FUNDAMENTALS OF DATA MANAGEMENT

COS20015

NAME: SYED OMAIR MAQDOOM MOHIUDDIN

STUDENT ID: 102873768

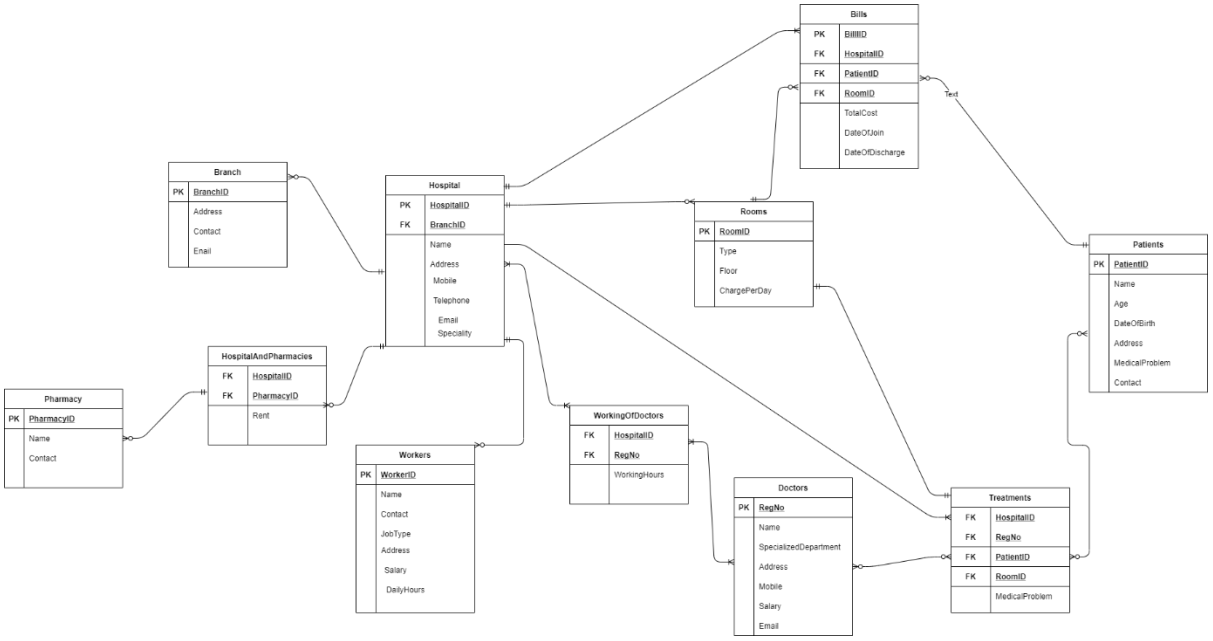
TUTOR: Harindu Korala

Design Report

This is a report for creating a hospital database. The data storage chosen was MYSQL. In the data base we have hospital branches if it contains any, doctors and workers who work for the hospital. Patients who visit hospital and are treated by 1 or many doctors. If they are admitted in hospital then the room allotted to them, bill at the time of discharge. Every hospital has one or more specialities, doctors are specialized in one or more department of medication. Each hospital may or may not have a pharmacy. So, looking at all these aspects I created a database to store all the information of the hospital.

Every table in the database has primary and foreign key constraints. They also contain indexes to retrieve the data from the database more quickly than otherwise.

ER Diagram for the Database:



Normalisation:

As you can see from the Entity relation diagram, it is in the 3NF. Normalization is like filtering some dirty water to remove all the dirt from the water. In filtering water, we have many filters and phases, in the same way we have many phases in normalisation (1NF, 2NF, 3Nf, …). Normalization helps us by getting rid of Redundancy. Redundancy is something which we have more than once. For instance, A patient can visit the hospital multiple times, or a doctor can treat a patient multiple times, so, this patient with their id will have multiple rows with similar data.

So, this redundant data leads to various problems like, INSERT anomalies, UPDATE anomalies, DELETE anomalies and many other. Using Normalisation, we can reduce data redundancy. Redundancy is expressed in terms of dependencies. Various Normal Forms are defined to have certain types of dependency. 1NF there are no composite or multi-valued attributes, in other words every attribute in the table should be single valued attributes. All the tables in the database are in 1NF.

To be in 2NF any relation should be in 1NF and it should not contain no Partial dependency. So, identification of functional dependency is necessary for second normal form. 2NF handles the UPDATE anomalies. Any primary key in case of 2NF cannot be a composite key in case it arises any partial dependency.

Consider a relation R (W, X, Y, Z) having functional dependency (WX->YZ, XY->Z}

Closure of (WX) = {W, X, Y, Z}. So, the Super key will be "WX".

In WX → YZ (WX is the Super Key and Y, Z is non-prime)

In XY → Z (XY is non-prime and Z is non=prime which is allowed in 2NF)

The above relation is in 2Nf because there is no prime attribute is deriving non-prime attribute that is there is no partial functional dependency.

Although it is not in 3NF because non=prime attribute is deriving non-prime attribute.

So, removing all the partial dependencies from the database gave the 2NF.

To get the last normalised form, the database should be in 1Nf as well as 2NF. In 3NF non-prime attributes are only allowed to be functionally dependent on Super Key (combination of different attributes to from a primary key) of relation. No transitive dependency on a Super key or candidate key. 3Nf virtually eliminates all the redundancies. The goal of the third normal form is to ensure referential integrity.

Consider a relation R(X ,Y, Z) having functional dependency (XY->Z, Z->X}

Closure of (XY) = {X, Y, Z}

Closure of (YZ) = {X, Y, Z}

So, the Candidate Keys are {XY, YZ}

XY → Z (prime deriving prime)

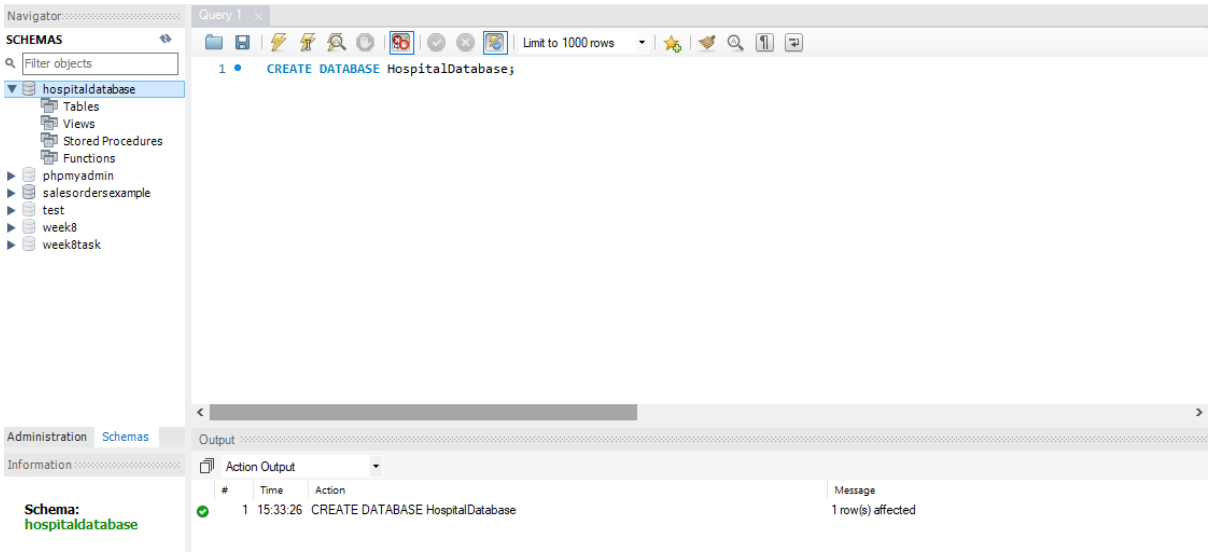
Z→ X (prime deriving prime).

There is no partial functional dependency and no transitive dependency. It is in 3NF.

From the above data, we can conclude that the database is in 3NF.

Creation of Database:

Command: **CREATE DATABASE HospitalDatabase;**

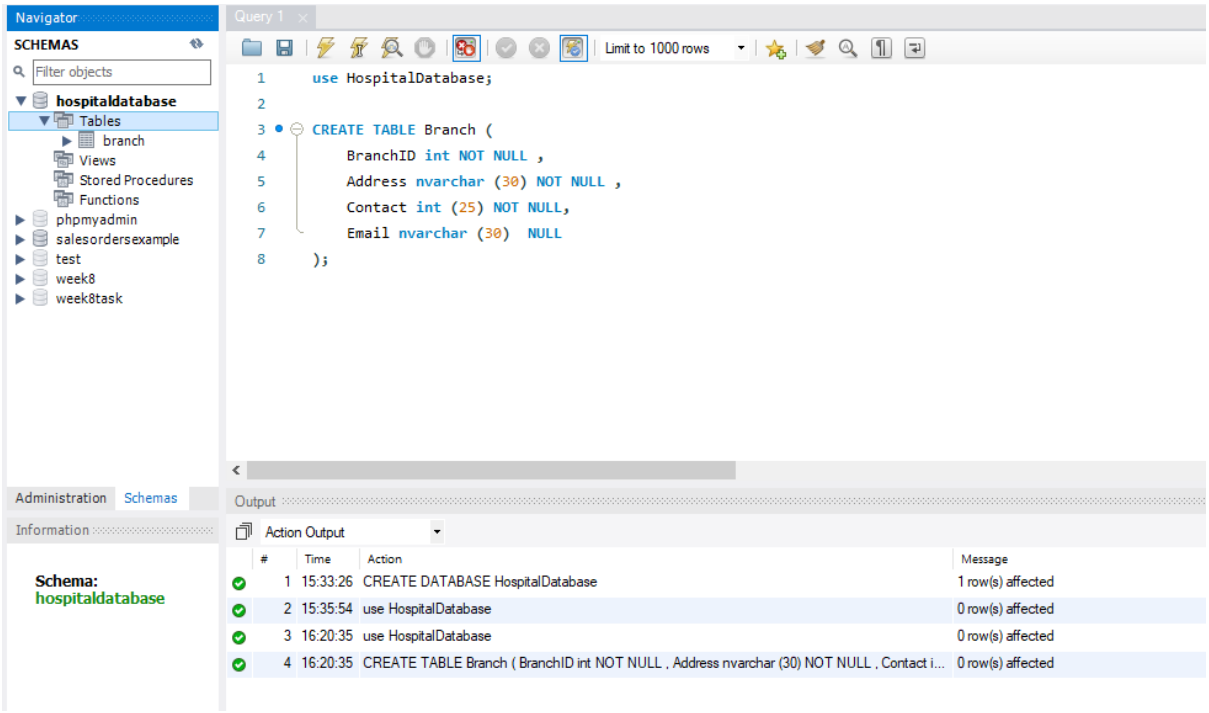


Addition of Tables:

