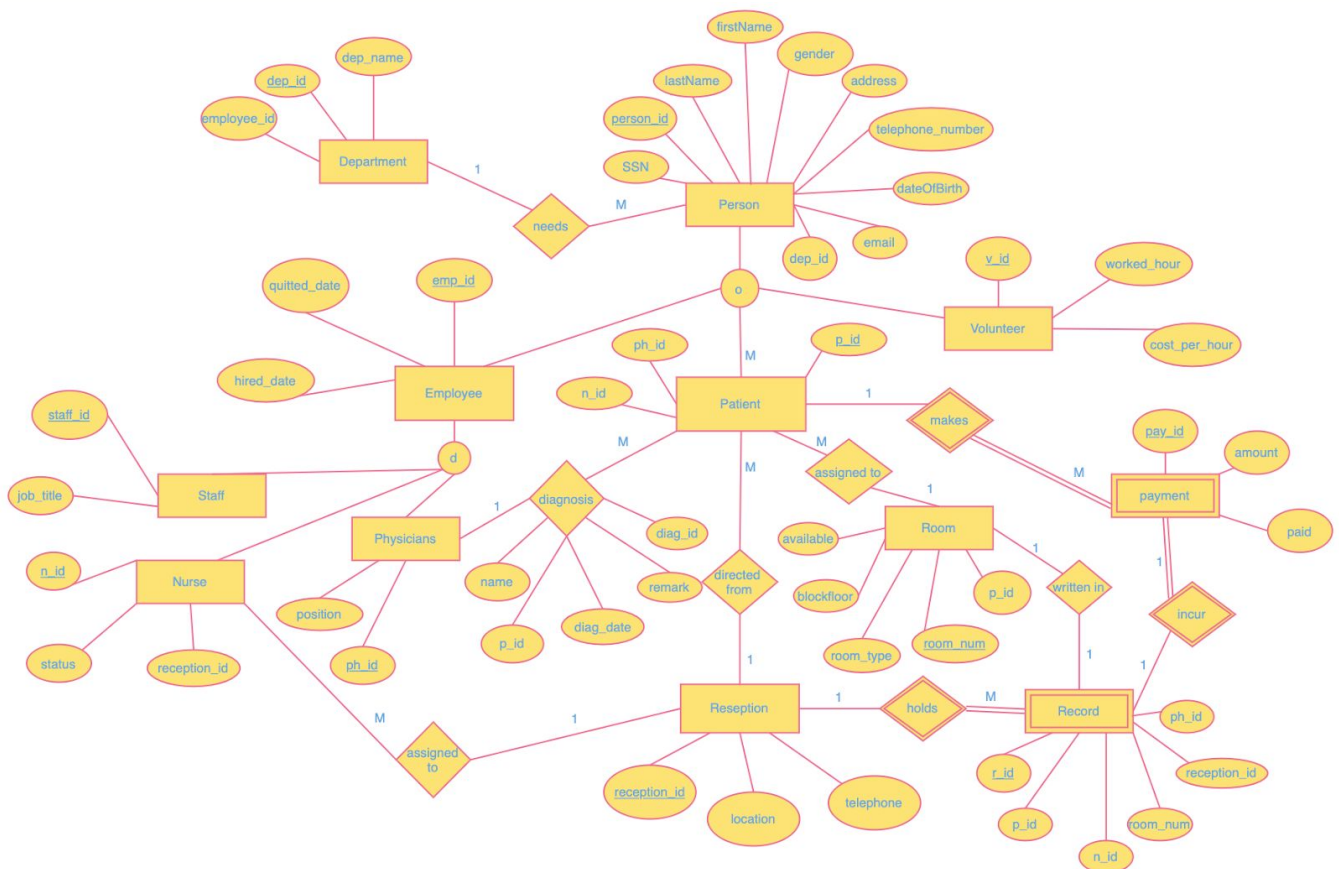


Nazerke Kulan 17-p
 Arailym Ospankhan 14-p
 Dिल्fina Kazhiyeva 18-p
 Assel Yembergenova 14-p

