EX: 13 MINI PROJECT

DOMAIN: HOSPITAL MAMAGEMENT SYSTEM

**INTRODUCTION:**

The main aim of the project is to create a database that holds the records of doctors, nurses and patients admitted. It should be effective, convenient to use and the data should be secure. It should easily store and retrieve data.

**ASSUMPTIONS MADE:**

The database is created with the following assumptions:

* Out patients must get an appointment to visit a doctor.
* In patients and out patients are differentiated based on lab results.
* The hospital should have many branches.
* Hospital should have nurses and contract staffs along with doctors.
* Each doctor, nurse, contact staff, ambulance driver, others should have an employee id.
* Hospital should have many sections/departments.
* Each department should have many wards.
* Staffs in bill counter are kept in others employee entity.
* Patients admitted should have an individual patient id.
* Each department should have a manager.

**REQUIREMENTS:**

The database is created by keeping the below requirements in mind:

* Data should be stored securely.
* Patient detail should be accurate.
* Data redundancy should not happen.
* Data should be retrieved at faster rate.
* Only authorized person should be allowed to access the database.
* Backups should be taken at regular intervals.

**RISKS:**

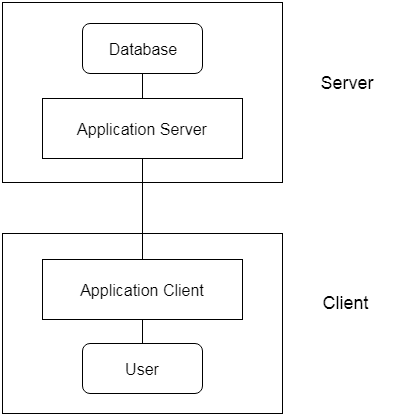
The following risks are encountered in a hospital management system.

* Patient reports can accidentally be changed.
* During any update accidental erasure of data can happen.
* Error can occur during the retrieval of backup data.
* Fear of data security breach.

**TYPE OF ARCHITECTURE USED:**

Three tier architecture is used because of the following reasons:

* It improves data integrity.
* Easy to maintain and modify.
* Security is improved since the client does not have direct access to the database.
* High performance, lightweight, persistent object.
* Better to re-use.



**GENERALIZATION USED:**

The type of ISA used is disjoint. Since, the doctors, nurses, contract staffs, other staffs, ambulance driver work in the same hospital their personal details are kept in a common table employee. The nurses in the hospital can be a manager or a registered nurse. Similarly, the patients can be in-patient or out-patient.

**SOFTWARE REQUIREMENTS:**

The following software are used for this project:

* ER-MODEL: Online ER drawing tool. (Diagrams.io)
* RELATIONAL MODEL: MYSQL Client (Maria DB 10.7 (x64)).
* Windows operating system 10(x64).

**REFERENCES:**

* Database System Concepts – By, Abraham Silberschatz, Henry F. Korth, S. Sudarshan.
* Medium.com
* Wikipedia.com

**CONVERTION OF ER DIAGRAM TO RELATIONAL TABLES:**

**HOSPITAL table:**

In the hospital entity, Licence number is the only key attribute. Address is a composite attribute having street, city, state and postal code as sub-attributes.

|  |  |  |
| --- | --- | --- |
| Li\_no | INT | PRI |
| Branch\_code | INT |  |
| Branch\_name | VARCHAR(30) |  |
| Street | VARCHAR(40) |  |
| City | VARCHAR(25) |  |
| State | VARCHAR(30) |  |
| Postal\_code | INT |  |

**DEPARTMENT table:**

In the department entity, li\_no and section id are the key attributes. Number of admissions, discharges, section name, dept manager, empty wards are the simple attributes.

|  |  |  |
| --- | --- | --- |
| Sec\_id | INT | PRI |
| Li\_no | INT | MUL |
| Sec\_name | VARCHAR(30) |  |
| Dept\_manager | VARCHAR(50) |  |
| Empty\_wards | INT |  |
| No\_admission | INT |  |
| No\_discharge | INT |  |

**WARD table:**

In the ward entity, ward no and sec id are the key attributes. Total beds, ward type, ward name, occupied are the other attributes. Empty bed is a derived attribute.

|  |  |  |
| --- | --- | --- |
| Ward\_no | INT | PRI |
| Sec\_id | INT | MUL |
| Ward\_name | VARCHAR(30) |  |
| Ward\_type | VARCHAR(30) |  |
| Occupied\_bed | INT |  |
| Total\_beds | INT |  |

**EMPLOYEE table:**

Employee table has employee id and li no are the key attributes. Gender, duty time, salary, experience are other attributes. Name is a composite attribute having first name, middle name, last name has sub-attributes. Mobile number is a multivalued attribute.

|  |  |  |
| --- | --- | --- |
| Emp\_id | INT | PRI |
| Li\_no | INT | MUL |
| First\_name | VARCHAR(20) |  |
| Mid\_name | VARCHAR(20) |  |
| Last\_name | VARCHAR(10) |  |
| Experience | INT |  |
| Salary | FLOAT |  |
| Gender | CHAR(2) |  |
| Street | VARCHAR(30) |  |
| City | VARCHAR(30) |  |
| State | VARCHAR(30) |  |
| Postal\_code | INT |  |

**AMBULANCE DRIVER table:**

In this entity emp id and amb no are the key attributes. Duty area, shift and no of trips are the other attributes.

|  |  |  |
| --- | --- | --- |
| Amb\_no | INT | PRI |
| Emp\_id | INT | MUL |
| Duty\_area | VARCHAR(25) |  |
| Shift | CHAR(10) |  |
| No\_trips | INT |  |

**CONTRACT table:**

In this entity, contract no and emp id are the key attributes. Work type, work duration, duty time and qualification are the other attributes.

|  |  |  |
| --- | --- | --- |
| Con\_no | INT | PRI |
| Emp\_id | INT | MUL |
| Work\_type | VARCHAR(30) |  |
| Duty\_time | CHAR(10) |  |
| Work\_duration | INT |  |
| Qualification | VARCHAR(20) |  |

**OTHERS table:**

In this table, id and emp id are the key attributes. Position held, duty time and qualification are the other attributes.

|  |  |  |
| --- | --- | --- |
| Id | INT | PRI |
| Emp\_id | INT | MUL |
| Position\_held | VARCHAR(30) |  |
| Duty\_time | CHAR(10) |  |
| Qualification | VARCHAR(30) |  |

**NURSE table:**

In nurse table nurse\_id, emp id are the key attributes. Qualification is a simple attribute.

|  |  |  |
| --- | --- | --- |
| Nurse\_id | INT | PRI |
| Emp\_id | INT | MUL |
| Qualification | VARCHAR(30) |  |

**NURSE MANAGER table:**

In this table id, emp id and sec id are the key attributes. Qualification is a simple attribute.

|  |  |  |
| --- | --- | --- |
| Id | INT | PRI |
| Emp\_id | INT | MUL |
| Sec\_id | INT | MUL |
| Qualification | VARCHAR(30) |  |

**ROSTER table:**

In this table reg id, ward no and working hours are the key attributes. Duty time and shift are simple attributes.

|  |  |  |
| --- | --- | --- |
| Reg\_id | INT | PRI |
| Ward\_no | INT | MUL |
| Working\_hour | INT | PRI |
| Shift | CHAR(5) |  |
| Duty\_time | CHAR(10) |  |

VISITING DOCTOR table:

In this entity, visit id and emp id are the key attributes. Specialization, no appointments, consult time and consult fee are simple attributes. Visiting days is a multivalued attribute which can take values mon, tue, wed, thurs, fri, sat or sun.

|  |  |  |
| --- | --- | --- |
| Visit\_id | INT | PRI |
| Emp\_id | INT | MUL |
| Specialization | VARCHAR(30) |  |
| Consult\_time | VARCHAR(15) |  |
| Consult\_fee | FLOAT |  |
| No\_appointments | INT |  |

**DOCTOR table:**

In this table, reg id, emp id are the key attributes. Specialization, no appointments, duty time, qualification are simple attributes.

|  |  |  |
| --- | --- | --- |
| Rec\_id | INT | PRI |
| Emp\_id | INT | MUL |
| Qualification | VARCHAR(15) |  |
| Specialization | VARCHAR(20) |  |
| Duty\_time | VARCHAR(10) |  |
| No\_appointments | INT |  |

**TRAINEE table:**

In this entity, trainee id and reg id are the key attributes. Qualification and duty time are simple attributes.

|  |  |  |
| --- | --- | --- |
| Trainee\_id | INT | PRI |
| Reg\_id | INT | MUL |
| Qualification | VARCHAR(15) |  |
| Duty\_time | VARCHAR(10) |  |

**PATIENT table:**

In this entity, patient id and reg id are the key attributes. Patient name and address are composite attributes. Mobile number is a multivalued attribute. Gender, martial status, dob, blood group, height, weight, and father/mother names are simple attributes.

|  |  |  |
| --- | --- | --- |
| Patient\_id | INT | PRI |
| Reg\_id | INT | MUL |
| First\_name | VARCHAR(20) |  |
| Mid\_name | VARCHAR(20) |  |
| Last\_name | VARCHAR(20) |  |
| Gender | CHAR(3) |  |
| Martial\_status | VARCHAR(20) |  |
| Height | INT |  |
| Weight | INT |  |
| Fat/mot\_name | VARCHAR(40) |  |
| Blood\_group | CHAR(5) |  |
| Dob | DATE |  |
| Street | VARCHAR(30) |  |
| City | VARCHAR(30) |  |
| State | VARCHAR(30) |  |
| Pin\_code | INT |  |

**LAB table:**

In this entity, test id and patient id are the key attributes. Date and time are simple attributes.

|  |  |  |
| --- | --- | --- |
| Test\_id | INT | PRI |
| Patient\_id | INT | MUL |
| Date | DATE |  |
| Time | TIME |  |

**LAB RESULT table:**

In this entity, test id and disease are the key attributes. Glucose, bp, hypothesis, procedure, observation are simple attributes.

