ФГБОУ ВО

«Заполярный государственный университет им. Н.М. Федоровского»

Кафедра __ИСиТ__

Специальность __ИС-21__

ОТЧЕТ

о выполнении лабораторной работы

Выполнила:

Старикова А.А.

Дата:

« 05 » мая 2023 г.

Норильск, 2023

Лабораторная работа №4

Тема: SQL. Запросы.

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

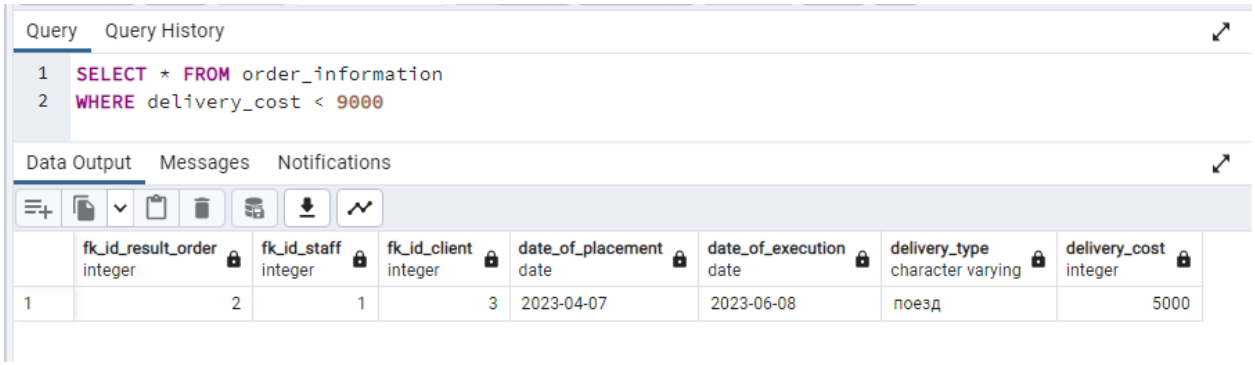
Задание: разработать запросы к базе данных, созданной и заполненной на предыдущих лабораторных работах, следующих видов:

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.

Для каждого запроса подписать, что именно он возвращает с учетом предметной области (запросы со смыслом, а не только синтаксически правильные операторы).

SELECT * FROM order_information

WHERE delivery_cost < 9000



Query Query History

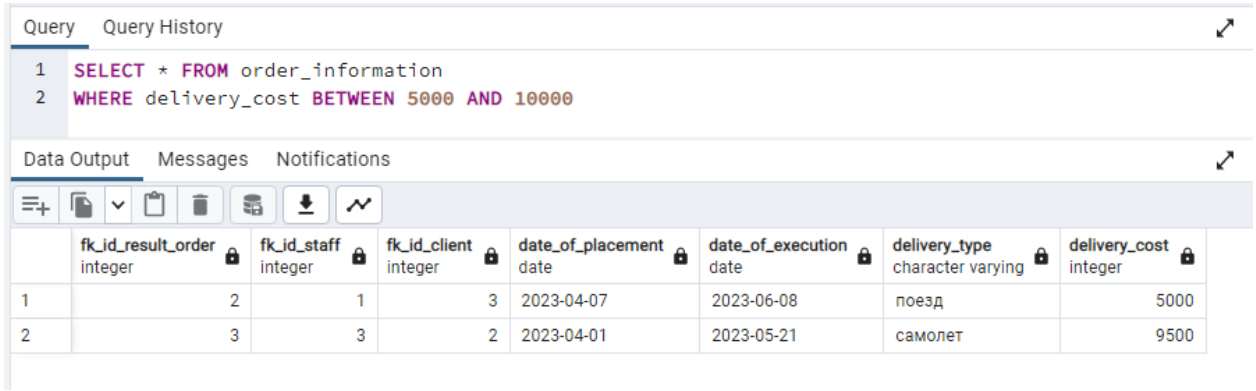
```
1  SELECT * FROM order_information
2  WHERE delivery_cost < 9000
```

Data Output Messages Notifications

	fk_id_result_order integer	fk_id_staff integer	fk_id_client integer	date_of_placement date	date_of_execution date	delivery_type character varying	delivery_cost integer
1	2	1	3	2023-04-07	2023-06-08	поезд	5000

SELECT * FROM order_information

WHERE delivery_cost BETWEEN 5000 AND 10000



Query Query History

```
1  SELECT * FROM order_information
2  WHERE delivery_cost BETWEEN 5000 AND 10000
```

Data Output Messages Notifications

	fk_id_result_order integer	fk_id_staff integer	fk_id_client integer	date_of_placement date	date_of_execution date	delivery_type character varying	delivery_cost integer
1	2	1	3	2023-04-07	2023-06-08	поезд	5000
2	3	3	2	2023-04-01	2023-05-21	самолет	9500

SELECT * FROM staff

WHERE staff_adress LIKE '%Нью-Йорк%'

```
SELECT * FROM staff
WHERE staff_adress LIKE '%Нью-Йорк%'
```

Data Output | Messages | Notifications

	id_staff [PK] integer	staff_fio character varying (60)	post character varying	staff_adress character varying	staff_number character varying (11)	fk_id_for_delete integer
1	1	Сигурни Уивер	менеджер по продажам	Манхэттэн, Нью-Йорк, США	89089287283	[null]

SELECT * FROM provider

WHERE company_name IN ('Ужасающий','Молчание ягнят')

