Database Management and Database Design

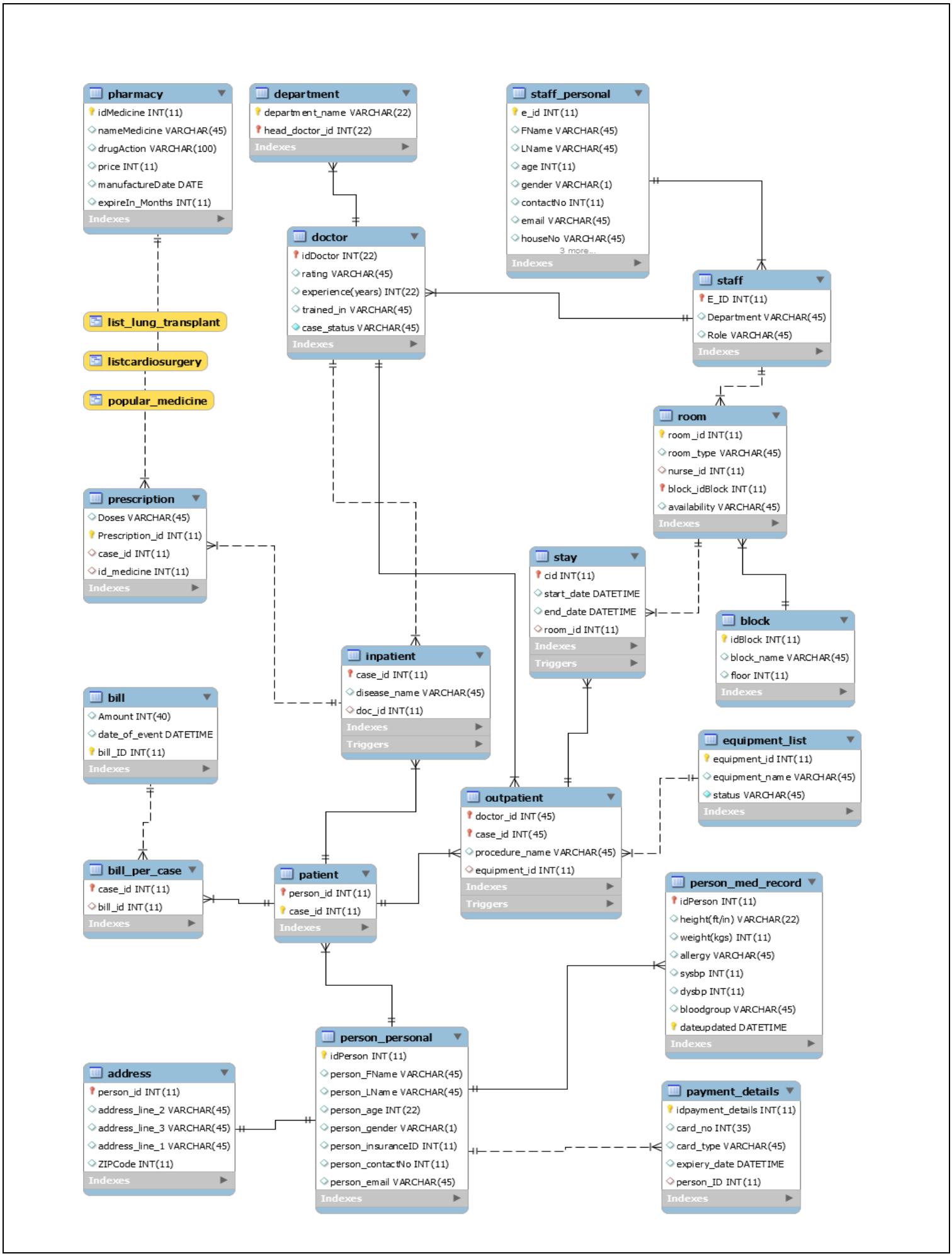
Hospital Database Management

1. Objective Statement
2. Full Schema
3. Users
4. Queries
5. Views
6. Triggers
7. Stored Procedure

Objective Statement

A wide-ranging hospital management database system is designed which address the administrational requirements of the hospitals. In this implementation, the database designed will act as an comprehensive information system which will help in managing all the aspects of hospital's operation such as patient's medical records, managerial administration, resource allocation, room allotment and other services.

