

**HOSPITAL DATABASE MANAGEMENT SYSTEM**  
**UCS310 -DBMS Project Report End Semester Lab Evaluation**

**Submitted By :**

**Darshan Patel(102017033)**

**Kartik Kumar Koli(102017030)**

**Kuldeep Raj Gour(102017031)**

**Subgroup:2CS2**

**Submitted To:**

**Mrs.Geeta Kasana**



**CSED Department, Thapar Institute of Engineering and  
Technology**

**JAN-MAY 2022**

# Contents

- 1. Basic Introduction
  - 1.1 Database Management System
  - 1.2 Relational Database Management System
  - 1.3 ER diagram
  - 1.4 Terminologies and Symbols of ER diagram
- 2. Hospital Database Management System
  - 2.1 Problem Statement
  - 2.2 Objectives of the system
  - 2.3 ER diagram: Hospital Database
  - 2.4 ER to Table
  - 2.5 Physical Schema
  - 2.6 Normalisation
  - 2.7 SQL Queries
  - 2.8 PL/SQL

## **1.BASIC INTRODUCTION**

### **1.1 DATABASE MANAGEMENT SYSTEM: -**

A Database-management system (DBMS) is a collection of interrelated data and a set of programs to access that data. This is a collection of related data with an implicit meaning and hence is a database. The primary goal of a DBMS is to provide a way to store and retrieve database information that is both convenient and efficient.







## 1.2 RELATIONAL DATABASE MANAGEMENT SYSTEM: -



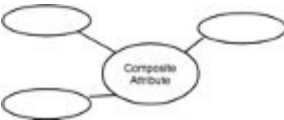


RDBMS is the collection of programs and capabilities that enables the user to interact with a relational database. A relational database management system (RDBMS) is a type of DBMS with a row-based table structure.

## 1.3 ER DIAGRAM: -

ER model stands for an Entity-Relationship model. It is a high-level data model. This model is used to define the data elements and relationship for a specified system. It develops a conceptual design for the database. It also develops a quite simple and easy to design view of data. In ER modelling, the database structure is portrayed as a diagram called an entity-relationship diagram.

## 1.4 TERMINOLOGIES AND SYMBOLS OF ER DIAGRAM: -

SYMBOL	DESCRIPTION
	This is a basic entity that is represented by a rectangle with its name inside.
	This is an entity that cannot solely be identified with its attributes (due to the absence of a primary key). It inherits the identifier of its parent entity and often integrates it with a partial key.
	A strong relationship is depicted by a single rhombus with its name inside. In this, an entity is independent, that is, its primary key for any child doesn't contain the primary key of the linked entity.
	A weak relationship is depicted by a double rhombus with the name inside. In this, the child is dependent on the parent entity as its primary key would contain a component of the parent's primary key.
	A basic attribute is represented by a single oval with its name written inside.
	This is a special attribute that is used to uniquely identify an entity. It is represented by an oval with its name underlined.

	<p>These are the attributes that can have multiple values (like the Name attribute can have First and Last name) and are represented by a double oval.</p>
	<p>A derived attribute might not be physically present in the database and could be logically derived from any other attribute (Represented by a dotted oval).</p>
	<p>Composite attributes are those attributes which are composed of many other simple attributes.</p>
	<p>This depicts that not all the entities in the set are a part of the relationship and is depicted by a single line.</p>
	<p>This means that all the entities in the set are in a relationship and are depicted by a double line.</p>

## 2.HOSPITAL MANAGEMENT SYSTEM

### 2.1 Problem Statement: -

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.