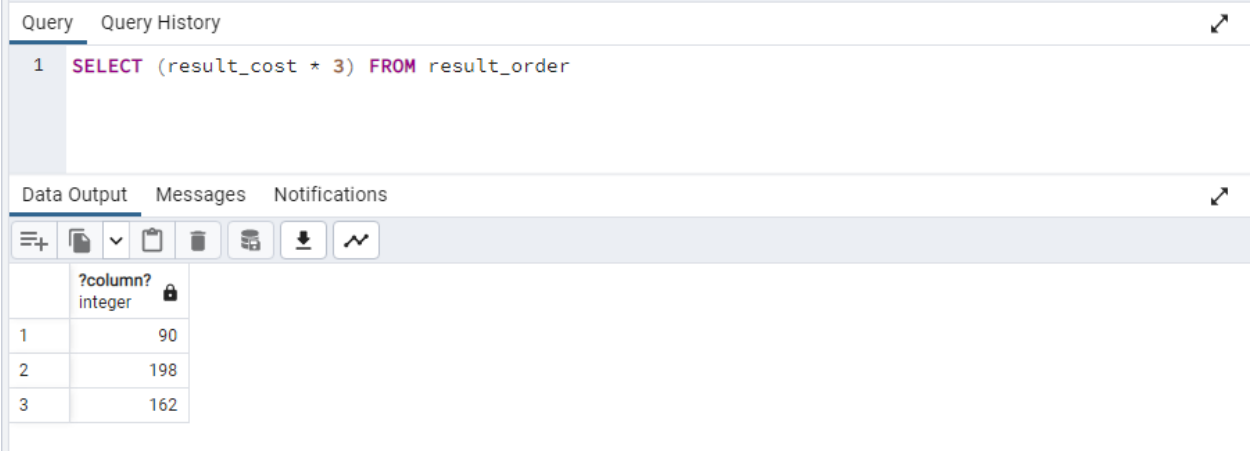
```
SELECT * FROM provider
WHERE company_name IN ('Ужасающий', 'Молчание ягнят')
```

Query | Query History

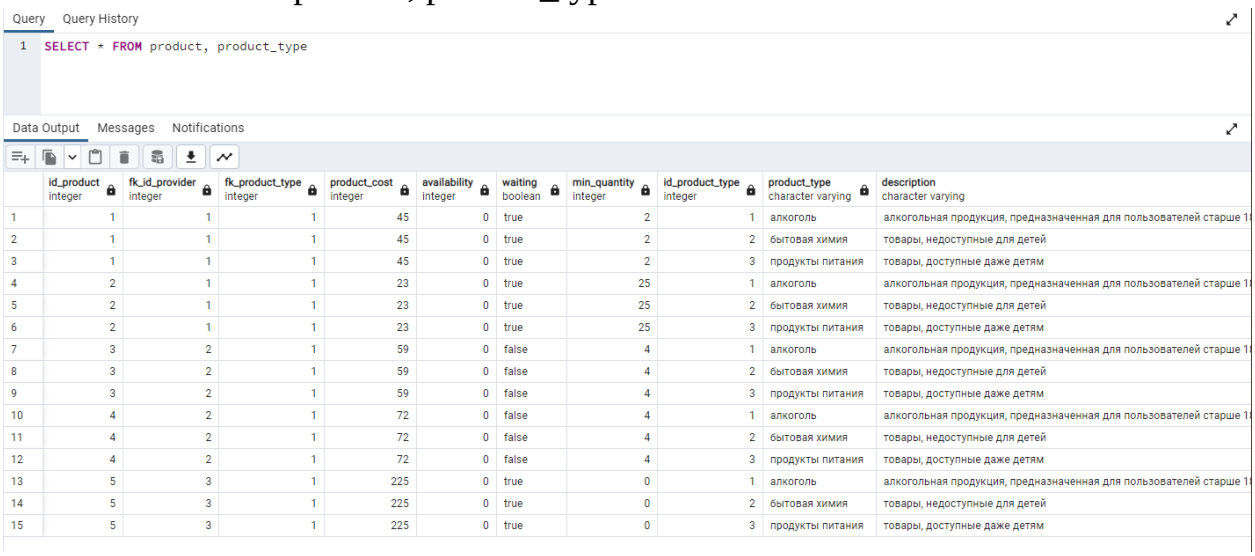
Data Output | Messages | Notifications

	id_provider [PK] integer	company_name character varying	provider_name character varying (60)	provider_adress character varying	provider_number character varying (11)
1	1	Ужасающий	Джоди Фостер	Лос-Анджелес, Калифорния, США	89059287733
2	3	Молчание ягнят	Брук Смит	Нью-Йорк, США	89135683321

