

Hospital Management System



**J Component-Project Report
ITE-1003**

**Database Management Systems
Winter Semester – 2020-21**

Submitted to

Prof. Bimal Kumar Ray

Submitted by

19BIT0378- Paras Dang

1. Choose a mini world for design and implementation of it's a database assigning an appropriate title for the database.

The mini world that we have chosen is a hospital for which we have designed a miniature backend.

The title of our database is Hospital Healthcare RECORD.

2. Write down the functional requirements for the database (in approximately 1500 words). The data requirements, apart from data to be stored in the database should also take into account the necessary integrity constraints that are reasonable for the database under consideration. The functional requirements should involve at least four different scenarios of removal of old data, four different scenarios for modification of existing data and eight different scenarios of data retrieval.

Data Requirements:

Information needs to be stored about the following entities:

- ❖ Patient information includes Patient_ID which would be the key attribute used to uniquely identify the patient, patient name (made of first and last name) and whether the patient is admitted in the hospital or not.
- ❖ Doctor information includes Doctor_ID which would be the key attribute used to uniquely identify the doctor, doctor name (made of first and last name), doctor's contact information and the specialization of the doctor.
- ❖ Lab technician information includes, the Technician_ID which is the key attribute used to uniquely identify the technician, the Technician_Name and the list of labs the technician is allowed to operate (as a multi-valued attribute).
- ❖ Lab reports are created by Lab Technician whenever a Doctor orders for them to be created for a patient. Lab reports contain information such as the Report_ID which would be the key attribute, the test which was conducted and the result. Patient requiring multiple tests would have different lab reports for each one.
- ❖ Drug information is required to be stored in the database as the prescriptions prescribed by the doctors to every patient contain a set a number of drugs. So, to prevent redundancy the drugs are stored in a different relation than prescription. Drug information contains the Drug_ID which would be the key attribute, Drug_name and the cost of the given drug.

- ❖ For every patient stored in the Hospital Record Management System, a record is maintained for the total charges incurred by the patient. The charges can either be covered under health insurance or be paid by the patient directly.
- ❖ Bills under a particular patient is identified by the Bill_number(partial key). A derived attribute of Total_Cost is calculated as the summation of two attributes Under_Insurance and Personal_cost.
- ❖ Prescription information contains the list of drugs prescribed by a doctor for a given patient. For a particular patient, the prescription can be uniquely identified by the Prescription number which is a partial key. The attribute Issue_Date stores the date on which the given prescription was issued whereas Validity_Date stores the date upto which the given prescription is valid.
- ❖ Record ID uniquely identifies the medical records. Medical record for each patient is singular. The description attribute holds all the information for the given medical record.

Information needs to be stored about the following relationships:

- ✓ The relationship visit is between the entities Patient and Doctor and contains Visit_Date which stores the last date on which a particular patient visited a particular doctor. The cardinality ratio is m:n for Patient to Doctor. Both have partial participation.
- ✓ The relationship gives is between Patient and Feedback. The cardinality ratio is m:1 for Patient and Feedback.
- ✓ The relationship accesses is between the entities Patient and Medical_Records.
- ✓ It has a cardinality of 1:1 for patient to medical record. Patient has partial participation whereas Medical_Records has total participation.
- ✓ The relationship updates is between the entities Doctor and Medical_Records. It has a cardinality ratio of 1:n for Doctor to Medical_Record. Both have partial participation.
- ✓ The relationship Creates is between Lab_Report and Lab_Technician . The cardinality ratio is 1:n for Lab_Technician to Lab_Report . Lab_Report has total participation whereas Lab_Technician has partial participation.
- ✓ The relationship Belongs_To is between Lab_Report and Patient. The cardinality ratio is 1:n for Patient to Lab_Report. Patient has partial participation whereas Lab_Report has total participation.
- ✓ The relationship orders for is between the entities Doctor and Lab_Report. The cardinality ratio is 1:n for Doctor to Lab_Report and Doctor has partial participation whereas Lab_Report has total participation.
- ✓ The relationship Incurs is between Bill and Patient. The cardinality ratio is 1:n for Patient to Bill . Bill has total participation whereas Patient has partial participation.
- ✓ The relationship Prescribed_To is between Patient and Prescription. The cardinality ratio is 1:n for Patient to Prescription. Prescription has total participation whereas Patient has partial participation.
- ✓ The relationship Prescribed_By is between Prescription and Doctor. The cardinality ratio is 1:n for Doctor to Prescription. Doctor has partial participation whereas Prescription has total participation.

- ✓ The relationship Contains is between Prescription and Drugs. The cardinality ratio is m:n for Prescription to Drugs. Prescription has total participation whereas Drugs has partial participation.

Functional Requirements:-

The database has the following functional requirements:-

The system enables the reception staff to add a new patient record. Other staff in the hospital such as nurses, doctors etc. can modify or update the records of the patient in the database.

The system enables nurses, doctors and other hospital staff to search for a specific patient in the database.

The system automatically creates an entry whenever a patient visits a doctor which contains the date of visit as well as the prescription prescribed by the doctor to the patient.

