AMERICAN INTERNATIONAL UNIVERSITY- BANGLADESH
Faculty of Science and Technology (FST)
Cse department

Introduction to Database
Faculty: Anjir Ahemd Chowdhury

HOSPITAL MANAGEMENT SYSTEM

07.12.2021

Group members

SANDIP MISRA	21-44506-1
MD. REASAT AHMED	21-44542-1
MAHBUBUL ISLAM ABDULLAH	21-44523-1

Introduction

Our project is about a Hospital Management System (HMS). It also improves the management visibility of hospitals, all information, and data regarding the patient, doctor and medicine could be seen by any department easily. A hospital stores extensive amounts of patient data for easy accessibility. The databases behind this collection of information are massive, with complex data structures and security protecting sensitive data. By using this management system we can save much of our time. It makes finding data easier than using pen and paper. By using our hospital management system we can find the data very easily and save plenty of time.

List Of Tables

I. DEPARTMENT
II. DOCTORS
III. DOCTORS_INFO
IV. PATIENTS
V. CHECK_UP
VI. PATIENT_ADMIT
VII. PATIENT_DISCHARGE
VIII. RAGULAR_PATIENT
IX. PRS_MEDICINE

Normalization

➔ First Normal Form (1NF)

For a table to be in the First Normal Form, it should follow the following 4 rules:

- It should only have single(atomic) valued attributes/columns.
- Values stored in a column should be of the same domain
- All the columns in a table should have unique names.
- The order in which data is stored, does not matter.

In our project by default, we follow the 1st law of 1st normalization.

In the PATIENT_DISCHARGED table we follow the 2nd law of 1st normalization. First, we determine the value of the MEDICINE column with the value of the PAYMENT column. Later we fix it and put in their respective columns. This table we also follow the 3rd law of 1st normalization. This table has TR_ADVISE, TR_GIVEN columns. But when we create tables, we assign TR_ADVS columns twice to the table. Then we solve these problems and we define the TR_ADVISE and TR_GIVEN columns.

At last, we follow the 4th law of 1st normalization. When we assign VALUES in columns, sometimes we assign it to the sequence and sometimes we assign the VALUES randomly.

➔ Second Normal Form (2NF)

For a table to be in the Second Normal Form:

- It should be in the First Normal form.
- It should not have Partial Dependency.

DOCTORS table following the 1st normalization from.

The DOCTORS table contains the columns like DR_ID, DR_NAME, QUALIFICATION, DEPARTMENT, JOIN_DATE, SALARY, PHONE_NO, EMAIL_ADDRESS. Here we see partial dependency and we solve this problem. Following the 2nd law, we have divided the DOCTORS table into DOCTORS (DR_ID, DR_NAME, QUALIFICATION, DEPARTMENT) and DOCTORS_INFO (JOIN_DATE, SALARY, EMAIL_ADDRESS, PHONE_NO) tables.

➔ Third Normal Form (3NF)

A table is said to be in the Third Normal Form when,

- It is in the Second Normal form.
- It doesn't have Transitive Dependency.

We did not find any specific place to apply this law in our project.

Case Study

In a hospital, there are many departments. Each department has a unique department Id, name, location, and facilities. Many doctors work for the hospital and the departments they are associated with. Each doctor has a name, qualification, email address, phone number, identity number, salary, date of joining. Doctors are referred to by their doctor id number. When any patient arrives in the hospital for a check-up. A patient ID number is generated and name, age, sex, address, phone number, diagnosis is also stored. After storing the necessary details, the patient is sent to the doctor for check-up. There are two types of patients in the hospital. Some regular patients and some new patients are coming for admission. When regular patients are coming to the hospital some formalities need to be filled out like payment, visiting date. After the check-up, the doctor prescribes the medicine. When a patient is admitted, details are stored. Information stored includes patient number, advance payment, mode of payment, room number, condition. After the treatment is completed, the doctor discharges the patient. Whenever a patient is discharged from the hospital, some formalities must be completed like payment made, treatment given, treatment advice, date of discharge.

