

UCS1412 – Database Lab

Mini Project

Pharmacy management system

Done by

Abhishek K (185001005)

Gokhulnath T (185001051)

Hemanth Kumar K M (185001059)

1) Abstract

Pharmacy management system is a management system that is designed to improve accuracy and to enhance safety and efficiency in the pharmaceutical store. This program can be used in any pharmaceutical shops having a database to maintain. It is a computer-based system which helps the pharmacist to improve inventory management, cost, medical safety etc. The system allows the user to enter a manufacturing and expiry date for a particular product or drug during opening stock and sales transaction. The software can print invoices, bills, receipts etc. It can also maintain the record of supplies sent in by the supplier. The system services and goals are established by consultation with system user. It also involves manual entry upon arrival of new batches of drugs and upon drug movement out of the pharmacy for a certain period. Pharmacy management system deals with the maintenance of drugs and consumables in the pharmacy unit. The main aim of the project is to manage the database of a pharmaceutical shop. This project provides insight into the design and implementation of a Pharmacy Management System. It is done by creating a database of the available medicines and equipment in the shop. The primary aim of pharmacy management system is to improve accuracy and enhance safety and efficiency in the pharmaceutical store and to develop software for its effective management. Pharmacy management system is useful to maintain correct database by providing an option to update the drugs in stock and is used to manage most pharmacy related activities.

2) ENTITY-RELATIONSHIP DIAGRAM

ER Diagram for this project consists of 9 regular entities and 3 relational/associative entities. The regular entities are:

- 1: PHARMACY
- 2: CUSTOMER
- 3: DOCTOR
- 4: MANUFACTURER
- 5: MEDIQUIPMENT
- 6: EMPLOYEE
- 7: HOSPITAL
- 8: BILL
- 9: SUPPLIER

The associative entities are:

- 1: PRESCRIBE
- 2: CONTRACT
- 3: WORKS

The attributes of each of these entities are:

1: PHARMACY {PHID, NAME, CITY, FAX, PHONE} where PHID is primary key.

Pharmacy Management System

Dept of ISE, BNMIT 2017-18 Page | 4

2: CUSTOMER {PID, NAME, SEX, CITY, PHONE, AGE, DID} where PID is primary key and DID is foreign key.

3: DOCTOR {DID, DNAME, SPECIALITY, AGE, MOBILE, GENDER} where DID is primary key.

4: MANUFACTURER {CID, NAME, EMAIL, MOBILE, CITY, PHARID} where CID is primary key and PHARID is foreign key.

5: MEDIQUIPMENT {CODE, TRADE_NAME, PRODUCT_TYPE, MFG_DATE, EXP_DATE, PRICE, CID} where CODE is primary key and CID is foreign key.

6: EMPLOYEE {NAME, CITY, DOJ, MOBILE, SALARY, AGE, SEX, PHARID} where MOBILE is primary key and PHARID is foreign key.

7: HOSPITAL {HID, NAME, EMAIL, PHONE, CITY, PHARID} where HID is primary key and PHARID is foreign key.

8: BILL {BID, DOB, AGE, PNAME, MOBILE, CITY, PRODUCT, AMOUNT, PHARID} where BID is primary key.

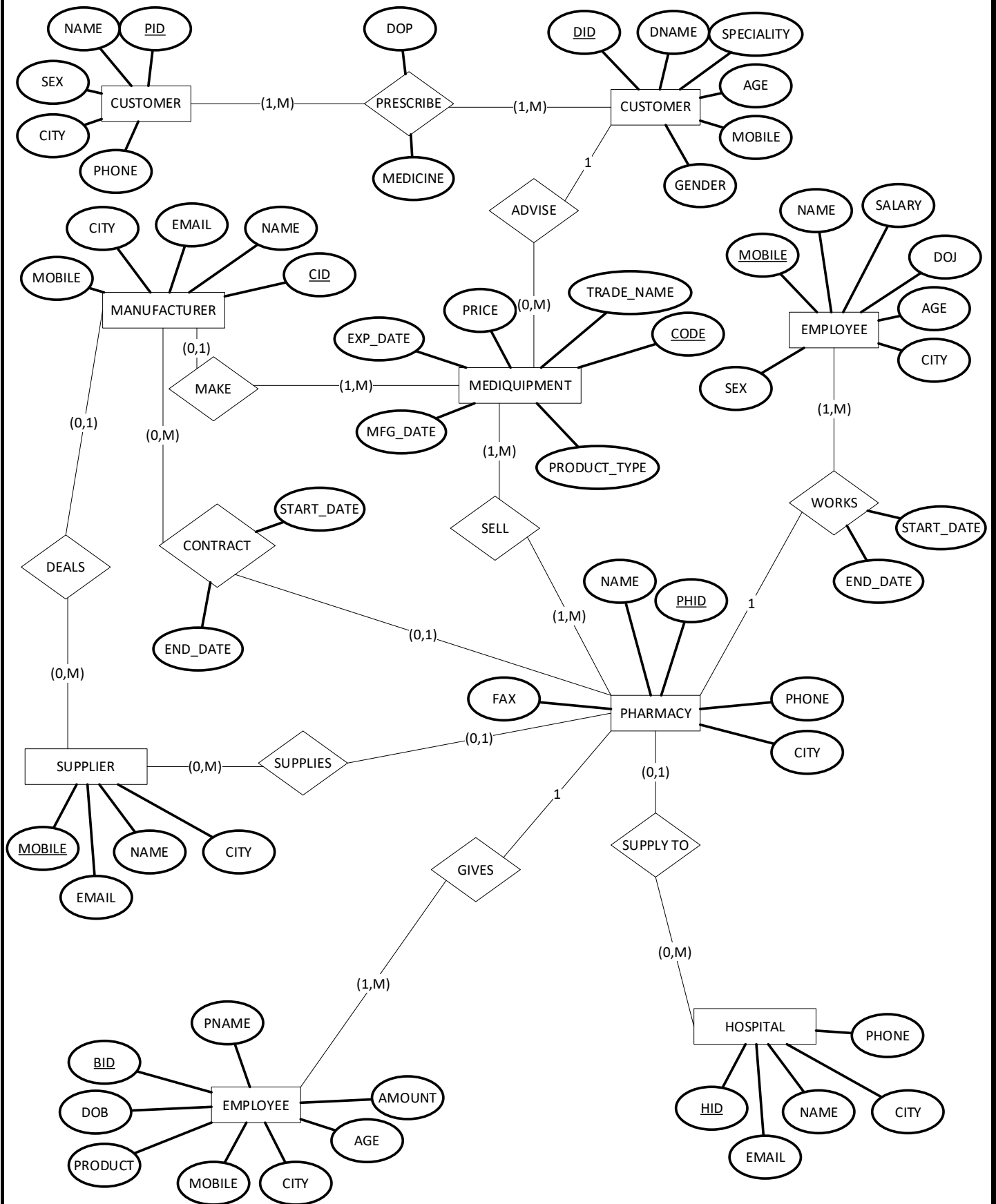
9: SUPPLIER {NAME, CITY, MOBILE, EMAIL, CID, PHARID} where MOBILE is primary key and CID, PHARID is foreign key.