The system provides the Doctor the functionality of updating Patient Medical-Records on a later day.

The system categorizes the Bill incurred by Patient under Health Insurance or Personal Cost, it also calculates the total amount and creates an entry for the same.

The system provides the functionality of admitting a patient to the hospital that was previously not admitted.

The system provides the functionality to the hospital staff for releasing a patient from the hospital.

The system has the ability to send lab results back to the nurses or doctors. The system enables the Doctor to check Patient history by searching it easily.

In case of a deceased patient, the system modifies the Patient records appropriately.

Scenario Based System Requirement :-

- **Data Removal :-**

The system provides the functionality to remove an entry with incorrectly inserted data.

Eg: In case a doctor determines the lab results to be incorrect, then the Lab test should be performed again and the previous results should be removed from the database.

In case a staff member vacates a job position, the system provides the functionality to remove the entry for the same.

In case the patient is related to any doctor/staff member then bill created must be recreated as the patient may get some concession hence the system provides functionality to remove the entry for the same.

In case a particular drug is replaced by another drug then the system provides functionality to remove the entry for the same.

- **Data Modification:-**

The system provides the functionality for a Doctor to modify patient record based on diagnosis conducted on a later date.

The system provides the functionality to modify the lab equipment details based on the use.

The system provides the functionality to modify the patient insurance details.

The patient attribute admitted can be modified as and when the patient is discharged from the hospital or admitted to the hospital.

- **Data Retrieval :-**

The system provides the functionality to Doctors, Nurses and other staff to retrieve patient records using specific queries.

The system provides the functionality to the Doctors as well as the patients to retrieve the lab test results.

The system provides the functionality to the Patients to retrieve the prescription information and the Drug Details associated with the prescription.

The system provides the functionality to the Patients and the hospital staff to retrieve the current outstanding Bill Details.

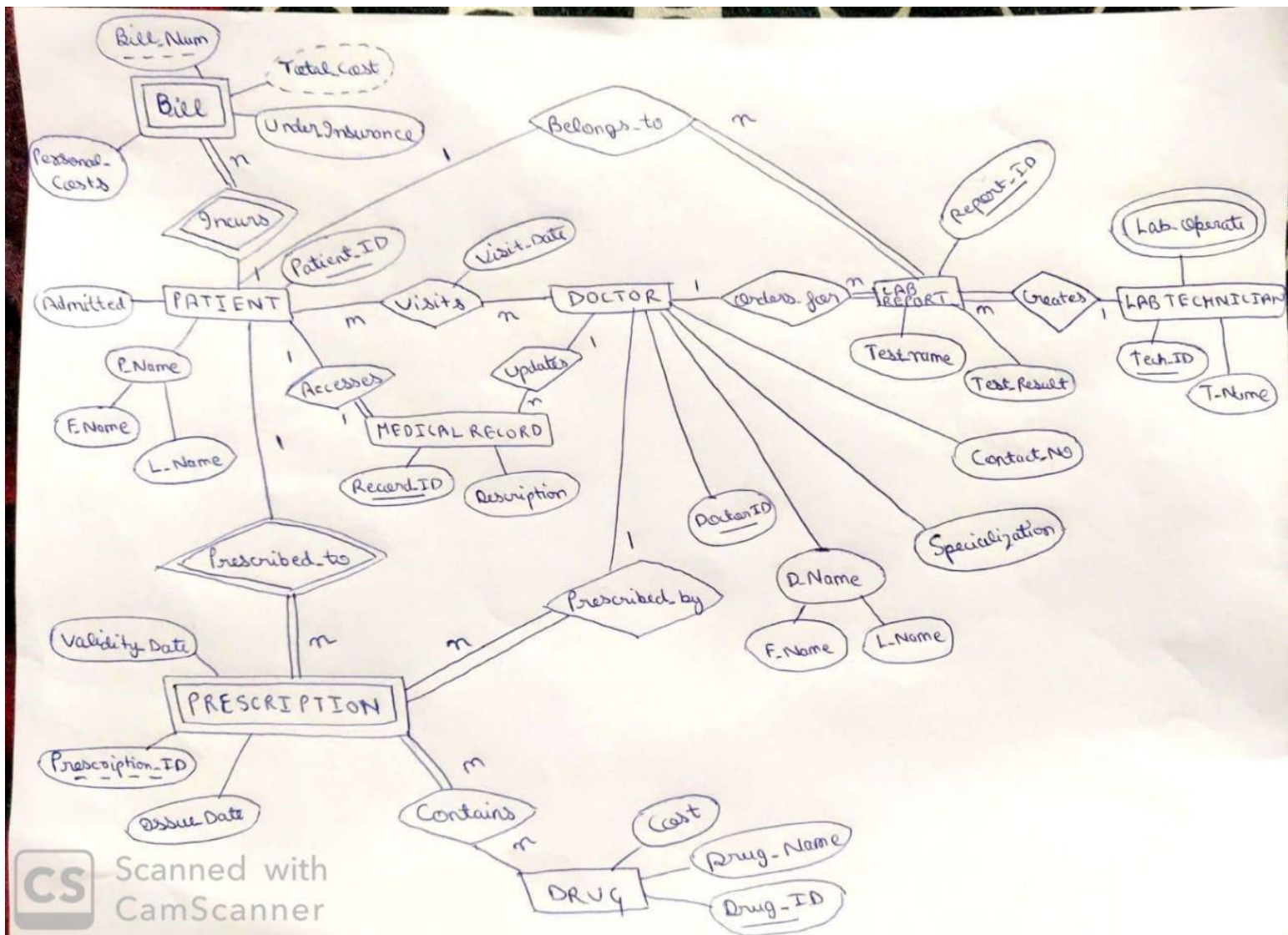
The system provides the functionality to the hospital owner to retrieve the information of any staff member and patient.

