DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
(ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

II B.Tech II Semester

Subject Name: DATABASE MANAGEMENT SYSTEMS LAB

Subject Code: C0519
Regulations: MR-22

Lab Manual

Academic Year: 2024-25

MALLA REDDY ENGINEERING COLLEGE (AUTONOMOUS)
MAIN CAMPUS
(An UGC Autonomous Institution, Approved by AICTE and Affiliated to JNTUH, Hyderabad, Accredited by NAAC with 'A++' Grade (III Cycle))
NBA Accredited Programmes – UG (CE, EEE, ME, ECE, & CSE), PG (CE-SE, EEE, EPS, ME-TE)
Maisammaguda(H), Gundlapochampally Village, Medchal Mandal,
Medchal-Malkajgiri District, Telangana State – 500100

MALLA REDDY ENGINEERING COLLEGE (AUTONOMOUS)

MR22 – ACADEMIC REGULATIONS (CBCS)

for B.Tech. (REGULAR) DEGREE PROGRAMME

Applicable for the students of B.Tech. (Regular) programme admitted from the Academic Year 2022-23 onwards

The B.Tech. Degree of Jawaharlal Nehru Technological University Hyderabad, Hyderabad shall be conferred on candidates who are admitted to the programme and who fulfill all the requirements for the award of the Degree.

VISION OF THE INSTITUTE

To be a premier center of professional education and research, offering quality programs in a socio-economic and ethical ambience.

MISSION OF THE INSTITUTE

- To impart knowledge of advanced technologies using state-of-the-art infrastructural facilities.
- To inculcate innovation and best practices in education, training and research.
- To meet changing socio-economic needs in an ethical ambience.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING – ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

DEPARTMENT VISION

To attain global standards in Computer Science and Engineering education, training and research to meet the growing needs of the industry with socio-economic and ethical considerations.

DEPARTMENT MISSION

- To impart quality education and research to undergraduate and postgraduate students in Computer Science and Engineering.
- To encourage innovation and best practices in Computer Science and Engineering utilizing state-of-the-art facilities.
- To develop entrepreneurial spirit and knowledge of emerging technologies based on ethical values and social relevance.

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

PEO1: Graduates will demonstrate technical skills, competency in AI & ML and exhibit team management capability with proper communication in a job environment

PEO2: Graduates will function in their profession with social awareness and responsibility

PEO3: Graduates will interact with their peers in other disciplines in industry and society and contribute to the economic growth of the country

PEO4: Graduates will be successful in pursuing higher studies in engineering or management

PROGRAMME OUTCOMES (POs)

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem analysis: Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and team work: Function effectively as an individual and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

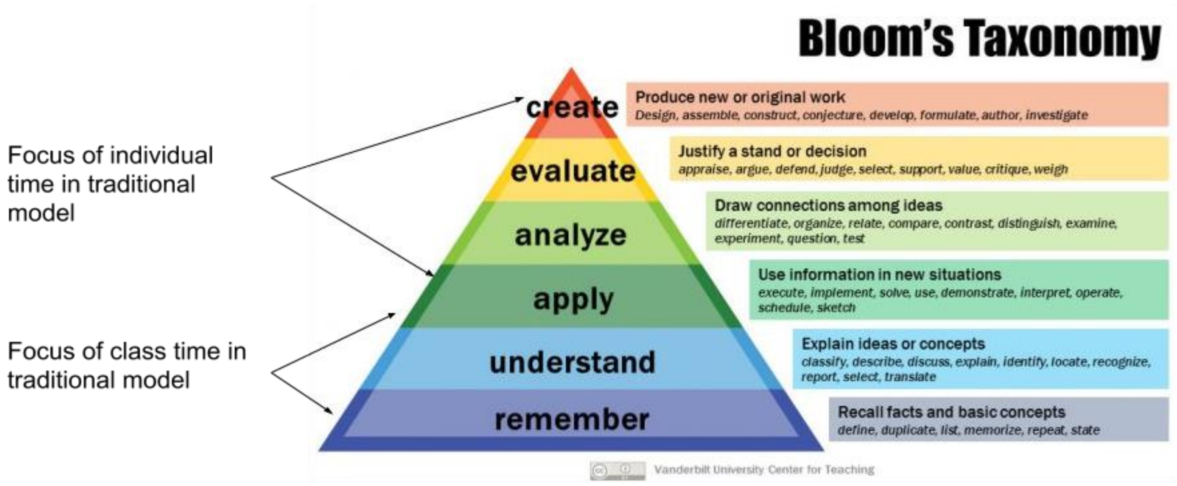
PROGRAMME SPECIFIC OUTCOMES (PSOs)

PSO1: Design and develop intelligent automated systems applying mathematical, analytical, programming and operational skills to solve real world problems

PSO2: Apply machine learning techniques, software tools to conduct experiments, interpret data and to solve complex problems

PSO3: Implement engineering solutions for the benefit of society by the use of AI and ML

BLOOM’S TAXONOMY (BT) TRIANGLE & BLOOM’S ACTION VERBS


Bloom’s Taxonomy
Focus of individual time in traditional model
Focus of class time in traditional model
create
evaluate
analyze
apply
understand
remember
Vanderbilt University Center for Teaching

Produce new or original work — *Design, assemble, construct, conjecture, develop, formulate, author, investigate*

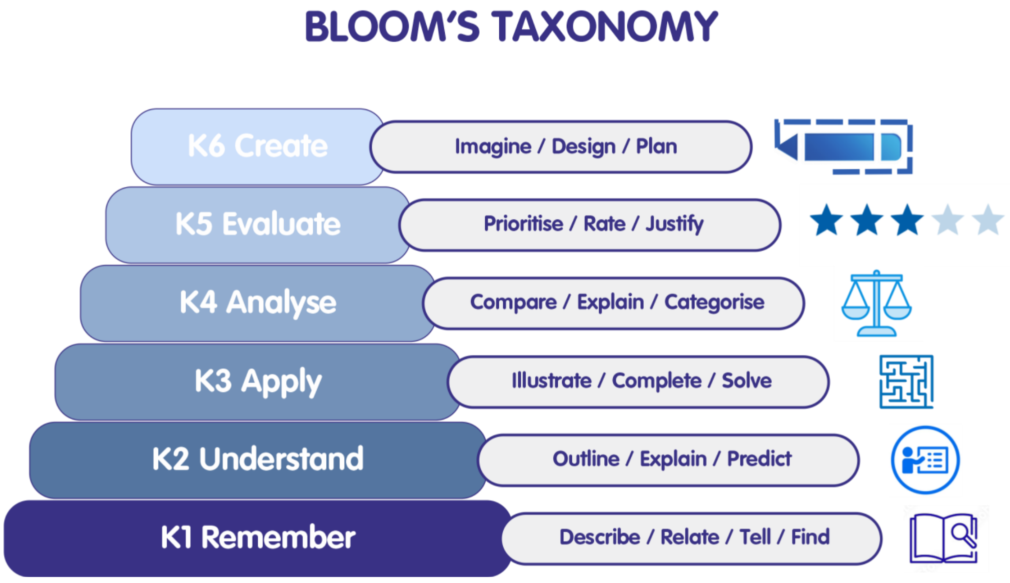
Justify a stand or decision — *appraise, argue, defend, judge, select, support, value, critique, weigh*

Draw connections among ideas — *differentiate, organize, relate, compare, contrast, distinguish, examine, experiment, question, test*

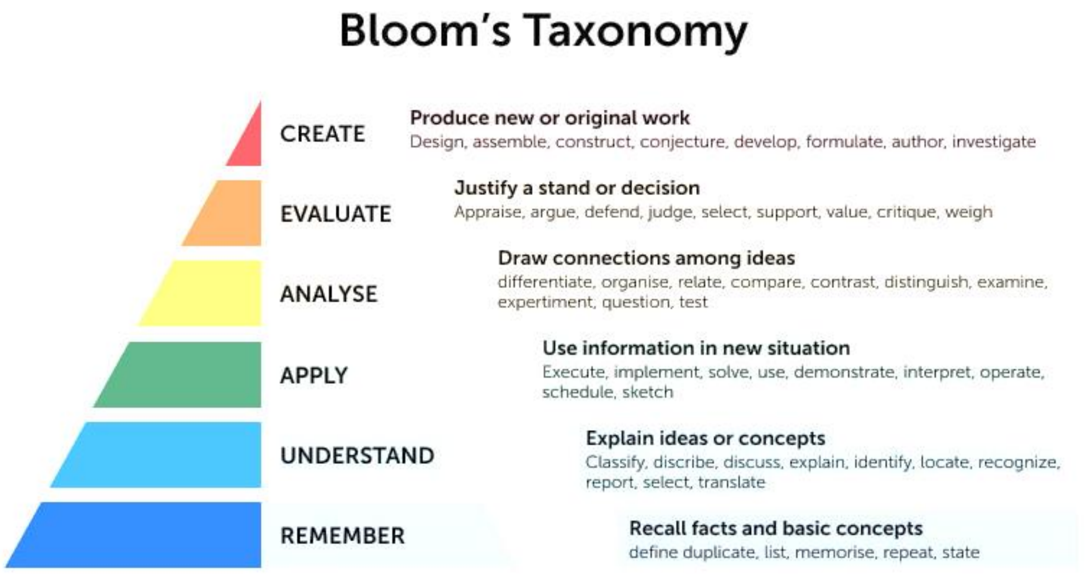
Use information in new situations — *execute, implement, solve, use, demonstrate, interpret, operate, schedule, sketch*

Explain ideas or concepts — *classify, describe, discuss, explain, identify, locate, recognize, report, select, translate*

Recall facts and basic concepts — *define, duplicate, list, memorize, repeat, state*



K6 Create	Imagine / Design / Plan
K5 Evaluate	Prioritise / Rate / Justify
K4 Analyse	Compare / Explain / Categorise
K3 Apply	Illustrate / Complete / Solve
K2 Understand	Outline / Explain / Predict
K1 Remember	Describe / Relate / Tell / Find



CREATE	**Produce new or original work** Design, assemble, construct, conjecture, develop, formulate, author, investigate
EVALUATE	**Justify a stand or decision** Appraise, argue, defend, judge, select, support, value, critique, weigh
ANALYSE	**Draw connections among ideas** differentiate, organise, relate, compare, contrast, distinguish, examine, expertiment, question, test
APPLY	**Use information in new situation** Execute, implement, solve, use, demonstrate, interpret, operate, schedule, sketch
UNDERSTAND	**Explain ideas or concepts** Classify, discribe, discuss, explain, identify, locate, recognize, report, select, translate
REMEMBER	**Recall facts and basic concepts** define duplicate, list, memorise, repeat, state

BLOOM’S ACTION VERBS

REVISED Bloom’s Taxonomy Action Verbs

Definitions	**I. Remembering**	**II. Understanding**	**III. Applying**	**IV. Analyzing**	**V. Evaluating**	**VI. Creating**
Bloom’s Definition	Exhibit memory of previously learned material by recalling facts, terms, basic concepts, and answers.	Demonstrate understanding of facts and ideas by organizing, comparing, translating, interpreting, giving descriptions, and stating main ideas.	Solve problems to new situations by applying acquired knowledge, facts, techniques and rules in a different way.	Examine and break information into parts by identifying motives or causes. Make inferences and find evidence to support generalizations.	Present and defend opinions by making judgments about information, validity of ideas, or quality of work based on a set of criteria.	Compile information together in a different way by combining elements in a new pattern or proposing alternative solutions.
Verbs	• Choose • Define • Find • How • Label • List • Match • Name • Omit • Recall • Relate • Select • Show • Spell • Tell • What • When • Where • Which • Who • Why	• Classify • Compare • Contrast • Demonstrate • Explain • Extend • Illustrate • Infer • Interpret • Outline • Relate • Rephrase • Show • Summarize • Translate	• Apply • Build • Choose • Construct • Develop • Experiment with • Identify • Interview • Make use of • Model • Organize • Plan • Select • Solve • Utilize	• Analyze • Assume • Categorize • Classify • Compare • Conclusion • Contrast • Discover • Dissect • Distinguish • Divide • Examine • Function • Inference • Inspect • List • Motive • Relationships • Simplify • Survey • Take part in • Test for • Theme	• Agree • Appraise • Assess • Award • Choose • Compare • Conclude • Criteria • Criticize • Decide • Deduct • Defend • Determine • Disprove • Estimate • Evaluate • Explain • Importance • Influence • Interpret • Judge • Justify • Mark • Measure • Opinion • Perceive • Prioritize • Prove • Rate • Recommend • Rule on • Select • Support • Value	• Adapt • Build • Change • Choose • Combine • Compile • Compose • Construct • Create • Delete • Design • Develop • Discuss • Elaborate • Estimate • Formulate • Happen • Imagine • Improve • Invent • Make up • Maximize • Minimize • Modify • Original • Originate • Plan • Predict • Propose • Solution • Solve • Suppose • Test • Theory

Anderson, L. W., & Krathwohl, D. R. (2001). A taxonomy for learning, teaching, and assessing, Abridged Edition. Boston, MA: Allyn and Bacon.

2022-23 Onwards (MR-22)	MALLA REDDY ENGINEERING COLLEGE (AUTONOMOUS)	B.Tech. VI Semester		
Code: C0519	**DATABASE MANAGEMENT SYSTEMS LAB**	**L**	**T**	**P**
Credits: 1		**-**	**-**	**2**

Course Objectives:
- Introduce ER data model, database design and normalization
- Learn SQL basics for data definition and data manipulation

Software Requirements: MySQL

LIST OF EXPERIMENTS:
1. Concept design with E-R Model
2. Relational Model
3. Normalization
4. Practicing DDL commands
5. Practicing DML commands
6. A. Querying (using ANY, ALL, UNION, INTERSECT, JOIN, Constraints etc.) Nested, Correlated subqueries
7. Queries using Aggregate functions, GROUP BY, HAVING and Creation and dropping of Views.
8. Triggers (Creation of insert trigger, delete trigger, update trigger)
9. Procedures
10. Usage of Cursors

TEXT BOOKS:
Database Management Systems, Raghurama Krishnan, Johannes Gehrke, Tata Mc Graw Hill,3rd Edition
Database System Concepts, Silberschatz, Korth, McGraw Hill, V edition.

REFERENCE BOOKS:
1. Database Systems design, Implementation, and Management, Peter Rob & Carlos Coronel 7thEdition.
2. Fundamentals of Database Systems, Elmasri Navrate, Pearson Education
3. Introduction to Database Systems, C.J. Date, Pearson Education
4. Oracle for Professionals, The X Team, S. Shah and V. Shah, SPD.
5. Database Systems Using Oracle: A Simplified guide to SQL and PL/SQL, Shah, PHI.
6. Fundamentals of Database Management Systems, M. L. Gillenson, Wiley Student Edition

Course Outcomes:
- Design database schema for a given application and apply normalization
- Acquire skills in using SQL commands for data definition and data manipulation.
- Develop solutions for database applications using procedures, cursors and triggers

List of Experiments

1	Railway Reservation System -(Redesigning IRCTC database) **Train** (train Number, name, source, destination, start_time, reach_time, traveltime, distance, class, days, type) **Ticket** (PNRNo, Transactionid, from_station, To_station, date_of_journey, class date_of_booking, total_ticket_fare, train number) **Passenger** (PNR No, Serial no, Name, Age, Reservation_status) **Train_Route**(Train_No, route_no, station_code, name, arrival_time, depart_time, distance, day) **Train_Ticket_fare**(Train_No, class, base_fare, reservation_charge, superfast_charge, other_charge, tatkal_charge, service_tax)
	Create all the tables specified above. Make underlined columns as primary key.(use number, number(m,n), varchar(n), date, time, timestamp data types appropriately) Insert atleast 5 rows to each table. (Check www.irctc.co.in website for actual data) 1. Use Interactive insertion for inserting rows to thetable. 2. Use ADT (varray) for class and days column in Traintable.
2	Write simple DDL/DML Queries to 1. Remove all the rows from Passenger tablepermanently. 2. Change the name of the Passenger table toPassenger_Details. 3. List all traindetails. 4. List all passengerdetails. 5. Give a list of trains in ascending order ofnumber. 6. List the senior citizen passengersdetails. 7. List the station names where code starts with'M'. 8. List the trains details within a range of numbers. 9. Change the super fast charge value in train fare as zero, if it isnull. 10. List the passenger names whose tickets are notconfirmed. 11. List the base_fare of all AC coaches available in each train. Find the ticket details where transaction id is notknown. 1) Use Interactive updation for updating the seat no for particular PNR NO. 2) Find the train names that are from Secunderabad to Mumbai, but do not have the sourceor destination in itsname. 3) 3) Find the train details that are on Thursday (Use the ADT column created).