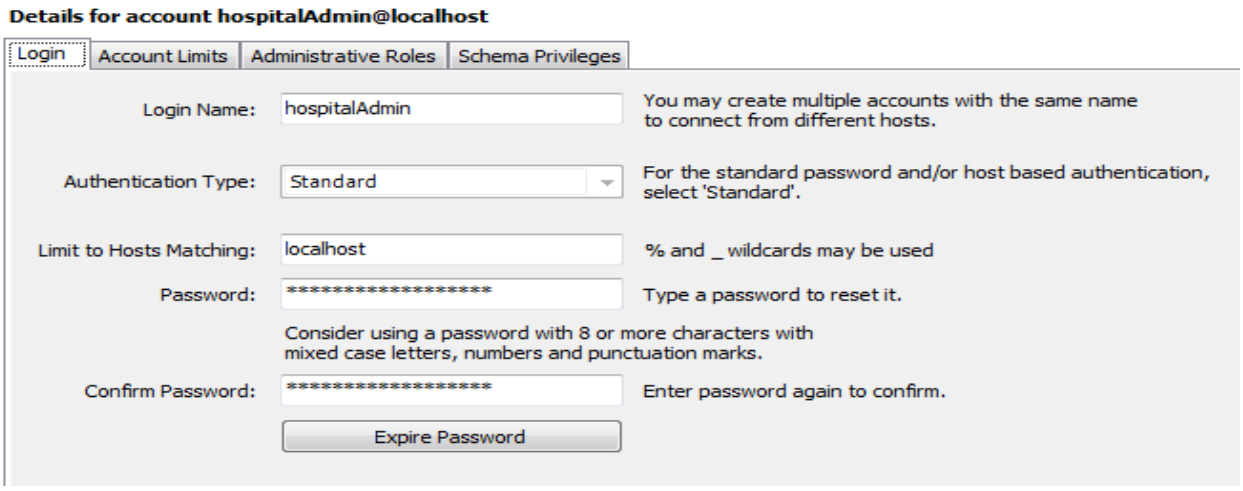
As the person will be allocated with a case_id which will be associated with the person per disease basis. There can be more than one case_ids for a person as the person may have multiple disease and may have separate treatment going for those. Apart from that the person is allocated with a room if he/she goes through any surgery/operations which will fall under the table 'procedure'. The duration of the 'stay' will be found in table: stay. The person will be allocated with the doctor for the treatment and a nurse during the 'stay' period. Block for the room allocated and room type can be determined and through that resource allocation can be planned in better way.