The system provides the functionality to the reception to retrieve the information of any doctor and patient (majorly contact information).

The system provides the functionality to the medical shop to retrieve the information of patient records.

The system provides the functionality to the hospital owner to retrieve the information of the money exchange.

3. Draw an ER/EER diagram based on the data requirements. Indicate key constraints, cardinality constraints and participation constraints on the diagram.



4. Convert the ER/EER diagram into a relational database schema diagram.

DOCTOR

<u>Doctor_ID</u>	F_Name	L_Name	Specialization	Contact_No
------------------	--------	--------	----------------	------------

PATIENT

<u>Patient_ID</u>	Admitted	F_Name	L_Name
-------------------	----------	--------	--------

LAB TECHNICIAN

<u>Tech_ID</u>	T_Name
----------------	--------

LAB TECH OPERATE

<u>Tech_ID</u>	<u>Lab_Operate</u>
----------------	--------------------

LAB REPORT

<u>Report_ID</u>	Doctor_ID	Patient_ID	Tech_ID	Test_Name	Test_Result
------------------	-----------	------------	---------	-----------	-------------

BILL

<u>Bill_Num</u>	<u>Patient_ID</u>	Personal_Costs	Under_Insurance
-----------------	-------------------	----------------	-----------------

MEDICAL RECORD

<u>Record_ID</u>	Doctor_ID	Patient_ID	Description
------------------	-----------	------------	-------------

VISITS

Visit_Date	<u>Patient_ID</u>	<u>Doctor_ID</u>
------------	-------------------	------------------

PRESCRIPTION

<u>Prescription_ID</u>	<u>Patient_ID</u>	Validity_Date	Issue_Date	Doctor_ID
------------------------	-------------------	---------------	------------	-----------

DRUG

<u>Drug_ID</u>	Drug_Name	Cost
----------------	-----------	------

CONTAINS

<u>Prescription_ID</u>	<u>Patient_ID</u>	<u>Drug_ID</u>
------------------------	-------------------	----------------

5. Implement the relational database schema incorporating appropriate (based on data requirements) integrity constraints and enter necessary sample data into the tables.

CREATION :-

```
create table doctor(doctor_id number(3) primary key , fname varchar(20) not null, lname varchar(20), specialization varchar(20), contact_no number(10) unique);
```

```
create table patient(patient_id number(3) primary key, fname varchar(20) not null, lname varchar(20), admitted varchar(3));
```

```
create table lab_technician(tech_id number(3) primary key, t_name varchar(30) not null);
```

```
create table lab_tech_operate(lab_operate number(3), tech_id references lab_technician, primary key(lab_operate, tech_id));
```

```
create table Lab_Report(report_id number(4) Primary key , test_name varchar(20) not null, test_result varchar(10), tech_id references lab_technician, doctor_id references doctor, patient_id references patient);
```

```
create table bill(patient_id references patient, bill_num number(4) not null, insurance_costs number(6), personal_costs number(6), constraint prime primary key(patient_id, bill_num));
```

```
create table medical_record(record_id number(4) primary key, description varchar(40), patient_id references patient , doctor_id references doctor );
```

```
create table visits(patient_id references patient, doctor_id references doctor, visit_date date not null, primary key(patient_id, doctor_id));
```

```
create table prescription(patient_id references patient, prescription_id number(4), validity_date date not null, issue_date date not null, doctor_id references doctor ,primary key(patient_id, prescription_id));
```

```
create table drug(drug_id number(3) primary key, drug_name varchar(20) unique, cost number(4));
```

```
create table contains(prescription_id not null, patient_id not null, drug_id references drug, foreign key(patient_id,prescription_id) references prescription(patient_id,prescription_id), primary key(prescription_id, patient_id, drug_id);
```

