

Rehabilitation Center Management Database

Now-a-days the percentage of people going into depression is increasing and hence they tend to smoke, drink and get drug addicted. The purpose of this project is that it manages the whole process of rehabilitation center system, so we can get the record and count of people who are undergoing the treatment and the number of people who got benefited with the rehabilitation process, so that we can take steps accordingly. This project includes patients which are addicted to drugs/alcohol/smoking etc. and have come to the rehabilitation center for the treatment. This system will help us to analyze the effectiveness of the treatment on every patient recorded in the system.

The database would be storing the details of the patients, the doctors, the appointment schedules, the list of addictions, locations, the medicine list, the level of severity, the treatment procedure, the diagnosis history, the result.

The techniques which will be used in this project are joins, views, procedures, triggers and user defined functions etc.

The list of tables which will be included in this project are:

Patient	PatientHistory	Location	Receptionist
Appointment	Room	Admit	Beds
Doctor	Therapist	Specialist	Treatment
Medicines	Diagnosis	DiagnosisHistory	Addiction
Recovery	Result		

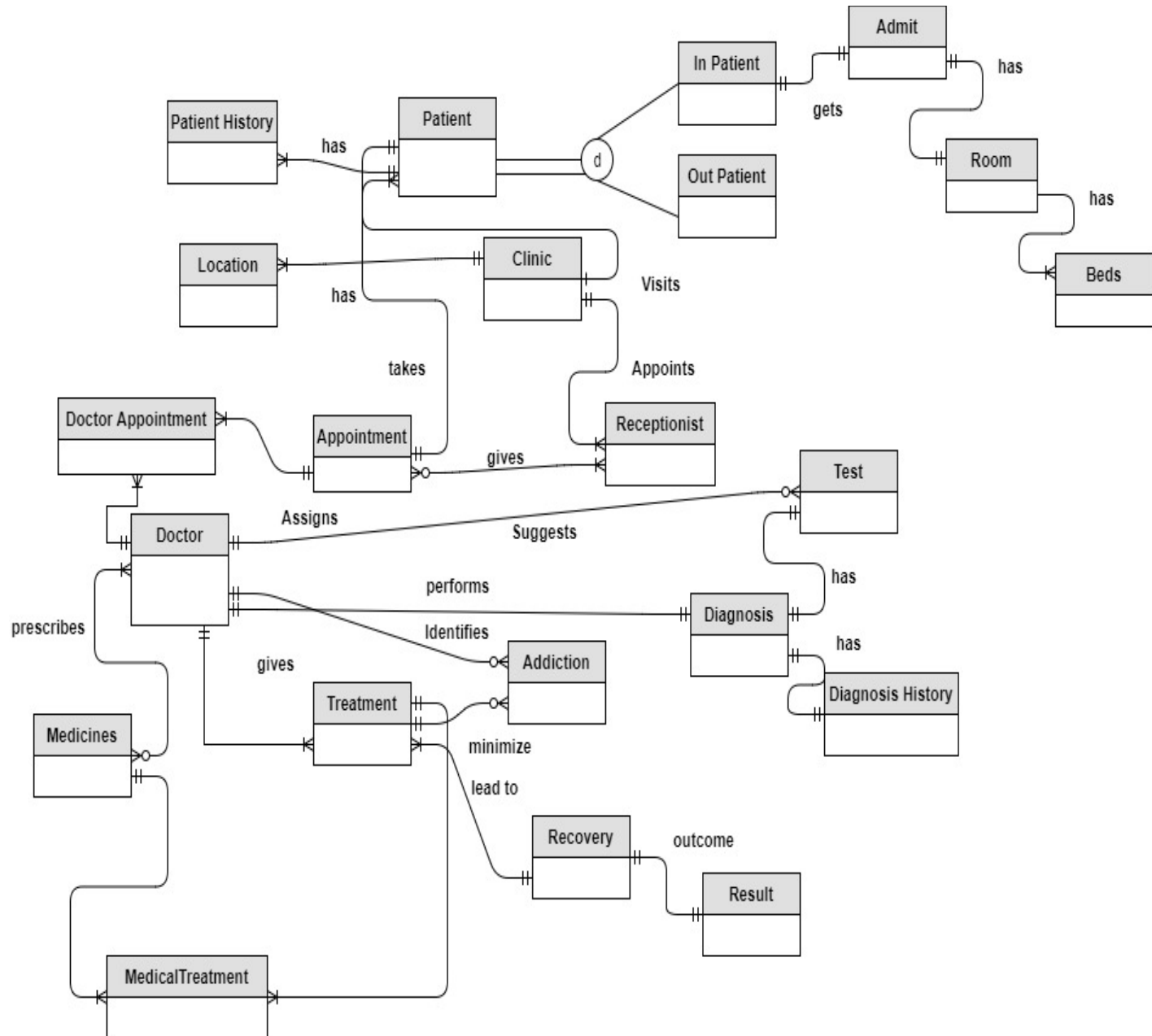
The main table in this project would be Treatment table as it will refer to most of the tables in the database and will have the attributes like TreatmentID, DateofTreatment, PatientID, DoctorID, AddictionID, DiagnosisID, AppointmentID, LocationID, ResultID, etc.

The Relationships between the tables would be as below:

- Mandatory one patient will take mandatory one appointment.
- Mandatory many patients will visit one clinic.
- Clinic has mandatory many locations.
- Clinic appoints mandatory many receptionist.
- Patient can be inpatient or outpatient.
- Mandatory one inpatient gets mandatory one admit.
- Mandatory one admit will have mandatory one room.
- Optional many rooms will have mandatory many beds.
- Mandatory many receptionist will give optional many appointments.
- Mandatory many appointments will have mandatory many doctors.
- Mandatory many doctors can prescribe optional many medicines.
- Mandatory many treatments will have mandatory many medicines.
- Mandatory one doctor can identify optional many addictions.
- Mandatory one doctor can have optional many tests.
- Mandatory one test can have mandatory one diagnosis.
- Mandatory one diagnosis will have mandatory one diagnosis history.
- Mandatory one doctor will perform mandatory one diagnosis.

- ➔ Mandatory one doctor will give multiple treatments.
- ➔ One doctor can prescribe optional many medicines.
- ➔ Mandatory many treatments can lead to one recovery.
- ➔ Mandatory one recovery will have mandatory one result.

Entity Relationship Diagram:





STORED PROCEDURES:

- 1) This procedure addClinic_sp inserts values in the clinic table as well as cliniclocation table as one clinic can have multiple locations. There is use of foreign key ClinicID in the cliniclocation table.

The screenshot displays the SQL Server Enterprise Manager interface. The main window shows the execution of a stored procedure named `addClinic_sp`. The procedure is defined as follows:

```
DELIMITER //
CREATE procedure addClinic_sp
(IN ClinicName varchar(255),
IN SAddress varchar(100),
IN Cty varchar(15),
IN Stte varchar(100),
IN County varchar(100))
BEGIN
START TRANSACTION;
INSERT INTO Clinic(Name)
VALUES (ClinicName);
SET @insertedID := LAST_INSERT_ID();
INSERT INTO cliniclocation(StreetAddress, City, State, Country, ClinicID)
VALUES(SAddress, Cty, Stte, County, @insertedID);
COMMIT;
//
call addClinic_sp('Redsox rehab center', '45 huntington ave', 'Colorado', 'WA', 'USA');
select * from cliniclocation;
```

The results pane shows the output of the procedure call, which is a table with 6 columns: ID, StreetAddress, City, State, Country, and ClinicID. The table contains two rows of data:

ID	StreetAddress	City	State	Country	ClinicID
9	43 Tree house Street	boston	MA	USA	12
14	45 huntington ave	Colorado	WA	USA	17

The bottom pane shows the Action Output, which includes the following messages:

#	Time	Action	Message	Duration / Fetch
554	03:17:46	call addClinic_sp('Redsox rehab center', '45 huntington ave', 'Colorado', 'WA', 'USA'...	0 row(s) affected	0.000 sec
555	03:17:46	call addClinic_sp('Redsox rehab center', '45 huntington ave', 'Colorado', 'WA', 'USA'...	10 row(s) returned	- / 0.000 sec

DELIMITER //

CREATE procedure addClinic_sp

(IN ClinicName varchar(255),

IN SAddress varchar(100),

IN Cty varchar(15),

IN Stte varchar(100),

IN County varchar(100))

BEGIN

START TRANSACTION;

INSERT INTO Clinic(Name)

VALUES (ClinicName);

SET @insertedID := LAST_INSERT_ID();

INSERT INTO cliniclocation(StreetAddress, City, State, Country, ClinicID)

```
VALUES(SAddress,Cty,Stte,County,@insertedID);
```

```
COMMIT;
```

```
END
```

```
//
```

```
call addClinic_sp('Redsox rehab center','45 huntington ave','Colorado','WA','USA');
```

```
select * from clinic;
```

```
select * from cliniclocation;
```

- 2) This procedure addDoctor_sp inserts values in the doctor table and in the addition table as the doctor will identify the addition. There is use of foreign key DoctorID in the addition table.

The screenshot displays the SQL Server Enterprise Manager interface. The top pane shows the SQL script for the 'addDoctor_sp' procedure. The script includes a 'DELIMITER //' statement, a 'CREATE procedure' block with parameters for DoctorName, DoctorAge, DoctorContact, AdditionName, and ConsultationFees, and a 'call addDoctor_sp' statement. The bottom pane shows the 'Result Grid' with two rows of data. The first row has ID 8, Name 'neelambari', Age 30, and Contact 8120982210. The second row has ID 9, Name 'neelambari', Age 30, and Contact 8120982210. The 'Output' pane at the bottom shows the execution of the 'select * from addition' and 'select * from doctor' queries, both returning 6 and 7 rows respectively.

```
1 DELIMITER //  
2 CREATE procedure addDoctor_sp  
3 (IN DoctorName varchar(255),  
4 IN DoctorAge varchar(10),  
5 IN DoctorContact varchar(15),  
6 IN AdditionName varchar(255),  
7 IN ConsultationFees int)  
8 BEGIN  
9 INSERT INTO Doctor(Name, Age, Contact)  
10 VALUES (DoctorName, DoctorAge, DoctorContact);  
11 set @doctorID:= last_insert_id();  
12 INSERT INTO addition(name, doctorID, Consultationfees)  
13 VALUES (AdditionName, @doctorID, ConsultationFees);  
14 END  
15 //  
16 call addDoctor_sp('neelambari', 30, '8120982210', 'Smoking', 500);  
17  
18  
19
```

ID	Name	Age	Contact
8	neelambari	30	8120982210
9	neelambari	30	8120982210

#	Time	Action	Message
563	03:36:13	select * from addition LIMIT 0, 1000	6 row(s) returned
564	03:39:58	select * from doctor LIMIT 0, 1000	7 row(s) returned

```
DELIMITER //
```

```
CREATE procedure addDoctor_sp
```

```
(IN DoctorName varchar(255),
```

```
IN DoctorAge varchar(10),
```

```
IN DoctorContact varchar(15),
IN AddictionName varchar(255),
IN ConsultationFees int)
BEGIN
INSERT INTO Doctor(Name, Age, Contact)
            VALUES (DoctorName, DoctorAge, DoctorContact);
set @doctorID:= last_insert_id();
INSERT INTO addiction(name, doctorID, Consultationfees)
            VALUES (AddictionName, @doctorID, ConsultationFees);
END
//
call addDoctor_sp('neelambari', 30, '8120982210', 'Smoking', 500);
select * from doctor;
select * from addiction;
```

- 3) The procedure addMedicine_sp inserts the medicine in the medicine table based on the data passed to the procedure.

The screenshot displays a SQL IDE interface. The top pane shows a script with the following SQL code:

```
1 DELIMITER //  
2 CREATE procedure addMedicine_sp  
3 (IN Name varchar(255),  
4 IN ExpiryDate date,  
5 IN Cost int)  
6 BEGIN  
7 INSERT INTO Medicine(Name,ExpiryDate,Cost)  
8 VALUES (Name,  
9 ExpiryDate,  
10 Cost);  
11 END  
12 //  
13  
14 call addMedicine_sp('BGS','2020/04/05',700);  
15  
16 select * from Medicine;
```

The bottom pane shows the 'Result Grid' with the following data:

ID	Name	ExpiryDate	Cost
1	Naltrexone	0000-00-00	200
2	Aspirin and Codeine	2018-05-02	200
3	Benzonatate	0000-00-00	500
4	Crocin	2017-07-12	890
5	Clomax	2019-05-22	600
6	BGS	2020-04-05	700

The 'Output' pane shows the execution of the procedure and the subsequent query:

#	Time	Action	Message
582	04:26:17	call addMedicine_sp('BGS','2020/04/05',700); select * from Medicine;	1 row(s) affected

DELIMITER //

CREATE procedure addMedicine_sp

(IN Name varchar(255),

IN ExpiryDate date,

IN Cost int)

BEGIN

INSERT INTO Medicine(Name,ExpiryDate,Cost)

VALUES (Name,ExpiryDate,Cost);

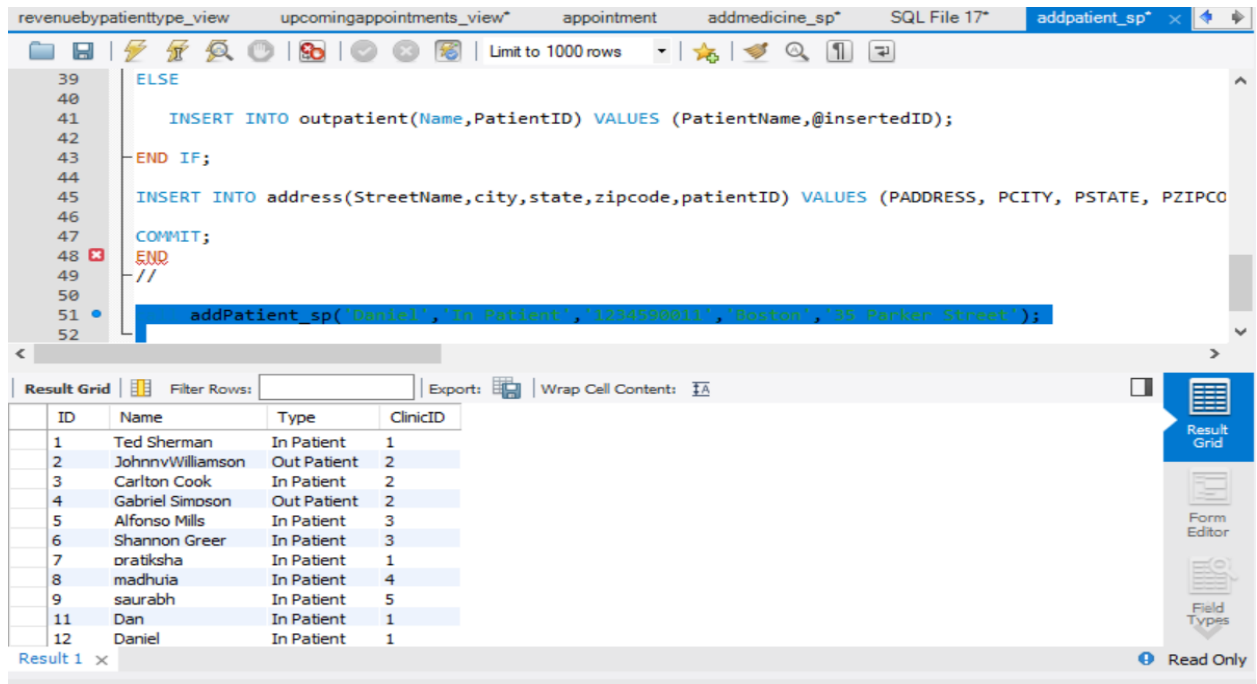
END

//

call addMedicine_sp('Crocin','2017/07/12',890);

select * from Medicine;

- 4) The procedure `addpatient_sp` inserts a new patient in patient table. Along with this, it also inserts the record in the In Patient or the Out Patient Table, based on the type mentioned in the query. Also, it creates the Patient address record while adding a new patient. There is use of foreign key `ClinicID` in the patient table, `PatientID` in the patientaddress, inpatient and outpatient table.



