

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

Факультет «Инфокоммуникационных технологий» Направление подготовки
«45.03.04 Интеллектуальные системы в гуманитарной среде»

О Т Ч Е Т

ЛАБОРАТОРНАЯ РАБОТА №4

Запросы

Выполнил:

Студент Смоленская Т.Д. К3243
(Фамилия И.О.) номер группы

Проверил:

Преподаватель Говоров А.И.
(Фамилия И.О.)

Санкт-Петербург
2020

Описание варианта:

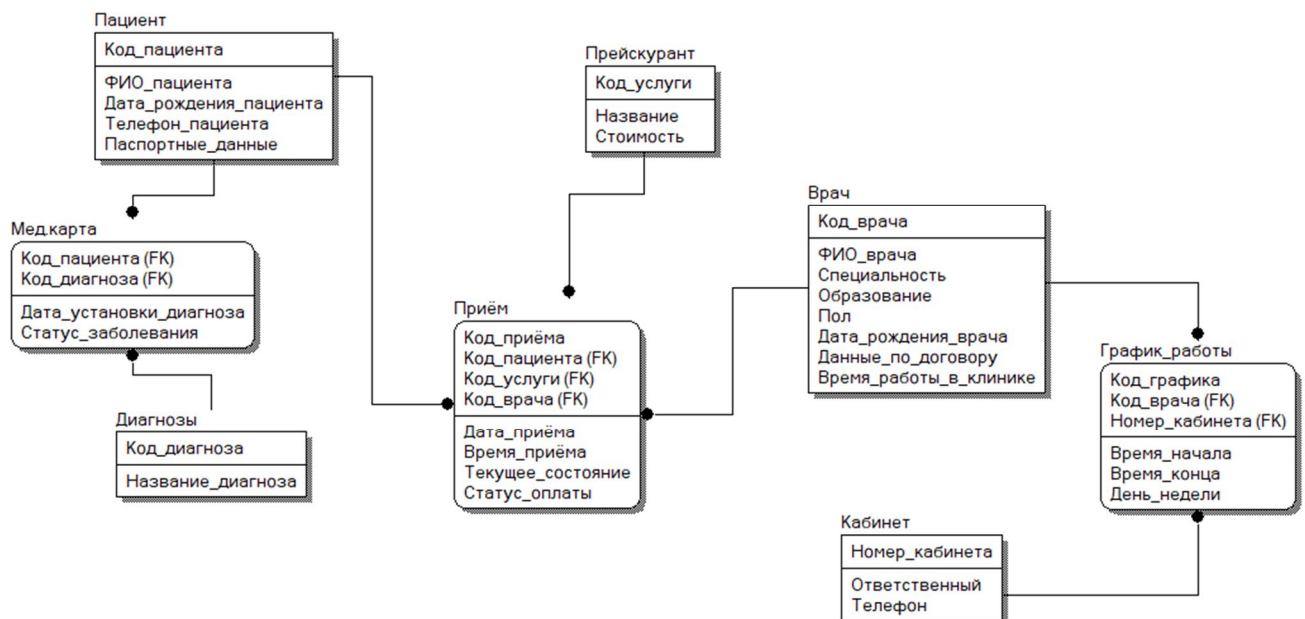
Создать программную систему, предназначенную для администратора лечебной клиники.

Прием пациентов ведут несколько врачей различных специализаций. На каждого пациента клиники заводится медицинская карта, в которой отражается вся информация по личным данным больного и истории его заболеваний (диагнозы). При очередном посещении врача в карте отражается дата и время приема, диагноз, текущее состояние больного, рекомендации по лечению. Так как прием ведется только на коммерческой основе, после очередного посещения пациент должен оплатить медицинские услуги (каждый прием оплачивается отдельно). Расчет стоимости посещения определяется врачом согласно прейскуранту по клинике.

Для ведения внутренней отчетности необходима следующая информация о врач: фамилия, имя, отчество, специальность, образование, пол, дата рождения и дата начала и окончания работы в клинике, данные по трудовому договору. Для каждого врача составляется график работы с указанием рабочих и выходных дней.

Прием пациентов врачи могут вести в разных кабинетах. Каждый кабинет имеет определенный режим работы, ответственного и внутренний телефон.