|  |  |  |
| --- | --- | --- |
| Test\_id | INT | PRI,MUL |
| Disease | VARCHAR(30) | PRI |
| Glucose | FLOAT |  |
| Bp | FLOAT |  |
| Hypothesis | VARCHAR(20) |  |
| Procedure | VARCHAR(30) |  |
| Observation | VARCHAR(30) |  |

**INPATIENT table:**

In this entity, inpatient id, patient id and test id are the key attributes. Treatment and theatre equipment are simple attributes. Medicine is a multivalued attribute.

|  |  |  |
| --- | --- | --- |
| Inpatient\_id | INT | PRI |
| Patient\_id | INT | MUL |
| Test\_id | INT | MUL |
| Treatment\_req | VARCHAR(50) |  |
| Theatre\_equipment | VARCHAR(50) |  |

**ADMISSION table:**

This is a weak entity. It totally depends on in patient entity. Admission no and inpatient id are the key attributes. Date of admission, date of discharge, time of admission, time of discharge are simple attributes.

|  |  |  |
| --- | --- | --- |
| Admission\_no | INT | PRI |
| Inpatient\_id | INT | PRI,MUL |
| Doa | DATE |  |
| Toa | TIME |  |
| Dod | DATE |  |
| Tod | TIME |  |

**CASE HISTORY table:**

In this entity, case no and patient id are the key attributes. Consult doctor, disease, surgery, last admit, last discharge are simple attributes.

|  |  |  |
| --- | --- | --- |
| Case\_no | INT | PRI |
| Patient\_id | INT | PRI,MUL |
| Consult\_doctor | VARCHAR(20) |  |
| Disease | VARCHAR(30) |  |
| Surgery | VARCHAR(30) |  |
| Last\_admit | DATE |  |
| Last\_discharge | DATE |  |

**OUT PATIENT table:**

In this entity, outpatient no and patient id are key attributes. Problems encountered is a simple attribute.

|  |  |  |
| --- | --- | --- |
| Outpatient\_no | INT | PRI |
| Patient\_id | INT | MUL |
| Problem\_encountered | VARCHAR(50) |  |

**APPOINTMENT table:**

In this entity, appointment no and outpatient no are the key attributes. Consult doctor, date, time and slot (1/0) are the simple attributes.

|  |  |  |
| --- | --- | --- |
| Appointment\_no | INT | PRI |
| Outpatient\_no | INT | MUL |
| Consult\_doctor | VARCHAR(30) |  |
| Date | DATE |  |
| Time | TIME |  |
| Slot | INT |  |

**PHARMACY table:**

In this entity, medicine id and li no are the key attributes. Medicine type, price, stock, mfg date and exp date are simple attributes.

|  |  |  |
| --- | --- | --- |
| Medic\_id | INT | PRI |
| Li\_no | INT | MUL |
| Medic\_type | VARCHAR(20) |  |
| Price | FLOAT |  |
| Stock | INT |  |
| Mfg\_date | DATE |  |
| Exp\_date | DATE |  |

**ORDERS table:** In this table, order no and medic id are the key attributes. Order date, arrival date, reorder level are simple attributes.

|  |  |  |
| --- | --- | --- |
| Order\_no | INT | PRI |
| Medic\_id | INT | PRI,MUL |
| Order\_date | DATE |  |
| Arrival\_date | DATE |  |
| Reorder\_level | INT |  |

**BILLING table:**

In this entity, bill no and li no are the key attributes. Date, time and amount are the simple attributes.

|  |  |  |
| --- | --- | --- |
| Bill\_no | INT | PRI |
| Li\_no | INT | MUL |
| Date | DATE |  |
| Time | TIME |  |
| Amount | FLOAT |  |

**PHARMACY BILLING table:**

In this entity, no, bill no are the key attributes. Medicine is a multivalued attribute. Prescription status (1/0), tax, discount are simple attributes.

|  |  |  |
| --- | --- | --- |
| No | INT | PRI |
| Bill\_no | INT | MUL |
| Prescription\_status | INT |  |
| Tax | FLOAT |  |
| Discount | FLOAT |  |

**INPATIENT BILLING table:**

In this entity, no, bill no and patient id are the key attributes. Days stayed, nurse charges, theater equipment, tax, lab charges, scan charge, ward charge, doctor fee are simple attributes.

|  |  |  |
| --- | --- | --- |
| No | INT | PRI |
| Bill\_no | INT | MUL |
| Patient\_id | INT | MUL |
| No\_days | INT |  |
| Nurse\_charges | FLOAT |  |
| Theatre\_equipment | FLOAT |  |
| Tax | FLOAT |  |
| Lab\_charge | FLOAT |  |
| Scan\_charge | FLOAT |  |
| Doctor\_charge | FLOAT |  |

**HEALTH INSURANCE table:**

In this entity, insurance id, no, patient id and account no are the key attributes. Claim status, amount claim, plan name, insure from, insure to, nature illness, relation insured are simple attributes.

|  |  |  |
| --- | --- | --- |
| Insurance\_id | INT | PRI |
| No | INT | PRI,MUL |
| Patient\_id | INT | PRI |
| Account\_no | INT | PRI |
| Claim\_status | INT |  |
| Plan\_name | VARCHAR(30) |  |
| Amount\_claim | FLOAT |  |
| Insure\_from | DATE |  |
| Insure\_to | DATE |  |
| Relation\_insured | VARCHAR(20) |  |

**OUTPATIENT BILLING table:**

In this entity, bill no, no and patient id are the key attributes. Lab charge, scan charge and doctor fee are simple attributes.

|  |  |  |
| --- | --- | --- |
| Bill\_no | INT | PRI |
| No | INT | MUL |
| Patient\_id | INT | MUL |
| Lab\_charge | FLOAT |  |
| Scan\_charge | FLOAT |  |
| Doctor\_fee | FLOAT |  |

**NORMALIZATION DONE:**

**HOSPITAL table:**

In hospital table we have address has composite attribute. We apply 1-NF to reduce the composite attribute address to smaller attributes like street, city, state, postal code.

|  |  |  |
| --- | --- | --- |
| Li\_no | INT | PRI |
| Branch\_code | INT |  |
| Branch\_name | VARCHAR(30) |  |
| Street | VARCHAR(40) |  |
| City | VARCHAR(25) |  |
| State | VARCHAR(30) |  |
| Postal\_code | INT |  |

**EMPLOYEE table:**

In this table address and name are composite attributes. So applying 1-NF we get,

|  |  |  |
| --- | --- | --- |
| Emp\_id | INT | PRI |
| Li\_no | INT | MUL |
| First\_name | VARCHAR(20) |  |
| Mid\_name | VARCHAR(20) |  |
| Last\_name | VARCHAR(10) |  |
| Experience | INT |  |
| Salary | FLOAT |  |
| Gender | CHAR(2) |  |
| Street | VARCHAR(30) |  |
| City | VARCHAR(30) |  |
| State | VARCHAR(30) |  |
| Postal\_code | INT |  |

Name is reduced to first name, middle name, last name and address is reduced to street, city, state, postal code.

Employee mobile number is a multivalued attribute. By applying 1-NF we get,

|  |  |  |
| --- | --- | --- |
| Emp\_id | INT | PRI,MUL |
| Mobile | CHAR(12) | PRI |

We create a separate table called emp\_mobile which has emp id and mobile hassimple attributes. Emp id along with mobile act has a primary key for the table.

**NURSE table:**

In nurse table we have an attribute called qualification. Since nurses can have a maximum of 3 possible degrees, we get data redundancy. To avoid it we create a table called nurse\_qualification which has qualification\_no and qualification has simple attributes.

|  |  |  |
| --- | --- | --- |
| Qualification\_id | INT | PRI |
| Qualification | VARCHAR(20) |  |

**VISITING DOCTOR table:**

In this table we have visit days attribute has multivalued. By applying 1-NF we get,

|  |  |  |
| --- | --- | --- |
| Visit\_id | INT | PRI,MUL |
| Days | CHAR(20) | PRI |

We create a separate table called visit\_days which has visit id and days has simple attributes. Visit id along with days forms the primary key for the table.

**DOCTOR table:**

In this table we have a simple attribute called qualification. Since this can take values MBBS or MS we have data redundancy. So to avoid it we create a separate table called doctor\_qualification which has qualification id and qualification has key attributes.

|  |  |  |
| --- | --- | --- |
| Qualification\_id | INT | PRI |
| Qualification | VARCHAR(20) |  |

**PATIENT table:**

In this table patient name and address are composite attributes. So, we apply 1NF,

|  |  |  |
| --- | --- | --- |
| Patient\_id | INT | PRI |
| Reg\_id | INT | MUL |
| First\_name | VARCHAR(20) |  |
| Mid\_name | VARCHAR(20) |  |
| Last\_name | VARCHAR(20) |  |
| Gender | CHAR(3) |  |
| Martial\_status | VARCHAR(20) |  |
| Height | INT |  |
| Weight | INT |  |
| Fat/mot\_name | VARCHAR(40) |  |
| Blood\_group | CHAR(5) |  |
| Dob | DATE |  |
| Street | VARCHAR(30) |  |
| City | VARCHAR(30) |  |
| State | VARCHAR(30) |  |
| Pin\_code | INT |  |

Patient name is reduced to first name, middle name, last name and address to street, city, state, pin code.

