CSLR 51 : DBMS LAB-1

**106119100**, Rajneesh Pandey, CSE-B

**Problem 1.** Consider the Following Database:

A software company wants to track project details

Employee(Empid , Empname, Address, Doj, Salary) : Empid as Primary key

Project (Projectno, Duration, Projectname) : Project no as Primary Key

Workson(Empid,Projno) :

Empid as Foreign key references Employee

Projectno as Foreign key references Project

**ER Diagram**

Text

Description automatically generated

CREATE DATABASE companyDB;

1. Display the Employee details in the descending order based on name.

CREATE TABLE Employee(

  EmpID varchar(255) NOT NULL,

  EmpName varchar(255),

  Address varchar(255),

  Doj varchar(255),

  Salary int,

  UNIQUE (EmpID),

  PRIMARY KEY (EmpID)

  );

INSERT INTO Employee(EmpID, EmpName, Address, Doj, Salary)

VALUES

 ('001','Harry','3427 Hall Valley Drive','12/28/1984',59000),

 ('002','Wilson','3950 Rinehart Road','5/18/1983',42000),

 ('003','Mildred','4768 Scenicview Drive','6/21/1969',78000),

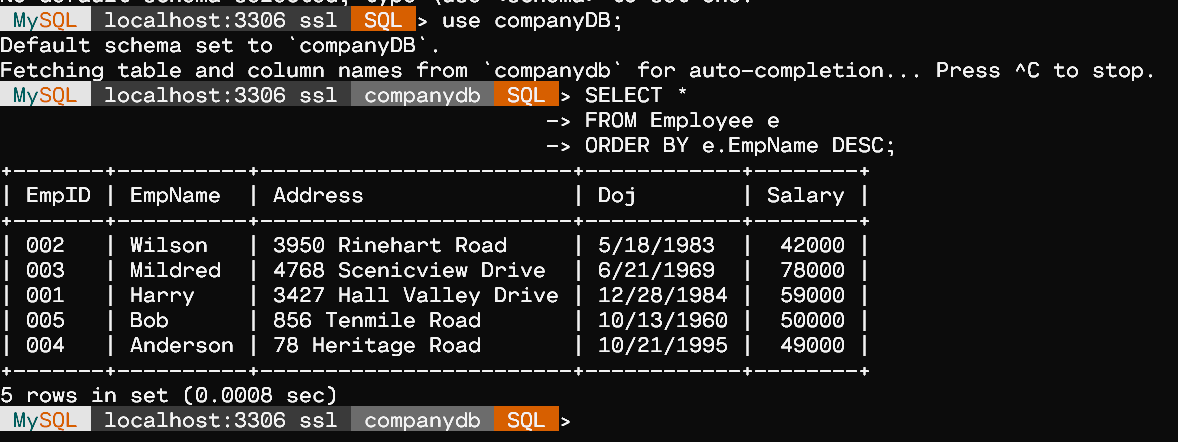
 ('004','Anderson','78 Heritage Road','10/21/1995',49000),

 ('005','Bob','856 Tenmile Road','10/13/1960',50000);

SELECT \*

FROM Employee e

ORDER BY e.EmpName DESC;



2. Display the project details if project id is given.

CREATE TABLE Project(

  Projectno varchar(255) NOT NULL,

  Duration int,

  Projectname varchar(255),

  UNIQUE (Projectno),

  PRIMARY KEY (Projectno)

  );

INSERT INTO Project(Projectno,Duration, Projectname)

