	1	5500	Главный специалист	100000
5	5	Анна	Усс	2002-04-22	1	1	5500	Главный специалист	100000
6	1	Алексей	Шикан	2003-07-06	2	2	56320	Химик	50000
7	2	Антон	Посметюк	2000-10-07	2	2	56320	Химик	50000
8	3	Александр	Цимбалюк	2003-06-18	2	2	56320	Химик	50000
9	4	Юлия	Лахмостова	2001-12-01	2	2	56320	Химик	50000
10	5	Анна	Усс	2002-04-22	2	2	56320	Химик	50000
11	1	Алексей	Шикан	2003-07-06	3	3	5500	Главный инженер	55000
12	2	Антон	Посметюк	2000-10-07	3	3	5500	Главный инженер	55000
13	3	Александр	Цимбалюк	2003-06-18	3	3	5500	Главный инженер	55000
14	4	Юлия	Лахмостова	2001-12-01	3	3	5500	Главный инженер	55000
15	5	Анна	Усс	2002-04-22	3	3	5500	Главный инженер	55000
16	1	Алексей	Шикан	2003-07-06	4	4	23140	Лаборант	35000
17	2	Антон	Посметюк	2000-10-07	4	4	23140	Лаборант	35000
18	3	Александр	Цимбалюк	2003-06-18	4	4	23140	Лаборант	35000
19	4	Юлия	Лахмостова	2001-12-01	4	4	23140	Лаборант	35000
20	5	Анна	Усс	2002-04-22	4	4	23140	Лаборант	35000

Total rows: 25 of 25 Query complete 00:00:00.078

е) Запрос с агрегирующей функцией AVG выводит среднее значение заработной платы среди сотрудников:

```
select AVG(post_salary) from post
```

## Скриншот № 6 (е) – запрос с агрегирующей функцией AVG

Query Query History

```
1 select AVG(post_salary) from post
2
```

Data Output Messages Notifications

	avg numeric
1	61000.000000000000

ф) Запрос с группировкой GROUP BY выводит данные о материалах с одинаковым названием и их количестве:

```
select materials_name, count(*) from materials
group by materials_name
```

Скриншот № 7 (f) – запрос с группировкой GROUP BY

Query		Query History
1	<code>select materials_name,count(*) from materials</code>	
2	<code>group by materials_name</code>	
Data Output		Messages Notifications
<div> <div>≡+</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>🗑️</div> <div>🗄️</div> <div>⬇️</div> <div>📈</div> </div>		
	materials_name character varying 🔒	count bigint 🔒
1	Растворы	1
2	Транзистор	1
3	Проводка	1
4	Линзы	2
5	Канцелярия	1
6	Электромагниты	1
7	New merch	1
8	Колбы	1

g) Запрос с сортировкой ORDER BY выводит данные о материалах с одинаковым названием и их количестве в порядке возрастания:

*select materials\_name,count(\*) from materials*

*group by materials\_name*

*order by materials\_name asc*

Скриншот № 8 (g) – запрос с группировкой ORDER BY

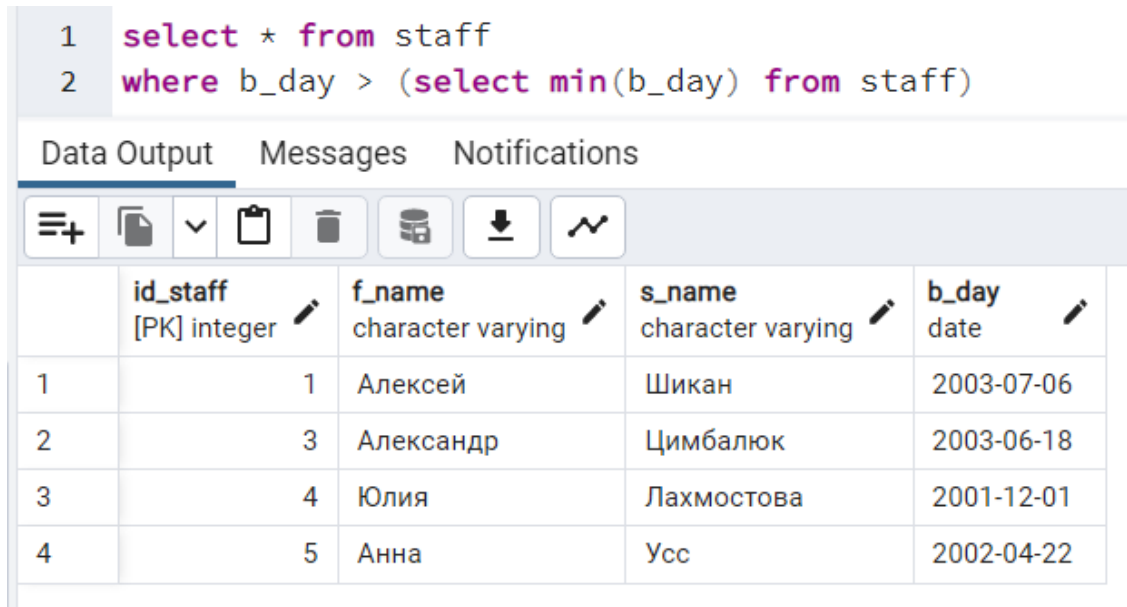
Query		Query History
1	<code>select materials_name,count(*) from materials</code>	
2	<code>group by materials_name</code>	
3	<code>order by materials_name asc</code>	
Data Output		Messages Notifications
<div> <div>≡+</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>🗑️</div> <div>🗄️</div> <div>⬇️</div> <div>📈</div> </div>		
	materials_name character varying 🔒	count bigint 🔒
1	New merch	1
2	Канцелярия	1
3	Колбы	1
4	Линзы	2
5	Проводка	1
6	Растворы	1
7	Транзистор	1
8	Электромагниты	1

h) Запрос с вложенным подзапросом (не менее 3 видов) выводит таблицу сотрудников без учета самого старшего по возрасту сотрудника:

```
select * from staff
```

```
where b_day > (select min(b_day) from staff)
```

Скриншот № 9 (h) – запрос с вложенным подзапросом



The screenshot shows a database query interface. At the top, there is a SQL query editor with two lines of code: `1 select * from staff` and `2 where b_day > (select min(b_day) from staff)`. Below the editor are three tabs: "Data Output", "Messages", and "Notifications". The "Data Output" tab is active, displaying a table with 5 columns: `id_staff` (integer, PK), `f_name` (character varying), `s_name` (character varying), and `b_day` (date). The table contains 4 rows of data.

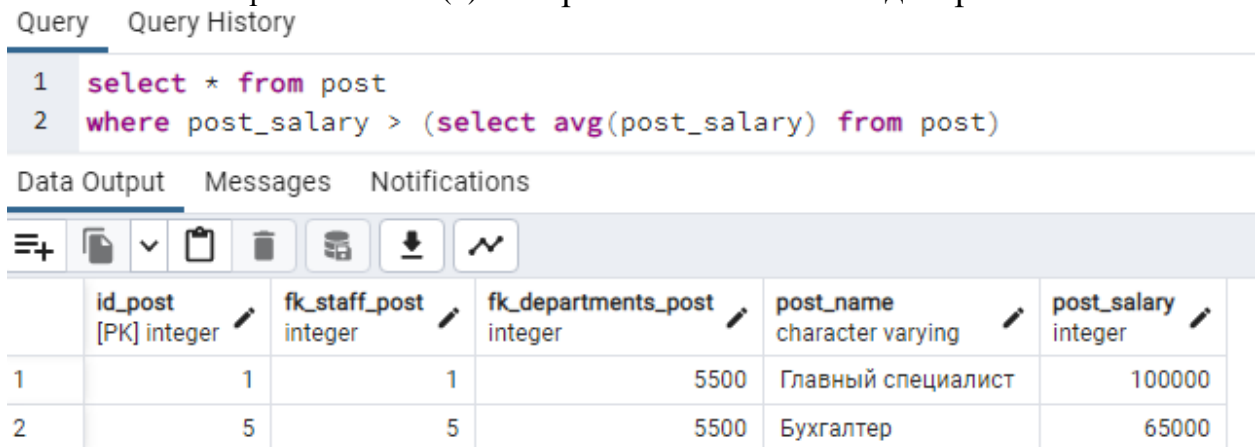
	<code>id_staff</code> [PK] integer	<code>f_name</code> character varying	<code>s_name</code> character varying	<code>b_day</code> date
1	1	Алексей	Шикан	2003-07-06
2	3	Александр	Цимбалюк	2003-06-18
3	4	Юлия	Лахмостова	2001-12-01
4	5	Анна	Усс	2002-04-22