SELECT (result_cost * 3) FROM result_order



SELECT * FROM product, product_type



Query | Query History

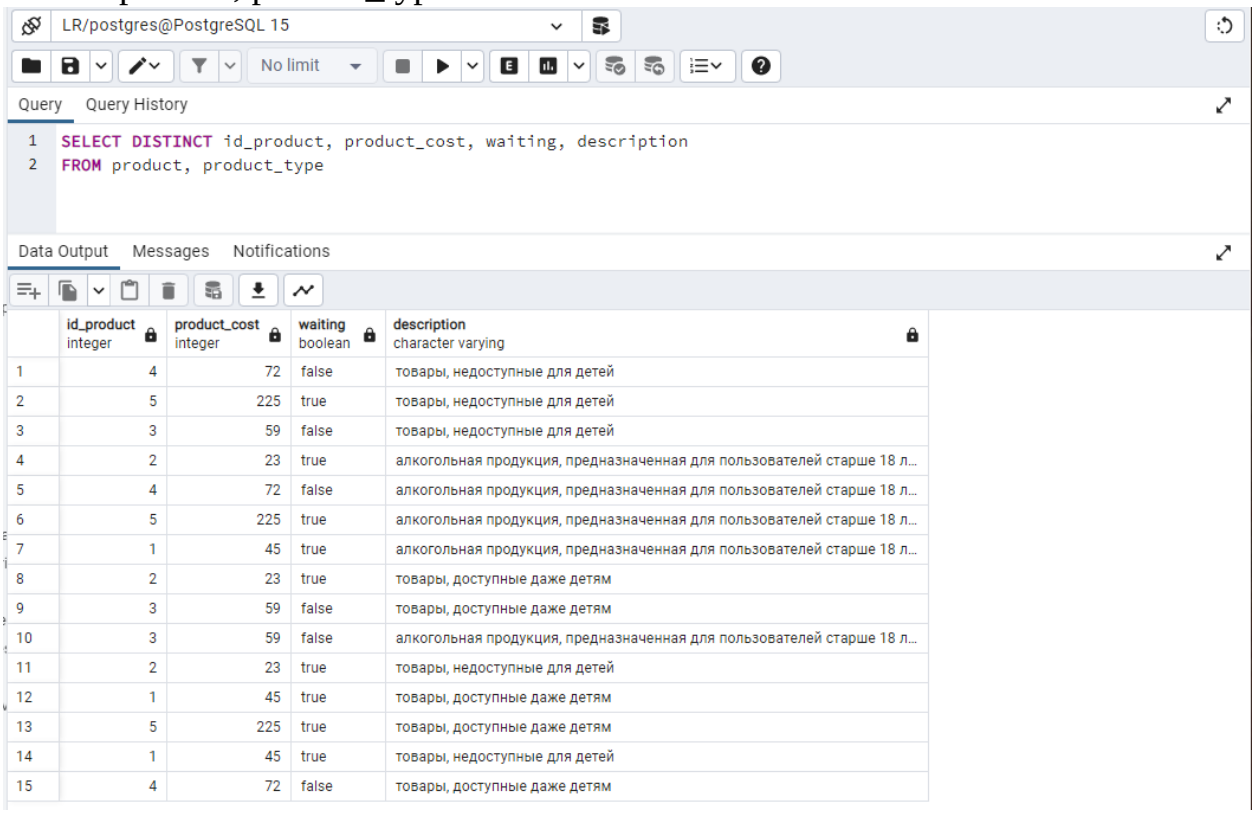
```
SELECT * FROM product, product_type
```

Data Output | Messages | Notifications

	id_product integer	fk_id_provider integer	fk_product_type integer	product_cost integer	availability integer	waiting boolean	min_quantity integer	id_product_type integer	product_type character varying	description character varying
1	1	1	1	45	0	true	2	1	алкоголь	алкогольная продукция, предназначенная для пользователей старше 1
2	1	1	1	45	0	true	2	2	бытовая химия	товары, недоступные для детей
3	1	1	1	45	0	true	2	3	продукты питания	товары, доступные даже детям
4	2	1	1	23	0	true	25	1	алкоголь	алкогольная продукция, предназначенная для пользователей старше 1
5	2	1	1	23	0	true	25	2	бытовая химия	товары, недоступные для детей
6	2	1	1	23	0	true	25	3	продукты питания	товары, доступные даже детям
7	3	2	1	59	0	false	4	1	алкоголь	алкогольная продукция, предназначенная для пользователей старше 1
8	3	2	1	59	0	false	4	2	бытовая химия	товары, недоступные для детей
9	3	2	1	59	0	false	4	3	продукты питания	товары, доступные даже детям
10	4	2	1	72	0	false	4	1	алкоголь	алкогольная продукция, предназначенная для пользователей старше 1
11	4	2	1	72	0	false	4	2	бытовая химия	товары, недоступные для детей
12	4	2	1	72	0	false	4	3	продукты питания	товары, доступные даже детям
13	5	3	1	225	0	true	0	1	алкоголь	алкогольная продукция, предназначенная для пользователей старше 1
14	5	3	1	225	0	true	0	2	бытовая химия	товары, недоступные для детей
15	5	3	1	225	0	true	0	3	продукты питания	товары, доступные даже детям

SELECT DISTINCT id_product, product_cost, waiting, description

FROM product, product_type



LR/postgres@PostgreSQL 15

No limit

Query | Query History

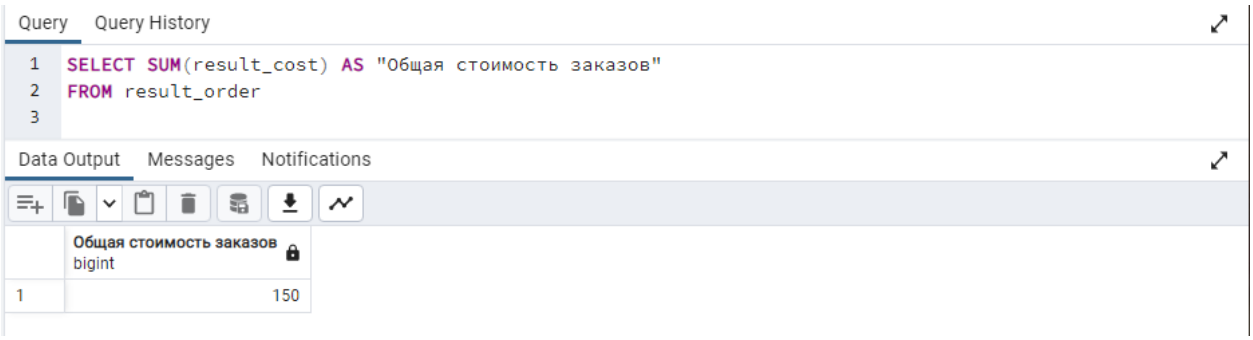
```
SELECT DISTINCT id_product, product_cost, waiting, description
FROM product, product_type
```