The screenshot shows the SQL Server Enterprise Manager interface. The top pane displays the SQL code for the `addpatient_sp` stored procedure. The bottom pane shows the execution results in a grid format.

```
39 ELSE
40
41     INSERT INTO outpatient(Name,PatientID) VALUES (PatientName,@insertedID);
42
43 END IF;
44
45 INSERT INTO address(StreetName,city,state,zipcode,patientID) VALUES (PADDRESS, PCITY, PSTATE, PZIPCODE);
46
47 COMMIT;
48 END
49 //
50
51 addPatient_sp('Daniel','In Patient','1234567891','Boston','35 Parker Street');
52
```

ID	Name	Type	ClinicID
1	Ted Sherman	In Patient	1
2	JohnnvWilliamson	Out Patient	2
3	Carlton Cook	In Patient	2
4	Gabriel Simson	Out Patient	2
5	Alfonso Mills	In Patient	3
6	Shannon Greer	In Patient	3
7	pratiksha	In Patient	1
8	madhuja	In Patient	4
9	saurabh	In Patient	5
11	Dan	In Patient	1
12	Daniel	In Patient	1

DELIMITER //

CREATE procedure addPatient_sp

(IN PatientName varchar(255),

IN PatientType varchar(10),

IN PCon varchar(15),

IN PCity varchar(100),

IN PAddress varchar(100),

IN PSTATE varchar(100),

IN PZIPCODE varchar(6))

BEGIN

START TRANSACTION;

SET @ClinicID := (select Clinic.ID from Clinic

LEFT JOIN ClinicLocation C ON C.ClinicID = Clinic.ID


```
WHERE C.City IN (PCity) AND C.StreetAddress LIKE (PAddress));

SET @insertedID := LAST_INSERT_ID();

IF PatientType = null THEN

    INSERT INTO Patient(Name,Type,ClinicID)

        VALUES (PatientName, 0, @ClinicID);

ELSE

    INSERT INTO Patient(Name,Type,ClinicID)

        VALUES (PatientName, PatientType, @ClinicID);

SET @patientID := LAST_INSERT_ID();

IF PatientType = 'In Patient' THEN

    INSERT INTO inpatient(Name,contact,patientID) VALUES (PatientName,PCon,@patientID);

ELSE

    INSERT INTO outpatient(Name,PatientID) VALUES (PatientName,@insertedID);

END IF;

INSERT INTO address(StreetName,city,state,zipcode,patientID) VALUES (PADDRESS, PCITY,
PSTATE, PZIPCODE, @patientID);

COMMIT;

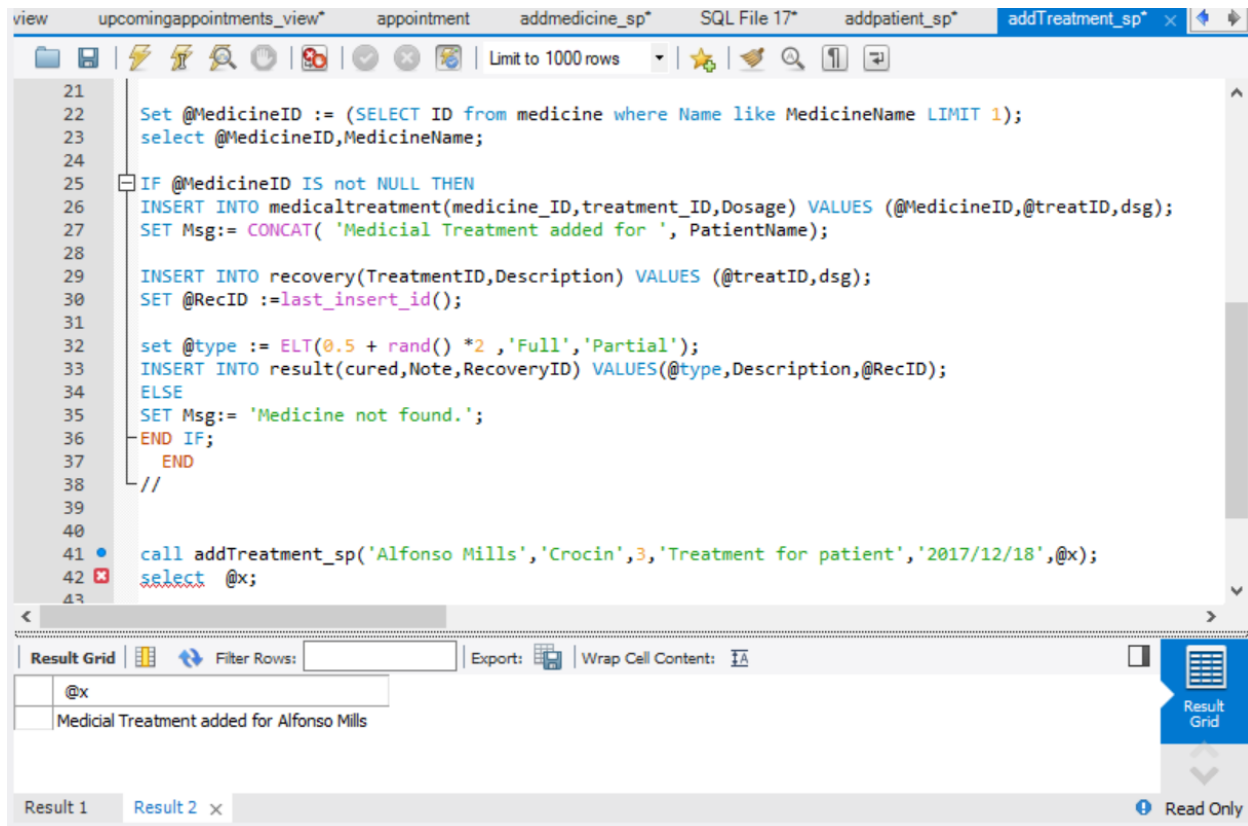
END

//

call addPatient_sp('Daniel','In Patient', '1234590011','Boston','35 Parker Street');

select * from Patient;
```

- 5) **The procedure addtreatment_sp inserts the treatment in the treatment table. Simultaneously, it adds the medicine prescribed in the medicinetratment table, inserts a record into the recovery table with the details of the treatment and the doctor. Also, it adds a record for this treatment in the result table and sets the message. There is the use of Foreign Key PatientID, DoctorID in treatment table, MedicineID in medicinetratment table, treatmentID in the recovery table, recoveryID in the result table.**



The screenshot shows a SQL Server Enterprise Manager window with a script editor and a results pane. The script editor contains the following T-SQL code:

```
21
22 Set @MedicineID := (SELECT ID from medicine where Name like MedicineName LIMIT 1);
23 select @MedicineID,MedicineName;
24
25 IF @MedicineID IS not NULL THEN
26     INSERT INTO medicaltreatment(medicine_ID,treatment_ID,Dosage) VALUES (@MedicineID,@treatID,dsg);
27     SET Msg:= CONCAT( 'Medicial Treatment added for ', PatientName);
28
29     INSERT INTO recovery(TreatmentID,Description) VALUES (@treatID,dsg);
30     SET @RecID :=last_insert_id();
31
32     set @type := ELT(0.5 + rand() *2 , 'Full', 'Partial');
33     INSERT INTO result(cured,Note,RecoveryID) VALUES(@type,Description,@RecID);
34 ELSE
35     SET Msg:= 'Medicine not found.';
36 END IF;
37 END
38 //
39
40
41 call addTreatment_sp('Alfonso Mills','Crocin',3,'Treatment for patient','2017/12/18',@x);
42 select @x;
```

The results pane shows two rows of data:

Result 1	Result 2
@x	
	Medicial Treatment added for Alfonso Mills

DELIMITER //

CREATE procedure addTreatment_sp

(IN PatientName varchar(255),

IN MedicineName varchar(10),

IN dsg int,

IN Description varchar(255),

IN TDate Datetime,

OUT Msg varchar(200))

BEGIN

SET @PtID := (select ID from Patient where Name like PatientName LIMIT 1);

```
SELECT P.addiction_ID,AD.DoctorID into @AdID,@DocID from patientaddiction P
LEFT JOIN addiction AD ON AD.ID = P.addiction_ID where patient_ID =@PtID LIMIT 1;

INSERT INTO treatment (Date,DoctorID,AddictionID,Description,patient_ID)
VALUES(TDate,@docID,@AdID,dsg,@PtID);
SET @treatID := last_insert_id();

Set @MedicineID := (SELECT ID from medicine where Name like MedicineName LIMIT 1);
select @MedicineID,MedicineName;

IF @MedicineID IS not NULL THEN

INSERT INTO medicaltreatment(medicine_ID,treatment_ID,Dosage) VALUES
(@MedicineID,@treatID,dsg);

SET Msg:= CONCAT( 'Medicial Treatment added for ', PatientName);

INSERT INTO recovery(TreatmentID,Description) VALUES (@treatID,dsg);
SET @RecID :=last_insert_id();

set @type := ELT(0.5 + rand() *2 , 'Full','Partial');
INSERT INTO result(cured>Note,RecoveryID) VALUES(@type,Description,@RecID);
ELSE
SET Msg:= 'Medicine not found.';
END IF;

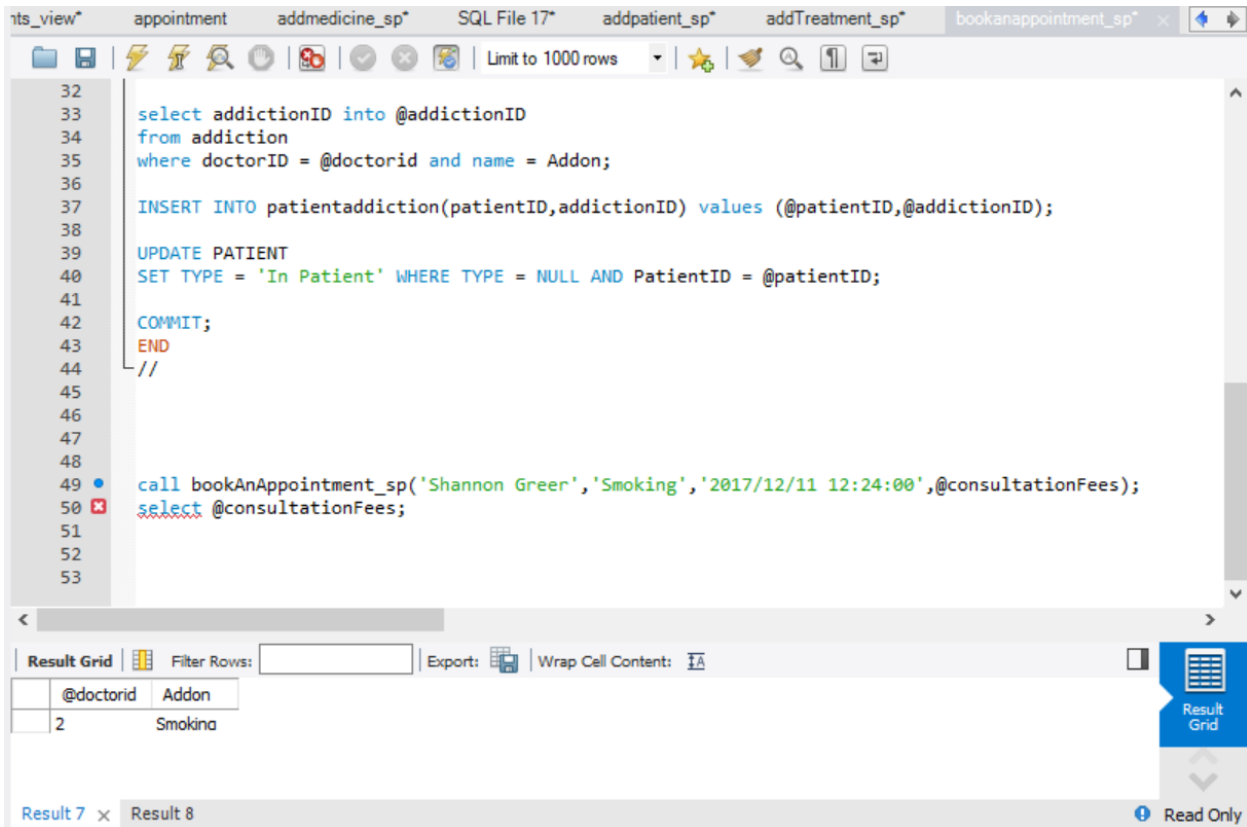
END

//

call addTreatment_sp('Alfonso Mills','Crocini',3,'Treatment for patient','2017/12/18',@x);
select @x;
```

- 6) The procedure **BookAnAppointment_sp** inserts an entry in the appointment table taking the patientname, addiction and appointment date as an input. It also adds a record in the

patientaddiction table. There is the use of Foreign Key PatientID, DoctorID, receptionistID in the appointment table.



The screenshot shows a SQL IDE with a script editor and a result grid. The script editor contains the following SQL code:

```
32
33 select addictionID into @addictionID
34 from addiction
35 where doctorID = @doctorid and name = Addon;
36
37 INSERT INTO patientaddiction(patientID,addictionID) values (@patientID,@addictionID);
38
39 UPDATE PATIENT
40 SET TYPE = 'In Patient' WHERE TYPE = NULL AND PatientID = @patientID;
41
42 COMMIT;
43 END
44 //
45
46
47
48
49 call bookAnAppointment_sp('Shannon Greer','Smoking','2017/12/11 12:24:00',@consultationFees);
50 select @consultationFees;
51
52
53
```

The result grid shows the following data:

@doctorid	Addon
2	Smoking

The result grid also shows the following data:

Result 7	Result 8

DELIMITER //

CREATE procedure bookAnAppointment_sp

(IN PatientName varchar(255),

IN Addon varchar(10),

IN AppointmentDate datetime,

OUT consultationFees INT)

BEGIN

START TRANSACTION;

SET @patientName := '%'+PatientName+'%';

select clinicID,ID into @clinicID,@patientID

```
from patient
where `Name` LIKE @patientName
LIMIT 1;
```

```
select doctorid into @doctorid
from addiction
where `name` LIKE Addon
order by rand()
LIMIT 1;
```

```
select ID into @receptionistID
from receptionist
where clinicID = @clinicID;
```

```
INSERT INTO appointment(`Date`,patientID,doctorID,receptionistID)
VALUES (AppointmentDate,@patientID,@doctorid,@receptionistID);
```

```
select @doctorid,Addon;
```

```
/*select consultationFees from addiction where DoctorID = @doctorid AND `Name` LIKE Addon order
by rand() limit 1;*/
```

```
set consultationFees := (select consultationFees from addiction where DoctorID = @doctorid AND
`Name` LIKE Addon order by rand() limit 1);
```

```
select addictionID into @addictionID
from addiction
where doctorID = @doctorid and name = Addon;
```

```
INSERT INTO patientaddiction(patientID,addictionID) values (@patientID,@addictionID);
```

```
UPDATE PATIENT
```

```
SET TYPE = 'In Patient' WHERE TYPE = NULL AND PatientID = @patientID;
```

COMMIT;

END

//

call bookAnAppointment_sp('Shannon Greer','Smoking','2017/12/11
12:24:00',@consultationFees);

select @consultationFees;

- 7) The procedure `diseaseDiagnosis_sp` inserts a record in the diagnosis table to store the diagnosis generated by the test carried out stored in the test table. There is the use of `PatientID`, `AddictionID`, `DoctorID` in the test table and the use of `DoctorID` and `TestID` in the Diagnosis Table.