3	Create (Alter table to add constraint) the necessary foreign keys by identifying the relationships in the table. 1) Add a suitable constraint to train table to always have train no in the range 10001 to 99999. 2) Add a suitable constraint for the column of station name, so that does not take duplicates. 3) Change the data type of arrival time, depart time (date -> timestamp or timestamp to date), and do the necessary process for updating the table with new values. 4) Add a suitable constraint for the class column that it should take values only as 1A, 2A, 3A, SL, C. 5) Add a not null constraint for the column distance in train_route.
4	Use SQL PLUS functions to. 1. Find the passengers whose date of journey is one month from today. 2. Print the train names in upper case. 3. Print the passenger names with left padding character. 4. Print the station codes replacing K with M. 5. Translate all the LC in class column (Train_fare) to POT anddisplay. 6. Display the fare details of all trains, if any value is ZERO, print as NULL value. 7. Display the pnrno and transaction id, if transaction id is null, print 'not generated'. 8. Print the date_of_jounrney in the format '27th November 2010'. 9. Find the maximum fare (total fare). 10. Find the average age of passengers in one ticket. 11. Find the maximum length of station name available in the database. 12. Print the fare amount of the passengers as rounded value. 13. Add the column halt time to train route. 14. Update values to it from arrival time and depart time. High Level: 15. Update values to arrival time and depart time using conversion functions. 16. Display the arrival time, depart time in the format HH:MI (24 hours andminutes).
5	Querying Aggregate Functions(COUNT,SUM,AVG,MAX and MIN) Bus: Bus(BusNo: String, Source: String, Destination: String, CoachType: String) Ticket: Ticket(TicketNo: string, DOJ: date, Address:string,ContactNo: string, BusNo:String, SeatNo :Integer, Source: String, Destination: String) Passenger: Passenger(PassportID: String, TicketNo:string,Name: String, ContactNo:string,Age: integer, Sex: character, Address: String); Reservation: Reservation(PNRNo: String, DOJ: Date, NoofSeats: integer , Address: String ,ContactNo: String, , BusNo: String,SeatNo:Integer) Cancellation: Cancellation (PNRNo: String,DOJ: Date, SeatNo:

	integer,ContactNo: String,Status: String) 1. Write a Query to display the information present in the passenger and cancellation tables 2. Display the number of days in a week on which the AP123 bus is available 3. Find number of tickets booked for each PNR_No using GROUP BY CLAUSE 4. Find the distinct PNR Numbers that are present.
6	Querying (using ANY, ALL, IN, Exists, NOT EXISTS, UNION, INTERSECT, Constraints etc.) 1. Display unique PNR_NO of all passengers 2. Display all the names of male passengers. 3. Display the ticket numbers and names of all the passengers. 4. Find the ticket numbers of the passengers whose name start with 'r' and ends with 'h'. 5. Find the names of Passengers whose age is between 30 and 45. 6. Display all the passengers names beginning with 'A'. 7. Display the sorted list of Passengers names
7	Joins , Nested Queries & Views: Create a table EMP with the following structure. COLUMN Name DATA Type --- EMPNO INTEGER(6) ENAME VARCHAR2(20) JOB VARCHAR2(10) MGR INTEGER (4) DEPTNO INTEGER (3) SAL INTEGER (7) 2. Create dept table with the following structure. COLUMN Name DATA Type --- DEPTNO INTEGER (2) DNAME VARCHAR2(10) LOC VARCHAR2(10) DEPTNO as the primary key 1. Display all the employees and the departments implementing a left outer join. 2. Display the employee name and department name in which they are working implementing a full outer join.

	3. Find the third highest salary of an employee. 4. Display all employee names and salary whose salary is greater than minimum salary of the company and job title starts with 'M'. 5.Write a query to display information about employees who earn more than any employee in dept 30. 6. Write a query to create and drop View
8	Write a simple PL/SQL block to. 1. Print the factorial of a given number. 2. Print the Fibonacci series
9	Write a cursor for the following. 1. Declare a cursor that defines a result set. 2. Open the cursor to establish the result set. 3. Fetch the data into local variables as needed from the cursor, one row at a time. 4. Close the cursor when done.
10	Write a PL/SQL procedure 1. For creation of stored procedure, Execution of procedure and modification of procedure.
11	Write a Trigger for the following: 1. Creation of insert trigger, delete trigger, update trigger.
12	Use TCL commands for your transactions. (Commit, Rollback, Savepoint)

:TASK 1 :

Q.Railway Reservation System -(Redesigning IRCTC database)

a: create a table containing the following data

Train (train Number, name, source, destination, start_time, reach_time, traveltime, distance, class,days, type)

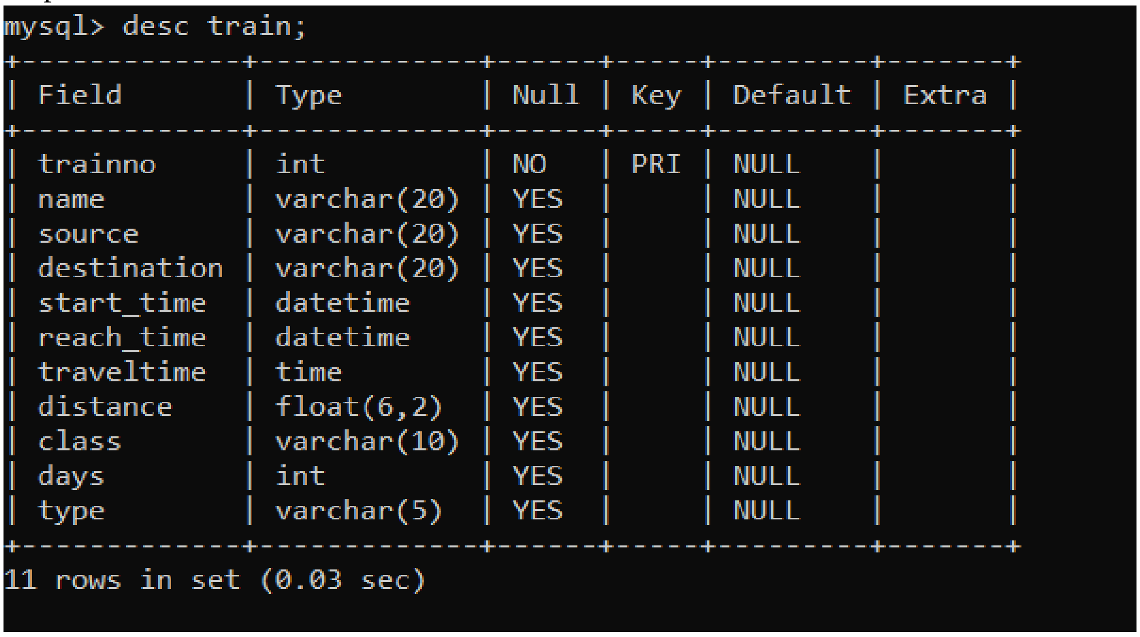
syntax:

create database dbmslab;

use dbmslab;

create table Train(trainno INT(6) PRIMARY KEY,name VARCHAR(20),source VARCHAR(20),destination VARCHAR(20),start_time DATETIME,reach_time DATETIME,traveltime TIME,distance FLOAT(6,2), class VARCHAR(10),days INT(2),type VARCHAR(5));

Output:



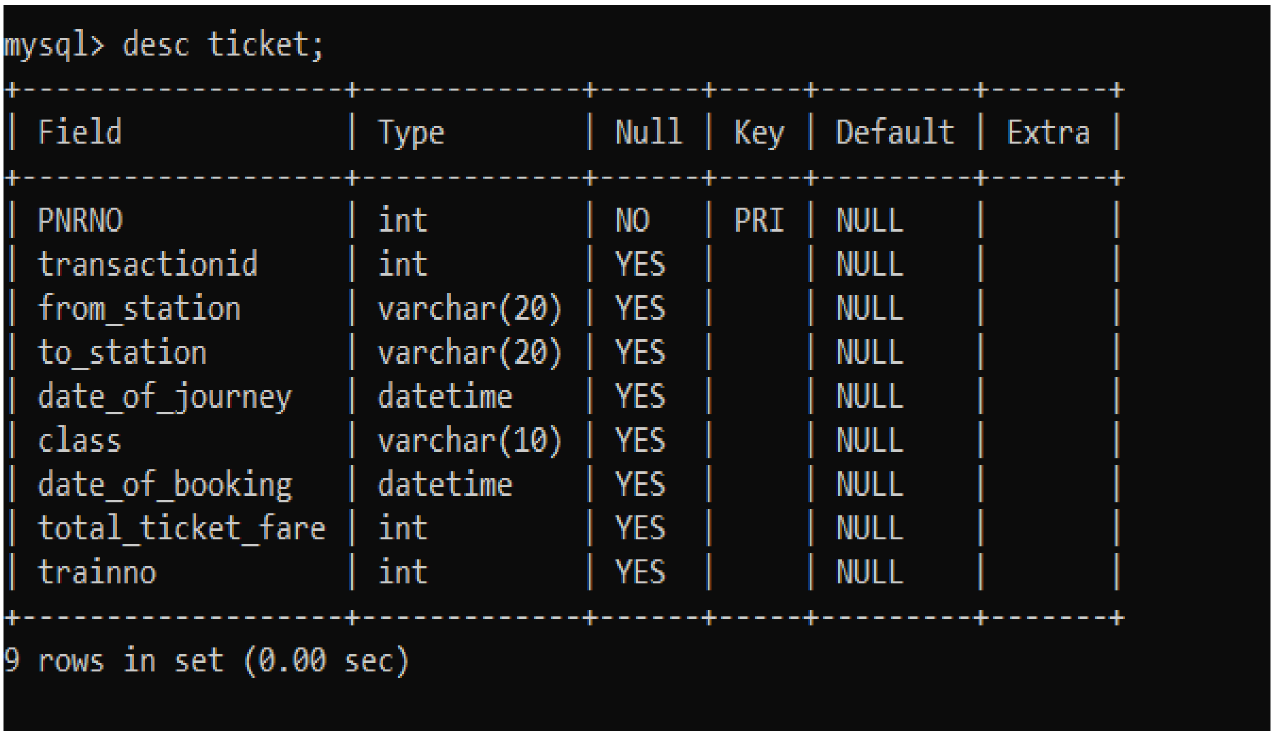
```
mysql> desc train;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| trainno     | int         | NO   | PRI | NULL    |       |
| name        | varchar(20) | YES  |     | NULL    |       |
| source      | varchar(20) | YES  |     | NULL    |       |
| destination | varchar(20) | YES  |     | NULL    |       |
| start_time  | datetime    | YES  |     | NULL    |       |
| reach_time  | datetime    | YES  |     | NULL    |       |
| traveltime  | time        | YES  |     | NULL    |       |
| distance    | float(6,2)  | YES  |     | NULL    |       |
| class       | varchar(10) | YES  |     | NULL    |       |
| days        | int         | YES  |     | NULL    |       |
| type        | varchar(5)  | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
11 rows in set (0.03 sec)
```

(b).create a table containing the following data
Ticket (PNRNo, Transactionid, from_station, To_station, date_of_journey, class date_of_booking,
total_ticket_fare, train number)

syntax:

create table Ticket (PNRNO INT(10) PRIMARY KEY, transactionid INT(10), from_station VARCHAR(20), to_station VARCHAR(20), date_of_journey DATETIME, class VARCHAR(10), date_of_booking DATETIME, total_ticket_fare INT(5), trainno INT(6));

Output:



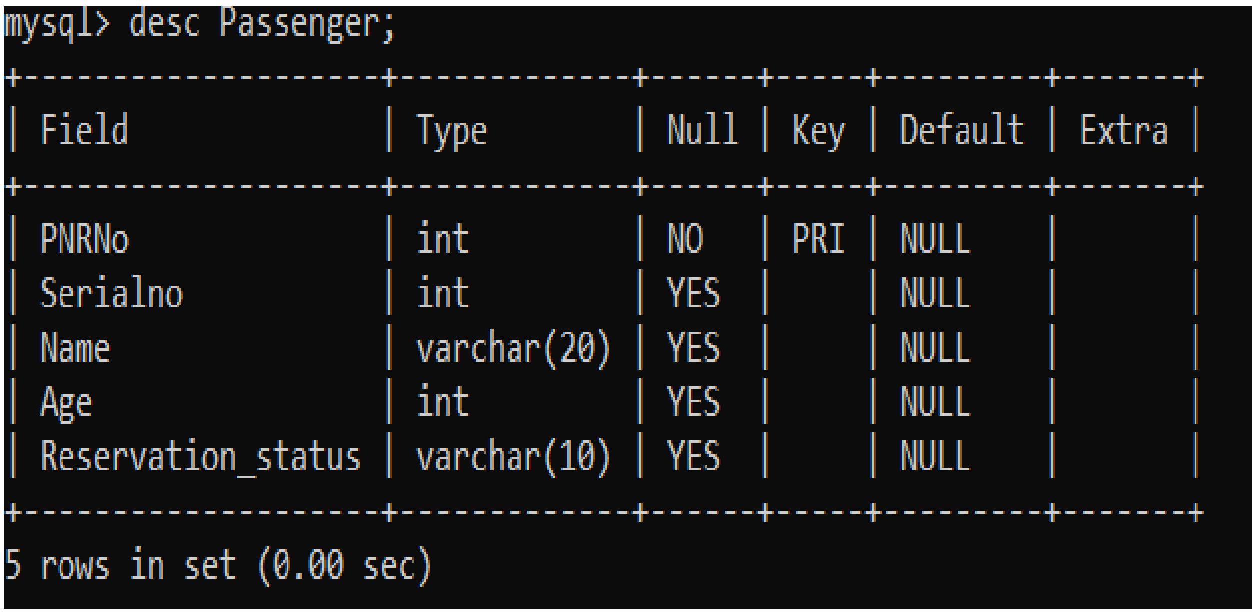
```
mysql> desc ticket;
+-------------------+-------------+------+-----+---------+-------+
| Field             | Type        | Null | Key | Default | Extra |
+-------------------+-------------+------+-----+---------+-------+
| PNRNO             | int         | NO   | PRI | NULL    |       |
| transactionid     | int         | YES  |     | NULL    |       |
| from_station      | varchar(20) | YES  |     | NULL    |       |
| to_station        | varchar(20) | YES  |     | NULL    |       |
| date_of_journey   | datetime    | YES  |     | NULL    |       |
| class             | varchar(10) | YES  |     | NULL    |       |
| date_of_booking   | datetime    | YES  |     | NULL    |       |
| total_ticket_fare | int         | YES  |     | NULL    |       |
| trainno           | int         | YES  |     | NULL    |       |
+-------------------+-------------+------+-----+---------+-------+
9 rows in set (0.00 sec)
```

(c). create a table containing the following data
Passenger (PNR No, Serial no, Name, Age, Reservation_status)

syntax:

create table Passenger (PNRNo INT(10) primary key , Serialno INT(10), Name VARCHAR(20), Age INT(3), Reservation_status VARCHAR(10));

Output:



```
mysql> desc Passenger;
+--------------------+-------------+------+-----+---------+-------+
| Field              | Type        | Null | Key | Default | Extra |
+--------------------+-------------+------+-----+---------+-------+
| PNRNo              | int         | NO   | PRI | NULL    |       |
| Serialno           | int         | YES  |     | NULL    |       |
| Name               | varchar(20) | YES  |     | NULL    |       |
| Age                | int         | YES  |     | NULL    |       |
| Reservation_status | varchar(10) | YES  |     | NULL    |       |
+--------------------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

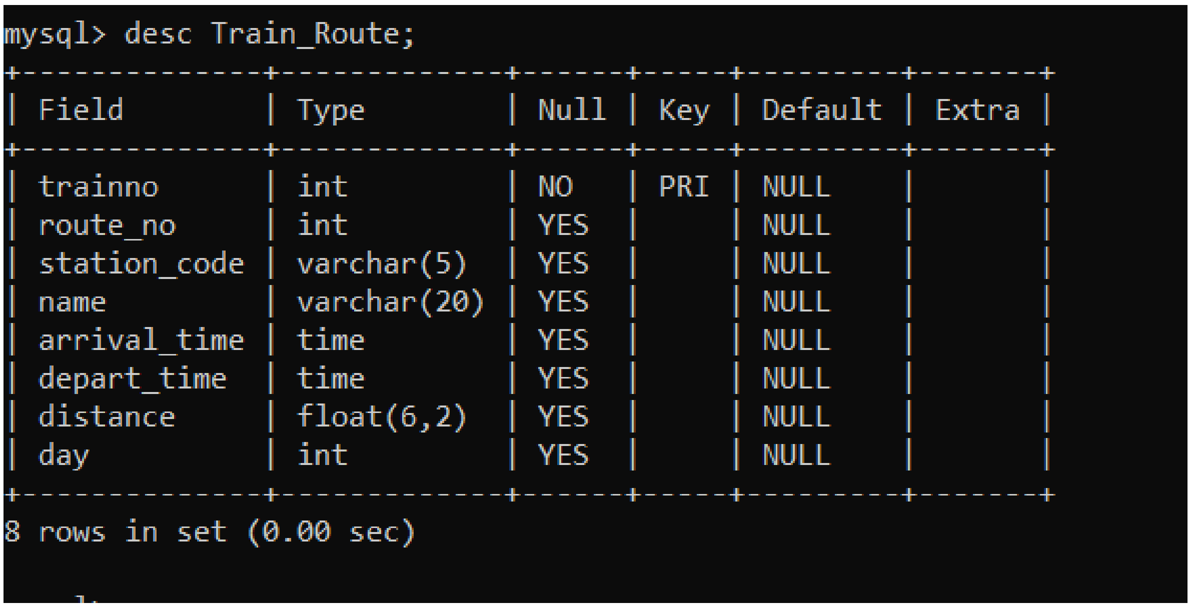
(d). create a table containing the following data Train_Route(Train_No, route_no, station_code, name, a

rrival_time, depart_time, distance, day)

syntax:

create table Train_Route(trainno INT(6) primary key, route_no INT(6), station_code VARCHAR(5), name VARCHAR(20), arrival_time TIME, depart_time TIME, distance FLOAT(6,2), day INT(2));

Output:



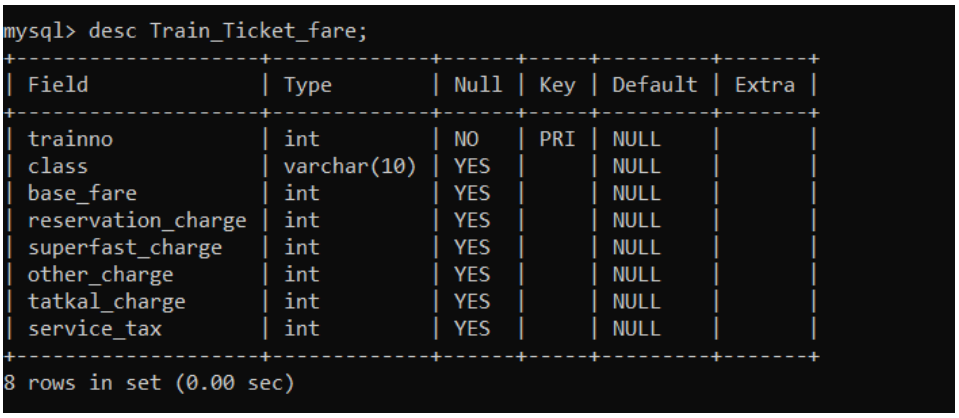
```
mysql> desc Train_Route;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| trainno      | int         | NO   | PRI | NULL    |       |
| route_no     | int         | YES  |     | NULL    |       |
| station_code | varchar(5)  | YES  |     | NULL    |       |
| name         | varchar(20) | YES  |     | NULL    |       |
| arrival_time | time        | YES  |     | NULL    |       |
| depart_time  | time        | YES  |     | NULL    |       |
| distance     | float(6,2)  | YES  |     | NULL    |       |
| day          | int         | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
8 rows in set (0.00 sec)
```

(e).create a table containing the following data Train_Ticket_fare(Train_No, class, base_fare, reservation_charge, superfast_charge, other_charge, tatkal_charge, service_tax)

syntax:

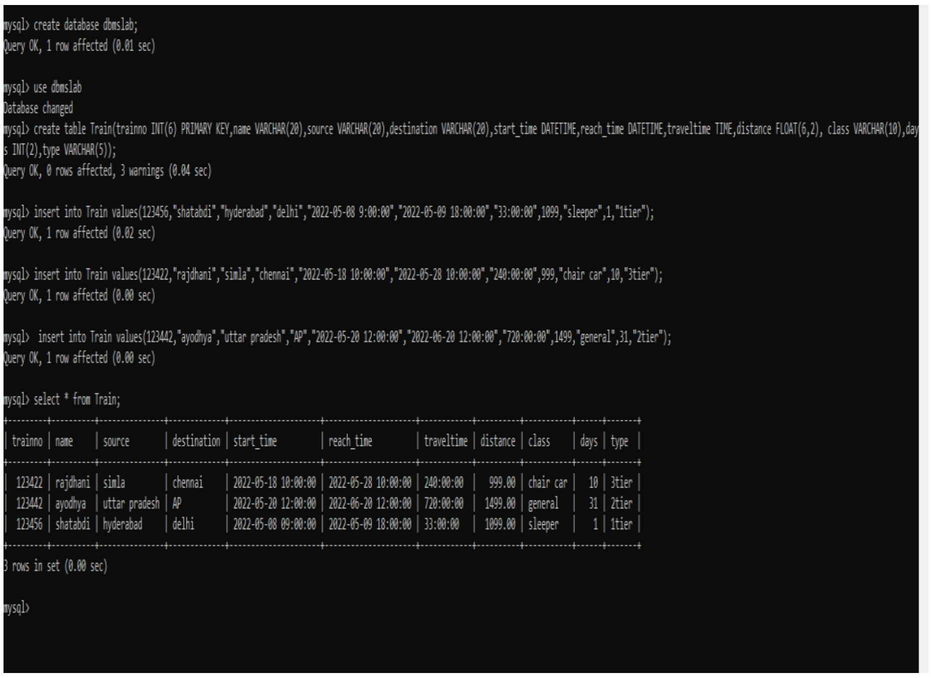
create table Train_Ticket_fare(trainno INT(6) primary key, class VARCHAR(10), base_fare INT(4), reservation_charge INT(4), superfast_charge INT(4), other_charge INT(4), tatkal_charge INT(4), service_tax INT(4);

Output:



```
mysql> desc Train_Ticket_fare;
+--------------------+-------------+------+-----+---------+-------+
| Field              | Type        | Null | Key | Default | Extra |
+--------------------+-------------+------+-----+---------+-------+
| trainno            | int         | NO   | PRI | NULL    |       |
| class              | varchar(10) | YES  |     | NULL    |       |
| base_fare          | int         | YES  |     | NULL    |       |
| reservation_charge | int         | YES  |     | NULL    |       |
| superfast_charge   | int         | YES  |     | NULL    |       |
| other_charge       | int         | YES  |     | NULL    |       |
| tatkal_charge      | int         | YES  |     | NULL    |       |
| service_tax        | int         | YES  |     | NULL    |       |
+--------------------+-------------+------+-----+---------+-------+
8 rows in set (0.00 sec)
```

Example for inserting the values into the table & how to show the data present in the table:



```
mysql> create database dbmslab;
Query OK, 1 row affected (0.01 sec)

mysql> use dbmslab
Database changed
mysql> create table Train(trainno INT(6) PRIMARY KEY,name VARCHAR(20),source VARCHAR(20),destination VARCHAR(20),start_time DATETIME,reach_time DATETIME,traveltime TIME,distance FLOAT(6,2), class VARCHAR(10),day
s INT(2),type VARCHAR(5));
Query OK, 0 rows affected, 3 warnings (0.04 sec)

mysql> insert into Train values(123456,"shatabdi","hyderabad","delhi","2022-05-08 9:00:00","2022-05-09 18:00:00","33:00:00",1099,"sleeper",1,"1tier");
Query OK, 1 row affected (0.02 sec)

mysql> insert into Train values(123422,"rajdhani","simla","chennai","2022-05-18 10:00:00","2022-05-28 10:00:00","240:00:00",999,"chair car",10,"3tier");
Query OK, 1 row affected (0.00 sec)

mysql>  insert into Train values(123442,"ayodhya","uttar pradesh","AP","2022-05-20 12:00:00","2022-06-20 12:00:00","720:00:00",1499,"general",31,"2tier");
Query OK, 1 row affected (0.00 sec)

mysql> select * from Train;
+---------+----------+---------------+-------------+---------------------+---------------------+------------+----------+-----------+------+-------+
| trainno | name     | source        | destination | start_time          | reach_time          | traveltime | distance | class     | days | type  |
+---------+----------+---------------+-------------+---------------------+---------------------+------------+----------+-----------+------+-------+
|  123422 | rajdhani | simla         | chennai     | 2022-05-18 10:00:00 | 2022-05-28 10:00:00 | 240:00:00  |   999.00 | chair car |   10 | 3tier |
|  123442 | ayodhya  | uttar pradesh | AP          | 2022-05-20 12:00:00 | 2022-06-20 12:00:00 | 720:00:00  |  1499.00 | general   |   31 | 2tier |
|  123456 | shatabdi | hyderabad     | delhi       | 2022-05-08 09:00:00 | 2022-05-09 18:00:00 | 33:00:00   |  1099.00 | sleeper   |    1 | 1tier |
+---------+----------+---------------+-------------+---------------------+---------------------+------------+----------+-----------+------+-------+
3 rows in set (0.00 sec)

mysql>
```

:TASK 2 :

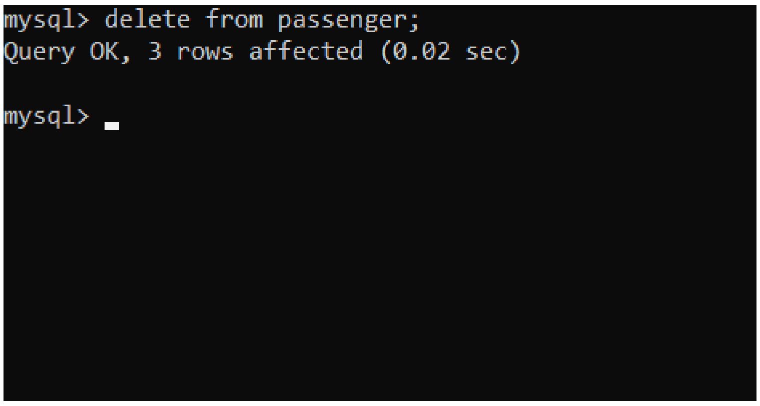
Q.Write simple DDL/DML Queries to

1. Remove all the rows from Passenger table permanently.

syntax:

TRUNCATE TABLE passenger;

Output:



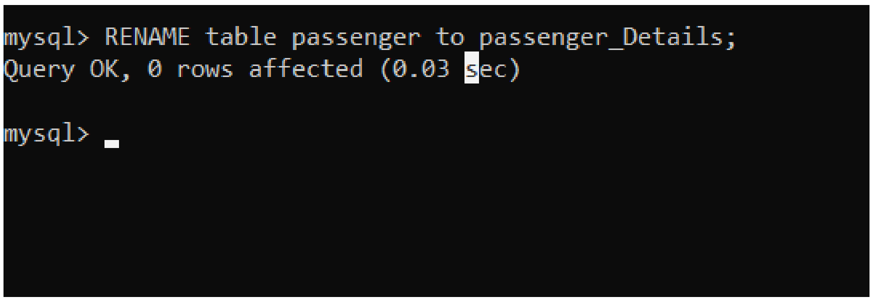
```
mysql> delete from passenger;
Query OK, 3 rows affected (0.02 sec)

mysql> _
```

2. Change the name of the Passenger table to Passenger_Details.

syntax:

RENAME table passenger to passenger_Details;

Output:



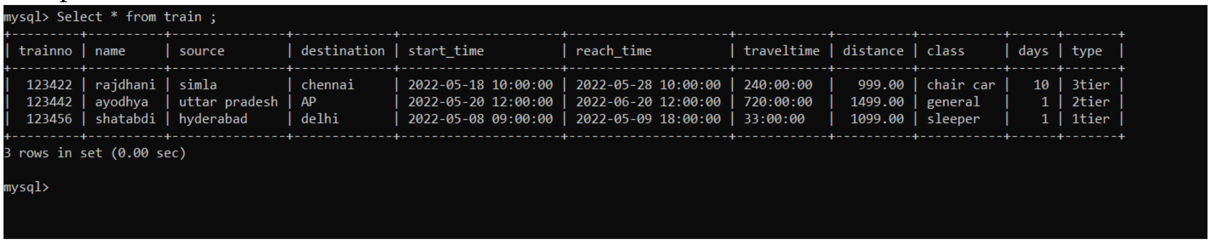
```
mysql> RENAME table passenger to passenger_Details;
Query OK, 0 rows affected (0.03 sec)

mysql> _
```

3. List all train details.

syntax:

Select * from train ;

Output:



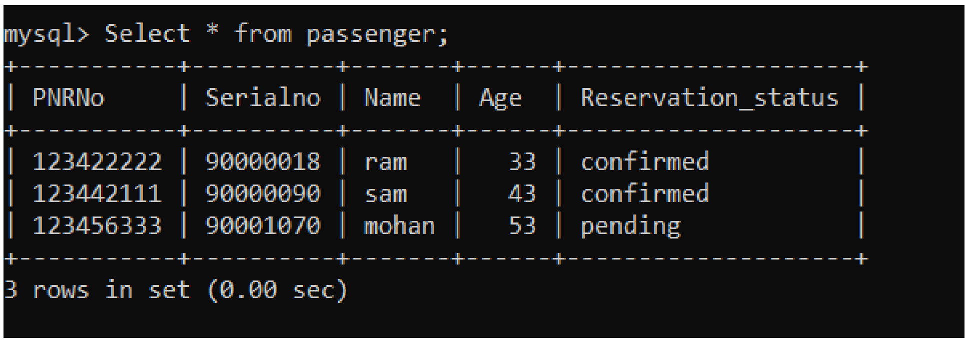
```
mysql> Select * from train ;
+---------+----------+---------------+-------------+---------------------+---------------------+------------+----------+-----------+------+-------+
| trainno | name     | source        | destination | start_time          | reach_time          | traveltime | distance | class     | days | type  |
+---------+----------+---------------+-------------+---------------------+---------------------+------------+----------+-----------+------+-------+
|  123422 | rajdhani | simla         | chennai     | 2022-05-18 10:00:00 | 2022-05-28 10:00:00 | 240:00:00  |   999.00 | chair car |   10 | 3tier |
|  123442 | ayodhya  | uttar pradesh | AP          | 2022-05-20 12:00:00 | 2022-06-20 12:00:00 | 720:00:00  |  1499.00 | general   |    1 | 2tier |
|  123456 | shatabdi | hyderabad     | delhi       | 2022-05-08 09:00:00 | 2022-05-09 18:00:00 | 33:00:00   |  1099.00 | sleeper   |    1 | 1tier |
+---------+----------+---------------+-------------+---------------------+---------------------+------------+----------+-----------+------+-------+
3 rows in set (0.00 sec)

mysql>
```

4. List all passenger details.

Syntax:

Select * from passenger;

Output:



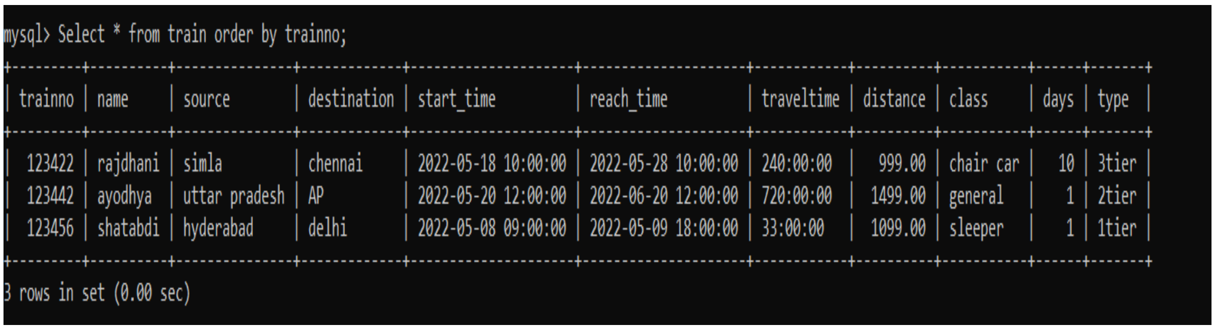
```
mysql> Select * from passenger;
+-----------+----------+-------+------+--------------------+
| PNRNo     | Serialno | Name  | Age  | Reservation_status |
+-----------+----------+-------+------+--------------------+
| 123422222 | 90000018 | ram   |   33 | confirmed          |
| 123442111 | 90000090 | sam   |   43 | confirmed          |
| 123456333 | 90001070 | mohan |   53 | pending            |
+-----------+----------+-------+------+--------------------+
3 rows in set (0.00 sec)
```

5. Give a list of trains in ascending order of number.

Syntax:

Select * from train order by trainno;

Output:



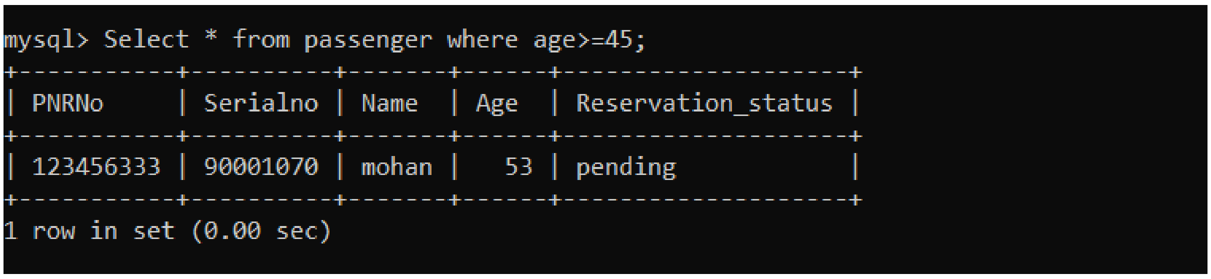
```
mysql> Select * from train order by trainno;
+---------+----------+---------------+-------------+---------------------+---------------------+------------+----------+-----------+------+-------+
| trainno | name     | source        | destination | start_time          | reach_time          | traveltime | distance | class     | days | type  |
+---------+----------+---------------+-------------+---------------------+---------------------+------------+----------+-----------+------+-------+
|  123422 | rajdhani | simla         | chennai     | 2022-05-18 10:00:00 | 2022-05-28 10:00:00 | 240:00:00  |   999.00 | chair car |   10 | 3tier |
|  123442 | ayodhya  | uttar pradesh | AP          | 2022-05-20 12:00:00 | 2022-06-20 12:00:00 | 720:00:00  |  1499.00 | general   |    1 | 2tier |
|  123456 | shatabdi | hyderabad     | delhi       | 2022-05-08 09:00:00 | 2022-05-09 18:00:00 | 33:00:00   |  1099.00 | sleeper   |    1 | 1tier |
+---------+----------+---------------+-------------+---------------------+---------------------+------------+----------+-----------+------+-------+
3 rows in set (0.00 sec)
```

6. List the senior citizen passengers details.

Syntax:

Select * from passenger where age>=45;

Output:



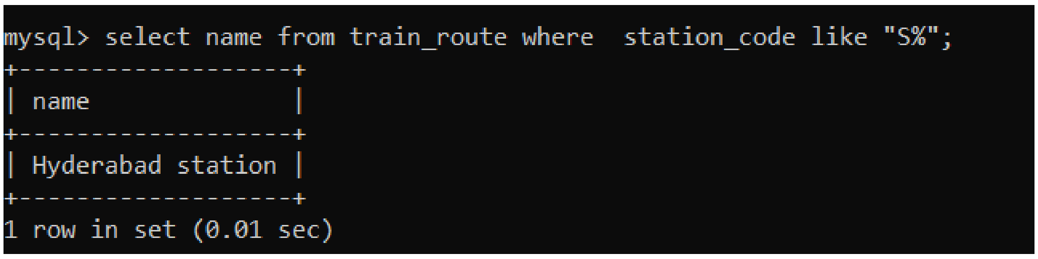
```
mysql> Select * from passenger where age>=45;
+-----------+----------+-------+------+--------------------+
| PNRNo     | Serialno | Name  | Age  | Reservation_status |
+-----------+----------+-------+------+--------------------+
| 123456333 | 90001070 | mohan |   53 | pending            |
+-----------+----------+-------+------+--------------------+
1 row in set (0.00 sec)
```

7. List the station names where code starts with 'S'.

Syntax:

select name from train_route where station_code like "S%";

Output:



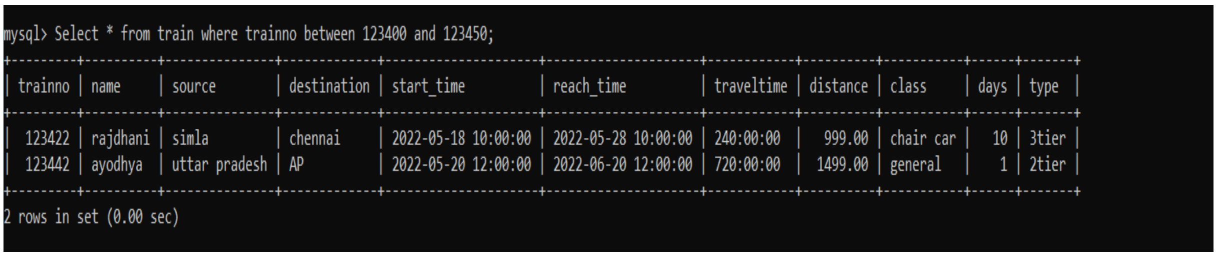
```
mysql> select name from train_route where  station_code like "S%";
+-------------------+
| name              |
+-------------------+
| Hyderabad station |
+-------------------+
1 row in set (0.01 sec)
```

8. List the trains details within a range of numbers.

Syntax:

Select * from train where trainno between 123400 and 123450;

Output:



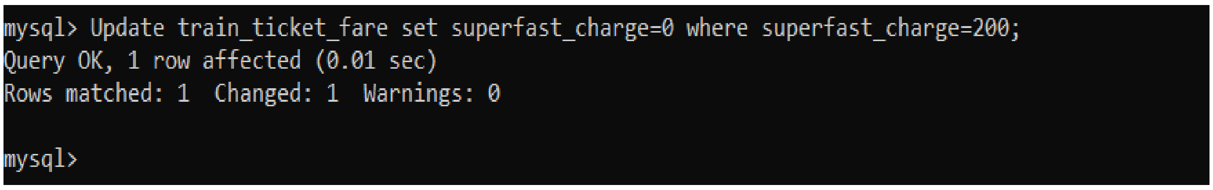
```
mysql> Select * from train where trainno between 123400 and 123450;
+---------+----------+---------------+-------------+---------------------+---------------------+------------+----------+-----------+------+-------+
| trainno | name     | source        | destination | start_time          | reach_time          | traveltime | distance | class     | days | type  |
+---------+----------+---------------+-------------+---------------------+---------------------+------------+----------+-----------+------+-------+
|  123422 | rajdhani | simla         | chennai     | 2022-05-18 10:00:00 | 2022-05-28 10:00:00 | 240:00:00  |   999.00 | chair car |   10 | 3tier |
|  123442 | ayodhya  | uttar pradesh | AP          | 2022-05-20 12:00:00 | 2022-06-20 12:00:00 | 720:00:00  |  1499.00 | general   |    1 | 2tier |
+---------+----------+---------------+-------------+---------------------+---------------------+------------+----------+-----------+------+-------+
2 rows in set (0.00 sec)
```

9. Change the super fast charge value in train fare as zero, if it is null.

Syntax:

Update train_ticket_fare set superfast_charge=0 where superfast_charge is NULL;

Output:



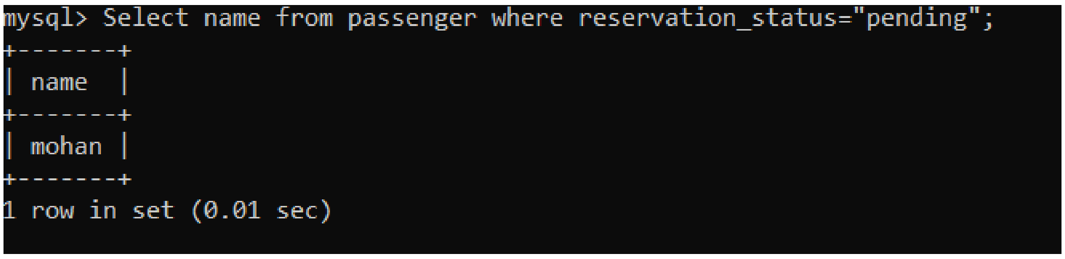
```
mysql> Update train_ticket_fare set superfast_charge=0 where superfast_charge=200;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql>
```

10. List the passenger names whose tickets are not confirmed.

Syntax:

Select name from passenger where reservation_status="pending";

Output:



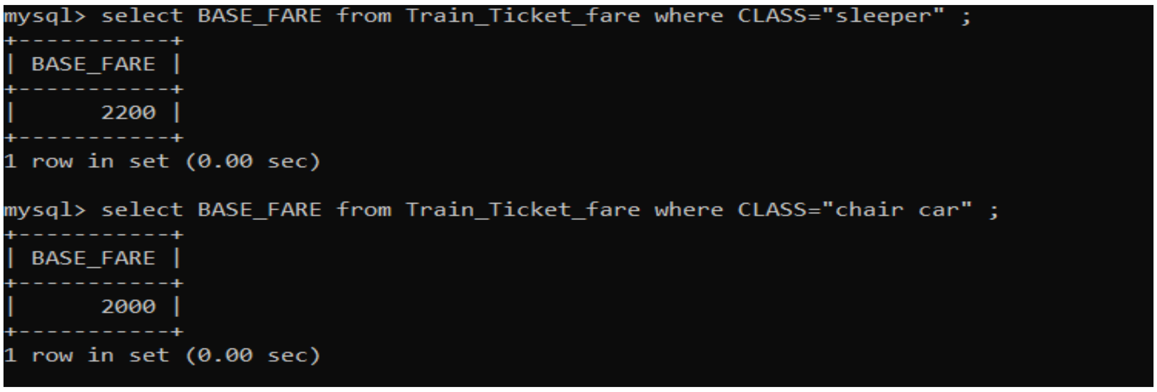
```
mysql> Select name from passenger where reservation_status="pending";
+-------+
| name  |
+-------+
| mohan |
+-------+
1 row in set (0.01 sec)
```

11. List the base_fare of all AC coaches available in each train.

Synatx:

select BASE_FARE from Train_Ticket_fare where CLASS=”chair car”;

output:



```
mysql> select BASE_FARE from Train_Ticket_fare where CLASS="sleeper" ;
+-----------+
| BASE_FARE |
+-----------+
|      2200 |
+-----------+
1 row in set (0.00 sec)

mysql> select BASE_FARE from Train_Ticket_fare where CLASS="chair car" ;
+-----------+
| BASE_FARE |
+-----------+
|      2000 |
+-----------+
1 row in set (0.00 sec)
```

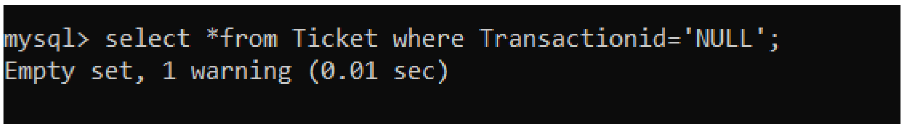
12.find the ticket details where transaction id is not known.

Synatx:

select *from Ticket where Transactionid='NULL';

Output:

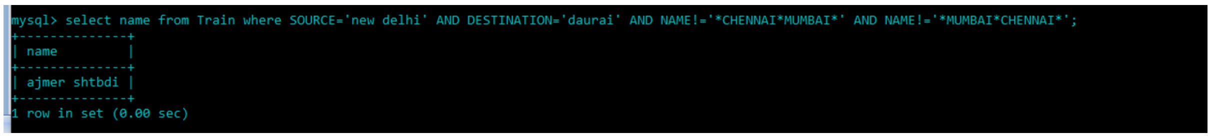
if there are no traansactions id with null data->



```
mysql> select *from Ticket where Transactionid='NULL';
Empty set, 1 warning (0.01 sec)
```

13. Find the train names that are from Chennai to Mumbai, but do not have the sourceor destination in itsname.

Syntax: select name from Train where SOURCE='CHENNAI' AND DESTINATION='MUMBAI' AND NAME!='*CHENNAI*MUMBAI*' AND NAME!='*MUMBAI*CHENNAI*';



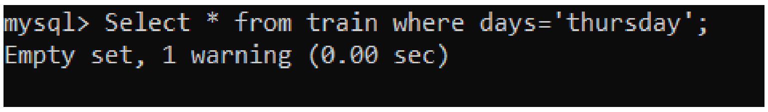
```
mysql> select name from Train where SOURCE='new delhi' AND DESTINATION='daurai' AND NAME!='*CHENNAI*MUMBAI*' AND NAME!='*MUMBAI*CHENNAI*';
+--------------+
| name         |
+--------------+
| ajmer shtbdi |
+--------------+
1 row in set (0.00 sec)
```

14. Find the train details that are on Thursday(Use the ADT column created)

Synatx:

Select * from train where days='thursday';

Output:



```
mysql> Select * from train where days='thursday';
Empty set, 1 warning (0.00 sec)
```

:TASK 3:

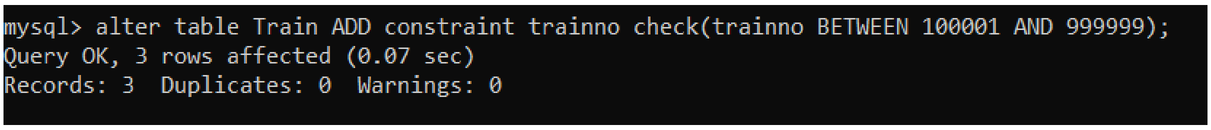
Q.Create (Alter table to add constraint) the necessary foreign keys by identifying the relationships in the table.

1) Add a suitable constraint to train table to always have train no in the range 10001 to 99999.

Syntax:

alter table Train ADD constraint trainno check(trainno BETWEEN 100001 AND 999999);

Output:



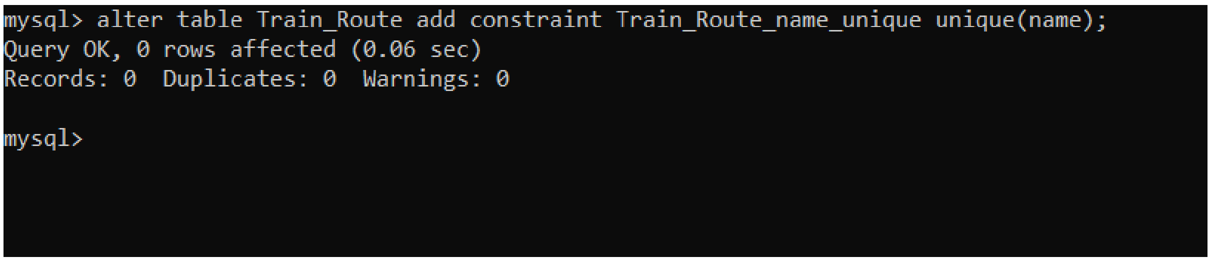
```
mysql> alter table Train ADD constraint trainno check(trainno BETWEEN 100001 AND 999999);
Query OK, 3 rows affected (0.07 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

2) Add a suitable constraint for the column of station name, so that does not take duplicates.

Syntax:

alter table Train_Route add constraint Train_Route_name_unique unique(name);

Output:



```
mysql> alter table Train_Route add constraint Train_Route_name_unique unique(name);
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
```

3) Change the data type of arrival time, depart time (date ->timestamp or timestamp to date), and do the necessary process for updating the table with new values.

Syntax:

alter table Train drop column start_time;

alter table train drop column reach_time;

alter table Train add start_time timestamp(0);

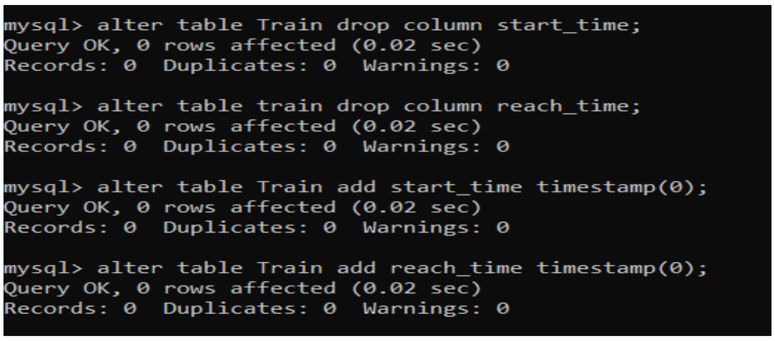
alter table Train add reach_time timestamp(0);

update Train set start_time=timestamp('2022-08-15 18:40:00'),reach_time=timestamp('2022-08-16 8:20:00') where trainno=123442;

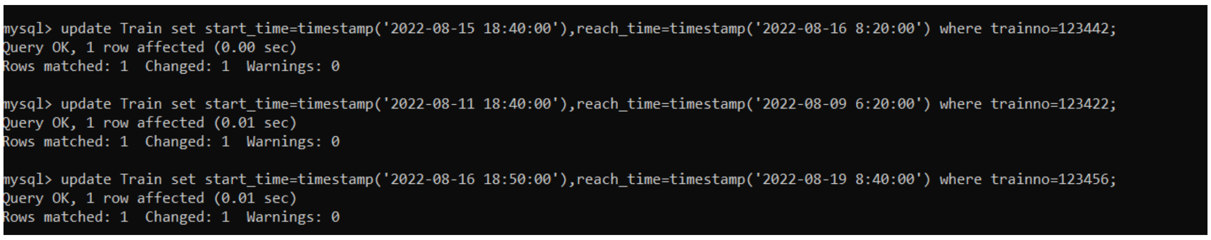
update Train set start_time=timestamp('2022-08-11 18:40:00'),reach_time=timestamp('2022-08-09 6:20:00') where trainno=123422;

update Train set start_time=timestamp('2022-08-16 18:50:00'),reach_time=timestamp('2022-08-19 8:40:00') where trainno=123456;

Output:



```
mysql> alter table Train drop column start_time;
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> alter table train drop column reach_time;
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> alter table Train add start_time timestamp(0);
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> alter table Train add reach_time timestamp(0);
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0
```



```
mysql> update Train set start_time=timestamp('2022-08-15 18:40:00'),reach_time=timestamp('2022-08-16 8:20:00') where trainno=123442;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> update Train set start_time=timestamp('2022-08-11 18:40:00'),reach_time=timestamp('2022-08-09 6:20:00') where trainno=123422;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> update Train set start_time=timestamp('2022-08-16 18:50:00'),reach_time=timestamp('2022-08-19 8:40:00') where trainno=123456;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

4) Add a suitable constraint for the class column that it should take values only as 1A, 2A, 3A, SL, C.

Syntax:

alter table train add constraint chk_valCHECK(class in(‘1A’,‘2A’,‘3A’,‘SL’,‘C’));

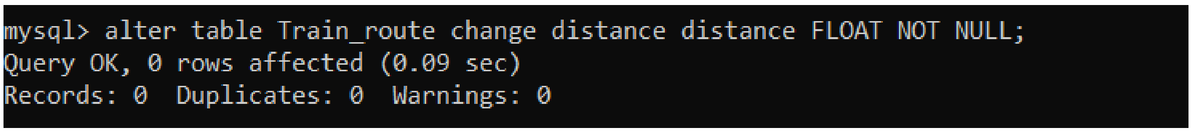
Output:

5) Add a not null constraint for the column distance in train_route.

Syntax:

alter table Train_route change distance distance FLOAT NOT NULL;

Output:



```
mysql> alter table Train_route change distance distance FLOAT NOT NULL;
Query OK, 0 rows affected (0.09 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

:TASK 4:

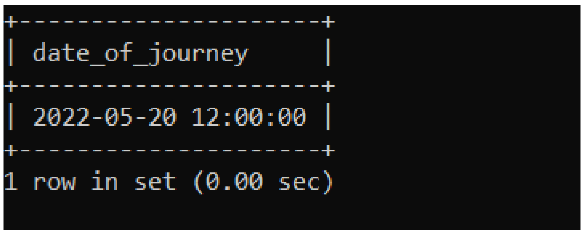
Q.Use SQL PLUS functions to.

1. Find the passengers whose date of journey is one month from today.

Syntax:

select date_of_journey from ticket where date_of_journey>date_add(now(),interval 30 day);

Output:



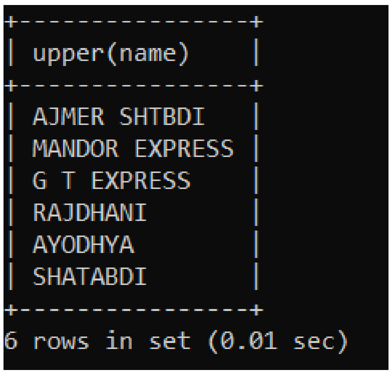
```
+---------------------+
| date_of_journey     |
+---------------------+
| 2022-05-20 12:00:00 |
+---------------------+
1 row in set (0.00 sec)
```

2. Print the train names in upper case.

Syntax:

select upper(name) from Train;

Output:



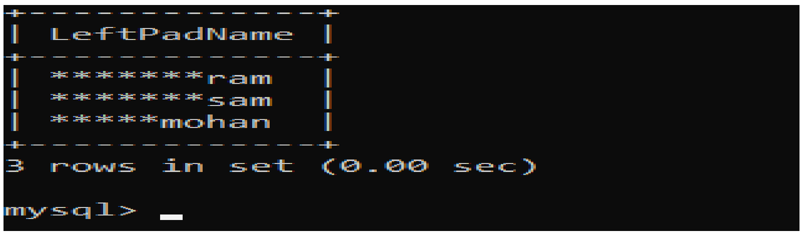
```
+----------------+
| upper(name)    |
+----------------+
| AJMER SHTBDI   |
| MANDOR EXPRESS |
| G T EXPRESS    |
| RAJDHANI       |
| AYODHYA        |
| SHATABDI       |
+----------------+
6 rows in set (0.01 sec)
```

3. Print the passenger names with left padding character.

Syntax:

ELECT LPAD(Name,10,"***") AS LeftPadName From passenger;

Output:



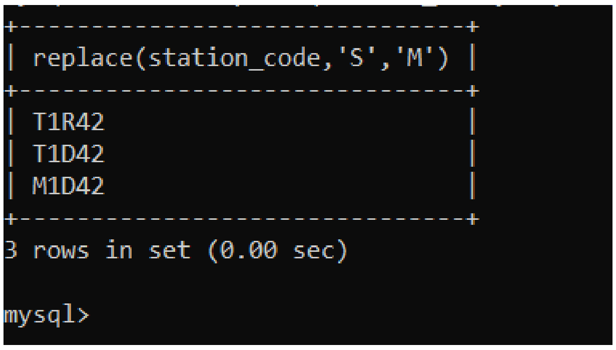
```
+-------------+
| LeftPadName |
+-------------+
| *******ram  |
| *******sam  |
| *****mohan  |
+-------------+
3 rows in set (0.00 sec)

mysql> _
```

4. Print the station codes replacing S with M.

Syntax:

select replace(station_code,'S','M') from Train_Route;

Output:



```
+------------------------------+
| replace(station_code,'S','M') |
+------------------------------+
| T1R42                        |
| T1D42                        |
| M1D42                        |
+------------------------------+
3 rows in set (0.00 sec)

mysql>
```

5. Translate all the LC in class column (Train_fare) to POT and display.

Syntax:

select translate(class,'LC','POT') from Train_ticket_fare;

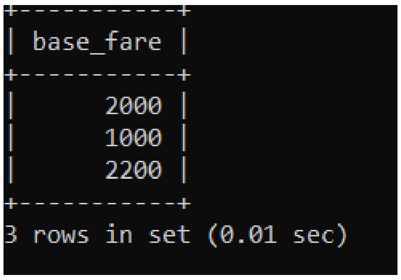
Output:

6. Display the fare details of all trains, if any value is ZERO, print as NULL value.

Syntax:

SELECT NULLIF(base_fare, 0) AS base_fare FROM train_ticket_fare;

Output:



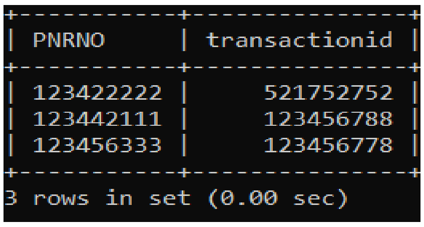
```
+-----------+
| base_fare |
+-----------+
|      2000 |
|      1000 |
|      2200 |
+-----------+
3 rows in set (0.01 sec)
```

7. Display the pnrno and transaction id, if transaction id is null, print 'not generated'.

Synatx:

SELECT pnrno, IF(transactionid IS NULL,'not generated') AS "transactionid" from ticket.

