**HOSPITAL DATABASE MANAGEMENT SYSTEM**

UCS310 -DBMS Project Report

EndSem Lab Evaluation

Submitted By :

Sanchit Raj (102116112)

Kushal (102116097) Raghav Sharma(102116102) Namit Sharma(102116105)

Submitted To: Dr.Manisha Kaushal

Assistant Professor, TIET

1

Contents

|  |  |  |
| --- | --- | --- |
| **S.no** | **Topic** | **Page Number** |
| **1.** | Introduction | 3-4 |
| **2.** | Objectives of the system | 4 |
| **3.** | ER diagram: Hospital  Database | 5-8 |
| **4.** | Physical Schema | 9-11 |
| **5.** | SQL commands | 12-17 |
| **6.** | Normalization | 18-20 |
| **7.** | PL/SQL Queries | 21-24 |
| **8.** | Conclusion | 24 |
| **9.** | References | 25 |

**HOSPITAL MANAGEMENT SYSTEM**

# Introduction: -

Due to increasing population and due to the emergence of new diseases and the pandemic season has led to the opening of new hospitals in various cities for providing quality health care and facilities to the patients.

Hospitals are the essential part of our lives, providing best medical facilities to people suffering from various ailments, which may be due to change in climatic conditions, increased work-load, emotional trauma, stress etc. It is necessary for the hospitals to keep track of its day-to-day activities & records of its patients, doctors, nurses, ward boys and other staff personnel that keep the hospital running smoothly & successfully.

But keeping track of all the activities and their records on paper is very cumbersome and error prone. It also is very inefficient and a time-consuming process Observing the continuous increase in population and number of people visiting the hospital. Recording and maintaining all these records are highly unreliable, inefficient, and error-prone. It is also not economically & technically feasible to maintain these records on paper.

Thus, keeping the working of the manual system as the basis of our project. We have developed an automated version of the manual system, named “Hospital Management System.”

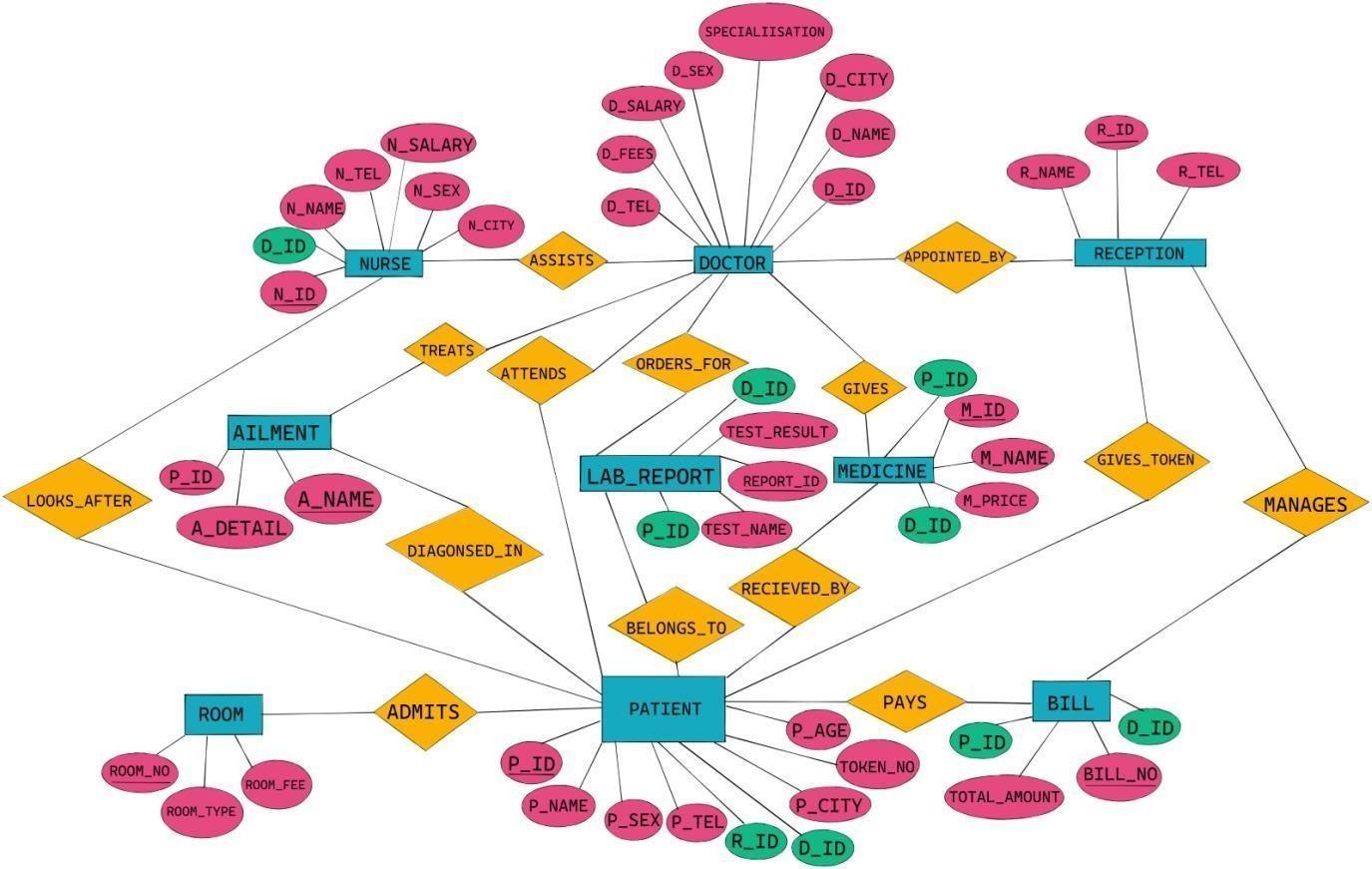
The main aim of our project is to provide a paperless hospital up to 90%. It also aims at providing low-cost reliable automation of the existing systems. The system also provides excellent security of data at every level of user-system interaction and provides robust & reliable storage and backup facilities.