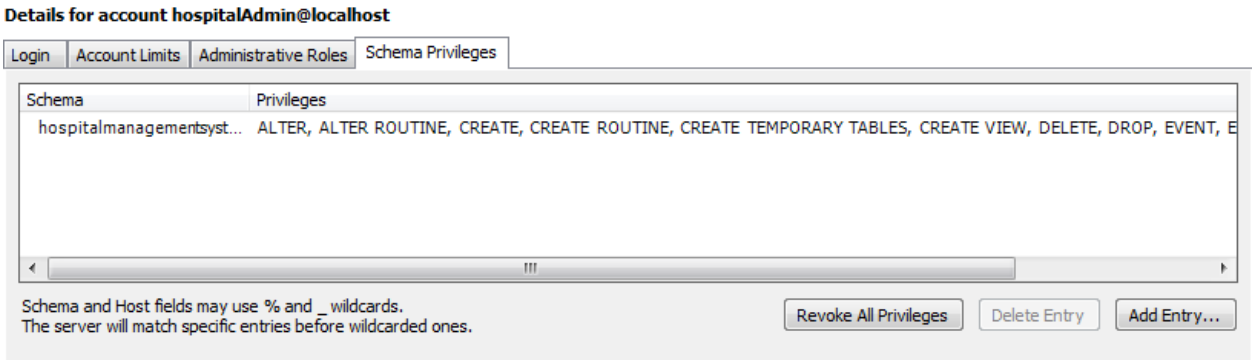

pharmacy
idMedicine INT(11)
nameMedicine VARCHAR(45)
drugAction VARCHAR(100)
price INT(11)
manufactureDate DATE
expireIn_Months INT(11)
Indexes
department
department_name VARCHAR(22)
head_doctor_id INT(22)
Indexes
staff_personal
e_id INT(11)
FName VARCHAR(45)
LName VARCHAR(45)
age INT(11)
gender VARCHAR(1)
contactNo INT(11)
email VARCHAR(45)
houseNo VARCHAR(45)
3 more...
Indexes
doctor
idDoctor INT(22)
rating VARCHAR(45)
experience(years) INT(22)
trained_in VARCHAR(45)
case_status VARCHAR(45)
Indexes
staff
E_ID INT(11)
Department VARCHAR(45)
Role VARCHAR(45)
Indexes
list_lung_transplant
listcardiosurgery
popular_medicine
room
room_id INT(11)
room_type VARCHAR(45)
nurse_id INT(11)
block_idBlock INT(11)
availability VARCHAR(45)
Indexes
prescription
Doses VARCHAR(45)
Prescription_id INT(11)
case_id INT(11)
id_medicine INT(11)
Indexes
stay
cid INT(11)
start_date DATETIME
end_date DATETIME
room_id INT(11)
Indexes
Triggers
block
idBlock INT(11)
block_name VARCHAR(45)
floor INT(11)
Indexes
inpatient
case_id INT(11)
disease_name VARCHAR(45)
doc_id INT(11)
Indexes
Triggers
bill
Amount INT(40)
date_of_event DATETIME
bill_ID INT(11)
Indexes
equipment_list
equipment_id INT(11)
equipment_name VARCHAR(45)
status VARCHAR(45)
Indexes
outpatient
doctor_id INT(45)
case_id INT(45)
procedure_name VARCHAR(45)
equipment_id INT(11)
Indexes
Triggers
bill_per_case
case_id INT(11)
bill_id INT(11)
Indexes
patient
person_id INT(11)
case_id INT(11)
Indexes
person_med_record
idPerson INT(11)
height(ft/in) VARCHAR(22)
weight(kgs) INT(11)
allergy VARCHAR(45)
sysbp INT(11)
dysbp INT(11)
bloodgroup VARCHAR(45)
dateupdated DATETIME
Indexes
person_personal
idPerson INT(11)
person_FName VARCHAR(45)
person_LName VARCHAR(45)
person_age INT(22)
person_gender VARCHAR(1)
person_insuranceID INT(11)
person_contactNo INT(11)
person_email VARCHAR(45)
Indexes
address
person_id INT(11)
address_line_2 VARCHAR(45)
address_line_3 VARCHAR(45)
address_line_1 VARCHAR(45)
ZIPCode INT(11)
Indexes
payment_details
idpayment_details INT(11)
card_no INT(35)
card_type VARCHAR(45)
expiery_date DATETIME
person_ID INT(11)
Indexes

USERS

Hospital Admin:

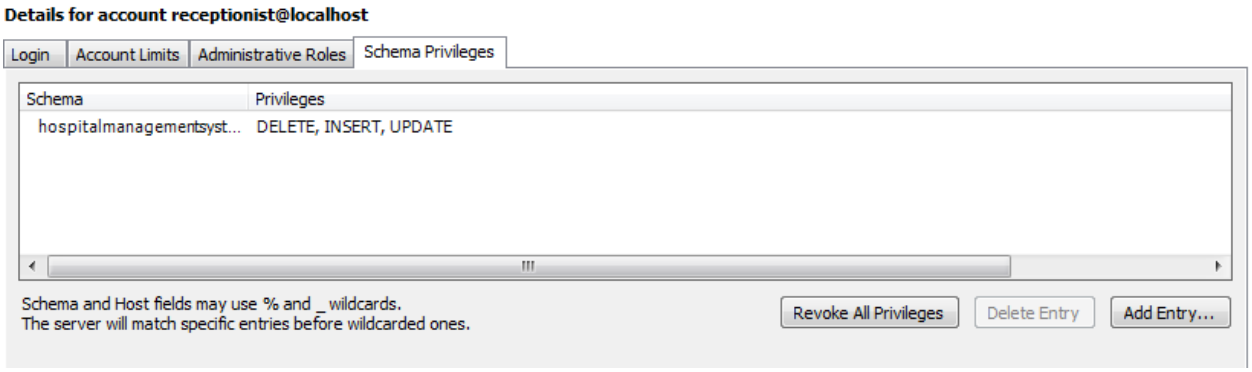
Got all privileges

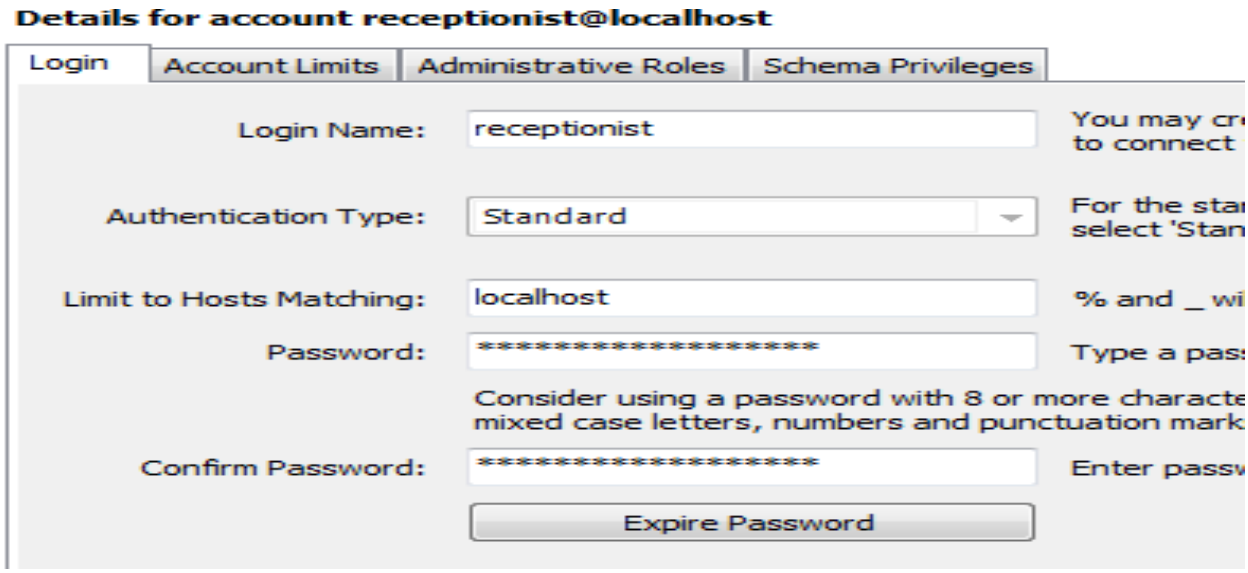




Receptionist Privileges

Delete Insert and update privileges





Tables:

1. **person_personal**
2. **address**
3. **payment_detail**
4. **person_med_record**
5. **patient**
6. **staff_personal**
7. **staff(E_ID, Department, Role)**
8. **doctor (idDoctor , rating, experience,trained_in, case_status)**
9. **inpatient (case_id, disease_name, doc_id)**
10. **pharmacy (idMedcine, nameMedicine, drugAction, price, manufactureDate, expireIn_Months)**
11. **prescription (doses, prescription_id, case_id, id_medicine)**
12. **bill_per_case (case_id, bill_id)**
13. **bill (amount, date_of_event, bill_id)**
14. **outpatient* (doctor_id, case_id, procedure_name, equipment_id) triggering equipment_list.status**
15. **equipment_list (equipment_id, equipment_name, status)**
16. **block (idBlock, block_name, floor)**
17. **room (room_id, room_type, nurse_id, block_idblock, availability)**
18. **stay* (cid, start_date, end_date, room_id) triggering room.availability**
19. **department (department_name, head_doctor_id)**