The screenshot displays the SQL Server Enterprise Manager interface. The top pane shows the code for the stored procedure `diseasediagnosis_sp`. The code includes an `INSERT INTO Test` statement, a `Set @TestID` statement, an `INSERT INTO diagnosis` statement, and a `call diseaseDiagnosis_sp` statement. The bottom pane shows the results of the procedure, which is a table with 5 columns: ID, Date, Note, TestID, and DoctorID. The results show 12 rows of data.

```
21 INSERT INTO Test(Name,DoctorID,Date,Cost,Result)
22     VALUES
23     (
24         @DoctorID,
25         TestDate,
26         TestCost,
27         @type);
28 Set @TestID := last_insert_id();
29 END IF;
30 INSERT INTO diagnosis (Date,Note,TestID,DoctorID)
31     VALUES (TestDate,(select diseaseDiagnosis (@AddictionID,@type)),@TestID,@DoctorID);
32
33 COMMIT;
34 END
35 //
36 call diseaseDiagnosis_sp('Alfonso Mills','Blood','2017/11/12',4450);
37 select * from diagnosis;
```

ID	Date	Note	TestID	DoctorID
6	2017-10-12 09:50:00	The person has Negligible addiction of Alcohol	2	6
7	2017-10-12 09:00:00	The person has Negligible addiction of Alcohol	3	1
8	2017-10-12 10:00:00	The person has Negligible addiction of Smoking	4	2
9	2017-10-12 11:00:00	The person has Negligible addiction of Smoking	5	1
10	2017-10-12 14:00:00	The person has Negligible addiction of Smoking	6	4
11	2017-10-12 13:00:00	The person has Negligible addiction of Smoking	2	2
12	2017-11-12 12:00:00	The person has Marginal addiction of Smoking	8	5
22	2017-11-12 00:00:00	The person has Negligible addiction of Smoking	25	2

DELIMITER //

CREATE procedure diseaseDiagnosis_sp

(IN PatientName varchar(255),

IN TestName varchar(100),

IN TestDate datetime,

IN TestCost INT)

BEGIN

START TRANSACTION;

SET @PatientID := (select ID from patient

WHERE `name` LIKE PatientName LIMIT 1);

SET @AddictionID := (select addiction_ID from patientaddiction

WHERE patient_ID = @PatientID LIMIT 1);

```
SET @DoctorID := (select doctorID from addiction
                    WHERE ID = @AddictionID);

IF (TestName != '') THEN

set @type := ELT(0.5 + rand() *3 , 'Negligible', 'Marginal', 'Critical');

INSERT INTO Test(Name, DoctorID, Date, Cost, Result)
            VALUES (TestName, @DoctorID, TestDate, TestCost, @type);

Set @TestID := last_insert_id();

END IF;

INSERT INTO diagnosis (Date, Note, TestID, DoctorID)
            VALUES (TestDate, (select diseaseDiagnosis (@AddictionID, @type)), @TestID, @DoctorID);

COMMIT;

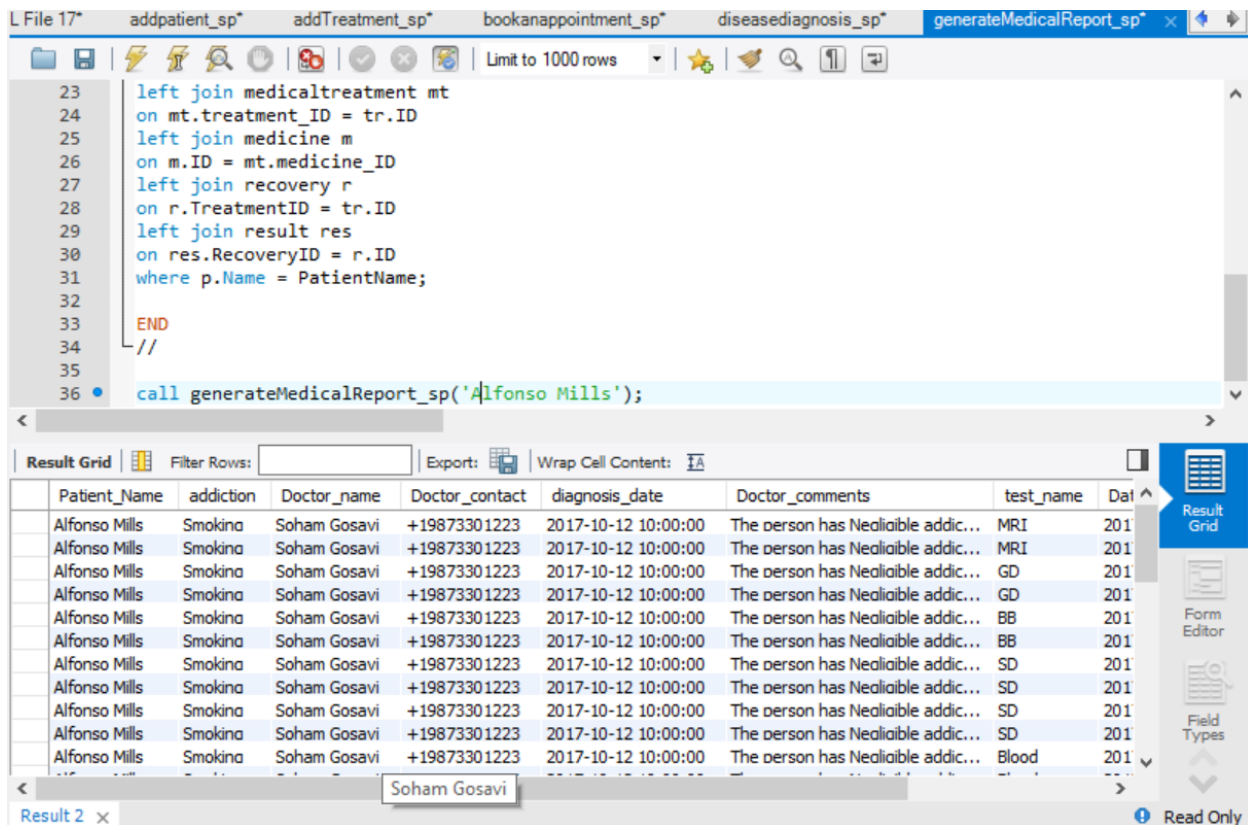
END

//

call diseaseDiagnosis_sp('Alfonso Mills', 'Blood', '2017/11/12', 4450);

select * from diagnosis;
```


- 8) The procedure generateMedicalReport_sp generates a patient wise report of all the Diagnosis done on the patient by the Doctor during the treatment. It uses joins on patientaddiction, addiction, doctor, diagnosis, test, treatment, medicaltreatment, medicine, recovery, result tables.



The screenshot shows a SQL IDE with a script editor and a result grid. The script editor contains the following SQL code:

```
23 left join medicaltreatment mt
24 on mt.treatment_ID = tr.ID
25 left join medicine m
26 on m.ID = mt.medicine_ID
27 left join recovery r
28 on r.TreatmentID = tr.ID
29 left join result res
30 on res.RecoveryID = r.ID
31 where p.Name = PatientName;
32
33 END
34 //
35
36 call generateMedicalReport_sp('Alfonso Mills');
```

The result grid displays the following data:

Patient_Name	addiction	Doctor_name	Doctor_contact	diagnosis_date	Doctor_comments	test_name	Date
Alfonso Mills	Smoking	Soham Gosavi	+19873301223	2017-10-12 10:00:00	The person has Neoliable addic...	MRI	2017-10-12 10:00:00
Alfonso Mills	Smoking	Soham Gosavi	+19873301223	2017-10-12 10:00:00	The person has Neoliable addic...	MRI	2017-10-12 10:00:00
Alfonso Mills	Smoking	Soham Gosavi	+19873301223	2017-10-12 10:00:00	The person has Neoliable addic...	GD	2017-10-12 10:00:00
Alfonso Mills	Smoking	Soham Gosavi	+19873301223	2017-10-12 10:00:00	The person has Neoliable addic...	GD	2017-10-12 10:00:00
Alfonso Mills	Smoking	Soham Gosavi	+19873301223	2017-10-12 10:00:00	The person has Neoliable addic...	BB	2017-10-12 10:00:00
Alfonso Mills	Smoking	Soham Gosavi	+19873301223	2017-10-12 10:00:00	The person has Neoliable addic...	BB	2017-10-12 10:00:00
Alfonso Mills	Smoking	Soham Gosavi	+19873301223	2017-10-12 10:00:00	The person has Neoliable addic...	SD	2017-10-12 10:00:00
Alfonso Mills	Smoking	Soham Gosavi	+19873301223	2017-10-12 10:00:00	The person has Neoliable addic...	SD	2017-10-12 10:00:00
Alfonso Mills	Smoking	Soham Gosavi	+19873301223	2017-10-12 10:00:00	The person has Neoliable addic...	SD	2017-10-12 10:00:00
Alfonso Mills	Smoking	Soham Gosavi	+19873301223	2017-10-12 10:00:00	The person has Neoliable addic...	SD	2017-10-12 10:00:00
Alfonso Mills	Smoking	Soham Gosavi	+19873301223	2017-10-12 10:00:00	The person has Neoliable addic...	Blood	2017-10-12 10:00:00

DELIMITER //

CREATE procedure generateMedicalReport_sp

(IN PatientName varchar(255))

BEGIN

select p.name as Patient_Name,a.name as addiction,doc.name as Doctor_name,doc.contact as
Doctor_contact,dia.date as diagnosis_date,dia.note as Doctor_comments,t.name as test_name,t.Date,t.cost
as test_cost

from patient p

left join patientaddiction pa

```
on p.id=pa.patient_ID
left join addiction a
on a.id = pa.addiction_ID
left join doctor doc
on doc.id = a.doctorid
left join diagnosis dia
on doc.id = dia.doctorid
left join test t
on t.DoctorID = dia.DoctorID
left join treatment tr
on tr.patient_ID = p.ID
AND tr.DoctorID = doc.ID
left join medicaltreatment mt
on mt.treatment_ID = tr.ID
left join medicine m
on m.ID = mt.medicine_ID
left join recovery r
on r.TreatmentID = tr.ID
left join result res
on res.RecoveryID = r.ID
where p.Name = PatientName;

END

//

call generateMedicalReport_sp('Alfonso Mills');
```

- 9) The procedure `getClinicLocation_sp` gets the details of all the clinics at a particular location entered. It uses `ClinicID` as the foreign key on table `cliniclocation`.

The screenshot shows a SQL Server Enterprise Manager interface. The top pane displays the SQL code for creating and executing a stored procedure. The bottom pane shows the results of the execution in a grid format.

```
2 CREATE procedure getClinicLocation_sp
3 (IN States varchar(255),
4  IN Countries varchar(10))
5 BEGIN
6
7
8  Select cl.StreetAddress,cl.City,cl.State,cl.Country,c.Name
9  from cliniclocation cl left join clinic c
10  on cl.ClinicID = c.ID
11  where state LIKE States AND country LIKE Countries;
12
13  END
14  //
15
16
17 call getClinicLocation_sp('MA','USA');
18
```

StreetAddress	City	State	Country	Name
121.Huntington avenue	Boston	MA	USA	TATA Rehab
35 Parker Street	Boston	MA	USA	Jupiter
35 Hemenwav Street	Boston	MA	USA	TATA Rehab
34.better	boston	MA	USA	NULL
34.better	boston	MA	USA	NULL
43 Smith	boston	MA	USA	NULL
43 Tree h	boston	MA	USA	NULL
43 Tree house Street	boston	MA	USA	NULL
43 Tree house Street	boston	MA	USA	NULL

DELIMITER //

```
CREATE procedure getClinicLocation_sp
```

```
(IN States varchar(255),
```

```
IN Countries varchar(10))
```

```
BEGIN
```

```
  Select cl.StreetAddress,cl.City,cl.State,cl.Country,c.Name
```

```
  from cliniclocation cl left join clinic c
```

```
  on cl.ClinicID = c.ID
```

```
  where state LIKE States AND country LIKE Countries;
```

```
  END
```

```
  //
```

```
call getClinicLocation_sp('MA','USA');
```

- 10) The procedure `makepayment_sp` inserts a record in the payment table based on the pending payments of the patient. It also stores the mode of payment used by the patient. It computes the pending amount based on the treatments taken by the patient and the past payments done by the patient. The foreign key used here is `PatientID` on the payment table and the `TreatmentID` on the treatment table.

The screenshot displays the SQL Server Enterprise Manager interface. The top pane shows the execution of a stored procedure `makePayment_sp` with the following code:

```
1 DELIMITER //
2
3 CREATE procedure makePayment_sp
4 (IN TrID INT,
5  IN Paymode varchar(15),
6  OUT msg varchar(200))
7
8 BEGIN
9   Set @PatientID := (select patient_ID from treatment where ID = TrID LIMIT 1);
10  Set @PType := (select Type from patient where ID = @PatientID LIMIT 1);
11  Set @patientName := (select name from patient where id = @PatientID);
12  SET @AmountDue := 0;
13  SET @Inprice := 0;
```

The bottom pane shows the 'Result Grid' with the following data:

ID	Amount	Mode	patient_ID
4	2750	Debit Card	4
5	2750	Debit Card	4
6	12400	Credit Card	5
NULL	NULL	NULL	NULL

The 'Output' pane shows the 'Action Output' with the following messages:

#	Time	Action	Message
664	05:33:20	call makePayment_sp(20,'Credit Card',@x)	1 row(s) returned
665	05:33:20	select @x LIMIT 0, 1000	1 row(s) returned
666	05:33:20	select * from payment LIMIT 0, 1000	6 row(s) returned

DELIMITER //

CREATE procedure makePayment_sp

(IN TrID INT,

IN Paymode varchar(15),

OUT msg varchar(200))

BEGIN

Set @PatientID := (select patient_ID from treatment where ID = TrID LIMIT 1);

Set @PType := (select Type from patient where ID = @PatientID LIMIT 1);

Set @patientName := (select name from patient where id = @PatientID);

```
SET @AmountDue := 0;

SET @Inprice := 0;

IF Not exists (select * from payment where patient_ID = @PatientID)
THEN
set @cprice := (select A.consultationFees from patientaddiction PA
                LEFT JOIN Addiction A ON A.ID= PA.addiction_ID
                LEFT JOIN Patient P ON P.ID = PA.patient_ID
                WHERE P.ID =@PatientID LIMIT 1);

set @testprice := ( select T.Cost from patientaddiction PA
                  LEFT JOIN Patient P ON P.ID = PA.patient_ID
                  LEFT JOIN Addiction A ON A.ID= PA.addiction_ID
                  LEFT JOIN Doctor D ON D.ID = A.DoctorID
                  LEFT JOIN Test T ON T.DoctorID = D.ID WHERE P.ID =@PatientID LIMIT 1);

IF(@PType != 'OutPatient') THEN

SELECT b.Price,DATEDIFF(A.DischargeDateTime,A.AdmitDateTime) into @Pri,@Days from Patient
P

    left join inpatient inp on p.id = inp.PatientID
    left join admit A ON A.PatientID =P.ID
    LEFT JOIN ROOM RM ON RM.AdmitID = A.ID
    LEFT JOIN Bed b ON b.RoomID = RM.ID
    Where P.ID = @PatientID LIMIT 1;

SET @Inprice:= @Pri * @Days;

END IF;

set @testprice := (select IFNULL (@testprice,0));

set @inprice := (select IFNULL (@testprice,0));

SET @AmountDue = (@inprice + @cprice + @testprice);

INSERT INTO Payment(Amount,Mode,patient_ID) VALUES (@AmountDue,Paymode,@PatientID);

SET msg := CONCAT(@patientName,' paid ',@AmountDue,' Successfully');

ELSE

SET msg := 'hi';
```

```
        END IF;

    END

//

call makePayment_sp(20,'Credit Card',@x);
select @x;

select * from payment;
```

Functions

- 1) **The function diseaseDiagnosis_func is used to generate a diagnosis note based on the type of addiction. It is used in the procedure diseaseDiagnosis_sp.**

```
DELIMITER //

CREATE FUNCTION `diseaseDiagnosis`(AddictionID INT,TestType varchar(100)) RETURNS
VARCHAR(100)

BEGIN

    DECLARE DiagnosisNote VARCHAR(100);

    select name into @AddictionName
    from addiction
    where ID = AddictionID;

    if(TestType = ")

    THEN

        set DiagnosisNote := 'The person has ' +@AddictionName+ ' addiction';

    ELSE
```

```
set DiagnosisNote := 'The person has '+@type+' addiction of '+@AddictionName;  
  
END IF;  
RETURN DiagnosisNote;  
END  
//  
DELIMITER ;
```

Views

- 1) **ClinicwisePatients View:** This view helps us to identify the inflow of patients classified based on different clinics.