Data Output | Messages | Notifications

	id_product integer	product_cost integer	waiting boolean	description character varying
1	4	72	false	товары, недоступные для детей
2	5	225	true	товары, недоступные для детей
3	3	59	false	товары, недоступные для детей
4	2	23	true	алкогольная продукция, предназначенная для пользователей старше 18 л...
5	4	72	false	алкогольная продукция, предназначенная для пользователей старше 18 л...
6	5	225	true	алкогольная продукция, предназначенная для пользователей старше 18 л...
7	1	45	true	алкогольная продукция, предназначенная для пользователей старше 18 л...
8	2	23	true	товары, доступные даже детям
9	3	59	false	товары, доступные даже детям
10	3	59	false	алкогольная продукция, предназначенная для пользователей старше 18 л...
11	2	23	true	товары, недоступные для детей
12	1	45	true	товары, доступные даже детям
13	5	225	true	товары, доступные даже детям
14	1	45	true	товары, недоступные для детей
15	4	72	false	товары, доступные даже детям

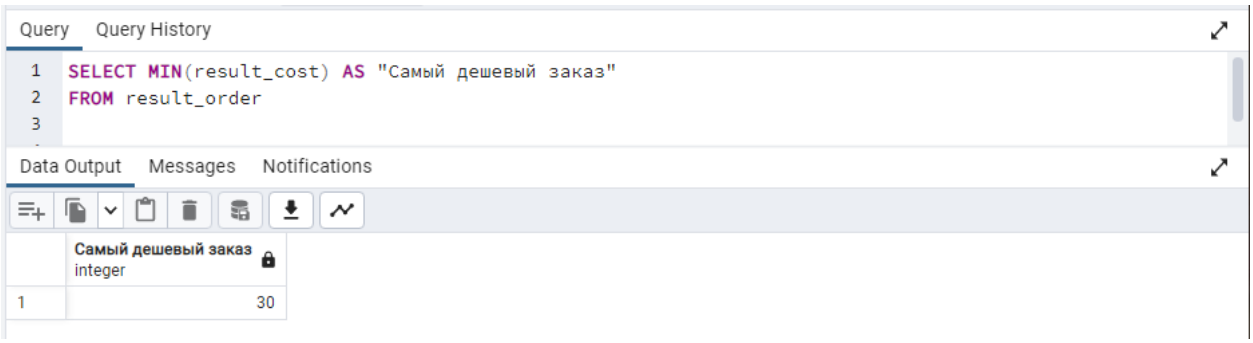
SELECT SUM(result_cost) AS "Общая стоимость заказов"

FROM result_order



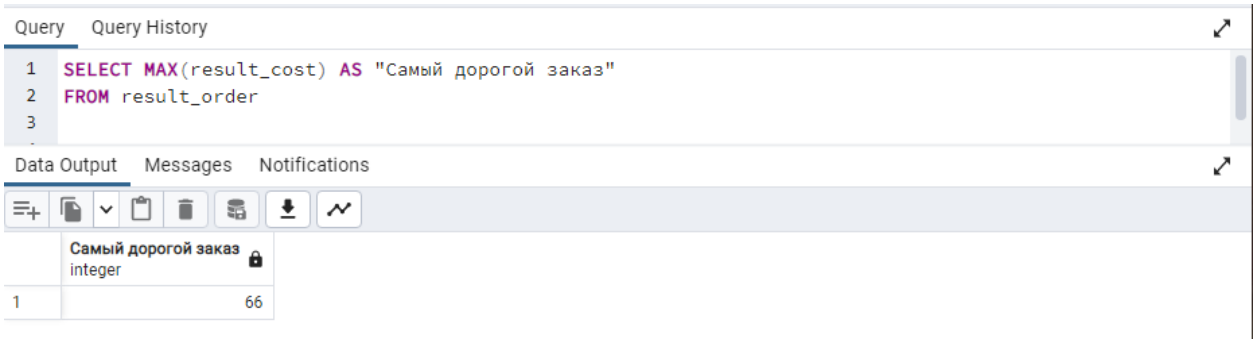
SELECT MIN(result_cost) AS "Самый дешевый заказ"