TASK 1 (5 points)

We have implemented a database structure to support the basic functions of a hospital: the Hospital consists of several departments which specialize in their sphere and need persons to be included in it. Our database keeps track of information about their employees at different positions and volunteers who held internships at this hospital and of course our patients who need treatment. We have nurses, physicians and staff as one of the employees. You may propose reasonable extensions to this basic set of functions, such as tracking the record for each patient visit to reception, and departmental information for doctors and nurses indicating their respective specializations. Physicians make diagnoses for patients for the treatment and for assigned rooms from which payments are comes out.

TASK 2 (5 points)



In our Hospital database there are 12 entities in total, where 10 of them are strong: *DEPARTMENT, PERSON, EMPLOYEE, STAFF, NURSE, PHYSICIAN, RECEPTION, PATIENT, VOLUNTEER, ROOM* and other 2 entities: "payment", "record" are weak entities, where they cannot exist without the Patient and Reception entities. And we have complex relation *Diagnoses* between patients and Physicians.

And about the *cardinality* between entities:

One department needs many people to be included in it.

Many nurses are assigned to only one reception office.

One or more PATIENT must refer to one PHYSICIAN.

Many patients can be assigned to one room.

One Reception office holds many records about patients.

One Patient can incur many payments.

There is one record for one room.

Many patients are directed from one reception office.

Also the diagram includes a *hierarchical model*, where data is organized into a tree like structure with each record having one parent record and many children.

And in our hierarchical model there are both types of specialization used.

Person and its subclasses are connected with *overlapping* constraints. Person is a supertype-entity and it has three subclasses: Employee, Volunteer and Patient connected with overlapping constraints where an entity instance of a supertype can be a member of multiple subtypes. For example Doctor can be the Patient at the same time.

And our Employee entity uses *disjoint* constraint. Employee is a super-type and Staff, Nurse and Physician are its subclasses, where instance can only be a member of one subtype. For example, one Employee cannot be Staff, Nurse and Physician simultaneously.

TASK 3 (20 points)

```
CREATE DATABASE HOSPITAL;
```

```
CREATE TABLE Department(  
  dep_id int PRIMARY KEY,  
  dep_name varchar(256) NOT NULL  
);
```

```
CREATE TABLE Person(  
  person_id int PRIMARY KEY,  
  SSN int UNIQUE NOT NULL,  
  lastName varchar(50) NOT NULL,  
  firstName varchar(50) NOT NULL,  
  gender varchar(10) NOT NULL,  
  address varchar(50),  
  telephone_number char(11),  
  email varchar(50),  
  dateOfBirth datetime,  
  dep_id int FOREIGN KEY REFERENCES Department(dep_id),  
  emp_id int,  
  p_id int,  
  v_id int  
);
```

```
CREATE TABLE Employee(  
  emp_id int PRIMARY KEY,  
  hired_date datetime NOT NULL,  
  quitted_date datetime,  
  staff_id int,  
  n_id int,  
  ph_id int  
);
```

```
CREATE TABLE Staff(  
  staff_id int PRIMARY KEY,  
  job_title varchar(50) NOT NULL  
);
```

```
CREATE TABLE Nurse(  
  n_id int PRIMARY KEY,  
  status varchar(30) NOT NULL,  
  reception_id int  
);
```

```

CREATE TABLE Physician(
ph_id int PRIMARY KEY,
position varchar(30) NOT NULL
);

CREATE TABLE Patient(
p_id int PRIMARY KEY,
ph_id int FOREIGN KEY REFERENCES Physician(ph_id),
pay_id int,
r_id int,
diag_id int
);

CREATE TABLE Volunteer(
v_id int PRIMARY KEY,
worked_hour int NOT NULL,
cost_per_hour int NOT NULL
);

CREATE TABLE Room(
room_num varchar(10) PRIMARY KEY,
room_type varchar(20) NOT NULL,
blockfloor int NOT NULL,
available char(1) NOT NULL
);

CREATE TABLE Reception(
reception_id int PRIMARY KEY,
telephone char(5) NOT NULL,
location varchar(50) NOT NULL
);

CREATE TABLE Payment(
pay_id int PRIMARY KEY,
amount int NOT NULL,
paid char(1) NOT NULL
);

CREATE TABLE Record(
r_id int PRIMARY KEY,
n_id int FOREIGN KEY REFERENCES Nurse(n_id),
room_num varchar(10) FOREIGN KEY REFERENCES Room(room_num)
);