1. Objectives of the system: -

The project “Hospital management system” is aimed to develop to maintain the day–to-day state of admission/discharge of patients, list of doctors, reports generation, etc. It is designed to achieve the following objectives:

1. To computerise all details regarding patient details & hospital details.
2. Scheduling the appointment of patients with doctors to make it convenient for both.
3. Scheduling the services of specialised doctors and emergencies properly so that facilities provided by hospitals are fully utilised in an effective and efficient manner.
4. If the medical store issues medicines to patients, it should reduce the stock status of the medical store and vice-versa.
5. It should be able to handle the test reports of patients conducted in the pathology lab of the hospital.
6. The inventory should be updated automatically whenever a transaction is made.
7. The information of the patients should be kept up to date and their records should be kept in the system for historical purposes.

# ER Diagram: Hospital Database: -



***Fig. 2.3 : ER Diagram***

* 1. **Patient’s** ID ‘P\_ID’ would be the key attribute used to uniquely identify the patient including their name, sex, street, city, contact number, date of birth and admission into hospital.
  2. **Doctor’s** ID ‘D\_ID’ would be the key attribute used to uniquely identify the doctor including their name, contact number, service fee, salary, gender, and their respective specialisations.
  3. If **Medicine** information is stored in the database as per prescriptions prescribed by the doctors to every patient, redundancy is bound to arise. To prevent the same, drugs are stored in a different relation than prescription i.e.. medicine information contains the ID, ‘M\_ID’, the key attribute, M\_NAME and the price(M\_PRICE) of the respective medicines.
  4. **Bill** (weak entity type) for a particular patient is identified by the ‘BILL\_NO.’ as a partial key and ‘TOTAL\_AMOUNT’ which describe the total amount billed.
  5. **Nurse** information includes ID, ‘N\_ID’, the key attribute used to uniquely identify the nurse including name, contact details and their gender.
  6. **Room** information for a specific patient includes ‘ROOM\_NO’ the key attribute used to uniquely identify the room, room type and the room fee.
  7. **Aliment** (weak entity type) includes information under a particular patient using ‘A\_NAME’ to identify the ailment and aliment details.
  8. **Reception** information includes ID ‘R\_ID’ the key attribute used to uniquely identify the reception, reception-name, and reception contact details.
  9. **LAB\_REPORT** information includes ‘REPORT\_ID’,‘TEST\_NAME’ and ‘TEST\_RESULT’.

**APPOINTED\_BY**: between the Doctor and the Reception and contains attribute ‘time’ which stores the time of appointment. The cardinality ratio is 1: n from Reception to doctor. Both have total participation.

**LOOKS\_AFTER:** between Nurse and Patient. The cardinality ratio m: n for Nurse to Patient. There is partial participation with Nurse and total with patient.

**ASSISTS:** between Doctor and Nurse which shows a nurse assists a doctor to reduce his work. The cardinality ration is 1: n from doctor to nurse. Both have total participation.

**TREATS:** between Doctor and the Ailment i.e., Doctor scrutinises and treats the ailment and takes necessary steps to treat it. The cardinality ratio is m: n. There is total participation with doctor and partial participation with ailment.

**GIVES:** is between Medicine and Doctor. The cardinality ratio is n: 1 for Doctor to Medicine. Both have total participation.

**ATTENDS**: between Doctor and Patient. The cardinality ratio is 1: n for Doctor to Patient. Both have total participation.

**GIVES\_TOKEN:** between Reception and Patient. The cardinality ratio is 1: n for Reception to Patient. Both have total participation.

**MANAGES:** between Reception and Bill. The cardinality ratio is 1: n for Reception to Bill. Both entities have total participation.

**ADMITS**: In between Patient and Room to store which room is allotted to the patient. The cardinality ratio is m: n. There is partial participation with Patient and total participation with the entity type room.

**ORDERS\_FOR**: is between Doctor and LAB\_REPORT with cardinality ratio 1: n from doctor to LAB\_REPORT.

**BELONGS\_TO**: between LAB\_REPORT and Patient with cardinality ratio 1: n from patient to LAB\_REPORT.

**PAYS:** between the weak entity type Bill and the strong entity type Patient. The cardinality ratio is 1: n for Patient to Bill.