1. Creating Branch table

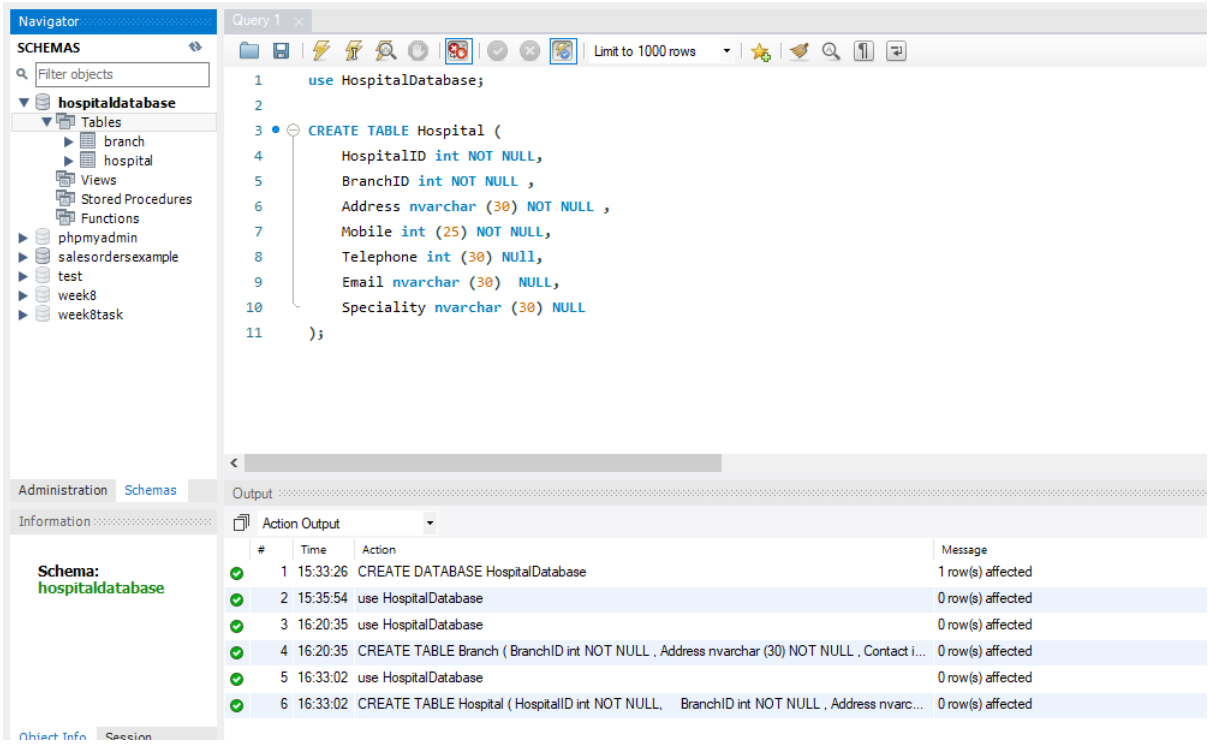
Command:

```
CREATE TABLE Branch (
        BranchID int NOT NULL ,
        Address nvarchar (30) NOT NULL ,
        Contact int (25) NOT NULL,
   Email nvarchar (30)  NULL
);
```



2. Creating Hospital Table
   Command:

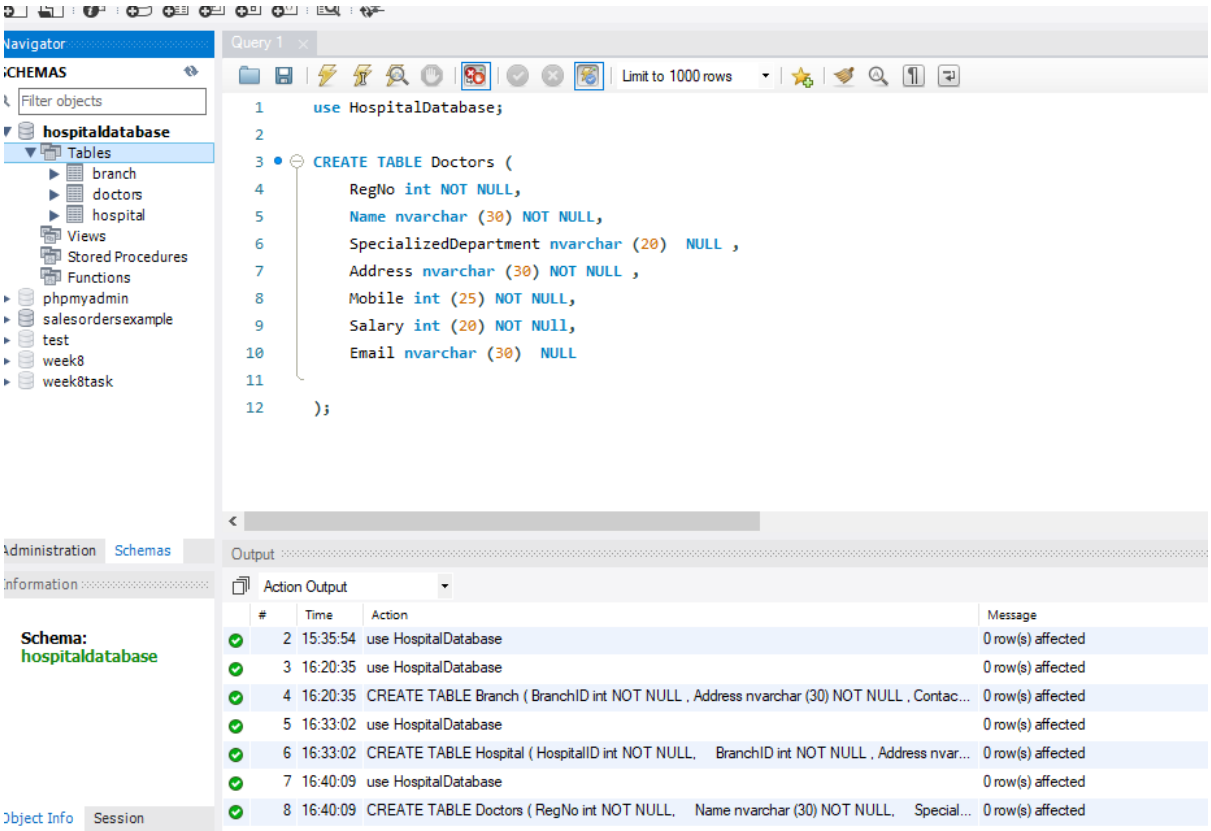
```
CREATE TABLE Hospital (
        HospitalID int NOT NULL,
   BranchID int NOT NULL ,
Name nvarchar (30) NOT NULL,
        Address nvarchar (30) NOT NULL ,
        Mobile int (25) NOT NULL,
   Telephone int (30) NULL,
   Email nvarchar (30)  NULL,
   Speciality nvarchar (30) NULL
);
```



3. Creating Doctors Table
   Command:

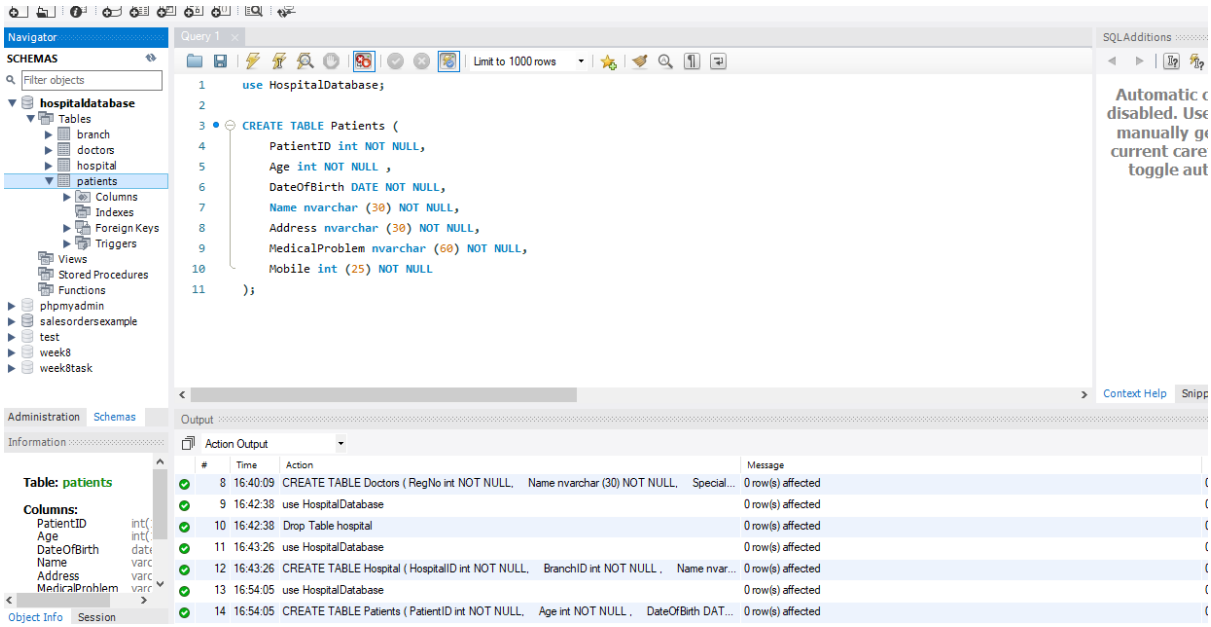
```
CREATE TABLE Doctors (
        RegNo int NOT NULL,
   Name nvarchar (30) NOT NULL,
   SpecializedDepartment nvarchar (20)  NULL ,
        Address nvarchar (30) NOT NULL ,
        Mobile int (25) NOT NULL,
   Salary int (20) NOT NUll,
   Email nvarchar (30)  NULL

);
```