ER-Diagram

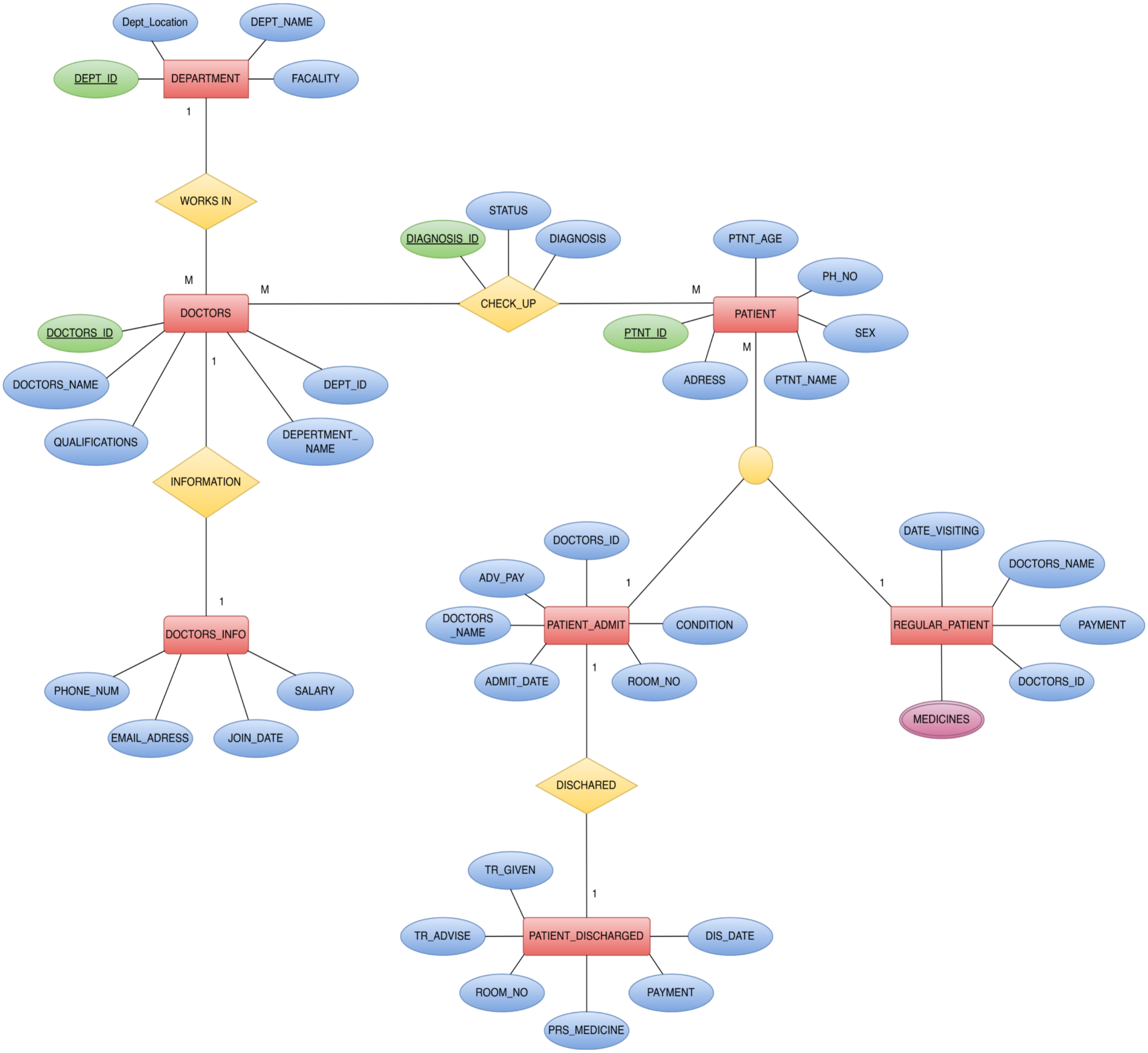
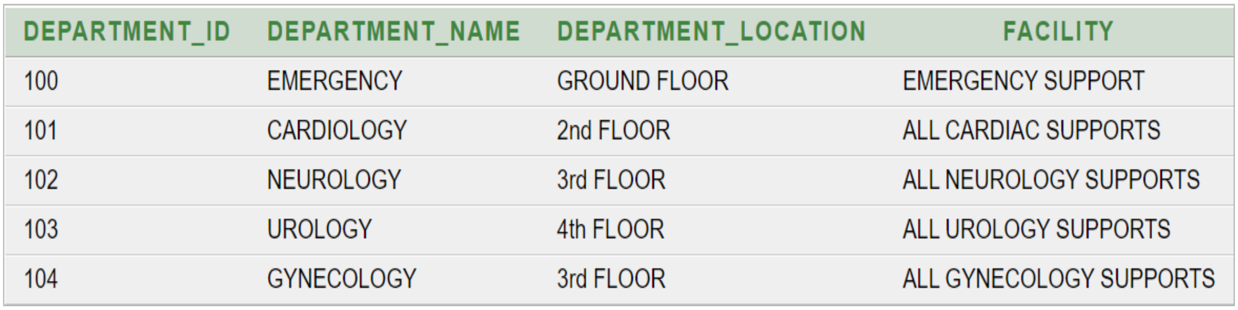

Dept_Location
DEPT_NAME
DEPT_ID
DEPARTMENT
FACALITY
1
WORKS IN
M
STATUS
DIAGNOSIS_ID
DIAGNOSIS
PTNT_AGE
PH_NO
M
DOCTORS
M
CHECK_UP
M
PATIENT
DOCTORS_ID
PTNT_ID
SEX
M
DOCTORS_NAME
1
DEPT_ID
ADRESS
PTNT_NAME
QUALIFICATIONS
DEPERTMENT_
NAME
INFORMATION
DOCTORS_ID
DATE_VISITING
ADV_PAY
DOCTORS_NAME
1
1
1
DOCTORS
_NAME
PATIENT_ADMIT
CONDITION
REGULAR_PATIENT
PAYMENT
DOCTORS_INFO
ADMIT_DATE
1
ROOM_NO
DOCTORS_ID
PHONE_NUM
SALARY
MEDICINES
EMAIL_ADRESS
JOIN_DATE
DISCHARED
TR_GIVEN
1
TR_ADVISE
PATIENT_DISCHARGED
DIS_DATE
ROOM_NO
PAYMENT
PRS_MEDICINE