**RECEIVED\_BY**: between Medicine and Patient. The cardinality ratio is 1: n for Medicine to Patient. Both have total participation.

**DIAGONSED\_IN**: between the weak entity type Ailment and strong entity type Patient. The cardinality ratio is 1: n for Patient to Ailment. Both have total participation.

**Pays:** between the weak entity type Bill and the strong entity type Patient. The cardinality ratio is 1: n for Patient to Bill.

# Physical Schema: -

**Table Name: NURSE**

|  |  |  |
| --- | --- | --- |
| **FIELDS** | **DATA TYPE** | **RELATIONSHIPS** |
| N\_ID | NUMBER (5) | PRIMARY KEY |
| D\_ID | NUMBER (5) | FOREIGN KEY |
| N\_NAME | VARCHAR (20) | NOT NULL |
| N\_SEX | VARCHAR (2) | NOT NULL |
| N\_TEL | NUMBER (10) | NOT NULL |
| N\_CITY | VARCHAR (10) | NOT NULL |
| N\_SALARY | NUMBER (10) | NOT NULL |

**Table Name: DOCTOR**

|  |  |  |
| --- | --- | --- |
| **FIELDS** | **DATA TYPE** | **RELATIONSHIPS** |
| D\_ID | NUMBER (5) | PRIMARY KEY |
| D\_NAME | VARCHAR (20) | NOT NULL |
| D\_SEX | VARCHAR (2) | NOT NULL |
| D\_TEL | NUMBER (10) | NOT NULL |
| D\_CITY | VARCHAR (10) | NOT NULL |
| D\_SALARY | NUMBER (10) | NOT NULL |
| D\_FEES | NUMBER (10) | NOT NULL |
| SPECIALISATION | VARCHAR (50) | NOT NULL |

**Table Name: RECEPTION**

|  |  |  |
| --- | --- | --- |
| **FIELDS** | **DATA TYPE** | **RELATIONSHIPS** |
| R\_ID | NUMBER (5) | PRIMARY KEY |
| R\_NAME | VARCHAR (20) | NOT NULL |
| R\_TEL | NUMBER (10) | NOT NULL |

**Table Name: PATIENT**

|  |  |  |
| --- | --- | --- |
| **FIELDS** | **DATA TYPE** | **RELATIONSHIPS** |
| P\_ID | NUMBER (5) | PRIMARY KEY |
| P\_NAME | VARCHAR (20) | NOT NULL |
| P\_AGE | NUMBER (10) | NOT NULL |
| P\_SEX | VARCHAR (2) | NOT NULL |
| P\_TEL | NUMBER (10) | NOT NULL |
| P\_CITY | VARCHAR (10) | NOT NULL |
| D\_ID | NUMBER (5) | FOREIGN KEY |
| R\_ID | NUMBER (5) | FOREIGN KEY |
| TOKEN\_NO | NUMBER (5) | NOT NULL |

**Table Name: BILL**

|  |  |  |
| --- | --- | --- |
| **FIELDS** | **DATA TYPE** | **RELATIONSHIPS** |
| BILL\_NO | NUMBER (10) | PRIMARY KEY |
| P\_ID | NUMBER (5) | PRIMARY KEY |
| D\_ID | NUMBER (5) | PRIMARY KEY |
| TOTAL\_AMOUNT | NUMBER (10) | NOT NULL |

**Table Name: ROOM**

|  |  |  |
| --- | --- | --- |
| **FIELDS** | **DATA TYPE** | **RELATIONSHIPS** |

|  |  |  |
| --- | --- | --- |
| ROOM\_NO. | NUMBER (5) | PRIMARY KEY |

|  |  |  |
| --- | --- | --- |
| ROOM\_FEE | NUMBER (5) | NOT NULL |
| ROOM\_TYPE | VARCHAR (20) | NOT NULL |

**Table Name: AILMENT**

|  |  |  |
| --- | --- | --- |
| **FIELDS** | **DATA TYPE** | **RELATIONSHIPS** |
| A\_NAME | VARCHAR (50) | PRIMARY KEY |
| P\_ID | NUMBER (5) | PRIMARY KEY |
| A\_DETAIL | VARCHAR (50) | NOT NULL |

**Table Name: LAB\_REPORT**

|  |  |  |
| --- | --- | --- |
| **FIELDS** | **DATA TYPE** | **RELATIONSHIPS** |
| REPORT\_ID | NUMBER (5) | PRIMARY KEY |
| TEST\_NAME | VARCHAR (20) | NOT\_NULL |
| TEST\_RESULT | VARCHAR (10) | NOT NULL |
| D\_ID | NUMBER (5) | FOREIGN KEY |
| P\_ID | NUMBER (5) | FOREIGN KEY |