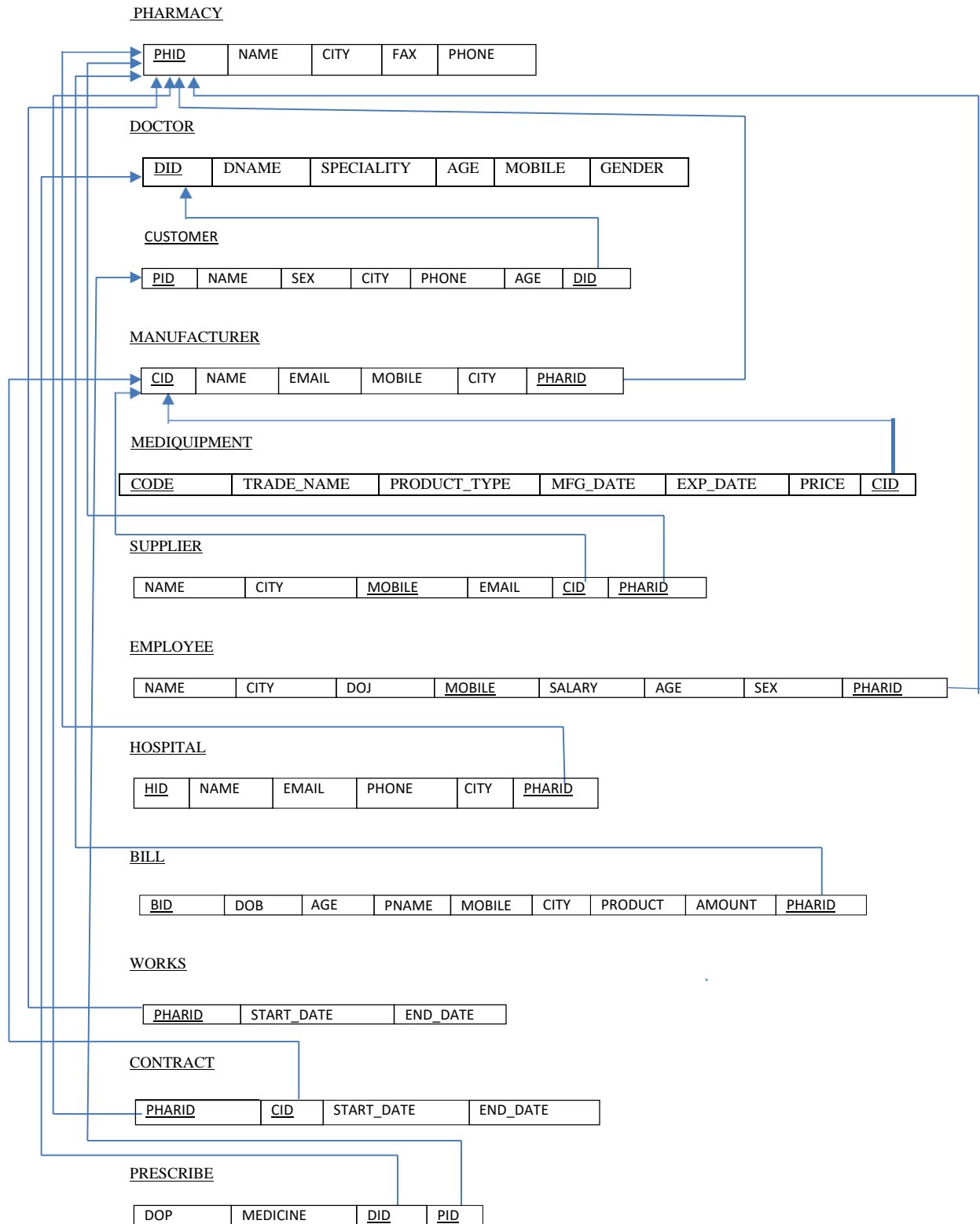
10: PRESCRIBE {DATE, MEDICINE, DID, PID} where DID, PID is foreign key.

11: CONTRACT {PHARID, CID, START_DATE, END_DATE} where PHARID, CID is foreign key.

12: WORKS {PHARID, START_DATE, END_DATE} where PHARID is foreign key.

ER Diagram of Pharmacy Management System





Relational Database Schema for Pharmacy Management System

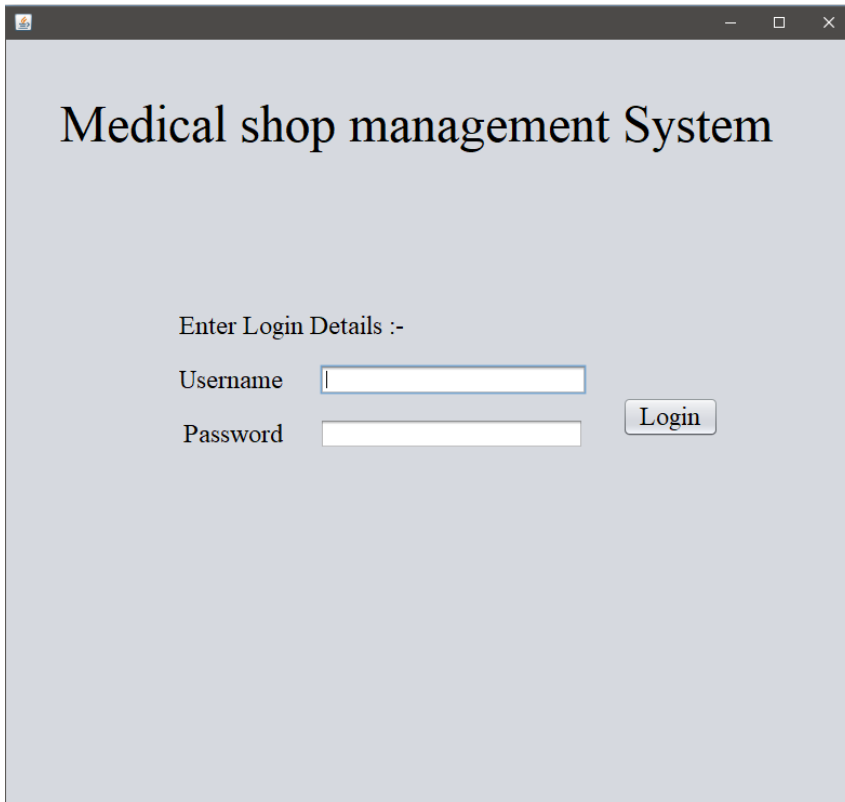
KEY ATTRIBUTES

<u>ENTITY</u>	<u>PRIMARY KEY</u>	<u>FOREIGN KEY</u>
PHARMACY	PHID	
DOCTOR	DID	
CUSTOMER	PID	DID
MANUFACTURER	CID	PHARID
MEDIQUIPMENT	CODE	CID
SUPPLIER	MOBILE	CID, PHARID
EMPLOYEE	MOBILE	PHARID
HOSPITAL	HID	PHARID
BILL	BID	PHARID
WORKS	PHARID	PHARID
CONTRACT	PHARID, CID	PHARID, CID
PRESCRIBE	PID, DID	DID,PID

SNAPSHOTS

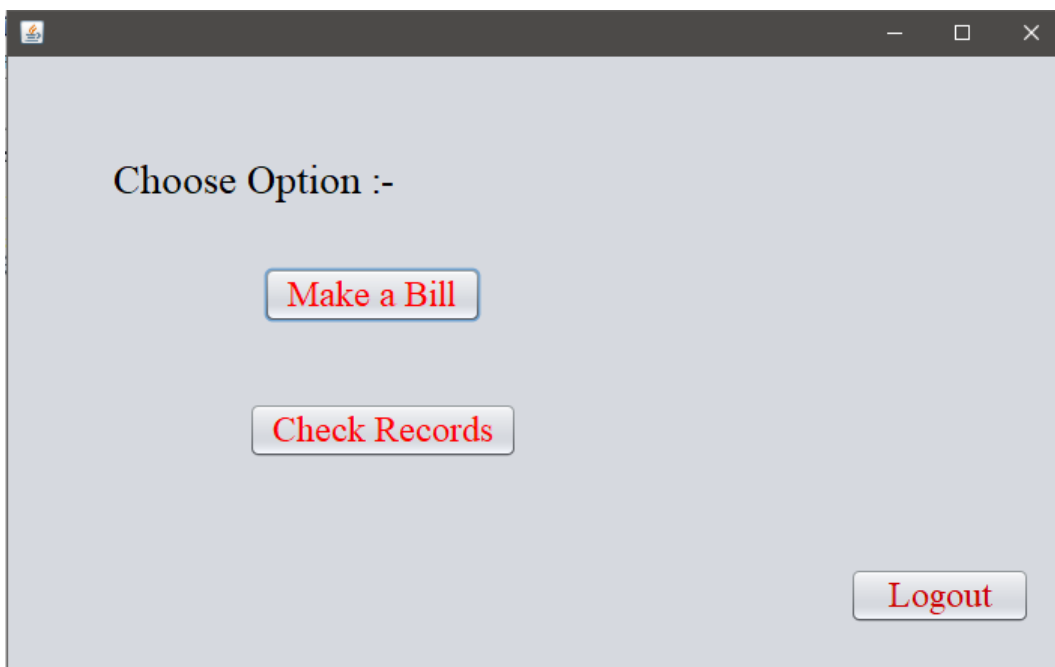
1)Login Page :-

This page asks the user to enter the username and password to login into the system.

A screenshot of a web browser window displaying the login page for the 'Medical shop management System'. The page has a light gray background. At the top, the title 'Medical shop management System' is centered in a large, black, serif font. Below the title, the text 'Enter Login Details :-' is centered. Underneath, there are two input fields: 'Username' and 'Password', each followed by a white text box with a blue border. To the right of the 'Password' field is a 'Login' button with a gray gradient and a black border. The browser window has a standard title bar with a small icon on the left and minus, maximize, and close buttons on the right.