TABLE CREATION

→ **DEPARTMENT TABLE :**

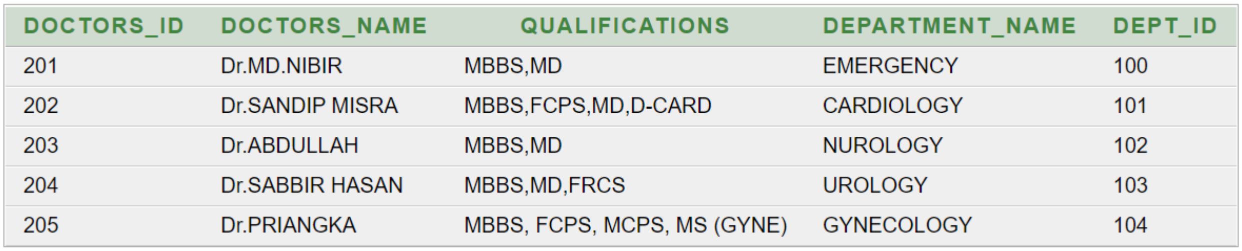


DEPARTMENT_ID	DEPARTMENT_NAME	DEPARTMENT_LOCATION	FACILITY
100	EMERGENCY	GROUND FLOOR	EMERGENCY SUPPORT
101	CARDIOLOGY	2nd FLOOR	ALL CARDIAC SUPPORTS
102	NEUROLOGY	3rd FLOOR	ALL NEUROLOGY SUPPORTS
103	UROLOGY	4th FLOOR	ALL UROLOGY SUPPORTS
104	GYNECOLOGY	3rd FLOOR	ALL GYNECOLOGY SUPPORTS

```
CREATE TABLE DEPARTMENT(DEPARTMENT_ID int primary key NOT NULL,
DEPARTMENT_NAME varchar(200), DEPARTMENT_LOCATION varchar(200), FACILITY
varchar(500));
```

```
INSERT into DEPARTMENT (DEPARTMENT_ID, DEPARTMENT_NAME,
DEPARTMENT_LOCATION , FACILITY) VALUES ('100', 'EMERGENCY', 'GROUND
FLOOR', 'EMERGENCY SUPPORT');

INSERT into DEPARTMENT (DEPARTMENT_ID, DEPARTMENT_NAME,
DEPARTMENT_LOCATION , FACILITY) VALUES ('101', 'CARDIOLOGY', '2nd FLOOR',
'ALL CARDIAC SUPPORTS');

INSERT into DEPARTMENT (DEPARTMENT_ID, DEPARTMENT_NAME,
DEPARTMENT_LOCATION , FACILITY) VALUES ('102', 'NEUROLOGY', '3rd FLOOR',
'ALL NEUROLOGY SUPPORTS');

INSERT into DEPARTMENT (DEPARTMENT_ID, DEPARTMENT_NAME,
DEPARTMENT_LOCATION , FACILITY) VALUES ('103', 'UROLOGY', '4th FLOOR',
'ALL UROLOGY SUPPORTS');

INSERT into DEPARTMENT (DEPARTMENT_ID, DEPARTMENT_NAME,
DEPARTMENT_LOCATION , FACILITY) VALUES ('104', 'GYNECOLOGY', '3rd FLOOR',
'ALL GYNECOLOGY SUPPORTS');
```

→ DOCTORS TABLE :



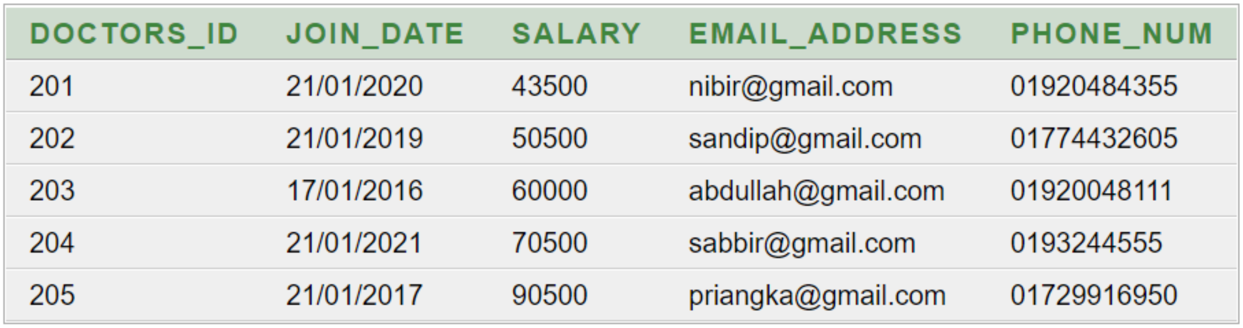
DOCTORS_ID	DOCTORS_NAME	QUALIFICATIONS	DEPARTMENT_NAME	DEPT_ID
201	Dr.MD.NIBIR	MBBS,MD	EMERGENCY	100
202	Dr.SANDIP MISRA	MBBS,FCPS,MD,D-CARD	CARDIOLOGY	101
203	Dr.ABDULLAH	MBBS,MD	NUROLOGY	102
204	Dr.SABBIR HASAN	MBBS,MD,FRCS	UROLOGY	103
205	Dr.PRIANGKA	MBBS, FCPS, MCPS, MS (GYNE)	GYNECOLOGY	104

```
CREATE TABLE DOCTORS(DOCTORS_ID int PRIMARY KEY NOT NULL, DOCTORS_NAME
varchar(200), QUALIFICATIONS varchar(200), DEPARTMENT_NAME varchar(200),
DEPT_ID int);
```

```
ALTER TABLE DOCTORS
ADD FOREIGN KEY (dept_id) REFERENCES DEPARTMENT(department_id);
```

```
Insert into DOCTORS (DOCTORS_ID, DOCTORS_NAME, QUALIFICATIONS ,
DEPARTMENT_NAME, DEPT_ID) VALUES ('201', 'Dr.MD.NIBIR ', 'MBBS, MD',
'EMERGENCY', '100');

INSERT into DOCTORS (DOCTORS_ID, DOCTORS_NAME, QUALIFICATIONS ,
DEPARTMENT_NAME,DEPT_ID) VALUES ('202', 'Dr.SANDIP MISRA', 'MBBS,FCPS, MD,
D-CARD','CARDIOLOGY','101');

INSERT into DOCTORS (DOCTORS_ID, DOCTORS_NAME, QUALIFICATIONS ,
DEPARTMENT_NAME ,DEPT_ID) VALUES ('203', 'Dr.ABDULLAH', 'MBBS,MD',
'NEUROLOGY', '102');

INSERT into DOCTORS (DOCTORS_ID, DOCTORS_NAME, QUALIFICATIONS ,
DEPARTMENT_NAME ,DEPT_ID) VALUES ('204', 'Dr.SABBIR HASAN',
'MBBS,MD,FRCS', 'UROLOGY', '103');

INSERT into DOCTORS (DOCTORS_ID, DOCTORS_NAME, QUALIFICATIONS ,
DEPARTMENT_NAME ,DEPT_ID) VALUES ('205', 'Dr.PRIANGKA', 'MBBS, FCPS, MCPS,
MS (GYNE)','GYNECOLOGY', '104');
```