INSERTION

```
insert into doctor values(21, 'Paras' , 'Dang', 'Neurology' , 8285990923);
insert into doctor values(25, 'Paramartha' , 'Barik', 'Cardiology' ,
9818952671);
insert into doctor values(26, 'Praful' , 'Garg', 'Opthalmology' ,
7268723145);
insert into doctor values(17, 'Parth' , 'kumar', 'Pathology' , 8587416548);
insert into doctor values(19, 'Ujjwal' , 'Garg', 'General Medicine' , 8587281987);
```

```
insert into patient values(100, 'Shreya', 'Kumar', 'yes');
insert into patient values(105, 'Harsh', 'Singh', 'no');
insert into patient values(210, 'Sanskar', 'Banerjee',
'no');
insert into patient values(325, 'Ishika', 'Sen', 'yes');
insert into patient values(192, 'Gauranga', 'Sen', 'no');
```

```
insert into bill values(100, 50, 1156, 230);
insert into bill values(100, 52, 560, 52);
insert into bill values(100, 7, 950, 100);
insert into bill values(105, 10, 550, 20);
insert into bill values(210, 5, 10620, 1200);
insert into bill values(210, 9, 400, 80);
insert into bill values(325, 2, 2200, 1900);
insert into bill values(325, 100, 6700, 400);
insert into bill values(325, 450, 34500, 120);
insert into bill values(325, 5, 250, 50);
insert into bill values(325, 1277, 10100, 5000);
insert into bill values(192, 4300, 0, 550);
insert into bill values(192, 69, 0, 420);
```

```
insert into lab_technician values(1, 'Avik
Kothari');
insert into lab_technician values(2, 'Shivam
Sethi');
insert into lab_technician values(3, 'Gauri
Pal');
insert into lab_technician values(4, 'Shikta
Malik');
insert into lab_technician values(5, 'Niranjan
Shukla');
```

```
insert into lab_tech_operate values(1,  
1);  
insert      into      lab_tech_operate  
values(1, 2);  
insert      into      lab_tech_operate  
values(2, 5);
```

```
insert into lab_tech_operate values(2,
3);
insert into lab_tech_operate values(3,
4);
insert into lab_tech_operate values(3,
5);
insert into lab_tech_operate values(4,
1);
insert into lab_tech_operate values(4,
2);
insert into lab_tech_operate
values(5, 2);
insert into lab_tech_operate values(5,
3);
```

```
insert into lab_report values(101 , 'Blood Sugar', 'High', 1, 25, 100);
insert into lab_report values(102 , 'Heamoglobin Test', 'Normal', 2, 19,
105);
insert into lab_report values(103 , 'Insulin Clearance', 'Normal', 5, 19,
210);
insert into lab_report values(104 , 'Retina Scan', 'Normal', 4, 26, 325);
insert into lab_report values(105 , 'Gastric Fluid', 'Low', 3, 19,
192);
insert into lab_report values(106 , 'Flu Test', 'Positive', 5, 17,
192);
insert into lab_report values(107 , 'Glaucoma Test', 'Negative', 4, 26, 325);
```

```
insert into drug values(674, 'Acetaminophen',
630);
insert into drug values(546, 'Tegretol', 210);
insert into drug values(231, 'Florastor', 745);
insert into drug values(786, 'Pantoprazole',
300);
insert into drug values(096, 'Omeprazole',
110);
```

```
insert into prescription values(100, 920, '12-apr-2018', '12-aug-2018',
21);
insert into prescription values(100, 135, '27-dec-2018', '15-feb-2019',
26);
insert into prescription values(105, 20, '03-aug-2019', '28-nov-2019',
25);
insert into prescription values(210, 8, '06-feb-2019', '05-feb-2020',
19);
insert into prescription values(210, 978, '21-dec-2019', '31-dec-
```

```
2019', 17);  
insert into prescription values(325, 121, '19-jul-2018', '01-sep-2019',  
25);  
insert into prescription values(325, 72, '05-jan-2019', '07-may-2019',  
21);  
insert into prescription values(325, 4221, '28-mar-2019', '04-oct-  
2019', 17);  
insert into prescription values(192, 6213, '11-apr-2019', '29-feb-  
2020', 19);
```

```
insert into visits values(100, 21, '12-apr-  
2018');  
insert into visits values(100, 26, '27-dec-  
2018');  
insert into visits values(105, 25, '10-oct-  
2019');  
insert into visits values(210, 26, '26-dec-  
2019');  
insert into visits values(325, 17, '18-jul-  
2018');
```

```
insert into visits values(325, 19, '05-jan-2019');
insert into visits values(325, 25, '26-mar-2019');
insert into visits values(192, 19, '09-apr-2019');
```

```
insert into medical_record values(3456, 'High blood sugar diagnosed' , 100, 25);
insert into medical_record values(5636, 'Normal Heamoglobin Levels Detected', 105, 19);
insert into medical_record values(8763, 'Insulin Clearance Levels are Normal', 210, 19);
insert into medical_record values(5543, 'Normal Eye pressure levels', 325, 26);
insert into medical_record values(6406, 'Conditions for flu detected' , 192, 17);
```

```
insert into contains values(100, 920, 546);
insert into contains values(100, 920, 674);
insert into contains values(100, 135, 231);
insert into contains values(100, 135, 546);
insert into contains values(100, 135, 096);
insert into contains values(105, 20, 674);
insert into contains values(105, 20, 231);
insert into contains values(210, 8, 546);
insert into contains values(210, 8, 096);
insert into contains values(210, 978, 231);
insert into contains values(210, 978, 546);
insert into contains values(210, 978, 786);
insert into contains values(325, 121, 096);
insert into contains values(325, 72, 096);
insert into contains values(325, 72, 674);
insert into contains values(325, 4221, 096);
insert into contains values(325, 4221, 674);
insert into contains values(325, 4221, 546);
insert into contains values(192, 6213, 231);
```


6. Write down the necessary SQL statements for implementation of functional requirements through SQL query, delete and update statement.