```

```
CREATE TABLE Diagnose(
diag_id int PRIMARY KEY,
name varchar(50) NOT NULL,
remark varchar(50) NOT NULL,
date datetime NOT NULL
);
```

--In some table we couldn't add our foreign key straightaway (because we still didn't create other tables) that's why we are adding them by altering the table

```
ALTER TABLE Person
ADD FOREIGN KEY (emp_id) REFERENCES Employee(emp_id),
    FOREIGN KEY (p_id) REFERENCES Patient(p_id),
    FOREIGN KEY (v_id) REFERENCES Volunteer(v_id);
```

```
ALTER TABLE Employee
ADD FOREIGN KEY (staff_id) REFERENCES Staff(staff_id),
    FOREIGN KEY (n_id) REFERENCES Nurse(n_id),
    FOREIGN KEY (ph_id) REFERENCES Physician(ph_id);
```

```
ALTER TABLE Nurse
ADD FOREIGN KEY (reception_id) REFERENCES Reception(reception_id);
```

```
ALTER TABLE Patient
ADD FOREIGN KEY (pay_id) REFERENCES Payment(pay_id),
    FOREIGN KEY (r_id) REFERENCES Record(r_id),
    FOREIGN KEY (diag_id) REFERENCES Diagnose(diag_id);
```

```
INSERT INTO Reception
VALUES
    (900, '72735', 'first floor'),
    (901, '31454', 'second floor'),
    (902, '84394', 'third floor'),
    (903, '32343', 'fourth floor'),
    (904, '71734', 'fifth floor'),
    (905, '73519', 'first floor'),
    (906, '48384', 'second floor'),
    (907, '32434', 'fifth floor'),
    (908, '72134', 'ninth floor'),
    (909, '34597', 'first floor');
```

```

INSERT INTO Nurse
VALUES
    (500, 'nurse', 900),
    (501, 'nurse', 900),
    (502, 'nurse', 902),
    (503, 'head nurse', 903),
    (504, 'nurse', 902),
    (505, 'nurse', 900),
    (506, 'nurse', 906),
    (507, 'head nurse', 904),
    (508, 'nurse', 904),
    (509, 'nurse', 904);

```

```

INSERT INTO Room
VALUES
    ('101A', 'single', 3, '1'),
    ('102A', 'single', 8, '0'),
    ('103A', 'double', 6, '1'),
    ('104A', 'triple', 9, '0'),
    ('105A', 'single', 10, '1'),
    ('201A', 'double', 5, '1'),
    ('202A', 'triple', 1, '0'),
    ('203A', 'quad', 2, '0'),
    ('204A', 'triple', 4, '1'),
    ('205A', 'single', 7, '1');

```

```

INSERT INTO Staff
VALUES
    (300, 'cleaner'),
    (301, 'watchman'),
    (302, 'cleaner'),
    (303, 'chef'),
    (304, 'security_guard'),
    (305, 'cleaner'),
    (306, 'security_guard'),
    (307, 'watchman'),
    (308, 'cleaner'),
    (309, 'security_guard');