Queries

1. Finding the person detail of all the female individuals in the database with age<30

   SELECT pp.person_Fname,pp.person_Lname, pp.person_age from person_personal pp where person_gender='F' AND person_age<30;

person_Fname	person_Lname	person_age
Rice	Venere	24
Michel	Penere	23
Kite	Snow	24
Willams	Timothy	23
Bill	Jackson	22

2. Finding the details of the patient by using their case value.(#3)

   SELECT b.bill_id, b.date_of_event, b.amount, pt.person_id, pp.person_Fname, pp.person_Lname,pp.person_age,pp.person_gender from bill b
   inner join bill_per_case bp
   on b.bill_id=bp.bill_id
   inner join patient pt
   on pt.case_id=bp.case_id
   inner join person_personal pp
   on pp.idperson=pt.person_id
   where pt.case_id=3;

bill_id	date_of_event	amount	person_id	person_Fname	person_Lname	person_age	person_gender
3	2016-12-11 00:00:00	90	3	Michel	Penere	23	F

3. Finding the costliest medicine

   SELECT idmedicine, nameMedicine, price

   FROM pharmacy

   WHERE price=(SELECT MAX(price) FROM pharmacy);

idmedicine	nameMedicine	price
15	Dygene	99

4. Finding most purchased medicine

```
SELECT
    `ph`.`nameMedicine` AS `Name of Medicine`,
    `ph`.`idMedicine` AS `Medicine ID`,
    COUNT(`p`.`id_medicine`) AS `Number Of Purchases`
FROM
    (`prescription` `p`
    LEFT JOIN `pharmacy` `ph` ON ((`ph`.`idMedicine` = `p`.`id_medicine`)))
GROUP BY `p`.`id_medicine`
ORDER BY COUNT(`p`.`id_medicine`) DESC
LIMIT 1
```

Name of Medicine	Medicine ID	Number Of Purchases
Allegra	2	3

5. Finding Available Doctors

```
SELECT CONCAT(sp.FName, ',', sp.LName) AS Doctor_Name, d.idDoctor AS Doctor_ID, d.rating AS Doctor_Rating, d.trained_in as Doctor_Specialization, d.case_status as Doctor_Availability FROM hospitalmanagementsystem.doctor d

inner join staff_personal sp on sp.e_id=d.idDoctor

where d.case_status="ON CASE";
```

Doctor_Name	Doctor_ID	Doctor_Rating	Doctor_Specialization	Doctor_Availability
Robin,Wills	1	3	Cardiology	ON CASE
John,Snow	4	3	Emergency medicine	ON CASE
Lalita,Manow	5	3	Paediatric cardiology	ON CASE
Mohit,Arora	7	3	Surgery	ON CASE
John,Snow	15	3	Pediatric Infection Analysis	ON CASE

6. Finding maximum amount paid by patient, patient's Full Name, patient's ID, Case ID of patient, person's insurance number, person's credit card number, person's credit card type, credit card expiry date.

```
SELECT CONCAT(pp.person_FName, ',', pp.person_LName) AS Patient_Name, pp.person_age,
 pp.person_insuranceID, b.amount, b.bill_id, b.date_of_event, bpc.case_id,
 pd.card_no, pd.card_type, pd.expiery_date
FROM   bill b
left join bill_per_case bpc
on b.bill_ID=bpc.bill_id
left join patient pt
on bpc.case_id=pt.case_id
left join person_personal pp
on pp.idPerson=pt.person_id
left join payment_details pd
on pd.person_ID=pp.idPerson
WHERE  b.amount=(SELECT MAX(Amount) FROM bill);
```

Patient_Name	person_age	person_insuranceID	amount	bill_id	date_of_event	case_id	card_no	card_type	expiery_date
John,Peter	20	423433242	900	1	2016-12-11 00:00:00	1	324234234	Master Card	2018-09-22 00:00:00