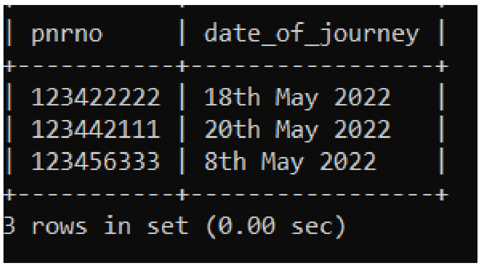
Output:



```
+-----------+---------------+
| PNRNO     | transactionid |
+-----------+---------------+
| 123422222 |     521752752 |
| 123442111 |     123456788 |
| 123456333 |     123456778 |
+-----------+---------------+
3 rows in set (0.00 sec)
```

8. Print the date_of_jounrney in the format '27th November 2010'.

Syntax:
SELECT pnrno,DATE_FORMAT(date_of_journey,'%D %M %Y') as date_of_journey from ticket;
Output:



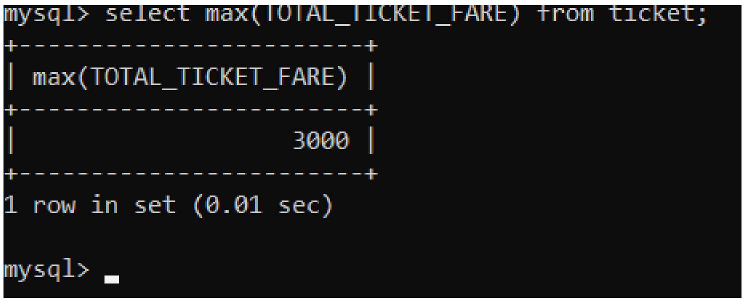
```
| pnrno     | date_of_journey |
+-----------+-----------------+
| 123422222 | 18th May 2022   |
| 123442111 | 20th May 2022   |
| 123456333 | 8th May 2022    |
+-----------+-----------------+
3 rows in set (0.00 sec)
```

9. Find the maximum fare (total fare)

Syntax:

select max(TOTAL_TICKET_FARE) from ticket;

Output:



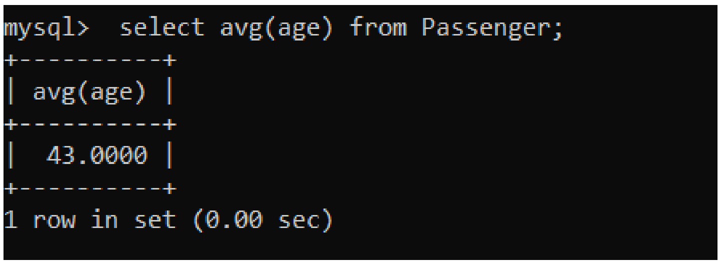
```
mysql> select max(TOTAL_TICKET_FARE) from ticket;
+------------------------+
| max(TOTAL_TICKET_FARE) |
+------------------------+
|                   3000 |
+------------------------+
1 row in set (0.01 sec)

mysql>
```

10. Find the average age of passengers in one ticket.

Syntax:

select avg(age) from Passenger;

Output:



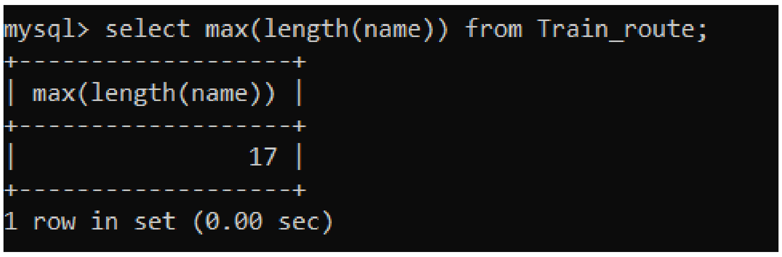
```
mysql>  select avg(age) from Passenger;
+----------+
| avg(age) |
+----------+
|  43.0000 |
+----------+
1 row in set (0.00 sec)
```

11.Find the maximum length of station name available in the database.

Syntax:

select max(length(name)) from Train_route;

Output:

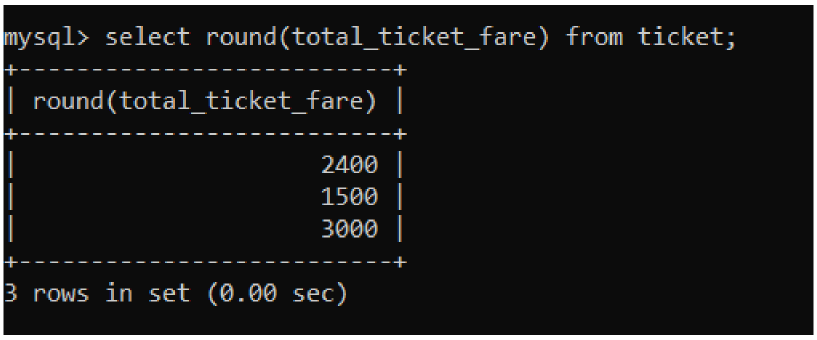

```
mysql> select max(length(name)) from Train_route;
+-------------------+
| max(length(name)) |
+-------------------+
|                17 |
+-------------------+
1 row in set (0.00 sec)
```

12. Print the fare amount of the passengers as rounded value.

Syntax:

select round(total_ticket_fare) from ticket;

Output:

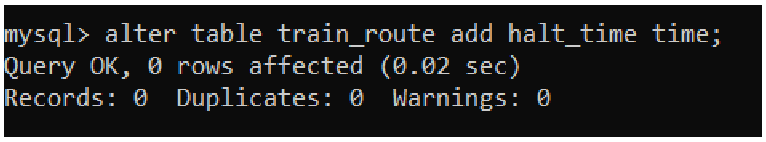

```
mysql> select round(total_ticket_fare) from ticket;
+--------------------------+
| round(total_ticket_fare) |
+--------------------------+
|                     2400 |
|                     1500 |
|                     3000 |
+--------------------------+
3 rows in set (0.00 sec)
```

13. Add the column halt time to train route.

Syntax:

alter table train_route add halt_time time;

Output:


```
mysql> alter table train_route add halt_time time;
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

14. Update values to it from arrival time and depart time.

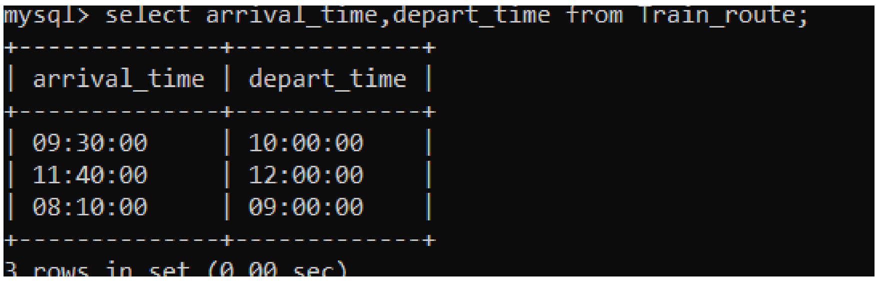
Syntax:
update train_route set halt_time=depart_time-arrival_time;

15. Display the arrival time, depart time in the format HH:MI (24 hours and minutes).

Syntax:

select arrival_time,depart_time from Train_route;

output:



```
mysql> select arrival_time,depart_time from Train_route;
+--------------+-------------+
| arrival_time | depart_time |
+--------------+-------------+
| 09:30:00     | 10:00:00    |
| 11:40:00     | 12:00:00    |
| 08:10:00     | 09:00:00    |
+--------------+-------------+
3 rows in set (0.00 sec)
```

TASK-5

QueryingAggregateFunctions(COUNT,SUM,AVG,MAXandMIN**)**

Aim: ToPracticeQueriesusingAggregatefunctionsforthefollowing

1. WriteaQuerytodisplaytheinformationpresentinthepassengerandcancellationtables
2. Displaythenumberofdaysinaweekon whichtheAP123busisavailable
3. FindnumberofticketsbookedforeachPNR_NousingGROUPBYCLAUSE
4. FindthedistinctPNRNumbersthatarepresent.

1. WriteaQuerytodisplaytheinformationpresentinthepassengerand cancellationtables

MYSQL>CREATETABLECANCELLATION2(PNRNOINTPRIMARYKEY,JOURNEYDATEDATETIME,NOOFSEATS INT,ADDRESS VARCHAR(20),CONTACTNO INT,STATUS VARCHAR(10),FOREIGNKEY(PNRNO)REFERENCESRESERVATION2(PNRNO));

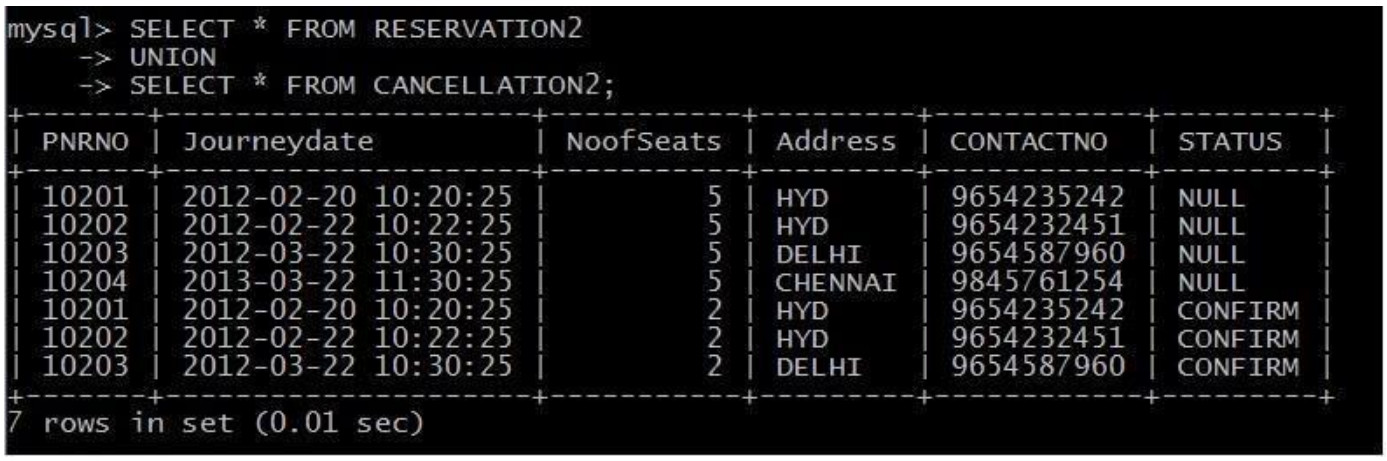
mysql> INSERT INTO CANCELLATION2 VALUES(10201,'2012-02-2010:20:25',2,'HYD',9654235242,'CONFIRM');

```
mysql> INSERT INTO CANCELLATION2 VALUES(10202,'2012-
02-2210:22:25',2,'HYD',9654232451,'CONFIRM');


mysql> INSERT INTO CANCELLATION2 VALUES(10203,'2012-
03-2210:30:25',2,'DELHI',9654587960,'CONFIRM');
```

MySQL>SELECT*

FROMRESERVATIONUNIONSELECT*

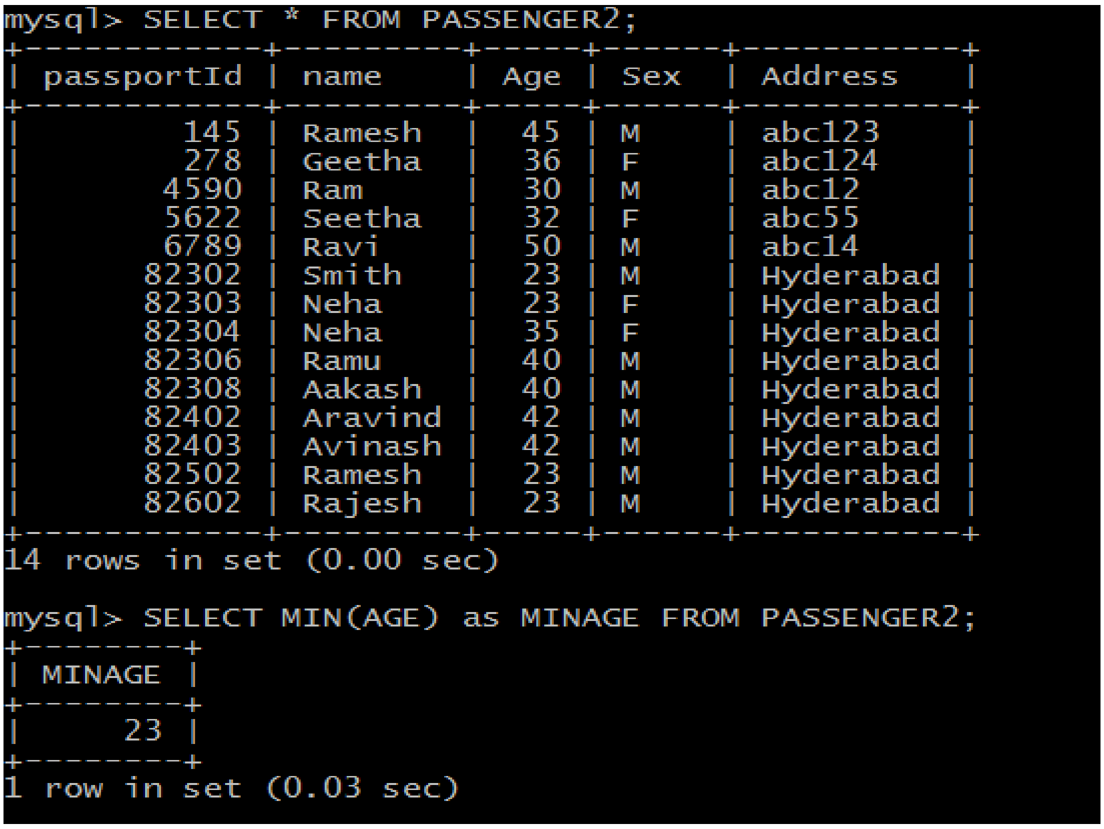
FROMCANCELLATION;



```
mysql> SELECT * FROM RESERVATION2
    -> UNION
    -> SELECT * FROM CANCELLATION2;
+-------+---------------------+-----------+---------+------------+---------+
| PNRNO | Journeydate         | NoofSeats | Address | CONTACTNO  | STATUS  |
+-------+---------------------+-----------+---------+------------+---------+
| 10201 | 2012-02-20 10:20:25 |         5 | HYD     | 9654235242 | NULL    |
| 10202 | 2012-02-22 10:22:25 |         5 | HYD     | 9654232451 | NULL    |
| 10203 | 2012-03-22 10:30:25 |         5 | DELHI   | 9654587960 | NULL    |
| 10204 | 2013-03-22 11:30:25 |         5 | CHENNAI | 9845761254 | NULL    |
| 10201 | 2012-02-20 10:20:25 |         2 | HYD     | 9654235242 | CONFIRM |
| 10202 | 2012-02-22 10:22:25 |         2 | HYD     | 9654232451 | CONFIRM |
| 10203 | 2012-03-22 10:30:25 |         2 | DELHI   | 9654587960 | CONFIRM |
+-------+---------------------+-----------+---------+------------+---------+
7 rows in set (0.01 sec)
```

2. Displaythe MinimumageofthePassenger

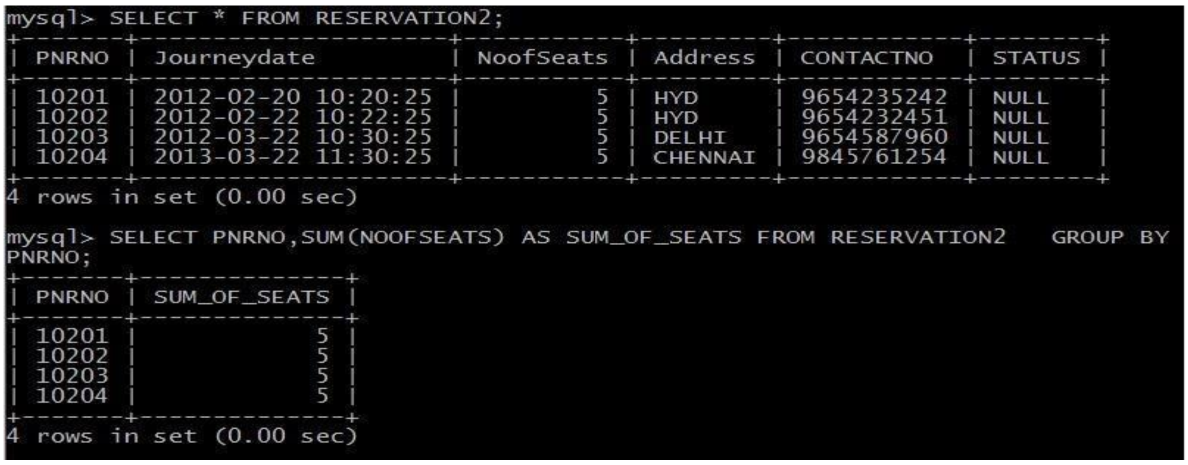
MySQL>SELECTMIN(AGE)asMINAGE FROMPASSENGER;