```

```

INSERT INTO Physician
VALUES
    (400, 'Staff Internist'),
    (401, 'Attending Physician'),
    (402, 'Surgical Attending Physician'),

```

```

(403, 'Senior Attending Physician'),
(404, 'Head Chief of Medicine'),
(405, 'Surgical Attending Physician'),
(406, 'Surgical Attending Physician'),
(407, 'MD Resident'),
(408, 'MD Resident'),
(409, 'Staff Internist');

```

INSERT INTO Payment

VALUES

```

(200, 10000, '1'),
(201, 80000, '0'),
(202, 150000, '1'),
(203, 23000, '1'),
(204, 5000, '0'),
(205, 60000, '0'),
(206, 40000, '1'),
(207, 40000, '0'),
(208, 50000, '1'),
(209, 80000, '1');

```

INSERT INTO Record

VALUES

```

(1000, 500, '101A'),
(1001, 500, '102A'),
(1002, 501, '101A'),
(1003, 503, '102A'),
(1004, 503, '101A'),
(1005, 504, '105A'),
(1006, 500, '201A'),
(1007, 502, '202A'),
(1008, 502, '203A'),
(1009, 505, '101A');

```

INSERT INTO Diagnose

VALUES

```

(800, 'hypertension', 'undetermined', '2019-09-23'),
(801, 'hyperlipidemia', 'serious', '2020-06-19'),
(802, 'diabetes', 'critical', '2011-01-27'),
(803, 'back pain', 'stable', '2016-09-30'),
(804, 'anxiety', 'undetermined', '2015-10-12'),
(805, 'obesity', 'serious', '2019-04-28'),
(806, 'allergic rhinitis', 'undetermined', '2018-07-07'),
(807, 'reflux esophagitis', 'critical', '2009-12-18'),
(808, 'pain in joint', 'serious', '2018-11-30'),
(809, 'asthma', 'critical', '2018-11-30');