→ DOCTORS_INFO TABLE :



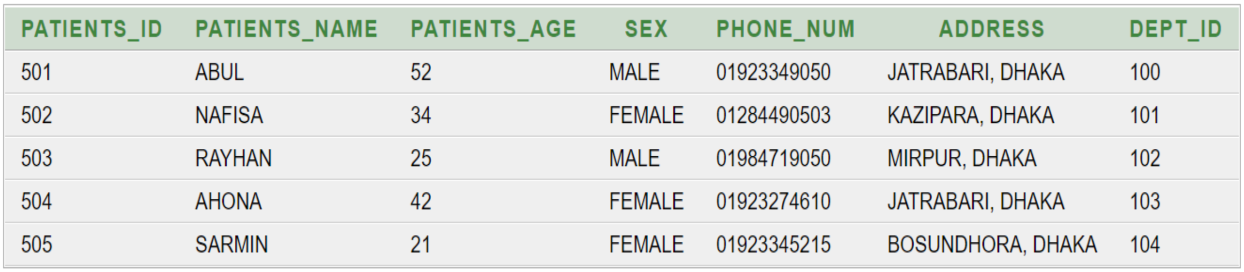
DOCTORS_ID	JOIN_DATE	SALARY	EMAIL_ADDRESS	PHONE_NUM
201	21/01/2020	43500	nibir@gmail.com	01920484355
202	21/01/2019	50500	sandip@gmail.com	01774432605
203	17/01/2016	60000	abdullah@gmail.com	01920048111
204	21/01/2021	70500	sabbir@gmail.com	0193244555
205	21/01/2017	90500	priangka@gmail.com	01729916950

```
CREATE TABLE DOCTORS_INFO(DOCTORS_ID int NOT NULL,JOIN_DATE varchar(500),
SALARY varchar(200) NOT NULL, EMAIL_ADDRESS varchar(200), PHONE_NUM
varchar(50));
```

```
ALTER TABLE DOCTORS_INFO
ADD FOREIGN KEY (doctors_id) REFERENCES DOCTORS(doctors_id);
```

```
INSERT into DOCTORS_INFO (JOIN_DATE, SALARY, EMAIL_ADDRESS,
PHONE_NUM,DOCTORS_ID) VALUES ('21/01/2020', '43500', 'nibir@gmail.com',
'01920484355','201');

INSERT into DOCTORS_INFO (JOIN_DATE, SALARY, EMAIL_ADDRESS,
PHONE_NUM,DOCTORS_ID) VALUES ('21/01/2019', '50500', 'sandip@gmail.com',
'01774432605','202');

INSERT into DOCTORS_INFO (JOIN_DATE, SALARY, EMAIL_ADDRESS, PHONE_NUM,
DOCTORS_ID) VALUES ('17/01/2016', '60000', 'abdullah@gmail.com',
'01920048111','203');

INSERT into DOCTORS_INFO (JOIN_DATE, SALARY, EMAIL_ADDRESS, PHONE_NUM,
DOCTORS_ID) VALUES ('21/01/2021', '70500', 'sabbir@gmail.com',
'0193244555','204');

INSERT into DOCTORS_INFO (JOIN_DATE, SALARY, EMAIL_ADDRESS, PHONE_NUM,
DOCTORS_ID) VALUES ('21/01/2017', '90500', 'priangka@gmail.com',
'01729916950','205');
```

➔ PATIENTS TABLE :