In patient table patient mobile is a multivalued attribute. So we create a separate table called patient\_mobile which takes patient id and mobile has key attributes.

|  |  |  |
| --- | --- | --- |
| Patient\_id | INT | PRI,MUL |
| Mobile | CHAR(12) | PRI |

**IN PATIENT table:**

This table has a multivalued attribute called medicine. On applying 1-NF we create a separate table called patient\_medic which takes two attributes in patient id and medicine.

|  |  |  |
| --- | --- | --- |
| Inpatient\_no | INT | PRI,MUL |
| Medicine | VARCHAR(50) | PRI |

**PHARMACY BILLING table:**

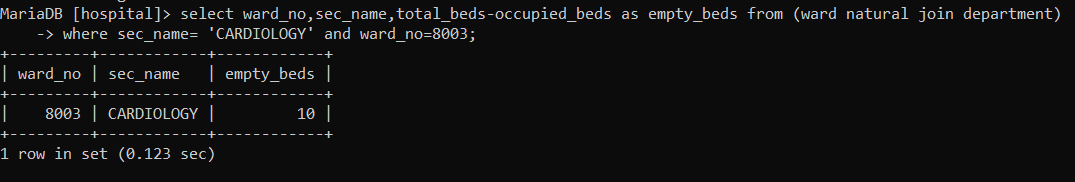
In this table we need to list what are the medicine included in the bill. If we create a separate attribute medicine we get data redundancy. So, to avoid it we create a separate table called medic\_cart which has attributes medic\_cart no and medic id.

|  |  |  |
| --- | --- | --- |
| Mediccart\_no | INT | PRI |
| Medic\_id | INT | MUL |

**QUERIES SOLVED:**

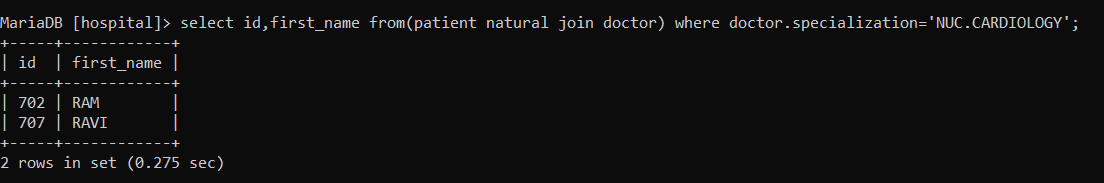
QUERY 1--> FIND THE NUMBER OF EMPTY BEDS IN WARD NO 8003 OF CARDIOLOGY DEPARTMENT ?

MariaDB [hospital]> select ward\_no,sec\_name,total\_beds-occupied\_beds as empty\_beds from (ward natural join department) where sec\_name= 'CARDIOLOGY' and ward\_no=8003;



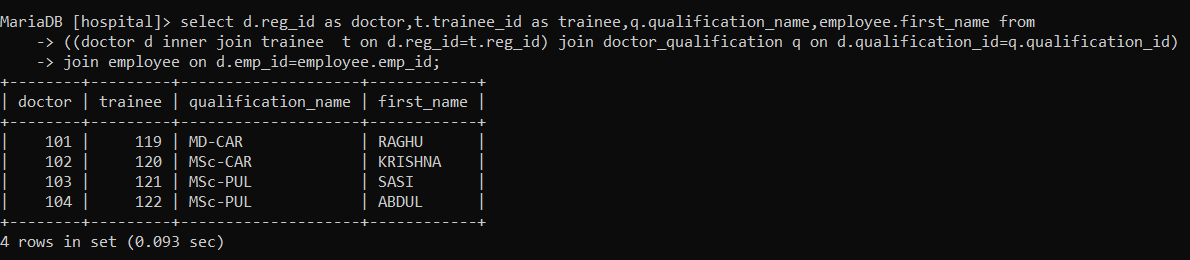
QUERY 2--> FIND THE PATIENTS WHO GETS TREATMENT FROM NUCLEAR CARDIOLOGY DOCTOR ?

MariaDB [hospital]> select id,first\_name from(patient natural join doctor) where doctor.specialization='NUC.CARDIOLOGY';



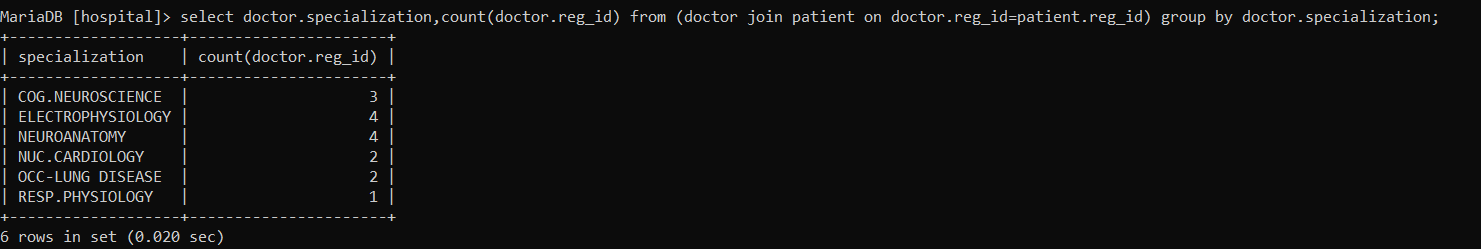
QUERY 3--> LIST THE DOCTOR AND CORESSPONDING TRAINEE WHO WORKS WITH HIM ?

MariaDB [hospital]> select d.reg\_id as doctor,t.trainee\_id as trainee,q.qualification\_name,employee.first\_name from ((doctor d inner join trainee t on d.reg\_id=t.reg\_id) join doctor\_qualification q on d.qualification\_id=q.qualification\_id) join employee on d.emp\_id=employee.emp\_id;//



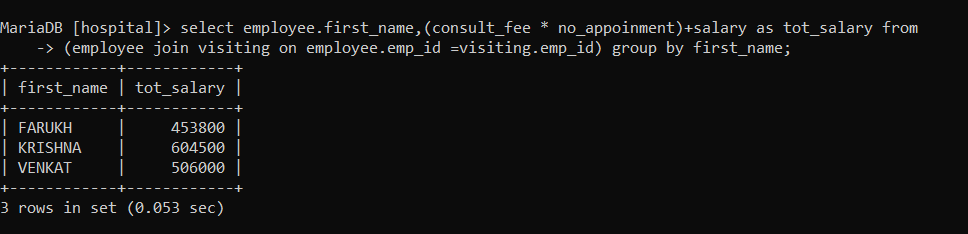
QUERY 4--> COUNT THE NUMBER OF DOCTORS WITH SAME SPECIALIZATION ?

MariaDB [hospital]> select doctor.specialization,count(doctor.reg\_id) from (doctor join patient on doctor.reg\_id=patient.reg\_id) group by doctor.specialization;



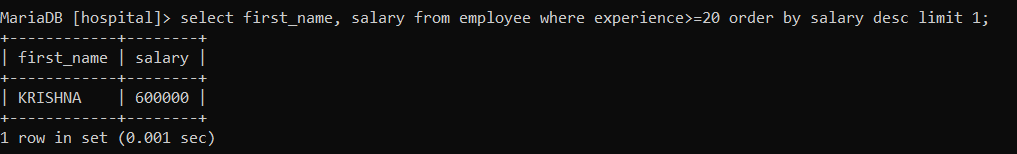
QUERY 5--> FOR VISTING DOCTORS CALCULATE THE TOTAL SALARY ?

MariaDB [hospital]> select employee.first\_name,(consult\_fee \* no\_appoinment)+salary as tot\_salary from (employee join visiting on employee.emp\_id =visiting.emp\_id) group by first\_name;



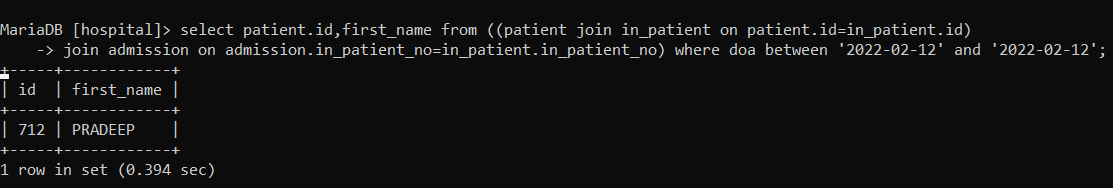
QUERY 6--> FIND THE EMPLOYEE WHO HAS EXPERIENCE MORE THAN 20 YEARS AND GETS THE HIGHEST SALARY ?

MariaDB [hospital]> select first\_name, salary from employee where experience>=20 order by salary desc limit 1;



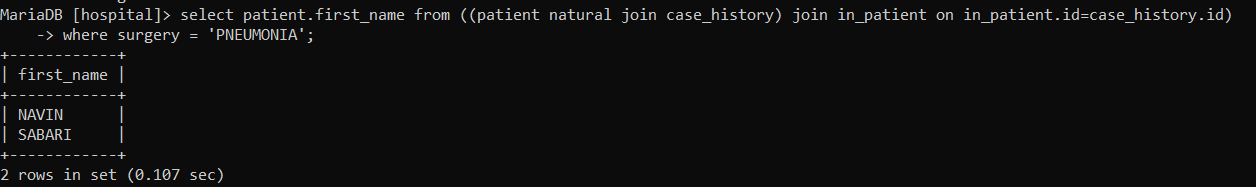
QUERY 7--> FIND THE PATIENT WHO GOT ADMIITED BETWEEN '2022-02-12' and '2022-02-12' ?

MariaDB [hospital]> select patient.id,first\_name from ((patient join in\_patient on patient.id=in\_patient.id) join admission on admission.in\_patient\_no=in\_patient.in\_patient\_no) where doa between '2022-02-12' and '2022-02-12';//



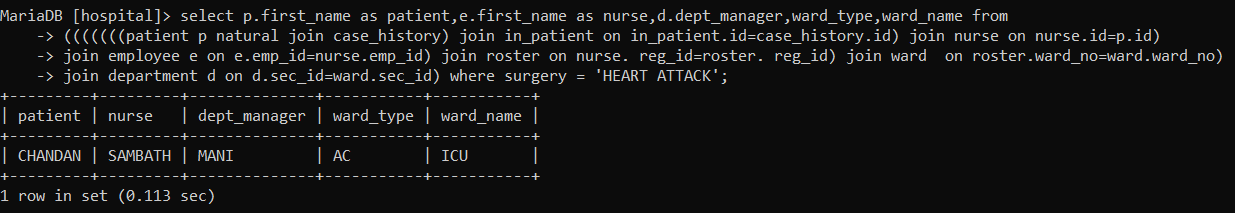
QUERY 8--> FIND THE PATIENT WHO HAD PNEUMONIA PREVIOUSLY ?

