

210103331 NURAI ABUGALIEVA
210103244 NURTILEU PAIZKHAN
210103169 ORZHANOV NAURYZBEK



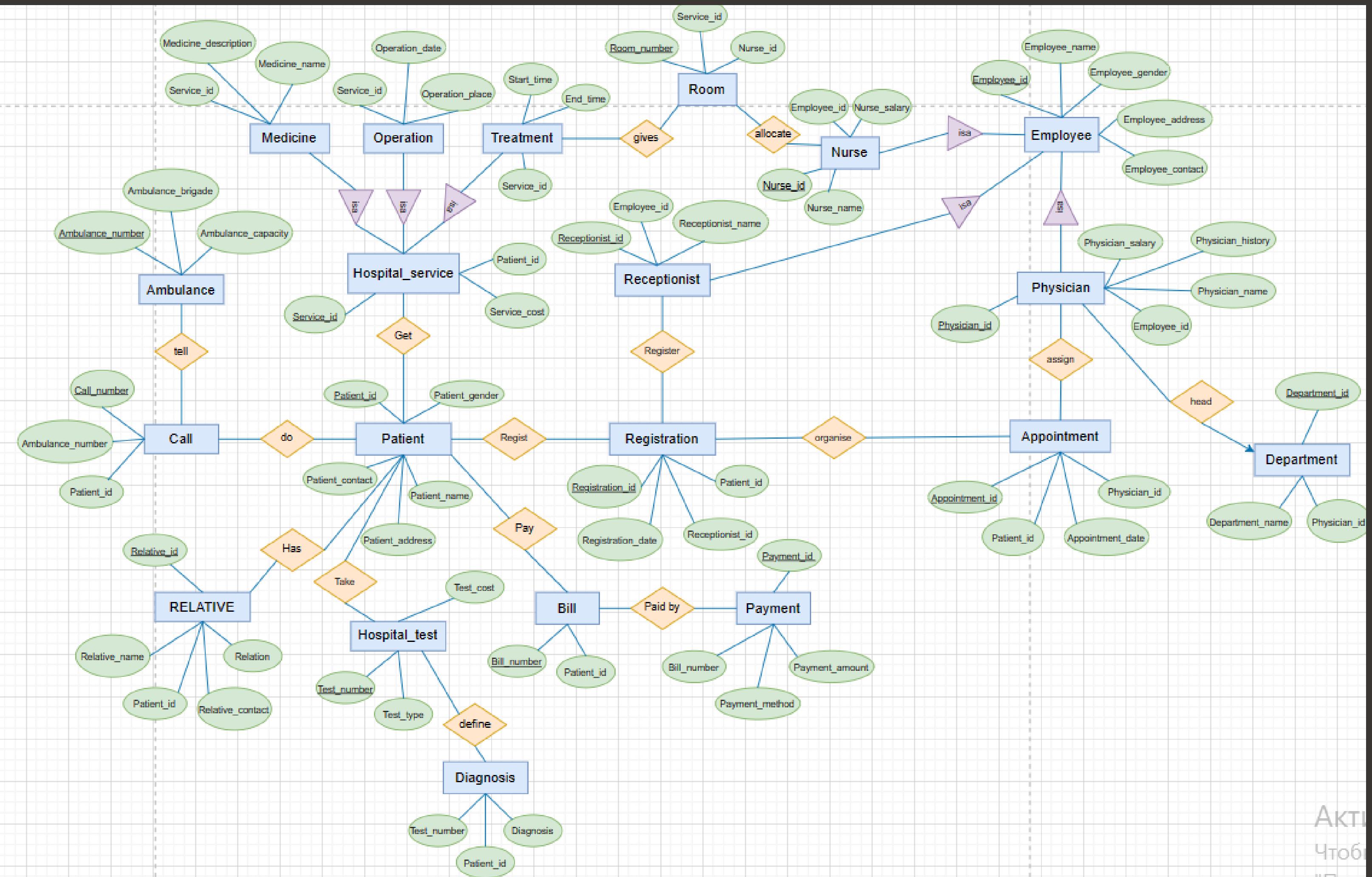
HOSPITAL DATABASE MANAGEMENT

INTRODUCTION

- This system helps to find a quick access to any information such as patients, employees, etc. It produces monthly reports ,controls data redundancy,helps patient to get the best service,calculates every bill.A user can search status of each room ,availability of doctors and other services easily.
- Any authorized user can add, delete and update data into the database base on their user-assigned-role.
- Data obsolescence can be handled by obsolete inventory write-offs , often charge obsolete at the end of the year - taking the loss and moving forward

REFERENCES:

- SQL Exercises/The Hospital - Wikibooks, open books for an open world
- https://www.researchgate.net/publication/330316043_Design_and_Implementation_of_a_Hospital_Database_Management_System_HDMS_for_Medical_Doctors



Functional Dependencies

1) Patient	Physician table is a subclass of Employee table	{Service_id} foreign key	{Patient_id} foreign key
	{Patient_id} → {Patient_name}, {Patient_address}, {Patient_contact}, {Patient_gender}		
	{Patient_id} key		
	{Patient_id}, {Patient_address}, {Patient_contact} → {Patient_name}, {Patient_gender}		
	{Patient_id}, {Patient_address}, {Patient_contact} superkey		
2) Employee			
	{Employee_id} → {Employee_name}, {Employee_address}, {Employee_contact}, {Employee_gender}		
	{Employee_id} key		
	{Employee_id}, {Employee_address}, {Employee_contact}, ->{Employee_name}, {Employee_gender}		
	{Employee_id}, {Employee_address}, {Employee_contact} superkey		
3) Nurse			
	{Nurse_id}, {Employee_id} → {Nurse_name}, {Nurse_salary}		
	{Nurse_id} key		
	{Employee_id} foreign key		
	Nurse table is a subclass of Employee table		
4) Receptionist			
	{Receptionist_id}, {Employee_id} → {Receptionist_name}		
	{Receptionist_id} key ; {Employee_id} foreign key		
	Receptionist table is a subclass of Employee table		
5) Physician			
	{Physician_id}, {Employee_id} → {Physician_name}, {Physician_salary}, {Physician_history}		
	{Physician_id} key		