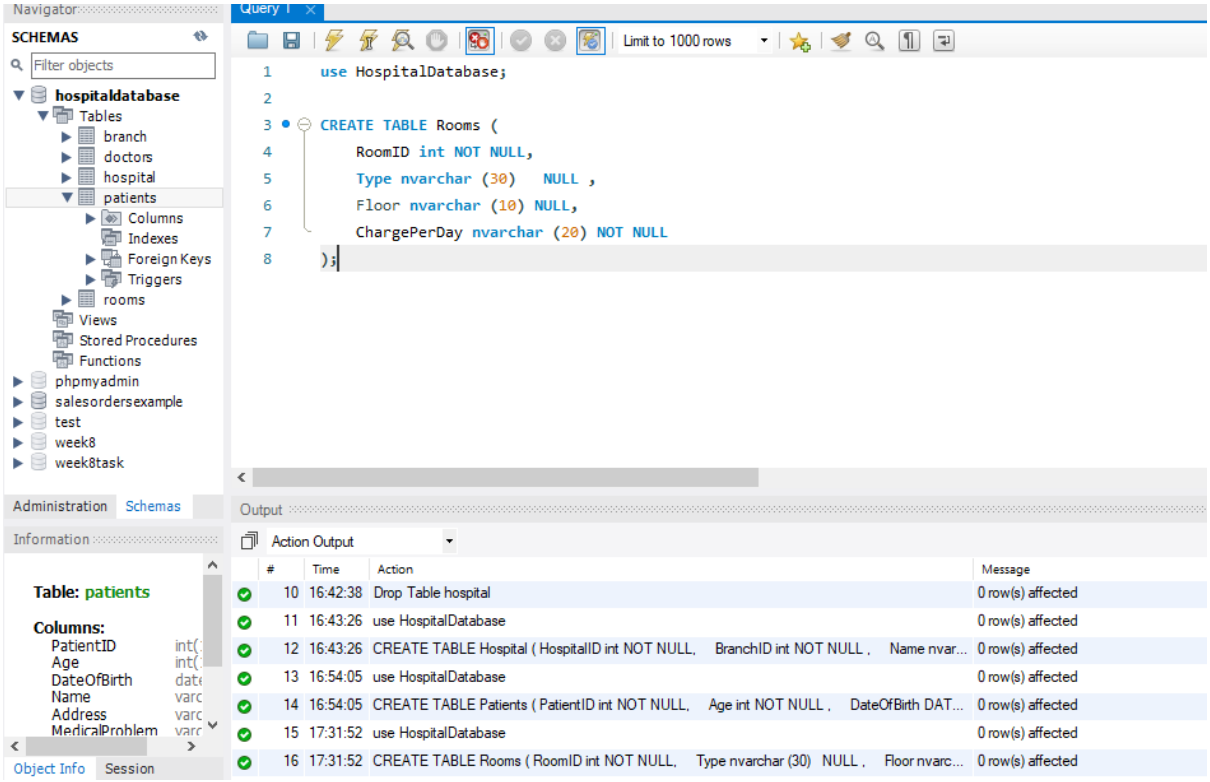
4. Creating Patients Table
   Command:

```
CREATE TABLE Patients (
        PatientID int NOT NULL,
   Age int NOT NULL ,
   DateOfBirth DATE NOT NULL,
   Name nvarchar (30) NOT NULL,
        Address nvarchar (30) NOT NULL,
   MedicalProblem nvarchar (60) NOT NULL,
        Mobile int (25) NOT NULL
);
```



5. Creating Rooms/Wards Table
   Command:

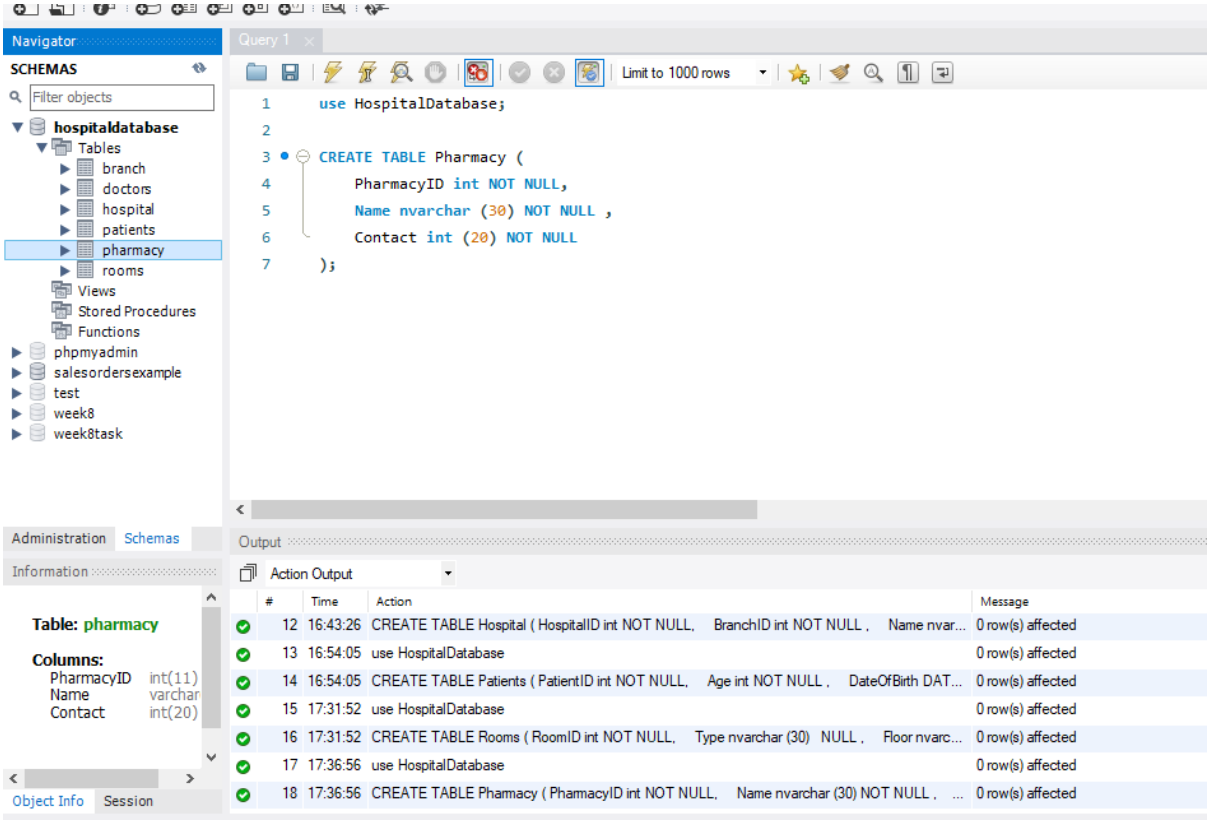
```
CREATE TABLE Rooms (
        RoomID int NOT NULL,
   Type nvarchar (30)   NULL ,
   Floor nvarchar (10) NULL,
   ChargePerDay nvarchar (20) NOT NULL
);
```



6. Create Pharmacy Table

Command:

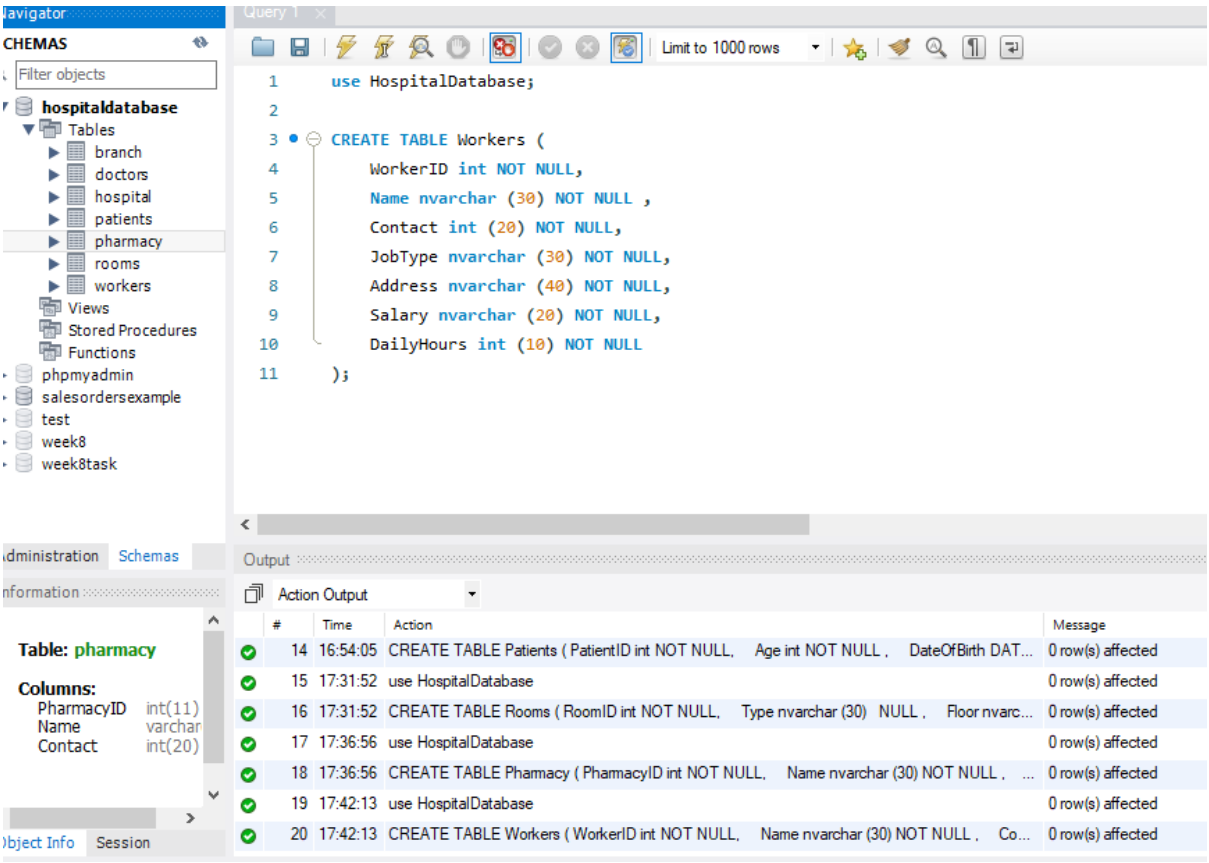
```
CREATE TABLE Pharmacy (
        PharmacyID int NOT NULL,
   Name nvarchar (30) NOT NULL ,
   Contact int (20) NOT NULL
);
```



7. **Create Workers Table**

   Command:

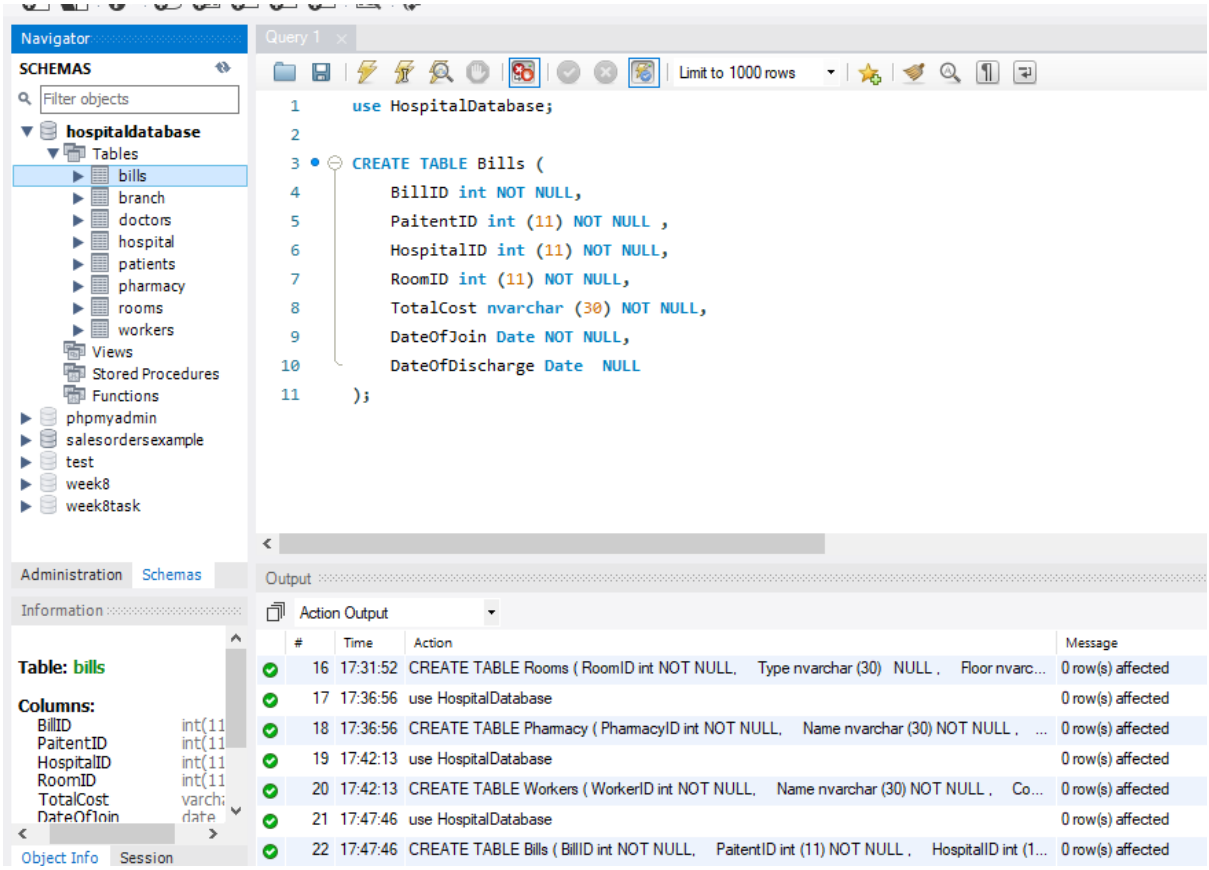
```
CREATE TABLE Workers (
        WorkerID int NOT NULL,
   Name nvarchar (30) NOT NULL ,
   Contact int (20) NOT NULL,
   JobType nvarchar (30) NOT NULL,
   Address nvarchar (40) NOT NULL,
   Salary nvarchar (20) NOT NULL,
   DailyHours int (10) NOT NULL
);
```