MariaDB [hospital]> select patient.first\_name from ((patient natural join case\_history) join in\_patient on in\_patient.id=case\_history.id) where surgery = 'PNEUMONIA';//



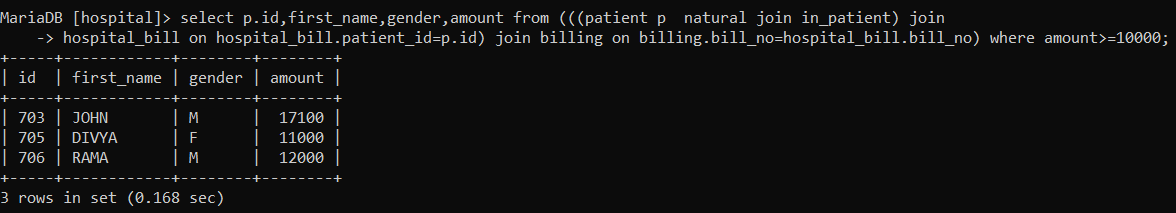
QUERY 9--> LIST THE PATIENT NAME,NURSE TREATING HIM,WARD IN WHICH HE IS ADMITTED AND THE DEPARTMENT MANAGER NAME WHO HAS UNDERGONE A SURGERY FOR HEART ATTACK ?

MariaDB [hospital]> select p.first\_name as patient,e.first\_name as nurse,d.dept\_manager,ward\_type,ward\_name from (((((((patient p natural join case\_history) join in\_patient on in\_patient.id=case\_history.id) join nurse on nurse.id=p.id) join employee e on e.emp\_id=nurse.emp\_id) join roster on nurse. reg\_id=roster. reg\_id) join ward on roster.ward\_no=ward.ward\_no)join department d on d.sec\_id=ward.sec\_id) where surgery = 'HEART ATTACK';//



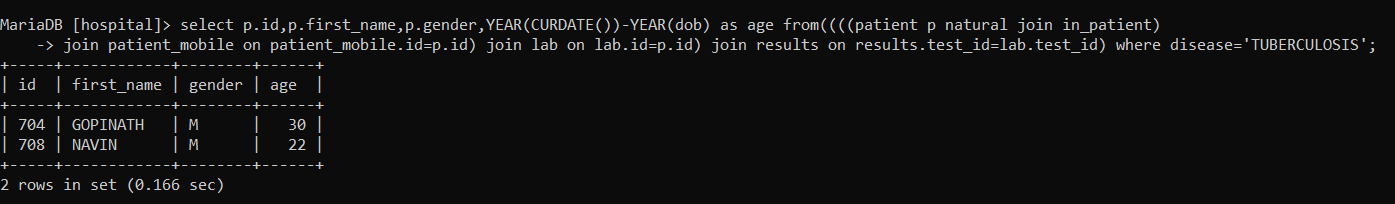
QUERY 10--> LIST THE PATIENT DETAILS WHOSE BILL AMOUNT IS GREATER THAN 10,000 ?

MariaDB [hospital]> select p.id,first\_name,gender,amount from (((patient p natural join in\_patient) join hospital\_bill on hospital\_bill.patient\_id=p.id) join billing on billing.bill\_no=hospital\_bill.bill\_no) where amount>=10000;//



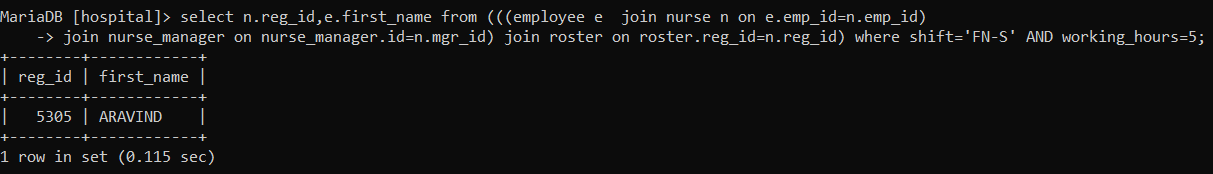
QUERY 11--> DISPLAY PATIENT NAME AND AGE WHO IS DIAGNOISED WITH TUBERCLOSIS ?

MariaDB [hospital]> select p.id,p.first\_name,p.gender,YEAR(CURDATE())-YEAR(dob) as age from((((patient p natural join in\_patient) join patient\_mobile on patient\_mobile.id=p.id) join lab on lab.id=p.id) join results on results.test\_id=lab.test\_id) where disease='TUBERCULOSIS';//



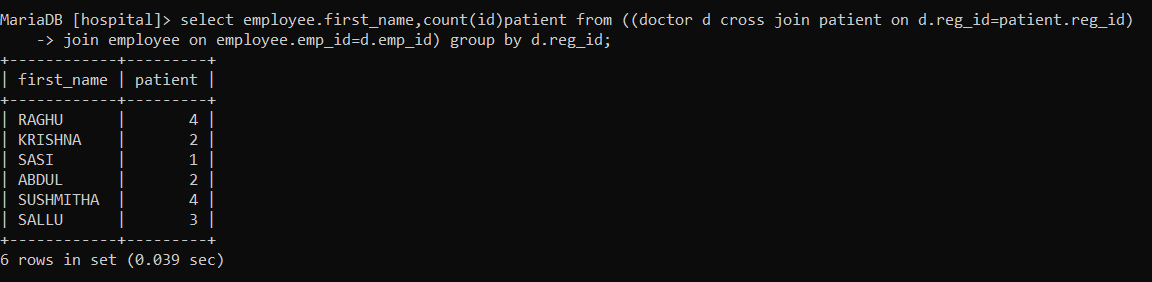
QUERY 12--> LIST THE NURSE DETAILS WHO HAVE ROSTER DUTY AT FORENOON SHIFT AND WORKING HOURS = 5 HOURS ?

MariaDB [hospital]> select n.reg\_id,e.first\_name from (((employee e join nurse n on e.emp\_id=n.emp\_id) join nurse\_manager on nurse\_manager.id=n.mgr\_id) join roster on roster.reg\_id=n.reg\_id) where shift='FN-S' AND working\_hours=5;//



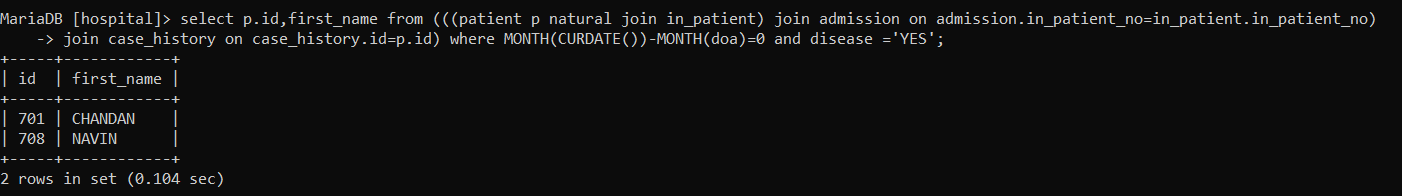
QUERY 13--> WRITE A QUERY TO FIND NUMBER OF PATIENTS WHO COME UNDER SAME DOCTOR ?

MariaDB [hospital]> select employee.first\_name,count(id)patient from ((doctor d cross join patient on d.reg\_id=patient.reg\_id) join employee on employee.emp\_id=d.emp\_id) group by d.reg\_id;//



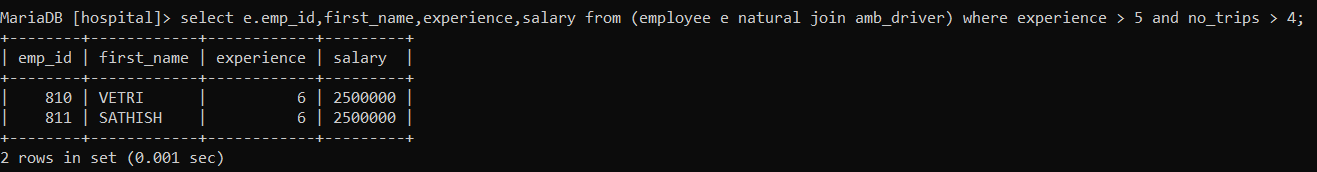
QUERY 14--> LIST THE PATIENTS WHO ARE ADMITTED TODAY AND HAS UNDERGONE SURGERY PREVIOUSLY ?

MariaDB [hospital]> select p.id,first\_name from (((patient p natural join in\_patient) join admission on admission.in\_patient\_no=in\_patient.in\_patient\_no) join case\_history on case\_history.id=p.id) where MONTH(CURDATE())-MONTH(doa)=0 and disease ='YES';//



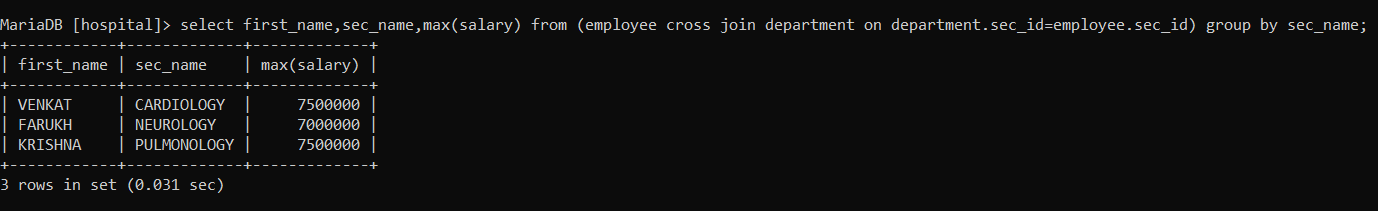
QUERY 15--> LIST THE DETAILS OF AMBULANCE DRIVER WHO HAS A EXPERIENCE OF MORE THAN 5 YEARS AND MAKES 4 TRIPS DAILY ?

MariaDB [hospital]> select e.emp\_id,first\_name,experience,salary from (employee e natural join amb\_driver) where experience > 5 and no\_trips > 4;//



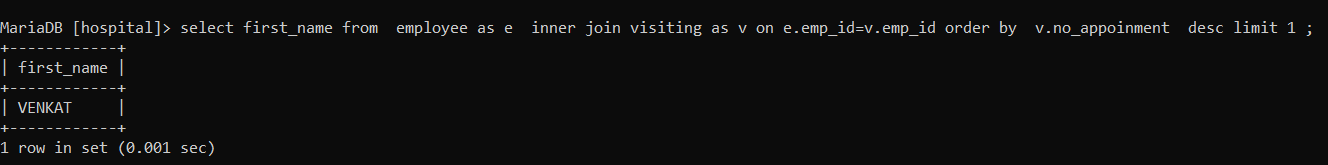
QUERY 16--> FIND THE EMPLOYEE WHO GETS THE MAXIMUM SALARY IN HIS/HER DEPARTMENT ?