VALUES

 ('P1',5,'WebSite'),

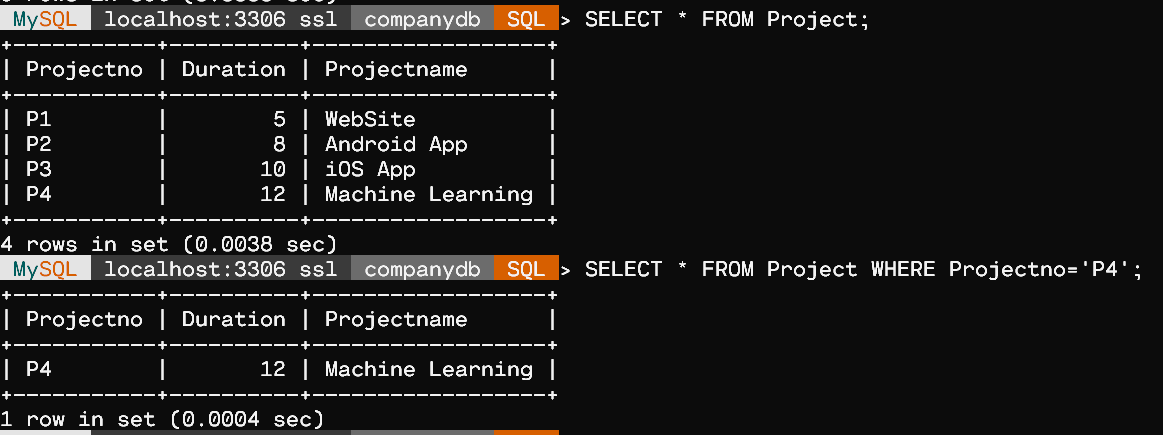
 ('P2',8,'Android App'),

 ('P3',10,'iOS App'),

 ('P4',12,'Machine Learning');

SELECT \* FROM Project;

SELECT \* FROM Project WHERE Projectno='P4';

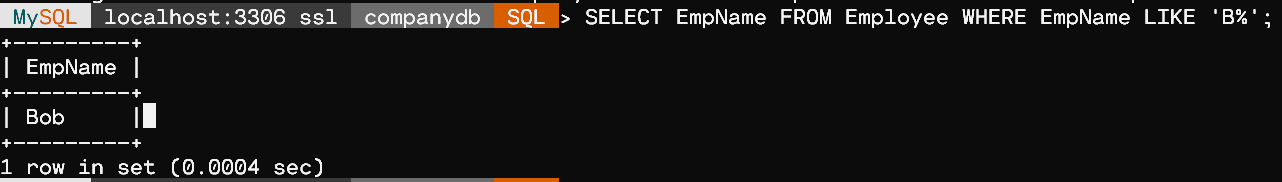


3. Display the employee names starting with ‘B'

SELECT EmpName

from Employee

where EmpName LIKE 'B%';



 4. Display the employee ID's working in a particular project if project no is given.

CREATE TABLE Workson(

  Empid varchar(255),

  Projectno  varchar(255),

  FOREIGN KEY (Empid) REFERENCES Employee(Empid),

  FOREIGN KEY (Projectno) REFERENCES Project(Projectno)

  );

INSERT INTO Workson(Projectno,Empid)

VALUES

  ('P3','002'),

  ('P2','004'),

  ('P1','003'),

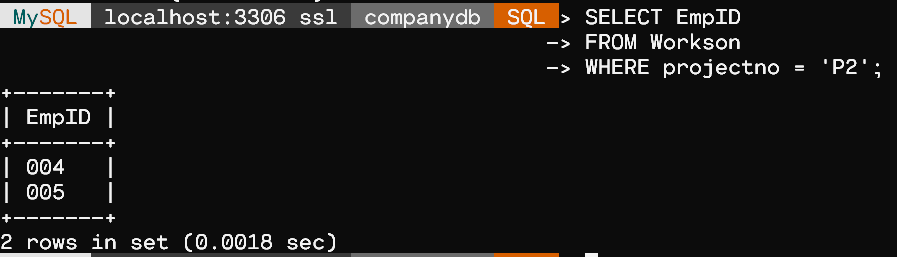
  ('P4','001'),

  ('P2','005');

SELECT EmpID

FROM Workson

WHERE projectno = 'P2';



**Problem2.** Consider the Following Database:

Student(Rollno, Name, Marks(of 6 subjects),total) : Rollno as Primary key

Department(Deptid, Deptname, HOD name) and Deptid as Primary key

StudDep(Rollno, Deptid).

