

ФГБОУ ВО

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

Кафедра __ИСиТ__

Специальность ИС-21

ОТЧЕТ

о выполнении лабораторной работы

Выполнил:

Костин Е.А.

Дата:

« » апреля 2023 г.

Норильск 2023

Лабораторная работа №4

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

Задание:

1. 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.
2. Для каждого запроса подписать, что именно он возвращает с учетом предметной области (запросы со смыслом, а не только синтаксически правильные операторы). Отчет по лабораторной работе должен содержать:
 1. Фамилию и номер группы учащегося, задание.
 2. Коды операций.
 3. Принтскрины всех выполненных операторов.

- a. запрос с условием на числовые данные (>,<=, between);
select * from drug where price between 300 and 1300
Указать лекарства с ценой от 300 до 1300

Query Query History

```
1 select * from drug
2 where price between 300 and 1300
```

Data Output Messages Notifications

	id_drug [PK] integer	name character varying (64)	price integer	fk_producer_id integer	fk_provider_id integer
1	2	Гаврилин	1300	3	2
2	3	Гексорал	800	3	3

- b. запрос с условием на текстовые данные (LIKE, IN);
select * from personal where secondname like '%нос'
Указать работников фамилия которых оканчивается на “НОС”

Query Query History

```
1 select * from personal
2 where secondname like '%нос'
```

Data Output Messages Notifications

	id_personal [PK] integer	firstname character varying (64)	secondname character varying (64)	fk_post_id integer
1	1	Максим	Долгонос	1
2	2	Глеб	Долгонос	1

с. запрос с вычисляемым полем;

```
select sum(price*2) from drug
```

Указать сумму всех лекарств если цена поднимется в 2 раза

Query		Query History
1		<pre>select sum(price*2) from drug</pre>
Data Output		Messages
		Notifications
	sum bigint	
1	25850	

d. запрос к нескольким таблицам (без явного указания JOIN);

```
select * from persindrug, personal
```

Вывести таблицу работников и работников продающих таблетки вместе

Query

Query History

1

select * from persindrug, personal

Data Output

Messages

Notifications

	fk_person_id integer	fk_drug_id integer	id_personal integer	firstname character varying (64)	secondname character varying (64)	fk_post_id integer
1	1	1	1	Максим	Долгонос	1
2	1	1	2	Глеб	Долгонос	1
3	1	1	3	Виталья	Богатырь	2
4	1	1	4	Дима	Маслеников	3
5	1	2	1	Максим	Долгонос	1
6	1	2	2	Глеб	Долгонос	1
7	1	2	3	Виталья	Богатырь	2
8	1	2	4	Дима	Маслеников	3
9	1	4	1	Максим	Долгонос	1
10	1	4	2	Глеб	Долгонос	1

Total rows: 44 of 44

Query complete 00:00:00.088

е. запрос с агрегирующей функцией (AVG, SUM, COUNT, MIN, MAX);

```
select SUM(pay) as "Сумма заработной платы" from post
```

Вывести сколько денег уходит на зарплаты по всем должностям

Query		Query History
1		<pre>select SUM(pay) as "Сумма заработной платы" from post</pre>
Data Output		Messages
Сумма заработной платы		
bigint		
1	10155000	

ф. запрос с группировкой (GROUP BY);

```
select secondname, count(*) from personal group by secondname
```

Вывести фамилии работников и количество человек с одинаковыми фамилиями группируя по фамилии

Query

Query History

1 select secondname, count(*) from personal

2 group by secondname

Data Output

Messages

Notifications

secondname

character varying (64)

count

bigint

1

Маслеников

1

2

Долгонос

2

3

Богатырь

1

g. запрос с сортировкой (ORDER BY);

```
select secondname, count(*) from personal  
  
group by secondname  
  
order by count desc
```

Вывести фамилии работников и количество человек с одинаковыми фамилиями сортируя по убыванию количества

Query

Query History

1

select secondname, count(*) from personal

2

group by secondname

3

order by count desc

Data Output

Messages

Notifications

≡+

📄

▼

📋

🗑️

🗄️

⬇️

📈

	secondname character varying (64) 🔒	count bigint 🔒
1	Долгонос	2
2	Маслеников	1
3	Богатырь	1

h. запрос с вложенным подзапросом (не менее 3 видов);