MariaDB [hospital]> select first\_name,sec\_name,max(salary) from (employee cross join department on department.sec\_id=employee.sec\_id) group by sec\_name;



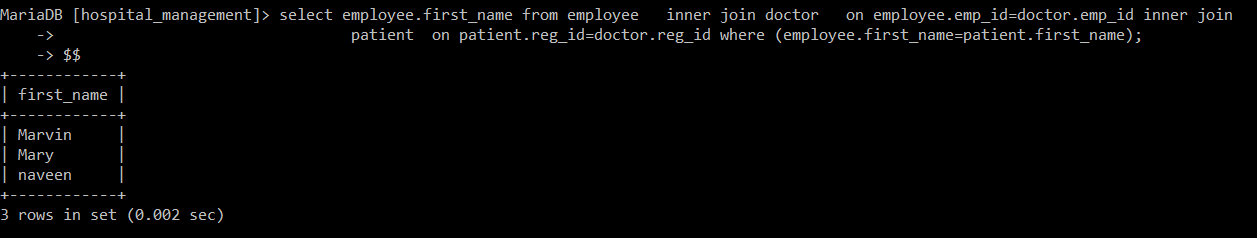
QUERY 17--> LIST THE VISITING DOCTOR WHO TAKES MAXIMUM NUMBER OF APPOINTMENTS ?

MariaDB [hospital]> select first\_name from employee as e inner join visiting as v on e.emp\_id=v.emp\_id order by v.no\_appoinment desc limit 1 ;



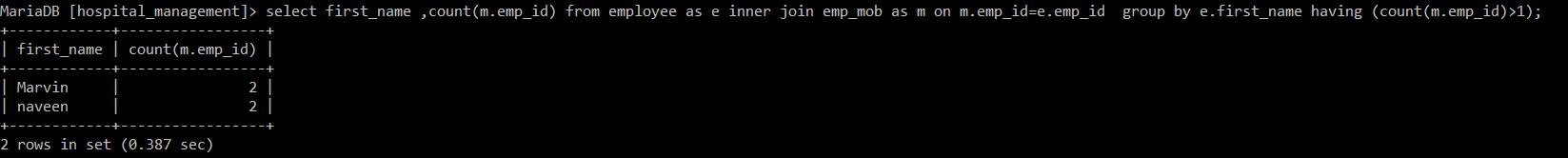
QUERY 18--> FIND THE EMPLOYEES WHOSE FIRST NAME AND PATIENTS FIRST NAME ARE SAME ?

MariaDB [hospitaL]> select employee.first\_name from employee inner join doctor on employee.emp\_id=doctor.emp\_id inner join patient on patient.reg\_id=doctor.reg\_id where (employee.first\_name=patient.first\_name);



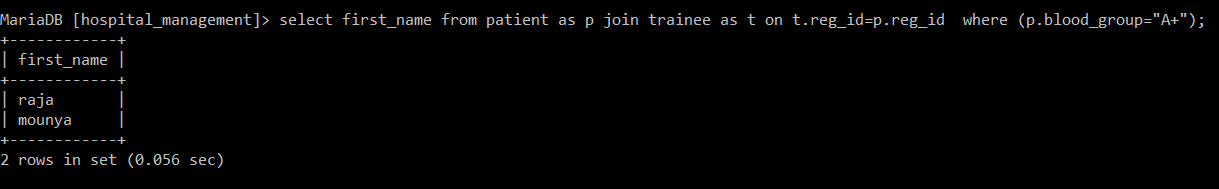
QUERY 19--> LIST THE EMPLOYEES WHO HAVE MORE THAN 1 MOBILE NUMBER ?

MariaDB [hospital]> select first\_name,count(m.emp\_id) from employee as e inner join emp\_mob as m on m.emp\_id=e.emp\_id group by e.first\_name having (count(m.emp\_id)>1);



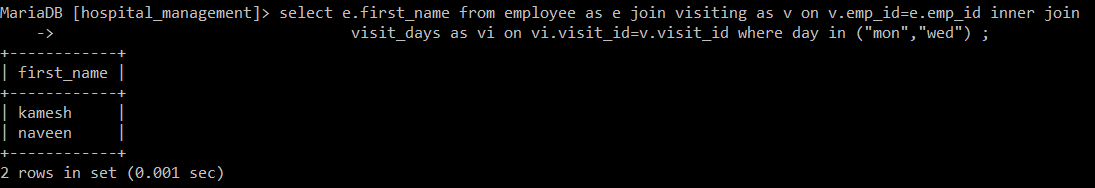
QUERY 20--> LIST THE PATIENT WHO HAS A+VE BLOOD GROUP ?

MariaDB [hospital]> select first\_name from patient as p join trainee as t on t.reg\_id=p.reg\_id where (p.blood\_group="A+");



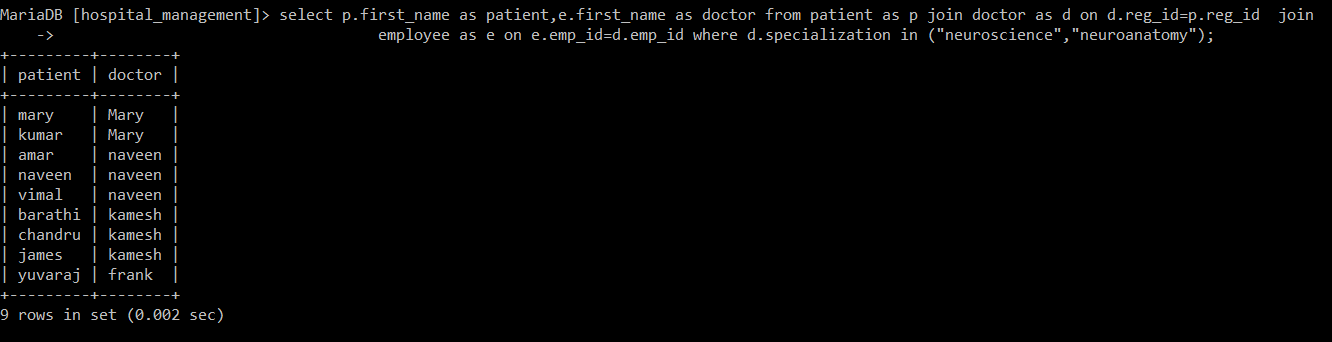
QUERY 21--> LIST THE DOCTOR WHO VISITS THE HOSPITAL ON MONDAY AND WEDNESDAY ?

MariaDB [hospital]> select e.first\_name from employee as e join visiting as v on v.emp\_id=e.emp\_id inner join visit\_days as vi on vi.visit\_id=v.visit\_id where day in ("mon","wed") ;



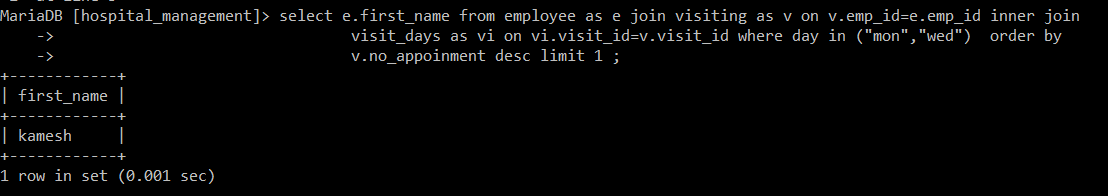
QUERY 22--> LIST THE PATIENT NAME AND DOCTOR TREATING HIM/HER AND HAS HIS SPECIALIZATION IN NEURO SCIENCE OR NEURO ANATOMY ?

MariaDB [hospital]> select p.first\_name as patient,e.first\_name as doctor from patient as p join doctor as d on d.reg\_id=p.reg\_id join employee as e on e.emp\_id=d.emp\_id where d.specialization in ("neuroscience","neuroanatomy");



QUERY 23--> LIST THE DOCTOR WHO VISITS THE HOSPITAL ON MONDAY AND WEDNESDAY AND SEE MAXIMUM NUMBER OF PATIENTS?

MariaDB [hospital]> select e.first\_name from employee as e join visiting as v on v.emp\_id=e.emp\_id inner join visit\_days as vi on vi.visit\_id=v.visit\_id where day in ("mon","wed") order by v.no\_appoinment desc limit 1 ;



QUERIES USING STORED PROCEDURES :

QUERY 24--> WRITE A PROCEDURE TO FIND THE OUT PATIENT WHO HAS A APPOINTMENT WITH DOCTOR ON 1 ST SLOT ?

MariaDB [hospital]> create procedure a1()

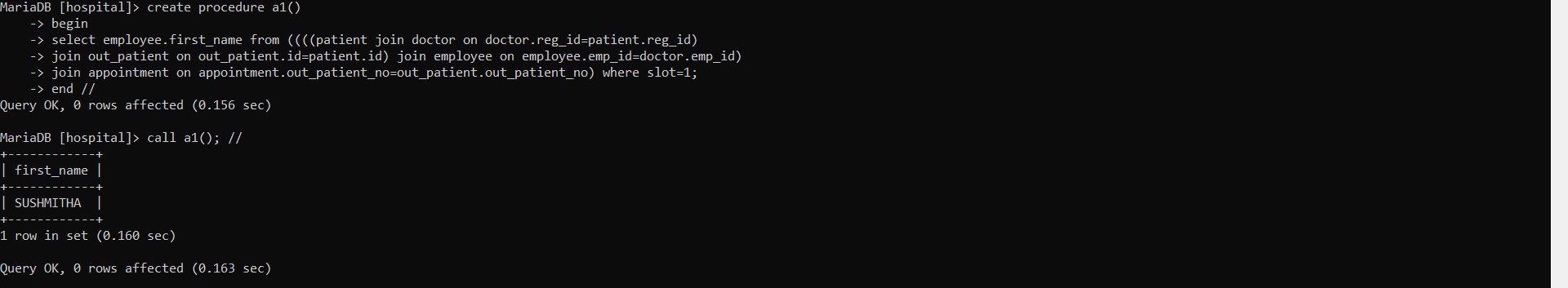
-> begin

-> select employee.first\_name from ((((patient join doctor on doctor.reg\_id=patient.reg\_id)

-> join out\_patient on out\_patient.id=patient.id) join employee on employee.emp\_id=doctor.emp\_id)

-> join appointment on appointment.out\_patient\_no=out\_patient.out\_patient\_no) where slot=1;

-> end //



QUERY 25--> WRITE A PROCEDURE WHICH GETS EMPLOYEE FIRST AS ARGUMENT AND DISPLAYS HIS QUALIFICATION,POSITION HELD AND HIS/HER SALARY ?