**Table Name: MEDICINE**

|  |  |  |
| --- | --- | --- |
| **FIELDS** | **DATA TYPE** | **RELATIONSHIPS** |
| M\_ID | NUMBER (5) | PRIMARY KEY |
| M\_NAME | VARCHAR (20) | NOT NULL |
| M\_PRICE | NUMBER (5) | NOT NULL |
| D\_ID | NUMBER (5) | FOREIGN KEY |
| P\_ID | NUMBER (5) | FOREIGN KEY |

# SQL Commands: -

**CREATE TABLE DOCTOR ( D\_ID NUMBER (5), D\_NAME VARCHAR (20),**

**D\_SEX VARCHAR (2),**

**D\_TEL NUMBER (10),**

**D\_CITY VARCHAR (10),**

**SPECIALISATION VARCHAR (50),**

**D\_FEES NUMBER (10),**

**D\_SALARY NUMBER (10),**

**CONSTRAINT PK\_DOCTOR PRIMARY KEY(D\_ID));**



**CREATE TABLE NURSE ( N\_ID NUMBER (5), N\_NAME VARCHAR (20),**

**N\_SEX VARCHAR (2),**

**N\_TEL NUMBER (10),**

**N\_CITY VARCHAR (10),**

**N\_SALARY NUMBER (10),**

**D\_ID NUMBER (5),**

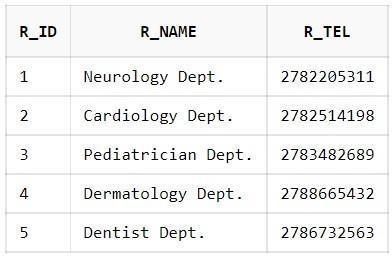
**CONSTRAINT PK\_NURSE PRIMARY KEY (N\_ID), CONSTRAINT FK\_NURSE FOREIGN KEY(D\_ID)** **REFERENCES DOCTOR);**



-

**CREATE TABLE RECEPTION ( R\_ID NUMBER (5), R\_NAME VARCHAR (20),**

**R\_TEL NUMBER (10),**

**CONSTRAINT PK\_RECEPTION PRIMARY KEY(R\_ID));** 

**CREATE TABLE PATIENT ( P\_ID NUMBER (5), P\_NAME VARCHAR (20),**

**P\_AGE NUMBER (10),**

**P\_SEX VARCHAR (2),**

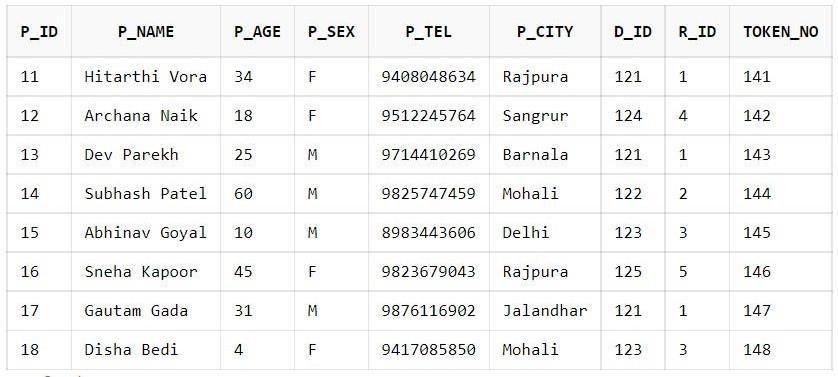
**P\_TEL NUMBER (10),**

**P\_CITY VARCHAR (10), D\_ID NUMBER (5), R\_ID NUMBER (5),**

**TOKEN\_NO NUMBER (5),**

**CONSTRAINT PK\_PATIENT PRIMARY KEY(P\_ID),**

**CONSTRAINT FK\_P1 FOREIGN KEY(D\_ID) REFERENCES DOCTOR(D\_ID), CONSTRAINT FK\_P2 FOREIGN KEY(R\_ID) REFERENCES RECEPTION(R\_ID));**