	{Employee_id} foreign key		
6) Department			
	{Department_id}, {Physician_id} → {Department_name}		
	{Department_id} key		
	{Physician_id} foreign key		
7) Hospital_service			
	{Service_id}, {Patient_id} → {Service_cost}		
	{Service_id} key		
	{Patient_id} foreign key		
8) Treatment			
	{Service_id} → {Start_time}, {End_time}		
	{Service_id} foreign key		
	Treatment table is a subclass of Hospital_service table		
9) Medicine			
	{Service_id} > {Medicine_name}, {Medicine_description}		
	{Service_id} foreign key		
	{Medicine_name} → {Medicine_description}		
	Medicine table is a subclass of Hospital_service table		
10) Operation			
	{Service_id} → {Operation_date}, {Operation_place}		
	{Service_id} foreign key		
11) Room			
	{Service_id}, {Nurse_id} → {Room_number}		
12) RELATIVE			
	{Relative_id}, {Patient_id} → {Relative_name}, {Relation}, {Relative_contact}		
	{Patient_id} foreign key ; {Relative_id} key		
	{Patient_id}, {Relative_id}, {Relative_contact} ->{Relative_name}, {Relation}		
	{Patient_id}, {Relative_id}, {Relative_contact} superkey		
13) Appointment			
	{Appointment_id}, {Patient_id}, {Physician_id} → Appointment_date		
	{Patient_id} foreign key		
	{Physician_id} foreign key		
	{Appointment_id} key		
14) Ambulance			
	{Ambulance_number} → {Ambulance_brigade}, {Ambulance_capacity}		
	{Ambulance_number} key		
15) Call			
	{Call_number} → {Ambulance_number}, {Patient_id}		
	{Call_number} key		
	{Ambulance_number} foreign key		
16) Hospital_test			
	{Test_number} → {Test_type}, {Test_cost}		
	{Test_number}, {Test_type} → {Test_cost}		
	{Test_number}, {Test_type} superkey		
	{Test_number} key		
17) Diagnosis			
	{Test_number}, {Patient_id} → {Diagnosis}		
	{Test_number} foreign key		
	{Patient_id} foreign key		
18) Bill			
	{Patient_id} <-> {Bill_number}		
	{Patient_id} foreign key		
19) Payment			
	{Payment_id}, {Bill_number} → {Payment_method}, {Payment_amount}		
	{Bill_number} foreign key; {Payment_id} key		
20) Registration			
	{Registration_id}, {Patient_id}, {Receptionist_id} → {Registration_date}		
	{Patient_id} foreign key		
	{Receptionist_id} foreign key		
	{Registration_id} key		