FROM result_order



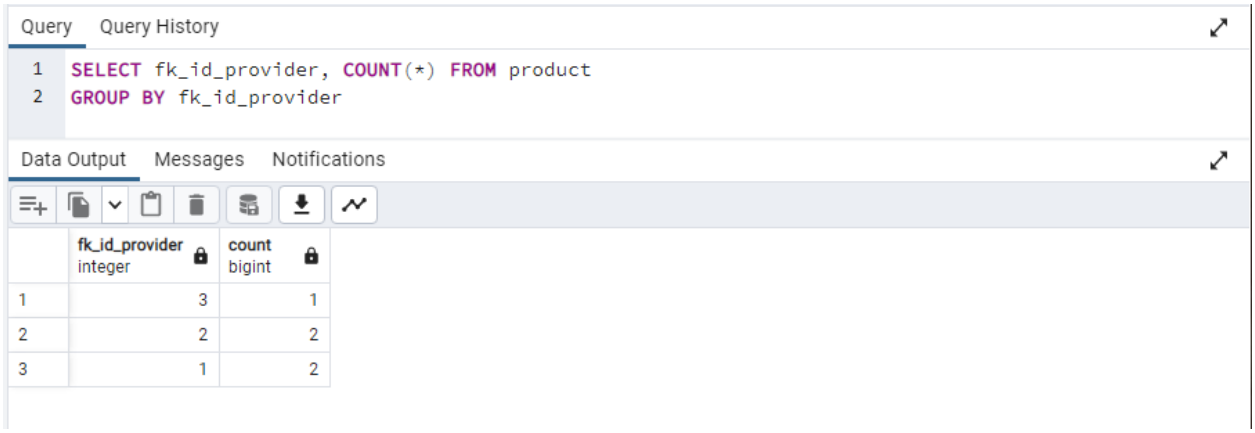
SELECT MAX(result_cost) AS "Самый дорогой заказ"

FROM result_order



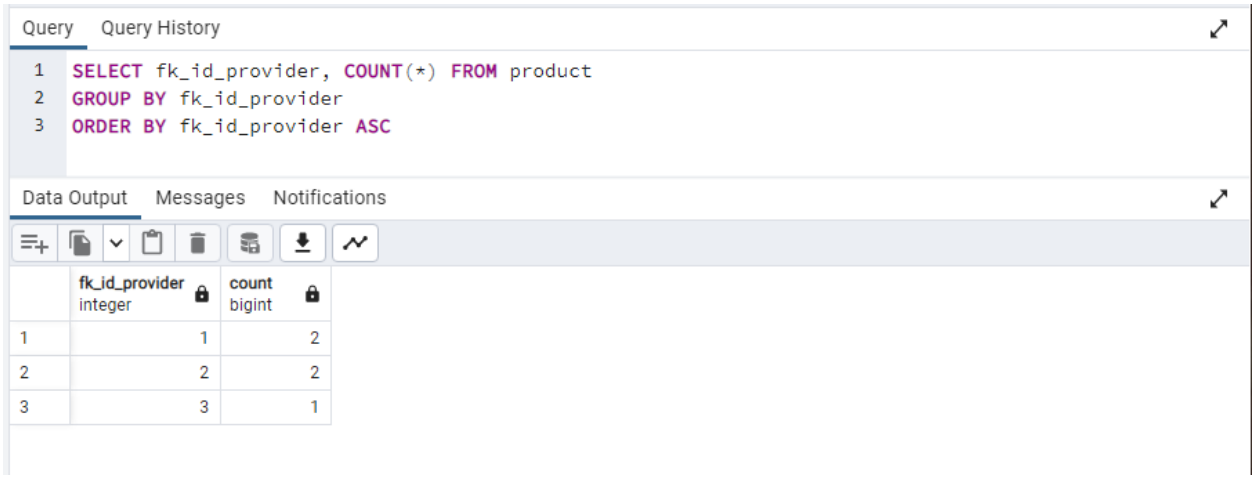
Вывод количества поставляемых продуктов от определенного поставщика

```
SELECT fk_id_provider, COUNT(*) FROM product
GROUP BY fk_id_provider
```



Вывод количества поставляемых продуктов от определенного поставщика по возрастанию ID поставщика

```
SELECT fk_id_provider, COUNT(*) FROM product
GROUP BY fk_id_provider
ORDER BY fk_id_provider ASC
```



Поиск заказанного продукта с наименьшим наличием товара на складе.

SELECT * FROM result_order

WHERE fk_id_product = (SELECT id_product FROM product WHERE min_quantity = (SELECT MAX(min_quantity) FROM product))

Query Query History

```
SELECT * FROM result_order
WHERE fk_id_product = (SELECT id_product FROM product WHERE min_quantity
                       = (SELECT MAX(min_quantity) FROM product))
```

Data Output Messages Notifications

	id_result_order [PK] integer	fk_id_product integer	quantity integer	discount integer	result_cost integer
1	1	2	5	5	30

Поиск товаров с ценой, меньше средней

SELECT * FROM product

WHERE product_cost < (SELECT AVG(product_cost) FROM product)

Query Query History

```
SELECT * FROM product
WHERE product_cost < (SELECT AVG(product_cost) FROM product)
```

Data Output Messages Notifications

	id_product [PK] integer	fk_id_provider integer	fk_product_type integer	product_cost integer	availability integer	waiting boolean	min_quantity integer
1	1	1	1	45	0	true	2
2	2	1	1	23	0	true	25
3	3	2	1	59	0	false	4
4	4	2	1	72	0	false	4

Поиск покупателя с самым длинным адресом

SELECT * FROM clients

WHERE CHARACTER_LENGTH(client_adress) =

(SELECT MAX(CHARACTER_LENGTH(client_adress))FROM clients)

Query Query History

```
SELECT * FROM clients
    WHERE CHARACTER_LENGTH(client_adress) =
    (SELECT MAX(CHARACTER_LENGTH(client_adress))FROM clients)
```

Data Output Messages Notifications

	id_client [PK] integer	client_fio character varying (60)	client_adress character varying	client_number character varying (11)
1	1	Катрин Коркоран	Филадельфия, Пенсильвания, США	89089284526