7. Find people with age more than 30 and their financial details

```
SELECT CONCAT(pp.person_FName, ',', pp.person_LName) AS Patient_Name, pp.person_age,
 pp.person_insuranceID, b.amount, b.bill_id, b.date_of_event, bpc.case_id,
 pd.card_no, pd.card_type, pd.expiery_date
FROM   bill b
left join bill_per_case bpc
on b.bill_ID=bpc.bill_id
left join patient pt
on bpc.case_id=pt.case_id
```

left join person_personal pp

on pp.idPerson=pt.person_id

left join payment_details pd

on pd.person_ID=pp.idPerson

where pp.person_age>30;

Patient_Name	person_age	person_insuranceID	amount	bill_id	date_of_event	case_id	card_no	card_type	expiery_date
Manish,Water	33	423432542	99	6	2016-11-06 00:00:00	6	342342343	FOREX Card	2018-01-12 00:00:00
Tiger,Woods	33	423432442	100	8	2016-11-08 00:00:00	8	332242312	Master Card	2018-11-12 00:00:00
Kate,Red	33	636336366	120	16	2016-12-09 00:00:00	16	NULL	NULL	NULL

8. **Finding people with bill amount greater than 100, their date of transaction, their bill_id, first name and last name.**

select b.amount, b.date_of_event, b.bill_id, bpc.case_id,p.person_id,pp.person_fname, pp.person_lname

from bill b

left join bill_per_case bpc

on b.bill_id=bpc.bill_id

left join patient p

on p.case_id=bpc.case_id

left join person_personal pp

on pp.idperson=p.person_id

where b.amount>100;

amount	date_of_event	bill_id	case_id	person_id	person_fname	person_lname
900	2016-12-11 00:00:00	1	1	1	John	Peter
456	2016-12-09 00:00:00	12	12	12	Mohit	Ramaini
120	2016-12-09 00:00:00	16	16	16	Kate	Red

9. Find the name of medicine taken by person with case id 3 and disease_name, id_medicine and ph.namemedicine

```
select ip.case_id, ip.disease_name, p.id_medicine, ph.namemedicine
from inpatient ip
left join prescription p
on ip.case_id=p.case_id
left join pharmacy ph
on p.id_medicine=ph.idmedicine where ip.case_id=3;
```

case_id	disease_name	id_medicine	namemedicine
3	Menangitis	2	Allegra

10. Find details of cases and disease for which 'Allegra' is taken.

```
select ip.case_id, ip.disease_name, p.id_medicine, ph.namemedicine
from inpatient ip
left join prescription p
on ip.case_id=p.case_id
left join pharmacy ph
on p.id_medicine=ph.idmedicine where ph.namemedicine='allegra';
```

case_id	disease_name	id_medicine	namemedicine
2	Cough	2	Allegra
3	Menangitis	2	Allegra
5	Ulcer Surgery	2	Allegra

4-Views

View 1

List the name and details of people who have the lung transplant.

Provide their full name, their case id and the procedure name

```
SELECT* from list_lung_transplant;


CREATE
    ALGORITHM = UNDEFINED
    DEFINER = `root`@`localhost`
    SQL SECURITY DEFINER
VIEW `list_lung_transplant` AS
    SELECT
        `o`.`procedure_name` AS `procedure_name`,
        `p`.`person_FName` AS `person_Fname`,
        `p`.`person_LName` AS `person_Lname`,
        `o`.`case_id` AS `case_id`
    FROM
        ((`outpatient` `o`
        LEFT JOIN `patient` `pa` ON ((`pa`.`case_id` = `o`.`case_id`)))
        LEFT JOIN `person_personal` `p` ON ((`p`.`idPerson` = `pa`.`person_id`)))
    WHERE
        (`o`.`procedure_name` = 'Lung Transplant')
```

procedure_name	person_Fname	person_Lname	case_id
Lung Transplant	Bill	Gates	9
Lung Transplant	Bill	Jackson	14