Выводит таблицу должностей с заработной платой выше среднего значения:

```
select * from post
```

```
where post_salary > (select avg(post_salary) from post)
```

Скриншот № 10 (h) – запрос с вложенным подзапросом



The screenshot shows a database query interface. At the top, there is a SQL query editor with two lines of code: `1 select * from post` and `2 where post_salary > (select avg(post_salary) from post)`. Below the editor are three tabs: "Query", "Query History", and "Notifications". The "Query" tab is active, displaying a table with 6 columns: `id_post` (integer, PK), `fk_staff_post` (integer), `fk_departments_post` (integer), `post_name` (character varying), and `post_salary` (integer). The table contains 2 rows of data.

	<code>id_post</code> [PK] integer	<code>fk_staff_post</code> integer	<code>fk_departments_post</code> integer	<code>post_name</code> character varying	<code>post_salary</code> integer
1	1	1	5500	Главный специалист	100000
2	5	5	5500	Бухгалтер	65000

Выводит материал с самой маленькой стоимостью:

```
select * from materials
```

```
where materials_cost = (select min(materials_cost) from materials)
```

Скриншот № 11 (h) – запрос с вложенным подзапросом

Query

Query History

1

select \* from materials

2

where materials\_cost = (select min(materials\_cost) from materials)

Data Output

Messages

Notifications

≡+

▼

	id_materials [PK] integer	fk_departments_materials integer	materials_name character varying	materials_cost integer
1	101	23140	Электромагниты	50

- i) Запрос с оператором UNION выводит один большой список с названиями материалов и оборудования:

*select materials\_name from materials*

*union all select device\_name from device*

Скриншот № 12 (i) – запрос с вложенным подзапросом

Query

Query History

1

select materials\_name from materials

2

union all select device\_name from device

Data Output

Messages

Notifications

≡+

materials\_name

character varying

1

Линзы

2

Проводка

3

Транзистор

4

Линзы

5

Колбы

6

Растворы

7

Канцелярия

8

Электромагниты

9

New merch

10

Принтер-сканер

11

ПК

12

Счеты

13

Амперметр

14

Вольтметр

15

Баромет

16

ХимОбор1

17

ХимОбор2

- j) Запрос с оператором INTERSECT выводит имена сотрудников, которые есть и в таблице “staff”, и в таблице “proba”:

*select f\_name from proba*



*INTERSECT select f\_name from staff*

Скриншот № 13 (j) – запрос с оператором INTERSECT

Query		Query History
1	<code>select f_name from proba</code>	
2	<code>INTERSECT select f_name from staff;</code>	

Data Output		Messages	Notifications
f_name character varying			
1	Анна		
2	Антон		

k) Запрос с оператором EXCEPT выводит все имена из таблицы “proba”, которых нет в таблице “staff”:

*select f\_name from proba*

*EXCEPT select f\_name from staff*

Скриншот № 14 (k) – запрос с оператором EXCEPT

Query		Query History
1	<code>select f_name from proba</code>	
2	<code>EXCEPT select f_name from staff</code>	

Data Output		Messages	Notifications
f_name character varying			
1	Маня		
2	Алексий		
3	Таня		
4	Саня		
5	Галя		

l) Запрос с выражением CASE выводит уникальный код (набор символов вроде штрих-кода) для разработки:

*select develop\_name,*

```

case when develop_name = 'ФизРазработка1' then '#28f98wr4'
when develop_name = 'ФизРазработка2' then '#ed53bts52'
when develop_name = 'ХимРазработка1' then '#po54b322b'
when develop_name = 'ХимРазработка2' then '#333fbny6l'
end develop_barcode
from develop

```

Скриншот № 15 (l) – запрос с выражением CASE

Query

Query History

```
1 select develop_name,
2 case when develop_name = 'ФизРазработка1' then '#28f98wr4'
3 when develop_name = 'ФизРазработка2' then '#ed53bts52'
4 when develop_name = 'ХимРазработка1' then '#po54b322b'
5 when develop_name = 'ХимРазработка2' then '#333fbny6l'
6 end develop_barcode
7 from develop
```

Data Output

Messages

Notifications

≡+

▼

	develop_name character varying	develop_barcode text
1	ФизРазработка1	#28f98wr4
2	ФизРазработка2	#ed53bts52
3	ХимРазработка1	#po54b322b
4	ХимРазработка2	#333fbny6l

м) Запрос с оператором JOIN (пять видов):

а. JOIN – выполняет объединение таблиц отделы и разработки по id:

```
select * from departments join develop on id_departments = fk_departments_develop
```

Скриншот № 16 (м.а) – запрос с оператором JOIN

1

select \* from departments join develop on id\_departments = fk\_departments\_develop

Data Output

Messages

Notifications

≡+

▼

	id_departments integer	departments_name character varying	id_develop integer	fk_departments_develop integer	develop_name character varying
1	23140	Отдел Физики	991	23140	ФизРазработка1
2	23140	Отдел Физики	992	23140	ФизРазработка2
3	56320	Отдел Химии	993	56320	ХимРазработка1
4	56320	Отдел Химии	994	56320	ХимРазработка2

б. INNER JOIN - выполняет объединение таблицы отделов и материалов, относящихся к этим отделам по id:

*select departments\_name, id\_materials, materials\_name from departments*

*INNER JOIN materials on id\_departments = fk\_departments\_materials*

Скриншот № 17 (m.b) – запрос с оператором INNER JOIN

Query		Query History	
1	select	departments_name, id_materials, materials_name	from departments
2	INNER JOIN	materials on	id_departments = fk_departments_materials
Data Output			
Messages			
Notifications			
	departments_name	id_materials	materials_name
	character varying	integer	character varying
1	Отдел Физики	102	Линзы
2	Отдел Физики	103	Проводка
3	Отдел Физики	104	Транзистор
4	Отдел Химии	105	Линзы
5	Отдел Химии	106	Колбы
6	Отдел Химии	107	Растворы
7	Управление	1010	Канцелярия
8	Отдел Физики	101	Электромагниты
9	Управление	1011	New merch

с. AS – позволяет вывести итоговую цену за покупку материала и его доставку:

*SELECT materials\_name, SUM(materials\_price\_delivery + materials\_cost) AS total\_price*