Rollno as foreign key references Student

Deptid as foreign key references Department

The total field is updated automatically

**ER Diagram:**

Diagram

Description automatically generated

CREATE DATABASE collegeDB;

CREATE TABLE Student(

    Rollno int NOT NULL,

    StudName varchar(255) NOT NULL,

    sub1 int,

    sub2 int,

    sub3 int,

    sub4 int,

    sub5 int,

    sub6 int,

    Total int,

    UNIQUE (Rollno),

    PRIMARY KEY(Rollno)

);

CREATE TABLE Department(

    DeptID int NOT NULL,

    Deptname varchar(255),

    HODname varchar(255),

    UNIQUE (DeptID),

    PRIMARY KEY (DeptID)

);

CREATE TABLE StudDep(

    Rollno int,

    DeptID int,

    FOREIGN KEY (Rollno) REFERENCES Student(Rollno),

    FOREIGN KEY (DeptID) REFERENCES Department(DeptID)

);

1.Insert 10 student details and 3 department details. Insert details in the studdep table.

INSERT INTO Student(Rollno,StudName, sub1, sub2, sub3, sub4, sub5, sub6)

VALUES

 (1,'Amar',70,80,90,80,100,90),

 (2,'Shivam',80,90,80,100,90,70),

 (3,'Radha',80,100,90,70,70,40),

 (4,'Nitin',40,60,50,80,60,70),

 (5,'Ritik',50,50,60,100,90,70),

 (6,'Vaibhav',100,100,100,100,100,90),

 (7,'Kartik',10,4,50,80,100,90),

 (8,'Ram',80,90,80,100,90,70),

 (9,'Tom',90,80,100,90,50,90),

 (10,'Katy',90,80,100,90,100,100);

SET SQL\_SAFE\_UPDATES = 0;

UPDATE Student SET Total = sub1 + sub2 + sub3 + sub4 + sub5 + sub6;

INSERT INTO Department(DeptID,Deptname,HODname)

VALUES

 (1,'CSE','Dinesh'),

 (2,'Mech','Lakshmi'),

 (3,'ECE','Surya');

INSERT INTO StudDep(Rollno,DeptID)

VALUES

 (1,1),

 (2,1),

 (3,3),

 (4,2),

 (5,1),

 (6,2),

 (7,1),

 (8,3),

 (9,3),

 (10,2);

2.Display the Student details if deptid is given.

SELECT \*

FROM Student

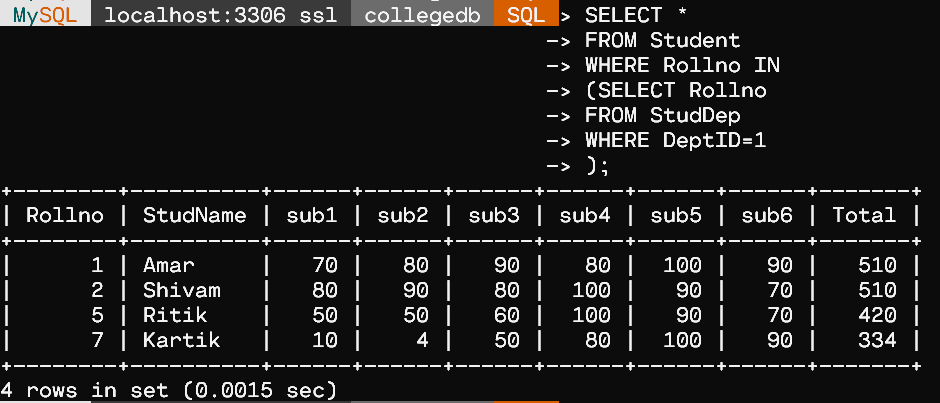
WHERE Rollno IN

(SELECT Rollno

FROM StudDep

WHERE DeptID=1

);



 3.Display the department details if rollno is given

SELECT \*

FROM Department

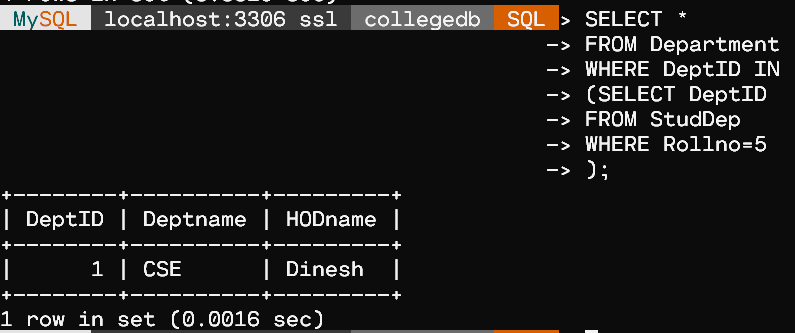
WHERE DeptID IN

(SELECT DeptID

FROM StudDep

WHERE Rollno=5

);