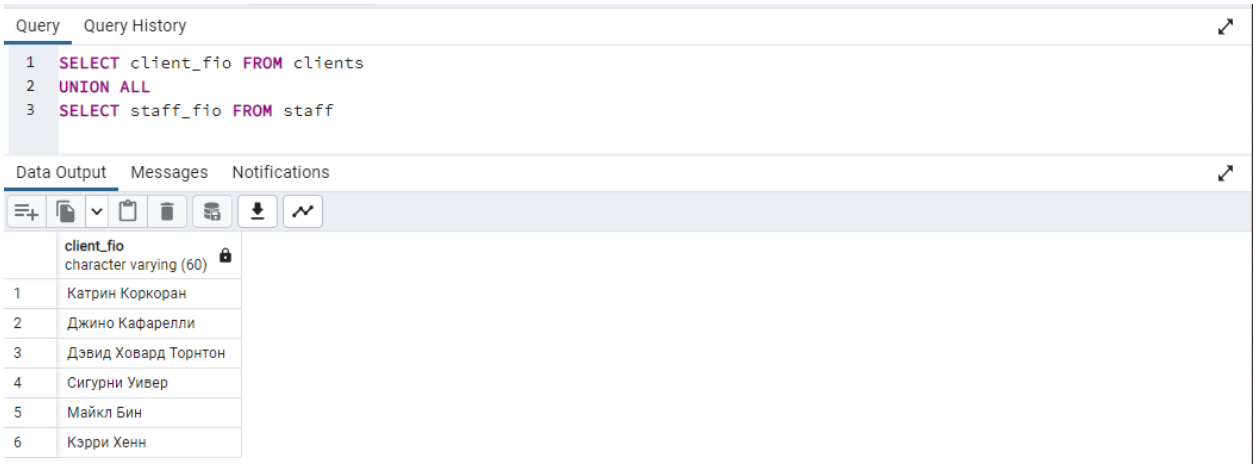
UNION используется для объединения результирующих наборов из 2 или более операторов SELECT.

Вывести все покупателей и продавцов

SELECT client_fio FROM clients

UNION ALL

SELECT staff_fio FROM staff



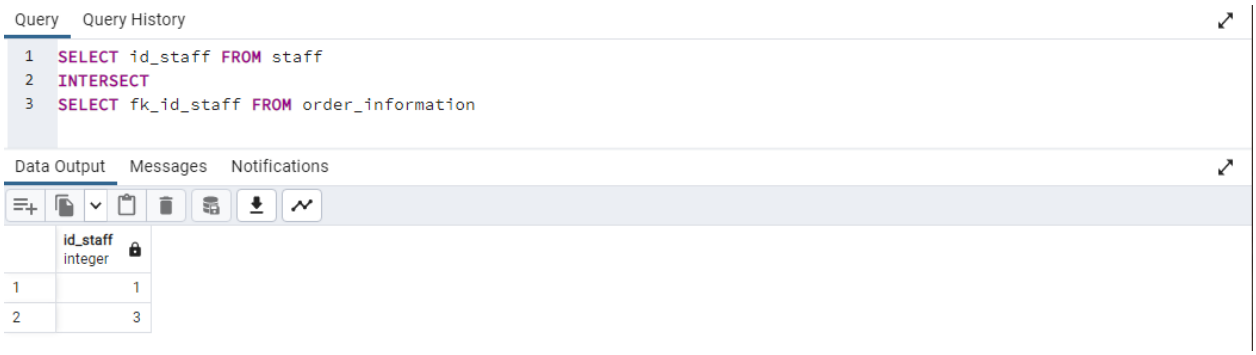
INTERSECT – выводит одинаковые строки из первого, второго и последующих наборов данных.

Менеджеры, которые что-то продали

SELECT id_staff FROM staff

INTERSECT

SELECT fk_id_staff FROM order_information



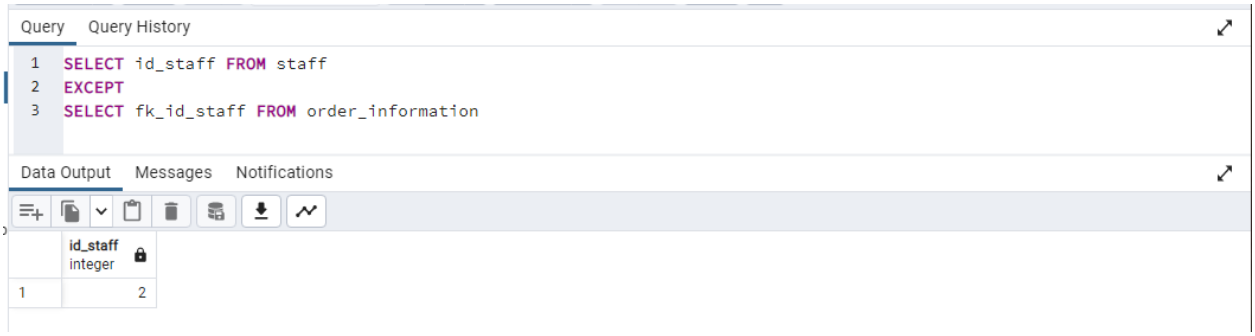
EXCEPT — выводит только те данные из первого набора строк, которых нет во втором наборе.

Сотрудники, которые ничего не продали.

SELECT id_staff FROM staff

EXCEPT

SELECT fk_id_staff FROM order_information



Выражение CASE – условный оператор языка SQL
Данный оператор позволяет осуществить проверку условий и возвратить в зависимости от выполнения того или иного условия тот или иной результат.