**CREATE TABLE MEDICINE ( M\_ID NUMBER (5), M\_NAME VARCHAR (20),**

**M\_PRICE NUMBER (5),**

**D\_ID NUMBER (5),**

**P\_ID NUMBER (5),**

**CONSTRAINT PK\_MEDICINE PRIMARY KEY(M\_ID),**

**CONSTRAINT FK\_MEDICINE FOREIGN KEY (D\_ID) REFERENCES DOCTOR,**

**CONSTRAINT FK\_MEDICINE\_PATIENT FOREIGN KEY (P\_ID) REFERENCES PATIENT);**

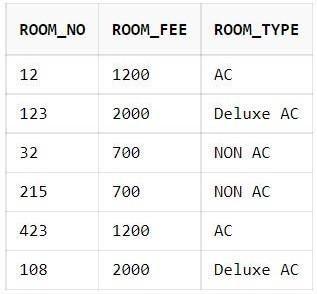


**CREATE TABLE ROOM ( ROOM\_NO NUMBER (5),**

**ROOM\_FEE NUMBER (5),**

**ROOM\_TYPE VARCHAR (20),**

**CONSTRAINT PK\_ROOM PRIMARY KEY(ROOM\_NO));**



**CREATE TABLE LAB\_REPORT ( REPORT\_ID NUMBER (5), TEST\_NAME VARCHAR (20),**

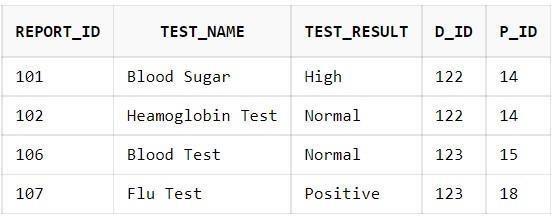
**TEST\_RESULT VARCHAR (10),**

**D\_ID NUMBER (5),**

**P\_ID NUMBER (5),**

**CONSTRAINT PK\_REPORT PRIMARY KEY(REPORT\_ID),**

**CONSTRAINT FK\_REPORT FOREIGN KEY (D\_ID) REFERENCES DOCTOR, CONSTRAINT FK\_REPORT\_PATIENT FOREIGN KEY (P\_ID) REFERENCES PATIENT);**



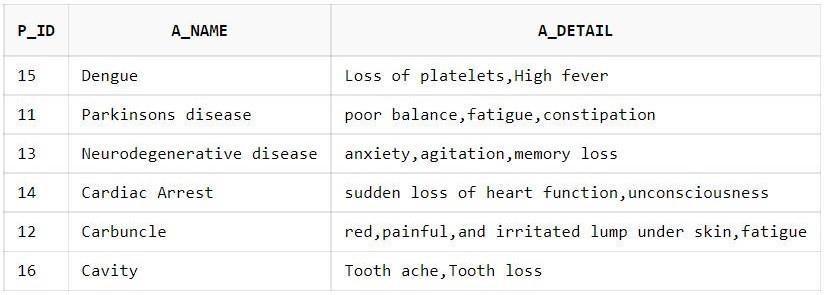
**CREATE TABLE AILMENT ( P\_ID NUMBER (5),**

**A\_NAME VARCHAR (50),**

**A\_DETAIL VARCHAR (50),**

**CONSTRAINT PK\_AILMENT PRIMARY KEY (P\_ID, A\_NAME),**

**CONSTRAINT FK\_AILMENT FOREIGN KEY (P\_ID) REFERENCES PATIENT);**

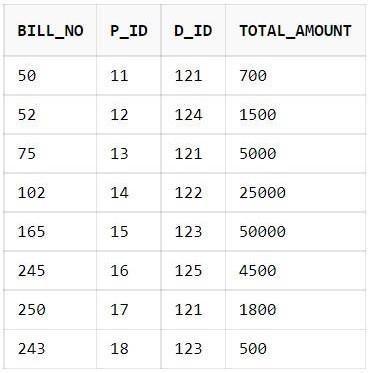


**CREATE TABLE BILL**

**( BILL\_NO NUMBER (10), P\_ID REFERENCES PATIENT,**

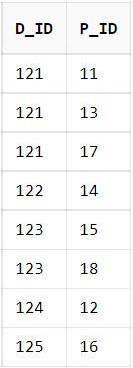
**D\_ID REFERENCES DOCTOR, TOTAL\_AMOUNT NUMBER (10),**

**CONSTRAINT PK\_BILL PRIMARY KEY (P\_ID, BILL\_NO, D\_ID));**



**CREATE TABLE TREATS ( D\_ID NUMBER (5), P\_ID NUMBER (5),**

**CONSTRAINT FK1\_TREATS FOREIGN KEY(D\_ID) REFERENCES DOCTOR(D\_ID), CONSTRAINT FK2\_TREATS FOREIGN KEY(P\_ID) REFERENCES PATIENT(P\_ID), CONSTRAINT PK\_TREATS PRIMARY KEY (D\_ID, P\_ID));**

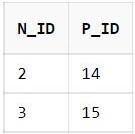


**CREATE TABLE LOOKS\_AFTER ( N\_ID NUMBER (5),**

**P\_ID NUMBER (5),**

**CONSTRAINT FK\_NP1 FOREIGN KEY(P\_ID) REFERENCES PATIENT(P\_ID), CONSTRAINT FK\_NP2 FOREIGN KEY(N\_ID) REFERENCES**

**NURSE(N\_ID), CONSTRAINT PK\_LOOKS\_AFTER PRIMARY KEY (N\_ID, P\_ID));**

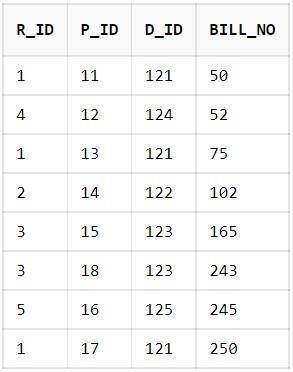


**CREATE TABLE MANAGES ( R\_ID NUMBER (5), P\_ID NUMBER (5), D\_ID NUMBER (5),**

**BILL\_NO NUMBER (10),**

**CONSTRAINT FK\_MANAGES1 FOREIGN KEY(R\_ID) REFERENCES RECEPTION(R\_ID), CONSTRAINT FK\_MANAGESBILL FOREIGN KEY (BILL\_NO, P\_ID, D\_ID) REFERENCES BILL (BILL\_NO, P\_ID, D\_ID),**