The screenshot displays a database management interface with a SQL editor and a results grid. The SQL editor contains the following code:

```
1 • create view clinicwisepatients_view as
2 select clinic.name as `Clinic Name`,count(patient.ID) as `Number of patients` from patient
3 left join clinic
4 on patient.clinicID = clinic.ID
5 left join cliniclocation
6 on clinic.ID = cliniclocation.clinicID
7 group by clinic.ID;
8
9
10 • select * from clinicwisepatients_view;
```

The results grid shows the following data:

Number of patients	Clinic Name
2	Jupiter
6	TATA Rehab
2	Sauous Rehab Clinic
1	Neptune
1	Goldern swan

The output pane shows the execution of the SQL statements:

#	Time	Action	Message
✓ 567	04:00:10	SELECT * FROM rehab_system.clinicwisepatients_view LIMIT 0, 1000	5 row(s) returned
✓ 568	04:00:52	select * from clinicwisepatients_view LIMIT 0, 1000	5 row(s) returned

```
create view clinicwisepatients_view as
select clinic.name as `Clinic Name`,count(patient.ID) as `Number of patients` from patient
left join clinic
on patient.clinicID = clinic.ID
left join cliniclocation
on clinic.ID = cliniclocation.clinicID
group by clinic.ID;

select * from clinicwisepatients_view;
```


- 2) **diagnosisforpatient_view**: This view helps us to find the diagnosis suggested by the doctor to the patient. It also provides us with the results of the diagnosis that was carried out for the patient.

The screenshot displays a database management interface. The top section shows a SQL query being executed. The query is as follows:

```
1 create view diagnosisforpatient_view as
2 select patient.name as `patient name`,doctor.Name as `doctor name`,test.name as `patients test`,diagno
3 from patient
4 left join appointment
5 on appointment.PatientID = patient.ID
6 left join doctor
7 on appointment.DoctorID = doctor.ID
8 left join test
9 on test.DoctorID = doctor.ID
10 left join diagnosis
11 on diagnosis.DoctorID = test.DoctorID
12 where doctor.name IS NOT NULL
13 group by patient.Name;
14
15 select * from diagnosisforpatient_view;
```

Below the query, the 'Result Grid' shows the output of the query. The grid has four columns: 'patient name', 'doctor name', 'patients test', and 'patient diagnosis'. The data is as follows:

patient name	doctor name	patients test	patient diagnosis
Alfonso Mills	Apoorva Gosavi	Ted Sherman	The person has Nealiable addiction of Alcohol
Carlton Cook	Soham Gosavi	MRI	The person has Nealiable addiction of Smoking
Gabriel Simoson	Soham Gosavi	MRI	The person has Nealiable addiction of Smoking
JohnnvWilliamson	Soham Gosavi	MRI	The person has Nealiable addiction of Smoking
Ted Sherman	Apoorva Gosavi	Ted Sherman	The person has Nealiable addiction of Alcohol

At the bottom, the 'Output' section shows the execution log. The log indicates that the view was created successfully and the query returned 5 rows.

#	Time	Action	Message
569	04:03:50	create view diagnosisforpatient_view as select patient.name as `patient name`,d...	Error Code: 1050. Table 'diagnosisforpatient_view' already exi
570	04:03:58	select * from diagnosisforpatient_view LIMIT 0, 1000	5 row(s) returned

```
create view diagnosisforpatient_view as
select patient.name as `patient name`,doctor.Name as `doctor name`,test.name as `patients
test`,diagnosis.note as `patient diagnosis`
from patient left join appointment
on appointment.PatientID = patient.ID
left join doctor on appointment.DoctorID = doctor.ID
left join test on test.DoctorID = doctor.ID
left join diagnosis on diagnosis.DoctorID = test.DoctorID
where doctor.name IS NOT NULL
group by patient.Name;

select * from diagnosisforpatient_view;
```

- 3) **inpatientdetails_view**: This view helps us to identify the details of all the in-patients. IT gives the information about their contact number, admit date, discharge date, the room, floor and the bed they were admitted in during their treatment. It also shows us if a particular patient is still admitted and is undergoing treatment or not.

The screenshot shows a database management tool interface. The top toolbar includes icons for file operations, search, and a 'Limit to 1000 rows' dropdown. The main area displays a SQL script with line numbers 1 through 19. The script creates a view named 'inpatientdetails_view' and then selects all data from it. The view's definition includes a SELECT statement with aliases for patient name, contact number, admit date, discharge date (using IFNULL for patients still admitted), room number, floor, and bed number. It uses multiple LEFT JOINs to connect the patient, inpatient, admit, room, and bed tables. The bottom section shows the 'Result Grid' with a table of data. The table has columns: admitted patient name, contact number, Admit date, discharge date, roomno, floor, and bed number. The data includes patients like Ted Sherman, Carlton Cook, Alfonso Mills, Shannon Greer, pratiksha, and madhuja. The 'discharge date' for pratiksha and madhuja is 'Patient is still admitted'. On the right side, there are buttons for 'Result Grid', 'Form Editor', and 'Read Only'.

```
1 • create view inpatientdetails_view as
2 select inpatient.name as `admitted patient name`,
3 inpatient.contact as `contact number`,
4 admit.AdmitDateTime as `Admit date`,
5 IFNULL(admit.DischargeDateTime,'Patient is still admitted') as `discharge date`,
6 room.RoomNo as `roomno`,room.Floor as `floor`,bed.ID as `bed number`
7 from patient
8 left join inpatient
9 on patient.ID = inpatient.PatientID
10 left join admit
11 on admit.PatientID = inpatient.ID
12 left join room
13 on room.AdmitID = admit.ID
14 left join bed
15 on bed.RoomID = room.ID
16 where inpatient.Name IS NOT NULL
17 group by patient.ID;
18
19 • select * from inpatientdetails_view;
```

admitted patient name	contact number	Admit date	discharge date	roomno	floor	bed number
Ted Sherman	8571003400	2017-10-13 09:00:00	2017-10-14 12:00:00	201	2	1
Carlton Cook	8571003410	2017-08-06 12:00:00	2017-08-24 14:00:00	105	1	4
Alfonso Mills	8571003500	2017-10-15 08:00:00	2017-10-19 10:00:00	101	1	2
Shannon Greer	8571013500	2017-09-23 10:00:00	2017-09-30 11:00:00	303	3	3
pratiksha	90123399	2017-12-12 09:00:00	Patient is still admitted	201	2	8
madhuja	90123399	2017-12-13 09:00:00	Patient is still admitted	301	3	6

```
create view inpatientdetails_view as
select inpatient.name as `admitted patient name`,
inpatient.contact as `contact number`,
admit.AdmitDateTime as `Admit date`,
IFNULL(admit.DischargeDateTime,'Patient is still admitted') as `discharge date`,
room.RoomNo as `roomno`,room.Floor as `floor`,bed.ID as `bed number`
from patient left join inpatient on patient.ID = inpatient.PatientID
left join admit on admit.PatientID = inpatient.ID
left join room on room.AdmitID = admit.ID
left join bed on bed.RoomID = room.ID
where inpatient.Name IS NOT NULL
group by patient.ID;
select * from inpatientdetails_view;
```

- 4) **revenuebymodeofpayment_view**: This view helps us to identify the amount that flowed into the clinic by the means of cash/credit card/debit card. It also shows us the total money earned by the clinic.

The screenshot shows a database management tool interface. The top section contains a SQL query editor with the following code:

```
1 • create view revenuebymodeofpayment_view as
2 select payment.mode,sum(payment.amount) as `revenue by payment-mode`
3 from patient
4 join payment
5 on patient.ID = payment.patient_ID
6 where payment.amount IS NOT NULL
7 group by patient.type with rollup;
8
9 • select * from revenuebymodeofpayment_view;
```

Below the editor, the 'Result Grid' tab is active, displaying the following data:

mode	revenue by payment-mode
Cash	15100
Credit Card	4800
Debit Card	8250
NULL	28150

At the bottom, the 'Output' section shows the 'Action Output' table:

#	Time	Action	Message
✓ 570	04:03:58	select * from diagnosisforpatient_view LIMIT 0, 1000	5 row(s) returned
✓ 571	04:06:04	select * from inpatientdetails_view LIMIT 0, 1000	7 row(s) returned
✓ 572	04:10:03	select * from revenuebymodeofpayment_view LIMIT 0, 1000	4 row(s) returned

```
create view revenuebymodeofpayment_view as
select payment.mode,sum(payment.amount) as `revenue by payment-mode`
from patient
join payment
on patient.ID = payment.patient_ID
where payment.amount IS NOT NULL
group by patient.type with rollup;

select * from revenuebymodeofpayment_view;
```

- 5) **revenuebypatientname_view**: This view helps us to identify the total amount paid by each of the patients to the clinic and also gives us the total amount earned by the clinic to tally.

The screenshot shows a database management interface. The top section contains SQL code for creating a view and querying it. The middle section displays the 'Result Grid' with a table of patient names and their total revenue. The bottom section shows the 'Output' log with execution details.

```
1 • create view revenuebypatientname_view as
2 select patient.name,sum(payment.amount) as `revenue by patient`
3 from patient
4 join payment
5 on patient.ID = payment.patient_ID
6 where payment.amount IS NOT NULL
7 group by patient.Name with rollup;
8
9 • select * from revenuebypatientname_view;
```

name	revenue by patient
Carlton Cook	4800
Gabriel Simoson	8250
JohnnvWilliamson	15100
NULL	28150

Output:

#	Time	Action	Message
✓ 571	04:06:04	select * from inpatientdetails_view LIMIT 0, 1000	7 row(s) returned
✓ 572	04:10:03	select * from revenuebymodeofpayment_view LIMIT 0, 1000	4 row(s) returned
✓ 573	04:12:15	select * from revenuebypatientname_view LIMIT 0, 1000	4 row(s) returned

```
create view revenuebypatientname_view as
select patient.name,sum(payment.amount) as `revenue by patient`
from patient
join payment
on patient.ID = payment.patient_ID
where payment.amount IS NOT NULL
group by patient.Name with rollup;

select * from revenuebypatientname_view;
```

- 6) **revenuebypatienttype_view** : This view helps us to calculate the earnings earned by the company based on the patient types (classified into In-Patients and Out-Patients) and tally it with the total earnings of the company.

The screenshot displays a database management interface. At the top, a toolbar includes icons for file operations, execution, and search, along with a 'Limit to 1000 rows' dropdown. Below the toolbar, a SQL editor contains the following code:

```
1 • create view revenuebypatienttype_view as
2   select patient.type,sum(payment.amount) as `revenue by patient-type`
3   from patient
4   join payment
5   on patient.ID = payment.patient_ID
6   where payment.amount IS NOT NULL
7   group by patient.type with rollup;
8
9 • select * from revenuebypatienttype_view;
```

Below the editor, the 'Result Grid' tab is active, showing the following data:

type	revenue by patient-type
In Patient	4800
Out Patient	23350
NULL	28150

On the right side of the interface, there are buttons for 'Result Grid', 'Form Editor', and 'Read Only'. At the bottom, an 'Output' section shows the 'Action Output' log:

#	Time	Action	Message
✓ 572	04:10:03	select * from revenuebypatienttype_view LIMIT 0, 1000	4 row(s) returned
✓ 573	04:12:15	select * from revenuebypatientname_view LIMIT 0, 1000	4 row(s) returned
✓ 574	04:13:51	select * from revenuebypatienttype_view LIMIT 0, 1000	3 row(s) returned

```
create view revenuebypatienttype_view as
select patient.type,sum(payment.amount) as `revenue by patient-type`
from patient
join payment
on patient.ID = payment.patient_ID
where payment.amount IS NOT NULL
group by patient.type with rollup;

select * from revenuebypatienttype_view;
```

- 7) **upcomingappointments_view**: This view will help to identify the upcoming appointments for the clinics. It gives us the Patient Name and the Doctor Name along with the Date and Time of the appointment.

The screenshot displays a database management interface. The top section shows a SQL script with the following code:

```
1 create view upcomingappointments_view as
2 select appointment.id,patient.name as `Patient Name`,doctor.name as `Doctor Name`,appointment.date
3 from appointment
4 left join patient
5 on appointment.PatientID = patient.id
6 left join doctor
7 on appointment.DoctorID = doctor.id
8 where appointment.date >= now();
9
10
11 select * from upcomingappointments_view;
12
```

Below the code, the 'Result Grid' shows the output of the query:

id	Patient Name	Doctor Name	date
10	Alfonso Mills	Apoorva Gosavi	2017-12-26 09:00:00

The bottom section shows the 'Output' of the SQL actions:

#	Time	Action	Message
578	04:19:25	UPDATE `rehab_system`.`appointment` SET `Date` = '2017-12-26 09:00' WHERE...	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
579	04:19:32	SELECT * FROM rehab_system.appointment LIMIT 0, 1000	10 row(s) returned
580	04:19:45	select * from upcomingappointments_view LIMIT 0, 1000	1 row(s) returned

create view upcomingappointments_viewaddMedicine_sp as

select appointment.id,patient.name as `Patient Name`,doctor.name as `Doctor
Name`,appointment.date

from appointment

left join patient

on appointment.PatientID = patient.id

left join doctor

on appointment.DoctorID = doctor.id

where appointment.date >= now();

select * from upcomingappointments_view;