1) Finding the name of all patients who got coronavirus in 2020

$$R := \prod_{P_name(P)} \prod_{P.P_id=R.P_id} \prod_{R_date(R)} \left(\sigma_{R_date \text{ like } \%2020\%} (R) \right) \prod_{R.P_id=D.P_id} \prod_{D.D=\text{coronavirus}(D)}$$

```

1 SELECT Patient_name, Diagnosis, Registration_date FROM Patient INNER JOIN Diagnosis ON Diagnosis.Patient_id=Patient.Patient_id INNER JOIN Registration ON
2 | Registration.Patient_id=Patient.Patient_id WHERE Registration_date like '%2020' and diagnosis='coronavirus' ;

```

PATIENT_NAME	DIAGNOSIS	REGISTRATION_DATE
Aikerim	coronavirus	10/07/2020

2) creating a view to better understand all payment costs

3) finding the total payment amount with bill for patients

$$R := \prod_{B_number, SUM(T_cost + S_cost) \rightarrow TOTAL} (Paid)$$

```

1 CREATE VIEW Paid AS SELECT Bill_number, Test_cost, Service_cost, Patient_name FROM Hospital_test INNER JOIN Diagnosis ON Hospital_test.Test_number=Diagnosis.Test_number INNER JOIN
2 Hospital_service ON Hospital_service.Patient_id=Diagnosis.Patient_id LEFT JOIN Bill ON Bill.Patient_id=Diagnosis.Patient_id RIGHT JOIN Patient ON Patient.Patient_id=Bill.Patient_id;
3 | SELECT Bill_number ,SUM(Test_cost+Service_cost) AS Total FROM Paid GROUP BY Bill_number

```

BILL_NUMBER	TOTAL
2	105000
10	15000

4) finding the names of patients who got both of medicine and operation service;

$$R := \prod_{P_name(P)} \prod_{H_service.P_id=P.P_id} \prod_{H_service.S_id=O.S_id} \cap \prod_{P_name(P)} \prod_{H_service.P_id=P.P_id} \prod_{H_service.M_id=M.S_id}$$

```

1 SELECT Patient_name FROM Patient LEFT JOIN Hospital_service ON Hospital_service.Patient_id=Patient.Patient_id INNER JOIN Operation ON Hospital_service.Service_id=Operation.Service_id
2 INTERSECT SELECT Patient_name FROM Patient LEFT JOIN Hospital_service ON Hospital_service.Patient_id=Patient.Patient_id INNER JOIN Medicine ON
3 Hospital_service.Service_id=Medicine.Service_id GROUP BY Patient_name ;

```

PATIENT_NAME
Aikerim
Almira

5) finding the names of physicians who have salary more than average salary of them

$$R := \prod_{P_name, P_salary} (\sigma_{P_salary > \text{AVG}(P_salary)}(P))$$

```

1 SELECT Physician_name, Physician_salary FROM Physician WHERE Physician_salary > (SELECT AVG(Physician_salary) FROM Physician);