4.Display the student names who got total greater than 500

SELECT StudName

FROM Student

WHERE Total>500;

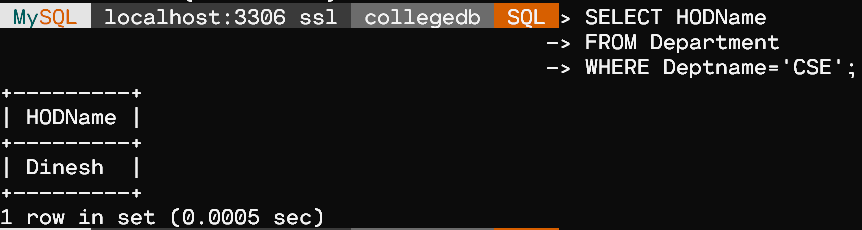


5.Display the HOD name of the CSE department

SELECT HODName

FROM Department

WHERE Deptname='CSE';



 6.Display the student rollnos of the CSE department

SELECT Rollno

FROM StudDep

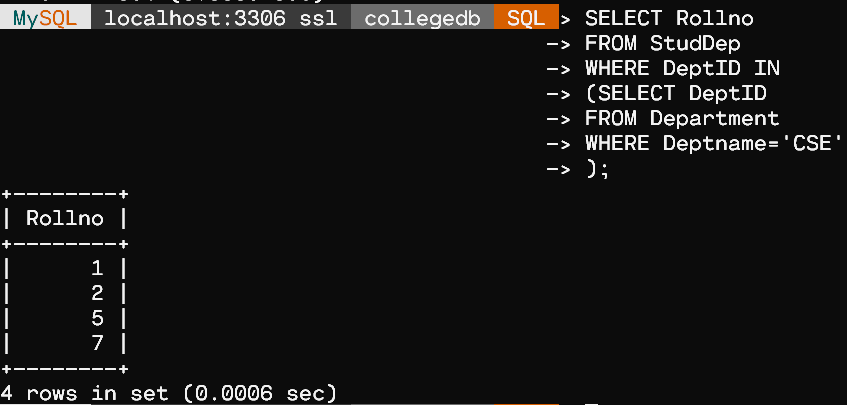
WHERE DeptID IN

(SELECT DeptID

FROM Department

WHERE Deptname='CSE'

);



**Problem 3**.Consider the Following Database:

salesperson(ssn, name, start\_year, dept\_no)

ssn – Primary Key

trip(ssn, from\_city, to\_city, departure\_date, return\_date, trip\_id))

ssn – Foreign key

trip\_id – Primary key

salerep\_expense(trip\_id, expense\_type,amount)

trip\_id – Foreign key

The expense types are ‘TRAVEL’, ‘STAY’ and ‘FOOD’

**ER DIAGRAM**

**Graphical user interface, diagram

Description automatically generated**

 CREATE DATABASE vacationDB;

CREATE TABLE salesperson(

  ssn int NOT NULL,

  personName varchar(255),

  start\_year varchar(255),

  dept\_no int,

  UNIQUE (ssn),

  PRIMARY KEY (ssn)

  );

CREATE TABLE trip(

  trip\_id int NOT NULL,

  ssn int,

  from\_city varchar(255),

  to\_city varchar(255),

  departure\_date varchar(255),

  return\_date varchar(255),

  UNIQUE (trip\_id),

  PRIMARY KEY (trip\_id),

  FOREIGN KEY (ssn) REFERENCES salesperson(ssn)

  );

CREATE TABLE salerep\_expense(

  trip\_id int,

  expense\_type  varchar(255),

/\* The expense types are ‘TRAVEL’, ‘STAY’ and ‘FOOD’ \*/

  amount int,

  FOREIGN KEY (trip\_id) REFERENCES trip(trip\_id)

  );

INSERT INTO salesperson(ssn,personName,start\_year,dept\_no)

VALUES

 (1000,'Harry','12/28/1984',100),

 (2000,'Wilson','5/18/1983',200),

 (3000,'Mildred','6/21/1969',300),

 (4000,'Anderson','10/21/1995',400),

 (5000,'Bob','10/13/1960',500);