```

```
INSERT INTO Volunteer
VALUES
```

```
(30, 2000, 30),
(31, 3000, 48),
(32, 1500, 150),
(33, 2600, 65),
(34, 2500, 34),
(35, 1300, 65),
(36, 2000, 23),
(37, 2500, 43),
(38, 1200, 423),
(39, 2000, 422);
```

```
INSERT INTO Patient
VALUES
```

```
(600, 400, 200, 1000, 800),
(601, 400, 201, 1001, 801),
(602, 402, 202, 1002, 802),
(603, 403, 203, 1003, 803),
(604, 403, 204, 1004, 804),
(605, 400, 205, 1005, 805),
(606, 400, 206, 1006, 806),
(607, 407, 207, 1007, 807),
(608, 402, 208, 1008, 808),
(609, 402, 209, 1009, 809);
```

```
INSERT INTO Employee
VALUES
```

```
(100, '2015-12-09', '2020-12-09', NULL, 503, NULL),
(101, '2011-02-09', '2019-12-30', NULL, NULL, 401),
(102, '2005-12-09', NULL, NULL, 503, NULL),
(103, '2018-10-09', '2019-12-13', 301, NULL, NULL),
(104, '2002-02-19', NULL, NULL, 505, NULL),
(105, '2008-10-19', NULL, NULL, NULL, 403),
(106, '2005-02-09', '2014-03-04', 303, NULL, NULL),
(107, '2016-04-09', '2016-07-28', 309, NULL, NULL),
(108, '2013-12-29', NULL, NULL, NULL, 408),
(109, '2016-09-17', '2016-11-30', 307, NULL, NULL);
```



```
INSERT INTO Department
```

```
VALUES
```

```
(1001, 'Surgery'),
(1002, 'General Medicine'),
(1003, 'Pharmacy Department'),
(1004, 'Physical Medicine'),
(1005, 'Rehabilitation Department'),
(1006, 'Psychiatry'),
(1007, 'Radiology Department'),
(1008, 'Dietary Department'),
(1009, 'Outpatient Department'),
(1010, 'Medical Department');
```

```
INSERT INTO Person
```

```
VALUES
```

```
(2020190, 10334, 'Abakova', 'Aruzhan', 'female', 'Zhakupov 5', '870723456',
'abakovaa@gmail.com', '2002-09-09', 1010, 100, NULL, NULL),
(2020191, 10434, 'Kabdibek', 'Arunaz', 'female', 'Botkina 6', '870723448',
'kabdibeka@gmail.com', '1978-09-16', 1001, 101, NULL, NULL),
(2020192, 10634, 'Yerlankyzy', 'Dana', 'female', 'Akmescit 45', '870723156',
'yerlankyzyd@gmail.com', '1972-01-09', 1010, 102, NULL, NULL),
(2020193, 12334, 'Mausimbaev', 'Dastan', 'male', 'Kurmangazy 15', '874723466',
'mausimbaevd@mail.com', '1995-09-09', 1006, NULL, 603, NULL),
(2020194, 17334, 'Alzhanov', 'Alnur', 'male', 'Satpayev 5', '870729056',
'adilzhanova@mail.com', '1983-04-07', 1004, NULL, 604, NULL),
(2020195, 18374, 'Konarova', 'Saule', 'female', 'Kunanbaev 10', '870726756',
'konarovas@mail.com', '2003-09-19', 1010, NULL, 605, NULL),
(2020196, 15634, 'Dumanova', 'Dina', 'female', 'Kopesov 89', '870893456',
'dumanovad@mail.com', '1976-11-23', 1004, NULL, 606, NULL),
(2020197, 11564, 'Yermekuly', 'Asylzhan', 'male', 'Zhunusbay 6', '874723478',
'yermekulya@mail.com', '1993-09-09', 1004, NULL, NULL, 32),
(2020198, 20345, 'Alzhanov', 'Serik', 'male', 'Satpayev 49', '870724579',
'asylzhanovs@mail.com', '1991-09-29', 1004, NULL, NULL, 33),
(2020199, 10874, 'Gaziz', 'Alnur', 'male', 'Zhakupov 89', '870776586', 'gaziza@mail.com',
'2001-12-13', 1002, NULL, NULL, 34);
```

PERSON TABLE

person_id	SSN	lastName	firstName	gender	address	id_number	email	dateOfBirth	dep_id	emp_id	p_id	v_id
2000190	10334	Abakova	Azurban	female	Zhakupov 5	87072456	abakova@gmail.com	09.09.2002	1010	100	NULL	NULL
2000191	10434	Kabibbek	Aunaz	female	Bol'kha 6	87072348	kabibbek@gmail.com	16.05.1978	1001	101	NULL	NULL
2000192	10634	Yelarsky	Daria	female	Akresht 45	87072156	yelarsky@gmail.com	09.01.1972	1010	102	NULL	NULL
2000193	12334	Muslimbaev	Dashin	male	Kurnagazy 15	87472366	muslimbaev@gmail.com	09.06.1985	1006	NULL	603	NULL
2000194	17334	Adilhanov	Ahtur	male	Sabayev 5	87072056	adilhanov@gmail.com	07.04.1983	1004	NULL	604	NULL
2000195	18374	Kononova	Saule	female	Kurambay 10	87072676	kononova@gmail.com	19.08.2003	1010	NULL	605	NULL
2000196	15634	Dumanova	Dora	female	Koprov 89	870893456	dumanova@gmail.com	23.11.1976	1004	NULL	606	NULL
2000197	11664	Yermukuly	Ayylzhan	male	Zhurustay 6	87472478	yermukulya@gmail.com	09.08.1993	1004	NULL	NULL	32
2000198	20345	Ayylzhanov	Serik	male	Sabayev 49	870724579	ayylzhanov@gmail.com	29.05.1991	1004	NULL	NULL	33
2000199	10874	Gauzi	Ahtur	male	Zhakupov 89	870775686	gauza@gmail.com	13.12.2001	1002	NULL	NULL	34