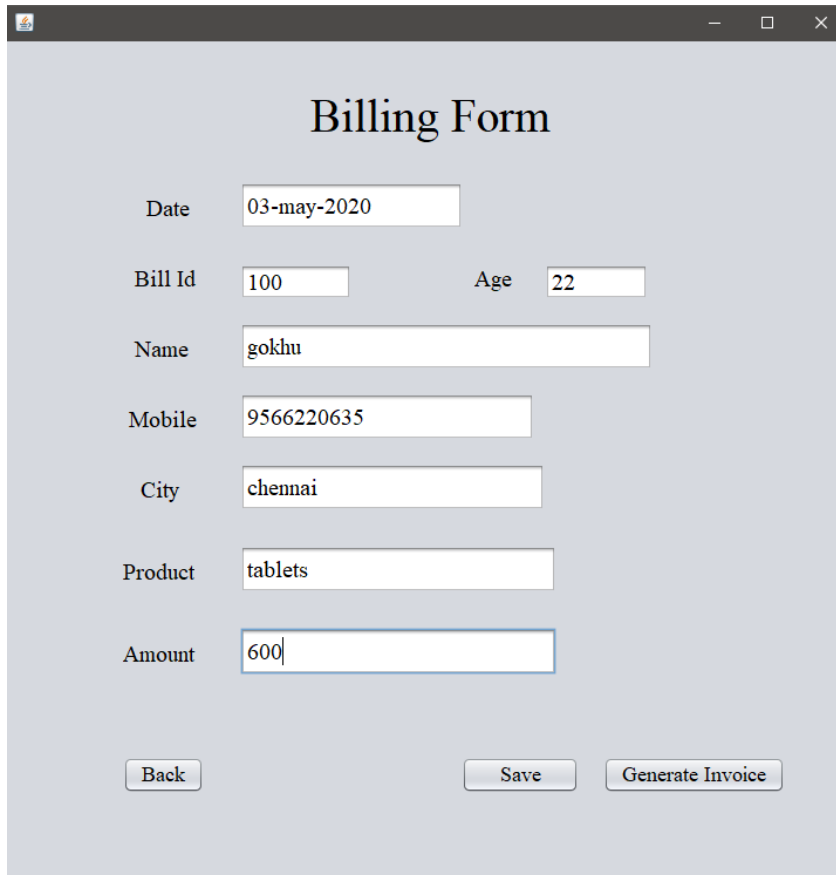
2)Option Page :-

This page lets the user select from the different options available in the form.

A screenshot of a web browser window displaying the option page for the 'Medical shop management System'. The page has a light gray background. At the top, the text 'Choose Option :-' is centered. Below this, there are three buttons with a gray gradient and a black border. The first button is labeled 'Make a Bill' in red text. The second button is labeled 'Check Records' in red text. The third button, located at the bottom right, is labeled 'Logout' in red text. The browser window has a standard title bar with a small icon on the left and minus, maximize, and close buttons on the right.

3)Billing Page :-

This page allows the user to generate a bill and save the bill, once all the data that are asked in the form are inserted.

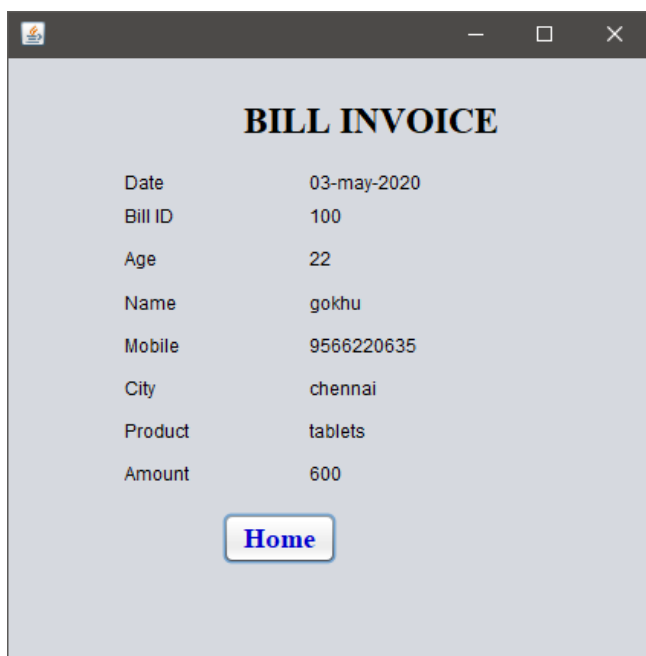


The screenshot shows a web application window titled "Billing Form". It contains several input fields for user data: Date (03-may-2020), Bill Id (100), Age (22), Name (gokhu), Mobile (9566220635), City (chennai), Product (tablets), and Amount (600). At the bottom, there are three buttons: "Back", "Save", and "Generate Invoice".

Field	Value
Date	03-may-2020
Bill Id	100
Age	22
Name	gokhu
Mobile	9566220635
City	chennai
Product	tablets
Amount	600

4)Bill Invoice Page :-

This page displays all the data that the user has entered in the billing page.



The screenshot shows a web application window titled "BILL INVOICE". It displays the data entered in the billing form in a list format. At the bottom, there is a "Home" button.

Field	Value
Date	03-may-2020
Bill ID	100
Age	22
Name	gokhu
Mobile	9566220635
City	chennai
Product	tablets
Amount	600

5)Category Page :-

This page displays the different categories that are available in the system and are connected to the database.

Select Category :-

Customer Employee Manufacturer

Mediquipment Pharmacy Hospital

Supplier Doctor

Back

6)Pharmacy Page :-

This page lets the user to insert data into the database from the front end, delete the data that the user wants to delete and view the database that is stored in the database.

Pharmacy Details

Insert Details

Pharmacy ID City

Name

FAX Phone

Insert

Delete Details

Enter Pharmacy Id

Delete

View All Details

View Back

Pharmacy_ID	Pharmacy_Name	City	Fax	Phone
1	gokhul_shop_1	chennai	-9765221	9001002003
2	gokhul_shop_2	mumbai	-9837990	9001002004
3	gokhul_shop_3	bangalore	-9811880	9001002001
4	abhi_shop_1	delhi	-9862028	9010126387
5	abhi_shop_2	chennai	-9817933	9661268268

After deletion,

Pharmacy Details

Insert Details

Pharmacy ID City

Name

FAX Phone


Delete Details

Enter Pharmacy Id

View All Details

Pharmacy_ID	Pharmacy_Name	City	Fax	Phone
2	gokhul_shop_2	mumbai	-9837990	9001002004
3	gokhul_shop_3	bangalore	-9811880	9001002001
4	abhi_shop_1	delhi	-9862028	9010126387
5	abhi_shop_2	chennai	-9817933	9661268268
6	abhi_shop_3	bangalore	-9866294	9662312312

Message

 Delete Successfull

Inserting the details,

Pharmacy Details

Insert Details

Pharmacy ID City

Name

FAX Phone


Delete Details

Enter Pharmacy Id

View All Details

Pharmacy_ID	Pharmacy_Name	City	Fax	Phone
2	gokhul_shop_2	mumbai	-9837990	9001002004
3	gokhul_shop_3	bangalore	-9811880	9001002001
4	abhi_shop_1	delhi	-9862028	9010126387
5	abhi_shop_2	chennai	-9817933	9661268268
6	abhi_shop_3	bangalore	-9866294	9662312312

Message

 Saved

7) Doctor Page :-

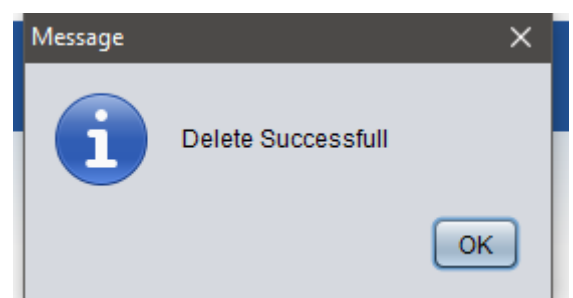
This page lets the user to insert data into the database from the front end, delete the data that the user wants to delete and view the database that is stored in the database.

The screenshot shows a window titled "Doctor Details". It contains three sections: "Insert Details", "Delete Details", and "View All Details". The "Insert Details" section has input fields for Doctor Id, Age, Name, Speciality, Gender, and Phone, along with an "Insert" button. The "Delete Details" section has an input field for "Enter Doctor Id" and a "Delete" button. The "View All Details" section has "View" and "Back" buttons. Below these sections is a table with the following data:

Doctor_ID	Doctor_Name	Speciality	Age	Mobile	Gender
10	gokhul	cardiologists	45	9831312512	m
11	abhishek	diabetologists	50	9826263264	m
12	hemanth	gynaecologists	30	9006001002	m
13	ajay	hygienists	32	9875400500	m
14	sowtham	lenroloists	28	9861621312	m

After deletion,

The screenshot shows the same "Doctor Details" window, but the "Delete" button in the "Delete Details" section is now disabled. The "Enter Doctor Id" field contains the value "10". The table below still shows the same data as before, including doctor ID 10.



Inserting the details,

Doctor Details

Insert Details

Doctor Id Age

Name

Speciality Gender

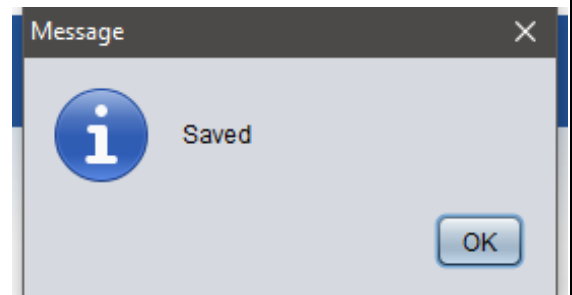
Phone

Delete Details

Enter Doctor Id

View All Details

Doctor_ID	Doctor_Name	Speciality	Age	Mobile	Gender
11	abhishek	diabetologists	50	9826263264	m
12	hemanth	gynaecologists	30	9006001002	m
13	ajay	hygienists	32	9875400500	m
14	gowtham	leprologists	28	9861621312	m
15	akashva	neurologists	28	9811254112	f



8)Customer Page :-

This page lets the user to insert data into the database from the front end, delete the data that the user wants to delete and view the database that is stored in the database.