Select:

1. View the medical records of patients, who have been billed the minimum number of times.

```
select * from medical_record where patient_id in (select patient_id from bill group by patient_id having count(*) = (select min(count(*)) from bill group by patient_id));
```

```
SQL> select * from medical_record where patient_id in (select patient_id from bill group by patient_id having count(*) = (select min(count(*)) from bill group by patient_id));
```

RECORD_ID	DESCRIPTION	PATIENT_ID	DOCTOR_ID
5636	Normal Heamoglobin Levels Detected	105	19

2. Display the patient_id, total billed amount of all the patients who have been prescribed the costliest drug

```
select patient_id, sum(insurance_costs + personal_costs) as "Total cost" from bill group by patient_id having patient_id in ( select patient_id from contains where drug_id in ( select drug_id from drug where cost = (select max(cost) from drug)));
```

```
SQL> select patient_id, sum(insurance_costs + personal_costs) as "Total cost" from bill group by patient_id having patient_id in ( select patient_id from contains where drug_id in ( 2 select drug_id from drug where cost = (select max(cost) from drug)));
```

PATIENT_ID	Total cost
100	3048
105	570
192	970
210	12300

3. Display the lab reports which have been ordered by a particular doctor involving a particular lab.

```
select * from lab_report where doctor_id = &docid and tech_id in ( select tech_id from lab_tech_operate where lab_operate=&lid );
```

```
SQL> select * from lab_report where doctor_id = &docid and tech_id in ( 2 select tech_id from lab_tech_operate where lab_operate=&lid );
Enter value for docid: 19
old 1: select * from lab_report where doctor_id = &docid and tech_id in (
new 1: select * from lab_report where doctor_id = 19 and tech_id in (
Enter value for lid: 5
old 2: select tech_id from lab_tech_operate where lab_operate=&lid )
new 2: select tech_id from lab_tech_operate where lab_operate=5 )
```

REPORT_ID	TEST_NAME	TEST_RESUL	TECH_ID	DOCTOR_ID	PATIENT_ID
102	Heamoglobin Test	Normal	2	19	105
105	Gastric Fluid	Low	3	19	192

4. View the patients associated to a particular drug

```
select * from patient where patient_id in (select patient_id from contains where
drug_id in(select drug_id from drug where drug_name = &drug_name));
```

```
SQL> select * from patient where patient_id in (select patient_id from contains where drug_id in(select drug_id from drug where drug_name = &drug_name));
Enter value for drug_name: 'Tegretol'
old 1: select * from patient where patient_id in (select patient_id from contains where drug_id in(select drug_id from drug where drug_name = &drug_name))
new 1: select * from patient where patient_id in (select patient_id from contains where drug_id in(select drug_id from drug where drug_name = 'Tegretol'))
```

PATIENT_ID	FNAME	LNAME	ADM
100	Shreya	Kumar	yes
325	Ishika	Sen	yes
210	Sanskar	Banerjee	no

5. View the patients admitted status.

```
select admitted from patient where fname = &fname and lname = &lname;
```

```
SQL> select admitted from patient where fname = &fname and lname = &lname;
Enter value for fname: 'Shreya'
Enter value for lname: 'Kumar'
old 1: select admitted from patient where fname = &fname and lname = &lname
new 1: select admitted from patient where fname = 'Shreya' and lname = 'Kumar'
```

ADM
yes

6. View the drug_id corresponds to a particular patient

```
select drug_id,patient_id from contains where patient_id in (select patient_id from patient
where fname = &fname and lname = &lname);
```

```
SQL> select drug_id,patient_id from contains where patient_id in (select patient_id from patient where fname = &fname and lname = &lname);
Enter value for fname: 'Shreya'
Enter value for lname: 'Kumar'
old 1: select drug_id,patient_id from contains where patient_id in (select patient_id from patient where fname = &fname and lname = &lname)
new 1: select drug_id,patient_id from contains where patient_id in (select patient_id from patient where fname = 'Shreya' and lname = 'Kumar')
```

DRUG_ID	PATIENT_ID
96	100
231	100
546	100
546	100
674	100

7. View test result of a particular patient.

```
select patient_id,test_result from lab_report where patient_id in(select patient_id from patient
where fname = &fname and lname = &lname);
```

```
SQL> select patient_id,test_result from lab_report where patient_id in(select patient_id from patient where fname = &fname and lname = &lname);
Enter value for fname: 'Harsh'
Enter value for lname: 'Singh'
old 1: select patient_id,test_result from lab_report where patient_id in(select patient_id from patient where fname = &fname and lname = &lname)
new 1: select patient_id,test_result from lab_report where patient_id in(select patient_id from patient where fname = 'Harsh' and lname = 'Singh')

PATIENT_ID TEST_RESUL
-----
105 Normal
```