**CONSTRAINT PK\_MANAGES PRIMARY KEY (BILL\_NO, P\_ID, R\_ID, D\_ID));**



**CREATE TABLE ADMITS ( ROOM\_NO NUMBER (5), P\_ID NUMBER (5), DATE\_OF\_ADMISSION DATE,**

**CONSTRAINT FK\_PRO1 FOREIGN KEY(P\_ID) REFERENCES PATIENT(P\_ID),**

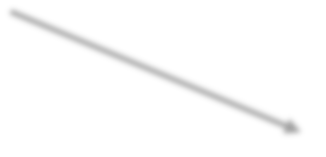
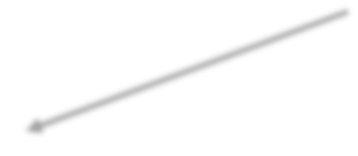
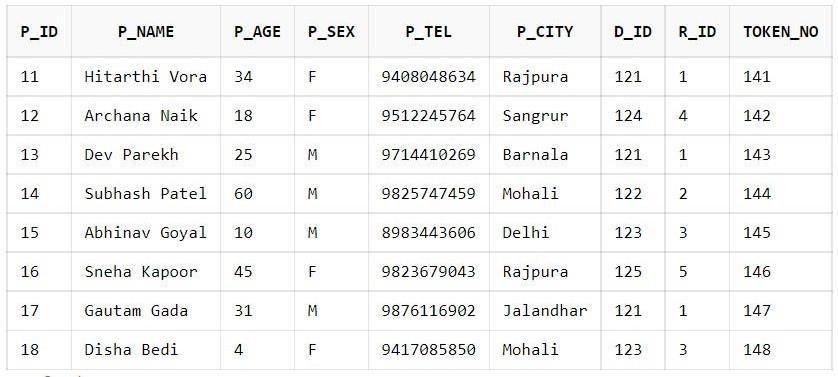
**CONSTRAINT FK\_PRO2 FOREIGN KEY(ROOM\_NO) REFERENCES**

**ROOM(ROOM\_NO), CONSTRAINT PK\_ADMITS PRIMARY KEY (ROOM\_NO, P\_ID));**

# Normalization: -

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Database**  **Type** | | **Tables** | | |
| Normalized | (1F) | DOCTOR, NURSE, RECEPTION, PATIENT, MEDICINE, ROOM,  LAB\_REPORT,AILMENT, BILL, TREATS,  LOOKS\_AFTER, MANAGES, ADMITS | | |
| Normalized | (2F) | DOCTOR, NURSE, RECEPTION, PATIENT, MEDICINE, ROOM,  LAB\_REPORT,AILMENT, BILL, TREATS,  LOOKS\_AFTER, MANAGES, ADMITS | | |
| Normalized | (3F) | DOCTOR, NURSE,  LOOKS\_AFTER, | RECEPTION, MANAGES, | BILL, TREATS, ADMITS |

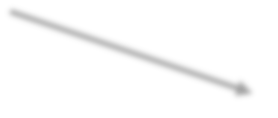
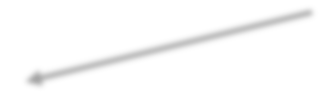
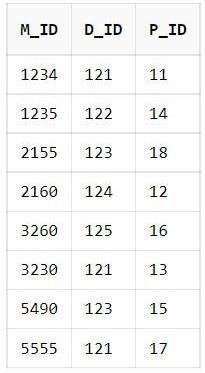
**PATIENT**



**APPOINTMENT PATIENTDETAILS**

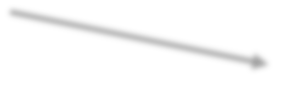
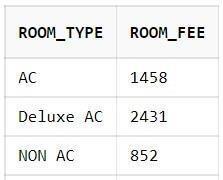
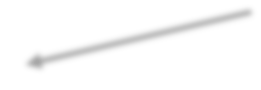
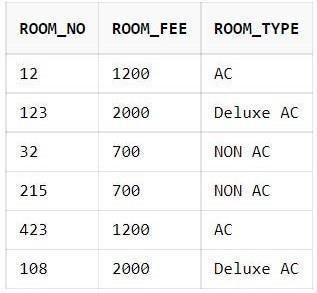
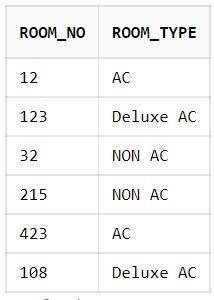
**MEDICINE**