8. **Create Bills Table**

Command:

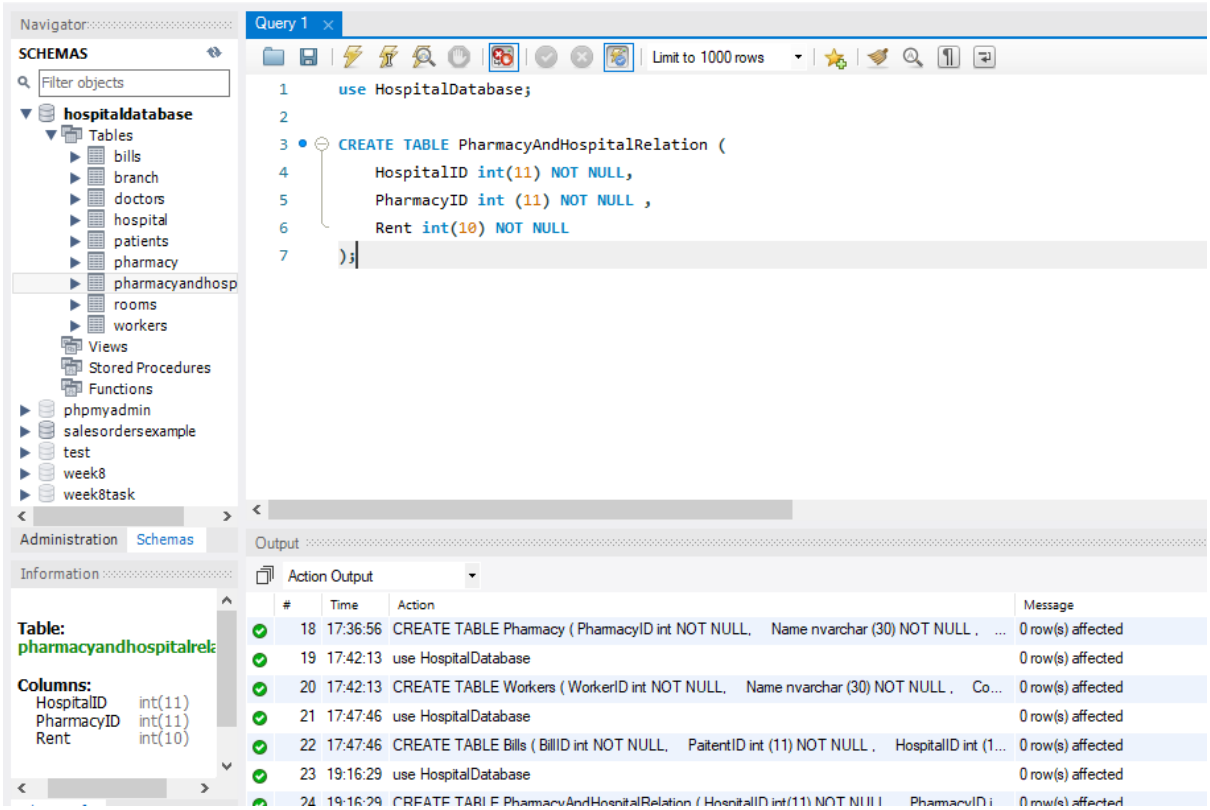
```
CREATE TABLE Bills (
        BillID int NOT NULL,
   PaitentID int (11) NOT NULL ,
   HospitalID int (11) NOT NULL,
   RoomID int (11) NOT NULL,
   TotalCost nvarchar (30) NOT NULL,
   DateOfJoin Date NOT NULL,
   DateOfDischarge Date  NULL
);
```



9. **Creating Pharmacy and Hospital Relation Table**

Command:

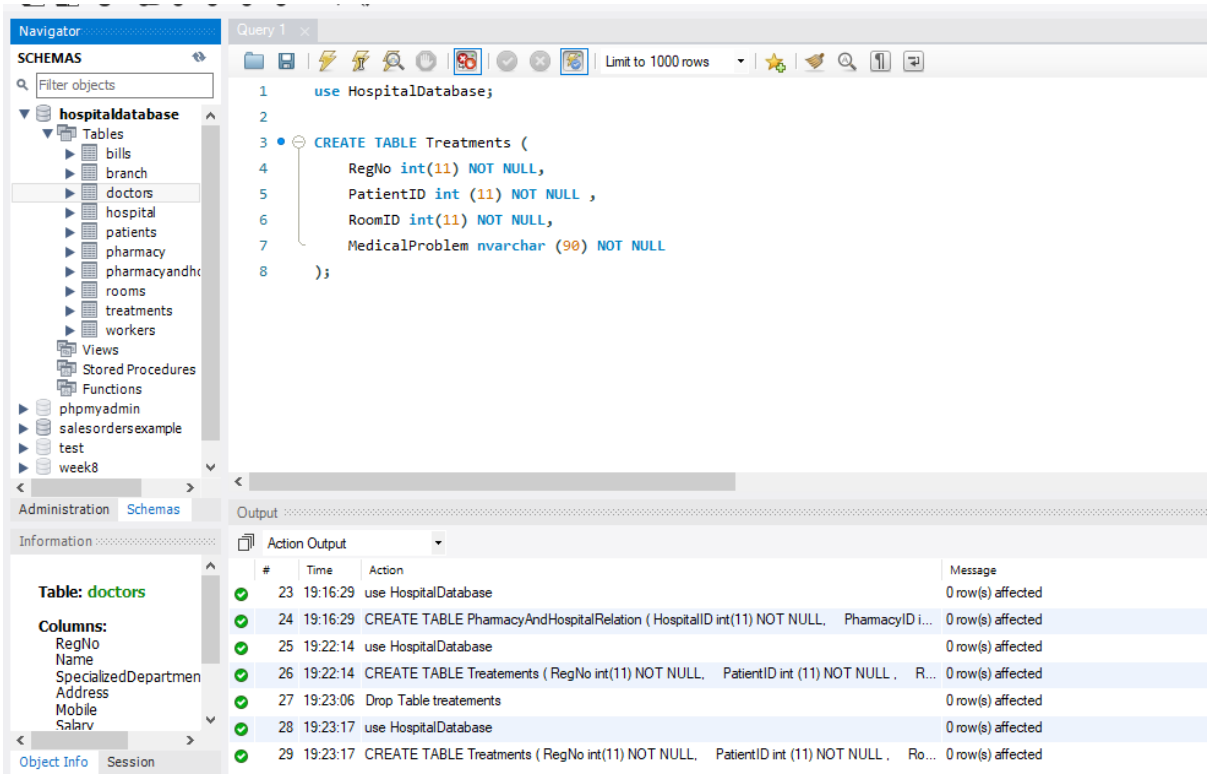
```
CREATE TABLE PharmacyAndHospitalRelation (
        HospitalID int(11) NOT NULL,
   PharmacyID int (11) NOT NULL ,
   Rent int(10) NOT NULL
);
```



10. Creating Treatments Table

Command:

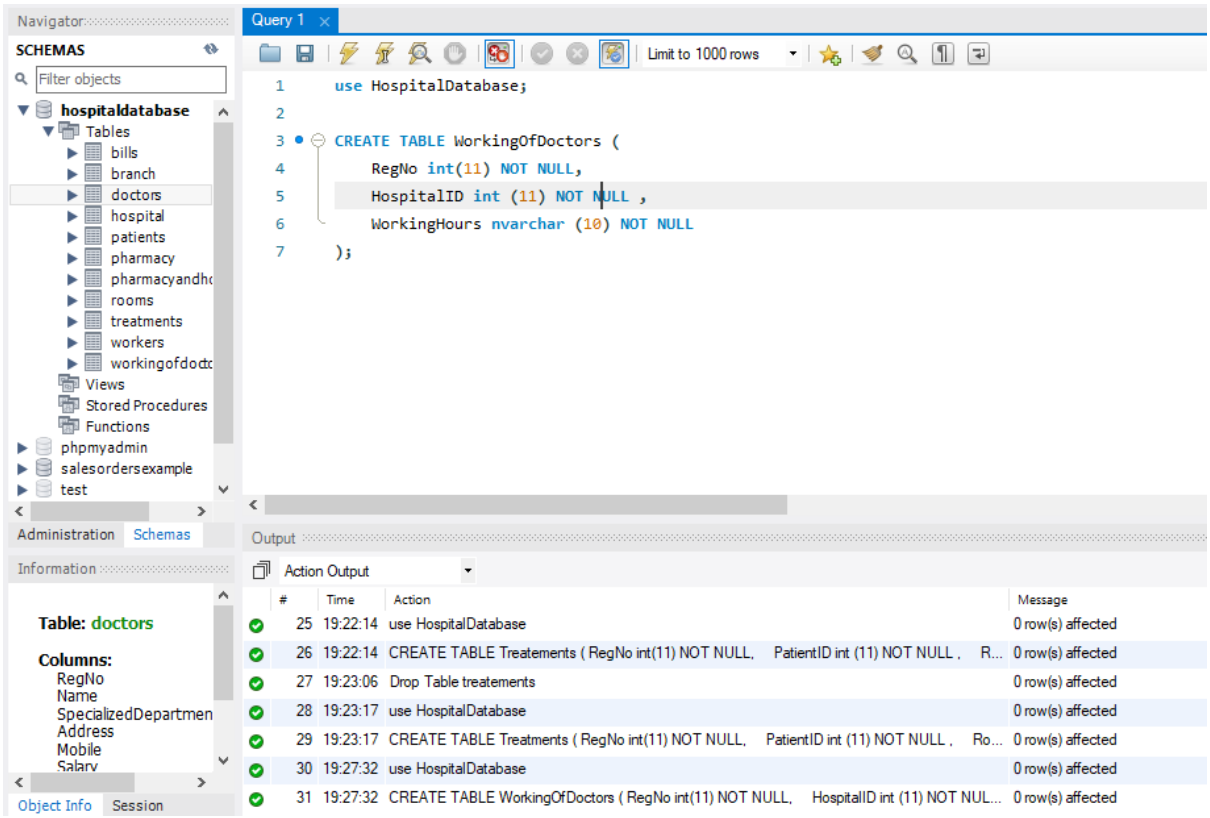
```
CREATE TABLE Treatments (
        RegNo int(11) NOT NULL,
   PatientID int (11) NOT NULL ,
   RoomID int(11) NOT NULL,
HospitalID int(11) NOT NULL,
   MedicalProblem nvarchar (90) NOT NULL
);
```