8. View the description of a particular patient.

```
select patient_id,description from medical_record where patient_id in(select patient_id from
patient where fname = &fname and lname = &lname);
```

```
SQL> select patient_id,description from medical_record where patient_id in(select patient_id from patient where fname = &fname and lname = &lname);
Enter value for fname: 'Harsh'
Enter value for lname: 'Singh'
old 1: select patient_id,description from medical_record where patient_id in(select patient_id from patient where fname = &fname and lname = &lname)
new 1: select patient_id,description from medical_record where patient_id in(select patient_id from patient where fname = 'Harsh' and lname = 'Singh')

PATIENT_ID DESCRIPTION
-----
105 Normal Heamoglobin Levels Detected
```

Update:

1. Update the prescribed drug “Acetaminophen” to “Omeprazole” for the patients whose lab report showed Normal Hemoglobin and has visited doctor “Ujjwal”

update contains

```
set drug_id = (select drug_id from drug where drug_name='Omeprazole')
where drug_id = (select drug_id from drug where drug_name='Acetaminophen') and
patient_id = (
select patient_id from patient where patient_id in (
select patient_id from lab_report where test_name='Hemoglobin Test'
and test_result='Normal' and doctor_id = (
select doctor_id from doctor where fname='Ujjwal'))
);
```

```
SQL> update contains
2 set drug_id = (select drug_id from drug where drug_name='Omeprazole')
3 where drug_id = (select drug_id from drug where drug_name='Acetaminophen') and
4 patient_id = (
5 select patient_id from patient where patient_id in (
6 select patient_id from lab_report where test_name='Hemoglobin Test'
7 and test_result='Normal' and doctor_id = (
8 select doctor_id from doctor where fname='Ujjwal'))
9 );
```

1 row updated.

```
SQL> select * from contains where patient_id = 105;
```

PATIENT_ID	PRESCRIPTION_ID	DRUG_ID
105	20	96
105	20	231

2. Update the doctor name associated to the medical_record table.

update medical_record

```
set doctor_id = (select doctor_id from doctor where fname = 'Praful' and lname = 'Garg') where doctor_id =
(select doctor_id from doctor where fname = 'Paramartha' and lname = 'Barik');
```

```
SQL> update medical_record
2 set doctor_id = (select doctor_id from doctor where fname = 'Praful' and lname = 'Garg') where doctor_id = (select doctor_id from doctor where fname = 'Paras' and lname = 'Dang');
```

0 rows updated.

```
SQL> update medical_record
2 set doctor_id = (select doctor_id from doctor where fname = 'Praful' and lname = 'Garg') where doctor_id = (select doctor_id from doctor where fname = 'Paramartha' and lname = 'Barik');
```

1 row updated.

```
SQL> select * from medical_record;
```

RECORD_ID	DESCRIPTION	PATIENT_ID	DOCTOR_ID
3456	High blood sugar diagnosed	100	26
6406	Conditions for flu detected	192	17
5636	Normal Hemoglobin Levels Detected	105	19
8763	Insulin Clearance Levels are Normal	210	19
5543	Normal Eye pressure levels	325	26

3. update technician Shivam Sethi to technician Avik Kothari.

```
update lab_report
```

```
set tech_id = (select tech_id from lab_technician where t_name = 'Avik Kothari') where tech_id = (select tech_id from lab_technician where t_name = 'Shivam Sethi');
```

```
SQL> update lab_report
  2  set tech_id = (select tech_id from lab_technician where t_name = 'Avik Kothari') where tech_id = (select tech_id from lab_technician where t_name = 'Shivam Sethi');

1 row updated.
```

```
SQL> select * from lab_report;
```

REPORT_ID	TEST_NAME	TEST_RESUL	TECH_ID	DOCTOR_ID	PATIENT_ID
101	Blood Sugar	High	1	25	100
107	Glaucoma Test	Negative	4	26	325
104	Retina Scan	Normal	4	26	325
102	Heamoglobin Test	Normal	1	19	105

4. Update doctor Praful garg contact number.

```
update doctor
```

```
set contact_no = 9675793271 where doctor_id = 26;
```



```
9818952671
      26 Praful      Garg      Ophthalmology
7268723145

DOCTOR_ID FNAME      LNAME      SPECIALIZATION
-----
CONTACT_NO
-----
      17 Parth      kumar      Pathology
8587416548

      19 Ujjwal      Garg      General Medicine
8587281987

SQL> update doctor
  2 set contact_no = 9675793271 where doctor_id = 26;

1 row updated.

SQL> select * from doctor;

DOCTOR_ID FNAME      LNAME      SPECIALIZATION
-----
CONTACT_NO
-----
      21 Paras      Dang      Neurology
8285990923

      25 Paramartha    Barik      Cardiology
9818952671

      26 Praful      Garg      Ophthalmology
9675793271

DOCTOR_ID FNAME      LNAME      SPECIALIZATION
-----
CONTACT_NO
-----
      17 Parth      kumar      Pathology
8587416548

      19 Ujjwal      Garg      General Medicine
8587281987
```

