

УНИВЕРСИТЕТ ИТМО  
ФАКУЛЬТЕТ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

**ОТЧЕТ  
О ЛАБОРАТОРНОЙ РАБОТЕ № 5**

По теме: Создание SQL-запросов в Pgadmin  
По дисциплине: Основы проектирования баз данных  
Специальность 09.02.07 «Информационные системы и программирование»

Преподаватель:

\_\_\_\_\_ Говоров А.И.

Дата: «\_\_» \_\_\_\_\_ 20\_\_ г.

Оценка \_\_\_\_\_

Выполнил:

Студент группы № Y2339

\_\_\_\_\_ Карелова А.С.

**Цель работы:** овладеть практическими навыками создания некоторого количества SQL-запросов.

**Порядок выполнения работы:** познакомиться с инструкциями по работе с отладчиком, написать программу по полученному заданию и, выполняя ее по командам, заполнить таблицы в отчете. Операции умножения и деления выполнять через сложение и вычитание в виде отдельных процедур. Вызов и возврат из процедуры выполнять через команды переходов. При делении учитывать остаток.

**Программное обеспечение:** пакеты лицензионного ПО общего назначения.

**Практическое задание:**

1. выбор значений, заданных атрибутов из более, чем двух таблиц, с сортировкой;
2. использование условий WHERE, состоящих из более, чем одного условия;
3. использование функций для работы с датами;
4. использование строковых функций;
5. запрос с использованием подзапросов;
6. вычисление групповой (агрегатной) функции;
7. вычисление групповой (агрегатной) функции с условием HAVING;
8. использование предикатов EXISTS, ALL, SOME и ANY;
9. использование запросов с операциями реляционной алгебры (объединение, пересечение и т.д.);
10. использование объединений запросов (inner join и т.д.).

## Выполнение задания:

На рисунке 1 представлена разработанная ранее система, предназначенная для работников частной клиники.

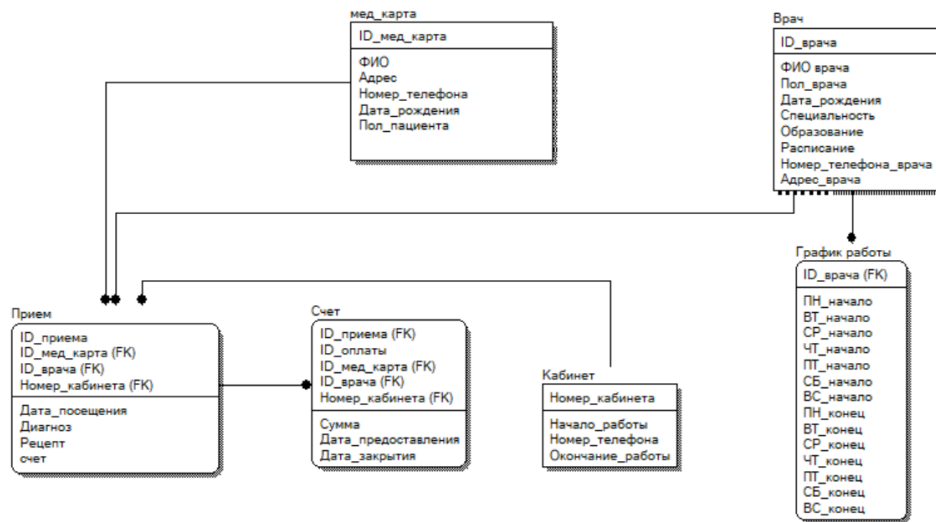


Рисунок 1 - Модель БД

1. На рисунке 2 представлены полные имена врачей и пациентов, которые были в определенные приемы, организованные по возрастанию ID приема.

```
SELECT a_id, d_full_name, mc_full_name FROM "Clinic1"."appointment", "Clinic1"."doctor", "Clinic1"."medical_card" WHERE a_doctor = d_id AND a_patient = mc_id ORDER BY a_id;
```

a_id integer	d_full_name text	mc_full_name text
1	Doctor 2	Patient 1
2	Doctor 1	Patient 5
3	Doctor 6	Patient 2
4	Doctor 3	Patient 3
5	Doctor 5	Patient 8
6	Doctor 8	Patient 3
7	Doctor 7	Patient 7
8	Doctor 8	Patient 3

Рисунок 2 - Результат запроса

2. На рисунке 3 представлена информация о врачах, которые принимали пациентов в период с 01.02.2020 по 03.02.2020, организованная по ID врача.

```
SELECT a_doctor, d_full_name, d_gender, d_dob, d_phone_num, d_profession, d_education, d_address FROM "Clinic1"."appointment", "Clinic1"."doctor" WHERE a_doctor = d_id AND a_date BETWEEN '2020-02-01' and '2020-02-03' ORDER BY a_doctor;
```

a_doctor integer	d_full_name text	d_gender boolean	d_dob date	d_phone_num integer	d_profession text	d_education text	d_address text
1	Doctor 1	true	1978-01-01	1265522	psychologist	8 years	Street 5 home ...
2	Doctor 2	false	1983-03-03	1234244	gynecologist	10 years	Street 6 home ...
3	Doctor 3	false	1989-04-01	5623447	optometrist	6 years	Street 7 home ...
6	Doctor 6	false	1982-03-03	1234243	gynecologist	10 years	Street 6 home ...