```

PHYSICIAN_NAME	PHYSICIAN_SALARY
Salikhan	1000000

6) finding the name of nurses who had looked after patients with cancer disease

$$R := \prod_{N_name(N)} \prod_{N.N_id=R.N_id} \prod_{N_id(R)} \prod_{H_service.H_service.S_id=R.S_id} \prod_{H_service.H_service.P_id=H.service.P_id} \prod_{D.D=cancer(D)}$$

```

1 SELECT Nurse_name FROM Nurse WHERE Nurse_id = ANY (SELECT Nurse_id FROM Room INNER JOIN Hospital_service ON Hospital_service.Service_id=Room.Service_id INNER JOIN Diagnosis ON
2 Diagnosis.Patient_id=Hospital_service.Patient_id WHERE Diagnosis='cancer') ORDER BY Nurse_name;

```

NURSE_NAME
Akhmet
Sabit

7) finding the names of patients who got medicine or treatment service;

$$R := \prod_{P_name(P)} \prod_{H_service.P_id=P.P_id} \prod_{H_service.S_id=T.S_id} \cup \prod_{P_name(P)} \prod_{H_service.P_id=P.P_id} \prod_{H_service.M_id=M.S_id}$$

```

1 SELECT Patient_name FROM Patient INNER JOIN Hospital_service ON Hospital_service.Patient_id=Patient.Patient_id INNER JOIN Treatment ON Hospital_service.Service_id=Treatment.Service_id
2 UNION SELECT Patient_name FROM Patient INNER JOIN Hospital_service ON Hospital_service.Patient_id=Patient.Patient_id INNER JOIN Medicine ON
3 Hospital_service.Service_id=Medicine.Service_id GROUP BY Patient_name ;

```

PATIENT_NAME
Abelard

8) finding the number of patients who had take blood typed test

$$R := \text{COUNT}(P_id)(D) \prod_{H_test, T_number=D.T_number} \sigma_{T_type=blood, H_test}$$

```

1 SELECT COUNT(Patient_id) FROM Diagnosis RIGHT JOIN Hospital_test ON Hospital_test.Test_number=Diagnosis.Test_number WHERE Test_type='blood' ;

```

COUNT(PATIENT_ID)
10

9) creating a view for showing patients who had operation abroad

10) finding a patients who had operation in Ankara from view

$$R := \sigma_{place \text{ like } \%Ankara\%, (abroad_operation)}$$

```

1 CREATE VIEW Abroad_operation AS SELECT Patient_id, Operation_place FROM Hospital_service INNER JOIN Operation ON Hospital_service.Service_id=Operation.Service_id WHERE
2 Operation_place NOT LIKE '\%Z\%';
3 | SELECT * FROM Abroad_operation WHERE Operation_place LIKE '\%Ankara\%' ;

```

PATIENT_ID	OPERATION_PLACE
108	TK,Ankara

11)If there will be new hearing test, eye test will cost twice as much

TEST NUMBER				TEST TYPE	TEST COST
	6010			hearing test	13000
	6100			eye test	26000

PAYMENT_ID	BILL_NUMBER	PAYMENT_METHOD	PAYMENT_AMOUNT
2300	1	insurance	100000
2301	2	card	700000
2302	3	insurance	1000000
2303	4	card	70000
2304	5	cash	9000
2306	7	insurance	5000
2308	8	cash	10000
2307	10	insurance	400000
2309	9	cash	900000

13)if there have some physician whose salary increased to 1000000, 11 year worked physician's salary will increase two times

PHYSICIAN_ID	PHYSICIAN_NAME	PHYSICIAN_SALARY	PHYSICIAN_HISTORY
1500	Askar	800000	6
1501	Askar	800000	6
1502	Alisher	800000	5
1503	Salikhan	1000000	2
1504	Alina	800000	6
1505	Ainur	800000	7
1506	Askarbek	2000000	11