`select * from drug where price < (select AVG(price) from drug)`

Вывести все лекарства дешевле средней цены лекарств

Query Query History

```
1 select * from drug where price <
2 (select AVG(price) from drug)
3
```

Data Output Messages Notifications

	id_drug [PK] integer	name character varying (64)	price integer	fk_producer_id integer	fk_provider_id integer
1	1	Фурацилин	200	1	3
2	2	Гаврилин	1300	3	2
3	3	Гексорал	800	3	3
4	4	Неактивированный уголь	110	2	2
5	6	Стоп-Запор	270	2	1
6	7	Боярышник	145	2	3

```
select * from personal
where fk_post_id = (select id_post from post where pay = (select
max(pay) from post))
```

Вывести человека который получает наибольшую зарплату

Query

Query History

```
1 select * from personal
2 where fk_post_id = (select id_post from post where pay = (select max(pay) from post))
```

Data Output

Messages

Notifications

<

```
select * from drug
where id_drug = (select fk_drug_id from drug_on_recipe where numrecipe
= (select MAX(numrecipe) from drug_on_recipe))
```

Вывести лекарство которое продали с самым большим номером рецепта

Query

Query History

1

2

3

```
select * from drug
where id_drug = (select fk_drug_id from drug_on_recipe where numrecipe = (select MAX(numrecipe) from drug_on_recipe))
```

Data Output

Messages

Notifications

	id_drug [PK] integer	name character varying (64)	price integer	fk_producer_id integer	fk_provider_id integer
1	2	Гаврилин	1300	3	2

i. запрос с оператором UNION;

```
select name from produser  
union all select name from provider
```

Вывести одним списком название поставщиков и производителей

Query Query History

1 select name from produser
2 union all select name from provider

Data Output Messages Notifications

≡+ 📄 ▼ 📋 🗑️ 🗄️ ⬇️ 📈

	name character varying (64) 🔒
1	Чих-Пых
2	Пух-Плюх
3	Жыж-Пыж
4	Баломут
5	Травы-Нави
6	Фармолаб.Inc

j. запрос с оператором INTERSECT;
select fk_person_id from persindrug
intersect select fk_person_id from persinservice
Вывести id работников которые продали лекарства и оказали услугу

Query		Query History	
1	select	fk_person_id	from persindrug
2	intersect select	fk_person_id	from persinservice

Data Output		Messages	Notifications
fk_person_id integer			
1	2		
2	1		

k. запрос с оператором EXCEPT;
select fk_DRUG_id from persindrug
except select fk_person_id from persinservice
Вывести id лекарств которые не совпадают с id продавцов которые оказывали услуги

QueryQuery History

```

1 select secondname,
2 case when secondname = 'Долгонос' then 'Безруков'
3 when secondname = 'Богатырь' then 'Богач'
4 when secondname = 'Маслеников' then 'Масlach'
5 end klichka
6 from personal

```

Data OutputMessagesNotifications

+

📄

▼

📋

🗑️

🗄️

⬇️

📈

	secondname character varying (64) 🔒	klichka text 🔒
1	Долгонос	Безруков
2	Долгонос	Безруков
3	Богатырь	Богач
4	Маслеников	Масlach

m. запрос с оператором JOIN (пять видов);

```
select * from post join personal on id_post = fk_post_id
```

Объединить таблицу сотрудников и должностей по id должности

Query

Query History

1

select * from post join personal on id_post = fk_post_id

Data Output

Messages

Notifications

	id_post integer	name character varying (64)	pay integer	id_personal integer	firstname character varying (64)	secondname character varying (64)	fk_post_id integer
1	1	Фармацевт	55000	1	Максим	Долгонос	1
2	1	Фармацевт	55000	2	Глеб	Долгонос	1
3	2	Юрист	100000	3	Виталья	Богатырь	2
4	3	Директор	10000000	4	Дима	Маслеников	3