Delete:

1. Delete the lab report of the patient whose admitted status is 'no' and has referred to at most two doctors.

```
delete from lab_report where patient_id in (
select patient_id from patient where admitted='no'
intersect
select patient_id from visits group by patient_id having count(*) <= 2);
```

```
SQL> Select report_id from lab_report where patient_id in (  
  2  select patient_id from patient where admitted='no'  
  3  intersect  
  4  select patient_id from visits group by patient_id having count(*) <= 2);
```

REPORT_ID

```
-----  
      102  
      103  
      105  
      106
```

```
SQL> delete from lab_report where patient_id in (  
  2  select patient_id from patient where admitted='no'  
  3  intersect  
  4  select patient_id from visits group by patient_id having count(*) <= 2);
```

4 rows deleted.

```
SQL> Select report_id from lab_report where patient_id in (  
  2  select patient_id from patient where admitted='no'  
  3  intersect  
  4  select patient_id from visits group by patient_id having count(*) <= 2);
```

no rows selected

2. Delete drugs which cost 300.

Delete from contains where drug_id in(select drug_id from drug where cost = 300);

19 rows selected.

```
SQL> Delete from contains where drug_id in(select drug_id from drug where cost = 300);
```

1 row deleted.

```
SQL> select * from contains;
```

PATIENT_ID	PRESCRIPTION_ID	DRUG_ID
210	8	96
210	8	546
105	20	231
105	20	674
325	72	96
325	72	674
325	121	96
100	135	96
100	135	231
100	135	546
100	920	546

PATIENT_ID	PRESCRIPTION_ID	DRUG_ID
100	920	674
210	978	231
210	978	546
325	4221	96
325	4221	546
325	4221	674
192	6213	231

18 rows selected.

3. Delete medical record of high blood sugar diagnosed patients.

```
Delete from medical_record where record_id in(select record_id from medical_record where description = 'High blood sugar diagnosed');
```

```
SQL> Delete from medical_record where record_id in(select record_id from medical_record where description = 'High blood sugar diagnosed');
```

1 row deleted.

```
SQL> select * from medical_record;
```

RECORD_ID	DESCRIPTION	PATIENT_ID	DOCTOR_ID
5636	Normal Heamoglobin Levels Detected	105	19
8763	Insulin Clearance Levels are Normal	210	19

4. Delete drug name Florastor

```
delete from contains where drug_id in (select drug_id from drug where drug_name = 'Florastor');
```

PATIENT_ID	PRESCRIPTION_ID	DRUG_ID
100	920	674
210	978	231
210	978	546
325	4221	96
325	4221	546
325	4221	674
192	6213	231

18 rows selected.

```
SQL> delete from contains where drug_id in (select drug_id from drug where drug_name = 'Florastor');
delete from contains where drug_id in (select drug_id from drug where drug_name = 'Florastor'
*
```

ERROR at line 1:
ORA-00907: missing right parenthesis

```
SQL> delete from contains where drug_id in (select drug_id from drug where drug_name = 'Florastor');
```

4 rows deleted.

```
SQL> select * from contains;
```

PATIENT_ID	PRESCRIPTION_ID	DRUG_ID
210	8	96
210	8	546
105	20	674
325	72	96
325	72	674
325	121	96
100	135	96
100	135	546
100	920	546
100	920	674
210	978	546

PATIENT_ID	PRESCRIPTION_ID	DRUG_ID
325	4221	96
325	4221	546
325	4221	674

14 rows selected.

7. Define and implement two PL/SQL function and two PL/SQL procedure involving cursor for the database under consideration.

1. Procedure

1. Print the total cost incurred by a patient and the percentage of it covered under personal

```
Set serveroutput on;
create or replace procedure print_cost(fn in varchar, ln in varchar) is
```

```

cursor bill_crs is select * from bill where patient_id = (select patient_id from patient where fname=fn and
lname=ln);
cost number(7) := 0;
p_cost number(7) := 0;
percentage number(3) := 0.00;
BEGIN
for bill_row in bill_crs LOOP
cost := cost + bill_row.insurance_costs + bill_row.personal_costs;
p_cost := p_cost + bill_row.personal_costs;
END LOOP;
percentage := (p_cost/cost*100);
dbms_output.put_line('Total cost incurred is ' || to_char(cost));
dbms_output.put_line('Percentage covered under personal cost ' || to_char(percentag));
END;

```

```

SQL> Set serveroutput on;
SQL> create or replace procedure print_cost(fn in varchar, ln in varchar) is
  2 cursor bill_crs is select * from bill where patient_id = (select patient_id from patient where fname=fn and lname=ln);
  3 cost number(7) := 0;
  4 p_cost number(7) := 0;
  5 percentage number(3) := 0.00;
  6 BEGIN
  7 for bill_row in bill_crs LOOP
  8 cost := cost + bill_row.insurance_costs + bill_row.personal_costs;
  9 p_cost := p_cost + bill_row.personal_costs;
 10 END LOOP;
 11 percentage := (p_cost/cost*100);
 12 dbms_output.put_line('Total cost incurred is ' || to_char(cost));
 13 dbms_output.put_line('Percentage covered under personal cost ' || to_char(percentage));
 14 END;
 15 /

Procedure created.

SQL> exec print_cost('Harsh', 'Singh');
Total cost incurred is 570
Percentage covered under personal cost 4

PL/SQL procedure successfully completed.

SQL>