```
mysql> SELECT * FROM PASSENGER2;
+------------+---------+-----+------+-----------+
| passportId | name    | Age | Sex  | Address   |
+------------+---------+-----+------+-----------+
|        145 | Ramesh  |  45 | M    | abc123    |
|        278 | Geetha  |  36 | F    | abc124    |
|       4590 | Ram     |  30 | M    | abc12     |
|       5622 | Seetha  |  32 | F    | abc55     |
|       6789 | Ravi    |  50 | M    | abc14     |
|      82302 | Smith   |  23 | M    | Hyderabad |
|      82303 | Neha    |  23 | F    | Hyderabad |
|      82304 | Neha    |  35 | F    | Hyderabad |
|      82306 | Ramu    |  40 | M    | Hyderabad |
|      82308 | Aakash  |  40 | M    | Hyderabad |
|      82402 | Aravind |  42 | M    | Hyderabad |
|      82403 | Avinash |  42 | M    | Hyderabad |
|      82502 | Ramesh  |  23 | M    | Hyderabad |
|      82602 | Rajesh  |  23 | M    | Hyderabad |
+------------+---------+-----+------+-----------+
14 rows in set (0.00 sec)

mysql> SELECT MIN(AGE) as MINAGE FROM PASSENGER2;
+--------+
| MINAGE |
+--------+
|     23 |
+--------+
1 row in set (0.03 sec)
```

3. FindnumberofticketsbookedforeachPNR_NousingGROUP BYCLAUSE

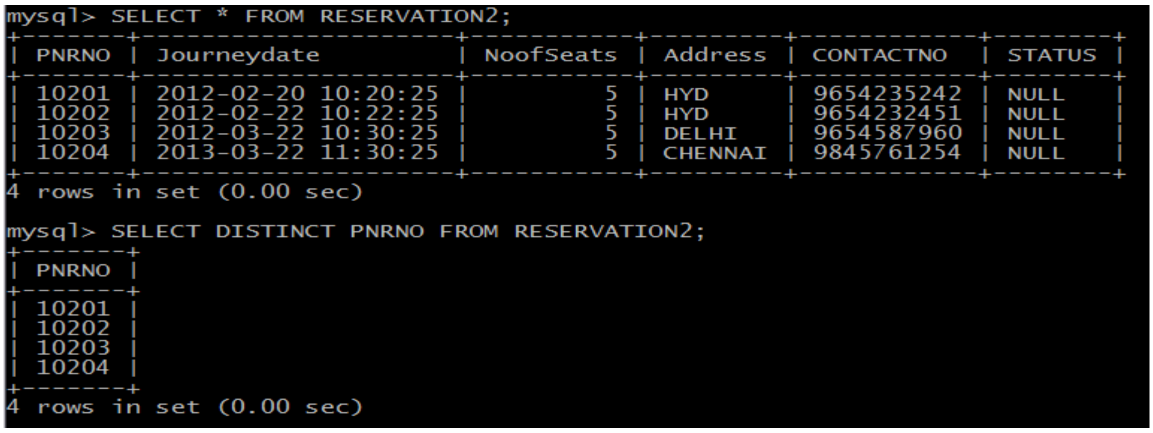
MySQL>SELECTPNRNO,SUM(No_of_SEATS)ASSUM_OF_SEATSFRO
MRESERVATION2 GROUPBY PNRNO;



```
mysql> SELECT * FROM RESERVATION2;
+-------+---------------------+-----------+---------+------------+--------+
| PNRNO | Journeydate         | NoofSeats | Address | CONTACTNO  | STATUS |
+-------+---------------------+-----------+---------+------------+--------+
| 10201 | 2012-02-20 10:20:25 |         5 | HYD     | 9654235242 | NULL   |
| 10202 | 2012-02-22 10:22:25 |         5 | HYD     | 9654232451 | NULL   |
| 10203 | 2012-03-22 10:30:25 |         5 | DELHI   | 9654587960 | NULL   |
| 10204 | 2013-03-22 11:30:25 |         5 | CHENNAI | 9845761254 | NULL   |
+-------+---------------------+-----------+---------+------------+--------+
4 rows in set (0.00 sec)

mysql> SELECT PNRNO,SUM(NOOFSEATS) AS SUM_OF_SEATS FROM RESERVATION2   GROUP BY
PNRNO;
+-------+--------------+
| PNRNO | SUM_OF_SEATS |
+-------+--------------+
| 10201 |            5 |
| 10202 |            5 |
| 10203 |            5 |
| 10204 |            5 |
+-------+--------------+
4 rows in set (0.00 sec)
```

4 Findthe distinct PNRNumbersthat arepresent.

MySQL>SELECTDISTINCTPNR_NOFROM RESERVATION2;

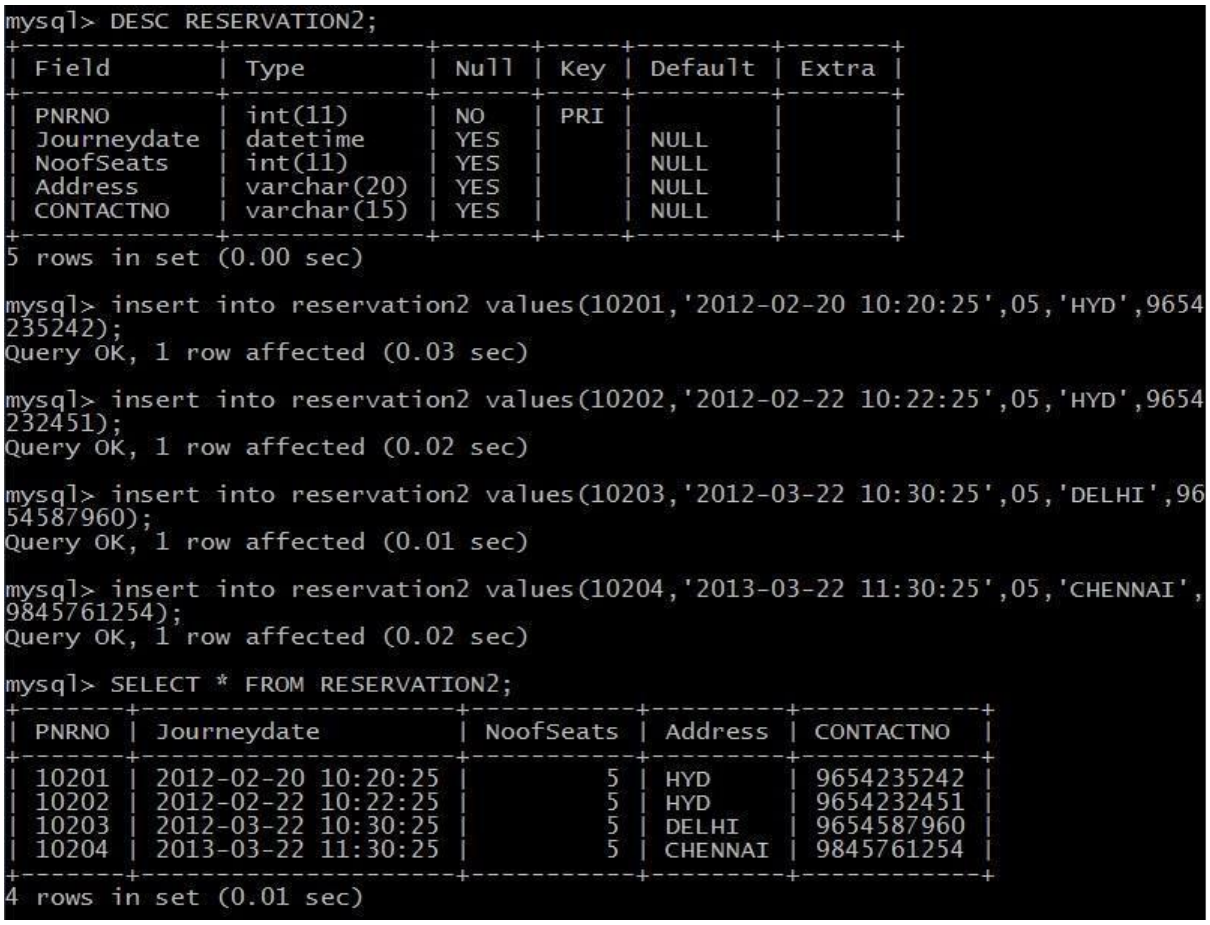


```
mysql> SELECT * FROM RESERVATION2;
+-------+---------------------+-----------+---------+------------+--------+
| PNRNO | Journeydate         | NoofSeats | Address | CONTACTNO  | STATUS |
+-------+---------------------+-----------+---------+------------+--------+
| 10201 | 2012-02-20 10:20:25 |         5 | HYD     | 9654235242 | NULL   |
| 10202 | 2012-02-22 10:22:25 |         5 | HYD     | 9654232451 | NULL   |
| 10203 | 2012-03-22 10:30:25 |         5 | DELHI   | 9654587960 | NULL   |
| 10204 | 2013-03-22 11:30:25 |         5 | CHENNAI | 9845761254 | NULL   |
+-------+---------------------+-----------+---------+------------+--------+
4 rows in set (0.00 sec)

mysql> SELECT DISTINCT PNRNO FROM RESERVATION2;
+-------+
| PNRNO |
+-------+
| 10201 |
| 10202 |
| 10203 |
| 10204 |
+-------+
4 rows in set (0.00 sec)
```

TASK–6

Querying (using ANY, ALL, IN, Exists, NOT EXISTS, UNION, INTERSECT, Constraints etc.)Aim: PracticethefollowingQueries:

1. DisplayuniquePNR_NOofallpassengers
2. Displayallthenamesofmalepassengers.
3. Displaytheticketnumbersandnames of allthepassengers.
4. Findtheticketnumbersofthepassengerswhosenamestart with‘r’and endswith‘h’.
5. FindthenamesofPassengerswhoseageisbetween30and45.
6. Displayall the passengersnamesbeginningwith‘A’.
7. DisplaythesortedlistofPassengersnames


```
mysql> DESC RESERVATION2;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| PNRNO       | int(11)     | NO   | PRI |         |       |
| Journeydate | datetime    | YES  |     | NULL    |       |
| NoofSeats   | int(11)     | YES  |     | NULL    |       |
| Address     | varchar(20) | YES  |     | NULL    |       |
| CONTACTNO   | varchar(15) | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)

mysql> insert into reservation2 values(10201,'2012-02-20 10:20:25',05,'HYD',9654
235242);
Query OK, 1 row affected (0.03 sec)

mysql> insert into reservation2 values(10202,'2012-02-22 10:22:25',05,'HYD',9654
232451);
Query OK, 1 row affected (0.02 sec)

mysql> insert into reservation2 values(10203,'2012-03-22 10:30:25',05,'DELHI',96
54587960);
Query OK, 1 row affected (0.01 sec)

mysql> insert into reservation2 values(10204,'2013-03-22 11:30:25',05,'CHENNAI',
9845761254);
Query OK, 1 row affected (0.02 sec)

mysql> SELECT * FROM RESERVATION2;
+-------+---------------------+-----------+---------+------------+
| PNRNO | Journeydate         | NoofSeats | Address | CONTACTNO  |
+-------+---------------------+-----------+---------+------------+
| 10201 | 2012-02-20 10:20:25 |         5 | HYD     | 9654235242 |
| 10202 | 2012-02-22 10:22:25 |         5 | HYD     | 9654232451 |
| 10203 | 2012-03-22 10:30:25 |         5 | DELHI   | 9654587960 |
| 10204 | 2013-03-22 11:30:25 |         5 | CHENNAI | 9845761254 |
+-------+---------------------+-----------+---------+------------+
4 rows in set (0.01 sec)
```

```
mysql>insertintopassenger2values(82302,'Smith',23,'M','Hyderabad');Q
ueryOK, 1rowaffected (0.02sec)

mysql> insert into passenger2
values(82303,'Neha',23,'F','Hyderabad');QueryOK, 1rowaffected
(0.01sec)

mysql>insertintopassenger2values(82304,'Neha',35,'F','Hyderabad');Que
ryOK, 1rowaffected (0.03sec)

mysql>insertintopassenger2values(82306,'Ramu',40,'M','Hyderabad');Q
ueryOK, 1rowaffected (0.02sec)

mysql>insertintopassenger2values(82308,'Aakash',40,'M','Hyderabad');
QueryOK, 1rowaffected (0.02sec)

mysql>insertintopassenger2values(82402,'Aravind',42,'M','Hyderabad');
QueryOK, 1rowaffected (0.02sec)

mysql>insertintopassenger2values(82403,'Avinash',42,'M','Hyderabad');
QueryOK, 1rowaffected (0.02sec)

mysql>insertintopassenger2values(82502,'Ramesh',23,'M','Hyderabad');
QueryOK, 1rowaffected (0.02sec)

mysql>insertintopassenger2values(82602,'Rajesh',23,'M','Hyderabad');Q
 ueryOK, 1rowaffected (0.02sec)
```

RESERVATION2

mysql>insertintoreservation2values(10201,'2012-02-

2010:20:25',05,'HYD',9654235242);QueryOK, 1rowaffected (0.03 sec)

mysql>insertintoreservation2values(10202,'2012-02-

2210:22:25',05,'HYD',9654232451);QueryOK, 1rowaffected (0.02 sec)

mysql> insert into reservation2 values(10203,'2012-03-22 10:30:25',05,'DELHI',96

54587960);QueryOK, 1rowaffected (0.01 sec)

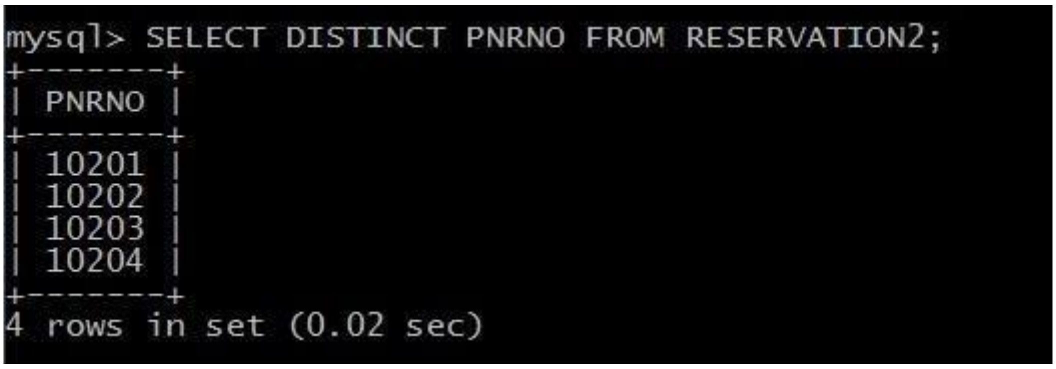
mysql>insertintoreservation2values(10204,'2013-03-

2211:30:25',05,'CHENNAI',9845761254);QueryOK, 1rowaffected (0.02 sec)

1. DisplayuniquePNR_NOofallreservationMysql>Select

   DISTINCTPNR_NO fromReservation;

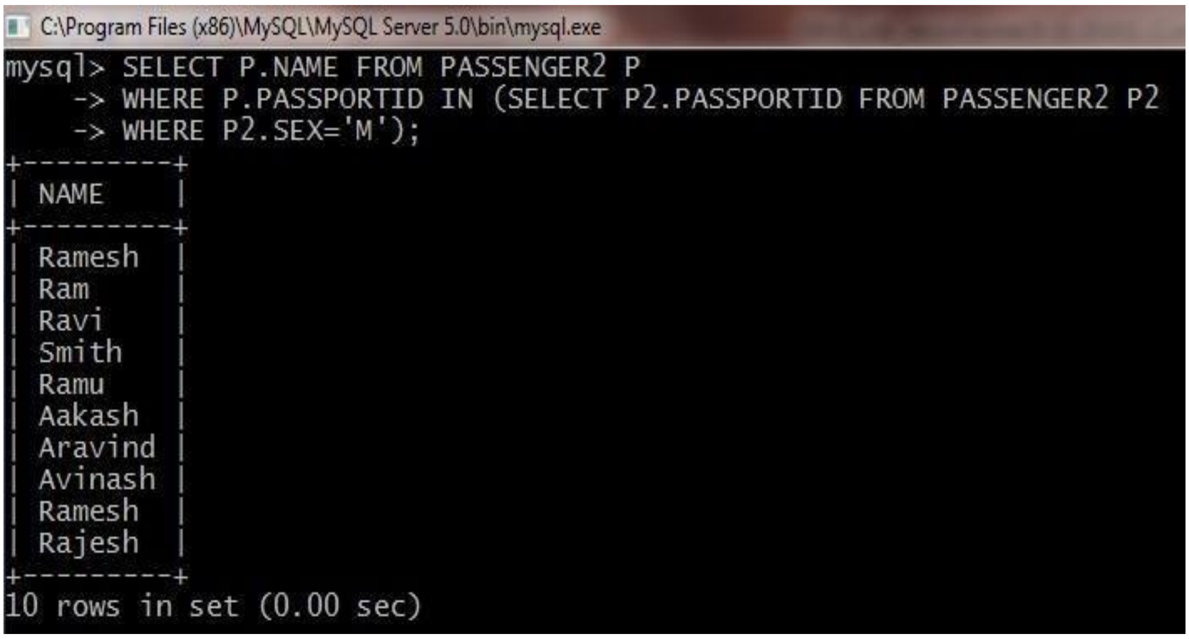
PNR_No
10201
10202
10203
10204



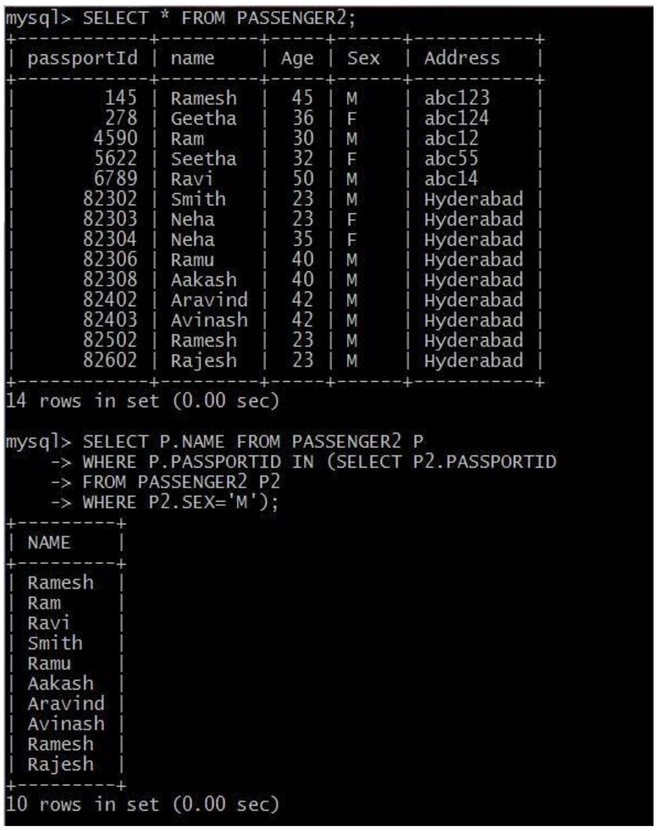
```
mysql> SELECT DISTINCT PNRNO FROM RESERVATION2;
+-------+
| PNRNO |
+-------+
| 10201 |
| 10202 |
| 10203 |
| 10204 |
+-------+
4 rows in set (0.02 sec)
```

2. Displayallthe namesofmalepassengers.