select * from drug left join drug_on_recipe on id_drug = fk_drug_id

объединить данные из таблицы лекарства с таблицей продажи лекарства по рецепту

связанных по ид лекарства

Query

Query History

1

select * from drug left join drug_on_recipe on id_drug = fk_drug_id

Data Output

Messages

Notifications

	id_drug integer	name character varying (64)	price integer	fk_producer_id integer	fk_provider_id integer	id_recipe integer	numrecipe integer	fk_drug_id integer
1	2	Гаврилин	1300	3	2	1	1729	2
2	5	Пилюзя	10100	1	1	2	1230	5
3	1	Фурацилин	200	1	3	3	1709	1
4	6	Стоп-Запор	270	2	1	[null]	[null]	[null]
5	4	Неактивированный уголь	110	2	2	[null]	[null]	[null]
6	3	Гексорал	800	3	3	[null]	[null]	[null]
7	7	Боярышник	145	2	3	[null]	[null]	[null]

select * from service right join persinservice on id_service = fk_service_id

объединить данные из таблицы продавца услуг с услугами связанные по ид услуги

Query

Query History

1 select * from service right join persinservice on id_service = fk_service_id

2

Data Output

Messages

Notifications

	id_service integer	name character varying (64)	price integer	fk_person_id integer	fk_service_id integer
1	1	Проверить зрение	500	2	1
2	4	Активировать уголь	10	2	4
3	2	Проверить давление	100	2	2
4	4	Активировать уголь	10	1	4
5	2	Проверить давление	100	1	2
6	3	Проверить температуру	50	1	3

```
select drug.name, producer.name, producer.inn from producer inner join drug on id_producer =
fk_producer_id
```

объединить таблицы поставщик и лекарства, выводя название, инн и какое лекарство поставляет поставщик

Query Query History	
1	<code>select drug.name, producer.name, producer.inn from producer inner join drug on id_producer = fk_producer_id</code>
2	
Data Output Messages Notifications	
	name character varying (64)
	name character varying (64)
	inn integer
1	Гексорал
2	Гаврилин
3	Пилюзя
4	Фурацилин
5	Боярышник
6	Стоп-Запор
7	Неактивированный уголь

```
select * from drug full join provider on drug.id_provider = provider.id_provider
```

полность объединить таблицу лекарст и потсавщиков

Query Query History	
1	<code>select * from drug full join provider on drug.id_provider = provider.id_provider</code>
2	
Data Output Messages Notifications	
	id_drug integer
	name character varying (64)
	price integer
	fk_producer_id integer
	id_provider integer
	id_provider integer
	name character varying (64)
	inn integer
	adress character varying (64)
1	4 Неактивированный уголь 110 2 2 2 Баломут 234078951 с.Яшкино
2	2 Гаврилин 1300 3 2 2 Баломут 234078951 с.Яшкино
3	7 Боярышник 145 2 3 3 Травы-Нави 173329862 г.Ногинск
4	3 Гексорал 800 3 3 3 Травы-Нави 173329862 г.Ногинск
5	1 Фурацилин 200 1 3 3 Травы-Нави 173329862 г.Ногинск
6	6 Стоп-Запор 270 2 1 1 Фармолаб.Ипс 999111222 г.Петнько
7	5 Пилюзя 10100 1 1 1 Фармолаб.Ипс 999111222 г.Петнько

n. иерархический запрос.

```
with recursive zaur (id_post, name, pay, important, path) as (
    select t1.id_post, t1.name, t1.pay, t1.important, cast (t1.name
as varchar(50)) as path
    from post t1 where t1.name = 'Фармацевт'
    union
    select t2.id_post, t2.name, t2.pay, t2.important, cast (zaur.path
|| '->' || t2.name as varchar(50))
    from post t2 join zaur on (zaur.important = t2.id_post))
select * from zaur
```

Отобразить иерархию должностей в аптеке

Query Query History

```
1 with recursive zaur (id_post, name, pay, important, path) as (  
2     select t1.id_post, t1.name, t1.pay, t1.important, cast (t1.name as varchar(50)) as path  
3     from post t1 where t1.name = 'Фармацевт'  
4     union  
5     select t2.id_post, t2.name, t2.pay, t2.important, cast (zaur.path || '->' || t2.name as varchar(50))  
6     from post t2 join zaur on (zaur.important = t2.id_post)  
7     select * from zaur
```

Data Output Messages Notifications



	id_post integer	name character varying (64)	pay integer	important integer	path character varying (50)
1	4	Фармацевт	999999	2	Фармацевт
2	2	Директор	100000	1	Фармацевт->Директор
3	1	Владелец	55000	[null]	Фармацевт->Директор->Владелец