Dump File:

```
CREATE DATABASE IF NOT EXISTS `rehab_system` /*!40100 DEFAULT CHARACTER
SET utf8 */;
USE `rehab_system`;
-- MySQL dump 10.13  Distrib 5.7.17, for Win64 (x86_64)
--
-- Host: 127.0.0.1    Database: rehab_system
-- -----
-- Server version 5.6.37

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO'
*/;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `addiction`
--

DROP TABLE IF EXISTS `addiction`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `addiction` (
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  `Name` enum('Smoking','Alcohol','Drugs') NOT NULL,
  `DoctorID` int(11) NOT NULL,
  `consultationFees` int(11) DEFAULT NULL,
  PRIMARY KEY (`ID`)
) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `addiction`
--

LOCK TABLES `addiction` WRITE;
/*!40000 ALTER TABLE `addiction` DISABLE KEYS */;
INSERT INTO `addiction` VALUES
(1,'Alcohol',1,1000),(2,'Smoking',2,3600),(3,'Drugs',3,7900),(4,'Smoking',
6,200),(5,'Drugs',7,890),(6,'Smoking',8,500);
/*!40000 ALTER TABLE `addiction` ENABLE KEYS */;
UNLOCK TABLES;
```

```
--
-- Table structure for table `address`
--

DROP TABLE IF EXISTS `address`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `address` (
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  `StreetName` varchar(45) NOT NULL,
  `City` varchar(45) NOT NULL,
  `State` varchar(45) DEFAULT NULL,
  `PatientID` int(11) NOT NULL,
  `ZipCode` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`ID`,`PatientID`),
  KEY `Address_PatientID` (`PatientID`),
  CONSTRAINT `Address_PatientID` FOREIGN KEY (`PatientID`) REFERENCES
`patient` (`ID`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `address`
--

LOCK TABLES `address` WRITE;
/*!40000 ALTER TABLE `address` DISABLE KEYS */;
/*!40000 ALTER TABLE `address` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `admit`
--

DROP TABLE IF EXISTS `admit`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `admit` (
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  `AdmitDateTime` datetime NOT NULL,
  `PatientID` int(11) NOT NULL,
  `DischargeDateTime` datetime DEFAULT NULL,
  PRIMARY KEY (`ID`,`PatientID`),
  KEY `Admit_PatientID` (`PatientID`),
  CONSTRAINT `Admit_PatientID` FOREIGN KEY (`PatientID`) REFERENCES
`inpatient` (`ID`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=15 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `admit`
--

LOCK TABLES `admit` WRITE;
```



```
/*!40000 ALTER TABLE `admit` DISABLE KEYS */;  
INSERT INTO `admit` VALUES (1,'2017-10-13 09:00:00',1,'2017-10-14  
12:00:00'), (2,'2017-10-15 08:00:00',3,'2017-10-19 10:00:00'), (3,'2017-09-  
23 10:00:00',4,'2017-09-30 11:00:00'), (4,'2017-08-06 12:00:00',2,'2017-08-  
24 14:00:00'), (5,'2017-08-06 13:00:00',2,'2017-08-24 15:00:00'), (10,'2017-  
12-12 09:00:00',6,NULL), (12,'2017-12-13 09:00:00',5,NULL), (14,'2017-12-01  
09:00:00',7,NULL);  
/*!40000 ALTER TABLE `admit` ENABLE KEYS */;  
UNLOCK TABLES;
```

```
--  
-- Table structure for table `appointment`  
--
```

```
DROP TABLE IF EXISTS `appointment`;  
/*!40101 SET @saved_cs_client      = @@character_set_client */;  
/*!40101 SET character_set_client = utf8 */;  
CREATE TABLE `appointment` (  
  `ID` int(11) NOT NULL AUTO_INCREMENT,  
  `Date` datetime NOT NULL,  
  `PatientID` int(11) NOT NULL,  
  `DoctorID` int(11) NOT NULL,  
  `ReceptionistID` int(11) NOT NULL,  
  PRIMARY KEY (`ID`),  
  KEY `DoctorID` (`DoctorID`),  
  KEY `PatientID` (`PatientID`),  
  KEY `ReceptionistID` (`ReceptionistID`),  
  CONSTRAINT `DoctorID` FOREIGN KEY (`DoctorID`) REFERENCES `doctor`  
(`ID`) ON DELETE NO ACTION ON UPDATE NO ACTION,  
  CONSTRAINT `PatientID` FOREIGN KEY (`PatientID`) REFERENCES `patient`  
(`ID`) ON DELETE NO ACTION ON UPDATE NO ACTION,  
  CONSTRAINT `ReceptionistID` FOREIGN KEY (`ReceptionistID`) REFERENCES  
`receptionist` (`ID`) ON DELETE NO ACTION ON UPDATE NO ACTION  
) ENGINE=InnoDB AUTO_INCREMENT=15 DEFAULT CHARSET=utf8;  
/*!40101 SET character_set_client = @saved_cs_client */;
```

```
--  
-- Dumping data for table `appointment`  
--
```

```
LOCK TABLES `appointment` WRITE;  
/*!40000 ALTER TABLE `appointment` DISABLE KEYS */;  
INSERT INTO `appointment` VALUES (1,'2017-10-12 12:00:00',1,1,1), (2,'2017-  
10-15 14:00:00',2,2,2), (3,'2017-09-23 09:00:00',3,2,2), (4,'2017-06-08  
11:00:00',4,2,2), (5,'2017-10-27 15:00:00',5,3,3), (6,'2017-12-11  
17:00:00',6,3,3), (7,'2017-12-11 12:24:00',1,2,1), (8,'2017-11-23  
08:24:00',6,3,3), (9,'2017-11-11 12:24:00',5,6,2), (10,'2017-12-26  
09:00:00',5,1,2), (11,'2017-12-11 12:24:00',7,8,1), (12,'2017-12-11  
12:24:00',7,1,1), (13,'2017-12-11 12:24:00',6,1,3), (14,'2017-12-11  
12:24:00',6,2,3);  
/*!40000 ALTER TABLE `appointment` ENABLE KEYS */;  
UNLOCK TABLES;
```

```
--
```

```
-- Table structure for table `bed`  
--
```

```
DROP TABLE IF EXISTS `bed`;  
/*!40101 SET @saved_cs_client      = @@character_set_client */;  
/*!40101 SET character_set_client = utf8 */;  
CREATE TABLE `bed` (  
  `ID` int(11) NOT NULL AUTO_INCREMENT,  
  `RoomID` int(11) NOT NULL,  
  `Price` int(11) DEFAULT NULL,  
  PRIMARY KEY (`ID`,`RoomID`),  
  KEY `Bed_RoomID` (`RoomID`),  
  CONSTRAINT `Bed_RoomID` FOREIGN KEY (`RoomID`) REFERENCES `room` (`ID`)  
ON DELETE NO ACTION ON UPDATE NO ACTION  
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8;  
/*!40101 SET character_set_client = @saved_cs_client */;
```

```
--  
-- Dumping data for table `bed`  
--
```

```
LOCK TABLES `bed` WRITE;  
/*!40000 ALTER TABLE `bed` DISABLE KEYS */;  
INSERT INTO `bed` VALUES  
(1,1,200),(2,2,300),(3,3,250),(4,4,400),(6,7,1200),(7,8,2400),(8,5,1000);  
/*!40000 ALTER TABLE `bed` ENABLE KEYS */;  
UNLOCK TABLES;
```

```
--  
-- Table structure for table `clinic`  
--
```

```
DROP TABLE IF EXISTS `clinic`;  
/*!40101 SET @saved_cs_client      = @@character_set_client */;  
/*!40101 SET character_set_client = utf8 */;  
CREATE TABLE `clinic` (  
  `ID` int(11) NOT NULL AUTO_INCREMENT,  
  `Name` varchar(45) NOT NULL,  
  PRIMARY KEY (`ID`)  
) ENGINE=InnoDB AUTO_INCREMENT=18 DEFAULT CHARSET=utf8;  
/*!40101 SET character_set_client = @saved_cs_client */;
```

```
--  
-- Dumping data for table `clinic`  
--
```

```
LOCK TABLES `clinic` WRITE;  
/*!40000 ALTER TABLE `clinic` DISABLE KEYS */;  
INSERT INTO `clinic` VALUES (1,'Jupiter'),(2,'TATA Rehab'),(3,'Saugus  
Rehab Clinic'),(4,'Neptune'),(5,'Goldern swan'),(6,'Ios swan'),(17,'Redsox  
rehab center');  
/*!40000 ALTER TABLE `clinic` ENABLE KEYS */;  
UNLOCK TABLES;
```

```
--
-- Table structure for table `cliniclocation`
--

DROP TABLE IF EXISTS `cliniclocation`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `cliniclocation` (
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  `StreetAddress` varchar(100) DEFAULT NULL,
  `City` varchar(45) DEFAULT NULL,
  `State` varchar(45) DEFAULT NULL,
  `Country` varchar(45) DEFAULT NULL,
  `ClinicID` int(11) NOT NULL,
  PRIMARY KEY (`ID`,`ClinicID`),
  KEY `Clinic_ClinicID` (`ClinicID`),
  CONSTRAINT `Clinic_ClinicID` FOREIGN KEY (`ClinicID`) REFERENCES
`clinic` (`ID`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=15 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `cliniclocation`
--

LOCK TABLES `cliniclocation` WRITE;
/*!40000 ALTER TABLE `cliniclocation` DISABLE KEYS */;
INSERT INTO `cliniclocation` VALUES (1,'121,Huntington
avenue','Boston','MA','USA',2),(2,'35 Parker
Street','Boston','MA','USA',1),(3,'35 Hemenway
Street','Boston','MA','USA',2),(4,'34,petter
','boston','MA','USA',7),(5,'34,petter ','boston','MA','USA',8),(6,' 43
Smith ','boston','MA','USA',9),(7,' 43 Tree
h','boston','MA','USA',10),(8,' 43 Tree house
Street','boston','MA','USA',11),(9,' 43 Tree house
Street','boston','MA','USA',12),(14,'45 huntington
ave','Colorado','WA','USA',17);
/*!40000 ALTER TABLE `cliniclocation` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Temporary view structure for view `clinicwisepatients_view`
--

DROP TABLE IF EXISTS `clinicwisepatients_view`;
/*!50001 DROP VIEW IF EXISTS `clinicwisepatients_view`*/;
SET @saved_cs_client      = @@character_set_client;
SET character_set_client = utf8;
/*!50001 CREATE VIEW `clinicwisepatients_view` AS SELECT
  1 AS `Number of patients`,
  1 AS `Clinic Name`*/;
SET character_set_client = @saved_cs_client;

--
```

```
-- Table structure for table `diagnosis`
--

DROP TABLE IF EXISTS `diagnosis`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `diagnosis` (
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  `Date` datetime NOT NULL,
  `Note` varchar(500) NOT NULL,
  `TestID` int(11) NOT NULL,
  `DoctorID` int(11) NOT NULL,
  PRIMARY KEY (`ID`),
  KEY `Diagnosis_DoctorID` (`DoctorID`),
  KEY `Diagnosis_TestID` (`TestID`),
  CONSTRAINT `Diagnosis_DoctorID` FOREIGN KEY (`DoctorID`) REFERENCES `doctor` (`ID`) ON DELETE NO ACTION ON UPDATE NO ACTION,
  CONSTRAINT `Diagnosis_TestID` FOREIGN KEY (`TestID`) REFERENCES `test` (`ID`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=23 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `diagnosis`
--

LOCK TABLES `diagnosis` WRITE;
/*!40000 ALTER TABLE `diagnosis` DISABLE KEYS */;
INSERT INTO `diagnosis` VALUES (1,'2017-09-12 11:30:00','The person has Negligible addiction of Alcohol',13,1),(2,'2017-09-12 10:25:00','The person has Negligible addiction of Alcohol',14,3),(3,'2017-10-12 10:40:00','The person has Negligible addiction of Alcohol',15,1),(4,'2017-10-12 10:30:00','The person has Marginal addiction of Alcohol',16,5),(5,'2017-10-12 11:15:00','The person has Negligible addiction of Alcohol',1,4),(6,'2017-10-12 09:50:00','The person has Negligible addiction of Alcohol',2,6),(7,'2017-10-12 09:00:00','The person has Negligible addiction of Alcohol',3,1),(8,'2017-10-12 10:00:00','The person has Negligible addiction of Smoking',4,2),(9,'2017-10-12 11:00:00','The person has Negligible addiction of Smoking',5,1),(10,'2017-10-12 14:00:00','The person has Negligible addiction of Smoking',6,4),(11,'2017-10-12 13:00:00','The person has Negligible addiction of Smoking',7,2),(12,'2017-11-12 12:00:00','The person has Marginal addiction of Smoking',8,5),(22,'2017-11-12 00:00:00','The person has Negligible addiction of Smoking',25,2);
/*!40000 ALTER TABLE `diagnosis` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Temporary view structure for view `diagnosisforpatient_view`
--

DROP TABLE IF EXISTS `diagnosisforpatient_view`;
/*!50001 DROP VIEW IF EXISTS `diagnosisforpatient_view`*/;
SET @saved_cs_client      = @@character_set_client;
```

```
SET character_set_client = utf8;
/*!50001 CREATE VIEW `diagnosisforpatient_view` AS SELECT
  1 AS `patient name`,
  1 AS `doctor name`,
  1 AS `patients test`,
  1 AS `patient diagnosis`*/;
SET character_set_client = @saved_cs_client;

--
-- Table structure for table `doctor`
--

DROP TABLE IF EXISTS `doctor`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `doctor` (
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  `Name` varchar(45) NOT NULL,
  `Age` int(11) DEFAULT NULL,
  `Contact` varchar(15) DEFAULT NULL,
  PRIMARY KEY (`ID`)
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `doctor`
--

LOCK TABLES `doctor` WRITE;
/*!40000 ALTER TABLE `doctor` DISABLE KEYS */;
INSERT INTO `doctor` VALUES (1,'Apoorva
Gosavi',29,'+12983301299'),(2,'Soham Gosavi',22,'+19873301223'),(3,'Shweta
Gosavi',45,'+17592058122'),(4,'sakshi',45,'9120098909'),(5,'sakshi',45,'91
20098909'),(6,'madhuja',45,'9120098909'),(8,'neelambari',30,'8120982210');
/*!40000 ALTER TABLE `doctor` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `inpatient`
--

DROP TABLE IF EXISTS `inpatient`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `inpatient` (
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  `Name` varchar(45) NOT NULL,
  `Contact` varchar(45) NOT NULL,
  `PatientID` int(11) NOT NULL,
  PRIMARY KEY (`ID`),
  KEY `In_PatientID` (`PatientID`),
  CONSTRAINT `In_PatientID` FOREIGN KEY (`PatientID`) REFERENCES `patient`
(`ID`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=utf8;
```

```
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `inpatient`
--

LOCK TABLES `inpatient` WRITE;
/*!40000 ALTER TABLE `inpatient` DISABLE KEYS */;
INSERT INTO `inpatient` VALUES (1,'Ted
Sherman','8571003400',1),(2,'Carlton Cook','8571003410',3),(3,'Alfonso
Mills','8571003500',5),(4,'Shannon
Greer','8571013500',6),(5,'madhuja','90123399',8),(6,'pratiksha','90123399
',7),(7,'saurabh','90123399',9),(8,'Dan','8768912000',11),(9,'Daniel','123
4590011',12);
/*!40000 ALTER TABLE `inpatient` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Temporary view structure for view `inpatientdetails_view`
--

DROP TABLE IF EXISTS `inpatientdetails_view`;
/*!50001 DROP VIEW IF EXISTS `inpatientdetails_view`*/;
SET @saved_cs_client      = @@character_set_client;
SET character_set_client = utf8;
/*!50001 CREATE VIEW `inpatientdetails_view` AS SELECT
  1 AS `admitted patient name`,
  1 AS `contact number`,
  1 AS `Admit date`,
  1 AS `discharge date`,
  1 AS `roomno`,
  1 AS `floor`,
  1 AS `bed number`*/;
SET character_set_client = @saved_cs_client;

--
-- Table structure for table `medicaltreatment`
--

DROP TABLE IF EXISTS `medicaltreatment`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `medicaltreatment` (
  `medicine_ID` int(11) NOT NULL,
  `treatment_ID` int(11) NOT NULL,
  `Dosage` int(11) DEFAULT NULL,
  KEY `fk_medicine_has_treatment_medicin1` (`medicine_ID`),
  KEY `fk_medicine_has_treatment_treatment1` (`treatment_ID`),
  CONSTRAINT `fk_medicine_has_treatment_medicin1` FOREIGN KEY
(`medicine_ID`) REFERENCES `medicine` (`ID`) ON DELETE NO ACTION ON UPDATE
NO ACTION,
  CONSTRAINT `fk_medicine_has_treatment_treatment1` FOREIGN KEY
(`treatment_ID`) REFERENCES `treatment` (`ID`) ON DELETE NO ACTION ON
UPDATE NO ACTION
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `medicaltreatment`
--

LOCK TABLES `medicaltreatment` WRITE;
/*!40000 ALTER TABLE `medicaltreatment` DISABLE KEYS */;
INSERT INTO `medicaltreatment` VALUES
(2,9,1),(3,10,3),(4,11,3),(1,12,2),(4,18,4),(2,19,5),(4,20,3);
/*!40000 ALTER TABLE `medicaltreatment` ENABLE KEYS */;
UNLOCK TABLES;
/*!50003 SET @saved_cs_client      = @@character_set_client */ ;
/*!50003 SET @saved_cs_results    = @@character_set_results */ ;
/*!50003 SET @saved_col_connection = @@collation_connection */ ;
/*!50003 SET character_set_client  = utf8 */ ;
/*!50003 SET character_set_results = utf8 */ ;
/*!50003 SET collation_connection  = utf8_general_ci */ ;
/*!50003 SET @saved_sql_mode      = @@sql_mode */ ;
/*!50003 SET sql_mode              = 'NO_ENGINE_SUBSTITUTION' */ ;
DELIMITER ;;
/*!50003 CREATE*/ /*!50017 DEFINER=`root`@`localhost`*/ /*!50003 TRIGGER
tr_insprescribedMedicine AFTER INSERT ON medicaltreatment
FOR EACH ROW
BEGIN
    set @docID := (select T.doctorID from treatment T
                    LEFT JOIN medicaltreatment MT ON
                    T.ID =MT.treatment_ID LIMIT 1);

    INSERT INTO prescribemedicine (doctor_ID,medicine_ID)
    VALUES (@docID,NEW.medicine_ID);

END */;;
DELIMITER ;
/*!50003 SET sql_mode              = @saved_sql_mode */ ;
/*!50003 SET character_set_client  = @saved_cs_client */ ;
/*!50003 SET character_set_results = @saved_cs_results */ ;
/*!50003 SET collation_connection  = @saved_col_connection */ ;

--
-- Table structure for table `medicine`
--

DROP TABLE IF EXISTS `medicine`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client  = utf8 */;
CREATE TABLE `medicine` (
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  `Name` varchar(200) NOT NULL,
  `ExpiryDate` date NOT NULL,
  `Cost` int(11) DEFAULT NULL,
  PRIMARY KEY (`ID`)
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8;
```

```
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `medicine`
--

LOCK TABLES `medicine` WRITE;
/*!40000 ALTER TABLE `medicine` DISABLE KEYS */;
INSERT INTO `medicine` VALUES (1,'Naltrexone','0000-00-00',200),(2,'Aspirin and Codeine','2018-05-02',200),(3,'Benzonatate','0000-00-00',500),(4,'Crocine','2017-07-12',890),(5,'Clomax','2019-05-22',600),(6,'BGS','2020-04-05',700);
/*!40000 ALTER TABLE `medicine` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `outpatient`
--

DROP TABLE IF EXISTS `outpatient`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `outpatient` (
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  `Name` varchar(45) NOT NULL,
  `PatientID` int(11) NOT NULL,
  PRIMARY KEY (`ID`),
  KEY `Out_PatientID` (`PatientID`),
  CONSTRAINT `Out_PatientID` FOREIGN KEY (`PatientID`) REFERENCES `patient` (`ID`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `outpatient`
--

LOCK TABLES `outpatient` WRITE;
/*!40000 ALTER TABLE `outpatient` DISABLE KEYS */;
INSERT INTO `outpatient` VALUES (1,'Johnny Williamson',2),(2,'Gabriel Simpson',4);
/*!40000 ALTER TABLE `outpatient` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `patient`
--

DROP TABLE IF EXISTS `patient`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `patient` (
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  `Name` varchar(45) NOT NULL,
```



```
`Type` enum('In Patient','Out Patient') DEFAULT NULL,
`ClinicID` int(11) NOT NULL,
PRIMARY KEY (`ID`,`ClinicID`),
KEY `Patient_ClinicID` (`ClinicID`),
CONSTRAINT `Patient_ClinicID` FOREIGN KEY (`ClinicID`) REFERENCES
`clinic` (`ID`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=13 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `patient`
--