## 2.2 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:

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

## 2.3 ER Diagram: Hospital Database: -

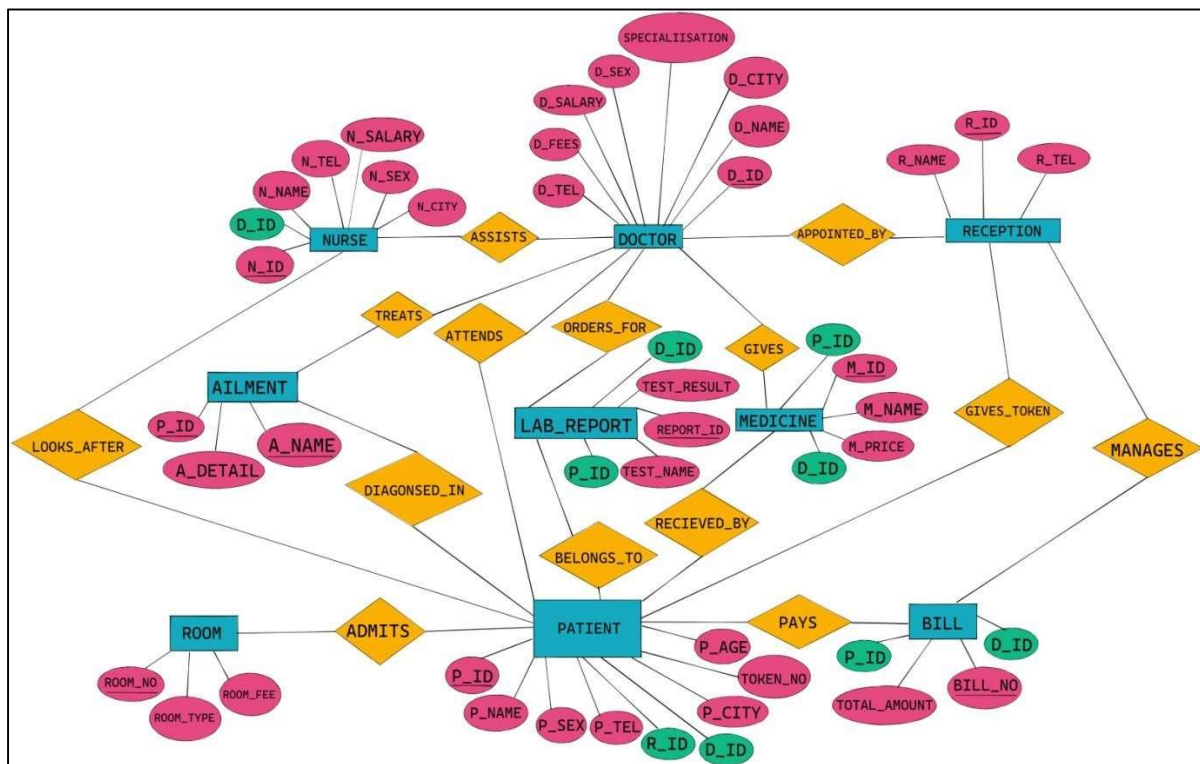


Fig. 2.3: ER Diagram



a) 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.

b) 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.

c) 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.

d) 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.

e) Nurse information includes ID, 'N\_ID', the key attribute used to uniquely identify the nurse including name, contact details and their gender.

- f) Room information for a specific patient includes 'ROOM\_NO' the key attribute used to uniquely identify the room, room type and the room fee.
- g) Ailment (weak entity type) includes information under a particular patient using 'A\_NAME' to identify the ailment and ailment details.
- h) Reception information includes ID 'R\_ID' the key attribute used to uniquely identify the reception, reception-name, and reception contact details.
- i) 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 ratio 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.

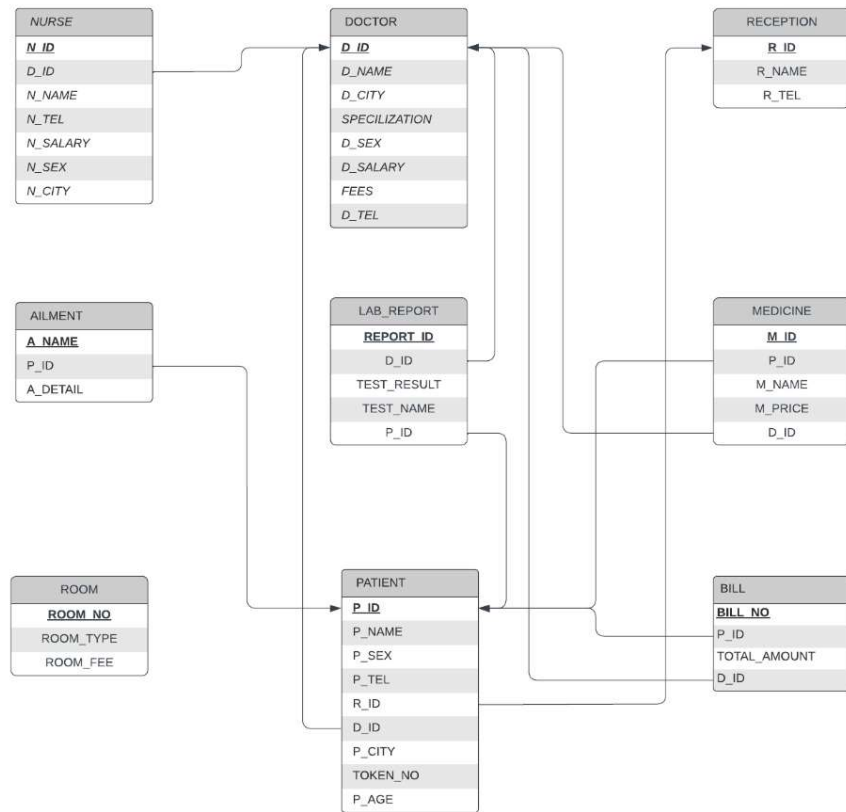
**DIAGNOSED\_IN:** between the weak entity type Ailment and strong entity type Patient. The cardinality ratio is 1: n for Patient to Ailment. Both have total participation.