INSERT INTO trip(trip\_id,from\_city,to\_city,departure\_date,return\_date,ssn)

VALUES

 (1,'Mandu','Sanchi','2021-09-13','2021-10-01',2000),

 (2,'Khajuraho','Chennai','2021-08-24','2021-09-27',1000),

 (3,'Ujjain','Bhopal','2021-11-01','2021-12-20',5000),

 (4,'Orchha','Indore','2021-12-27','2022-01-15',3000),

 (5,'Pachmarhi','Chennai','2021-11-23','2021-12-20',4000),

 (6,'Jammu','Chennai','2021-11-20','2021-12-20',4000);

INSERT INTO salerep\_expense(trip\_id,expense\_type,amount)

VALUES

 (1,'TRAVEL',2000),

 (2,'TRAVEL',1500),

 (3,'TRAVEL',2100),

 (4,'TRAVEL',1300),

 (5,'TRAVEL',1800),

 (6,'TRAVEL',1000),

 (1,'STAY',500),

 (2,'STAY',600),

 (4,'STAY',620),

 (6,'STAY',600),

 (1,'FOOD',800),

 (2,'FOOD',100),

 (3,'FOOD',700),

 (4,'FOOD',600);

1.Give the details(all attributes of trip relation)for trips that exceed Rs2000

    SELECT trip\_id,from\_city,to\_city,departure\_date,return\_date

    FROM

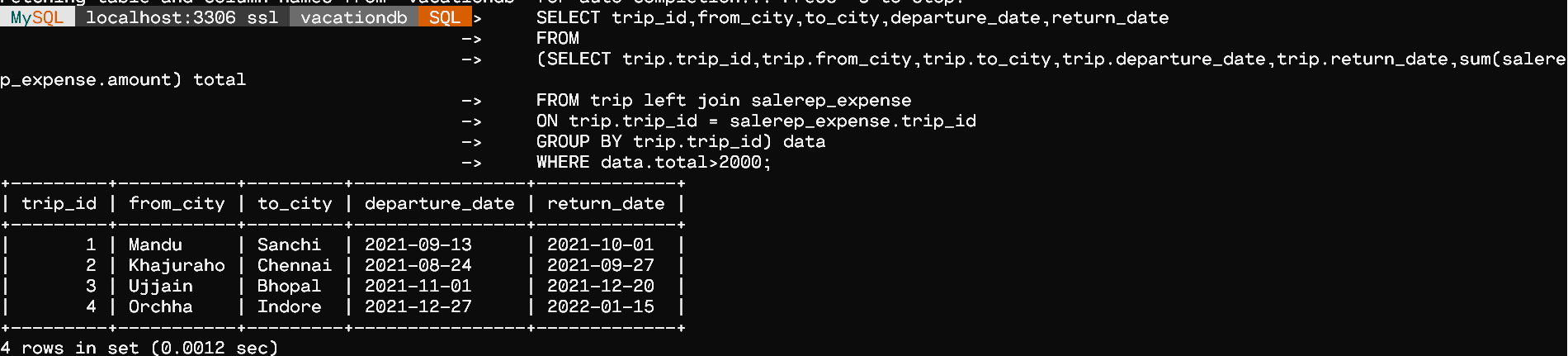
    (SELECT trip.trip\_id,trip.from\_city,trip.to\_city,trip.departure\_date,trip.return\_date,sum(salerep\_expense.amount) total

    FROM trip left join salerep\_expense

    ON trip.trip\_id = salerep\_expense.trip\_id

    GROUP BY trip.trip\_id) data

    WHERE data.total>2000;