Customer Details

Insert Details

Customer Id Age

Name

City Sex

Phone

Delete Details

Enter Customer Id

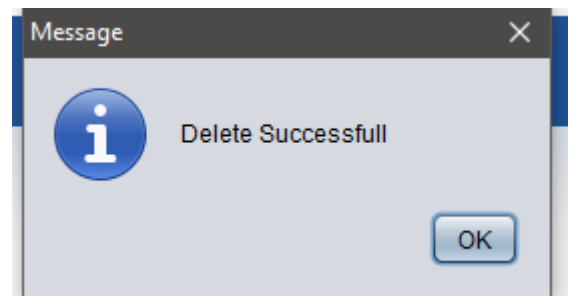
View All Details

Customer_ID	Customer_Name	Sex	City
100	anshu	m	chean
101	iqbal	m	rampu
102	priya	f	chenna
103	kanubil	m	natpa

After deletion,

The application window titled "Customer Details" contains three sections: "Insert Details", "Delete Details", and "View All Details". The "Delete Details" section shows "Enter Customer Id" with the value "100" and a "Delete" button. The "View All Details" section has a "View" button and a table of customer data.

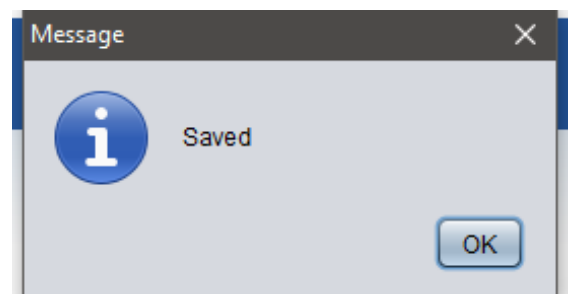
Customer_ID	Customer_Name	Sex	City
100	anshu	m	cheam
101	iqbal	m	rampu
102	priya	f	chenn
103	kaushik	m	patna



Inserting the details,

The application window titled "Customer Details" shows the "Insert Details" section with fields for Customer Id (100), Age (20), Name (gokhul), City (chennai), Sex (M), and Phone (9566220635). The "View All Details" section shows a table with 5 customers, including the newly added one.

Customer_ID	Customer_Name	Sex	City
101	iqbal	m	rampu
102	priya	f	chenn
103	kaushik	m	patna
104	niru	f	bayce



9)Manufacturer Page :-

This page lets the user to insert data into the database from the front end, delete the data that the user wants to delete and view the database that is stored in the database.

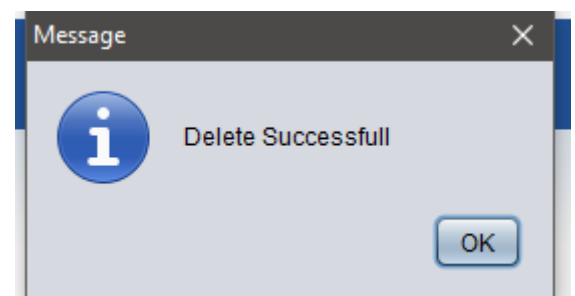
The screenshot shows a web application window titled "Manufacturer Details". It contains three main sections: "Insert Details", "Delete Details", and "View All Details". The "Insert Details" section has input fields for Manufacturer Id, Name, Email, Mobile, and City, with an "Insert" button. The "Delete Details" section has an input field for "Enter Manufacturer Id" and a "Delete" button. The "View All Details" section has "View" and "Back" buttons. Below these sections is a table with 5 rows of manufacturer data.

Manufacturer_ID	Name	Email	Mobile	City
1001	medicyl	med2@gmail	9212123134	mysore
1002	vitade	vit3@gmail	9812121368	chennai
1003	annuva	ann4@gmail	9001002115	delhi
1004	mandavus	man5@gmail	9122113114	hyderabad
1005	inovine	ino5@gmail	9881828112	bangalore

After deletion,

This screenshot shows the same "Manufacturer Details" application window after a deletion. The "Delete Details" section now has "1000" entered in the "Enter Manufacturer Id" field. The table below now contains 4 rows of data, as manufacturer ID 1001 has been removed.

Manufacturer_ID	Name	Email	Mobile	City
1000	hoivo	hoivo1@gmail	9112113114	bangalore
1001	medicyl	med2@gmail	9212123134	mysore
1002	vitade	vit3@gmail	9812121368	chennai
1003	annuva	ann4@gmail	9001002115	delhi
1004	mandavus	man5@gmail	9122113114	hyderabad



Inserting the details,

Manufacturer Details

Insert Details

Manufacturer Id

Name

Email

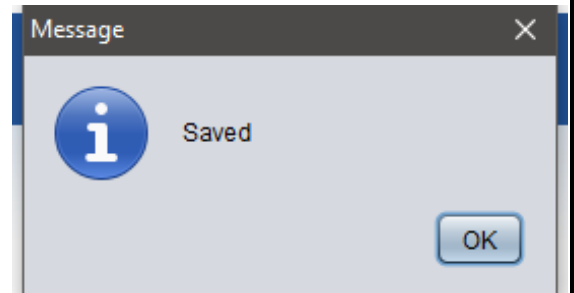
Mobile City

Delete Details

Enter Manufacturer Id

View All Details

Manufacturer_ID	Name	Email	Mobile	City
1001	medicyl	med2@gmail	9212123134	mysore
1002	vitade	vit3@gmail	9812121368	chennai
1003	annuva	ann4@gmail	9001002115	delhi
1004	mandavus	man5@gmail	9122113114	hyderabad
1005	inovine	ino5@gmail	9881828112	bangalore



10)Mediquipment Page :-

This page lets the user to insert data into the database from the front end, delete the data that the user wants to delete and view the database that is stored in the database.

Mediquipment Details

Insert Details

Code Product Type

Name

Mfg_Date Exp_Date

Price

Delete Details

Enter Code

View All Details

Code	Trade_Name	Product_Type	Mfg_Date	Exp_Date
12122	paracetamol	capsule	2016-08-06 00:00:00.0	2020-08-20 00:00:00.0
12100	quinapril	tablets	2016-08-07 00:00:00.0	2021-08-19 00:00:00.0
12115	zafirlukast	tablets	2016-09-08 00:00:00.0	2019-09-17 00:00:00.0
12116	crocim	svrun	2016-05-09 00:00:00.0	2018-05-16 00:00:00.0

After deletion,

Mediquipment Details

Insert Details

Code Product Type

Name

Mfg_Date Exp_Date

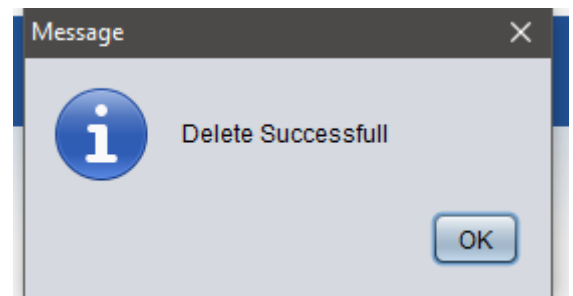
Price

Delete Details

Enter Code

View All Details

Code	Trade_Name	Product_Type	Mfg_Date	Exp_Date
12122	paracetamol	capsule	2016-08-06 00:00:00.0	2020-08-20 00:00:00.0
12100	quinapril	tablets	2016-08-07 00:00:00.0	2021-08-19 00:00:00.0
12115	zafirlukast	tablets	2016-09-08 00:00:00.0	2019-09-17 00:00:00.0
12116	crocin	syrup	2016-05-09 00:00:00.0	2018-05-16 00:00:00.0



Inserting the details,

Mediquipment Details

Insert Details

Code Product Type

Name

Mfg_Date Exp_Date

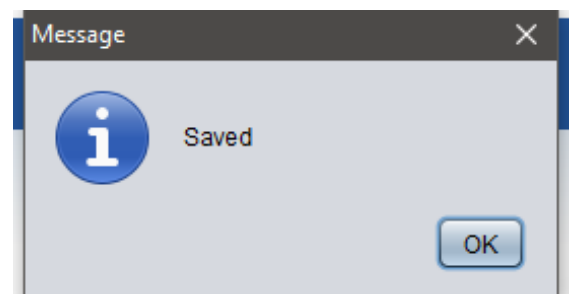
Price

Delete Details

Enter Code

View All Details

Code	Trade_Name	Product_Type	Mfg_Date	Exp_Date
12100	quinapril	tablets	2016-08-07 00:00:00.0	2021-08-19 00:00:00.0
12115	zafirlukast	tablets	2016-09-08 00:00:00.0	2019-09-17 00:00:00.0
12116	crocin	syrup	2016-05-09 00:00:00.0	2018-05-16 00:00:00.0
12117	injection	operation	2016-06-18 00:00:00.0	2023-04-23 00:00:00.0



11)Employee Page :-

This page lets the user to insert data into the database from the front end, delete the data that the user wants to delete and view the database that is stored in the database.