**Pays:** between the weak entity type Bill and the strong entity type Patient.

The cardinality ratio is 1: n for Patient to Bill.

## 2.4. ER to Table:





## 2.5 Physical Schema:-

**Table Name: NURSE**

<b>FIELDS</b>	<b>DATA TYPE</b>	<b>RELATIONSHIPS</b>
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**

<b>FIELDS</b>	<b>DATA TYPE</b>	<b>RELATIONSHIPS</b>
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

**Table Name: RECEPTION**

<b>FIELDS</b>	<b>DATA TYPE</b>	<b>RELATIONSHIPS</b>
R_ID	NUMBER (5)	PRIMARY KEY
R_NAME	VARCHAR (20)	NOT NULL
R_TEL	NUMBER (10)	NOT NULL

**Table Name: PATIENT**

<b>FIELDS</b>	<b>DATA TYPE</b>	<b>RELATIONSHIPS</b>
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**

<b>FIELDS</b>	<b>DATA TYPE</b>	<b>RELATIONSHIPS</b>
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: AILMENT**

<b>FIELDS</b>	<b>DATA TYPE</b>	<b>RELATIONSHIPS</b>
A_NAME	VARCHAR (50)	PRIMARY KEY
P_ID	NUMBER (5)	PRIMARY KEY
A_DETAIL	VARCHAR (50)	NOT NULL

**Table Name: LAB REPORT**

<b>FIELDS</b>	<b>DATA TYPE</b>	<b>RELATIONSHIPS</b>
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**

<b>FIELDS</b>	<b>DATA TYPE</b>	<b>RELATIONSHIPS</b>
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

2.6 Normalization: -

<b>Database Type</b>	<b>Tables</b>
Normalized (1F)	DOCTOR, NURSE, RECEPTION, PATIENT, MEDICINE, ROOM,

	LAB_REPORT,AILMENT, LOOKS_AFTER,	BILL, TREATS, MANAGES, ADMITS
Normalized (2F)	DOCTOR, NURSE, MEDICINE, ROOM, LAB_REPORT,AILMENT, LOOKS_AFTER,	RECEPTION, PATIENT, BILL, TREATS, MANAGES, ADMITS
Normalized (3F)	DOCTOR, NURSE, LOOKS_AFTER,	RECEPTION, BILL, TREATS, MANAGES, ADMITS

### PATIENT

P_ID	P_NAME	P_AGE	P_SEX	P_TEL	P_CITY	D_ID	R_ID	TOKEN_NO
11	Hitarthi Vora	34	F	9408048634	Rajpura	121	1	141
12	Archana Naik	18	F	9512245764	Sangrur	124	4	142
13	Dev Parekh	25	M	9714410269	Barnala	121	1	143
14	Subhash Patel	60	M	9825747459	Mohali	122	2	144
15	Abhinav Goyal	10	M	8983443606	Delhi	123	3	145
16	Sneha Kapoor	45	F	9823679043	Rajpura	125	5	146
17	Gautam Gada	31	M	9876116902	Jalandhar	121	1	147
18	Disha Bedi	4	F	9417085850	Mohali	123	3	148



### APPOINTMENT

P_ID	D_ID	R_ID	TOKEN_NO
11	121	1	141
12	124	4	142
13	121	1	143
14	122	2	144
15	123	3	145
16	125	5	146
17	121	1	147
18	123	3	148

### PATIENT DETAILS

P_ID	P_NAME	P_AGE	P_SEX	P_TEL	P_CITY
11	Hitarthi Vora	34	F	9408048634	Rajpura
12	Archana Naik	18	F	9512245764	Sangrur
13	Dev Parekh	25	M	9714410269	Barnala
14	Subhash Patel	60	M	9825747459	Mohali
15	Abhinav Goyal	10	M	8983443606	Delhi
16	Sneha Kapoor	45	F	9823679043	Rajpura
17	Gautam Gada	31	M	9876116902	Jalandhar
18	Disha Bedi	4	F	9417085850	Mohali

## MEDICINE

M_ID	M_NAME	M_PRICE	D_ID	P_ID
1234	Phenytoin	450	121	11
1235	Lacosamide	350	122	14
2155	Kerlone	1000	123	18
2160	Zebeta	850	124	12
3260	Gramicidin	400	125	16
3230	Ofloxacin	550	121	13
5490	Albuterol	300	123	15
5555	Amoxicillin	250	121	17

## PRESCRIBES

M_ID	D_ID	P_ID
1234	121	11
1235	122	14
2155	123	18
2160	124	12
3260	125	16
3230	121	13
5490	123	15
5555	121	17

## MEDICINE DETAILS

M_ID	M_NAME	M_PRICE
1234	Phenytoin	450
1235	Lacosamide	350
2155	Kerlone	1000
2160	Zebeta	850
3260	Gramicidin	400
3230	Ofloxacin	550
5490	Albuterol	300
5555	Amoxicillin	250