```

2. Print the doctor ID associated to a particular patient.

```

Set serveroutput on;
create or replace procedure print_doctor_id(fn in varchar, ln in varchar) is
cursor dcid_crs is select * from visits where patient_id = (select patient_id from patient where fname=fn
and lname=ln);
dc_id number(4) := 0;
BEGIN
for dcid_row in dcid_crs LOOP
dc_id := dc_id + dcid_row.doctor_id; END
LOOP;
dbms_output.put_line('Doctor_id ' || to_char(dc_id));
END;

SQL> Set serveroutput on;
SQL> create or replace procedure print_doctor_id(fn in varchar, ln in varchar) is
  2 cursor dcid_crs is select * from visits where patient_id = (select patient_id from patient where fname=fn and lname=ln);
  3 dc_id number(4) := 0;
  4 BEGIN
  5 for dcid_row in dcid_crs LOOP
  6 dc_id := dc_id + dcid_row.doctor_id;
  7 END LOOP;
  8 dbms_output.put_line('Doctor_id ' || to_char(dc_id));
  9 END;
 10 /

Procedure created.

SQL> exec print_doctor_id('Harsh', 'Singh');
Doctor_id 25

PL/SQL procedure successfully completed.

```

2. Function

1. From the bill of a particular patient, obtain the net cost incurred by him/her for everything apart from drugs, such as visitation-fees and lab-test fees, etc.

Set serveroutput on;

```

create or replace function cost_extra(fn in varchar, ln in varchar) return number is
cursor bill_crs is select * from bill where patient_id = (select patient_id from patient where fname=fn and lname=ln);
cursor drug_crs is select * from contains where patient_id = (select patient_id from patient where fname=fn and lname=ln);
total_cost number(7) := 0;
drug_cost number(5) := 0;
temp_cost drug.cost%type;
BEGIN
for bill_row in bill_crs LOOP
total_cost := total_cost + bill_row.insurance_costs + bill_row.personal_costs; END
LOOP;
for drug_row in drug_crs LOOP
select cost into temp_cost from drug where drug_id = drug_row.drug_id;
drug_cost := drug_cost + temp_cost;
END LOOP;
return (total_cost-drug_cost);
END;

```

```

SQL> Set serveroutput on;
SQL> create or replace function cost_extra(fn in varchar, ln in varchar) return number is
  2 cursor bill_crs is select * from bill where patient_id = (select patient_id from patient where fname=fn and lname=ln);
  3 cursor drug_crs is select * from contains where patient_id = (select patient_id from patient where fname=fn and lname=ln);
  4 total_cost number(7) := 0;
  5 drug_cost number(5) := 0;
  6 temp_cost drug.cost%type;
  7 BEGIN
  8 for bill_row in bill_crs LOOP
  9 total_cost := total_cost + bill_row.insurance_costs + bill_row.personal_costs;
10 END LOOP;
11 for drug_row in drug_crs LOOP
12 select cost into temp_cost from drug where drug_id = drug_row.drug_id;
13 drug_cost := drug_cost + temp_cost;
14 END LOOP;
15 return (total_cost-drug_cost);
16 END;
17 /

Function created.

SQL> variable cost number;
SQL> exec :cost := cost_extra('Harsh', 'Singh');

PL/SQL procedure successfully completed.

SQL> print cost;

      COST
-----
      -805

SQL> exec :cost := cost_extra('Gauranga', 'Sen');

PL/SQL procedure successfully completed.

SQL> print cost;

      COST
-----
       225

```

2. Print the report_id of the patient.

```

Set serveroutput on;
create or replace function print_report_id(fn in varchar, ln in varchar) return number is
cursor rp_id_crs is select * from lab_report where patient_id = (select patient_id from patient where fname=fn and lname=ln);
rp_id lab_report.report_id%type := 0;
BEGIN
for rp_id_row in rp_id_crs LOOP
rp_id := rp_id + rp_id_row.report_id;
END LOOP;
return (rp_id);
END;

```

```
SQL> Set serveroutput on;
SQL> create or replace function print_report_id(fn in varchar, ln in varchar) return number is
  2 cursor rpid_crs is select * from lab_report where patient_id = (select patient_id from patient where fname=fn and lname=ln);
  3 rp_id lab_report.report_id%type := 0;
  4 BEGIN
  5 for rpid_row in rpid_crs LOOP
  6 rp_id := rp_id + rpid_row.report_id;
  7 END LOOP;
  8 return (rp_id);
  9 END;
10 /

Function created.

SQL> variable report_id_patient number;
SQL> exec :report_id_patient := print_report_id('Ishika', 'Sen');

PL/SQL procedure successfully completed.

SQL> print report_id_patient;

REPORT_ID_PATIENT
-----
                211
```

THANKS