*FROM materials*

*GROUP BY id\_materials*

*ORDER BY total\_price DESC*

Скриншот № 18 (m.c) – запрос с оператором AS

Query		Query History	
1	SELECT	materials_name, SUM(materials_price_delivery + materials_cost)	AS total_price
2	FROM	materials	
3	GROUP BY	id_materials	
4	ORDER BY	total_price	DESC
Data Output			
Messages			
Notifications			
	materials_name	total_price	
	character varying	bigint	
1	New merch	20616	
2	Растворы	6199	
3	Линзы	2950	
4	Канцелярия	2500	
5	Транзистор	2349	
6	Колбы	1859	
7	Линзы	1762	
8	Проводка	300	
9	Электромагниты	65	

d. NATURAL JOIN – выводит (объединение таблиц) имена работников и их должности:

```
SELECT id_staff, f_name, s_name, post_name
```

```
FROM staff
```

```
NATURAL JOIN post
```

Скриншот № 19 (m.d) – запрос с оператором NATURAL JOIN

Query

Query History

1SELECT id\_staff, f\_name, s\_name, post\_name

2FROM staff

3NATURAL JOIN post

Data Output

Messages

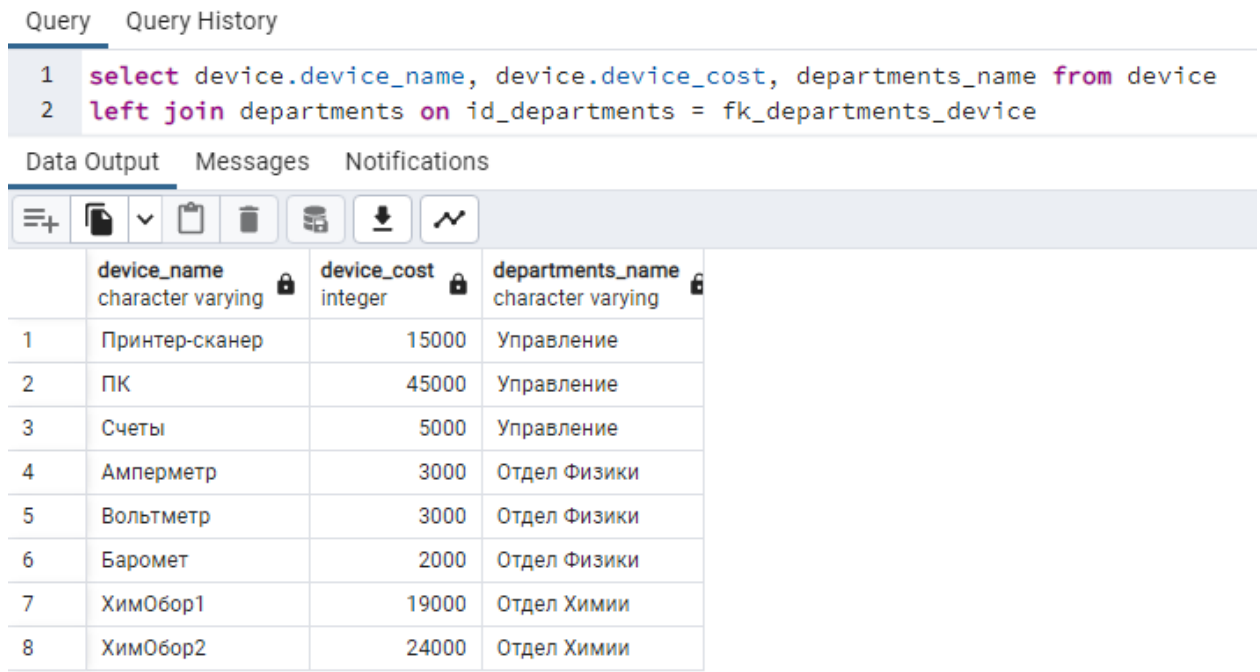
Notifications

	id_staff integer	f_name character varying	s_name character varying	post_name character varying
1	1	Алексей	Шикан	Главный специалист
2	2	Антон	Посметюк	Главный специалист
3	3	Александр	Цимбалюк	Главный специалист
4	4	Юлия	Лахмостова	Главный специалист
5	5	Анна	Усс	Главный специалист
6	1	Алексей	Шикан	Химик
7	2	Антон	Посметюк	Химик
8	3	Александр	Цимбалюк	Химик
9	4	Юлия	Лахмостова	Химик
10	5	Анна	Усс	Химик
11	1	Алексей	Шикан	Главный инженер
12	2	Антон	Посметюк	Главный инженер
13	3	Александр	Цимбалюк	Главный инженер
14	4	Юлия	Лахмостова	Главный инженер
15	5	Анна	Усс	Главный инженер
16	1	Алексей	Шикан	Лаборант
17	2	Антон	Посметюк	Лаборант
18	3	Александр	Цимбалюк	Лаборант
19	4	Юлия	Лахмостова	Лаборант
20	5	Анна	Усс	Лаборант
21	1	Алексей	Шикан	Бухгалтер
22	2	Антон	Посметюк	Бухгалтер
23	3	Александр	Цимбалюк	Бухгалтер
24	4	Юлия	Лахмостова	Бухгалтер
25	5	Анна	Усс	Бухгалтер

- е. LEFT JOIN - вывод объединенной таблицы отделов и приборов из ЭТИХ ОТДЕЛОВ:

```
select device.device_name, device.device_cost, departments_name from device  
left join departments on id_departments = fk_departments_device
```

Скриншот № 20 (m.e) – запрос с оператором LEFT JOIN



The screenshot shows a database query interface with a 'Query' tab selected. The query text is as follows:

```
1 select device.device_name, device.device_cost, departments_name from device  
2 left join departments on id_departments = fk_departments_device
```

Below the query, the 'Data Output' tab is active, displaying the results of the query in a table. The table has four columns: 'device\_name' (character varying), 'device\_cost' (integer), and 'departments\_name' (character varying). The results are as follows:

	device_name character varying	device_cost integer	departments_name character varying
1	Принтер-сканер	15000	Управление
2	ПК	45000	Управление
3	Счеты	5000	Управление
4	Амперметр	3000	Отдел Физики
5	Вольтметр	3000	Отдел Физики
6	Баромет	2000	Отдел Физики
7	ХимОбор1	19000	Отдел Химии
8	ХимОбор2	24000	Отдел Химии

- н) Иерархический запрос – выводит таблицу иерархии должностей в научной лаборатории:

```
with recursive post1 (id_id, parent_id, post_name, post_salary, path) as (  
select p1.id_id, p1.parent_id, p1.post_name, p1.post_salary, cast (p1.post_name as  
varchar(70)) as path  
from post p1 where p1.post_name = 'Лаборант'  
union  
select p2.id_id, p2.parent_id, p2.post_name, p2.post_salary, cast (post1.path ||  
p2.post_name as varchar(70))  
from post p2 join post1 on (post1.parent_id = p2.id_id))  
select * from post1
```

## Скриншот № 21(n) – иерархический запрос

Query Query History

```
1 with recursive post1 (id_id, parent_id, post_name, post_salary, path) as (  
2 select p1.id_id, p1.parent_id, p1.post_name, p1.post_salary, cast (p1.post_name as varchar(70)) as path  
3 from post p1 where p1.post_name = 'Лаборант'  
4 union  
5 select p2.id_id, p2.parent_id, p2.post_name, p2.post_salary, cast (post1.path || '->' || p2.post_name as varchar(70))  
6 from post p2 join post1 on (post1.parent_id = p2.id_id)  
7 select * from post1
```

Data Output Messages Notifications



	id_id integer	parent_id integer	post_name character varying	post_salary integer	path character varying (70)	
1		5	4	Лаборант	35000	Лаборант
2		4	3	Химик	50000	Лаборант->Химик
3		3	2	Главный инженер	55000	Лаборант->Химик->Главный инженер
4		2	1	Бухгалтер	65000	Лаборант->Химик->Главный инженер->Бухгалтер
5		1	[null]	Главный специалист	100000	Лаборант->Химик->Главный инженер->Бухгалтер->Главный специалист