LOCK TABLES `patient` WRITE;
/*!40000 ALTER TABLE `patient` DISABLE KEYS */;
INSERT INTO `patient` VALUES (1,'Ted Sherman','In Patient',1),(2,'Johnny
Williamson','Out Patient',2),(3,'Carlton Cook','In
Patient',2),(4,'Gabriel Simpson','Out Patient',2),(5,'Alfonso Mills','In
Patient',3),(6,'Shannon Greer','In Patient',3),(7,'pratiksha','In
Patient',1),(8,'madhuja','In Patient',4),(9,'saurabh','In
Patient',5),(11,'Dan','In Patient',1),(12,'Daniel','In Patient',1);
/*!40000 ALTER TABLE `patient` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `patientaddiction`
--

DROP TABLE IF EXISTS `patientaddiction`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `patientaddiction` (
  `patient_ID` int(11) NOT NULL,
  `addiction_ID` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `patientaddiction`
--

LOCK TABLES `patientaddiction` WRITE;
/*!40000 ALTER TABLE `patientaddiction` DISABLE KEYS */;
INSERT INTO `patientaddiction` VALUES (1,1),(2,3),(3,2),(4,1),(5,2),(6,2);
/*!40000 ALTER TABLE `patientaddiction` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `payment`
--

DROP TABLE IF EXISTS `payment`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
```

```
CREATE TABLE `payment` (  
  `ID` int(11) NOT NULL AUTO_INCREMENT,  
  `Amount` int(11) NOT NULL,  
  `Mode` enum('Cash','Credit Card','Debit Card') DEFAULT NULL,  
  `patient_ID` int(11) NOT NULL,  
  PRIMARY KEY (`ID`,`patient_ID`),  
  KEY `fk_Payment_patient1` (`patient_ID`),  
  CONSTRAINT `fk_Payment_patient1` FOREIGN KEY (`patient_ID`) REFERENCES  
  `patient` (`ID`) ON DELETE NO ACTION ON UPDATE NO ACTION  
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8;  
/*!40101 SET character_set_client = @saved_cs_client */;  
  
--  
-- Dumping data for table `payment`  
--  
  
LOCK TABLES `payment` WRITE;  
/*!40000 ALTER TABLE `payment` DISABLE KEYS */;  
INSERT INTO `payment` VALUES (1,15100,'Cash',2),(2,4800,'Credit  
Card',3),(3,2750,'Debit Card',4),(4,2750,'Debit Card',4),(5,2750,'Debit  
Card',4),(6,12400,'Credit Card',5);  
/*!40000 ALTER TABLE `payment` ENABLE KEYS */;  
UNLOCK TABLES;  
  
--  
-- Table structure for table `prescribemedicine`  
--  
  
DROP TABLE IF EXISTS `prescribemedicine`;  
/*!40101 SET @saved_cs_client      = @@character_set_client */;  
/*!40101 SET character_set_client = utf8 */;  
CREATE TABLE `prescribemedicine` (  
  `doctor_ID` int(11) NOT NULL,  
  `medicine_ID` int(11) NOT NULL,  
  KEY `fk_doctor_has_medicine_doctor1` (`doctor_ID`),  
  KEY `fk_doctor_has_medicine_medicinel` (`medicine_ID`),  
  CONSTRAINT `fk_doctor_has_medicine_doctor1` FOREIGN KEY (`doctor_ID`)  
  REFERENCES `doctor` (`ID`) ON DELETE NO ACTION ON UPDATE NO ACTION,  
  CONSTRAINT `fk_doctor_has_medicine_medicinel` FOREIGN KEY  
  (`medicine_ID`) REFERENCES `medicine` (`ID`) ON DELETE NO ACTION ON UPDATE  
  NO ACTION  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
/*!40101 SET character_set_client = @saved_cs_client */;  
  
--  
-- Dumping data for table `prescribemedicine`  
--  
  
LOCK TABLES `prescribemedicine` WRITE;  
/*!40000 ALTER TABLE `prescribemedicine` DISABLE KEYS */;  
INSERT INTO `prescribemedicine` VALUES  
  (1,1),(1,2),(1,3),(2,1),(2,2),(3,1),(3,2),(1,4),(1,4);  
/*!40000 ALTER TABLE `prescribemedicine` ENABLE KEYS */;  
UNLOCK TABLES;
```

```
--
-- Table structure for table `receptionist`
--

DROP TABLE IF EXISTS `receptionist`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `receptionist` (
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  `Name` varchar(45) NOT NULL,
  `ClinicID` int(11) NOT NULL,
  `Gender` enum('M','F','Other') DEFAULT NULL,
  `DateOfJoining` datetime DEFAULT NULL,
  PRIMARY KEY (`ID`,`ClinicID`),
  KEY `Recep_ClinicID` (`ClinicID`),
  CONSTRAINT `Recep_ClinicID` FOREIGN KEY (`ClinicID`) REFERENCES `clinic`
(`ID`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `receptionist`
--

LOCK TABLES `receptionist` WRITE;
/*!40000 ALTER TABLE `receptionist` DISABLE KEYS */;
INSERT INTO `receptionist` VALUES (1,'Fannie Daniel',1,'M','2017-06-03
09:00:00'),(2,'Angelica Lindsey',2,'F','2016-03-12 11:00:00'),(3,'Abe
Colleen',3,'M','2015-12-29 13:00:00');
/*!40000 ALTER TABLE `receptionist` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `recovery`
--

DROP TABLE IF EXISTS `recovery`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `recovery` (
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  `TreatmentID` int(11) NOT NULL,
  `Description` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`ID`),
  KEY `Recovery_TreatmentID` (`TreatmentID`),
  CONSTRAINT `Recovery_TreatmentID` FOREIGN KEY (`TreatmentID`) REFERENCES
`treatment` (`ID`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=13 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `recovery`
--
```

```
LOCK TABLES `recovery` WRITE;
/*!40000 ALTER TABLE `recovery` DISABLE KEYS */;
INSERT INTO `recovery` VALUES (1,1,'recoved from addiction'),(2,2,'recoved
from addiction'),(3,3,'recoved from addiction'),(4,4,'recoved from
addiction'),(5,5,'recoved from addiction'),(6,6,'recoved from
addiction'),(12,20,'3');
/*!40000 ALTER TABLE `recovery` ENABLE KEYS */;
UNLOCK TABLES;
```

```
--
-- Table structure for table `result`
--
```

```
DROP TABLE IF EXISTS `result`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `result` (
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  `Cured` enum('Full','Partial') NOT NULL,
  `Note` varchar(45) DEFAULT NULL,
  `RecoveryID` int(11) NOT NULL,
  PRIMARY KEY (`ID`),
  KEY `Result_RecoveryID` (`RecoveryID`),
  CONSTRAINT `Result_RecoveryID` FOREIGN KEY (`RecoveryID`) REFERENCES
`recovery` (`ID`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=13 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;
```

```
--
-- Dumping data for table `result`
--
```

```
LOCK TABLES `result` WRITE;
/*!40000 ALTER TABLE `result` DISABLE KEYS */;
INSERT INTO `result` VALUES (1,'Full','recoved',1),(2,'Partial','partially
recoved',2),(3,'Full','recoved fully',3),(4,'Partial','partially recoved
from
addition',4),(5,'Full','recoved',5),(6,'Full','recoved',6),(12,'Partial','
Treatment for patient',12);
/*!40000 ALTER TABLE `result` ENABLE KEYS */;
UNLOCK TABLES;
```

```
--
-- Temporary view structure for view `revenuebymodeofpayment_view`
--
```

```
DROP TABLE IF EXISTS `revenuebymodeofpayment_view`;
/*!50001 DROP VIEW IF EXISTS `revenuebymodeofpayment_view`*/;
SET @saved_cs_client      = @@character_set_client;
SET character_set_client = utf8;
/*!50001 CREATE VIEW `revenuebymodeofpayment_view` AS SELECT
  1 AS `mode`,
  1 AS `revenue by payment-mode`*/;
```

```
SET character_set_client = @saved_cs_client;

--
-- Temporary view structure for view `revenuebypatientname_view`
--

DROP TABLE IF EXISTS `revenuebypatientname_view`;
/*!50001 DROP VIEW IF EXISTS `revenuebypatientname_view`*/;
SET @saved_cs_client      = @@character_set_client;
SET character_set_client = utf8;
/*!50001 CREATE VIEW `revenuebypatientname_view` AS SELECT
  1 AS `name`,
  1 AS `revenue by patient`*/;
SET character_set_client = @saved_cs_client;

--
-- Temporary view structure for view `revenuebypatienttype_view`
--

DROP TABLE IF EXISTS `revenuebypatienttype_view`;
/*!50001 DROP VIEW IF EXISTS `revenuebypatienttype_view`*/;
SET @saved_cs_client      = @@character_set_client;
SET character_set_client = utf8;
/*!50001 CREATE VIEW `revenuebypatienttype_view` AS SELECT
  1 AS `type`,
  1 AS `revenue by patient-type`*/;
SET character_set_client = @saved_cs_client;

--
-- Table structure for table `room`
--

DROP TABLE IF EXISTS `room`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `room` (
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  `RoomNo` int(11) NOT NULL,
  `AdmitID` int(11) NOT NULL,
  `Floor` int(11) DEFAULT NULL,
  PRIMARY KEY (`ID`,`AdmitID`),
  KEY `Room_AdmitID` (`AdmitID`),
  CONSTRAINT `Room_AdmitID` FOREIGN KEY (`AdmitID`) REFERENCES `admit`
(`ID`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `room`
--

LOCK TABLES `room` WRITE;
/*!40000 ALTER TABLE `room` DISABLE KEYS */;
```

```
INSERT INTO `room` VALUES
(1,201,1,2),(2,101,2,1),(3,303,3,3),(4,105,4,1),(5,201,10,2),(7,301,12,3),
(8,401,14,4);
/*!40000 ALTER TABLE `room` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `test`
--

DROP TABLE IF EXISTS `test`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client  = utf8 */;
CREATE TABLE `test` (
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  `Name` varchar(25) NOT NULL,
  `Date` datetime NOT NULL,
  `DoctorID` int(11) NOT NULL,
  `Cost` int(11) DEFAULT NULL,
  `Result` enum('Negligible','Marginal','Critical') DEFAULT NULL,
  PRIMARY KEY (`ID`),
  KEY `Test_DoctorID` (`DoctorID`),
  CONSTRAINT `Test_DoctorID` FOREIGN KEY (`DoctorID`) REFERENCES `doctor`
(`ID`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=26 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client  = @saved_cs_client */;

--
-- Dumping data for table `test`
--

LOCK TABLES `test` WRITE;
/*!40000 ALTER TABLE `test` DISABLE KEYS */;
INSERT INTO `test` VALUES (1,'Ted Sherman','2017-02-12
10:20:00',1,2300,'Negligible'),(2,'Heart rate','2017-04-12
11:00:00',1,2300,'Negligible'),(3,'MRI','2017-06-12
10:00:00',1,2800,'Marginal'),(4,'CTScan','2017-06-12
14:00:00',1,2800,'Critical'),(5,'MA','2017-07-12
13:30:00',1,2200,'Negligible'),(6,'AB','2017-07-12
12:00:00',1,2230,'Marginal'),(7,'CC','2017-08-12
14:00:00',1,2230,'Marginal'),(8,'DD','2017-09-12
15:00:00',1,3000,'Negligible'),(9,'MS','2017-09-12
13:00:00',1,3000,'Negligible'),(10,'ff','2017-10-12
15:00:00',1,3000,'Marginal'),(11,'DF','2017-10-12
16:00:00',1,5400,'Negligible'),(12,'MRI','2017-10-12
16:00:00',2,4400,'Negligible'),(13,'GD','2017-10-12
17:00:00',2,4400,'Negligible'),(14,'BB','2017-10-12
18:00:00',2,4400,'Negligible'),(15,'SD','2017-10-12
19:00:00',2,4400,'Negligible'),(16,'SD','2017-11-12
20:00:00',2,4450,'Marginal'),(25,'Blood','2017-11-12
00:00:00',2,4450,'Negligible');
/*!40000 ALTER TABLE `test` ENABLE KEYS */;
UNLOCK TABLES;
```

```
--
-- Table structure for table `treatment`
--

DROP TABLE IF EXISTS `treatment`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `treatment` (
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  `Date` datetime NOT NULL,
  `DoctorID` int(11) NOT NULL,
  `AddictionID` int(11) NOT NULL,
  `Description` varchar(255) DEFAULT NULL,
  `patient_ID` int(11) NOT NULL,
  PRIMARY KEY (`ID`),
  KEY `Treat_AddictionID` (`AddictionID`),
  KEY `Treat_DoctorID` (`DoctorID`),
  KEY `Treat_PatientID` (`patient_ID`),
  CONSTRAINT `Treat_AddictionID` FOREIGN KEY (`AddictionID`) REFERENCES
`addiction` (`ID`) ON DELETE NO ACTION ON UPDATE NO ACTION,
  CONSTRAINT `Treat_DoctorID` FOREIGN KEY (`DoctorID`) REFERENCES `doctor`
(`ID`) ON DELETE NO ACTION ON UPDATE NO ACTION,
  CONSTRAINT `Treat_PatientID` FOREIGN KEY (`patient_ID`) REFERENCES
`patient` (`ID`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=21 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `treatment`
--

LOCK TABLES `treatment` WRITE;
/*!40000 ALTER TABLE `treatment` DISABLE KEYS */;
INSERT INTO `treatment` VALUES (1,'2017-08-08 12:00:00',1,1,'Daily alcohol
consumption',1),(2,'2017-09-24 11:00:00',2,2,'Severe consumption of
drugs',2),(3,'2017-10-23 15:00:00',2,3,'Consumes 2 packets of ciggrates
daily',3),(4,'2017-10-30 16:00:00',2,1,'heavy alcohol
consumption',4),(5,'2017-11-05 15:50:00',2,2,'drugs',5),(6,'2017-12-12
12:20:00',3,3,'ciggrates consumption',4),(20,'2017-12-18
00:00:00',2,2,'3',5);
/*!40000 ALTER TABLE `treatment` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Temporary view structure for view `upcomingappointments_view`
--

DROP TABLE IF EXISTS `upcomingappointments_view`;
/*!50001 DROP VIEW IF EXISTS `upcomingappointments_view`*/;
SET @saved_cs_client      = @@character_set_client;
SET character_set_client = utf8;
/*!50001 CREATE VIEW `upcomingappointments_view` AS SELECT
  1 AS `id`,
  1 AS `Patient Name`,
```

```
1 AS `Doctor Name`,
1 AS `date`*/;
SET character_set_client = @saved_cs_client;

--
-- Dumping events for database 'rehab_system'
--

--
-- Dumping routines for database 'rehab_system'
--
/*!50003 DROP FUNCTION IF EXISTS `diseaseDiagnosis` */;
/*!50003 SET @saved_cs_client      = @@character_set_client */ ;
/*!50003 SET @saved_cs_results    = @@character_set_results */ ;
/*!50003 SET @saved_col_connection = @@collation_connection */ ;
/*!50003 SET character_set_client  = utf8 */ ;
/*!50003 SET character_set_results = utf8 */ ;
/*!50003 SET collation_connection  = utf8_general_ci */ ;
/*!50003 SET @saved_sql_mode      = @@sql_mode */ ;
/*!50003 SET sql_mode              = 'NO_ENGINE_SUBSTITUTION' */ ;
DELIMITER ;;
CREATE DEFINER=`root`@`localhost` FUNCTION `diseaseDiagnosis`(AddictionID
INT,TestType varchar(100)) RETURNS varchar(100) CHARSET utf8
BEGIN
    DECLARE DiagnosisNote  VARCHAR(100);

select `name` into @AddictionName
from addiction
where ID = AddictionID;

IF(@type = '')

THEN
set DiagnosisNote := concat('The person has ' ,@AddictionName,'
addiction');

ELSE

    set DiagnosisNote := concat('The person has ',@type,' addiction of
',@AddictionName);

END IF;

RETURN DiagnosisNote;
END ;;
DELIMITER ;
/*!50003 SET sql_mode              = @saved_sql_mode */ ;
/*!50003 SET character_set_client  = @saved_cs_client */ ;
/*!50003 SET character_set_results = @saved_cs_results */ ;
/*!50003 SET collation_connection  = @saved_col_connection */ ;
/*!50003 DROP PROCEDURE IF EXISTS `addClinic_sp` */;
/*!50003 SET @saved_cs_client      = @@character_set_client */ ;
/*!50003 SET @saved_cs_results    = @@character_set_results */ ;
```



```
/*!50003 SET @saved_col_connection = @@collation_connection */ ;
/*!50003 SET character_set_client = utf8 */ ;
/*!50003 SET character_set_results = utf8 */ ;
/*!50003 SET collation_connection = utf8_general_ci */ ;
/*!50003 SET @saved_sql_mode = @@sql_mode */ ;
/*!50003 SET sql_mode = 'NO_ENGINE_SUBSTITUTION' */ ;
DELIMITER ;;
CREATE DEFINER=`root`@`localhost` PROCEDURE `addClinic_sp`(IN ClinicName
varchar(255),
IN SAddress varchar(100),
IN Cty varchar(15),
IN Stte varchar(100),
IN County varchar(100))
BEGIN

START TRANSACTION;

INSERT INTO Clinic(Name)
VALUES (ClinicName);
SET @insertedID := LAST_INSERT_ID();

INSERT INTO cliniclocation(StreetAddress, City, State, Country, ClinicID)
VALUES (SAddress, Cty, Stte, County, @insertedID);

COMMIT;
END ;;
DELIMITER ;
/*!50003 SET sql_mode = @saved_sql_mode */ ;
/*!50003 SET character_set_client = @saved_cs_client */ ;
/*!50003 SET character_set_results = @saved_cs_results */ ;
/*!50003 SET collation_connection = @saved_col_connection */ ;
/*!50003 DROP PROCEDURE IF EXISTS `addDoctor_sp` */;
/*!50003 SET @saved_cs_client = @@character_set_client */ ;
/*!50003 SET @saved_cs_results = @@character_set_results */ ;
/*!50003 SET @saved_col_connection = @@collation_connection */ ;
/*!50003 SET character_set_client = utf8 */ ;
/*!50003 SET character_set_results = utf8 */ ;
/*!50003 SET collation_connection = utf8_general_ci */ ;
/*!50003 SET @saved_sql_mode = @@sql_mode */ ;
/*!50003 SET sql_mode = 'NO_ENGINE_SUBSTITUTION' */ ;
DELIMITER ;;
CREATE DEFINER=`root`@`localhost` PROCEDURE `addDoctor_sp`(IN DoctorName
varchar(255),
IN DoctorAge varchar(10),
IN DoctorContact varchar(15),
IN AddictionName varchar(255),
IN ConsultationFees int)
BEGIN

INSERT INTO Doctor (Name, Age, Contact)
VALUES (DoctorName,
```

```
        DoctorAge,  
        DoctorContact);  
  
set @doctorID:= last_insert_id();  
  
INSERT INTO addiction(name,doctorID,Consultationfees)  
        VALUES (AddictionName,@doctorID,ConsultationFees);  
  
END ;;  
DELIMITER ;  
/*!50003 SET sql_mode            = @saved_sql_mode */ ;  
/*!50003 SET character_set_client = @saved_cs_client */ ;  
/*!50003 SET character_set_results = @saved_cs_results */ ;  
/*!50003 SET collation_connection = @saved_col_connection */ ;  
/*!50003 DROP PROCEDURE IF EXISTS `addMedicine_sp` */;  
/*!50003 SET @saved_cs_client     = @@character_set_client */ ;  
/*!50003 SET @saved_cs_results     = @@character_set_results */ ;  
/*!50003 SET @saved_col_connection = @@collation_connection */ ;  
/*!50003 SET character_set_client   = utf8 */ ;  
/*!50003 SET character_set_results   = utf8 */ ;  
/*!50003 SET collation_connection   = utf8_general_ci */ ;  
/*!50003 SET @saved_sql_mode       = @@sql_mode */ ;  
/*!50003 SET sql_mode              = 'NO_ENGINE_SUBSTITUTION' */ ;  
DELIMITER ;;  
CREATE DEFINER=`root`@`localhost` PROCEDURE `addMedicine_sp`(IN Name  
varchar(255),  
IN ExpiryDate date,  
IN Cost int)  
BEGIN  
  
INSERT INTO Medicine(Name,ExpiryDate,Cost)  
        VALUES (Name,  
        ExpiryDate,  
        Cost);  
  
END ;;  
DELIMITER ;  
/*!50003 SET sql_mode            = @saved_sql_mode */ ;  
/*!50003 SET character_set_client = @saved_cs_client */ ;  
/*!50003 SET character_set_results = @saved_cs_results */ ;  
/*!50003 SET collation_connection = @saved_col_connection */ ;  
/*!50003 DROP PROCEDURE IF EXISTS `addPatient_sp` */;  
/*!50003 SET @saved_cs_client     = @@character_set_client */ ;  
/*!50003 SET @saved_cs_results     = @@character_set_results */ ;  
/*!50003 SET @saved_col_connection = @@collation_connection */ ;  
/*!50003 SET character_set_client   = utf8 */ ;  
/*!50003 SET character_set_results   = utf8 */ ;  
/*!50003 SET collation_connection   = utf8_general_ci */ ;  
/*!50003 SET @saved_sql_mode       = @@sql_mode */ ;  