**PRESCRIBES**

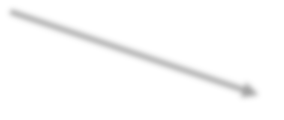
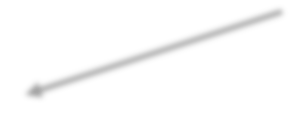
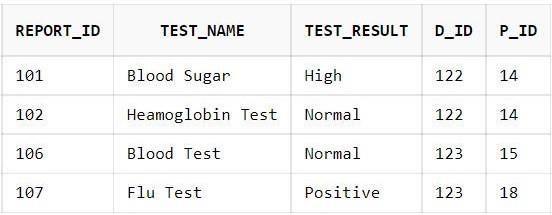
**MEDICINE DETAILS**

**ROOM**

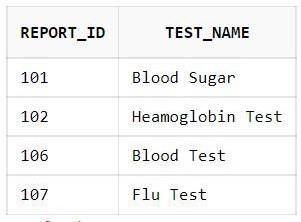


**ROOM DETAIL**

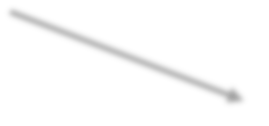
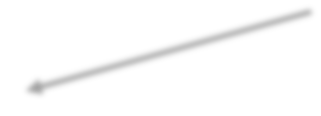
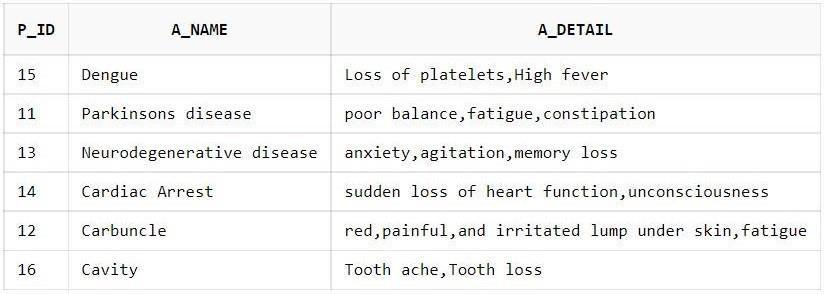
**ROOM CHARGES**

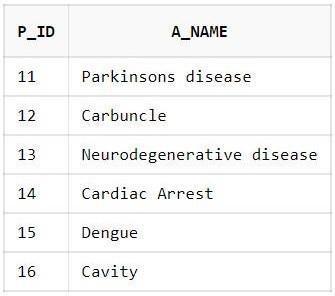
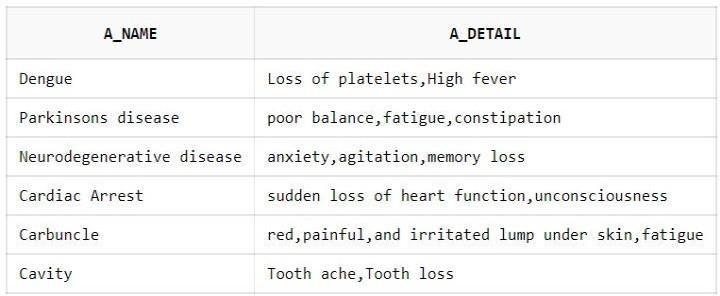
**LAB REPORT**

**LAB TESTS TEST REPORT**

**AILMENT**

**DIAGNOSIS AILMENT DETAILS**

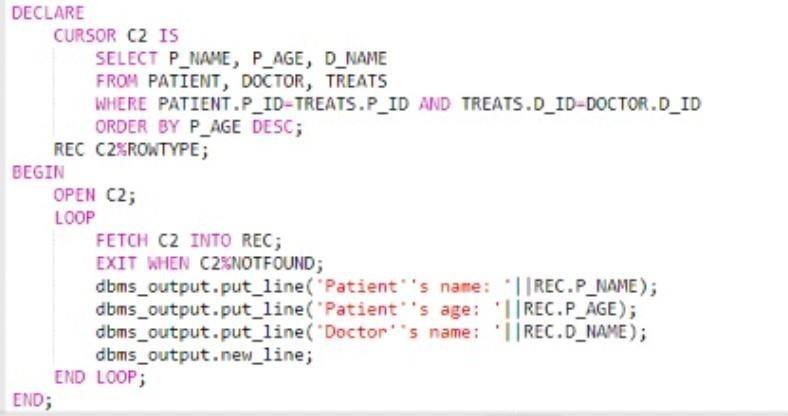
# PL/SQL Queries: -

1. **Write a SQL query to create a trigger for checking Rooms availability.**

****

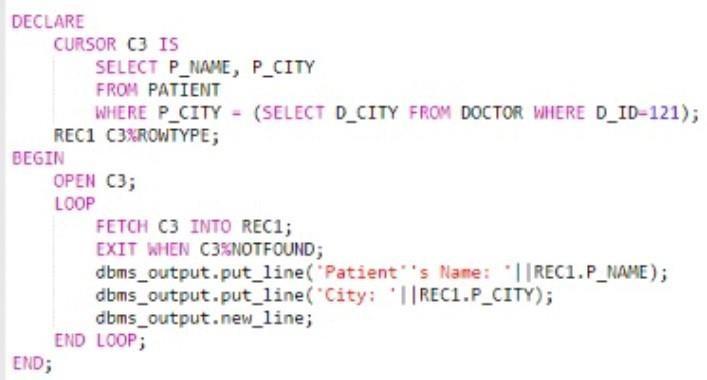
****

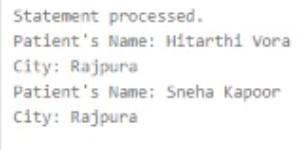
1. **List the name of the patient descending order of their age with their doctor's name.**