SELECT description,

CASE

WHEN description = 'алкогольная продукция, предназначенная для пользователей старше 18 лет'

THEN 'алкашка'

WHEN description = 'товары, недоступные для детей'

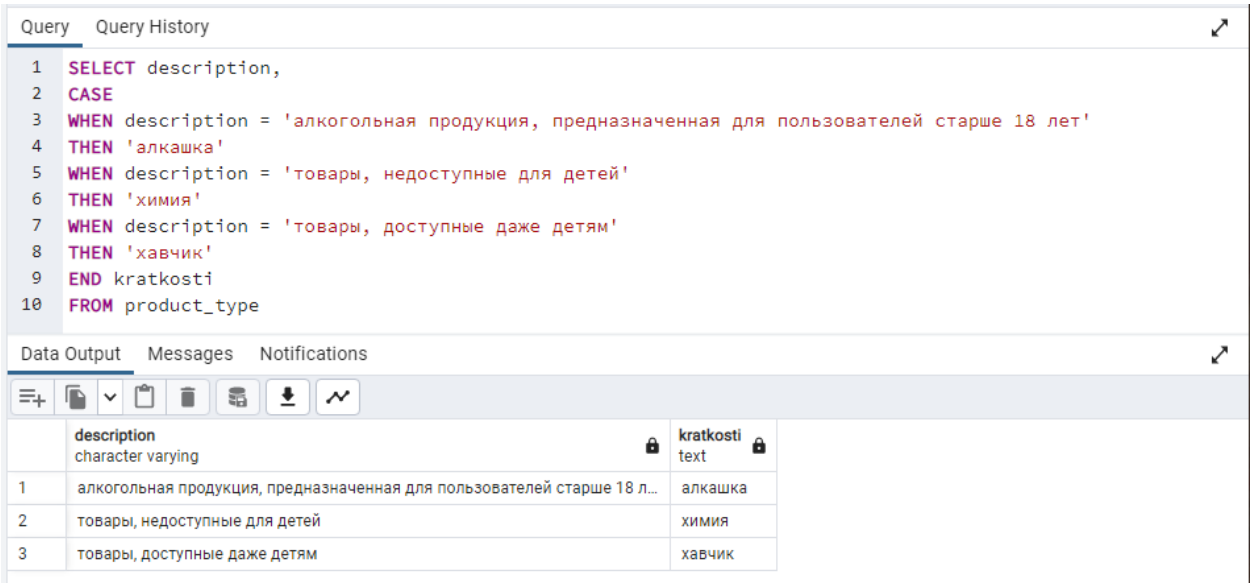
THEN 'химия'

WHEN description = 'товары, доступные даже детям'

THEN 'хавчик'

END kratkosti

FROM product_type



Из строк левой_таблицы и правой_таблицы объединяются и возвращаются только те строки, по которым выполняются условия_соединения.

SELECT staff_fio, id_staff, fk_id_result_order, delivery_type FROM staff

JOIN

order_information ON id_staff = fk_id_staff

Query Query History

```
1 SELECT staff_fio, id_staff, fk_id_result_order, delivery_type FROM staff
2 JOIN
3 order_information ON id_staff = fk_id_staff
```

Data Output Messages Notifications

	staff_fio character varying (60)	id_staff integer	fk_id_result_order integer	delivery_type character varying
1	Сигурни Уивер	1	2	поезд
2	Сигурни Уивер	1	1	самолет
3	Кэрри Хенн	3	3	самолет

Для недостающих данных вместо строк правой_таблицы вставляются NULL-значения.

SELECT staff_fio, id_staff, fk_id_result_order, delivery_type FROM staff

LEFT JOIN

order_information ON id_staff = fk_id_staff

Query Query History

```
1 SELECT staff_fio, id_staff, fk_id_result_order, delivery_type FROM staff
2 LEFT JOIN
3 order_information ON id_staff = fk_id_staff
```

Data Output Messages Notifications

	staff_fio character varying (60)	id_staff integer	fk_id_result_order integer	delivery_type character varying
1	Сигурни Уивер	1	2	поезд
2	Сигурни Уивер	1	1	самолет
3	Кэрри Хенн	3	3	самолет
4	Майкл Бин	2	[null]	[null]

Для недостающих данных вместо строк левой_таблицы вставляются NULL-значения.

SELECT staff_fio, id_staff, fk_id_result_order, delivery_type FROM staff

RIGHT JOIN

order_information ON id_staff = fk_id_staff

Query Query History

```
1  SELECT staff_fio, id_staff, fk_id_result_order, delivery_type FROM staff
2  RIGHT JOIN
3  order_information ON id_staff = fk_id_staff
```

Data Output Messages Notifications

	staff_fio character varying (60)	id_staff integer	fk_id_result_order integer	delivery_type character varying
1	Сигурни Уивер	1	2	поезд
2	Сигурни Уивер	1	1	самолет
3	Кэрри Хенн	3	3	самолет

Если для строк левой_таблицы и правой_таблицы выполняются условия_соединения, то они объединяются в одну строку. Для строк, для которых не выполняются условия_соединения, NULL-значения вставляются на место левой_таблицы, либо на место правой_таблицы, в зависимости от того данных какой таблицы в строке не имеется

SELECT staff_fio, id_staff, fk_id_result_order,

delivery_type, delivery_cost FROM staff

FULL JOIN

order_information ON id_staff = fk_id_staff

Query Query History

```
1  SELECT staff_fio, id_staff, fk_id_result_order,
2  delivery_type, delivery_cost FROM staff
3  FULL JOIN
4  order_information ON id_staff = fk_id_staff
```

