

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**ЗАПОЛЯРНЫЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМ. Н. М.
ФЕДОРОВСКОГО**

ОТЧЕТ

**по лабораторной работе
по дисциплине “Базы данных”**

Тема: Запросы

Группа: ИС-21

Студент: Игнатьев Данила

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).

Код и скрины:

- 1. WHERE(SELECT salary FROM specialties WHERE specialty_id = doctors.specialty_id) > 30000
- 2. WHERE(SELECT salary FROM specialties WHERE specialty_id = doctors.specialty_id) = 30000
- 3. WHERE(SELECT salary FROM specialties WHERE specialty_id = doctors.specialty_id) BETWEEN 30000 AND 40000

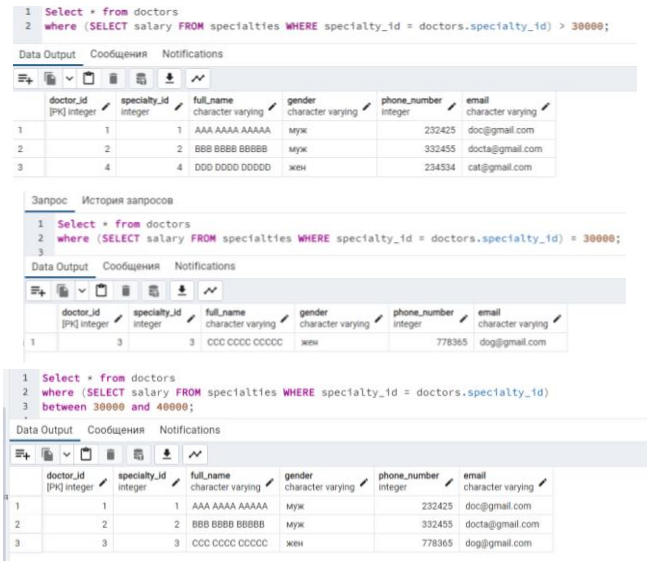


Рис.1 - запросы с условием на числовые данные (>,<=, between)

- 1. SELECT * FROM medical_records WHERE report LIKE 'A%'
- 2. SELECT * FROM medical_records WHERE report LIKE '%F%'
- 3. SELECT id, payment, offices_num FROM record WHERE offices_num IN (101, 102)

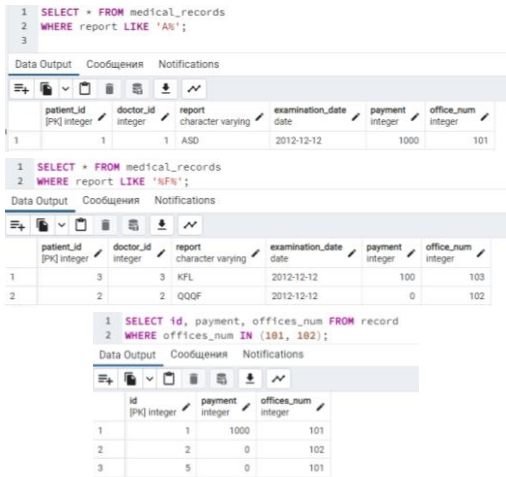
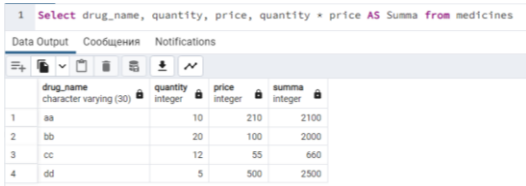


Рис.2 - запросы с условием на текстовые данные (LIKE, IN)

SELECT drug_name, quantity, price, quantity * price AS Summa from medicines



The screenshot shows a query editor with the following SQL statement:

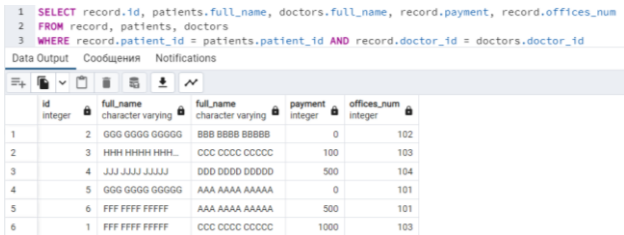
```
1 SELECT drug_name, quantity, price, quantity * price AS Summa from medicines
```