## ROOM

ROOM_NO	ROOM_FEE	ROOM_TYPE
12	1200	AC
123	2000	Deluxe AC
32	700	NON AC
215	700	NON AC
423	1200	AC
108	2000	Deluxe AC

## ROOM DETAIL

ROOM_NO	ROOM_TYPE
12	AC
123	Deluxe AC
32	NON AC
215	NON AC
423	AC
108	Deluxe AC

## ROOM CHARGES

ROOM_TYPE	ROOM_FEE
AC	1458
Deluxe AC	2431
NON AC	852

### LAB REPORT

REPORT_ID	TEST_NAME	TEST_RESULT	D_ID	P_ID
101	Blood Sugar	High	122	14
102	Heamoglobin Test	Normal	122	14
106	Blood Test	Normal	123	15
107	Flu Test	Positive	123	18



### LAB TESTS

REPORT_ID	TEST_NAME
101	Blood Sugar
102	Heamoglobin Test
106	Blood Test
107	Flu Test

### TEST REPORT

REPORT_ID	D_ID	P_ID	TEST_RESULT
101	122	14	High
102	122	14	Normal
106	123	15	Normal
107	123	18	Positive

### AILMENT

P_ID	A_NAME	A_DETAIL
15	Dengue	Loss of platelets,High fever
11	Parkinsons disease	poor balance,fatigue,constipation
13	Neurodegenerative disease	anxiety,agitation,memory loss
14	Cardiac Arrest	sudden loss of heart function,unconsciousness
12	Carbuncle	red,painful,and irritated lump under skin,fatigue
16	Cavity	Tooth ache,Tooth loss



### DIAGNOSIS

P_ID	A_NAME
11	Parkinsons disease
12	Carbuncle
13	Neurodegenerative disease
14	Cardiac Arrest
15	Dengue
16	Cavity

### AILMENT DETAILS

A_NAME	A_DETAIL
Dengue	Loss of platelets,High fever
Parkinsons disease	poor balance,fatigue,constipation
Neurodegenerative disease	anxiety,agitation,memory loss
Cardiac Arrest	sudden loss of heart function,unconsciousness
Carbuncle	red,painful,and irritated lump under skin,fatigue
Cavity	Tooth ache,Tooth loss

## 2.7 SQL Code:

### Table DOCTOR:

```
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));
```

TABLE DOCTOR

Column	Null?	Type
D_ID	NOT NULL	NUMBER(5,0)
D_NAME	-	VARCHAR2(20)
D_SEX	-	VARCHAR2(2)
D_TEL	-	NUMBER(10,0)
D_CITY	-	VARCHAR2(10)
SPECIALISATION	-	VARCHAR2(50)
D_FEES	-	NUMBER(10,0)
D_SALARY	-	NUMBER(10,0)

[Download CSV](#)

8 rows selected.

### Inserting Values in DOCTOR:

```
insert into doctor values(121,'Ayushi','F' ,9637912702,'Rajpura','Neurology',300,150000);  
insert into doctor values(122,'Chhavi','F' ,8840638864,'Chandigarh','Cardiology',400,140000);  
insert into doctor values(123,'Ekta','F' ,9876256487,'Panchkula','Pediatrician',200,154300);  
insert into doctor values(124,'Arnav','M' ,8853821122,'Zirakpur','Dermatology',100,152000);  
insert into doctor values(125,'Jay','M' ,7424905270,'Mohali','Dentist',350,110000);  
select * from doctor
```



D_ID	D_NAME	D_SEX	D_TEL	D_CITY	SPECIALISATION	D_FEES	D_SALARY
121	Ayushi	F	9637912702	Rajpura	Neurology	300	150000
122	Chhavi	F	8840638864	Chandigarh	Cardiology	400	140000
123	Ekta	F	9876256487	Panchkula	Pediatrician	200	154300
124	Arnav	M	8853821122	Zirakpur	Dermatology	100	152000
125	Jay	M	7424905270	Mohali	Dentist	350	110000

[Download CSV](#)

5 rows selected.

### Table NURSE:

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);

TABLE NURSE

Column	Null?	Type
N_ID	NOT NULL	NUMBER(5,0)
N_NAME	-	VARCHAR2(20)
N_SEX	-	VARCHAR2(2)
N_TEL	-	NUMBER(10,0)
N_CITY	-	VARCHAR2(10)
N_SALARY	-	NUMBER(10,0)
D_ID	-	NUMBER(5,0)

[Download CSV](#)

7 rows selected.