```
mysql>Selectp.namefrompassenger2p
        where
                 p.passportidIN(selectp2.passportidfrompassenger2p2
         where    p2.sex='M');
```

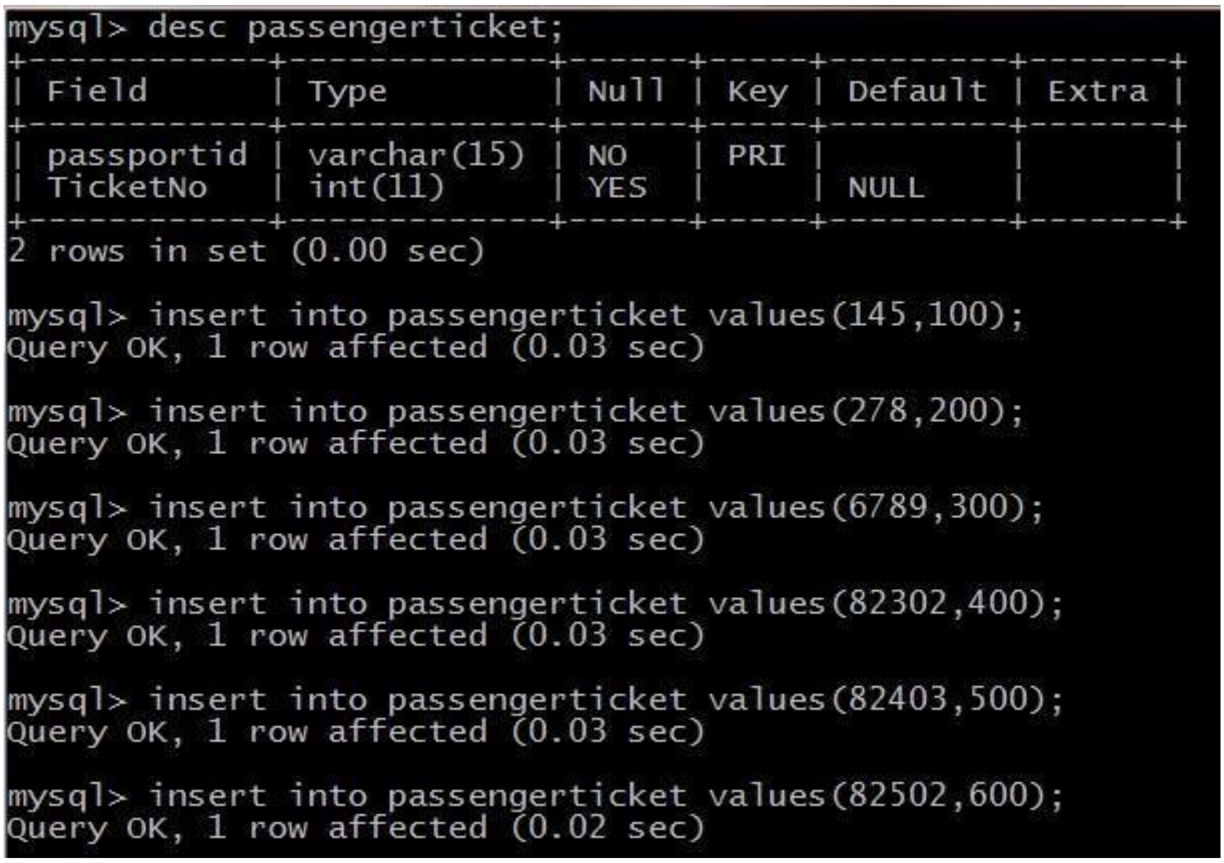


```
C:\Program Files (x86)\MySQL\MySQL Server 5.0\bin\mysql.exe
mysql> SELECT P.NAME FROM PASSENGER2 P
    -> WHERE P.PASSPORTID IN (SELECT P2.PASSPORTID FROM PASSENGER2 P2
    -> WHERE P2.SEX='M');
+---------+
| NAME    |
+---------+
| Ramesh  |
| Ram     |
| Ravi    |
| Smith   |
| Ramu    |
| Aakash  |
| Aravind |
| Avinash |
| Ramesh  |
| Rajesh  |
+---------+
10 rows in set (0.00 sec)
```



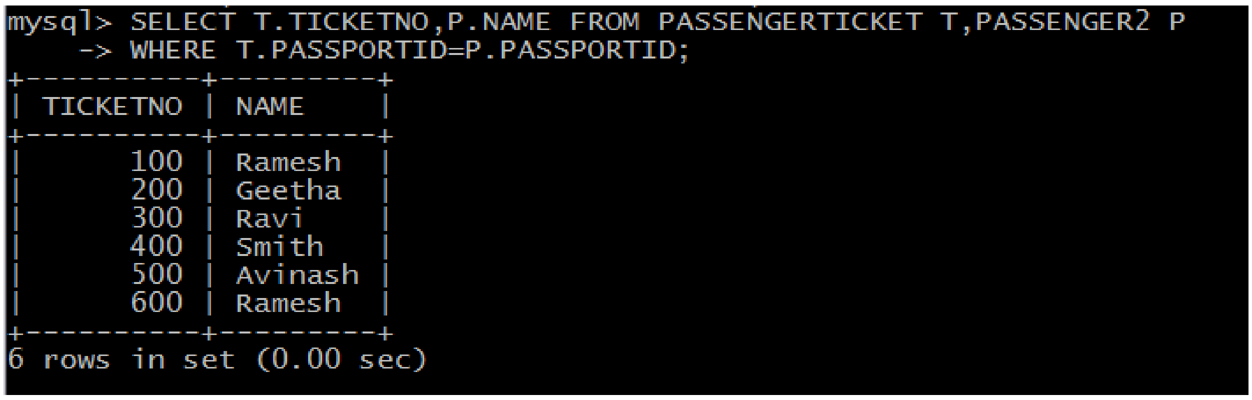
```
mysql> SELECT * FROM PASSENGER2;
+------------+---------+-----+------+-----------+
| passportId | name    | Age | Sex  | Address   |
+------------+---------+-----+------+-----------+
|        145 | Ramesh  |  45 | M    | abc123    |
|        278 | Geetha  |  36 | F    | abc124    |
|       4590 | Ram     |  30 | M    | abc12     |
|       5622 | Seetha  |  32 | F    | abc55     |
|       6789 | Ravi    |  50 | M    | abc14     |
|      82302 | Smith   |  23 | M    | Hyderabad |
|      82303 | Neha    |  23 | F    | Hyderabad |
|      82304 | Neha    |  35 | F    | Hyderabad |
|      82306 | Ramu    |  40 | M    | Hyderabad |
|      82308 | Aakash  |  40 | M    | Hyderabad |
|      82402 | Aravind |  42 | M    | Hyderabad |
|      82403 | Avinash |  42 | M    | Hyderabad |
|      82502 | Ramesh  |  23 | M    | Hyderabad |
|      82602 | Rajesh  |  23 | M    | Hyderabad |
+------------+---------+-----+------+-----------+
14 rows in set (0.00 sec)

mysql> SELECT P.NAME FROM PASSENGER2 P
    -> WHERE P.PASSPORTID IN (SELECT P2.PASSPORTID
    -> FROM PASSENGER2 P2
    -> WHERE P2.SEX='M');
+---------+
| NAME    |
+---------+
| Ramesh  |
| Ram     |
| Ravi    |
| Smith   |
| Ramu    |
| Aakash  |
| Aravind |
| Avinash |
| Ramesh  |
| Rajesh  |
+---------+
10 rows in set (0.00 sec)
```

3. Displaytheticketnumbersandnamesofallthepassengers.



```
mysql> desc passengerticket;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| passportid | varchar(15) | NO   | PRI |         |       |
| TicketNo   | int(11)     | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)

mysql> insert into passengerticket values(145,100);
Query OK, 1 row affected (0.03 sec)

mysql> insert into passengerticket values(278,200);
Query OK, 1 row affected (0.03 sec)

mysql> insert into passengerticket values(6789,300);
Query OK, 1 row affected (0.03 sec)

mysql> insert into passengerticket values(82302,400);
Query OK, 1 row affected (0.03 sec)

mysql> insert into passengerticket values(82403,500);
Query OK, 1 row affected (0.03 sec)

mysql> insert into passengerticket values(82502,600);
Query OK, 1 row affected (0.02 sec)
```

mysql>selectt.ticketno,p.namefrompassengertickett,passenger2pwheret.passportid=p.passportid;

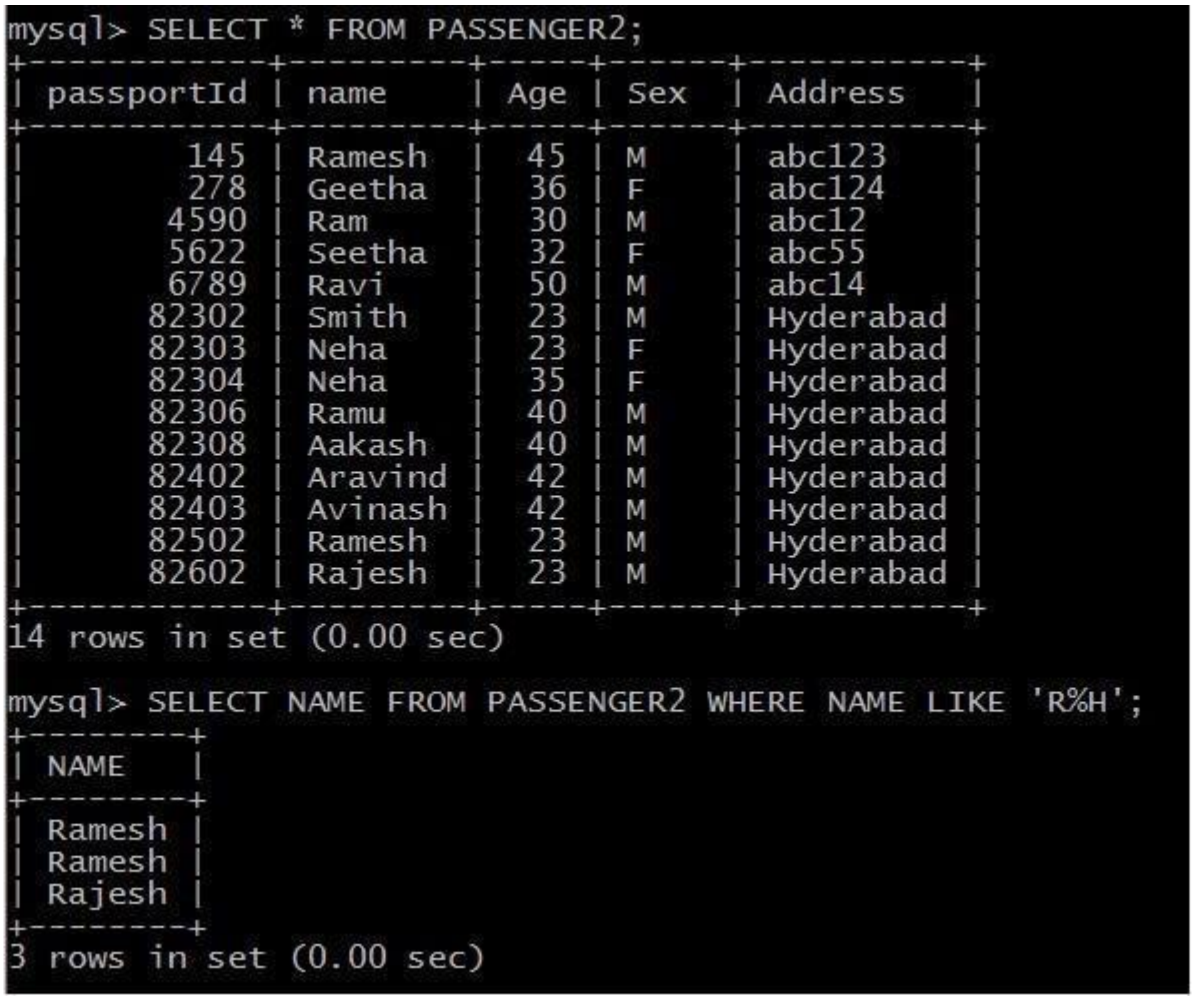


```
mysql> SELECT T.TICKETNO,P.NAME FROM PASSENGERTICKET T,PASSENGER2 P
    -> WHERE T.PASSPORTID=P.PASSPORTID;
+----------+---------+
| TICKETNO | NAME    |
+----------+---------+
|      100 | Ramesh  |
|      200 | Geetha  |
|      300 | Ravi    |
|      400 | Smith   |
|      500 | Avinash |
|      600 | Ramesh  |
+----------+---------+
6 rows in set (0.00 sec)
```

4. Findtheticketnumbersofthe passengerswhosenamestartwith‘r’andendswith‘h’.

MySQL>SELECTNameFROMPassengerWHEREnameLIKE‘R%H’

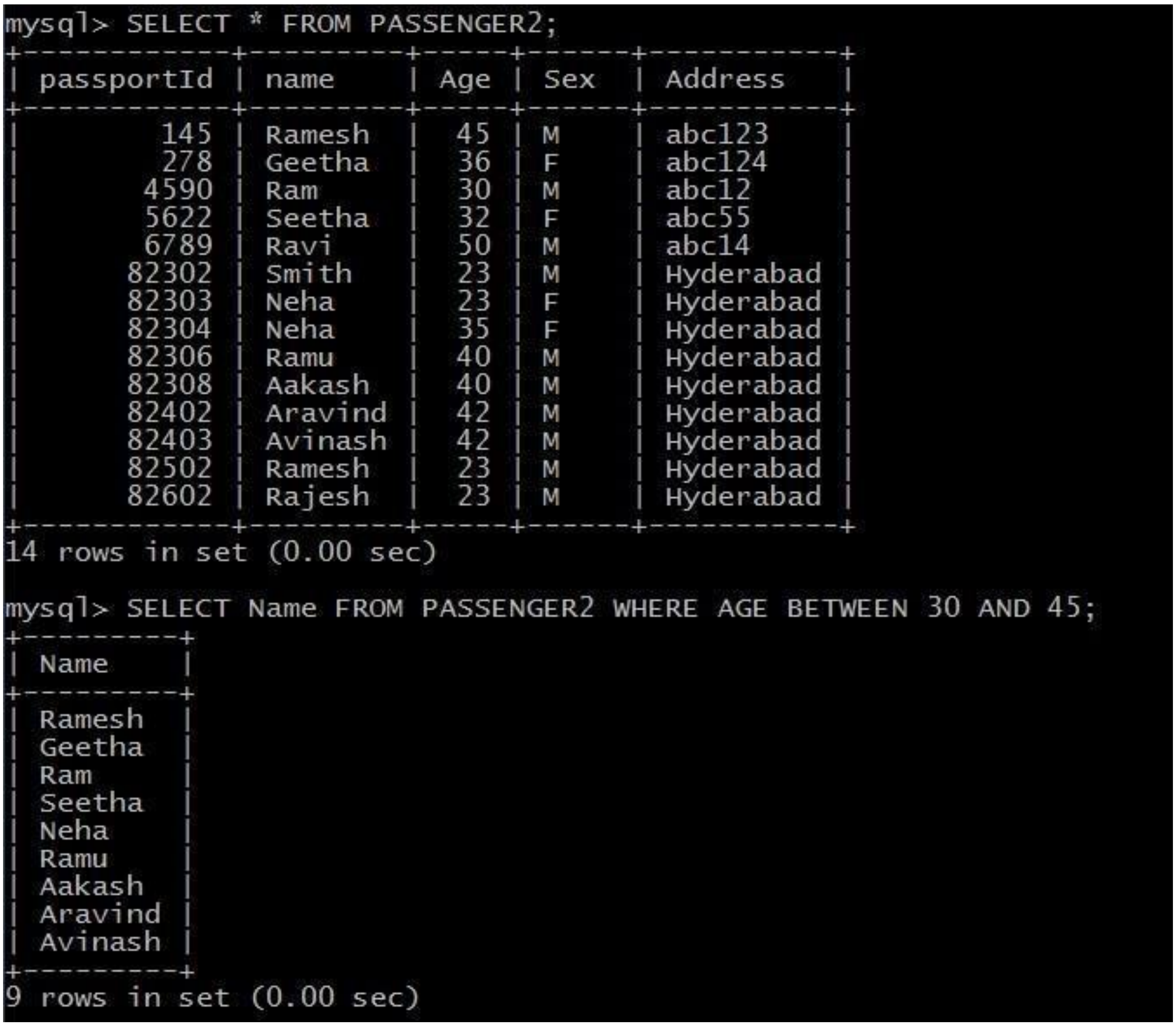
Name
Rajesh
Ramesh
Ramesh