Below the editor is a table with the following data:

	drug_name character varying (30)	quantity integer	price integer	summa integer
1	aa	10	210	2100
2	bb	20	100	2000
3	cc	12	55	660
4	dd	5	500	2500

Рис.3 - запрос с вычисляемым полем

SELECT record.id, patients.full_name, doctors.full_name, record_payment, record.offices_num
FROM record, patients, doctors
WHERE record.patient_id = patients.patient_id AND record.doctor_id = doctors.doctor_id



The screenshot shows a query editor with the following SQL statement:

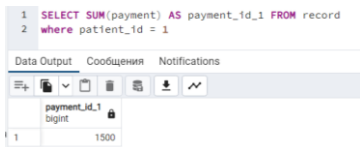
```
1 SELECT record.id, patients.full_name, doctors.full_name, record.payment, record.offices_num  
2 FROM record, patients, doctors  
3 WHERE record.patient_id = patients.patient_id AND record.doctor_id = doctors.doctor_id
```

Below the editor is a table with the following data:

	id integer	full_name character varying	full_name character varying	payment integer	offices_num integer
1	2	GGG GGGG GGGGG	BBB BBBB BBBB	0	102
2	3	HHH HHHH HHH	CCC CCCC CCCCC	100	103
3	4	JJJ JJJJ JJJJ	DDD DDDD DDDDD	500	104
4	5	GGG GGGG GGGGG	AAA AAAA AAAA	0	101
5	6	FFF FFFF FFFFF	AAA AAAA AAAA	500	101
6	1	FFF FFFF FFFFF	CCC CCCC CCCCC	1000	103

Рис.4 - запрос к нескольким таблицам (без явного указания JOIN)

SELECT SUM(payment) AS payment_id_1 FROM record WHERE patient_id = 1



The screenshot shows a query editor with the following SQL statement:

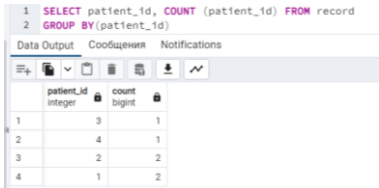
```
1 SELECT SUM(payment) AS payment_id_1 FROM record  
2 where patient_id = 1
```

Below the editor is a table with the following data:

	payment_id_1 bigint
1	1500

Рис.5 - запрос с агрегирующей функцией (AVG, SUM, COUNT, MIN, MAX)

SELECT patient_id, COUNT (patient_id) FROM record GROUP BY(patient_id)



The screenshot shows a query editor with the following SQL statement:

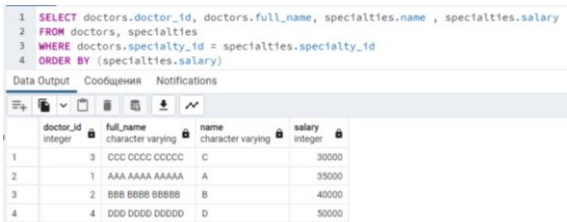
```
1 SELECT patient_id, COUNT (patient_id) FROM record  
2 GROUP BY(patient_id)
```

Below the editor is a table with the following data:

	patient_id integer	count bigint
1	3	1
2	4	1
3	2	2
4	1	2

Рис.6 - запрос с группировкой (GROUP BY)

SELECT doctors.doctor_id, doctors.full_name, specialties.name, specialties.salary FROM doctors, specialties
WHERE doctors.specialty_id = specialties.specialty_id ORDER BY (specialties.salary)



The screenshot shows a query editor with the following SQL statement:

```
1 SELECT doctors.doctor_id, doctors.full_name, specialties.name , specialties.salary  
2 FROM doctors, specialties  
3 WHERE doctors.specialty_id = specialties.specialty_id  
4 ORDER BY (specialties.salary)
```

Below the editor is a table with the following data:

	doctor_id integer	full_name character varying	name character varying	salary integer
1	3	CCC CCCC CCCCC	C	30000
2	1	AAA AAAA AAAA	A	35000
3	2	BBB BBBB BBBB	B	40000
4	4	DDD DDDD DDDDD	D	50000