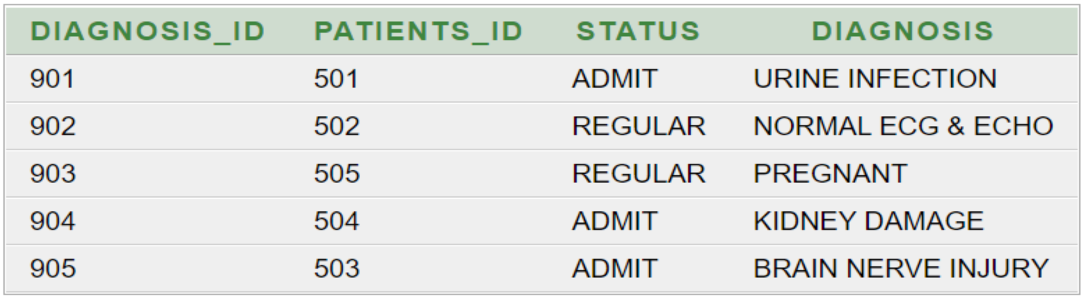
PATIENTS_ID	PATIENTS_NAME	PATIENTS_AGE	SEX	PHONE_NUM	ADDRESS	DEPT_ID
501	ABUL	52	MALE	01923349050	JATRABARI, DHAKA	100
502	NAFISA	34	FEMALE	01284490503	KAZIPARA, DHAKA	101
503	RAYHAN	25	MALE	01984719050	MIRPUR, DHAKA	102
504	AHONA	42	FEMALE	01923274610	JATRABARI, DHAKA	103
505	SARMIN	21	FEMALE	01923345215	BOSUNDHORA, DHAKA	104

```
CREATE TABLE PATIENTS(PATIENTS_ID int primary key NOT NULL, PATIENTS_NAME
varchar(200),  PATIENTS_AGE varchar(500), SEX varchar2(200),PHONE_NUM
varchar(50), ADDRESS varchar(200), DEPT_ID int);
```

```
ALTER TABLE PATIENTS
ADD FOREIGN KEY (DEPT_ID) REFERENCES DEPARTMENT(department_id);
```

```
INSERT into PATIENTS (PATIENTS_ID, PATIENTS_NAME, PATIENTS_AGE , SEX,
PHONE_NUM, ADDRESS, DEPT_ID) VALUES ('501', 'ABUL', '52', 'MALE',
'01923349050', 'JATRABARI, DHAKA',100);

INSERT into PATIENTS (PATIENTS_ID, PATIENTS_NAME, PATIENTS_AGE , SEX,
PHONE_NUM, ADDRESS, DEPT_ID) VALUES ('502', 'NAFISA', '34', 'FEMALE',
'01284490503', 'KAZIPARA, DHAKA',101);

INSERT into PATIENTS (PATIENTS_ID, PATIENTS_NAME, PATIENTS_AGE , SEX,
PHONE_NUM, ADDRESS, DEPT_ID) VALUES ('503', 'RAYHAN', '25', 'MALE',
'01984719050', 'MIRPUR, DHAKA',102);

INSERT into PATIENTS (PATIENTS_ID, PATIENTS_NAME, PATIENTS_AGE , SEX,
PHONE_NUM, ADDRESS, DEPT_ID) VALUES ('504', 'AHONA', '42', 'FEMALE',
'01923274610', 'JATRABARI, DHAKA',103);

INSERT into PATIENTS (PATIENTS_ID, PATIENTS_NAME, PATIENTS_AGE , SEX,
PHONE_NUM, ADDRESS, DEPT_ID) VALUES ('505', 'SARMIN', '21', 'FEMALE',
'01923345215', 'BOSUNDHORA, DHAKA',104);
```

➔ CHECK_UP TABLE :



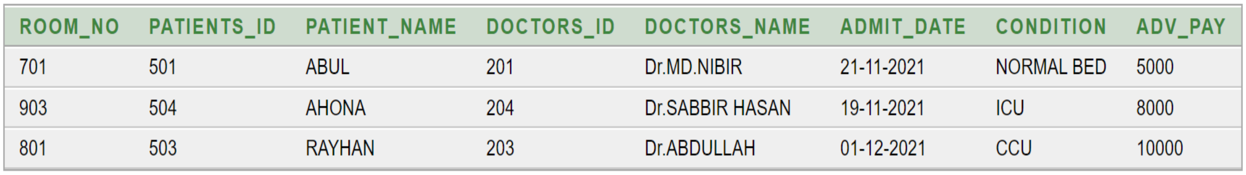
DIAGNOSIS_ID	PATIENTS_ID	STATUS	DIAGNOSIS
901	501	ADMIT	URINE INFECTION
902	502	REGULAR	NORMAL ECG & ECHO
903	505	REGULAR	PREGNANT
904	504	ADMIT	KIDNEY DAMAGE
905	503	ADMIT	BRAIN NERVE INJURY

```
CREATE TABLE CHECK_UP(DIAGNOSIS_ID int primary key NOT NULL, PATIENTS_ID
int,  STATUS varchar(500), DIAGNOSIS varchar(200));
```

```
ALTER TABLE CHECK_UP
ADD FOREIGN KEY (PATIENTS_ID) REFERENCES PATIENTS(PATIENTS_ID);
```

```
INSERT into CHECK_UP(DIAGNOSIS_ID, PATIENTS_ID, STATUS, DIAGNOSIS) VALUES
('901', '501', 'ADMIT', 'URINE INFECTION');

INSERT into CHECK_UP(DIAGNOSIS_ID, PATIENTS_ID, STATUS, DIAGNOSIS) VALUES
('902', '502', 'REGULAR', 'NORMAL ECG & ECHO');

INSERT into CHECK_UP(DIAGNOSIS_ID, PATIENTS_ID, STATUS, DIAGNOSIS) VALUES
('903', '505', 'REGULAR', 'PREGNANT');

INSERT into CHECK_UP(DIAGNOSIS_ID, PATIENTS_ID, STATUS, DIAGNOSIS) VALUES
('904', '504', 'ADMIT', 'KIDNEY DAMAGE');

INSERT into CHECK_UP(DIAGNOSIS_ID, PATIENTS_ID, STATUS, DIAGNOSIS) VALUES
('905', '503', 'ADMIT', 'BRAIN NERVE INJURY');
```