DEPARTMENT TABLE

dep_id	dep_name
1001	Surgery
1002	General Medicine
1003	Pharmacy Department
1004	Physical Medicine
1005	Rehabilitation Department
1006	Psychiatry
1007	Radiology Department
1008	Dietary Department
1009	Outpatient Department
1010	Medical Department

EMPLOYEE TABLE

emp_id	hire_date	qualif_date	staff_id	n_id	ph_id
100	2015-12-09	2025-12-08	NULL	503	NULL
101	2011-02-09	2030-12-30	NULL	NULL	401
102	2005-12-09	NULL	301	NULL	NULL
103	2018-10-09	2028-12-13	NULL	505	NULL
104	2020-02-19	NULL	NULL	NULL	403
105	2008-10-19	NULL	303	NULL	NULL
106	2005-02-09	2024-03-04	309	NULL	NULL
107	2016-04-09	2026-07-28	NULL	509	NULL
108	2013-12-29	NULL	NULL	NULL	408
109	2016-09-17	2026-11-31	307	NULL	NULL

PATIENT TABLE

p_id	ph_id	pay_id	r_id	diag_id
600	400	200	1000	800
601	400	201	1001	801
602	402	202	1002	802
603	403	200	1002	803
604	403	200	1001	804
605	400	205	1005	805
606	400	202	1006	806
607	407	207	1002	807
608	402	208	1008	808
609	402	209	1002	809

VOLUNTEER TABLE

v_id	hourly_salary	worked_hour
30	2000	30
31	3000	48
32	1500	150
33	2600	65
34	2500	34
35	1300	65
36	2000	23
37	2500	43
38	1200	423
39	2000	422

STAFF TABLE

staff_id	job_title
300	cleaner
301	waitman
302	cleaner
303	chef
304	security_guard
305	cleaner
306	security_guard
307	waitman
308	cleaner
309	security_guard

NURSE TABLE

n_id	status	reception_id
500	nurse	900
501	nurse	900
502	nurse	902
503	head_nurse	903
504	nurse	902
505	nurse	900
506	nurse	906
507	head_nurse	904
508	nurse	904
509	nurse	904

PHYSICIANS TABLE

ph_id	position
400	Staff Internist
401	Attending Physician
402	Surgical Attending Physician
403	Senior Attending Physician
404	Head Chief of Medicine
405	Surgical Attending Physician
406	Surgical Attending Physician
407	MD Resident
408	Attending Psychiatrist
409	Staff Internist

ROOM TABLE

room_num	room_type	blockfloor	available
101A	single	3	1
102A	single	8	0
103A	double	6	1
104A	triple	9	0
105A	single	10	1
201A	double	5	1
202A	triple	1	0
203A	quad	2	0
204A	triple	4	1
205A	single	7	1

RECEPTION TABLE

reception_id	telephone	location
900	872745	first floor
901	872705	second floor
902	872785	third floor
903	872755	fourth floor
904	872775	fifth floor
905	872725	sixth floor
906	872715	seventh floor
907	872735	eighth floor
908	872765	ninth floor
909	872795	tenth floor

PAYMENT TABLE

pay_id	amount	paid
200	10000	1
201	8000	0
202	15000	1
203	2250	1
204	5000	0
205	9000	0
206	11600	1
207	5500	0
208	17000	1
209	6000	1

RECORD TABLE

r_id	n_id	room_num
1000	500	101A
1001	501	102A
1002	502	103A
1003	503	104A
1004	504	105A
1005	505	201A
1006	506	202A
1007	507	203A
1008	508	204A
1009	509	205A