### Inserting values into NURSE:

insert into nurse values(1,'Mitali Vora','F',9879811559,'Patiala',10000,121);

```

insert into nurse values(2,'Sakshi Desai','F',9428120909,'Ludhiana',15000,122);
insert into nurse values(3,'Hirai Parekh','F',9726276215,'Dera Basi',12000,123);
insert into nurse values(4,'Ketki Parekh','F',9924841494,'Kharar',17000,124);
insert into nurse values(5,'Sita Kumari','F',8085562345,'Bathinda',8000,125);
select * from nurse

```

N_ID	N_NAME	N_SEX	N_TEL	N_CITY	N_SALARY	D_ID
1	Mitali Vora	F	9879811559	Patiala	10000	121
2	Sakshi Desai	F	9428120909	Ludhiana	15000	122
3	Hirai Parekh	F	9726276215	Dera Basi	12000	123
4	Ketki Parekh	F	9924841494	Kharar	17000	124
5	Sita Kumari	F	8085562345	Bathinda	8000	125

[Download CSV](#)

5 rows selected.

#### Table RECEPTION:

```

CREATE TABLE RECEPTION (
R_ID NUMBER (5),
R_NAME VARCHAR (20),
R_TEL NUMBER (10),
CONSTRAINT PK_RECEPTION PRIMARY KEY(R_ID));

```

TABLE RECEPTION

Column	Null?	Type
R_ID	NOT NULL	NUMBER(5,0)
R_NAME	-	VARCHAR2(20)
R_TEL	-	NUMBER(10,0)

[Download CSV](#)

3 rows selected.

#### Inserting values into RECEPTION:

```

insert into reception values(1,'Neurology Dept.',2782205311);
insert into reception values(2,'Cardiology Dept.',2782514198);
insert into reception values(3,'Pediatrician Dept.',2783482689);
insert into reception values(4,'Dermatology Dept.',2788665432);

```

```
insert into reception values(5,'Dentist Dept.',2786732563);
```

```
select * from reception
```

R_ID	R_NAME	R_TEL
1	Neurology Dept.	2782205311
2	Cardiology Dept.	2782514198
3	Pediatrician Dept.	2783482689
4	Dermatology Dept.	2788665432
5	Dentist Dept.	2786732563

[Download CSV](#)

5 rows selected.

#### **Table PATIENT:**

```
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));
```

TABLE PATIENT

Column	Null?	Type
P_ID	NOT NULL	NUMBER(5,0)
P_NAME	-	VARCHAR2(20)
P_AGE	-	NUMBER(10,0)
P_SEX	-	VARCHAR2(2)
P_TEL	-	NUMBER(10,0)
P_CITY	-	VARCHAR2(10)
D_ID	-	NUMBER(5,0)
R_ID	-	NUMBER(5,0)
TOKEN_NO	-	NUMBER(5,0)

[Download CSV](#)

9 rows selected.

**Inserting values into PATIENT:**

```

insert into patient values(11,'Hitarthi Vora',34,'F',9408048634,'Rajpura',121,1,141);
insert into patient values(12,'Archana Naik',18,'F',9512245764,'Sangrur',124,4,142);
insert into patient values(13,'Dev Parekh',25,'M',9714410269,'Barnala',121,1,143);
insert into patient values(14,'Subhash Patel',60,'M',9825747459,'Mohali',122,2,144);
insert into patient values(15,'Abhinav Goyal',10,'M',8983443606,'Delhi',123,3,145);
insert into patient values(16,'Sneha Kapoor',45,'F',9823679043,'Rajpura',125,5,146);
insert into patient values(17,'Gautam Gada',31,'M',9876116902,'Jalandhar',121,1,147);
insert into patient values(18,'Disha Bedi',4,'F',9417085850,'Mohali',123,3,148);
select * from patient

```

P_ID	P_NAME	P_AGE	P_SEX	P_TEL	P_CITY	D_ID	R_ID	TOKEN_NO
11	Hitarthi Vora	34	F	9408048634	Rajpura	121	1	141
12	Archana Naik	18	F	9512245764	Sangrur	124	4	142
13	Dev Parekh	25	M	9714410269	Barnala	121	1	143
14	Subhash Patel	60	M	9825747459	Mohali	122	2	144
15	Abhinav Goyal	10	M	8983443606	Delhi	123	3	145
16	Sneha Kapoor	45	F	9823679043	Rajpura	125	5	146
17	Gautam Gada	31	M	9876116902	Jalandhar	121	1	147
18	Disha Bedi	4	F	9417085850	Mohali	123	3	148

[Download CSV](#)

8 rows selected.

**Table MEDICINE:**

```
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);
```

TABLE MEDICINE

Column	Null?	Type
M_ID	NOT NULL	NUMBER(5,0)
M_NAME	-	VARCHAR2(20)
M_PRICE	-	NUMBER(5,0)
D_ID	-	NUMBER(5,0)
P_ID	-	NUMBER(5,0)

[Download CSV](#)

5 rows selected.