MariaDB [hospital]> delimiter //

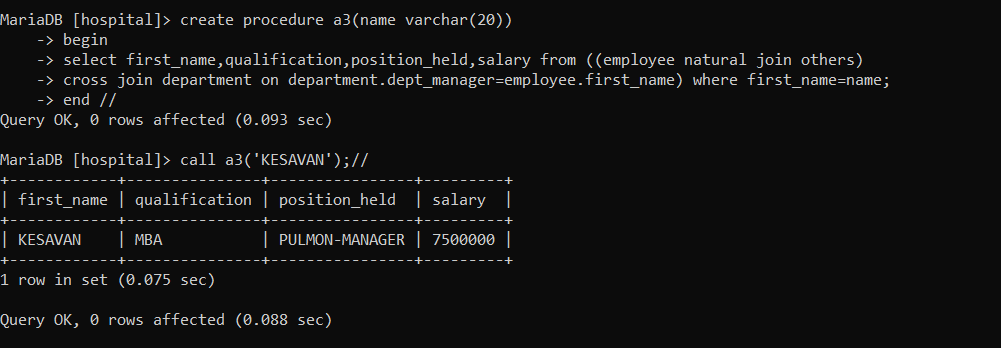
MariaDB [hospital]> create procedure a3(name varchar(20))

-> begin

-> select first\_name,qualification,position\_held,salary from ((employee natural join others)

-> cross join department on department.dept\_manager=employee.first\_name) where first\_name=name;

-> end //



QUERIES USING STORED FUNCTIONS :

QUERY 26--> WRITE A FUNCTION WHISH GETS BP AS ARGUMENT AND DISPLAYS THE PATIENT NAME WHO HAS IT ?

MariaDB [hospital]> delimiter //

MariaDB [hospital]> create function func1(bp varchar(10))

-> returns varchar(30)

-> deterministic

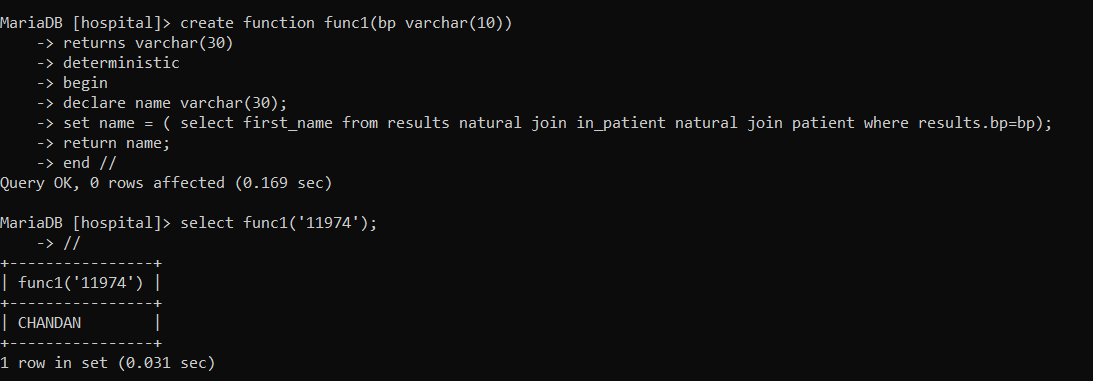
-> begin

-> declare name varchar(30);

-> set name = ( select first\_name from results natural join in\_patient natural join patient where results.bp=bp);

-> return name;

-> end //



QUERY 27--> CREATE A FUNCTION TO FIND WHETHER THE PATIENT HAS HIGH BP OR NOT ?

MariaDB [hospital]> create function bp\_check(bp varchar(10))

-> returns varchar(20)

-> deterministic

-> begin

-> declare result varchar(20);

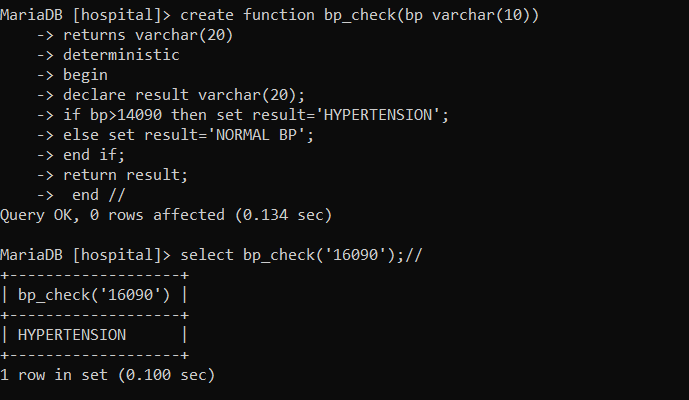
-> if bp>14090 then set result='HYPERTENSION';

-> else set result='NORMAL BP';

-> end if;

-> return result;

-> end //



QUERY 28--> CREATE A FUNCTION TO FIND THE PATIENT WHO STAYED MAXIMUM NUMBER OF DAYS IN THE HOSPITAL ?

MariaDB [hospital]> create function maximum\_days()

-> returns varchar(20)

-> deterministic

-> begin

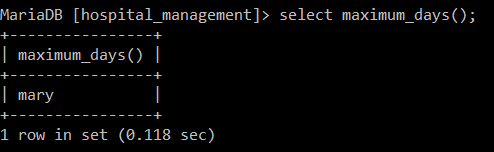
-> declare name varchar(20);

-> set name =(select first\_name from patient as p join in\_patient as i on i.id=p.id join

-> admission as a on a.in\_patient\_no=i.in\_patient\_no order by datediff(a.doa,a.dod) limit 1);

-> return name;

-> end$$



QUERY 29--> CREATE A FUNCTION TO FIND THE PATIENT WHO STAYED MINIMUM NUMBER OF DAYS IN THE HOSPITAL ?

MariaDB [hospital]> create function minimum\_days()

-> returns varchar(20)

-> deterministic

-> begin

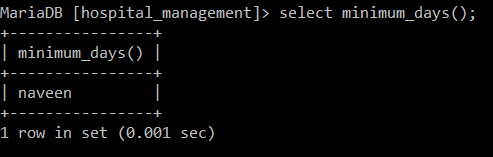
-> declare name varchar(20);

-> set name =(select first\_name from patient as p join in\_patient as i on i.id=p.id join

-> admission as a on a.in\_patient\_no=i.in\_patient\_no order by datediff(a.doa,a.dod) desc limit 1);

-> return name;

-> end$$



QUERIES USING CASE CONSTRUCT :

QUERY 30--> WRITE A QUERY USING CASE STATEMENT TO GRADE EMPLOYEES BASED ON EXPERIENCE ?

MariaDB [hospital]> select emp\_id,first\_name,salary,

-> case

-> when experience<6 then 'LESS EXPERIENCED'

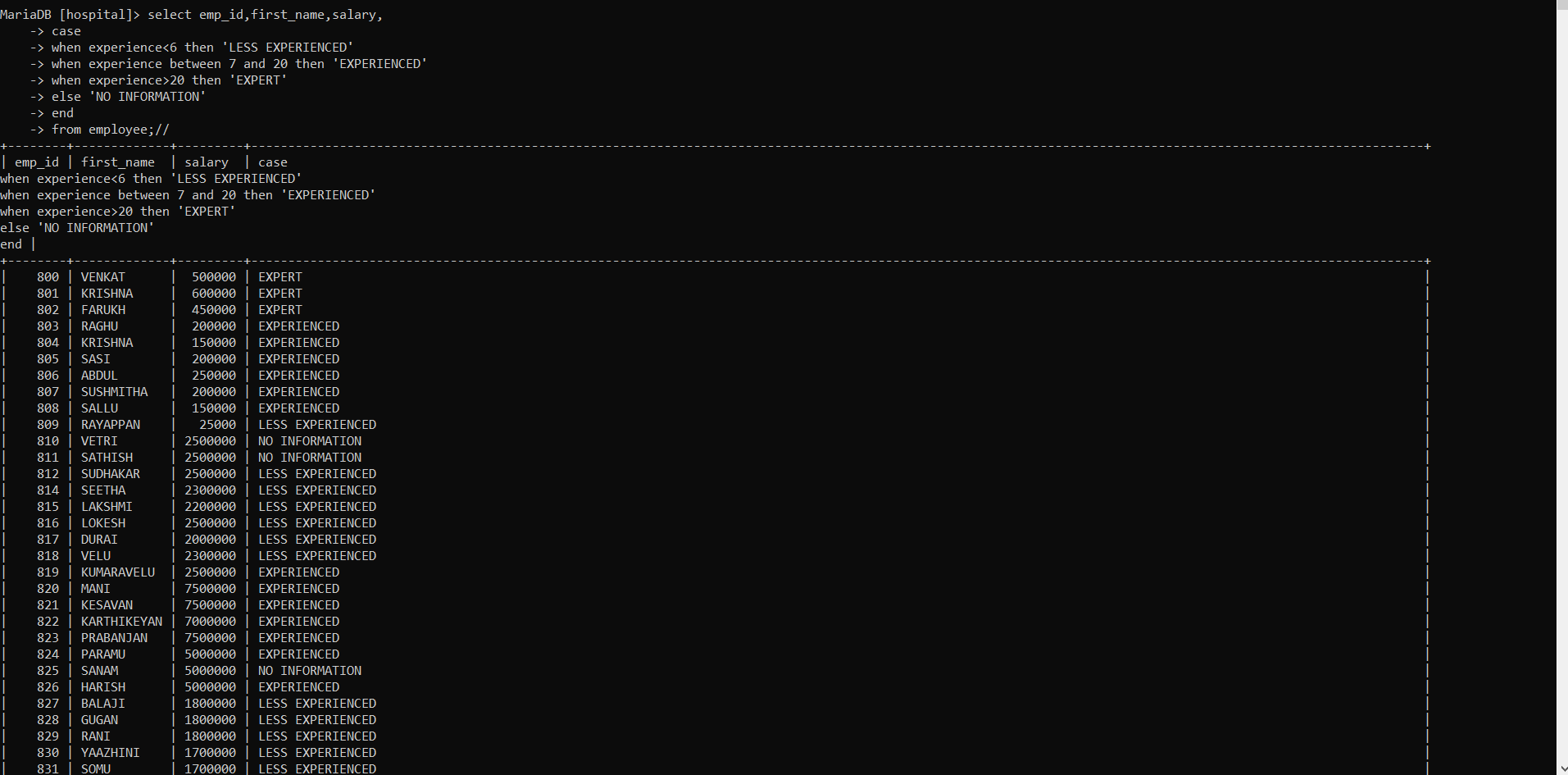
-> when experience between 7 and 20 then 'EXPERIENCED'