11. Create WorkingOfDoctors Table

Command:

```
CREATE TABLE WorkingOfDoctors (
        RegNo int(11) NOT NULL,
   HospitalID int (11) NOT NULL ,
   WorkingHours nvarchar (10) NOT NULL
);
```



Primary and Foreign Key Constraints:

Adding Primary Keys:

- **ALTER TABLE Branch ADD**

  CONSTRAINT Branch_PK PRIMARY KEY

  (

  BranchID

) ;

- **ALTER TABLE Doctors ADD**

  CONSTRAINT Doctors_PK PRIMARY KEY

  (

  RegNo

) ;

- **ALTER TABLE Patients ADD**
  CONSTRAINT Patients_PK PRIMARY KEY
  (
  PatientID
) ;

- **ALTER TABLE Pharmacy ADD**
  CONSTRAINT Pharmacy_PK PRIMARY KEY
  (
  PharmacyID
) ;

- **ALTER TABLE Rooms ADD**
  CONSTRAINT Rooms_PK PRIMARY KEY
  (
  RoomID
) ;

- **ALTER TABLE Workers ADD**
  CONSTRAINT Workers_PK PRIMARY KEY
  (
  WorkerID
) ;

- **ALTER TABLE Bills ADD**

```
CONSTRAINT Bills_PK PRIMARY KEY
(
        BillID
) ;
```

- ```
  ALTER TABLE Hospital ADD
  CONSTRAINT Hospital_PK PRIMARY KEY
  (
          HospitalID
  ) ;
  ```

Adding Foreign Keys:

- ```
  ALTER TABLE Hospital ADD
          CONSTRAINT Hospital_FK01 FOREIGN KEY
          (
                  BranchID
          ) REFERENCES Branch (
                  BranchID
          );
  ```

- ```
  ALTER TABLE pharmacyandhospitalrelation ADD
          CONSTRAINT PharmacyAndHospital_FK01 FOREIGN KEY
          (
                  HospitalID
          ) REFERENCES Hospital (
                  HospitalID
          ),
    ADD CONSTRAINT PharmacyAndHospital_FK02 FOREIGN KEY
          (
                  PharmacyID
          ) REFERENCES Pharmacy (
  ```
  ```

```
        PharmacyID
    );
```

- ```
  ALTER TABLE Treatments ADD
      CONSTRAINT Treatments_FK01 FOREIGN KEY
      (
          HospitalID
      ) REFERENCES Hospital (
          HospitalID
      ),
    ADD CONSTRAINT Treatments_FK02 FOREIGN KEY
      (
          PatientID
      ) REFERENCES Patients (
          PatientID
      ),
     ADD CONSTRAINT Treatments_FK03 FOREIGN KEY
      (
          RegNo
      ) REFERENCES Doctors (
          RegNo
      ),
    ADD CONSTRAINT Treatments_FK04 FOREIGN KEY
      (
          RoomID
      ) REFERENCES Rooms (
          RoomID
    );
  ```

- ```
  ALTER TABLE Bills ADD
      CONSTRAINT Bills_FK01 FOREIGN KEY
      (
          HospitalID
      ) REFERENCES Hospital (
          HospitalID
  ```