Data Output Messages Notifications

	staff_fio character varying (60)	id_staff integer	fk_id_result_order integer	delivery_type character varying	delivery_cost integer
1	Сигурни Уивер	1	2	поезд	5000
2	Сигурни Уивер	1	1	самолет	14250
3	Кэрри Хенн	3	3	самолет	9500
4	Майкл Бин	2	[null]	[null]	[null]

Объединение каждой строки левой_таблицы со всеми строками правой_таблицы

SELECT staff_fio, id_staff, fk_id_result_order,

delivery_type, delivery_cost FROM staff

CROSS JOIN order_information

Query | Query History

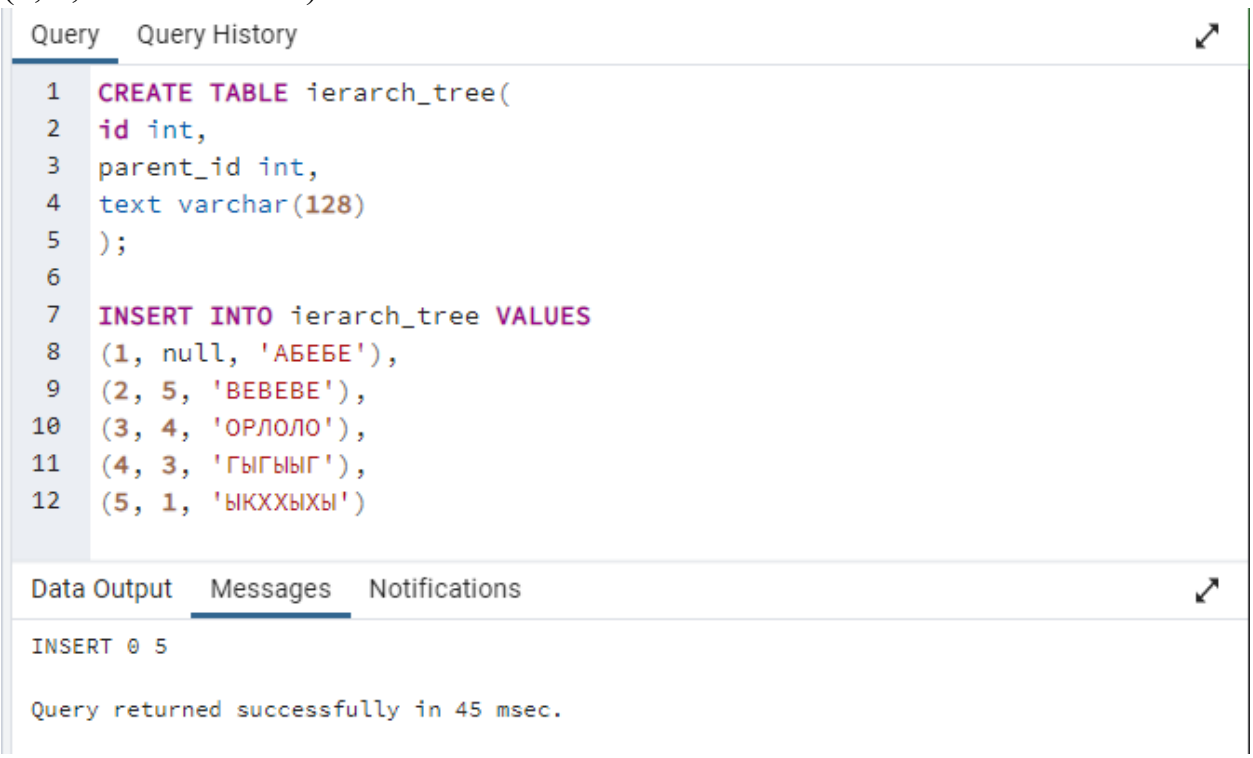
```
1  SELECT staff_fio, id_staff, fk_id_result_order,
2  delivery_type, delivery_cost FROM staff
3  CROSS JOIN order_information
```

Data Output | Messages | Notifications

	staff_fio character varying (60)	id_staff integer	fk_id_result_order integer	delivery_type character varying	delivery_cost integer
1	Сигурни Уивер	1	2	поезд	5000
2	Майкл Бин	2	2	поезд	5000
3	Кэрри Хенн	3	2	поезд	5000
4	Сигурни Уивер	1	1	самолет	14250
5	Майкл Бин	2	1	самолет	14250
6	Кэрри Хенн	3	1	самолет	14250
7	Сигурни Уивер	1	3	самолет	9500
8	Майкл Бин	2	3	самолет	9500
9	Кэрри Хенн	3	3	самолет	9500

```
CREATE TABLE ierarch_tree(
id int,
parent_id int,
text varchar(128)
);

INSERT INTO ierarch_tree VALUES
(1, null, 'АБЕБЕ'),
(2, 5, 'ВЕВЕВЕ'),
(3, 4, 'ОРЛОЛО'),
(4, 3, 'ГЫГЫЫГ'),
(5, 1, 'ЫКХХЫХЫ')
```



```
WITH RECURSIVE temp1 (id, parent_id, text, path) as (
SELECT t1.id, t1.parent_id, t1.text, CAST (t1.text AS varchar (50)) AS path
FROM ierarch_tree t1 WHERE t1.text =  'BEBEBE'
UNION
SELECT t2.id, t2.parent_id, t2.text, CAST (temp1.path || '->'|| t2.text AS varchar(50))
FROM ierarch_tree t2 INNER JOIN temp1 ON (temp1.parent_id = t2.id))
SELECT * FROM temp1
```