/*!50003 SET sql_mode              = 'NO_ENGINE_SUBSTITUTION' */ ;  
DELIMITER ;;  
CREATE DEFINER=`root`@`localhost` PROCEDURE `addPatient_sp`(IN PatientName  
varchar(255),  
IN PatientType varchar(10),
```

```
IN PCon varchar(15),
IN PCity varchar(100),
IN PAddress varchar(100))
BEGIN
START TRANSACTION;
    SET @ClinicID := (select Clinic.ID from Clinic
                      LEFT JOIN ClinicLocation C ON C.ClinicID = Clinic.ID
                      WHERE C.City IN (PCity) AND C.StreetAddress LIKE
(PAddress));
    INSERT INTO Patient (Name, Type, ClinicID)
                      VALUES (PatientName,
                              PatientType,
                              @ClinicID);
SET @insertedID := LAST_INSERT_ID();
IF PatientType = 'In Patient' THEN

    INSERT INTO inpatient (name, contact, PatientID)
                      VALUES (PatientName, PCon,
                              @insertedID);

ELSE
    INSERT INTO outpatient (Name, PatientID) VALUES
(PatientName, @insertedID);

END IF;
COMMIT;
END ;;
DELIMITER ;
/*!50003 SET sql_mode            = @saved_sql_mode */ ;
/*!50003 SET character_set_client = @saved_cs_client */ ;
/*!50003 SET character_set_results = @saved_cs_results */ ;
/*!50003 SET collation_connection = @saved_col_connection */ ;
/*!50003 DROP PROCEDURE IF EXISTS `addTreatment_sp` */;
/*!50003 SET @saved_cs_client      = @@character_set_client */ ;
/*!50003 SET @saved_cs_results    = @@character_set_results */ ;
/*!50003 SET @saved_col_connection = @@collation_connection */ ;
/*!50003 SET character_set_client  = utf8 */ ;
/*!50003 SET character_set_results = utf8 */ ;
/*!50003 SET collation_connection  = utf8_general_ci */ ;
/*!50003 SET @saved_sql_mode       = @@sql_mode */ ;
/*!50003 SET sql_mode              = 'NO_ENGINE_SUBSTITUTION' */ ;
DELIMITER ;;
CREATE DEFINER=`root`@`localhost` PROCEDURE `addTreatment_sp`(IN
PatientName varchar(255),
IN MedicineName varchar(10),
IN dsq int,
IN Description varchar(255),
IN TDate Datetime,
OUT Msg varchar(200))
BEGIN

SET @PtID := (select ID from Patient where Name like PatientName LIMIT 1);
```

```
SELECT P.addiction_ID,AD.DoctorID into @AdID,@DocID from patientaddiction
P
LEFT JOIN addiction AD ON AD.ID = P.addiction_ID where patient_ID =@PtID
LIMIT 1;
```

```
INSERT INTO treatment (Date,DoctorID,AddictionID,Description,patient_ID)
VALUES (TDate,@docID,@AdID,dsg,@PtID);
SET @treatID := last_insert_id();
```

```
Set @MedicineID := (SELECT ID from medicine where Name like MedicineName
LIMIT 1);
select @MedicineID,MedicineName;
```

```
IF @MedicineID IS not NULL THEN
INSERT INTO medicaltreatment(medicine_ID,treatment_ID,Dosage) VALUES
(@MedicineID,@treatID,dsg);
SET Msg:= CONCAT( 'Medicial Treatment added for ', PatientName);
```

```
INSERT INTO recovery(TreatmentID,Description) VALUES (@treatID,dsg);
SET @RecID :=last_insert_id();
```

```
set @type := ELT(0.5 + rand() *2 , 'Full', 'Partial');
INSERT INTO result(cured,Note,RecoveryID)
VALUES (@type,Description,@RecID);
ELSE
SET Msg:= 'Medicine not found.';
END IF;
```

```
END ;;
DELIMITER ;
/*!50003 SET sql_mode            = @saved_sql_mode */ ;
/*!50003 SET character_set_client = @saved_cs_client */ ;
/*!50003 SET character_set_results = @saved_cs_results */ ;
/*!50003 SET collation_connection = @saved_col_connection */ ;
/*!50003 DROP PROCEDURE IF EXISTS `bookAnAppointment_sp` */;
/*!50003 SET @saved_cs_client      = @@character_set_client */ ;
/*!50003 SET @saved_cs_results     = @@character_set_results */ ;
/*!50003 SET @saved_col_connection = @@collation_connection */ ;
/*!50003 SET character_set_client  = utf8 */ ;
/*!50003 SET character_set_results = utf8 */ ;
/*!50003 SET collation_connection  = utf8_general_ci */ ;
/*!50003 SET @saved_sql_mode       = @@sql_mode */ ;
/*!50003 SET sql_mode              = 'NO_ENGINE_SUBSTITUTION' */ ;
```

```
DELIMITER ;;
CREATE DEFINER=`root`@`localhost` PROCEDURE `bookAnAppointment_sp`(IN
PatientName varchar(255),
IN Addon varchar(10),
IN AppointmentDate datetime,
OUT consultationFees INT)
BEGIN
START TRANSACTION;
```

```
select clinicID,ID into @clinicID,@patientID
from patient
where `Name` LIKE PatientName
```

```
LIMIT 1;

select doctorid into @doctorid
from addiction
where `name` LIKE Addon
order by rand()
LIMIT 1;

select ID into @receptionistID
from receptionist
where clinicID = @clinicID;

INSERT INTO appointment(`Date`,patientID,doctorID,receptionistID)
VALUES
(AppointmentDate,@patientID,@doctorid,@receptionistID);
select @doctorid,Addon;
select consultationFees from addiction where DoctorID = @doctorid AND
`Name` LIKE Addon order by rand() limit 1;
set consultationFees := (select consultationFees from addiction where
DoctorID = @doctorid AND `Name` LIKE Addon order by rand() limit 1);

select addictionID into @addictionID
from addiction
where doctorID = @doctorid and name = Addon;

INSERT INTO patientaddiction(patientID,addictionID) values
(@patientID,@addictionID);
COMMIT;
END ;;
DELIMITER ;
/*!50003 SET sql_mode            = @saved_sql_mode */ ;
/*!50003 SET character_set_client = @saved_cs_client */ ;
/*!50003 SET character_set_results = @saved_cs_results */ ;
/*!50003 SET collation_connection = @saved_col_connection */ ;
/*!50003 DROP PROCEDURE IF EXISTS `diseaseDiagnosis_sp` */;
/*!50003 SET @saved_cs_client      = @@character_set_client */ ;
/*!50003 SET @saved_cs_results     = @@character_set_results */ ;
/*!50003 SET @saved_col_connection = @@collation_connection */ ;
/*!50003 SET character_set_client  = utf8 */ ;
/*!50003 SET character_set_results = utf8 */ ;
/*!50003 SET collation_connection  = utf8_general_ci */ ;
/*!50003 SET @saved_sql_mode       = @@sql_mode */ ;
/*!50003 SET sql_mode              = 'NO_ENGINE_SUBSTITUTION' */ ;
DELIMITER ;;
CREATE DEFINER=`root`@`localhost` PROCEDURE `diseaseDiagnosis_sp`(IN
PatientName varchar(255),
IN TestName varchar(100),
IN TestDate datetime,
IN TestCost INT)
BEGIN
START TRANSACTION;
SET @PatientID := (select ID from patient
WHERE `name` LIKE PatientName LIMIT 1);
```

```
SET @AddictionID := (select addiction_ID from patientaddiction
                     WHERE patient_ID = @PatientID LIMIT 1);
SET @DoctorID := (select doctorID from addiction
                  WHERE ID = @AddictionID);
IF (TestName != '') THEN

set @type := ELT(0.5 + rand() *3 , 'Negligible', 'Marginal', 'Critical');

INSERT INTO Test (Name, DoctorID, Date, Cost, Result)
              VALUES (TestName,
                      @DoctorID,
                      TestDate,
                      TestCost,
                      @type);
Set @TestID := last_insert_id();
END IF;
INSERT INTO diagnosis (Date, Note, TestID, DoctorID)
              VALUES (TestDate, (select diseaseDiagnosis
                                  (@AddictionID, @type)), @TestID, @DoctorID);

COMMIT;
END ;;
DELIMITER ;
/*!50003 SET sql_mode            = @saved_sql_mode */ ;
/*!50003 SET character_set_client = @saved_cs_client */ ;
/*!50003 SET character_set_results = @saved_cs_results */ ;
/*!50003 SET collation_connection = @saved_col_connection */ ;
/*!50003 DROP PROCEDURE IF EXISTS `generateMedicalReport_sp` */;
/*!50003 SET @saved_cs_client     = @@character_set_client */ ;
/*!50003 SET @saved_cs_results   = @@character_set_results */ ;
/*!50003 SET @saved_col_connection = @@collation_connection */ ;
/*!50003 SET character_set_client = utf8 */ ;
/*!50003 SET character_set_results = utf8 */ ;
/*!50003 SET collation_connection = utf8_general_ci */ ;
/*!50003 SET @saved_sql_mode     = @@sql_mode */ ;
/*!50003 SET sql_mode            = 'NO_ENGINE_SUBSTITUTION' */ ;
DELIMITER ;;
CREATE DEFINER=`root`@`localhost` PROCEDURE `generateMedicalReport_sp`(IN
PatientName varchar(255))
BEGIN

select p.name as Patient_Name, a.name as addiction, doc.name as
Doctor_name, doc.contact as Doctor_contact, dia.date as
diagnosis_date, dia.note as Doctor_comments, t.name as
test_name, t.Date, t.cost as test_cost
from patient p
left join patientaddiction pa
on p.id=pa.patient_ID
left join addiction a
on a.id = pa.addiction_ID
left join doctor doc
on doc.id = a.doctorid
left join diagnosis dia
on doc.id = dia.doctorid
```

```
left join test t
on t.DoctorID = dia.DoctorID
left join treatment tr
on tr.patient_ID = p.ID
AND tr.DoctorID = doc.ID
left join medicaltreatment mt
on mt.treatment_ID = tr.ID
left join medicine m
on m.ID = mt.medicine_ID
left join recovery r
on r.TreatmentID = tr.ID
left join result res
on res.RecoveryID = r.ID
where p.Name = PatientName;

END ;;
DELIMITER ;
/*!50003 SET sql_mode            = @saved_sql_mode */ ;
/*!50003 SET character_set_client = @saved_cs_client */ ;
/*!50003 SET character_set_results = @saved_cs_results */ ;
/*!50003 SET collation_connection = @saved_col_connection */ ;
/*!50003 DROP PROCEDURE IF EXISTS `getClinicLocation_sp` */;
/*!50003 SET @saved_cs_client      = @@character_set_client */ ;
/*!50003 SET @saved_cs_results     = @@character_set_results */ ;
/*!50003 SET @saved_col_connection = @@collation_connection */ ;
/*!50003 SET character_set_client  = utf8 */ ;
/*!50003 SET character_set_results = utf8 */ ;
/*!50003 SET collation_connection  = utf8_general_ci */ ;
/*!50003 SET @saved_sql_mode       = @@sql_mode */ ;
/*!50003 SET sql_mode              = 'NO_ENGINE_SUBSTITUTION' */ ;
DELIMITER ;;
CREATE DEFINER=`root`@`localhost` PROCEDURE `getClinicLocation_sp`(IN
States varchar(255),
IN Countries varchar(10))
BEGIN

Select cl.StreetAddress,cl.City,cl.State,cl.Country,c.Name
from cliniclocation cl left join clinic c
on cl.ClinicID = c.ID
where state LIKE States AND country LIKE Countries;

END ;;
DELIMITER ;
/*!50003 SET sql_mode            = @saved_sql_mode */ ;
/*!50003 SET character_set_client = @saved_cs_client */ ;
/*!50003 SET character_set_results = @saved_cs_results */ ;
/*!50003 SET collation_connection = @saved_col_connection */ ;
/*!50003 DROP PROCEDURE IF EXISTS `makePayment_sp` */;
/*!50003 SET @saved_cs_client      = @@character_set_client */ ;
/*!50003 SET @saved_cs_results     = @@character_set_results */ ;
/*!50003 SET @saved_col_connection = @@collation_connection */ ;
/*!50003 SET character_set_client  = utf8 */ ;
/*!50003 SET character_set_results = utf8 */ ;
```

```
/*!50003 SET collation_connection = utf8_general_ci */ ;
/*!50003 SET @saved_sql_mode = @@sql_mode */ ;
/*!50003 SET sql_mode = 'NO_ENGINE_SUBSTITUTION' */ ;
DELIMITER ;;
CREATE DEFINER=`root`@`localhost` PROCEDURE `makePayment_sp`(IN TrID INT,
IN Paymode varchar(15),
OUT msg varchar(200))
BEGIN
Set @PatientID := (select patient_ID from treatment where ID = TrID LIMIT
1);
Set @PType := (select Type from patient where ID = @PatientID LIMIT 1);
Set @patientName := (select name from patient where id = @PatientID);
SET @AmountDue := 0;
SET @Inprice := 0;

IF Not exists (select * from payment where patient_ID = @PatientID)
THEN
set @cprice := (select A.consultationFees from patientaddiction PA
LEFT JOIN Addiction A ON A.ID= PA.addiction_ID
LEFT JOIN Patient P ON P.ID = PA.patient_ID
WHERE P.ID =@PatientID LIMIT 1);

select @cprice;
set @testprice := ( select T.Cost from patientaddiction PA
LEFT JOIN Patient P ON P.ID = PA.patient_ID
LEFT JOIN Addiction A ON A.ID= PA.addiction_ID
LEFT JOIN Doctor D ON D.ID = A.DoctorID
LEFT JOIN Test T ON T.DoctorID = D.ID WHERE P.ID
=@PatientID LIMIT 1);
select @testprice;
IF(@PType != 'OutPatient') THEN
SELECT b.Price,DATEDIFF(A.DischargeDateTime,A.AdmitDateTime) into
@Pri,@Days from Patient P
left join inpatient inp on p.id = inp.PatientID
left join admit A ON A.PatientID =P.ID
LEFT JOIN ROOM RM ON RM.AdmitID = A.ID
LEFT JOIN Bed b ON b.RoomID = RM.ID
Where P.ID = @PatientID LIMIT 1;
SET @Inprice:= @Pri * @Days;
END IF;
select @inprice;
select @cprice;
select @testprice;
set @testprice := (select IFNULL (@testprice,0));
set @inprice := (select IFNULL (@testprice,0));
select @testprice;
SET @AmountDue = (@inprice + @cprice + @testprice);
INSERT INTO Payment(Amount,Mode,patient_ID) VALUES
(@AmountDue,Paymode,@PatientID);
SET msg := CONCAT(@patientName,' paid ',@AmountDue,' Successfully');
SELECT msg;

select @amountDue;
ELSE
SET msg := 'hi';
```



```
END IF;

END ;;
DELIMITER ;
/*!50003 SET sql_mode            = @saved_sql_mode */ ;
/*!50003 SET character_set_client = @saved_cs_client */ ;
/*!50003 SET character_set_results = @saved_cs_results */ ;
/*!50003 SET collation_connection = @saved_col_connection */ ;

--
-- Final view structure for view `clinicwisepatients_view`
--

/*!50001 DROP VIEW IF EXISTS `clinicwisepatients_view`*/;
/*!50001 SET @saved_cs_client          = @@character_set_client */;
/*!50001 SET @saved_cs_results        = @@character_set_results */;
/*!50001 SET @saved_col_connection    = @@collation_connection */;
/*!50001 SET character_set_client      = utf8 */;
/*!50001 SET character_set_results     = utf8 */;
/*!50001 SET collation_connection      = utf8_general_ci */;
/*!50001 CREATE ALGORITHM=UNDEFINED */
/*!50013 DEFINER=`root`@`localhost` SQL SECURITY DEFINER */
/*!50001 VIEW `clinicwisepatients_view` AS select count(`patient`.`ID`) AS
`Number of patients`,`clinic`.`Name` AS `Clinic Name` from ((`patient`
left join `clinic` on((`patient`.`ClinicID` = `clinic`.`ID`))) left join
`cliniclocation` on((`clinic`.`ID` = `cliniclocation`.`ClinicID`))) group
by `clinic`.`ID` */;
/*!50001 SET character_set_client      = @saved_cs_client */;
/*!50001 SET character_set_results     = @saved_cs_results */;
/*!50001 SET collation_connection      = @saved_col_connection */;

--
-- Final view structure for view `diagnosisforpatient_view`
--

/*!50001 DROP VIEW IF EXISTS `diagnosisforpatient_view`*/;
/*!50001 SET @saved_cs_client          = @@character_set_client */;
/*!50001 SET @saved_cs_results        = @@character_set_results */;
/*!50001 SET @saved_col_connection    = @@collation_connection */;
/*!50001 SET character_set_client      = utf8 */;
/*!50001 SET character_set_results     = utf8 */;
/*!50001 SET collation_connection      = utf8_general_ci */;
/*!50001 CREATE ALGORITHM=UNDEFINED */
/*!50013 DEFINER=`root`@`localhost` SQL SECURITY DEFINER */
/*!50001 VIEW `diagnosisforpatient_view` AS select `patient`.`Name` AS
`patient name`,`doctor`.`Name` AS `doctor name`,`test`.`Name` AS `patients
test`,`diagnosis`.`Note` AS `patient diagnosis` from ((((`patient` left
join `appointment` on((`appointment`.`PatientID` = `patient`.`ID`))) left
join `doctor` on((`appointment`.`DoctorID` = `doctor`.`ID`))) left join
`test` on((`test`.`DoctorID` = `doctor`.`ID`))) left join `diagnosis`
on((`diagnosis`.`DoctorID` = `test`.`DoctorID`))) where ((`doctor`.`Name`
is not null) and (`diagnosis`.`TestID` is not null)) group by
`patient`.`Name` */;
/*!50001 SET character_set_client      = @saved_cs_client */;
```

```
/*!50001 SET character_set_results      = @saved_cs_results */;
/*!50001 SET collation_connection       = @saved_col_connection */;

--
-- Final view structure for view `inpatientdetails_view`
--

/*!50001 DROP VIEW IF EXISTS `inpatientdetails_view`*/;
/*!50001 SET @saved_cs_client          = @@character_set_client */;
/*!50001 SET @saved_cs_results        = @@character_set_results */;
/*!50001 SET @saved_col_connection     = @@collation_connection */;
/*!50001 SET character_set_client      = utf8 */;
/*!50001 SET character_set_results     = utf8 */;
/*!50001 SET collation_connection      = utf8_general_ci */;
/*!50001 CREATE ALGORITHM=UNDEFINED */
/*!50013 DEFINER=`root`@`localhost` SQL SECURITY DEFINER */
/*!50001 VIEW `inpatientdetails_view` AS select `inpatient`.`Name` AS
`admitted patient name`,`inpatient`.`Contact` AS `contact
number`,`admit`.`AdmitDateTime` AS `Admit
date`,ifnull(`admit`.`DischargeDateTime`,`Patient is still admitted`) AS
`discharge date`,`room`.`RoomNo` AS `roomno`,`room`.`Floor` AS
`floor`,`bed`.`ID` AS `bed number` from ((((`patient` left join
`inpatient` on((`patient`.`ID` = `inpatient`.`PatientID`))) left join
`admit` on((`admit`.`PatientID` = `inpatient`.`ID`))) left join `room`
on((`room`.`AdmitID` = `admit`.`ID`))) left join `bed` on((`bed`.`RoomID`
= `room`.`ID`))) where (`inpatient`.`Name` is not null) group by
`patient`.`ID` */;
/*!50001 SET character_set_client      = @saved_cs_client */;
/*!50001 SET character_set_results     = @saved_cs_results */;
/*!50001 SET collation_connection      = @saved_col_connection */;

--
-- Final view structure for view `revenuebymodeofpayment_view`
--

/*!50001 DROP VIEW IF EXISTS `revenuebymodeofpayment_view`*/;
/*!50001 SET @saved_cs_client          = @@character_set_client */;
/*!50001 SET @saved_cs_results        = @@character_set_results */;
/*!50001 SET @saved_col_connection     = @@collation_connection */;
/*!50001 SET character_set_client      = utf8 */;
/*!50001 SET character_set_results     = utf8 */;
/*!50001 SET collation_connection      = utf8_general_ci */;
/*!50001 CREATE ALGORITHM=UNDEFINED */
/*!50013 DEFINER=`root`@`localhost` SQL SECURITY DEFINER */
/*!50001 VIEW `revenuebymodeofpayment_view` AS select `payment`.`Mode` AS
`mode`,sum(`payment`.`Amount`) AS `revenue by payment-mode` from
(`patient` join `payment` on((`patient`.`ID` = `payment`.`patient_ID`)))
where (`payment`.`Amount` is not null) group by `payment`.`Mode` with
rollup */;
/*!50001 SET character_set_client      = @saved_cs_client */;
/*!50001 SET character_set_results     = @saved_cs_results */;
/*!50001 SET collation_connection      = @saved_col_connection */;
```