```
        ),
    ADD CONSTRAINT Bills_FK02 FOREIGN KEY
        (
            PaitentID
        ) REFERENCES Patients (
            PatientID
        ),
    ADD CONSTRAINT Bills_FK03 FOREIGN KEY
        (
            RoomID
        ) REFERENCES Rooms (
            RoomID
    );
```

Adding Indexes:

- CREATE INDEX BranchID ON Branch(BranchID);
- CREATE INDEX RegNo ON Doctors(RegNo);
- CREATE INDEX HospitalID ON Hospital(HospitalID);
- CREATE INDEX PatientID ON Patients(PatientID);
- CREATE INDEX PharmacyID ON Pharmacy(PharmacyID);
- CREATE INDEX PharmacyAndHospital ON pharmacyandhospitalrelation(PharmacyID, HospitalID);
- CREATE INDEX RoomsID ON Rooms(RoomID);
- CREATE INDEX TreatmentsIndex ON Treatments(RegNo, PatientID, RoomID, HospitalID);
- CREATE INDEX WorkingDoctorIndex ON workingofdoctors(RegNo, HospitalID);
- CREATE INDEX WorkersIndex ON workers(WorkerID);

Adding Data in the Database:

➢

Result Grid | Filter Rows: | Export: | Wrap Cell

RegNo	HospitalID	WorkingHours
10	101	5
14	101	5
11	101	4
12	101	6
13	101	6
15	101	6
18	101	5
21	101	4
20	101	5
14	101	5
18	101	6
17	101	5
23	101	6
29	101	6
25	101	5
39	101	5
24	101	5
19	101	5
22	101	5
31	101	5

workingofdoctors 81

Output

➢

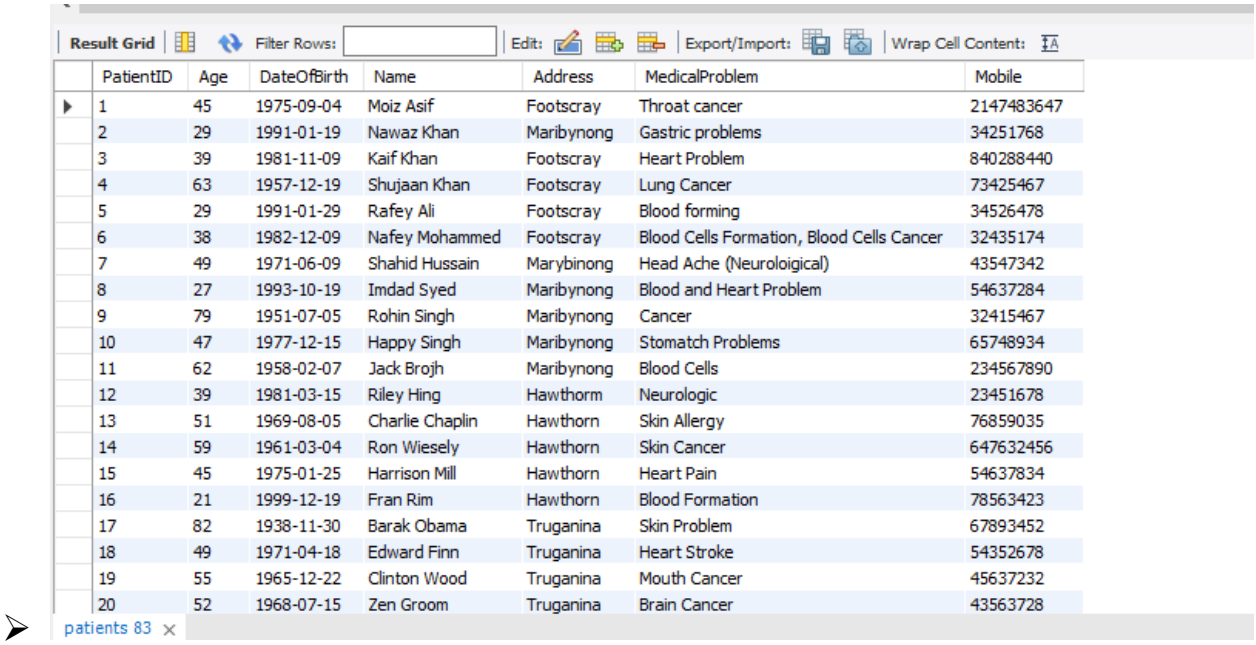
➢

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

RegNo	PatientID	RoomID	HospitalID	MedicalProblem
10	1	1	101	Throat Cancer
14	2	2	101	Gastric Problem
11	3	3	101	Heart problem
12	4	4	101	Lung Cancer
13	5	5	101	Blood Forming
13	6	12	101	Blood Cells Formation, Blood cancer
15	7	2	101	Head Ache (Neurological)
18	8	6	101	Blood and Heart Problem
21	8	6	101	Blood and Heart Problem
20	9	13	101	Cancer
14	10	8	101	Stomatch Problem
18	11	1	101	Blood Cells Formation
15	12	3	101	Brain (Neurologic)
17	13	4	101	Skin Allergy
20	14	5	101	Skin Cancer
21	15	6	101	Heart Pain
23	16	7	101	Blood formation
29	17	7	101	Skin Problem
26	18	8	101	Heart Stroke
25	19	9	101	Mouth Cancer

treatments 82

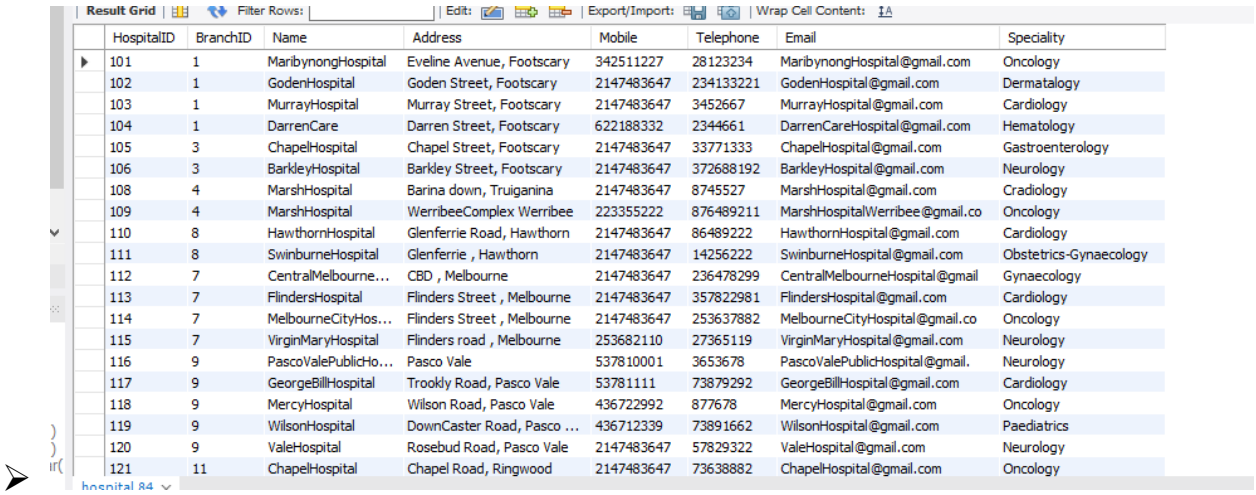
➢



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

PatientID	Age	DateOfBirth	Name	Address	MedicalProblem	Mobile
1	45	1975-09-04	Moiz Asif	Footscray	Throat cancer	2147483647
2	29	1991-01-19	Nawaz Khan	Maribynong	Gastric problems	34251768
3	39	1981-11-09	Kaif Khan	Footscray	Heart Problem	840288440
4	63	1957-12-19	Shujaan Khan	Footscray	Lung Cancer	73425467
5	29	1991-01-29	Rafey Ali	Footscray	Blood forming	34526478
6	38	1982-12-09	Nafey Mohammed	Footscray	Blood Cells Formation, Blood Cells Cancer	32435174
7	49	1971-06-09	Shahid Hussain	Marybinong	Head Ache (Neuroloigical)	43547342
8	27	1993-10-19	Imdad Syed	Maribynong	Blood and Heart Problem	54637284
9	79	1951-07-05	Rohin Singh	Maribynong	Cancer	32415467
10	47	1977-12-15	Happy Singh	Maribynong	Stomatch Problems	65748934
11	62	1958-02-07	Jack Brojh	Maribynong	Blood Cells	234567890
12	39	1981-03-15	Riley Hing	Hawthorm	Neurologic	23451678
13	51	1969-08-05	Charlie Chaplin	Hawthorn	Skin Allergy	76859035
14	59	1961-03-04	Ron Wiesely	Hawthorn	Skin Cancer	647632456
15	45	1975-01-25	Harrison Mill	Hawthorn	Heart Pain	54637834
16	21	1999-12-19	Fran Rim	Hawthorn	Blood Formation	78563423
17	82	1938-11-30	Barak Obama	Truganina	Skin Problem	67893452
18	49	1971-04-18	Edward Finn	Truganina	Heart Stroke	54352678
19	55	1965-12-22	Clinton Wood	Truganina	Mouth Cancer	45637232
20	52	1968-07-15	Zen Groom	Truganina	Brain Cancer	43563728

patients 83



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

HospitalID	BranchID	Name	Address	Mobile	Telephone	Email	Speciality
101	1	MaribynongHospital	Eveline Avenue, Footscary	342511227	28123234	MaribynongHospital@gmail.com	Oncology
102	1	GodenHospital	Goden Street, Footscary	2147483647	234133221	GodenHospital@gmail.com	Dermatalogy
103	1	MurrayHospital	Murray Street, Footscary	2147483647	3452667	MurrayHospital@gmail.com	Cardiology
104	1	DarrenCare	Darren Street, Footscary	622188332	2344661	DarrenCareHospital@gmail.com	Hematology
105	3	ChapelHospital	Chapel Street, Footscary	2147483647	33771333	ChapelHospital@gmail.com	Gastroenterology
106	3	BarkleyHospital	Barkley Street, Footscary	2147483647	372688192	BarkleyHospital@gmail.com	Neurology
108	4	MarshHospital	Barina down, Truiganina	2147483647	8745527	MarshHospital@gmail.com	Cradiology
109	4	MarshHospital	WerribeeComplex Werribee	223355222	876489211	MarshHospitalWerribee@gmail.co	Oncology
110	8	HawthornHospital	Glenferrie Road, Hawthorn	2147483647	86489222	HawthornHospital@gmail.com	Cardiology
111	8	SwinburneHospital	Glenferrie , Hawthorn	2147483647	14256222	SwinburneHospital@gmail.com	Obstetrics-Gynaecology
112	7	CentralMelbourne...	CBD , Melbourne	2147483647	236478299	CentralMelbourneHospital@gmail	Gynaecology
113	7	FlindersHospital	Flinders Street , Melbourne	2147483647	357822981	FlindersHospital@gmail.com	Cardiology
114	7	MelbourneCityHos...	Flinders Street , Melbourne	2147483647	253637882	MelbourneCityHospital@gmail.co	Oncology
115	7	VirginMaryHospital	Flinders road , Melbourne	253682110	27365119	VirginMaryHospital@gmail.com	Neurology
116	9	PascoValePublicHo...	Pasco Vale	537810001	3653678	PascoValePublicHospital@gmail.	Neurology
117	9	GeorgeBillHospital	Trookly Road, Pasco Vale	53781111	73879292	GeorgeBillHospital@gmail.com	Cardiology
118	9	MercyHospital	Wilson Road, Pasco Vale	436722992	877678	MercyHospital@gmail.com	Oncology
119	9	WilsonHospital	DownCaster Road, Pasco ...	436712339	73891662	WilsonHospital@gmail.com	Paediatrics
120	9	ValeHospital	Rosebud Road, Pasco Vale	2147483647	57829322	ValeHospital@gmail.com	Neurology
121	11	ChapelHospital	Chapel Road, Ringwood	2147483647	73638882	ChapelHospital@gmail.com	Oncology

hospital 84

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

RegNo	Name	SpecializedDepartment	Address	Mobile	Salary	Email
10	Syed Ali	Oncologist	Footscray	2147483647	90000	SyedAli@gmail.com
11	Syed Omair	Cardiologist	Footscray	2147483647	100000	SyedOmair@yahoo.com
12	Mohammed Tabrez	Oncologist	Maribynong	2147483647	980000	Tabrez117@gmail.com
13	Sai Ram	Hematalogist	Maribynong	2147483647	102000	Ram12456@gmail.com
14	Geordeo pool	Gastroenterologist	Footscray	2147483647	99000	Geordeo231@yahoo.com
15	Akib Khan	Neurologist	Footscray	964662277	110000	Khanakib@gmail.com
16	Irfan	Psychologist	Footscray	536628822	90000	Irfan33738@gmail.com
17	Fazal Mohammed	Dermatologist	Truganina	2147483647	106000	FazalMoammed@gmail.com
18	Shadaan Afreen	Hematologist	Truganina	2147483647	110000	Shadaan0028@gmail.com
19	Mifzal Hamid	Neurologist	Williams Landing	467388222	120000	Mifzalhamid456@gmail.com
20	Mudassir Syed	Oncoloist	Hawthorn	2147483647	116000	MudassirSyed7882@gmail...
21	Fuzail Taha	Cardiologist	Hawthorn	353672272	109000	Taha2273@gmail.com
22	Abdulla parvez	Neurologist	Williamslanding	36473837	104000	Abdulla12@yahoo.com
23	Mohammed Rahmat	Hematologist	Truganina	36782263	102000	Rahmat375854@yahoo.com
24	Zeeshan Fasi	Dermatologist	Williamslanding	364733321	120000	Fasi4553@yahoo.com
25	Mahmood Hussain	Oncologist	Werribee	474738322	110000	Mahmood45753@gmail.com
26	Shiley Rick	Cardiologist	Werribee	452772333	104000	Shiley3562@yahoo.com
27	Duen Pham	Gynacologist	Hawthorn	46732366	1230000	Pham456@gmail.com
28	Zaa Thang	Gastrologist	Truganina	2147483647	1090000	Thangzaa@gmail.com
29	Riley Shan	Hematologist	Hawthron	556228711	1000000	Riley2345@gmail.com

doctors 85

➤

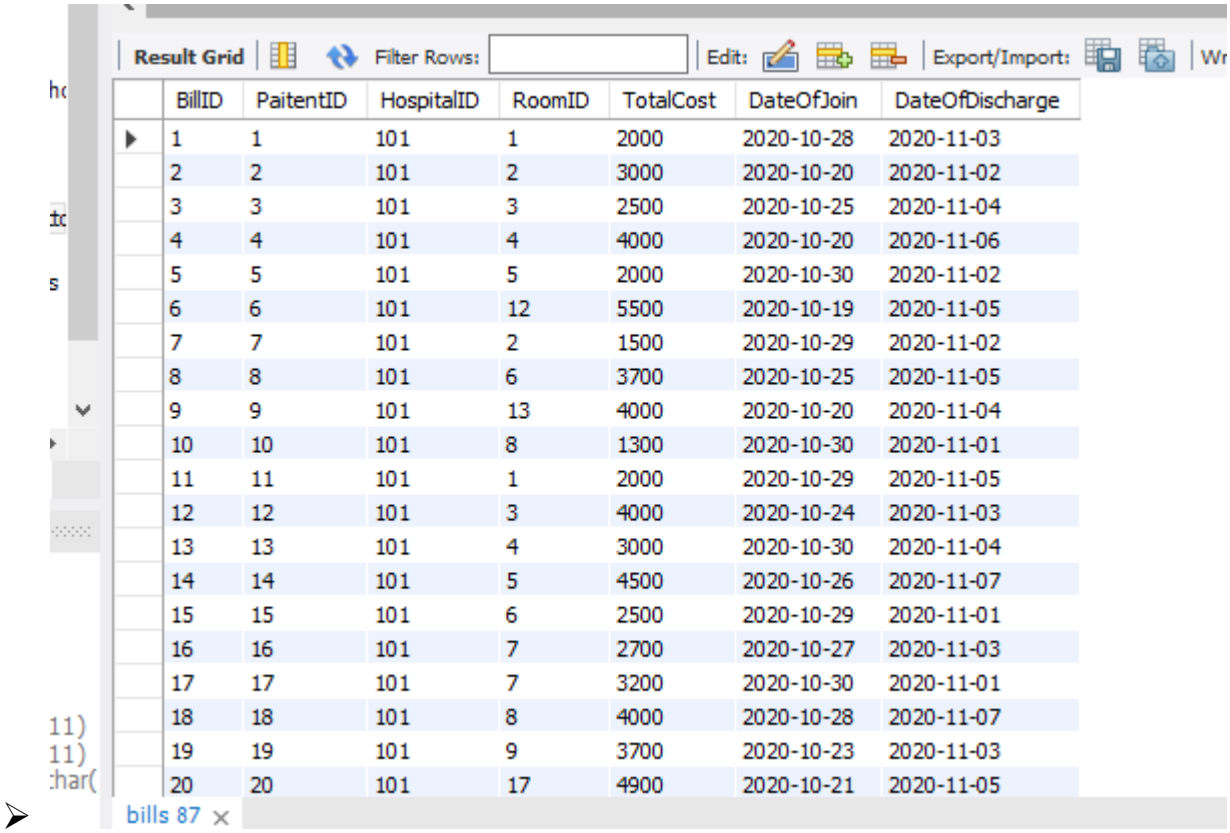
patients
pharmacy
pharmacyandh
rooms
treatments
workers
workingofdoct
ws
red Procedures
ctions
admin
rdersexample
Schemas
ofdoctors
int(11)
talID int(11)
gHours varchar(

Result Grid | Filter Rows: | Edit: | Export/Import:

BranchID	Address	Contact	Email
1	Footscray	123456789	footscrayhospital@gmail.com
2	adelide	2147483647	Adelidehospital@gmail.com
3	Footscray	342516780	Marybinonghospital@gmail.com
4	truganina	2147483647	TruganinaHospital@gmail.com
5	tarniet	2147483647	TarnietHospital@gmail.com
6	werribee	234156278	WerribeMercyHospital@gmail.com
7	MelbourneCentral	2147483647	MelbourneCentralHospital@gmail
8	Hawthorn	2147483647	HawthornHospital@gmail.com
9	Pascovale	2147483647	PascovaleHospital@gmail.com
10	Altona Medows	738913311	AltonaHospital@gmail.com
11	Ringwood	2147483647	RingwoodHospital@gmail.com
12	Richmond	2147483647	RichmondHospital@gmail.com
13	Pakenham	2147483647	PakenhamHospital@gmail.com
14	Coburg	781263451	CoburgHospital@gmail.com
15	Flinders	354611220	FlindersHospital@gmail.com
16	BroadMedows	2147483647	BroadMedowsHospital@gmail.com
17	Derrimut	2147483647	FDerrimutHospital@gmail.com
18	Sunshine	2147483647	SunshineHospital@gmail.com
19	Sunbury	827783391	SunburyHospital@gmail.com
20	Rafelo	2147483647	RafeloHospital@gmail.com

branch 86

➤



Result Grid | Filter Rows: | Edit: | Export/Import:

BillID	PaitentID	HospitalID	RoomID	TotalCost	DateOfJoin	DateOfDischarge
1	1	101	1	2000	2020-10-28	2020-11-03
2	2	101	2	3000	2020-10-20	2020-11-02
3	3	101	3	2500	2020-10-25	2020-11-04
4	4	101	4	4000	2020-10-20	2020-11-06
5	5	101	5	2000	2020-10-30	2020-11-02
6	6	101	12	5500	2020-10-19	2020-11-05
7	7	101	2	1500	2020-10-29	2020-11-02
8	8	101	6	3700	2020-10-25	2020-11-05
9	9	101	13	4000	2020-10-20	2020-11-04
10	10	101	8	1300	2020-10-30	2020-11-01
11	11	101	1	2000	2020-10-29	2020-11-05
12	12	101	3	4000	2020-10-24	2020-11-03
13	13	101	4	3000	2020-10-30	2020-11-04
14	14	101	5	4500	2020-10-26	2020-11-07
15	15	101	6	2500	2020-10-29	2020-11-01
16	16	101	7	2700	2020-10-27	2020-11-03
17	17	101	7	3200	2020-10-30	2020-11-01
18	18	101	8	4000	2020-10-28	2020-11-07
19	19	101	9	3700	2020-10-23	2020-11-03
20	20	101	17	4900	2020-10-21	2020-11-05

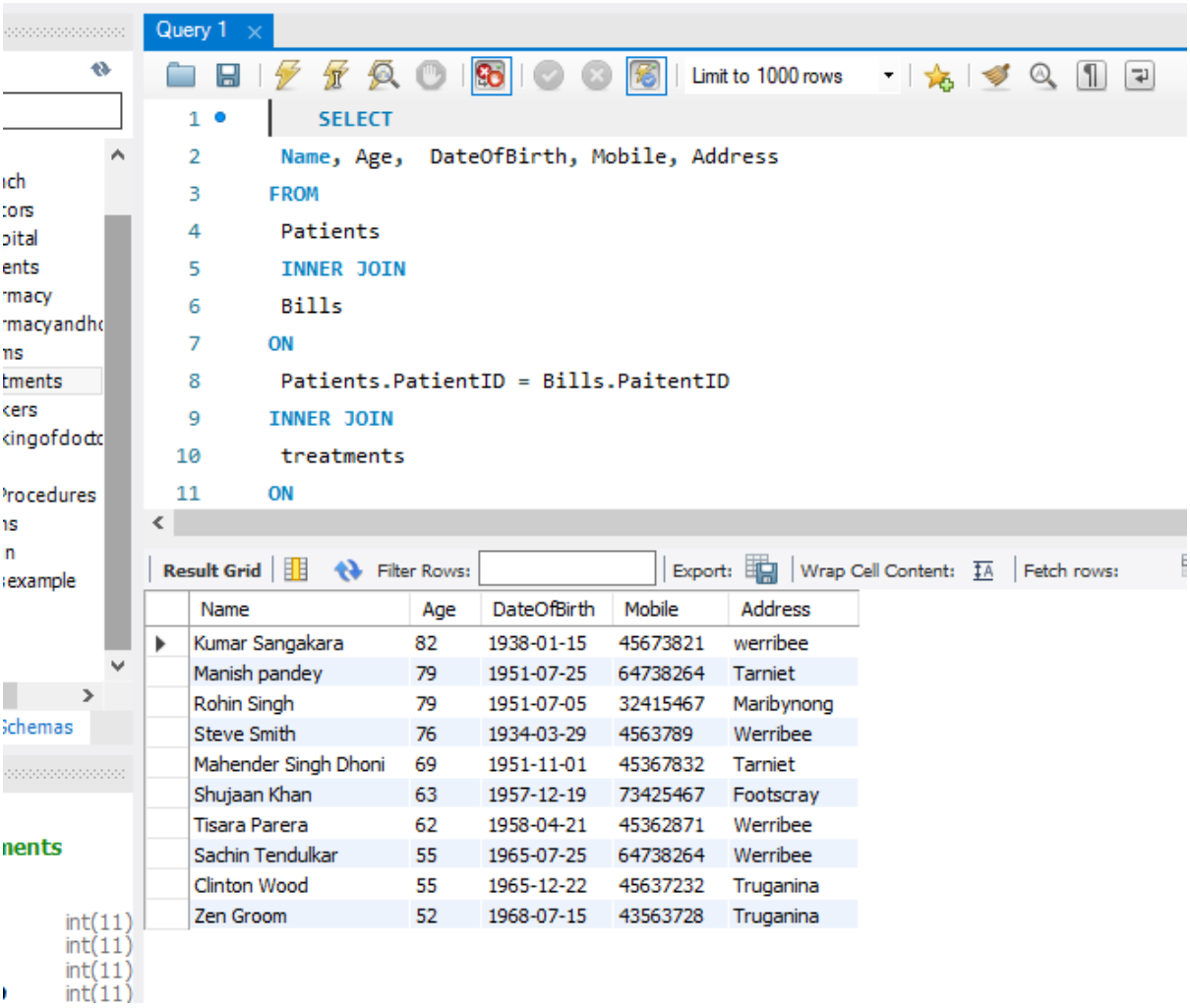
bills 87

Commands to show typical use cases of the database:

1. **Shows the patients details according to their age in descending order**