**Inserting values into MEDICINE:**

```
insert into MEDICINE values(1234, 'Phenytoin', 450, 121, 11);  
insert into MEDICINE values(1235, 'Lacosamide', 350, 122, 14);  
insert into MEDICINE values(2155, 'Kerolone', 1000, 123, 18);  
insert into MEDICINE values(2160, 'Zebeta', 850, 124, 12);  
insert into MEDICINE values(3260, 'Gramicidin', 400, 125, 16);  
insert into MEDICINE values(3230, 'Ofloxacin', 550, 121, 13);  
insert into MEDICINE values(5490, 'Albuterol', 300, 123, 15);  
insert into MEDICINE values(5555, 'Amoxicilin', 250, 121, 17);  
select * from MEDICINE
```

M_ID	M_NAME	M_PRICE	D_ID	P_ID
1234	Phenytoin	450	121	11
1235	Lacosamide	350	122	14
2155	Kerolone	1000	123	18
2160	Zebeta	850	124	12
3260	Gramicidin	400	125	16
3230	Ofloxacin	550	121	13
5490	Albuterol	300	123	15
5555	Amoxicilin	250	121	17

[Download CSV](#)

8 rows selected.

### Table ROOM:

```
CREATE TABLE ROOM (
ROOM_NO NUMBER (5),
ROOM_FEE NUMBER (5),
ROOM_TYPE VARCHAR (20),
CONSTRAINT PK_ROOM PRIMARY KEY(ROOM_NO));
```

TABLE ROOM

Column	Null?	Type
ROOM_NO	NOT NULL	NUMBER(5,0)
ROOM_FEE	-	NUMBER(5,0)
ROOM_TYPE	-	VARCHAR2(20)

[Download CSV](#)

3 rows selected.

### Inserting values into ROOM:

```
insert into ROOM values(12, 1200, 'AC');
insert into ROOM values(123, 2000, 'Deluxe AC');
insert into ROOM values(32, 700, 'NON AC');
insert into ROOM values(215, 700, 'NON AC');
insert into ROOM values(423, 1200, 'AC');
insert into ROOM values(108, 2000, 'Deluxe AC');
select * from ROOM
```

ROOM_NO	ROOM_FEE	ROOM_TYPE
12	1200	AC
123	2000	Deluxe AC
32	700	NON AC
215	700	NON AC
423	1200	AC
108	2000	Deluxe AC

[Download CSV](#)

6 rows selected.

### TABLE LAB\_REPORT:

```
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);
```

TABLE LAB\_REPORT

Column	Null?	Type
REPORT_ID	NOT NULL	NUMBER(5,0)
TEST_NAME	-	VARCHAR2(20)
TEST_RESULT	-	VARCHAR2(10)
D_ID	-	NUMBER(5,0)
P_ID	-	NUMBER(5,0)

[Download CSV](#)

5 rows selected.

### Inserting values into LAB\_REPORT:

```
insert into LAB_REPORT values(101, 'Blood Sugar', 'High', 122, 14);
```

```
insert into LAB_REPORT values(102, 'Haemoglobin Test', 'Normal', 122, 14);
```

```
insert into LAB_REPORT values(106, 'Blood Test', 'Normal', 123, 15);
```

```
insert into LAB_REPORT values(107, 'Flu Test', 'Positive', 123, 18);
```

```
select * from LAB_REPORT
```

REPORT_ID	TEST_NAME	TEST_RESULT	D_ID	P_ID
101	Blood Sugar	High	122	14
102	Haemoglobin Test	Normal	122	14
106	Blood Test	Normal	123	15
107	Flu Test	Positive	123	18

[Download CSV](#)

4 rows selected.

#### Table AILMENT:

```
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);
```

TABLE AILMENT

Column	Null?	Type
P_ID	NOT NULL	NUMBER(5,0)
A_NAME	NOT NULL	VARCHAR2(50)
A_DETAIL	-	VARCHAR2(50)

[Download CSV](#)

3 rows selected.

#### Inserting values into AILMENT:

```
insert into AILMENT values(15, 'Dengue', 'Loss of platelets,High fever');
```

```
insert into AILMENT values(11, 'Parkinsons', 'Poor balance,fatigue,constipation');
```

```
insert into AILMENT values(13, 'Neurodegenerative disease', 'Anxiety,agitation,memory loss');
```

```
insert into AILMENT values(14, 'Cardiac Arrest', 'Sudden loss of heart function,unconsciousness');
```

```
insert into AILMENT values(12, 'Carbuncle', 'red,painful,and irritated lump under skin,fatigue');
```

```
insert into AILMENT values(16, 'Cavity', 'Tooth ache,Tooth loss');
```

```
select * from AILMENT
```



P_ID	A_NAME	A_DETAIL
15	Dengue	Loss of platelets,High fever
11	Parkinsons	Poor balance,fatigue,constipation
13	Neurodegenerative disease	Anxiety,agitation,memory loss
14	Cardiac Arrest	Sudden loss of heart function,unconsciousness
12	Carbuncle	red,painful,and irritated lump under skin,fatigue
16	Cavity	Tooth ache,Tooth loss

[Download CSV](#)

6 rows selected.