➔ PATIENT_ADMIT TABLE :



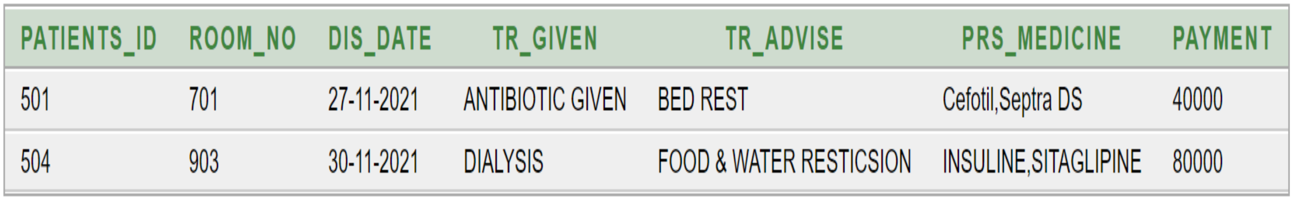
ROOM_NO	PATIENTS_ID	PATIENT_NAME	DOCTORS_ID	DOCTORS_NAME	ADMIT_DATE	CONDITION	ADV_PAY
701	501	ABUL	201	Dr.MD.NIBIR	21-11-2021	NORMAL BED	5000
903	504	AHONA	204	Dr.SABBIR HASAN	19-11-2021	ICU	8000
801	503	RAYHAN	203	Dr.ABDULLAH	01-12-2021	CCU	10000

```
CREATE TABLE PATIENT_ADMIT(ROOM_NO int NOT NULL, PATIENTS_ID
int,PATIENT_NAME VARCHAR(500),DOCTORS_ID int, DOCTORS_NAME varchar(100),
ADMIT_DATE varchar(500), CONDITION varchar(200),ADV_PAY varchar(200));
```

```
ALTER TABLE PATIENT_ADMIT
ADD FOREIGN KEY (PATIENTS_ID) REFERENCES PATIENTS(PATIENTS_ID);
```

```
Insert into PATIENT_ADMIT(ROOM_NO, PATIENTS_ID, PATIENT_NAME, DOCTORS_ID,
DOCTORS_NAME, ADMIT_DATE, CONDITION, ADV_PAY) VALUES ('701', '501',
'ABUL', '201', 'Dr.MD.NIBIR', '21-11-2021', 'NORMAL BED', '5000');

INSERT into PATIENT_ADMIT(ROOM_NO, PATIENTS_ID, PATIENT_NAME, DOCTORS_ID,
DOCTORS_NAME, ADMIT_DATE, CONDITION, ADV_PAY) VALUES ('903',
'504','AHONA', '204', 'Dr.SABBIR HASAN', '19-11-2021', 'ICU', '8000');

INSERT into PATIENT_ADMIT(ROOM_NO, PATIENTS_ID, PATIENT_NAME, DOCTORS_ID,
DOCTORS_NAME, ADMIT_DATE, CONDITION, ADV_PAY) VALUES ('801', '503',
'RAYHAN','203', 'Dr.ABDULLAH', '01-12-2021', 'CCU', '10000');
```

➔ **PATIENT_DISCHARGE TABLE:**



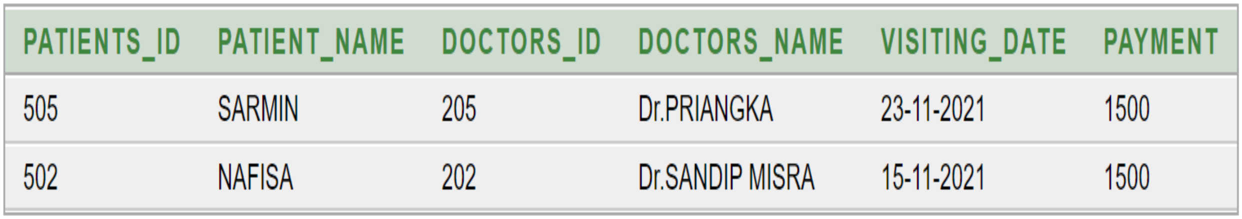
PATIENTS_ID	ROOM_NO	DIS_DATE	TR_GIVEN	TR_ADVISE	PRS_MEDICINE	PAYMENT
501	701	27-11-2021	ANTIBIOTIC GIVEN	BED REST	Cefotil,Septra DS	40000
504	903	30-11-2021	DIALYSIS	FOOD & WATER RESTICSION	INSULINE,SITAGLIPINE	80000

```
CREATE TABLE PATIENT_DISCHARGE(PATIENTS_ID int,ROOM_NO int,  DIS_DATE
varchar(500), TR_GIVEN varchar(200),TR_ADVISE varchar(200),PRS_MEDICINE
varchar(200), PAYMENT varchar(100));
```

```
ALTER TABLE PATIENT_DISCHARGE
ADD FOREIGN KEY (PATIENTS_ID) REFERENCES PATIENTS(PATIENTS_ID);
```

```
INSERT into PATIENT_DISCHARGE(PATIENTS_ID,ROOM_NO, DIS_DATE, TR_GIVEN,
TR_ADVISE, PRS_MEDICINE,PAYMENT) VALUES ('501', '701', '27-11-2021',
'ANTIBIOTIC GIVEN', ' BED REST','Cefotil,Septra DS','40000');

INSERT into PATIENT_DISCHARGE(PATIENTS_ID,ROOM_NO, DIS_DATE, TR_GIVEN,
TR_ADVISE, PRS_MEDICINE,PAYMENT) VALUES ('504', '903', '30-11-2021',
'DIALYSIS', 'FOOD & WATER RESTRICTION','INSULIN,SITAGLIPTIN','80000');
```