2.Print the ssn of salesperson who took trips to chennai more than once

    SELECT ssn

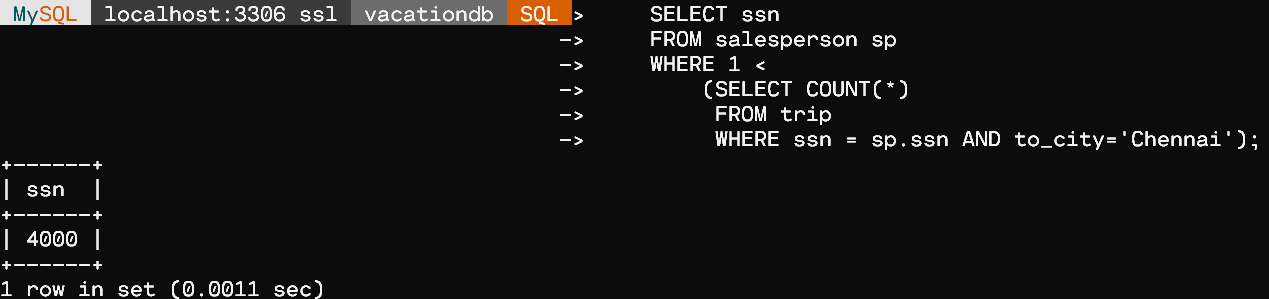
    FROM salesperson sp

    WHERE 1 <

        (SELECT COUNT(\*)

         FROM trip

         WHERE ssn = sp.ssn AND to\_city='Chennai');



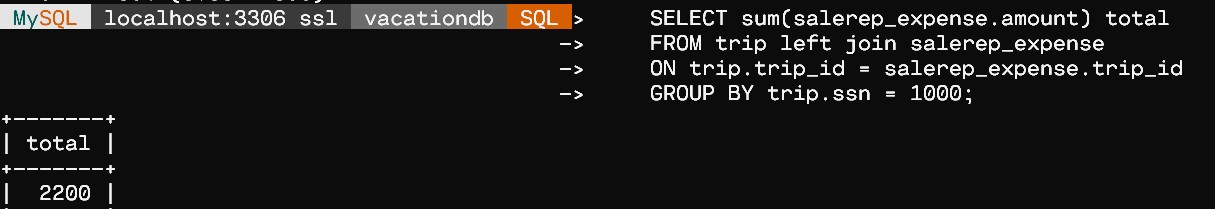
3.Print the total trip expenses incurred by the salesperson with ssn = 1000

    SELECT sum(salerep\_expense.amount) total

    FROM trip left join salerep\_expense

    ON trip.trip\_id = salerep\_expense.trip\_id

    GROUP BY trip.ssn = 1000;

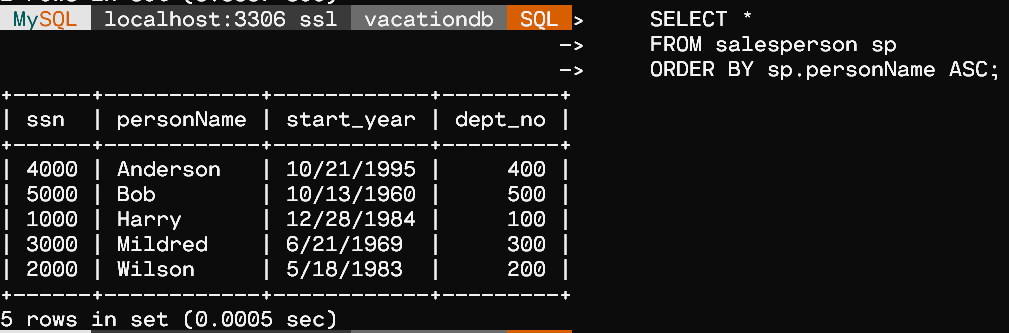


4.Display the salesperson details in the sorted order based on name

    SELECT \*

    FROM salesperson sp

    ORDER BY sp.personName ASC;



**Problem 4.** Consider the Following Database:

car(serial\_no, model, manufacturer, price)

serial\_no – Primary key

options(serial\_no, option\_name, price)

serial\_no – Foreign key

sales(salesperson\_id, serial\_no, date, sale\_price)

serial\_no – Foreign key

salesperson\_id – Foreign key

salesperson(salesperson\_id, name, phone)

salesperson\_id – Primary key

**ER DIAGRAM**

Graphical user interface, diagram

Description automatically generated

CREATE DATABASE carDB;

CREATE TABLE Car(

    serial\_no int NOT NULL,

    model varchar(255) NOT NULL,

    manufacturer varchar(255) NOT NULL,

    price int NOT NULL,

    UNIQUE (serial\_no),

    PRIMARY KEY(serial\_no)

);