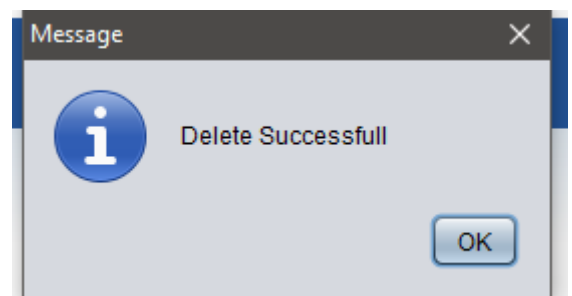
The screenshot shows the 'Employee Details' application window. It has three main sections: 'Insert Details', 'Delete Details', and 'View All Details'. The 'Insert Details' section contains input fields for Name, City, DOJ, Mobile, Age, Salary, and Sex, along with an 'Insert' button. The 'Delete Details' section has an 'Enter Mobile' field and a 'Delete' button. The 'View All Details' section has a 'View' button and a 'Back' button. Below these sections is a table displaying the current database records.

Name	City	DOJ	Mobile	Salary
arjun	bangalore	2014-08-02 00:00:00.0	9110220330	15000
ankita	mumbai	2013-03-03 00:00:00.0	9112221312	20000
ganesh	chennai	2014-04-03 00:00:00.0	9112222331	24000
mayur	delhi	2014-05-02 00:00:00.0	9113114118	25000

After deletion,

This screenshot shows the 'Employee Details' application window after a deletion. The 'Enter Mobile' field in the 'Delete Details' section now contains the value '9110220330'. The table at the bottom now only displays four records, as the employee with mobile number 9110220330 has been removed.

Name	City	DOJ	Mobile	Salary
arjun	bangalore	2014-08-02 00:00:00.0	9110220330	15000
ankita	mumbai	2013-03-03 00:00:00.0	9112221312	20000
ganesh	chennai	2014-04-03 00:00:00.0	9112222331	24000
mayur	delhi	2014-05-02 00:00:00.0	9113114118	25000



Inserting the details,

Employee Details

Insert Details

Name

City DOJ

Mobile Age

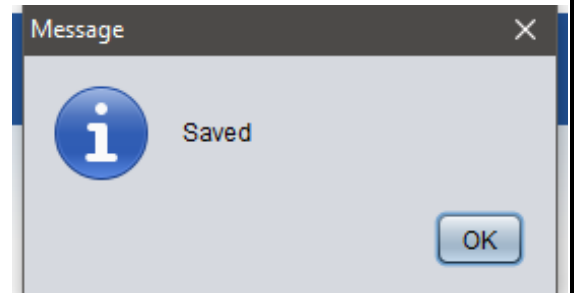
Salary Sex

Delete Details

Enter Mobile

View All Details

Name	City	DOJ	Mobile	Salary
ankita	mumbai	2013-03-03 00:00:00.0	9112221312	20000
ganesh	chennai	2014-04-03 00:00:00.0	9112222331	24000
mayur	delhi	2014-05-02 00:00:00.0	9113114118	25000
mayank	bangalore	2015-06-05 00:00:00.0	9116100200	26000



12) Supplier Page :-

This page lets the user to insert data into the database from the front end, delete the data that the user wants to delete and view the database that is stored in the database.

Supplier Details

Insert Details

Name

Email

Phone City

Delete Details

Enter Phone

View All Details

Name	City	Mobile	Email
priyank	bangalore	9812131212	pri1@gmail
trishal	mumbai	9811210000	tri2@gmail
raj	pune	9800100211	raj3@gmail
aryan	chennai	9761610610	ary4@gmail
vishak	hyderabad	9661252213	vis5@gmail

13) Hospital Page :-

This page lets the user to insert data into the database from the front end, delete the data that the user wants to delete and view the database that is stored in the database.

The screenshot shows a web application window titled "Hospital Details". It contains three main sections: "Insert Details", "Delete Details", and "View All Details".

- Insert Details:** Includes input fields for Hospital Id, Name, Email, Phone, and City, followed by an "Insert" button.
- Delete Details:** Includes an input field for "Enter Hospital Id" and a "Delete" button.
- View All Details:** Includes a "View" button and a "Back" button.

Below these sections is a table displaying the current data in the database:

Hospital_Id	Name	Email	Phone	City
50	krishna	kri6@gmail	9771700700	delhi
51	jayadeva	jay5@gmail	9772800800	bangalore
52	apollo	apo7@gmail	9812900912	mysore
53	paras	par8@gmail	9310315315	delhi
54	netvello	net9@gmail	9410055000	chennai

SQL QUERY

- 1) This query displays the name and salary of all employee's whose salary is greater than the average salary of all employee's working in Bangalore.

```
SQL> select name,salary
2      from employee
3      group by name,salary
4      having salary > (select avg(salary)
5                        from employee
6                        where city='bangalore')
7      order by 2 asc;
```

NAME	SALARY
ganesh	24000
mayur	25000
mayank	26000
satyam	27000

- 2) This query updates the medicine's price in mediquipment table to 300, if the manufacturer is from Pune.

CODE	TRADE_NAME	PRODUCT_TYPE	MFG_DATE	EXP_DATE
PRICE	CID			
12205	crocin	syrup	15-DEC-17	15-DEC-18
499				
12112	paramax	tablets	29-APR-16	20-FEB-20
308.16	1009			

11 rows selected.

```
SQL> update mediquipment set price=300
2   where cid in (select cid
3                 from manufacturer
4                 where city='pune');
1 row updated.
```

(Before execution)

CODE	TRADE_NAME	PRODUCT_TYPE	MFG_DATE	EXP_DATE
PRICE	CID			
12205	crocin	syrup	15-DEC-17	15-DEC-18
499				
12112	paramax	tablets	29-APR-16	20-FEB-20
300	1009			

11 rows selected.

SQL>

(After execution)

- 3) This query displays the name of pharmacy, pharmacy id and the count of the manufacturer id, if a pharmacy is taking medicine's from more than one manufacturer.

```
SQL> select p.name,pharid,count(distinct cid) as count1
2   from pharmacy p, manufacturer
3   where phid = pharid
4   group by p.name, pharid
5   having count(*)>1;
```

NAME	PHARID	COUNT1
abbott	1	2
allergan	2	3

SQL> |

- 4) This query displays the pharmacy name, employee name and employee salary, who are working in a pharmacy except the female employee.

```
SQL> select p.name, e.name, e.salary
2      from employee e, pharmacy p
3      where phid=pharid
4      minus
5      select p.name,e.name, e.salary
6      from employee e, pharmacy p
7      where e.sex='F';
```

NAME	NAME	SALARY
abbott	arjun	15000
janssen	ganesh	24000
novartis	daniel	19000
roche	mayank	26000
schwarz	satyam	27000

PROCEDURE :-

This procedure update's the medicine price by 100 in the mediquipment table, where the trade name is paramax.

CODE	TRADE_NAME	PRODUCT_TYPE	MFG_DATE	EXP_DATE
PRICE	CID			
12205	crocin	syrup	15-DEC-17	15-DEC-18
499				
12112	paramax	tablets	29-APR-16	20-FEB-20
300	1009			

11 rows selected.

```
SQL> drop procedure update_medi;
Procedure dropped.

SQL> create procedure update_medi as
2  begin
3  update mediquipment set price = price+100 where trade_name='paramax';
4  end;
5  /

Procedure created.

SQL> begin
2  update_medi;
3  end;
4  /

PL/SQL procedure successfully completed.
```

(Before procedure is executed)

CODE	TRADE_NAME	PRODUCT_TYPE	MFG_DATE	EXP_DATE
PRICE	CID			
12205	crocin	syrup	15-DEC-17	15-DEC-18
499				
12112	paramax	tablets	29-APR-16	20-FEB-20
400	1009			

(After procedure is executed)

Triggers:

- 1) This trigger will change the price of a medicine to 499, if the price is greater than 500.

```
SQL> create trigger price_check
2   before insert or update on mediequipment
3   for each row
4   when (new.price>500)
5   begin
6     :new.price:= 499;
7   end;
8   /

Trigger created.

SQL> insert into mediequipment values(12205,'vinyl','syrup','13-sep-16','13-mar-18',550,1003);
```

(Trigger is executed and a insertion is performed)

CODE	TRADE_NAME	PRODUCT_TYPE	MFG_DATE	EXP_DATE
PRICE	CID			
12205	vinyl	syrup	13-SEP-16	13-MAR-18
499	1003			
12112	paramax	tablets	29-APR-16	20-FEB-20
400	1009			

(Inserted price is less than 500)