-> when experience>20 then 'EXPERT'

-> else 'NO INFORMATION'

-> end

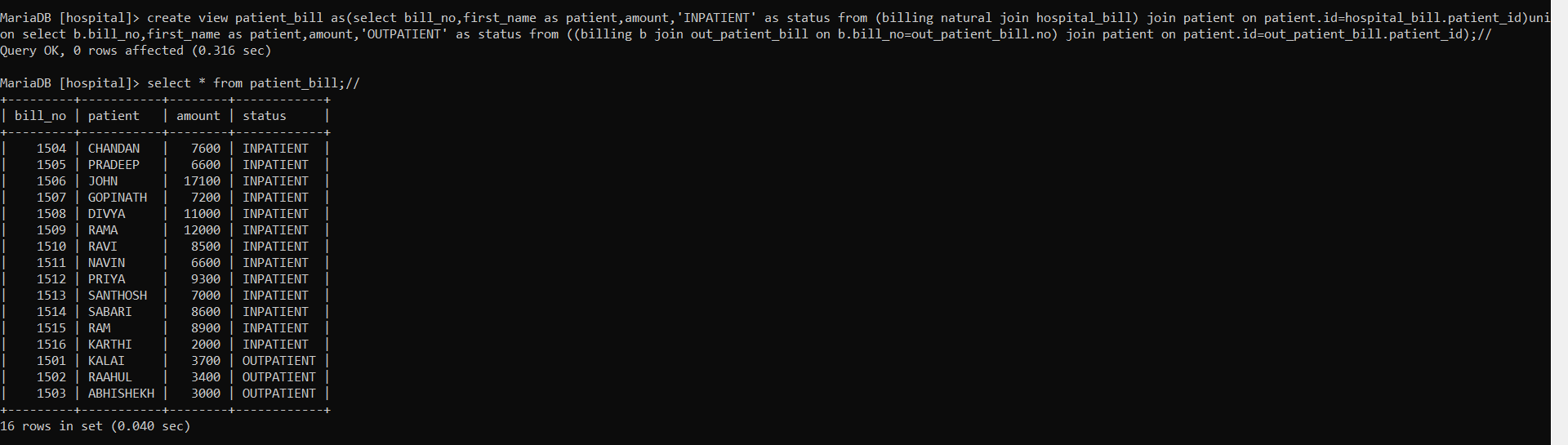
-> from employee;//



QUERY USING VIEW STATEMENT :

QUERY 31--> CREATE A VIEW FOR A EMPLOYEE IN BILL COUNTER.THE TABLE SHOULD CONTAIN BILL NO,PATIENT NAME,AMOUNT TO BE PAIED AND STATUS(INPATIENT/OUTPATIENT) ?

MariaDB [hospital]> create view patient\_bill as(select bill\_no,first\_name as patient,amount,'INPATIENT' as status from (billing natural join hospital\_bill) join patient on patient.id=hospital\_bill.patient\_id)union select b.bill\_no,first\_name as patient,amount,'OUTPATIENT' as status from ((billing b join out\_patient\_bill on b.bill\_no=out\_patient\_bill.no) join patient on patient.id=out\_patient\_bill.patient\_id);//



QUERY USING CURSORS :

QUERY 32--> CREATE A CURSOR TO FIND NUMBER OF PATIENTS HAVING O+VE BLOOD GROUP ?

MariaDB [hospital]> create procedure tot\_patient()

-> begin

-> declare a int;

-> declare pat\_id int;

-> declare done int default false;

-> declare cur cursor for select id from patient where blood\_group='O+VE';

-> declare continue handler for not found set done=true;

-> set a=0;

-> open cur;

-> repeat\_loop: loop

-> fetch cur into pat\_id;

-> if done then leave repeat\_loop;

-> end if;

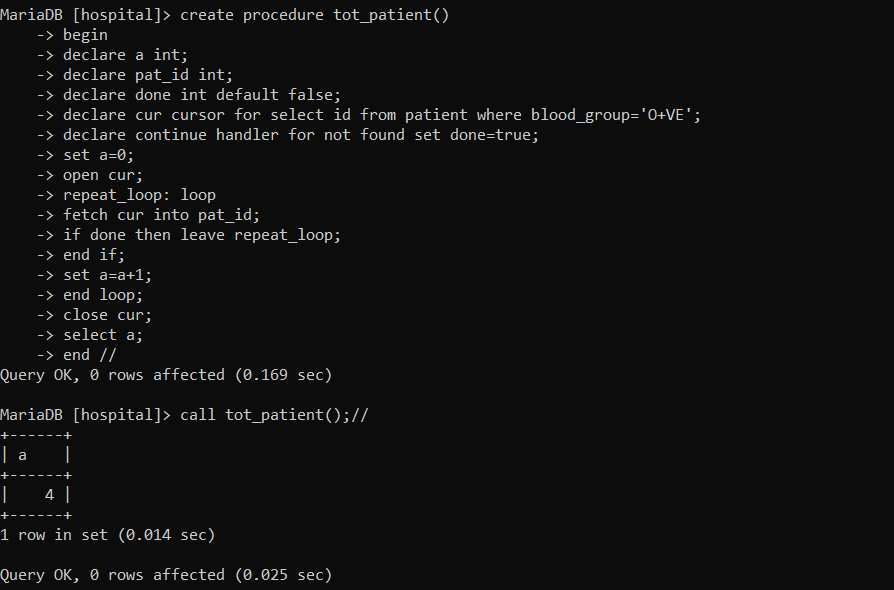
-> set a=a+1;

-> end loop;

-> close cur;

-> select a as no\_of o+ve;

-> end //



QUERY 33--> CREATE A CURSOR TO FIND NUMBER OF PATIENTS BORN BETWEEN 1990 AND 2000 AND LIVING IN EDISON STREET OR NELSON STREET ?

MariaDB [hospital]>create procedure get\_count()

-> begin

-> declare total int;

-> declare first\_name varchar(20);

-> declare sage int;

-> declare done int default 0;

-> declare cur cursor for select first\_name from patient where (year(dob) between 1990 and 2000) and street in ("edison street","nelson street");

-> declare continue handler for not found set done=1;

-> set total=0;

-> open cur;

-> read\_loop:loop

-> fetch cur into first\_name;

-> if done=1 then leave read\_loop;

-> end if;

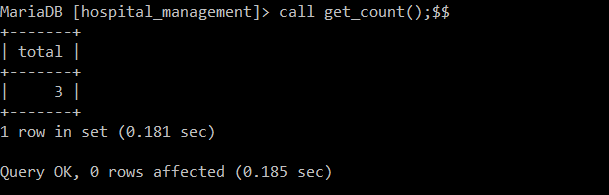
-> set total=total+1;

-> end loop;

-> close cur;

-> select total;

-> end $$



QUERIES USING TRIGGER :

QUERY 34--> CREATE A TRIGGER SUCH THAT ON INSERTION OF A NEW RECORD ON IN PATIENT TABLE AUTOMATICALLY GENERATE AND INSERT THE RECORD IN ADMISSION AND IN IN PATIENT BILLING TABLE.

ALSO UPDATE THE NUMBER OF OCCUPIED BEDS IN THE WARD TABLE .

MariaDB [hospital]> create trigger tg1 after insert on in\_patient for each row

-> begin

-> declare inpatient\_no int;

-> declare wardno int;

-> declare a int;

-> declare b int;

-> declare c int;

-> declare patid int;

-> set a = 6313 ;

-> set b = 1516;

-> set c =13;

-> set inpatient\_no=new.in\_patient\_no;

-> set patid= new.id;

-> set wardno=new.ward\_no;

-> insert into admission values(a,inpatient\_no,CURDATE(),CURTIME(),DATE\_ADD(CURDATE(),INTERVAL 2 DAY),'08:00:00');

-> set a=a+1;

-> update ward set occupied\_beds=occupied\_beds+1 where ward\_no=wardno;

-> insert into billing values(b,2348,DATE\_ADD(CURDATE(),INTERVAL 2 DAY),'05:00:00','2000.00');

-> insert into hospital\_bill values(c,b,patid,2,400,300,400,100,500,400);

-> set b=b+1;

-> set c=c+1;

-> end//



QUERY 35--> CREATE A TRIGGER SUCH THAT ON UPDATING A RECORD ON CASE HISTORY TABLE STORE THE OLD CONTENTS TO A BACKUP TABLE.