Рис.7 - запрос с сортировкой (ORDER BY)

1. `SELECT doctors.doctor_id, doctors.full_name, specialties.name, specialties.salary FROM doctors, specialties WHERE doctors.specialty_id = (SELECT specialty_id FROM specialties WHERE name = 'A') AND specialties.salary = (SELECT salary FROM specialties WHERE name = 'A')`
2. `SELECT * FROM specialties WHERE specialties.specialty_id IN(SELECT specialties.specialty_id FROM specialties WHERE specialties.salary > 35000)`
3. `SELECT * FROM specialties WHERE EXISTS(SELECT * FROM offices WHERE offices.day_of_the_week_id = schedule.day_of_the_week_id)`

1 `SELECT doctors.doctor_id, doctors.full_name, specialties.name , specialties.salary`
 2 `FROM doctors, specialties`
 3 `WHERE doctors.specialty_id = (SELECT specialty_id FROM specialties WHERE name = 'A')`
 4 `AND specialties.salary = (SELECT salary FROM specialties WHERE name = 'A')`

doctor_id	full_name	name	salary
1	AAA AAAA AAAA	A	35000

1 `SELECT * FROM specialties`
 2 `WHERE specialties.specialty_id IN(SELECT specialties.specialty_id FROM specialties WHERE specialties.salary > 35000)`

specialty_id	name	salary	working_hours	description	status
1	2 B	40000	18:00:00	nothing	не требуются
2	4 D	50000	18:00:00	nothing	не требуются

1 `SELECT schedule.day_of_the_week FROM schedule`
 2 `WHERE EXISTS(SELECT * FROM offices WHERE offices.day_of_the_week_id = schedule.day_of_the_week_id)`

day_of_the_week
1 SSS
2 DDD
3 AAA
4 WWW

Рис.8 - запросы с вложенным подзапросом

`SELECT full_name, gender FROM doctors`
`UNION`
`SELECT full_name, gender FROM patients`

1 `SELECT full_name, gender FROM doctors`
 2 `UNION`
 3 `SELECT full_name, gender FROM patients`

full_name	gender
GGG GGGG GGGG	муж
JJJ JJJ JJJJ	жен
FFF FFFF FFFF	муж
DDD DDDD DDDD	жен
HHH HHHH HHH	жен
BBB BBBB BBBB	муж
CCC CCCC CCCC	жен
AAA AAAA AAAA	муж
FFF FFFF FFFF	trap

Рис.9 - запрос с оператором UNION

`SELECT staff_name FROM staff`
`INTERSECT`
`SELECT full_name FROM doctors`

1 `SELECT staff_name FROM staff`
 2 `INTERSECT`
 3 `SELECT full_name FROM doctors`

staff_name
FFF FFFF FFFF
AAA AAAA AAAA
CCC CCCC CCCC
DDD DDDD DDDD
BBB BBBB BBBB

Рис.10 - запрос с оператором INTERSECT

```
SELECT staff_name FROM staff
EXCEPT
SELECT full_name FROM doctors
```

staff_name
1 keHG GDH HDG
2 BDF dBD DDD
3 FDG GGFHH dGG
4 HDHD HFH JJJJ

Рис.11 - запрос с оператором EXCEPT

```
SELECT name, salary,
CASE WHEN salary <= 35000 THEN 'Бонусы и премии'
ELSE 'Только премии' END AS financian_support FROM specialties
```

name	salary	financian_support
1 A	35000	Бонусы и премии
2 B	40000	только премии
3 C	30000	Бонусы и премии
4 D	50000	только премии
5 F	42000	только премии

Рис.12 - запрос с выражением CASE

- 1. SELECT doctors.full_name, doctors.gender, specialties.name, specialties.salary FROM doctors INNER JOIN specialties ON doctors.specialty_id = specialty_id
- 2. SELECT staff1.staff_name, staff2.job FROM staff staff1 INNER JOIN staff staff2 ON staff2.job AND staff1.staff_name = staff2.staff_name ORDER BY job
- 3. SELECT doctors.full_name, staff.staff_name FROM doctors CROSS JOIN staff
- 4. SELECT doctors.full_name, staff.job FROM doctors LEFT JOIN staff ON doctors.full_name = staff.staff_name
- 5. SELECT doctors.full_name, staff.job FROM doctors RIGHT JOIN staff ON doctors.full_name = staff.staff_name

full_name	gender	name	salary
1 AAA AAAA AAAA	муж	A	35000
2 BBB BBBB BBBB	муж	B	40000
3 CCC CCCC CCCC	жен	C	30000
4 DDD DDDD DDDD	жен	D	50000
5 FFF FFFF FFFF	trap	F	42000