- 2) This trigger will display a message if the age of a doctor is greater than 80.

```
SQL> create trigger age_check1
2   before insert or update on doctor
3   for each row
4   when (new.age>80)
5   begin
6     dbms_output.put_line('DOCTOR AGE CANNOT BE MORE THAN 80!!!');
7   end;
8   /

Trigger created.

SQL> set serveroutput on;
SQL> insert into doctor values(50,'abhishek','cardiologist',85,9566220635,'m');
DOCTOR AGE CANNOT BE MORE THAN 80!!!
```

Database creation commands

Connection commands:

```
connect system/gokhul;  
create user medicalshop22 identified by gokhul;  
grant connect, resource to medicalshop22;  
connect medicalshop22/gokhul;  
@ C:\Users\gokhu\Desktop\pharmacy.sql
```

Database commands:

```
create table pharmacy  
(  
    phid integer primary key,  
    name varchar2(20),  
    city varchar2(20),  
    fax varchar2(20),  
    phone number(10));
```

```
create table doctor  
(  
    did number(2) primary key,  
    dname varchar2(20),  
    speciality varchar2(20),  
    age number(2) not null,  
    mobile integer,  
    gender varchar2(6));
```

```
create table customer  
(  
    pid number(3) primary key,  
    name varchar2(20),
```

```
sex varchar2(6),  
city varchar2(20),  
phone integer,  
age number(2),  
did references doctor(did) on delete set null);
```

```
create table manufacturer
```

```
(  
  cid number(4) primary key,  
  name varchar2(20),  
  email varchar2(20),  
  mobile integer,  
  city varchar2(20),  
  pharid references pharmacy(phid) on delete set null);
```

```
create table mediequipment
```

```
(  
  code number(5) primary key,  
  Trade_name varchar2(20),  
  Product_Type varchar2(20),  
  Mfg_date date,  
  Exp_date date,  
  price number(10,2),  
  cid references manufacturer(cid) on delete set null);
```

```
create table supplier
```

```
(  
  name varchar2(20),  
  city varchar2(20),  
  mobile integer primary key,
```



```
email varchar2(20),  
cid references manufacturer(cid) on delete set null,  
pharid references pharmacy(phid) on delete set null);
```

```
create table employee
```

```
(  
name varchar2(20),  
city varchar2(20),  
doj date,  
mobile integer primary key,  
salary number(10),  
age number(2),  
sex varchar2(1),  
pharid references pharmacy(phid) on delete set null);
```

```
create table hospital
```

```
(  
hid number(2) primary key,  
name varchar2(20),  
email varchar2(40),  
phone integer,  
city varchar2(20),  
pharid references pharmacy(phid) on delete set null);
```

```
create table bill
```

```
(  
bid number(3) primary key,  
DOB date,  
age number(3),  
pname varchar2(20),
```

```
mobile integer,  
city varchar2(20),  
product varchar2(20),  
amount number(10,2),  
pharid references pharmacy(phid) on delete set null);
```

create table works

```
(  
pharid references pharmacy(phid) on delete cascade,  
start_date date,  
end_date date,  
primary key(pharid));
```

create table contract

```
(  
pharid references pharmacy(phid) on delete cascade,  
cid references manufacturer(cid) on delete cascade,  
start_date date,  
end_date date,  
primary key(pharid,cid));
```

create table prescribe

```
(  
DOP date,  
medicine varchar2(20),  
did references doctor(did) on delete cascade,  
pid references customer(pid) on delete cascade,  
primary key(pid,did));
```

```
select name,salary
      from employee
group by name,salary
having salary > (select avg(salary)
      from employee
      where city='bangalore')
order by 2 asc;
```

```
update mediquipment set price=300
      where cid in (select cid
      from manufacturer
      where city='pune');
```

```
select p.name,pharid,count(distinct cid) as count1
      from pharmacy p, manufacturer
      where phid = pharid
group by p.name, pharid
having count(*)>1;
```

```
select p.name, e.name, e.salary
      from employee e, pharmacy p
      where phid=pharid
minus
select p.name,e.name, e.salary
      from employee e, pharmacy p
      where e.sex='f';
```

```
create procedure update_medi as
```

```
begin
```

```
update mediquipment set price = price+100 where trade_name='paramax';
```

```
end;
```

```
/
```

```
create trigger price_check
```

```
before insert or update on mediquipment
```

```
for each row
```

```
when (new.price>500)
```

```
begin
```

```
:new.price:= 499;
```

```
end;
```

```
/
```

```
create trigger age_check1
```

```
before insert or update on doctor
```

```
for each row
```

```
when (new.age>80)
```

```
begin
```

```
dbms_output.put_line('DOCTOR AGE CANNOT BE  
MORE THAN 80!!!');
```

```
end;
```

```
/
```

```
commit;
```

Application Code

1) Billing Page:

```
private void InsertActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try{Class.forName("oracle.jdbc.OracleDriver");  
        Connection con1=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-  
C3RN1T1:1521:XE","medicalshop22","gokhul");  
        String sql="Insert into bill(bid,dob,age,pname,mobile,city,product,amount) values(?,?,?,?,?,?,?)";  
        PreparedStatement pst=con1.prepareStatement(sql);  
        pst.setString(1,b.getText());  
        pst.setString(2,a.getText());  
        pst.setString(3,c.getText());  
        pst.setString(4,d.getText());  
        pst.setString(5,e.getText());  
        pst.setString(6,f.getText());  
        pst.setString(7,g.getText());  
        pst.setString(8,h.getText());  
        pst.execute();  
  
        JOptionPane.showMessageDialog(null,"Saved");  
        pst.close();  
        con1.close();  
    }catch(Exception e) {JOptionPane.showMessageDialog(null,"Error="+e);}  
}
```

2) Pharmacy Page:

```
private void InsertActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try{Class.forName("oracle.jdbc.OracleDriver");  
        Connection con1=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-  
C3RN1T1:1521:XE","medicalshop22","gokhul");  
        String sql="Insert into pharmacy(phid,name,city,fax,phone) values(?,?,?,?,?)";  
        PreparedStatement pst=con1.prepareStatement(sql);  
        pst.setString(1,jTextField1.getText());  
        pst.setString(2,jTextField3.getText());  
        pst.setString(3,jTextField2.getText());  
        pst.setString(4,jTextField4.getText());  
        pst.setString(5,jTextField5.getText());  
        pst.execute();  
        jTextField1.setText("");  
        jTextField2.setText("");  
        jTextField4.setText("");  
        jTextField3.setText("");  
        jTextField5.setText("");  
        JOptionPane.showMessageDialog(null,"Saved");  
        pst.close();  
        con1.close();  
    }catch(Exception e) {JOptionPane.showMessageDialog(null,"Error="+e);}  
}
```

```

private void viewActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{Class.forName("oracle.jdbc.OracleDriver");
        Connection con1=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-
C3RN1T1:1521:XE","medicalshop22","gokhul");
        Statement stmt=con1.createStatement();
        ResultSet rs=stmt.executeQuery("select * from pharmacy");
        String show="";
        show="Pharmacy_ID\t"+"Pharmacy_Name\t"+"City\t"+"Fax\t"+"Phone\n";
        show+="_____\t"+"_____\t"+"_____\t"+"_____\t"+"_____\n";
        while(rs.next())
        {
            int phid=rs.getInt("phid");
            String name=rs.getString("name");
            String city=rs.getString("city");
            String fax = rs.getString("fax");
            String phone=rs.getString("phone");
            show+=phid+"\t"+name+"\t\t"+city+"\t"+fax+"\t"+phone+"\n";
            jTextArea1.setText(show);
        }
        rs.close();
        stmt.close();
        con1.close();
    }catch(Exception e) {JOptionPane.showMessageDialog(null,"Error="+e);}
}

```

```

private void DeleteActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{Class.forName("oracle.jdbc.OracleDriver");

        Connection con1=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-
C3RN1T1:1521:XE","medicalshop22","gokhul");

        String phid = jTextField6.getText();

        String sql = "delete from pharmacy where phid="+phid;

        PreparedStatement stmt2=con1.prepareStatement(sql);

        stmt2.execute();

        jTextField6.setText("");

        JOptionPane.showMessageDialog(null,"Delete Successfull");
    }
    catch(Exception e){JOptionPane.showMessageDialog(null,"Error="+e);}
}