CREATE TABLE salesperson(

    salesperson\_id int NOT NULL,

    spname varchar(255) NOT NULL,

    phone DECIMAL(12) NOT NULL,

    UNIQUE (salesperson\_id),

    PRIMARY KEY(salesperson\_id)

);

CREATE TABLE car\_options(

    serial\_no int,

    option\_name varchar(255) NOT NULL,

    price int NOT NULL,

    FOREIGN KEY (serial\_no) REFERENCES Car(serial\_no)

);

CREATE TABLE sales(

    salesperson\_id int,

    serial\_no int,

    sale\_date DATE,

    sale\_price int,

    FOREIGN KEY (serial\_no) REFERENCES Car(serial\_no),

    FOREIGN KEY (salesperson\_id) REFERENCES salesperson(salesperson\_id)

);

INSERT INTO Car(serial\_no,model,manufacturer,price)

VALUES

 (1,'Swift','Suzuki',700000),

 (2,'City','Honda',900000),

 (3,'Nano','Tata',500000),

 (4,'Fortuner','Toyota',1000000);

INSERT INTO salesperson(salesperson\_id,spname,phone)

VALUES

 (1,'John',8168915356),

 (2,'Tom',9368570708),

 (3,'Martin',7895247308);

INSERT INTO car\_options(serial\_no,option\_name,price)

VALUES

 (1,'Black',850000),

 (1,'Blue',760000),

 (1,'White',900000),

 (2,'Black',850000),

 (2,'Blue',760000),

 (4,'Blue',760000),

 (4,'White',900000);

INSERT INTO sales(salesperson\_id,serial\_no,sale\_date,sale\_price)

VALUES

 (2,1,'2021-01-15',850000),

 (1,2,'2021-04-07',750000),

 (3,3,'2021-03-23',600000),

 (1,4,'2021-03-12',900000);

1. For the sales person named ‘John’ list the following information for all the cars sold :

                serial no, manufacturer, sale\_price

    SELECT \*

    FROM Car

    WHERE serial\_no IN(

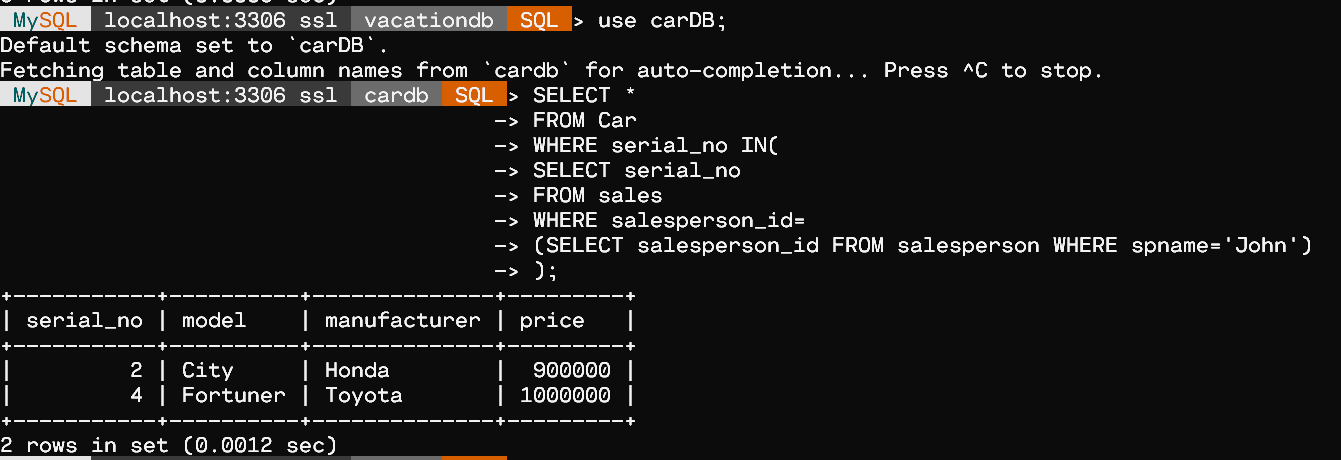
        SELECT serial\_no

        FROM sales

        WHERE salesperson\_id=

            (SELECT salesperson\_id FROM salesperson WHERE spname='John')

    );