Модель БД:



Перечень запросов:

Выбор значений с сортировкой

1. Показать ФИО пациентов и даты приёма пациентов с сортировкой по дате приёма:

```
1 SELECT fio_patient, date_meet FROM hospital.patient, hospital.meet
2 WHERE patient.id_patient=meet.id_patient ORDER BY date_meet;
```

	fio_patient text	date_meet date
1	Ivanov I.I.	2019-12-16
2	Antonov I.D.	2020-03-18
3	Mikhailov S.U.	2020-04-28
4	Petrov D.A.	2020-05-16
5	Yurieva N.B.	2020-05-16

2. Показать ФИО докторов, их специальность, начало и конец работы с сортировкой по специальности врача в порядке возрастания:

```
1 SELECT fio_doctor, specialty, time_start, time_end
2 FROM hospital.doctor, hospital.schedule WHERE doctor.id_doctor=schedule.id_doctor
3 ORDER BY specialty ASC;
```

	fio_doctor text	specialty text	time_start time without time zone	time_end time without time zone
1	Frank A.L.	Аллерголог	10:00:00	17:00:00
2	Vanchuk M.T.	Главный врач	09:00:00	19:00:00
3	Tygymbek A.V.	Главный врач	10:00:00	20:00:00
4	Bykov V.D.	Отоларингол...	08:00:00	20:00:00
5	Bibekov V.M.	Пульмонолог	11:00:00	18:30:00

3. Показать ФИО пациентов, их даты установки диагноза и статус оплаты с сортировкой по дате установки диагноза в порядке убывания:

```
1 SELECT fio_patient, date_disease, payment_state
2 FROM hospital.patient, hospital.med_card, hospital.meet
3 WHERE patient.id_patient=med_card.id_patient
4 AND patient.id_patient=meet.id_patient ORDER BY date_disease DESC
```

	fio_patient text	date_disease date	payment_state boolean
1	Petrov D.A.	2020-05-17	true
2	Yurieva N.B.	2020-05-16	true
3	Mikhailov S.U.	2020-04-30	true
4	Antonov I.D.	2020-03-29	true
5	Ivanov I.I.	2019-12-14	true

4. Показать название услуги, стоимость услуги, дату приёма, время приёма и ФИО доктора с сортировкой по стоимости услуги в порядке возрастания:

```
1 SELECT name_service, price_service, date_meet, time_meet, fio_doctor
2 FROM hospital.pricelist, hospital.meet, hospital.doctor
3 WHERE pricelist.id_service=meet.id_service AND meet.id_doctor=doctor.id_doctor
4 ORDER BY price_service ASC
```

Data Output Explain Notifications

	name_service text	price_service integer	date_meet date	time_meet time without time zone	fio_doctor text
1	Консультация	800	2020-05-16	14:20:00	Tygymbek A.V.
2	Первичная консул...	1000	2019-12-16	13:10:00	Bykov V.D.
3	Анализ крови	1300	2020-05-16	09:15:00	Vanchuk M.T.
4	Анализ крови	1300	2020-04-28	16:55:00	Vanchuk M.T.
5	Базовое обследов...	1500	2020-03-18	17:30:00	Bykov V.D.

Использование условий WHERE, состоящих из более, чем одного условия

5. Показать названия услуг, их стоимость, ФИО пациентов и статус оплаты, если стоимость услуги равна больше 1000, а статус оплаты равен true:

```
1 SELECT name_service, price_service, fio_patient, payment_state
2 FROM hospital.pricelist, hospital.patient, hospital.meet
3 WHERE patient.id_patient=meet.id_patient AND pricelist.id_service=meet.id_service
4 AND price_service > 1000 AND payment_state=true
```

Data Output Explain Notifications

	name_service text	price_service integer	fio_patient text	payment_state boolean
1	Базовое обследов...	1500	Antonov I.D.	true
2	Анализ крови	1300	Mikhailov S.U.	true
3	Анализ крови	1300	Petrov D.A.	true

6. Показать ФИО пациентов с названием диагноза «Коронавирус» и статусом заболевания «Обнаружено»:

```
1 SELECT fio_patient, name_disease, status_disease
2 FROM hospital.patient, hospital.disease, hospital.med_card
3 WHERE patient.id_patient=med_card.id_patient
4 AND disease.id_disease=med_card.id_disease AND disease.name_disease='Coronavirus'
5 AND med_card.status_disease='Обнаружено'
```

Data Output Explain Notifications

	fio_patient text	name_disease text	status_disease text
1	Petrov D.A.	Coronavirus	Обнаружено
2	Mikhailov S.U.	Coronavirus	Обнаружено

7. Показать ФИО врачей мужского пола со специальностью «Главный врач»:

```
1 SELECT fio_doctor, gender, specialty FROM hospital.doctor WHERE gender='Мужской'
2 AND specialty='Главный врач'
```

	fio_doctor text	gender text	specialty text	
1	Vanchuk M.T.	Мужской	Главный врач	
2	Tygymbek A.V.	Мужской	Главный врач	

8. Показать ФИО пациентов со статусом приёма «Вылечено» и датой приёма не ранее 2020-01-01:

```
1 SELECT fio_patient, current_state FROM hospital.patient, hospital.meet
2 WHERE current_state='Вылечено' AND date_meet>DATE('2020-01-01')
3 AND patient.id_patient=meet.id_patient
```

	fio_patient text	current_state text	
1	Yurieva N.B.	Вылечено	

Использование функций для работы с датами

9. Показать ФИО пациентов и дату их рождения, если дата рождения меньше 1970-01-01:

```
1 SELECT fio_patient, birthday_date_patient FROM hospital.patient
2 WHERE patient.birthday_date_patient<DATE('1970-01-01')
3
```

	fio_patient text	birthday_date_patient date	
1	Yurieva N.B.	1965-04-05	
2	Antonov I.D.	1956-11-25	