➔ REGULAR_PATIENT TABLE :



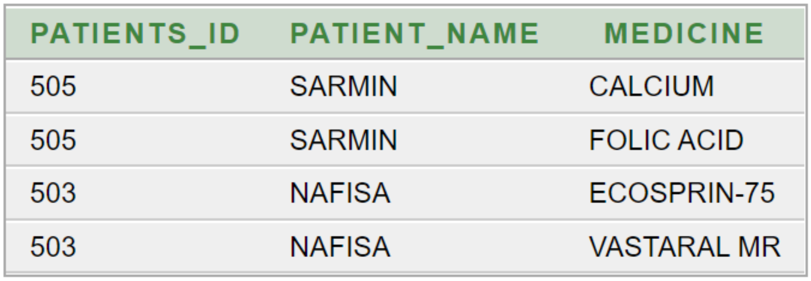
PATIENTS_ID	PATIENT_NAME	DOCTORS_ID	DOCTORS_NAME	VISITING_DATE	PAYMENT
505	SARMIN	205	Dr.PRIANGKA	23-11-2021	1500
502	NAFISA	202	Dr.SANDIP MISRA	15-11-2021	1500

```
CREATE TABLE REGULAR_PATIENT (PATIENTS_ID int NOT NULL, PATIENT_NAME
VARCHAR(500), DOCTORS_ID int, DOCTORS_NAME varchar(100), VISITING_DATE
varchar(100), PAYMENT varchar(100));
```

```
ALTER TABLE REGULAR_PATIENT
ADD FOREIGN KEY (PATIENTS_ID) REFERENCES PATIENTS(PATIENTS_ID);
```

```
INSERT into REGULAR_PATIENT(PATIENTS_ID,PATIENT_NAME,DOCTORS_ID,
DOCTORS_NAME, VISITING_DATE,PAYMENT) VALUES ('505','SARMIN', '205',
'Dr.PRIANGKA', '23-11-2021', '1500');

INSERT into REGULAR_PATIENT(PATIENTS_ID,PATIENT_NAME,DOCTORS_ID,
DOCTORS_NAME, VISITING_DATE,PAYMENT) VALUES ('502', 'NAFISA','202',
'Dr.SANDIP MISRA', '15-11-2021', '1500');
```

➔ PRS_MEDICINE TABLE :



PATIENTS_ID	PATIENT_NAME	MEDICINE
505	SARMIN	CALCIUM
505	SARMIN	FOLIC ACID
503	NAFISA	ECOSPRIN-75
503	NAFISA	VASTARAL MR

```
CREATE TABLE PRS_MEDICINE(PATIENTS_ID int NOT NULL, PATIENT_NAME
varchar(100), MEDICINE varchar(100));
```

```
ALTER TABLE PRS_MEDICINE
ADD FOREIGN KEY (PATIENTS_ID) REFERENCES PATIENTS(PATIENTS_ID)
```

```
INSERT into PRS_MEDICINE(PATIENTS_ID,PATIENT_NAME, MEDICINE) VALUES
('505', 'SARMIN', 'CALCIUM');

INSERT into PRS_MEDICINE(PATIENTS_ID,PATIENT_NAME, MEDICINE) VALUES
('505', 'SARMIN', 'FOLIC ACID');

INSERT into PRS_MEDICINE(PATIENTS_ID,PATIENT_NAME, MEDICINE) VALUES
('502', 'NAFISA', 'ECOSPRIN-75');

INSERT into PRS_MEDICINE(PATIENTS_ID,PATIENT_NAME, MEDICINE) VALUES
('502', 'NAFISA', 'VASTAREL MR');
```

Query

Simple :

1) Display all the records in the department table?

```
select * from DEPARTMENT
```

$$\sigma_{*}\,(DEPARTMENT)$$

2) Display doctors_id ,doctors_name,dept_id , qualification in Doctors table?

```
select DOCTORS_ID,DOCTORS_NAME, DEPT_ID, QUALIFICATIONS from DOCTORS
```

$$\sigma_{DOCTORS\_ID,DOCTORS\_NAME,\,DEPT\_ID,QUALIFICATIONS}\;(doctors)$$

3)Display all the records in the admit_patient table?

```
select * from patient_admit
```

$$\sigma_{*}\,(patient\_admit)$$

4)Display all the records in the PATIENT_DISCHARGE table?

```
select * from PATIENT_DISCHARGE
```

$$\sigma_{*}\,(patient\_discharge)$$

5) Display patients_name ,patients_id in the patients table?

```
select patients_name, patients_id from PATIENTS
```

$$\sigma_{patients\_name,\,patients\_id}\,(patients)$$

6) Display doctors_id, salary in doctors_info?

```
select doctors_id, salary from DOCTORS_INFO
```

$$\sigma_{doctors\_id,\,salary}\,(doctors\_info)$$

7)Display patient_name , patients_id in regular_patient table?

```
 Select patient_name, patients_id from regular_patient
```

$$\sigma_{patients\_name,\,patients\_id}\,(regular\_patient)$$

Complex :