MariaDB [hospital]> create trigger tg2 after update on case\_history for each row

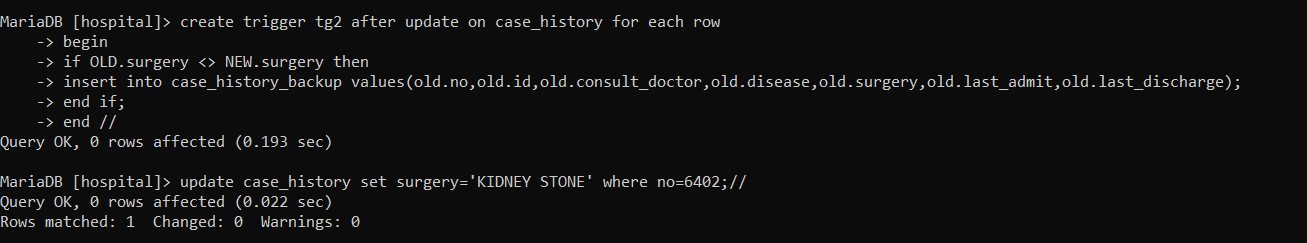
-> begin

-> if OLD.surgery <> NEW.surgery then

-> insert into case\_history\_backup values(old.no,old.id,old.consult\_doctor,old.disease,old.surgery,old.last\_admit,old.last\_discharge);

-> end if;

-> end //



QUERY 36--> CREATE A TRIGGER SUCH THAT ON DELETING A RECORD FROM EMPLOYEE TABLE AUTOMATICALLY BACKUP THE RECORD TO A EMPLOYEE\_BACKUP TABLE.

MariaDB [hospital]> create trigger tg3 before delete on employee for each row

-> begin

-> declare empid int;

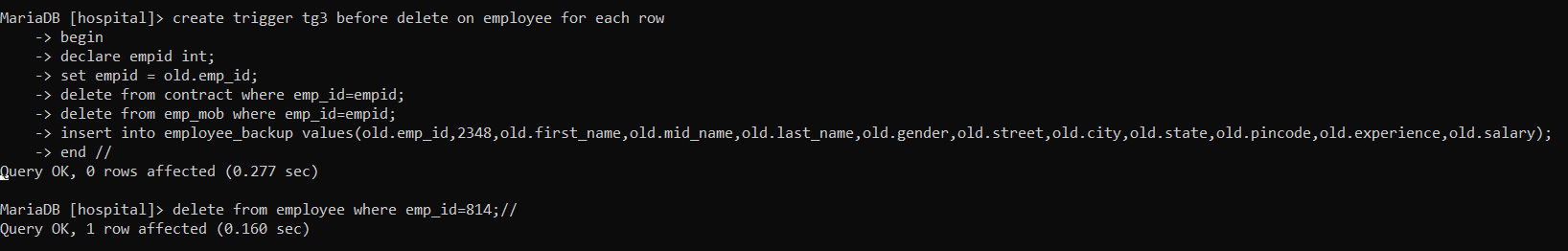
-> set empid = old.emp\_id;

-> delete from contract where emp\_id=empid;

-> delete from emp\_mob where emp\_id=empid;

-> insert into employee\_backup values(old.emp\_id,2348,old.first\_name,old.mid\_name,old.last\_name,old.gender,old.street,old.city,old.state,old.pincode,old.experience,old.salary);

-> end //

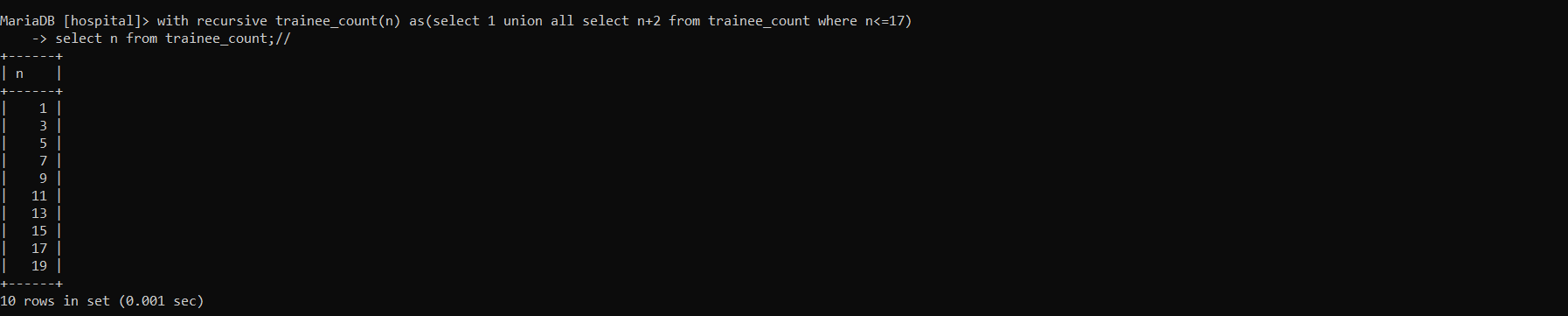


QUERIES USING RECURSSION :

QUERY 37--> CONSIDER A DOCTOR HAS 0 TRAINEES.IF AT DAY1, 1 TRAINEE JOINS HIM.AT DAY2, 2 TRAINEES JOINS AND AT DAY3, 2 TRAINEES JOIN AND SO ON ..

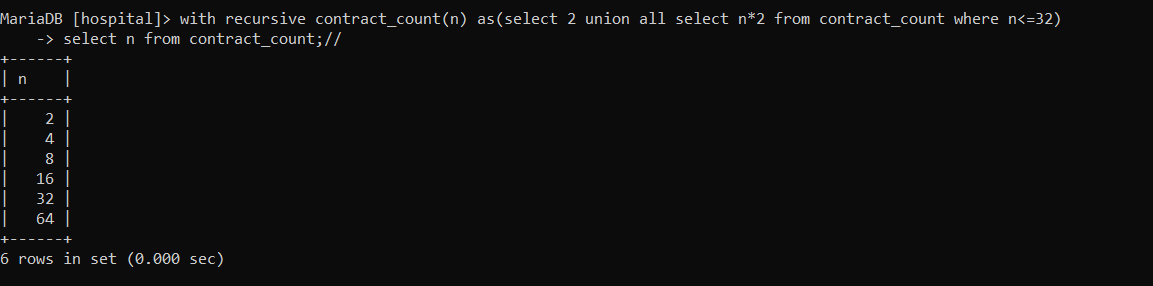
WRITE A RECURSIVE QUERY TO CALCULATE NUMBER OF TRAINEES AT THE END OF 10 DAYS.

MariaDB [hospital]> with recursive trainee\_count(n) as(select 1 union all select n+2 from trainee\_count where n<=17) select n from trainee\_count;//



QUERY 38--> SUPPOSE IN A HOSPITAL THE CONTRACT STAFFS JOIN IN THE MULTIPLES OF 2.WRITE A RECURSIVE QUERY TO CALCULATE NUMBER OF CONTRACT STAFFS AT THE END OF 6TH DAY.

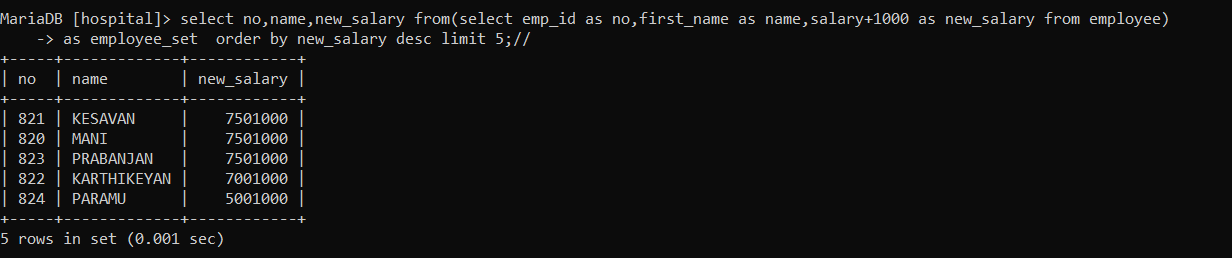
MariaDB [hospital]> with recursive contract\_count(n) as(select 2 union all select n\*2 from contract\_count where n<=32) select n from contract\_count;//



QUERIES USING DERIVED TABLES :

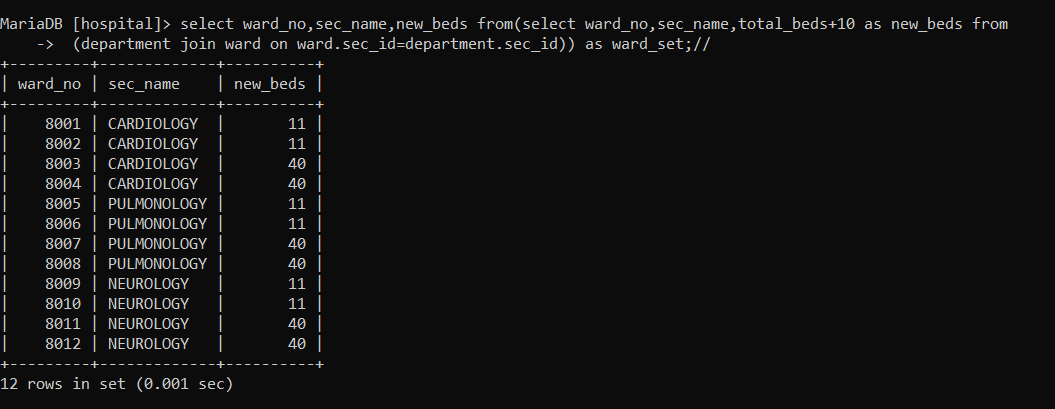
QUERY 39--> INCREASE THE SALARY OF EMPLOYEES BY 1000 BY USING THE CONCEPT OF DERIVED TABLES.LIST THE TOP 5 EMPLOYEES WHO GETS THE HIGHEST SALARY.

MariaDB [hospital]> select no,name,new\_salary from(select emp\_id as no,first\_name as name,salary+1000 as new\_salary from employee) as employee\_set order by new\_salary desc limit 5;//



QUERY 40--> CONSIDER THE HOSPITAL IS RENOVATED.SO, INCREASE THE NUMBER OF BEDS IN EACH WARD BY 10 BY USING THE CONCEPT OF DERIVED TABLES.

MariaDB [hospital]> select ward\_no,sec\_name,new\_beds from(select ward\_no,sec\_name,total\_beds+10 as new\_beds from (department join ward on ward.sec\_id=department.sec\_id)) as ward\_set;//



CORRELATED QUERIES :

QUERY 41--> LIST THE DETAILS OF EMPLOYEES WHO GETS SALARY GREATER THAN THE AVERAGE SALARY OF HIS/HER DEPARTMENT ?

MariaDB [hospital]> select \* from employee o where salary > (select avg(salary) from employee i where i.sec\_id=o.sec\_id);