10. Показать ФИО пациентов с датой приёма 2020-05-16:

```
1 SELECT fio_patient, date_meet FROM hospital.patient, hospital.meet
2 WHERE patient.id_patient=meet.id_patient AND date_meet=DATE('2020-05-16')
3
```

	fio_patient text	date_meet date	
1	Petrov D.A.	2020-05-16	
2	Yurieva N.B.	2020-05-16	

Использование строковых функций

11. Показать длину ФИО пациентов в символах:

```
1 SELECT fio_patient, LENGTH("fio_patient") FROM hospital.patient
2
```

	fio_patient text	length integer	
1	Ivanov I.I.	11	
2	Petrov D.A.	11	
3	Mikhailov S.U.	14	
4	Yurieva N.B.	12	
5	Antonov I.D.	12	
6	Valichev O.E.	13	

12. Привести строку в верхний регистр:

```
1 SELECT name_disease, UPPER("name_disease") FROM hospital.disease
```

	name_disease text	upper text	
1	OP3	OP3	
2	Coronavirus	CORONAV...	
3	Pneumonia	PNEUMO...	
4	Asthma	ASTHMA	
5	Influenza	INFLUENZA	

Использование подзапросов

13. Показать ФИО пациентов с таким кодом пациента из медкарты, когда код болезни равен 2:

```
1 SELECT fio_patient FROM hospital.patient WHERE patient.id_patient IN
2 (SELECT med_card.id_patient FROM hospital.med_card WHERE med_card.id_disease=2)
```

	fio_patient text	
1	Petrov D.A.	
2	Mikhailov S.U.	
3	Yurieva N.B.	

14. Показать фамилию доктора с таким кодом доктора, когда код пациента равен 14:

```

1 SELECT fio_doctor FROM hospital.doctor WHERE doctor.id_doctor IN
2 (SELECT meet.id_doctor FROM hospital.meet WHERE id_patient=14)

```

	fio_doctor	
1	Tygymbek A.V.	

Использование групповой (агрегатной) функции

15. Посчитать количество пациентов с определённым кодом диагноза и сгруппировать по коду диагноза:

```

1 SELECT id_disease, COUNT(id_patient) FROM hospital.med_card
2 GROUP BY id_disease

```

	id_disease	count	
1	3	1	
2	5	1	
3	2	3	

16. Показать максимальную стоимость из прейскуранта:

```

1 SELECT max(price_service) FROM hospital.pricelist

```

	max	
1	3200	

Использование групповой (агрегатной) функции с условием HAVING

17. Посчитать количество докторов с временем работы в клинике равном «26 years», сгруппированных по времени работы в клинике:

```

1 SELECT work_time_in_clinic, COUNT(fio_doctor) FROM hospital.doctor
2 GROUP BY work_time_in_clinic HAVING work_time_in_clinic='26 years'

```

	work_time_in_clinic	count	
1	26 years	1	

18. Показать названия услуг с минимальной ценой больше 1000, сгруппированных по названию услуги:

```
1 SELECT name_service, MIN(price_service) AS Minimum
2 FROM hospital.pricelist GROUP BY name_service
3 HAVING MIN(price_service)>1000
```

	name_service text	minimum integer
1	Анализ крови	1300
2	Полное обследова...	3200
3	MPT	1200
4	Базовое обследов...	1500

Использование предикатов EXISTS, ALL, SOME и ANY

19. Показать коды пациентов, которых принимал доктор с кодом доктора равному 11 и у которых статус оплаты равен true:

```
1 SELECT id_patient FROM hospital."meet"
2 WHERE id_doctor = 11 AND
3 EXISTS (SELECT * FROM hospital."meet" WHERE payment_state = true)
```

	id_patient integer
1	11
2	15

20. Показать название диагноза, дату установки диагноза с кодом диагноза равному 2 и датой установки диагноза не ранее 2020-04-01:

```
1 SELECT name_disease, date_disease FROM hospital.disease, hospital.med_card
2 WHERE disease.id_disease=2 AND disease.id_disease=med_card.id_disease
3 AND EXISTS (SELECT * FROM hospital.med_card WHERE date_disease>DATE('2020-04-01'))
```

	name_disease text	date_disease date
1	Coronavirus	2020-05-17
2	Coronavirus	2020-04-30
3	Coronavirus	2020-05-16

Вывод: К базе данных «hospital» было совершено 20 запросов разного типа и с разным уровнем сложности. В ходе лабораторной работы был изучен основной функционал языка SQL, позволяющий выводить основные значения и данные базы данных для наших целей.