12)if the insurance will be cancelled ,it should automatically paid by cash

14) Finding patients who had treatment between '09-30-2020' AND '11-11-2020

$$R := \prod_{P_id} (H_service) \bowtie_{H_service.S_id = T.S_id} (\sigma_{start_t = '09-30-2020' \text{ AND } end_t = '11-11-2020'} (T))$$

```

1 SELECT Patient_id FROM Hospital_service WHERE EXISTS (SELECT Start_time, End_time FROM Treatment WHERE
2 Hospital_service.Service_id = Treatment.Service_id AND Start_time = '09-30-2020' AND End_time = '11-11-2020');

```

Results	Explain	Describe	Saved SQL	History
PATIENT_ID				
102				

15) finding the nurses who have the same name with receptionists

$$R := \prod_{N_name} (\sigma_{N_name = R.name} (\prod_{R_name}(R) \bowtie_{R.emp_id = N.emp_id} \prod_{N_name}(N)))$$

```

1 SELECT Nurse_name FROM Nurse WHERE Nurse_name IN (SELECT Receptionist_name FROM Receptionist);

Results Explain Describe Saved SQL History

```

NURSE_NAME
Sabit

16) finding the number of doctors depending on their experience

$$R := \prod_{P_history, \exists \text{count}(P_id) > 1} (P)$$

```

1 SELECT COUNT(Physician_id), Physician_history FROM Physician
2 GROUP BY Physician_history HAVING COUNT(Physician_id) > 1;

Results Explain Describe Saved SQL History

```

COUNT(PHYSICIAN_ID)	PYHICIAN_HISTORY
6	
7	
5	

17) finding for all patients living in Almaty order by Patient name

$$R := \tau_{P_name} (\sigma_{P_address \text{ like } \% \text{Almaty}\%} (Patient))$$

```

1 SELECT * FROM Patient WHERE Patient_address LIKE '%Almaty%' ORDER BY Patient_name;

Results Explain Describe Saved SQL History

```

PATIENT_ID	PATIENT_NAME	PATIENT_ADDRESS	PATIENT_CONTACT	PATIENT_GENDER
100	Aikerim	Almaty, 10th microdistrict	87783422345	female
102	Alina	Almaty, 9th microdistrict	87784556989	female

18) finding the responsible doctors of each department

$$R := \prod_{P_name}(P) \bowtie_{P.P_id = D.P_id} \prod_{D_name}(D)$$

```

1 SELECT Physician_name, Department_name FROM Physician RIGHT JOIN Department ON Department.Physician_id = Physician.Physician_id;

Results Explain Describe Saved SQL History

```

PHYSICIAN_NAME	DEPARTMENT_NAME
Askar	Pediatrics
Askar	Cardiology

19) finding all patients who came with a help of ambulance

$$R := \prod_{P_name} (\sigma_{P.P_id = Call.P_id} (\prod_{P_name}(P) \bowtie_{P.P_id = Call.P_id} Call))$$

```

1 SELECT Patient_name FROM Patient WHERE Patient_id = ANY(SELECT Patient_id FROM Call);

Results Explain Describe Saved SQL History

```

PATIENT_NAME
Aikerim
Damir

20) finding the patients who got used notsep medicine

$$R := \prod_{P_id, M_name} (\prod_{P_id}(P) \bowtie_{H_service.S_id = M.S_id} (\sigma_{M.name = 'Notsep'} (M)))$$

```

1 SELECT Patient_id, Medicine_name FROM Hospital_service INNER JOIN Medicine ON
2 Hospital_service.Service_id = Medicine.Service_id WHERE Medicine_name = 'Notsep';

Results Explain Describe Saved SQL History

```

PATIENT_ID	MEDICINE_NAME
102	Notsep
103	Notsep

ALTER TABLE Patient ADD Patient_age
ALTER table Patient drop column Patient_age

TRANSACTION

```

1 START TRANSACTION;
2 SELECT Ambulance_number,Ambulance_brigade FROM Ambulance WHERE Ambulance_capacity='big';
3 SELECT Patient_name,Relative_name FROM Patient INNER JOIN Relative ON Relative.Patient_id=Patient.Patient_id ;
4 COMMIT;