```
SELECT
 Name, Age,  DateOfBirth, Mobile, Address
FROM
 Patients
 INNER JOIN
 Bills
ON
 Patients.PatientID = Bills.PaitentID
INNER JOIN
 treatments
ON
 treatments.RoomID = Bills.RoomID
GROUP BY
 treatments.RoomID
 ORDER BY
 Age
```

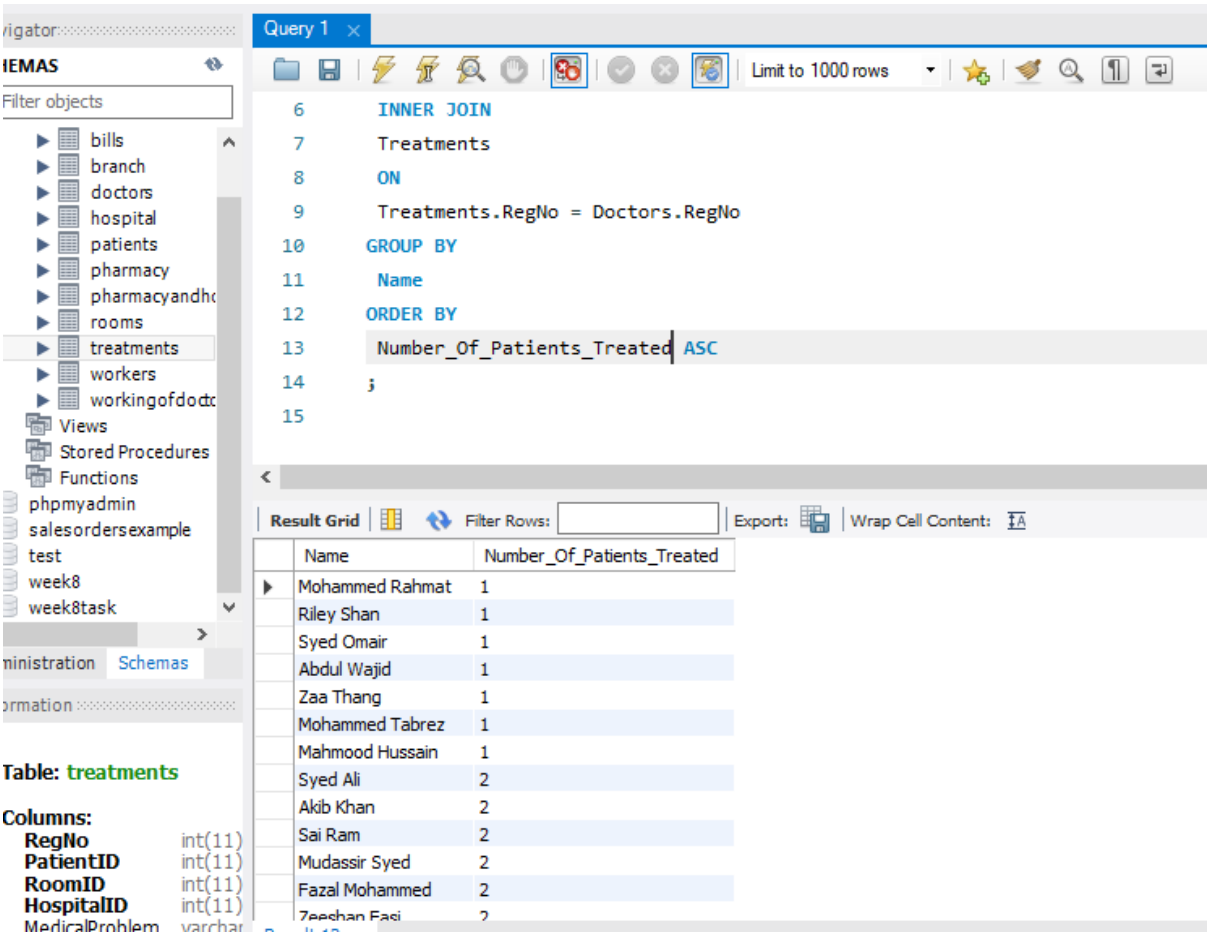
```
DESC
LIMIT 10
```



2. **Shows the number of patients treated by any doctor**