DIAGNOSE TABLE

diag_id	name	remark	diag_date
800	hypertension	undetermined	2019-09-23
801	hyperlipidemia	serious	2020-06-19
802	diabetes	critical	2011-01-27
803	back pain	stable	2016-08-30
804	anxiety	undetermined	2015-10-12
805	obesity	serious	2019-04-28
806	allergic rhinitis	undetermined	2018-07-07
807	reflux esophagitis	critical	2009-12-18
808	pain in joint	serious	2018-11-30
809	asthma	critical	2017-03-03

-----CREATING PROCEDURES-----

#1(to insert new record to person table)

```
CREATE PROCEDURE insertPerson
@perId int, @ssn_ int, @ln varchar(30), @fn varchar(30), @g varchar(10), @add varchar(30),
@tn varchar(20), @em varchar(30), @dob datetime, @depId int, @empid int, @pid int, @vid int
AS
insert into Person values(@perId, @ssn_, @ln, @fn, @g, @add, @tn, @em, @dob, @depId, @empid,
@pid, @vid)
GO

EXEC insertPerson @perId = 20201992, @ssn_ = 10374, @ln = 'Gaziz', @fn = 'Alnur', @g =
'male', @add = 'Zhakupov 89', @tn = '870776586', @em = 'gaziza@mail.com', @dob =
'09-12-2001', @depId = 1010, @empid = NULL, @pid = NULL, @vid = 34;
```

#2(to insert new record to employee table)

```
CREATE PROCEDURE insertEmployee @ei int, @hd datetime , @qd datetime, @stid int, @nid int,
@phid int
AS
insert into Employee values(@ei, @hd, @qd, @stid, @nid, @phid)
GO

EXEC insertEmployee @ei = 112, @hd = '2016-09-17', @qd = '2019-11-18', @stid = 307, @nid =
NULL, @phid = NULL
```

#3(to insert new record to department table)

```
CREATE PROCEDURE insertDepartment @di int, @depName varchar(50) , @ei int
AS
insert into Department values(@di, @depName, @ei)
GO

EXEC insertDepartment @di = 453, @depName = 'Medical Department2', @ei = 112
```

#4(to update the existing record from room table)

```
CREATE PROCEDURE updateRoom @avail char(1), @n varchar(10)
AS
update Room set available = @avail where room_num = @n
GO

EXEC updateRoom @avail = '0', @n = '205A'
```

By procedure “updateRoom” we can update values of column “available”. In real life, if patient recovers, he will be discharged, then room will be available (1) or if the patient is admitted to the hospital, he will be lie in one of the hospital rooms. Then room will be not available(0)

#5(to update the existing record from Payment table)

```
CREATE PROCEDURE updatePayment1 @paid int , @id int
AS
update Payment set paid = @paid where pay_id = @id
GO
```

```
EXEC updatePayment1 @paid = 1, @id = 207
```

By procedure "updatePayment" we can update values of column "paid". In real life, if patient pays the money for his treatment, then "paid" will be update to "1". 1 is paid, 0 is not paid.

#6(to update the existing record from Nurse table)

```
CREATE PROCEDURE updateNurse @status varchar(10) , @id int
AS
update Nurse set status = @status where n_id = @id
GO
```

```
EXEC updateNurse @status = 'head nurse', @id = 508
```

By procedure "updateNurse" we update values of column "status". In real life, if a nurse gets raises, the value of the column will update from "nurse" to "head nurse".

#7(to delete the existing record from Patient table)

```
CREATE PROCEDURE deletePatient @id int
AS
delete from Patient where p_id = @id
GO
```

```
EXEC deletePatient @id = 609
```

By procedure "deletePatient" we can easily delete the Patient from the database.

#8(to delete the existing record from Volunteer table)

```
CREATE PROCEDURE deleteVol @id int
AS
delete from Volunteer where v_id = @id
GO
```

```
EXEC deleteVol @id = 39
```

By procedure "deletePatient" we can delete records from table "Patient". In real life, if a patient recovers and does not visit the hospital , then the table will not contain information about this patient.