INNER JOIN

staff_name	job
1 BDF dBD DDD	cleaner
2 HDHD HFH JJJJ	cleaner
3 FDG GGFHH dGG	cloakroom attendant
4 keHG GDH HDG	cloakroom attendant
5 AAA AAAA AAAA	doctor
6 BBB BBBB BBBB	doctor
7 CCC CCCC CCCC	doctor
8 DDD DDDD DDDD	doctor
9 FFF FFFF FFFF	doctor

SELF JOIN

```

1 SELECT doctors.full_name, staff.staff_name FROM doctors
2 CROSS JOIN staff

```

	full_name character varying	staff_name character varying
1	AAA AAAA AAAAA	AAA AAAA AAAAA
2	BBB BBBB BBBB	AAA AAAA AAAAA
3	CCC CCCC CCCC	AAA AAAA AAAAA
4	DDD DDDD DDDD	AAA AAAA AAAAA
5	FFF FFFF FFFF	AAA AAAA AAAAA

CROSS JOIN

```

1 SELECT doctors.full_name, staff.job FROM doctors
2 LEFT JOIN staff ON doctors.full_name = staff.staff_name

```

	full_name character varying	job character varying
1	AAA AAAA AAAAA	doctor
2	BBB BBBB BBBB	doctor
3	CCC CCCC CCCC	doctor
4	DDD DDDD DDDD	doctor
5	FFF FFFF FFFF	doctor

LEFT JOIN

```

1 SELECT doctors.full_name, staff.job FROM doctors
2 RIGHT JOIN staff ON doctors.full_name = staff.staff_name

```

	full_name character varying	job character varying
1	AAA AAAA AAAAA	doctor
2	[null]	cleaner
3	[null]	cleaner
4	BBB BBBB BBBB	doctor
5	[null]	cloakroom attendant
6	CCC CCCC CCCC	doctor
7	DDD DDDD DDDD	doctor
8	FFF FFFF FFFF	doctor
9	[null]	cloakroom attendant

RIGHT JOIN

Рис.13 - запрос с оператором JOIN (пять видов)

```

WITH RECURSIVE staff_query AS
(
    SELECT staff_id, staff_name, job, boss_id FROM staff s
    WHERE boss_id is null
    UNION ALL
    SELECT s.staff_id, s.staff_name, s.job, s.boss_id FROM staff s
    INNER JOIN staff_query ss ON ss.staff_id = s.boss_id
)
SELECT * FROM staff_query

```

```

1 WITH RECURSIVE staff_query AS
2 (
3     SELECT staff_id, staff_name, job, boss_id FROM staff s
4     WHERE boss_id is null
5     UNION ALL
6     SELECT s.staff_id, s.staff_name, s.job, s.boss_id FROM staff s
7     INNER JOIN staff_query ss ON ss.staff_id = s.boss_id
8 )
9 SELECT * FROM staff_query

```

	staff_id integer	staff_name character varying	job character varying	boss_id integer
1	10	hgfhghthf fc fcdcs	chief	[null]
2	1	AAA AAAA AAAAA	senior doctor	10
3	8	FFF FFFF FFFF	senior doctor	10
4	14	.dnch wjvbe efv	senior nurse	10
5	2	BDF iBD DDD	cleaner	14
6	3	HDHD HFH JJJJ	cleaner	14
7	5	FDG GGFH dGG	cloakroom attendant	14
8	9	kehG GDH HDG	cloakroom attendant	14
9	4	BBB BBBB BBBB	doctor	2
10	6	CCC CCCC CCCC	doctor	2
11	7	DDD DDDD DDDD	doctor	2
12	11	sdtd dfb tgr	nurse	7
13	12	sdcs efv	nurse	6
14	13	wfwd bhb olk	nurse	4

Рис.14 - иерархический запрос