View 2

Provide the list of people who are having cardiosurgery.

```
SELECT* from listcardiosurgery;

CREATE
   ALGORITHM = UNDEFINED
   DEFINER = `root`@`localhost`
   SQL SECURITY DEFINER
VIEW `listcardiosurgery` AS
   SELECT
      `o`.`procedure_name` AS `procedure_name`,
      `p`.`person_FName` AS `person_Fname`,
      `p`.`person_LName` AS `person_Lname`,
      `o`.`case_id` AS `case_id`
   FROM
      ((`outpatient` `o`
      LEFT JOIN `patient` `pa` ON ((`pa`.`case_id` = `o`.`case_id`)))
      LEFT JOIN `person_personal` `p` ON ((`p`.`idPerson` = `pa`.`person_id`)))
   WHERE
      (`o`.`procedure_name` = 'Cardio Surgery')
```

	procedure_name	person_Fname	person_Lname	case_id
▶	Cardio Surgery	Barak	Obama	10
	Cardio Surgery	Tiger	Woods	8

View 3

Give the list of medicine which is bought the most

```
SELECT * FROM hospitalmanagementsystem.popular_medicine;


CREATE
    ALGORITHM = UNDEFINED
    DEFINER = `root`@`localhost`
    SQL SECURITY DEFINER
VIEW `popular_medicine` AS
    SELECT
        `ph`.`nameMedicine` AS `Name of Medicine`,
        `ph`.`idMedicine` AS `Medicine ID`,
        COUNT(`p`.`id_medicine`) AS `Number Of Purchases`
    FROM
        (`prescription` `p`
        LEFT JOIN `pharmacy` `ph` ON ((`ph`.`idMedicine` = `p`.`id_medicine`)))
    GROUP BY `p`.`id_medicine`
    ORDER BY COUNT(`p`.`id_medicine`) DESC
    LIMIT 1
```

	Name of Medicine	Medicine ID	Number Of Purchases
▶	Allegra	2	3

4.View

Provide details of people (full name, address, person id, zip) along with their blood group type.

```
CREATE
    ALGORITHM = UNDEFINED
    DEFINER = `root`@`localhost`
    SQL SECURITY DEFINER
VIEW `bloodgroupdetail` AS
    SELECT
        `p`.`idPerson` AS `idperson`,
        CONCAT(`p`.`person_FName`, `p`.`person_LName`) AS `concat(p.person_fname,p.person_lname)`,
        `a`.`address_line_1` AS `address_line_1`,
        `a`.`address_line_2` AS `address_line_2`,
        `a`.`address_line_3` AS `address_line_3`,
        `a`.`ZIPCode` AS `zipcode`,
        `pm`.`bloodgroup` AS `bloodgroup`
    FROM
        ((`person_personal` `p`
        LEFT JOIN `person_med_record` `pm` ON ((`p`.`idPerson` = `pm`.`idPerson`)))
        LEFT JOIN `address` `a` ON ((`a`.`person_id` = `p`.`idPerson`)))
```

idperson	concat(p.person_fname,p.person_lname)	address_line_1	address_line_2	address_line_3	zipcode	bloodgroup
1	JohnPeter	shillman Way	20c	Boston	2120	A+
2	RiceVenere	Montana Way	29c	Boston	2210	B-
3	MichelPenere	Mcgreevy Way	29b	Boston	29221	O
4	RobortSharma	Willams Way	221	Boston	29221	A+
5	KiteSnow	Tremount	29c	Boston	2210	AB
6	ManishWater	Peterborough	21	Boston	2210	B-
7	WillamsTimothy	Johnwalkier	22	Boston	2210	A+
8	TigerWoods	Valkayrie	29c	Boston	2210	A+

Triggers

Trigger 1

Inpatient-After Insert

```
CREATE DEFINER=`root`@`localhost` TRIGGER `hospitalmanagementsystem`.`inpatient_AFTER_INSERT`
AFTER INSERT ON `hospitalmanagementsystem`.`inpatient`
FOR EACH ROW
BEGIN
set @doc_id=new.doc_id;
UPDATE  hospitalmanagementsystem.doctor SET case_status='OFF CASE' where doc_id=@doc_id;
END
```