### Table BILL:

```
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));
```

TABLE BILL

Column	Null?	Type
BILL_NO	NOT NULL	NUMBER(10,0)
P_ID	NOT NULL	NUMBER(5,0)
D_ID	NOT NULL	NUMBER(5,0)
TOTAL_AMOUNT	-	NUMBER(10,0)

[Download CSV](#)

4 rows selected.

### Inserting values into BILL:

```
insert into BILL values(50, 11, 121, 700);
insert into BILL values(52, 12, 124, 1500);
insert into BILL values(75, 13, 121, 5000);
insert into BILL values(102, 14, 122, 25000);
insert into BILL values(165, 15, 123, 50000);
insert into BILL values(245, 16, 125, 4500);
insert into BILL values(250, 17, 121, 1800);
```

insert into BILL values(243, 18, 123, 500);

select \* from BILL

BILL_NO	P_ID	D_ID	TOTAL_AMOUNT
50	11	121	700
52	12	124	1500
75	13	121	5000
102	14	122	25000
165	15	123	50000
245	16	125	4500
250	17	121	1800
243	18	123	500

[Download CSV](#)

8 rows selected.

#### Table TREATS:

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));

TABLE TREATS

Column	Null?	Type
D_ID	NOT NULL	NUMBER(5,0)
P_ID	NOT NULL	NUMBER(5,0)

[Download CSV](#)

2 rows selected.

#### Inserting values into TREATS:

insert into TREATS values(121, 11);

insert into TREATS values(121, 13);

insert into TREATS values(121, 17);

insert into TREATS values(122, 14);

```

insert into TREATS values(123, 15);
insert into TREATS values(123, 18);
insert into TREATS values(124, 12);
insert into TREATS values(125, 16);
select * from TREATS

```

D_ID	P_ID
121	11
121	13
121	17
122	14
123	15
123	18
124	12
125	16

[Download CSV](#)

8 rows selected.

#### Table LOOKS\_AFTER:

```

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));

```

TABLE LOOKS\_AFTER

Column	Null?	Type
N_ID	NOT NULL	NUMBER(5,0)
P_ID	NOT NULL	NUMBER(5,0)

[Download CSV](#)

2 rows selected.

#### Inserting values into LOOKS\_AFTER:

```

insert into LOOKS_AFTER values(2, 14);
insert into LOOKS_AFTER values(3, 15);

```

select \* from LOOKS\_AFTER

N_ID	P_ID
2	14
3	15

[Download CSV](#)

2 rows selected.

### Table MANAGES:

```
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));
```

TABLE MANAGES

Column	Null?	Type
R_ID	NOT NULL	NUMBER(5,0)
P_ID	NOT NULL	NUMBER(5,0)
D_ID	NOT NULL	NUMBER(5,0)
BILL_NO	NOT NULL	NUMBER(10,0)

[Download CSV](#)

4 rows selected.

### Inserting values into MANAGES:

```
insert into MANAGES values(1, 11, 121, 50);  
insert into MANAGES values(4, 12, 124, 52);  
insert into MANAGES values(1, 13, 121, 75);  
insert into MANAGES values(2, 14, 122, 102);  
insert into MANAGES values(3, 15, 123, 165);  
insert into MANAGES values(3, 18, 123, 243);
```

```
insert into MANAGES values(5, 16, 125, 245);
```

```
insert into MANAGES values(1, 17, 121, 250);
```

```
select * from MANAGES
```

R_ID	P_ID	D_ID	BILL_NO
1	11	121	50
4	12	124	52
1	13	121	75
2	14	122	102
3	15	123	165
3	18	123	243
5	16	125	245
1	17	121	250

[Download CSV](#)

8 rows selected.

#### Table ADMITS:

```
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));
```

TABLE ADMITS

Column	Null?	Type
ROOM_NO	NOT NULL	NUMBER(5,0)
P_ID	NOT NULL	NUMBER(5,0)
DATE_OF_ADMISSION	-	DATE

[Download CSV](#)

3 rows selected.

#### Inserting values into ADMITS:

```
insert into ADMITS values(12, 15, '07-NOV-21');
```

```
insert into ADMITS values(32,17,'18-OCT-21');
```

```
select * from ADMITS
```

ROOM_NO	P_ID	DATE_OF_ADMISSION
12	15	07-NOV-21
32	17	18-OCT-21

[Download CSV](#)

2 rows selected.