8)Display patients_name in the patients table where the second letter of patient_name is ‘A’?

```
select patients_name from patients where patients_name like '_A%'
```

$$\Pi_{patients\_name}(\ \sigma_{patients\_name\ like\ '\_A\%'}(Patients))$$

9) Display patients_name in the patients table where the last letter of patient_name is ‘A’?

```
select patients_name from patients where patients_name like '%A'
```

$$\Pi_{patients\_name}(\ \sigma_{patients\_name\ like\ '\%\_A'}(Patients))$$

10) Display all doctors who join before 21/01/2019?

```
select * from DOCTORS_INFO where JOIN_DATE<='21/01/2019'
```

$$\pi_{*}\ (\ \sigma_{JOIN\_DATE<='21/01/2019'}\ (doctors\_info))$$

Sub-Query:

11) Display all the doctors_name in the doctors table where salary is greater than avg salary?

```
select DOCTORS_NAME from DOCTORS where DOCTORS_ID in(select DOCTORS_ID
from DOCTORS_INFO where SALARY>(select avg(SALARY) from DOCTORS_INFO))
```

$$\pi_{doctors\_name}(\sigma_{doctors\_id\ in(\pi_{doctors\_id}(\sigma_{salary>(\sigma_{avg(\ (SALARY))}(doctors\_info))}(doctors\_info))}(doctors))$$

12)Display all the doctors_name in the doctors table where JOIN_DATE before or Equal '21/01/2019'?

```
Select DOCTORS_NAME from DOCTORS where DOCTORS_ID not in (select
DOCTORS_ID from DOCTORS_INFO where JOIN_DATE<='21/01/2019')
```

$$\pi_{doctors\_name}(\sigma_{doctor\_id\ not\ in\ (\pi_{doctor\_id}(\sigma_{doctor\_id>JOIN\_DATE<='21/01/2019'}(doctors\_info)))}(doctors))$$

Function:

13) Count how many patients_id in patients_table?

```
select count (patients_id) from patients
```

$$\sigma_{count\ (patients\_id)}(Patients)$$

14) Display a round of average salary in the doctors_info table?

```
select avg(round (SALARY)) from DOCTORS_INFO
```

$$\sigma_{avg(round\ (SALARY))}(Patients)$$

15) Display sumision of payment in patient_Discharge table?

```
select sum (PAYMENT) from PATIENT_DISCHARGE
```

$$\sigma_{sum\ (PAYMENT)}(Patients)$$

Joining:

16) Display all the REGULAR_PATIENT who are in the PRS_MEDICINE table?

```
SELECT * FROM PRS_MEDICINE JOIN REGULAR_PATIENT ON
PRS_MEDICINE.PATIENTS_ID= REGULAR_PATIENT.PATIENTS_ID
```

$$\pi_{*}\ (\sigma_{PRS\_MEDICINE.PATIENTS\_ID=\ REGULAR\_PATIENT.PATIENTS\_ID}(PRS\_MEDICINE \bowtie REGULAR\_PATIENT))$$

17) Display all the doctors_id that are in Doctors and doctors _Info TABLE?

```
SELECT * FROM DOCTORS
JOIN DOCTORS_INFO ON DOCTORS.DOCTORS_ID= DOCTORS_INFO.DOCTORS_ID
```

$$\pi_{*}\ (\sigma_{DOCTORS.DOCTORS\_ID=\ DOCTORS\_INFO.DOCTORS\_ID}(DOCTORS \bowtie DOCTORS\_INFO))$$

Views:

18) Create a view called doctors_bio based on the doctors_name, qualifications, department name Where doctors_id are 202 and 205.

```
create view doctors_bio as select doctors_name,qualifications,
DEPARTMENT_NAME from doctors
where DOCTORS_ID=202 or DOCTORS_ID= 205
```

$$Create\ view\ doctors\_bio\ as$$
$$\pi_{doctors\_name, qualifications,\ DEPARTMENT\_NAME}\ (\ \sigma_{DOCTORS\_ID=202\ or\ DOCTORS\_ID=\ 205}(doctors))$$

19)Create a view called patients_bio based on the patients_name, patients_age and sex where dept_id are 102 ,103 and 100.

```
create view patients_bio as select patients_name ,patients_age , sex
from patients
where DEPT_ID=102 or DEPT_ID= 103 or DEPT_ID= 100
```

$$Create\ view\ patients\_bio\ as$$
$$\pi_{patients\_name\ ,patients\_age\ ,\ sex}\ (\ \sigma_{DEPT\_ID=102\ or\ DEPT\_ID=\ 103\ or\ DEPT\_ID=\ 100}(doctors))$$

Group By:

20)Display doctors_id in the doctors table group by doctors_id.

```
select DOCTORS_ID from DOCTORS_INFO group by  DOCTORS_ID
```

$$\sigma_{DOCTORS\_ID}\ \gamma_{DOCTORS\_ID}(doctors\_info)$$

Contribution

SANDIP MISRA	**File Format, Introduction, Normalizations, Case study, Table Alter, Query(7), Relational Algebra.**
MD. REASAT AHMED	**Normalizations, Er-DiagramUpdate, Table Creation , Query(6) .**
MAHBUBUL ISLAM ABDULLAH	**Cover Page, Er- diagram Update, Table Creation, Query(7) .**