Trigger 2

Inpatient After Update

```
CREATE DEFINER=`root`@`localhost` TRIGGER `hospitalmanagementsystem`.`inpatient_AFTER_UPDATE`
AFTER UPDATE ON `hospitalmanagementsystem`.`inpatient`
FOR EACH ROW
BEGIN
set @doc_id=new.doc_id;
UPDATE  hospitalmanagementsystem.doctor SET case_status='OFF CASE' where doc_id=@doc_id;
END
```

Trigger 3

Outpatient After Insert

```
CREATE DEFINER=`root`@`localhost` TRIGGER `hospitalmanagementsystem`.`outpatient_AFTER_INSERT`

AFTER INSERT ON `hospitalmanagementsystem`.`outpatient`

FOR EACH ROW

BEGIN

set @newequipid=new.equipment_id;

UPDATE  hospitalmanagementsystem.equipment_list SET status='NA' where
equipment_id=@newequipid;

END
```

Trigger 4

Outpatient After Delete

```
CREATE DEFINER=`root`@`localhost` TRIGGER
`hospitalmanagementsystem`.`outpatient_AFTER_UPDATE`

AFTER UPDATE ON `hospitalmanagementsystem`.`outpatient`

FOR EACH ROW

BEGIN

set @newequipid=new.equipment_id;

UPDATE  hospitalmanagementsystem.equipment_list SET status='NA' where
equipment_id=@newequipid;

END
```

Stored Procedure

1. **To find the patient and its details using the case_id of people.**

call PatientCaseInfo(3);

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `PatientCaseInfo`(IN param1 int)
BEGIN
SELECT b.bill_id, b.date_of_event, b.amount, pt.person_id, pp.person_Fname,
pp.person_Lname,pp.person_age,pp.person_gender from bill b
inner join  bill_per_case bp
on b.bill_id=bp.bill_id
inner join patient pt
on pt.case_id=bp.case_id
inner join person_personal pp
on pp.idperson=pt.person_id
where pt.case_id=param1;
END
```

bill_id	date_of_event	amount	person_id	person_Fname	person_Lname	person_age	person_gender
3	2016-12-11 00:00:00	90	3	Michel	Penere	23	F

2.To find the details of the doctor and the department based on the staff_id.

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `staff_details`(IN param1 int)
BEGIN
SELECT s.e_id, sp.fname,sp.lname, sp.age, s.department, s.role, sp.email, sp.contactno
 from staff s
inner join  staff_personal sp
on s.e_id=sp.e_id
where s.e_id=param1;
END
```

e_id	fname	lname	age	department	role	email	contactno
3	Ronit	Rajen	30	Anesthesia	Doctor	killlmn@yahoo.com	787787879

3.To find the details of patient who paid more than a particular amount for their treatment.

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `amount_paid_details`(in param1 int)
BEGIN
select b.amount, b.date_of_event, b.bill_id, bpc.case_id,p.person_id,pp.person_fname,
pp.person_lname
from bill b
left join bill_per_case bpc
on b.bill_id=bpc.bill_id
left join patient p
on p.case_id=bpc.case_id
left join person_personal pp
on pp.idperson=p.person_id
where b.amount>param1;
END
```

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

amount	date_of_event	bill_id	case_id	person_id	person_fname	person_lname
900	2016-12-11 00:00:00	1	1	1	John	Peter
456	2016-12-09 00:00:00	12	12	12	Mohit	Ramaini
120	2016-12-09 00:00:00	16	16	16	Kate	Red

4. To find the details a case, the medicine name, the disease name and the medicine id by providing the case_id of the patient.

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `diseasemedicine`(in param1 int)
BEGIN
select ip.case_id, ip.disease_name, p.id_medicine, ph.namemedicine
from inpatient ip
left join prescription p
on ip.case_id=p.case_id
left join pharmacy ph
on p.id_medicine=ph.idmedicine where ip.case_id=param1;
END
```

case_id	disease_name	id_medicine	namemedicine
1	Asthama	4	Codeine