## 2.8 PL/SQL Queries:

**1. Write a SQL query to count the number of AC rooms available.**

**Ans.**

```
declare
    cn number;
    message varchar2(30);
begin
    select count(*) into cn from ROOM where room_type='AC';
    if cn>0 then
        dbms_output.put_line('No. of AC rooms available=' || cn);
    else
        message:='No AC rooms available';
        dbms_output.put_line(message);
    end if;
end;
```

**Output:**

```
Statement processed.
No. of AC rooms available=2
```

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

**Ans.**

```
declare cursor c1
is
```

```
select P_NAME,P_AGE,D_NAME from PATIENT,DOCTOR,TREATS where
PATIENT.P_ID=TREATS.P_ID and TREATS.D_ID=DOCTOR.D_ID order by P_AGE desc;
```

```
var1 c1%rowtype;
```

```
begin
```

```
for var1 in c1 loop
```

```
    dbms_output.put_line('Patient's Name:' || var1.P_NAME);
```

```
    dbms_output.put_line('Patient's Age:' || var1.P_AGE);
```

```
    dbms_output.put_line('Doctor's Name:' || var1.D_NAME);
```

```
end loop;
```

```
end;
```

### Output:

```
Statement processed.
Patient's Name:Subhash Patel
Patient's Age:60
Doctor's Name:Chhavi
Patient's Name:Sneha Kapoor
Patient's Age:45
Doctor's Name:Jay
Patient's Name:Hitarthi Vora
Patient's Age:34
Doctor's Name:Ayushi
Patient's Name:Gautam Gada
Patient's Age:31
Doctor's Name:Ayushi
Patient's Name:Dev Parekh
Patient's Age:25
Doctor's Name:Ayushi
Patient's Name:Archana Naik
Patient's Age:18
Doctor's Name:Arnav
Patient's Name:Abhinav Goyal
Patient's Age:10
Doctor's Name:Ekta
Patient's Name:Disha Bedi
Patient's Age:4
Doctor's Name:Ekta
```

### 3.List the room numbers that are available.

**Ans.**

```
declare cursor c2
```

```
is
```

```
select ROOM_NO from ROOM where ROOM_NO not in(select ROOM_NO from ADMITS);
```

```
var2 c2%rowtype;
```

```
begin
```

```
for var2 in c2 loop
```

```
    dbms_output.put_line('Room no.' || var2.ROOM_NO || ' is available');
```

```
end loop;  
end;
```

**Output:**

```
Statement processed.  
Room no.108 is available  
Room no.215 is available  
Room no.423 is available  
Room no.123 is available
```

**4. List the name of patients who have taken token from Cardiology Dept.**

**Ans.**

```
declare cursor c3  
is  
    select P_NAME from PATIENT where R_ID=(select R_ID from RECEPTION where  
R_NAME='Cardiology Dept.');
```

```
    var3 c3%rowtype;  
begin  
    for var3 in c3 loop  
        dbms_output.put_line('Name of Patient: ' || var3.P_NAME);  
    end loop;  
end;
```

**Output:**

```
Statement processed.  
Name of Patient: Subhash Patel
```

**5.Trigger to show that one room can be occupied by only one patient.**

**Ans.**

```
create or replace trigger checkRoom  
before insert on admits for each row  
declare
```



```

    rowcount int;
begin
    select count(*) into rowcount from admits where room_no = :new.room_no;
    if rowcount <>0 then
        raise_application_error(-20001, 'Room already occupied');
    end if;
end;

```

```

insert into admits values(12,18,'12-NOV-2021')

```

#### **Ouput:**

```

ORA-20001: Room already occupied ORA-06512: at "SQL_UHLISSFADWCMQNZYQCZJGNRTX.CHECKROOM", line 6
ORA-06512: at "SYS.DBMS_SQL", line 1721

```

#### **5. Create a procedure to display no. of patients being treated by different doctors.**

**Ans.**

```

create or replace procedure myprocedure
is
    cursor c4
is
    select D_ID,count(P_ID) as patients from treats group by D_ID;
    var4 c4%rowtype;
begin
    for var4 in c4 loop
        dbms_output.put_line('Doctor ID: '||var4.D_ID||'      No. of patients being
treated:'||var4.patients);
    end loop;
end;

begin
myprocedure;
end;

```

**Output:**

```
Statement processed.  
Doctor ID: 121      No. of patients being treated:3  
Doctor ID: 122      No. of patients being treated:1  
Doctor ID: 123      No. of patients being treated:2  
Doctor ID: 124      No. of patients being treated:1  
Doctor ID: 125      No. of patients being treated:1
```

**6. Show exception handling when we are trying to retrieve information of non existing doctors.**

**Ans.**

```
DECLARE  
  
    doc doctor.d_id%type;  
    dname doctor.d_name%type;  
  
BEGIN  
  
    doc:=126;  
  
    select d_name into dname from doctor where d_id = doc;  
  
    dbms_output.put_line('Name ' || dname);  
  
    EXCEPTION when no_data_found THEN  
  
    dbms_output.put_line('No such doctors');  
  
    when others then  
  
    dbms_output.put_line('Error');  
  
end;
```

**Output:**

```
Statement processed.  
No such doctors
```