```


3) Doctor Page:

```
private void InsertActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try{Class.forName("oracle.jdbc.OracleDriver");  
        Connection con1=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-  
C3RN1T1:1521:XE","medicalshop22","gokhul");  
        String sql="Insert into doctor(did,dname,speciality,age,mobile,gender) values(?,?,?,?,?)";  
        PreparedStatement pst=con1.prepareStatement(sql);  
        pst.setString(1,jTextField1.getText());  
        pst.setString(2,jTextField3.getText());  
        pst.setString(3,jTextField4.getText());  
        pst.setString(4,jTextField2.getText());  
        pst.setString(5,jTextField6.getText());  
        pst.setString(6,jTextField5.getText());  
        pst.execute();  
        jTextField1.setText("");  
        jTextField2.setText("");  
        jTextField4.setText("");  
        jTextField3.setText("");  
        jTextField5.setText("");  
        jTextField6.setText("");  
        JOptionPane.showMessageDialog(null,"Saved");  
        pst.close();  
        con1.close();  
    }catch(Exception e) {JOptionPane.showMessageDialog(null,"Error="+e);}  
}
```

```

private void viewActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{Class.forName("oracle.jdbc.OracleDriver");

        Connection con1=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-
C3RN1T1:1521:XE","medicalshop22"," gokhul");

        Statement stmt=con1.createStatement();

        ResultSet rs=stmt.executeQuery("select * from doctor");

        String show="";

        show="Doctor_ID\t"+"Doctor_Name\t"+"Speciality\t"+"Age\t"+"Mobile\t"+"Gender\n";
        show+="_____\t"+"_____\t"+"_____\t"+"_____\t"+"_____\t"+"_____\n";
        while(rs.next())
        {
            int did=rs.getInt("did");

            String dname=rs.getString("dname");

            String speciality=rs.getString("speciality");

            String age = rs.getString("age");

            String mobile=rs.getString("mobile");

            String gender=rs.getString("gender");

            show+=did+"\t"+dname+"\t"+speciality+"\t"+age+"\t"+mobile+"\t"+gender+"\n";

            jTextArea1.setText(show);
        }

        rs.close();

        stmt.close();

        con1.close();

    }catch(Exception e) {JOptionPane.showMessageDialog(null,"Error="+e);}
}

```

```

private void DeleteActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{Class.forName("oracle.jdbc.OracleDriver");

        Connection con1=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-
C3RN1T1:1521:XE","medicalshop22","gokhul");

        String did = jTextField7.getText();

        String sql = "delete from doctor where did="+did;

        PreparedStatement stmt2=con1.prepareStatement(sql);

        stmt2.execute();

        jTextField7.setText("");

        JOptionPane.showMessageDialog(null,"Delete Successfull");
    }
    catch(Exception e){JOptionPane.showMessageDialog(null,"Error="+e);}
}

```

4) Customer Page:

```
private void InsertActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try{Class.forName("oracle.jdbc.OracleDriver");  
        Connection con1=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-  
C3RN1T1:1521:XE","medicalshop22","gokhul");  
        String sql="Insert into customer(pid,name,sex,city,phone,age) values(?,?,?,?,?)";  
        PreparedStatement pst=con1.prepareStatement(sql);  
        pst.setString(1,jTextField1.getText());  
        pst.setString(2,jTextField3.getText());  
        pst.setString(3,jTextField5.getText());  
        pst.setString(4,jTextField4.getText());  
        pst.setString(5,jTextField6.getText());  
        pst.setString(6,jTextField2.getText());  
        pst.execute();  
        jTextField1.setText("");  
        jTextField2.setText("");  
        jTextField4.setText("");  
        jTextField3.setText("");  
        jTextField5.setText("");  
        jTextField6.setText("");  
        JOptionPane.showMessageDialog(null,"Saved");  
        pst.close();  
        con1.close();  
    }catch(Exception e) {JOptionPane.showMessageDialog(null,"Error="+e);}  
}
```

```

private void viewActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{Class.forName("oracle.jdbc.OracleDriver");
        Connection con1=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-
C3RN1T1:1521:XE","medicalshop22"," gokhul");

        Statement stmt=con1.createStatement();
        ResultSet rs=stmt.executeQuery("select * from customer");

        String show="";
        show="Customer_ID\t"+"Customer_Name\t"+"Sex\t"+"City\t"+"Phone\t"+"Age\n";
        show+="_____\\t"+"_____\\t"+"__\\t"+"____\\t"+"____\\t"+"____\\n";
        while(rs.next())
        {
            int pid=rs.getInt("pid");
            String name=rs.getString("name");
            String sex=rs.getString("sex");
            String city = rs.getString("city");
            String phone=rs.getString("phone");
            String age=rs.getString("age");
            show+=pid+"\t"+name+"\t\t"+sex+"\t"+city+"\t"+phone+"\t"+age+"\n";
            jTextArea1.setText(show);
        }
        rs.close();
        stmt.close();
        con1.close();
    }catch(Exception e) {JOptionPane.showMessageDialog(null,"Error="+e);}
}

```

```
private void DeleteActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try{Class.forName("oracle.jdbc.OracleDriver");  
        Connection con1=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-  
C3RN1T1:1521:XE","medicalshop22","gokhul");  
        String pid = jTextField7.getText();  
        String sql = "delete from customer where pid="+pid;  
        PreparedStatement stmt2=con1.prepareStatement(sql);  
        stmt2.execute();  
        jTextField7.setText("");  
        JOptionPane.showMessageDialog(null,"Delete Successfull");  
    }  
    catch(Exception e){JOptionPane.showMessageDialog(null,"Error="+e);}  
}
```

5) Manufacturer Page:

```
private void InsertActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try{Class.forName("oracle.jdbc.OracleDriver");  
        Connection con1=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-  
C3RN1T1:1521:XE","medicalshop22","gokhul");  
        String sql="Insert into manufacturer(cid,name,email,mobile,city) values(?,?,?,?,?)";  
        PreparedStatement pst=con1.prepareStatement(sql);  
        pst.setString(1,jTextField1.getText());  
        pst.setString(2,jTextField2.getText());  
        pst.setString(3,jTextField3.getText());  
        pst.setString(4,jTextField4.getText());  
        pst.setString(5,jTextField5.getText());  
        pst.execute();  
        jTextField1.setText("");  
        jTextField2.setText("");  
        jTextField4.setText("");  
        jTextField3.setText("");  
        jTextField5.setText("");  
        JOptionPane.showMessageDialog(null,"Saved");  
        pst.close();  
        con1.close();  
    }catch(Exception e) {JOptionPane.showMessageDialog(null,"Error="+e);}  
}
```

```

private void viewActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{Class.forName("oracle.jdbc.OracleDriver");
        Connection con1=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-
C3RN1T1:1521:XE","medicalshop22"," gokhul");

        Statement stmt=con1.createStatement();
        ResultSet rs=stmt.executeQuery("select * from manufacturer");
        String show="";
        show="Manufacturer_ID\t"+"Name\t"+"Email\t"+"Mobile\t"+"City\n";
        show+="_____\\t"+"____\\t"+"____\\t"+"____\\t"+"____\\n";
        while(rs.next())
        {
            int cid=rs.getInt("cid");
            String name=rs.getString("name");
            String email=rs.getString("email");
            String mobile = rs.getString("mobile");
            String city=rs.getString("city");
            show+=cid+"\t\t"+name+"\t\t"+email+"\t\t"+mobile+"\t\t"+city+"\n";
            jTextArea1.setText(show);
        }
        rs.close();
        stmt.close();
        con1.close();
    }catch(Exception e) {JOptionPane.showMessageDialog(null,"Error="+e);}
}

```



```

private void DeleteActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{Class.forName("oracle.jdbc.OracleDriver");

        Connection con1=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-
C3RN1T1:1521:XE","medicalshop22","gokhul");

        String cid = jTextField6.getText();

        String sql = "delete from manufacturer where cid="+cid;

        PreparedStatement stmt2=con1.prepareStatement(sql);

        stmt2.execute();

        jTextField6.setText("");

        JOptionPane.showMessageDialog(null,"Delete Successfull");
    }
    catch(Exception e){JOptionPane.showMessageDialog(null,"Error="+e);}
}

```

6) Mediquipment Page:

```
private void InsertActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try{Class.forName("oracle.jdbc.OracleDriver");  
        Connection con1=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-  
C3RN1T1:1521:XE","medicalshop22","gokhul");  
        String sql="Insert into mediquipment(code,trade_name,product_type,mfg_date,exp_date,price)  
values(?,?,?,?,?,?)";  
        PreparedStatement pst=con1.prepareStatement(sql);  
        pst.setString(1,jTextField1.getText());  
        pst.setString(2,jTextField3.getText());  
        pst.setString(3,jTextField2.getText());  
        pst.setString(4,jTextField4.getText());  
        pst.setString(5,jTextField5.getText());  
        pst.setString(6,jTextField6.getText());  
        pst.execute();  
        jTextField1.setText("");  
        jTextField2.setText("");  
        jTextField4.setText("");  
        jTextField3.setText("");  
        jTextField5.setText("");  
        jTextField6.setText("");  
        JOptionPane.showMessageDialog(null,"Saved");  
        pst.close();  
        con1.close();  
    }catch(Exception e) {JOptionPane.showMessageDialog(null,"Error="+e);}  
}
```

```

private void viewActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{Class.forName("oracle.jdbc.OracleDriver");
        Connection con1=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-
C3RN1T1:1521:XE","medicalshop22","gokhul");
        Statement stmt=con1.createStatement();
        ResultSet rs=stmt.executeQuery("select * from mediequipment");
        String show="";
        show="Code\t"+"Trade_Name\t"+"Product_Type\t"+"Mfg_Date\t"+"Exp_Date\t"+"Price\n";
        show+="_____\t"+"_____\t"+"_____\t"+"_____\t\t"+"_____\t\t"+"_____\t\t"+"_____\n";
        while(rs.next())
        {
            int code=rs.getInt("code");
            String trade_name=rs.getString("trade_name");
            String product_type=rs.getString("product_type");
            String mfg_date = rs.getString("mfg_date");
            String exp_date=rs.getString("exp_date");
            int price=rs.getInt("price");
            show+=code+"\t"+trade_name+"\t"+product_type+"\t"+mfg_date+"\t"+exp_date+"\t"+price+"\n";
            jTextArea1.setText(show);
        }
        rs.close();
        stmt.close();
        con1.close();
    }catch(Exception e) {JOptionPane.showMessageDialog(null,"Error="+e);}
}