```
SELECT
 Name,
count(PatientID) AS Number_Of_Patients_Treated
FROM
 Doctors
INNER JOIN
 Treatments
ON
Treatments.RegNo = Doctors.RegNo
GROUP BY
 Name
```

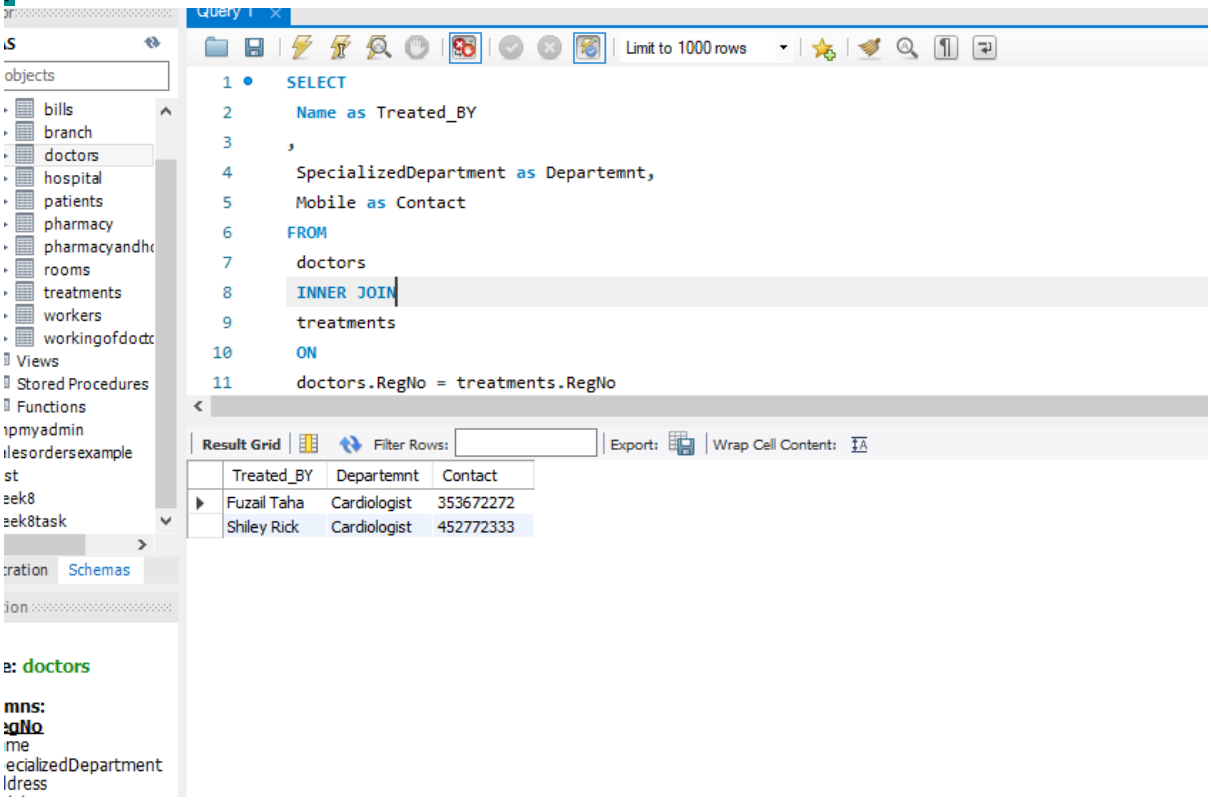
```
ORDER BY
Number_Of_Patients_Treated ASC
;
```



3. **Shows the details of doctors who treated the patients with medical problem “heart Pain”**

```
SELECT
Name as Treated_BY
,
SpecializedDepartment as Departemnt,
Mobile as Contact
FROM
doctors
INNER JOIN
treatments
ON
```

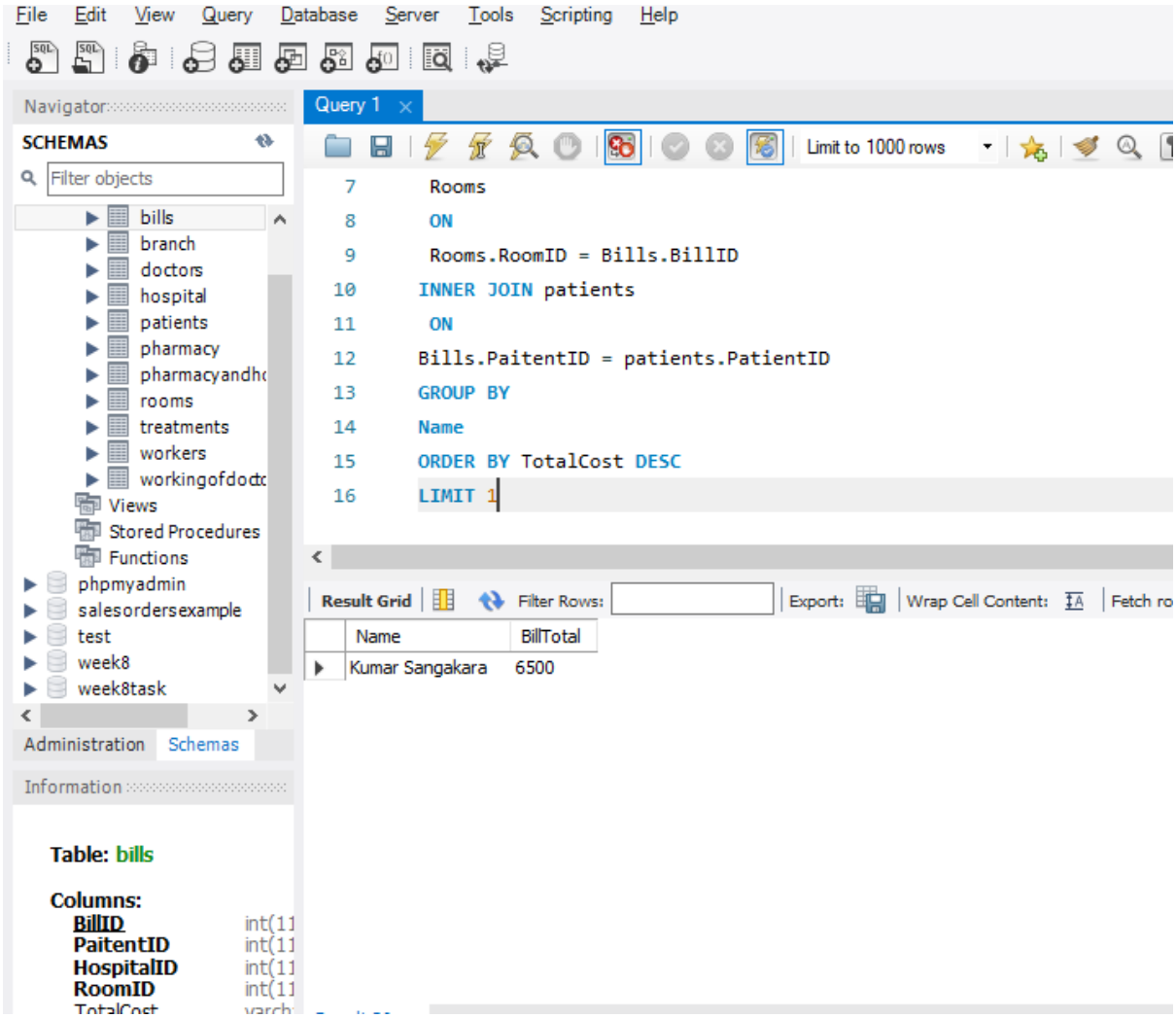
```
doctors.RegNo = treatments.RegNo
WHERE
treatments.MedicalProblem = 'Heart Pain'
;
```



4. **Shows the Maximum Cost of Bill Paid by the Patient**

```
SELECT
Name ,
TotalCost as BillTotal
FROM
Bills
INNER JOIN
Rooms
ON
Rooms.RoomID = Bills.BillID
INNER JOIN patients
ON
Bills.PaitentID = patients.PatientID
GROUP BY
Name
ORDER BY TotalCost DESC
```

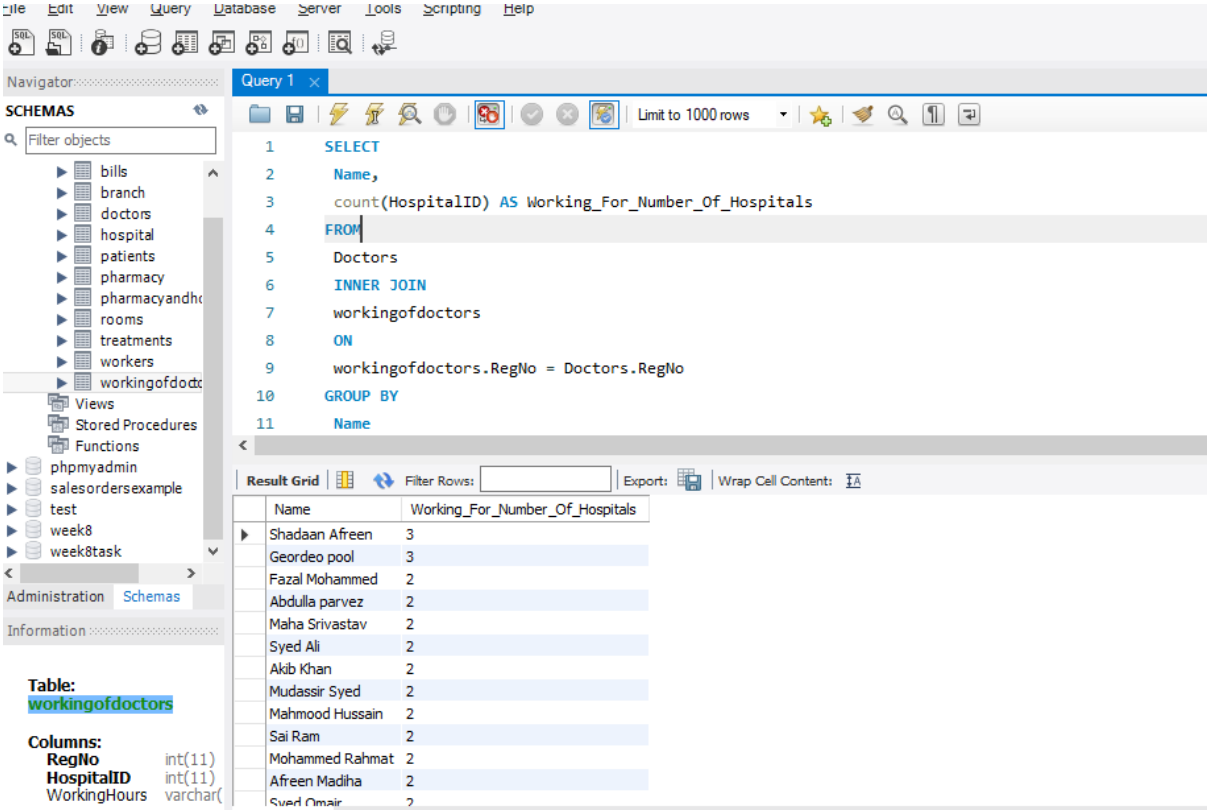
```
LIMIT 1
```



5. **Count of hospitals which a doctor works for**

```
SELECT
 Name,
count(HospitalID) AS Working_For_Number_Of_Hospitals
FROM
 Doctors
INNER JOIN
 workingofdoctors
ON
 workingofdoctors.RegNo = Doctors.RegNo
GROUP BY
 Name
```

```
ORDER BY
Working_For_Number_Of_Hospitals DESC
;
```



Main Usage of Database:

This database can be used as a hospital database to store data.In this we can store the details of different branches of hospitals, details of he doctors who work for different branches and hospitals. Also, details of patients, records of the operations and treatments, records of the workers and their shifts. You can easily see the pays and salaries of the workers and doctors. Easily update or change the details of bills if required.