*Рисунок 3 - Результат запроса*

3. На рисунке 4 представлена информация об интервале между датой приема и датой рождения врача, организованная по возрастанию разницы (показывает возраст врача на дату приема).

```
SELECT d_full_name, age ("Clinic1"."appointment"."a_date", "Clinic1"."doctor"."d_dob") from "Clinic1"."appointment", "Clinic1"."doctor" WHERE a_doctor = d_id ORDER BY age;
```

d_full_name text	age interval
Doctor 3	30 years 10 mons 2 days
Doctor 7	31 years 10 mons 4 days
Doctor 2	36 years 10 mons 29 days
Doctor 6	37 years 10 mons 30 days
Doctor 1	42 years 1 mon 1 day
Doctor 5	43 years 1 mon 3 days
Doctor 8	53 years 5 mons 3 days
Doctor 8	53 years 5 mons 4 days

*Рисунок 4 - Результат запроса*

4. На рисунке 5 представлена информация о количестве символов в полных именах пациентов, организованно по возрастанию номера ID пациента.

```
SELECT mc_full_name, length("Clinic1"."medical_card"."mc_full_name") FROM "Clinic1"."medical_card" ORDER BY mc_id;
```

	mc_full_name text	length integer
1	Patient 1	9
2	Patient 2	9
3	Patient 3	9
4	Patient 4	9
5	Patient 5	9
6	Patient 6	9
7	Patient 7	9
8	Patient 8	9

*Рисунок 5 - Результат запроса*

5. На рисунке 6 представлена информация пациентов с диагнозом miosis: id, имена пациентов и то, как долго они не оплачивали; организовано по возрастанию id пациента.

```
SELECT a_patient, mc_full_name, age("Clinic1"."payment"."p_date_closed", "Clinic1"."payment"."p_date_opened") FROM "Clinic1"."payment", "Clinic1"."medical_card", "Clinic1"."appointment" WHERE a_diagnosis = 'miosis' AND a_patient = mc_id AND a_payment = p_id ORDER BY mc_id;
```

a_patient integer	mc_full_name text	age interval
3	Patient 3	2 days
7	Patient 7	25 days

*Рисунок 6 - Результат запроса*

6. На рисунке 7 предоставлена информация о максимальной сумме оплаты за прием.

```
SELECT max("payment"."p_sum") as Max_sum from "Clinic1"."payment" join "Clinic1"."appointment" on "appointment"."a_payment" = "payment"."p_id";
```

	max_sum integer
1	3000

Рисунок 7 - Результат запроса

7. Вывод номеров счетов, за которые должны были заплатить больше, чем 1000 , организованная по номеру счетов (рисунок 8).

```
SELECT p_id, max(p_sum) from "Clinic1"."payment" GROUP BY p_id
having max("p_sum") > 1000 ORDER BY p_id;
```

p_id [PK] integer	max integer
2	3000
5	3000
7	3000

Рисунок 8 - Результат запроса

8. На рисунке 9 представлена информация о пациентах, которые зарегистрированы, но ещё никогда не были на приёме.

```
SELECT distinct * from "Clinic1"."medical_card" WHERE not mc_id =
any (SELECT a_patient FROM "Clinic1"."appointment");
```

	mc_id [PK] integer	mc_full_name text	mc_gender boolean	mc_dob date	mc_address text	mc_phone_num integer
1	6	Patient 6	false	1970-04-13	Street 11 home 1...	3482328
2	4	Patient 4	false	2001-02-20	Street 4 home 1 f...	3445119

Рисунок 9 - Результат запроса

9. На рисунке 10 представлена информация об врачах, которые уже принимали пациентов, организованно по возрастанию ID врача.

```
SELECT "d_id" FROM "Clinic1"."doctor" INTERSECT SELECT "a_doc-
tor" FROM "Clinic1"."appointment" ORDER BY d_id;
```

	d_id integer	
1		1
2		2
3		3
4		5
5		6
6		7
7		8

*Рисунок 10 - Результат запроса*

10. На рисунке 11 представлена информация пациентах, которые есть и в таблице “мед\_карта”, и в таблице “прием”.

```
SELECT mc_id FROM "Clinic1"."medical_card" UNION SELECT "a_pa-
tient" FROM "Clinic1"."appointment" ORDER BY "mc_id";
```

mc_id integer	
	1
	2
	3
	4
	5
	6
	7
	8

*Рисунок 11 - Результат запроса*

**Вывод:** в ходе выполнения лабораторной работы №5 были получены практические навыки создания SQL-запросов.