```

```

private void DeleteActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{Class.forName("oracle.jdbc.OracleDriver");

        Connection con1=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-
C3RN1T1:1521:XE","medicalshop22","gokhul");

        String code = jTextField7.getText();

        String sql = "delete from mediequipment where code="+code;

        PreparedStatement stmt2=con1.prepareStatement(sql);

        stmt2.execute();

        jTextField6.setText("");

        JOptionPane.showMessageDialog(null,"Delete Successfull");
    }
    catch(Exception e){JOptionPane.showMessageDialog(null,"Error="+e);}
}

```

7) Supplier Page:

```
private void InsertActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try{Class.forName("oracle.jdbc.OracleDriver");  
        Connection con1=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-  
C3RN1T1:1521:XE","medicalshop22","gokhul");  
        String sql="Insert into supplier(name,city,mobile,email) values(?,?,?,?)";  
        PreparedStatement pst=con1.prepareStatement(sql);  
        pst.setString(1,jTextField1.getText());  
        pst.setString(2,jTextField4.getText());  
        pst.setString(3,jTextField3.getText());  
        pst.setString(4,jTextField2.getText());  
        pst.execute();  
        jTextField1.setText("");  
        jTextField2.setText("");  
        jTextField4.setText("");  
        jTextField3.setText("");  
        JOptionPane.showMessageDialog(null,"Saved");  
        pst.close();  
        con1.close();    }catch(Exception e) {JOptionPane.showMessageDialog(null,"Error="+e);}  
    }
```

```

private void viewActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{Class.forName("oracle.jdbc.OracleDriver");
        Connection con1=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-
C3RN1T1:1521:XE","medicalshop22"," gokhul");

        Statement stmt=con1.createStatement();
        ResultSet rs=stmt.executeQuery("select * from supplier");
        String show="";
        show="Name\t"+"City\t"+"Mobile\t"+"Email\n";
        show+="____\t"+"____\t"+"____\t"+"____\n";
        while(rs.next())
        {
            String name = rs.getString("name");
            String city=rs.getString("city");
            String mobile=rs.getString("mobile");
            String email = rs.getString("email");
            show+=name+"\t"+city+"\t"+mobile+"\t"+email+"\n";
            JTextArea1.setText(show);
        }
        rs.close();
        stmt.close();
        con1.close();
    }catch(Exception e) {JOptionPane.showMessageDialog(null,"Error="+e);}
}

```

```

private void DeleteActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{Class.forName("oracle.jdbc.OracleDriver");

        Connection con1=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-
C3RN1T1:1521:XE","medicalshop22","gokhul");

        String mobile = jTextField5.getText();

        String sql = "delete from supplier where mobile="+mobile;

        PreparedStatement stmt2=con1.prepareStatement(sql);

        stmt2.execute();

        jTextField5.setText("");

        JOptionPane.showMessageDialog(null,"Delete Successful");
    }
    catch(Exception e){JOptionPane.showMessageDialog(null,"Error="+e);}
}

```

8) Employee Page:

```
private void InsertActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try{Class.forName("oracle.jdbc.OracleDriver");  
        Connection con1=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-  
C3RN1T1:1521:XE","medicalshop22","gokhul");  
        String sql="Insert into employee(name,city,doj,mobile,salary,age,sex) values(?,?,?,?,?,?,?)";  
        PreparedStatement pst=con1.prepareStatement(sql);  
        pst.setString(1,jTextField1.getText());  
        pst.setString(2,jTextField2.getText());  
        pst.setString(3,jTextField3.getText());  
        pst.setString(4,jTextField4.getText());  
        pst.setString(5,jTextField6.getText());  
        pst.setString(6,jTextField5.getText());  
        pst.setString(7,jTextField7.getText());  
        pst.execute();  
        jTextField1.setText("");  
        jTextField2.setText("");  
        jTextField4.setText("");  
        jTextField3.setText("");  
        jTextField5.setText("");  
        jTextField6.setText("");  
        jTextField7.setText("");  
        JOptionPane.showMessageDialog(null,"Saved");  
        pst.close();  
        con1.close();  
    }catch(Exception e) {JOptionPane.showMessageDialog(null,"Error="+e);}  
}
```



```

private void viewActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{Class.forName("oracle.jdbc.OracleDriver");
        Connection
con1=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","medicalshop22","gokhul");
        Statement stmt=con1.createStatement();
        ResultSet rs=stmt.executeQuery("select * from employee");
        String show="";
        show="Name\t"+"City\t"+"DOJ\t"+"Mobile\t"+"Salary\t"+"Age\t"+"Sex\n";
        show+="____\t"+"____\t"+"____\t"+"____\t"+"____\t"+"____\t"+"____\n";
        while(rs.next())
        {
            String name=rs.getString("name");
            String city=rs.getString("city");
            String doj=rs.getString("doj");
            String mobile = rs.getString("mobile");
            int salary=rs.getInt("salary");
            int age=rs.getInt("age");
            String sex=rs.getString("sex");
            show+=name+"\t"+"city+"\t"+"doj+"\t"+"mobile+"\t"+"salary+"\t"+"age+"\t"+"sex+"\n";
            jTextArea1.setText(show);
        }
        rs.close();
        stmt.close();
        con1.close();
    }catch(Exception e) {JOptionPane.showMessageDialog(null,"Error="+e);}
}

```

```

private void DeleteActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{Class.forName("oracle.jdbc.OracleDriver");

        Connection
con1=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","medicalshop22","gokhul");

        String mobile = jTextField8.getText();

        String sql = "delete from employee where mobile="+mobile;

        PreparedStatement stmt2=con1.prepareStatement(sql);

        stmt2.execute();

        jTextField8.setText("");

        JOptionPane.showMessageDialog(null,"Delete Successful");
    }
    catch(Exception e){JOptionPane.showMessageDialog(null,"Error="+e);}
}

```

9) Hospital Page:

```
private void InsertActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try{Class.forName("oracle.jdbc.OracleDriver");  
    Connection con1=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-  
C3RN1T1:1521:XE","medicalshop22"," gokhul");  
        String sql="Insert into hospital(hid,name,email,phone,city) values(?,?,?,?,?)";  
        PreparedStatement pst=con1.prepareStatement(sql);  
        pst.setString(1,jTextField1.getText());  
        pst.setString(2,jTextField2.getText());  
        pst.setString(3,jTextField3.getText());  
        pst.setString(4,jTextField4.getText());  
        pst.setString(5,jTextField5.getText());  
        pst.execute();  
        jTextField1.setText("");  
        jTextField2.setText("");  
        jTextField4.setText("");  
        jTextField3.setText("");  
        jTextField5.setText("");  
        JOptionPane.showMessageDialog(null,"Saved");  
        pst.close();  
        con1.close();  
    }catch(Exception e) {JOptionPane.showMessageDialog(null,"Error="+e);}  
}
```

```

private void viewActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{Class.forName("oracle.jdbc.OracleDriver");
        Connection
con1=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","medicalshop22","gokhul");
        Statement stmt=con1.createStatement();
        ResultSet rs=stmt.executeQuery("select * from hospital");
        String show="";
        show="Hospital_Id\t"+"Name\t"+"Email\t"+"Phone\t"+"City\n";
        show+="_____\t"+"_____\t"+"_____\t"+"_____\t"+"_____\n";
        while(rs.next())
        {
            int hid=rs.getInt("hid");
            String name=rs.getString("name");
            String email=rs.getString("email");
            String phone = rs.getString("phone");
            String city=rs.getString("city");
            show+=hid+"\t"+name+"\t"+email+"\t"+phone+"\t"+city+"\n";
            jTextArea1.setText(show);
        }
        rs.close();
        stmt.close();
        con1.close();
    }catch(Exception e) {JOptionPane.showMessageDialog(null,"Error="+e);}
}

```

```

private void DeleteActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{Class.forName("oracle.jdbc.OracleDriver");

        Connection
con1=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","medicalshop22","gokhul");

        String hid = jTextField6.getText();

        String sql = "delete from hospital where hid="+hid;

        PreparedStatement stmt2=con1.prepareStatement(sql);

        stmt2.execute();

        jTextField6.setText("");

        JOptionPane.showMessageDialog(null,"Delete Successful");

    }

    catch(Exception e){JOptionPane.showMessageDialog(null,"Error="+e);}
}

```