--

```
-- Final view structure for view `revenuebypatientname_view`
--

/*!50001 DROP VIEW IF EXISTS `revenuebypatientname_view`*/;
/*!50001 SET @saved_cs_client      = @@character_set_client */;
/*!50001 SET @saved_cs_results    = @@character_set_results */;
/*!50001 SET @saved_col_connection = @@collation_connection */;
/*!50001 SET character_set_client  = utf8 */;
/*!50001 SET character_set_results = utf8 */;
/*!50001 SET collation_connection  = utf8_general_ci */;
/*!50001 CREATE ALGORITHM=UNDEFINED */
/*!50013 DEFINER=`root`@`localhost` SQL SECURITY DEFINER */
/*!50001 VIEW `revenuebypatientname_view` AS select `patient`.`Name` AS
`name`,sum(`payment`.`Amount`) AS `revenue by patient` from (`patient`
join `payment` on((`patient`.`ID` = `payment`.`patient_ID`))) where
(`payment`.`Amount` is not null) group by `patient`.`Name` with rollup */;
/*!50001 SET character_set_client  = @saved_cs_client */;
/*!50001 SET character_set_results = @saved_cs_results */;
/*!50001 SET collation_connection  = @saved_col_connection */;

--
-- Final view structure for view `revenuebypatienttype_view`
--

/*!50001 DROP VIEW IF EXISTS `revenuebypatienttype_view`*/;
/*!50001 SET @saved_cs_client      = @@character_set_client */;
/*!50001 SET @saved_cs_results    = @@character_set_results */;
/*!50001 SET @saved_col_connection = @@collation_connection */;
/*!50001 SET character_set_client  = utf8 */;
/*!50001 SET character_set_results = utf8 */;
/*!50001 SET collation_connection  = utf8_general_ci */;
/*!50001 CREATE ALGORITHM=UNDEFINED */
/*!50013 DEFINER=`root`@`localhost` SQL SECURITY DEFINER */
/*!50001 VIEW `revenuebypatienttype_view` AS select `patient`.`Type` AS
`type`,sum(`payment`.`Amount`) AS `revenue by patient-type` from
(`patient` join `payment` on((`patient`.`ID` = `payment`.`patient_ID`)))
where (`payment`.`Amount` is not null) group by `patient`.`Type` with
rollup */;
/*!50001 SET character_set_client  = @saved_cs_client */;
/*!50001 SET character_set_results = @saved_cs_results */;
/*!50001 SET collation_connection  = @saved_col_connection */;

--
-- Final view structure for view `upcomingappointments_view`
--

/*!50001 DROP VIEW IF EXISTS `upcomingappointments_view`*/;
/*!50001 SET @saved_cs_client      = @@character_set_client */;
/*!50001 SET @saved_cs_results    = @@character_set_results */;
/*!50001 SET @saved_col_connection = @@collation_connection */;
/*!50001 SET character_set_client  = utf8 */;
/*!50001 SET character_set_results = utf8 */;
/*!50001 SET collation_connection  = utf8_general_ci */;
/*!50001 CREATE ALGORITHM=UNDEFINED */
```

```
/*!50013 DEFINER=`root`@`localhost` SQL SECURITY DEFINER */
/*!50001 VIEW `upcomingappointments_view` AS select `appointment`.`ID` AS
`id`,`patient`.`Name` AS `Patient Name`,`doctor`.`Name` AS `Doctor
Name`,`appointment`.`Date` AS `date` from ((`appointment` left join
`patient` on((`appointment`.`PatientID` = `patient`.`ID`))) left join
`doctor` on((`appointment`.`DoctorID` = `doctor`.`ID`))) where
(`appointment`.`Date` >= now()) */;
/*!50001 SET character_set_client      = @saved_cs_client */;
/*!50001 SET character_set_results     = @saved_cs_results */;
/*!50001 SET collation_connection     = @saved_col_connection */;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

-- Dump completed on 2017-12-14  6:19:33
```