2.List the serial\_no and model of cars that have no options

    SELECT serial\_no,model

    FROM

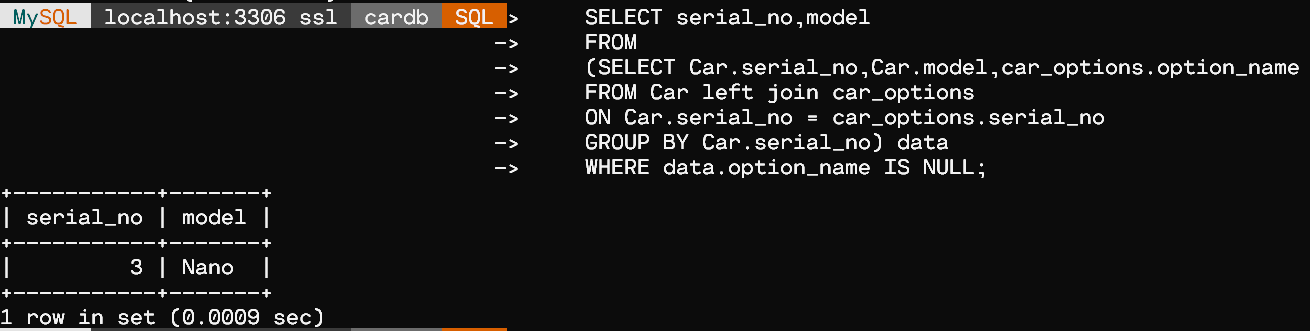
    (SELECT Car.serial\_no,Car.model,car\_options.option\_name

    FROM Car left join car\_options

    ON Car.serial\_no = car\_options.serial\_no

    GROUP BY Car.serial\_no) data

    WHERE data.option\_name IS NULL;



3.List the serial\_no, model, sale\_price for the cars that have optional parts.

    SELECT serial\_no,model,SP

    FROM

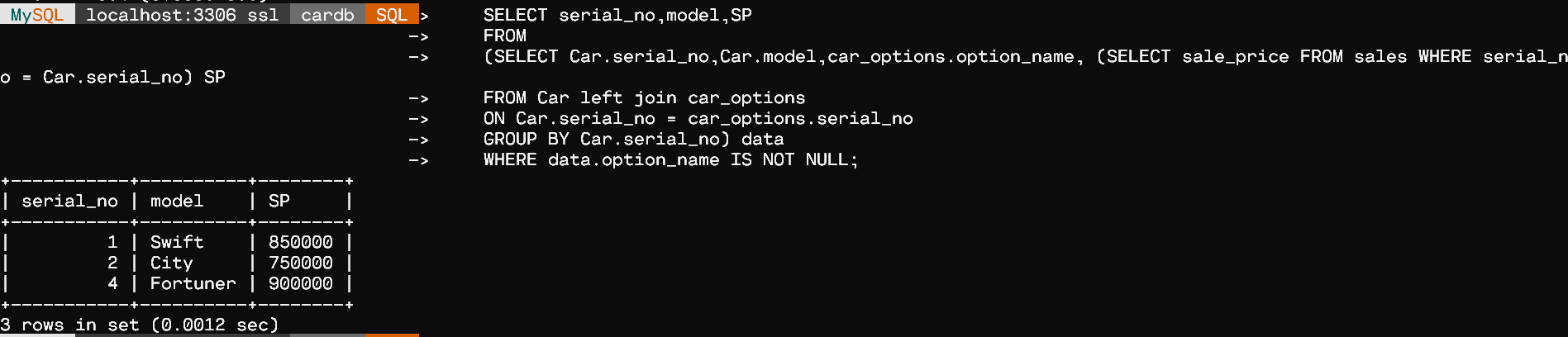
    (SELECT Car.serial\_no,Car.model,car\_options.option\_name, (SELECT sale\_price FROM sales WHERE serial\_no = Car.serial\_no) SP

    FROM Car left join car\_options

    ON Car.serial\_no = car\_options.serial\_no

    GROUP BY Car.serial\_no) data

    WHERE data.option\_name IS NOT NULL;



4.Modify the phone no of a particular sales person

    UPDATE salesperson

    SET phone = 8941999954

    WHERE spname='Tom';



    SELECT \* FROM salesperson;