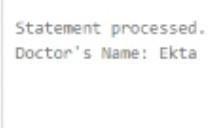
1. **List the name of the patients who have the same city as the doctor id ‘121’**



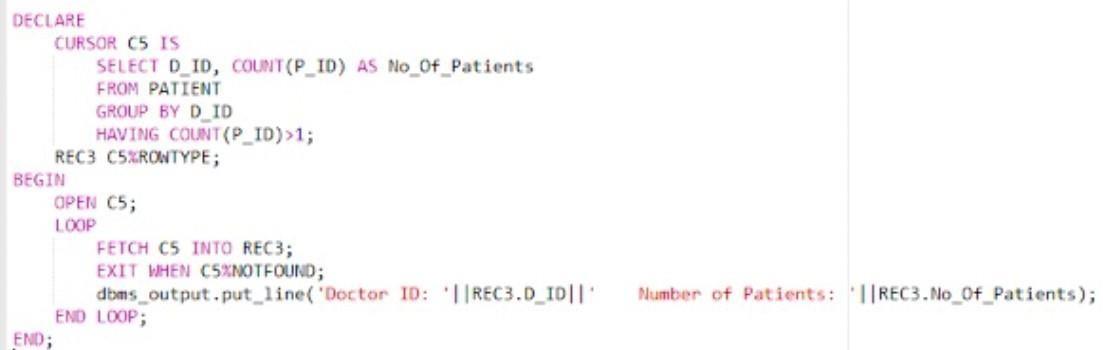


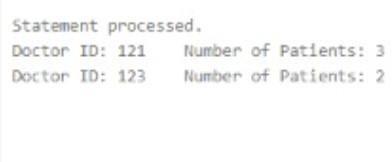
1. **List the name of the doctor who treat the patients with the date of admission between 01-Nov-2021 and 08-Nov-2021**



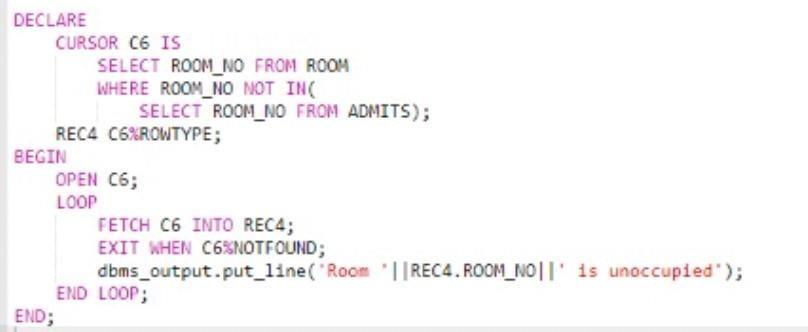


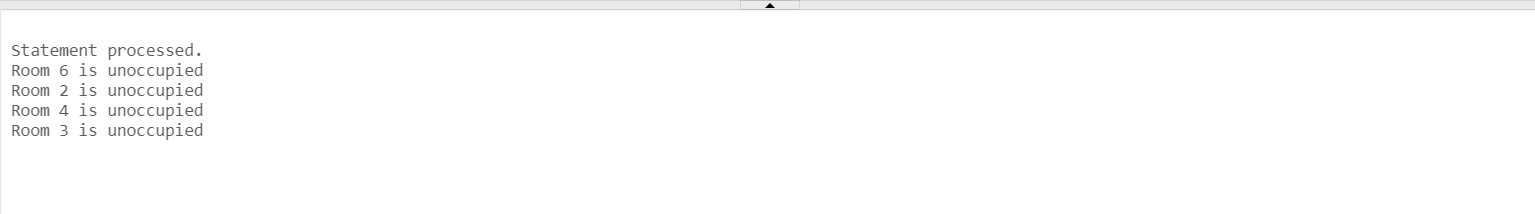
1. **List the name of doctors treating more than 1 patient.**





1. **List the room numbers that are unoccupied.**



****

**8. Conclusion: -**

A hospital management system is a software solution that is designed to help healthcare providers manage their operations efficiently. The system is typically used by hospitals, clinics, and other healthcare facilities to manage patient records, appointments, billing, and other administrative tasks.

In summary, a hospital management system can provide the following benefits:

Improved patient care: With a hospital management system, healthcare providers can access patient records quickly and easily, allowing them to provide better care to their patients.

Increased efficiency: The system can automate many administrative tasks, such as scheduling appointments and managing billing, freeing up staff time and increasing efficiency.

Better financial management: The system can help healthcare providers manage their finances more effectively, with features such as billing and accounting tools.

1. **References: -**

* LMS slides
* W3Schools
* Oracle documentations