```

История | Формат | Получить автосохранённый запрос

Связать параметры

Создание закладки SQL запроса:

	Ambulance_number	Ambulance_brigade	Помощь
	8000	А	
	Patient_name	Relative_name	Помощь
	Damir	Aikyn	
	Alina	Aiaulym	

1)Two selection query at the same time,which one of them shows patients with relatives,second shows ambulance information with big capacity

```

1 START TRANSACTION;
2 INSERT INTO Patient VALUES(120,'Aisultan','Almaty,10th microdistrict',87783422347,'male');
3 DELETE FROM PATIENT WHERE Patient_id=100;
4 DELETE FROM Hospital_service WHERE Patient_id=100;
5 COMMIT;

```

Очистить | Формат | Получить автосохранённый запрос

Связать параметры

Создание закладки SQL запроса:

Разделитель : | Показать данный запрос снова | Оставить поле запроса | Откат после

Скрыть поле запроса

MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0,0001 сек.)

2)While someone inserting into patient values,other deleting Patient with id 100 from patient and hospital_service tables

```

1 START TRANSACTION;
2 UPDATE Payment SET Payment_method='cash' WHERE Bill_number=6;
3 UPDATE Relative SET Relation='brother' WHERE Relative_name='Alpamys';
4 COMMIT;
5

```

SELECT * | SELECT | INSERT | UPDATE | DELETE | Очистить | Формат

Связать параметры

Создание закладки SQL запроса:

Разделитель : | Показать данный запрос снова | Оставить поле запроса

MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0,0001 сек.)

3)updating at the same time Payment and Relative tables

INDEX

```

1 SELECT Patient_name,Diagnosis,Registration_date FROM Patient INNER JOIN Diagnosis ON Diagnosis.Patient_id=Patient.Patient_id INNER JOIN
2 Registration ON Registration.Patient_id=Patient.Patient_id WHERE
3 Registration_date like '%2020' and diagnosis='coronavirus' ;

```

Results | Explain | Describe | Saved SQL | History

PATIENT_NAME	DIAGNOSIS	REGIS
Aikerim	coronavirus	10/07/2020

1 rows returned in 0.31 seconds | Download

before indexing

```

1 SELECT COUNT(Patient_id) FROM Diagnosis RIGHT JOIN Hospital_test ON Hospital_test.Test_number=Diagnosis.Test_number WHERE Test_type='blood' ;
2

```

Results | Explain | Describe | Saved SQL | History

COUNT(PATIENT_ID)
10

1 rows returned in 0.19 seconds | Download

```

1 SELECT Patient_name,Diagnosis,Registration_date FROM Patient INNER JOIN Diagnosis ON Diagnosis.Patient_id=Patient.Patient_id INNER JOIN
2 Registration ON Registration.Patient_id=Patient.Patient_id WHERE
3 Registration_date like '%2020' and diagnosis='coronavirus' ;
4 CREATE INDEX Reg_index ON Registration(Registration_date);
5 CREATE INDEX Diag_index ON Diagnosis(Diagnosis);

```

Results | Explain | Describe | Saved SQL | History

PATIENT_NAME	DIAGNOSIS	REGIS
Aikerim	coronavirus	10/07/2020

1 rows returned in 0.00 seconds | Download

after

```

1 SELECT COUNT(Patient_id) FROM Diagnosis RIGHT JOIN Hospital_test ON Hospital_test.Test_number=Diagnosis.Test_number WHERE Test_type='blood' ;
2 CREATE INDEX test_index ON Hospital_test(Test_type);
3

```

Results | Explain | Describe | Saved SQL | History

COUNT(PATIENT_ID)
10

1 rows returned in 0.02 seconds | Download