```
mysql> SELECT * FROM PASSENGER2;
+------------+---------+-----+------+-----------+
| passportId | name    | Age | Sex  | Address   |
+------------+---------+-----+------+-----------+
|        145 | Ramesh  |  45 | M    | abc123    |
|        278 | Geetha  |  36 | F    | abc124    |
|       4590 | Ram     |  30 | M    | abc12     |
|       5622 | Seetha  |  32 | F    | abc55     |
|       6789 | Ravi    |  50 | M    | abc14     |
|      82302 | Smith   |  23 | M    | Hyderabad |
|      82303 | Neha    |  23 | F    | Hyderabad |
|      82304 | Neha    |  35 | F    | Hyderabad |
|      82306 | Ramu    |  40 | M    | Hyderabad |
|      82308 | Aakash  |  40 | M    | Hyderabad |
|      82402 | Aravind |  42 | M    | Hyderabad |
|      82403 | Avinash |  42 | M    | Hyderabad |
|      82502 | Ramesh  |  23 | M    | Hyderabad |
|      82602 | Rajesh  |  23 | M    | Hyderabad |
+------------+---------+-----+------+-----------+
14 rows in set (0.00 sec)

mysql> SELECT NAME FROM PASSENGER2 WHERE NAME LIKE 'R%H';
+--------+
| NAME   |
+--------+
| Ramesh |
| Ramesh |
| Rajesh |
+--------+
3 rows in set (0.00 sec)
```

5. Findthe namesofPassengerswhoseageisbetween30 and45.

MySQL>SELECTNameFROMPASSENGERWHEREAGEBETWEEN30AND45

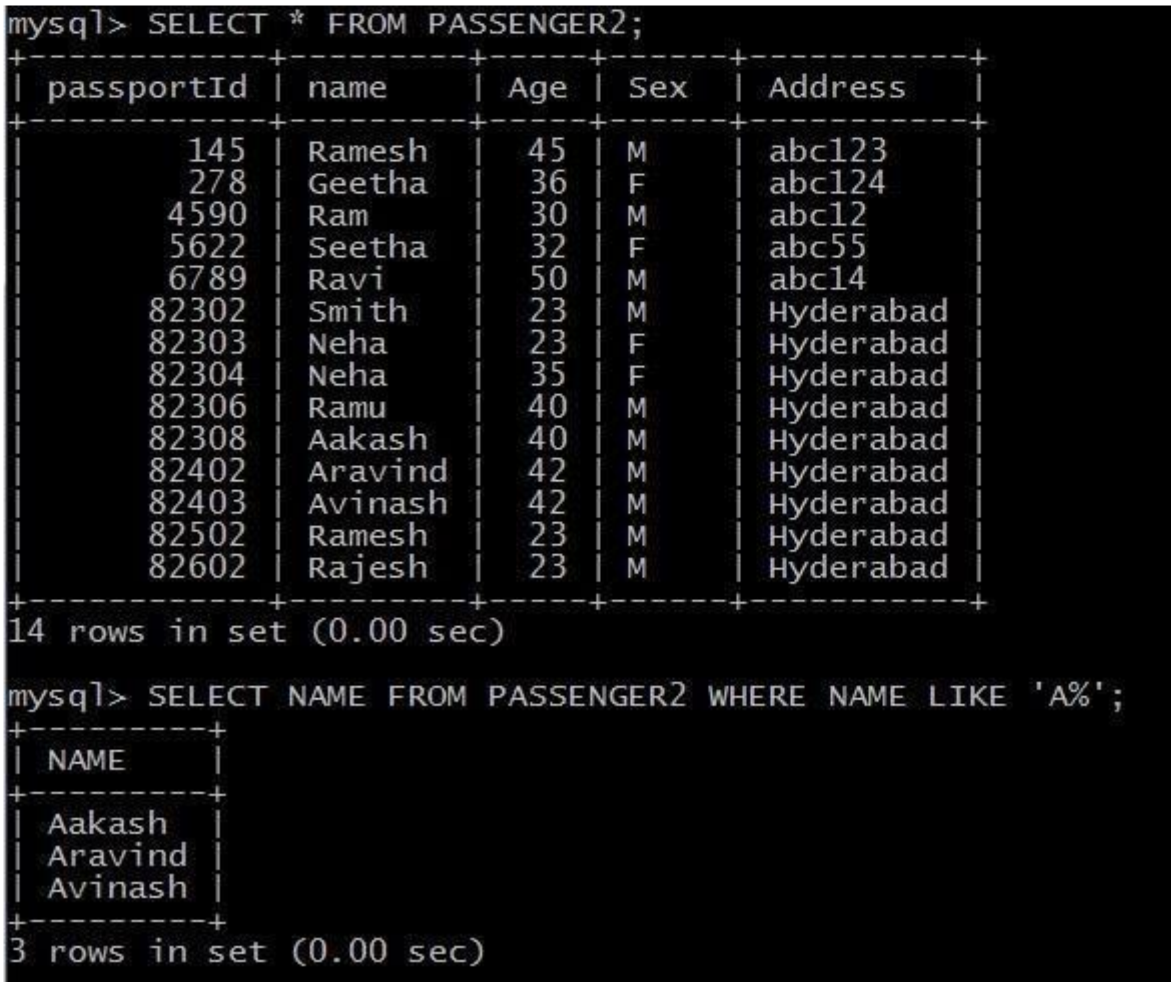


```
mysql> SELECT * FROM PASSENGER2;
+------------+---------+-----+------+-----------+
| passportId | name    | Age | Sex  | Address   |
+------------+---------+-----+------+-----------+
|        145 | Ramesh  |  45 | M    | abc123    |
|        278 | Geetha  |  36 | F    | abc124    |
|       4590 | Ram     |  30 | M    | abc12     |
|       5622 | Seetha  |  32 | F    | abc55     |
|       6789 | Ravi    |  50 | M    | abc14     |
|      82302 | Smith   |  23 | M    | Hyderabad |
|      82303 | Neha    |  23 | F    | Hyderabad |
|      82304 | Neha    |  35 | F    | Hyderabad |
|      82306 | Ramu    |  40 | M    | Hyderabad |
|      82308 | Aakash  |  40 | M    | Hyderabad |
|      82402 | Aravind |  42 | M    | Hyderabad |
|      82403 | Avinash |  42 | M    | Hyderabad |
|      82502 | Ramesh  |  23 | M    | Hyderabad |
|      82602 | Rajesh  |  23 | M    | Hyderabad |
+------------+---------+-----+------+-----------+
14 rows in set (0.00 sec)

mysql> SELECT Name FROM PASSENGER2 WHERE AGE BETWEEN 30 AND 45;
+---------+
| Name    |
+---------+
| Ramesh  |
| Geetha  |
| Ram     |
| Seetha  |
| Neha    |
| Ramu    |
| Aakash  |
| Aravind |
| Avinash |
+---------+
9 rows in set (0.00 sec)
```

6. Displayallthepassengersnamesbeginningwith‘A’.

MySQL>SELECT*FROMPASSENGERWHERENAME LIKE‘A%’;

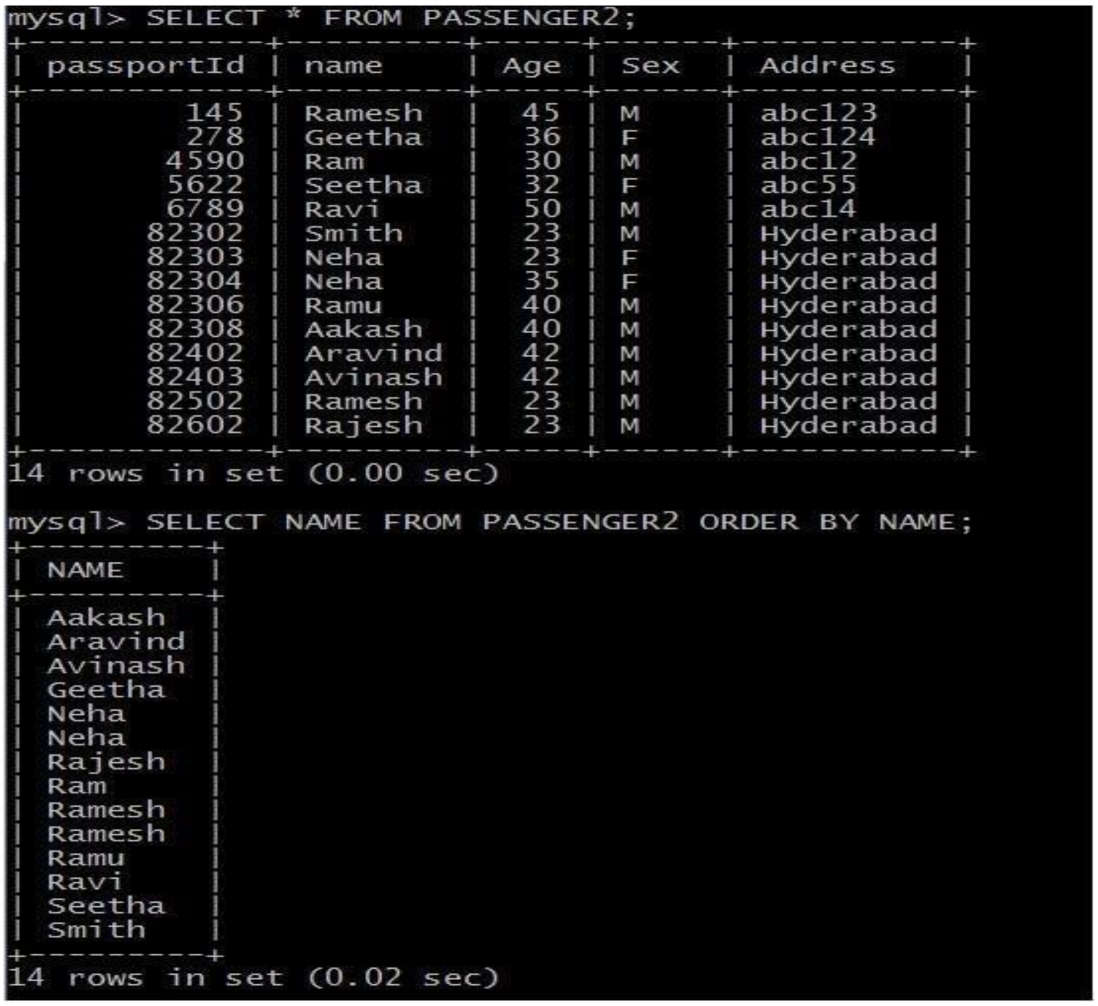
Name
Akash
Arivind
Avinash



```
mysql> SELECT * FROM PASSENGER2;
+------------+---------+-----+------+-----------+
| passportId | name    | Age | Sex  | Address   |
+------------+---------+-----+------+-----------+
|        145 | Ramesh  |  45 | M    | abc123    |
|        278 | Geetha  |  36 | F    | abc124    |
|       4590 | Ram     |  30 | M    | abc12     |
|       5622 | Seetha  |  32 | F    | abc55     |
|       6789 | Ravi    |  50 | M    | abc14     |
|      82302 | Smith   |  23 | M    | Hyderabad |
|      82303 | Neha    |  23 | F    | Hyderabad |
|      82304 | Neha    |  35 | F    | Hyderabad |
|      82306 | Ramu    |  40 | M    | Hyderabad |
|      82308 | Aakash  |  40 | M    | Hyderabad |
|      82402 | Aravind |  42 | M    | Hyderabad |
|      82403 | Avinash |  42 | M    | Hyderabad |
|      82502 | Ramesh  |  23 | M    | Hyderabad |
|      82602 | Rajesh  |  23 | M    | Hyderabad |
+------------+---------+-----+------+-----------+
14 rows in set (0.00 sec)

mysql> SELECT NAME FROM PASSENGER2 WHERE NAME LIKE 'A%';
+---------+
| NAME    |
+---------+
| Aakash  |
| Aravind |
| Avinash |
+---------+
3 rows in set (0.00 sec)
```

7. DisplaythesortedlistofPassengersnames

MySQL>SELECTNAMEFROMPASSENGERORDERBYNAME;



```
mysql> SELECT * FROM PASSENGER2;
+------------+---------+-----+------+-----------+
| passportId | name    | Age | Sex  | Address   |
+------------+---------+-----+------+-----------+
|        145 | Ramesh  |  45 | M    | abc123    |
|        278 | Geetha  |  36 | F    | abc124    |
|       4590 | Ram     |  30 | M    | abc12     |
|       5622 | Seetha  |  32 | F    | abc55     |
|       6789 | Ravi    |  50 | M    | abc14     |
|      82302 | Smith   |  23 | M    | Hyderabad |
|      82303 | Neha    |  23 | F    | Hyderabad |
|      82304 | Neha    |  35 | F    | Hyderabad |
|      82306 | Ramu    |  40 | M    | Hyderabad |
|      82308 | Aakash  |  40 | M    | Hyderabad |
|      82402 | Aravind |  42 | M    | Hyderabad |
|      82403 | Avinash |  42 | M    | Hyderabad |
|      82502 | Ramesh  |  23 | M    | Hyderabad |
|      82602 | Rajesh  |  23 | M    | Hyderabad |
+------------+---------+-----+------+-----------+
14 rows in set (0.00 sec)

mysql> SELECT NAME FROM PASSENGER2 ORDER BY NAME;
+---------+
| NAME    |
+---------+
| Aakash  |
| Aravind |
| Avinash |
| Geetha  |
| Neha    |
| Neha    |
| Rajesh  |
| Ram     |
| Ramesh  |
| Ramesh  |
| Ramu    |
| Ravi    |
| Seetha  |
| Smith   |
+---------+
14 rows in set (0.02 sec)
```

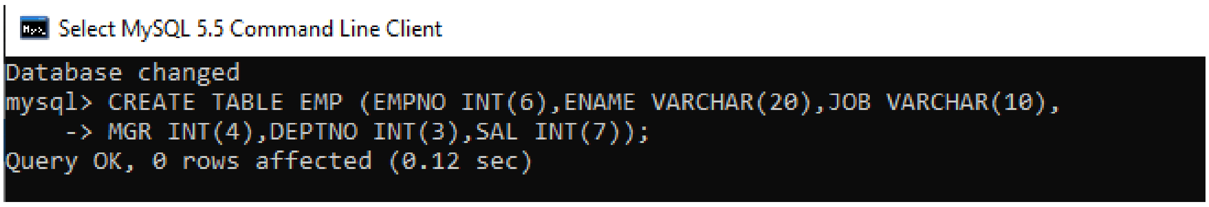
TASK 7:

Create a table EMP with the following structure.

COLUMN Name	DATA Type
EMPNO	INTEGER(6)
ENAME	VARCHAR2(20)
JOB	VARCHAR2(10)
MGR	INTEGER (4)
DEPTNO	INTEGER (3)
SAL	INTEGER (7)

1.

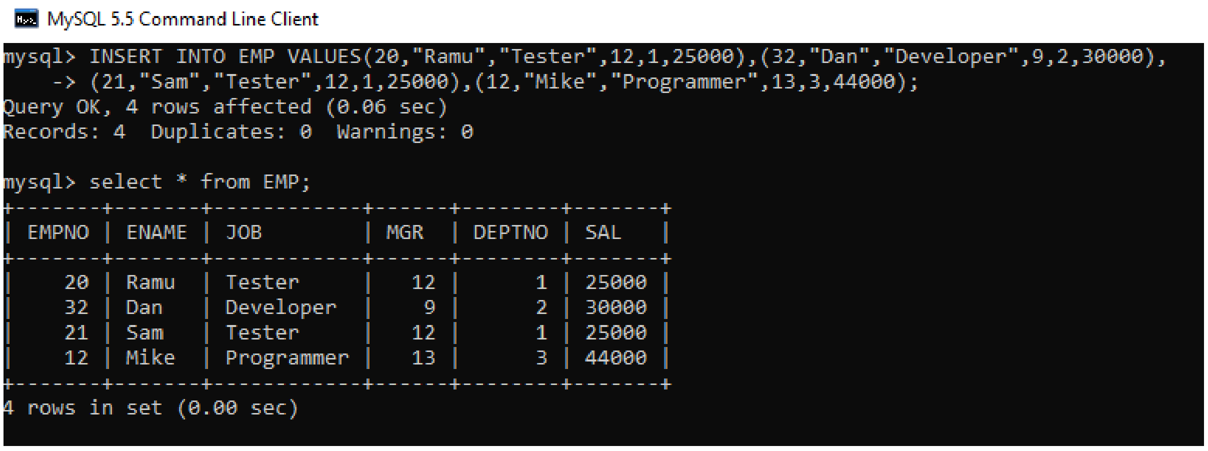
Creating Table Emp:



Select MySQL 5.5 Command Line Client

```
Database changed
mysql> CREATE TABLE EMP (EMPNO INT(6),ENAME VARCHAR(20),JOB VARCHAR(10),
    -> MGR INT(4),DEPTNO INT(3),SAL INT(7));
Query OK, 0 rows affected (0.12 sec)
```

Inserting the values into the table:



MySQL 5.5 Command Line Client

```
mysql> INSERT INTO EMP VALUES(20,"Ramu","Tester",12,1,25000),(32,"Dan","Developer",9,2,30000),
    -> (21,"Sam","Tester",12,1,25000),(12,"Mike","Programmer",13,3,44000);
Query OK, 4 rows affected (0.06 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> select * from EMP;
+-------+-------+------------+------+--------+-------+
| EMPNO | ENAME | JOB        | MGR  | DEPTNO | SAL   |
+-------+-------+------------+------+--------+-------+
|    20 | Ramu  | Tester     |   12 |      1 | 25000 |
|    32 | Dan   | Developer  |    9 |      2 | 30000 |
|    21 | Sam   | Tester     |   12 |      1 | 25000 |
|    12 | Mike  | Programmer |   13 |      3 | 44000 |
+-------+-------+------------+------+--------+-------+
4 rows in set (0.00 sec)
```

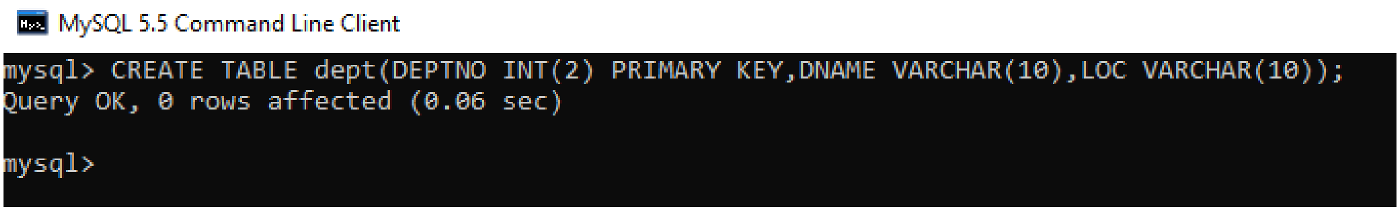
Create dept table with the following structure.

COLUMN Name	DATA Type
DEPTNO	INTEGER (2)
DNAME	VARCHAR2(10)
LOC	VARCHAR2(10)

DEPTNO as the primary key

2.

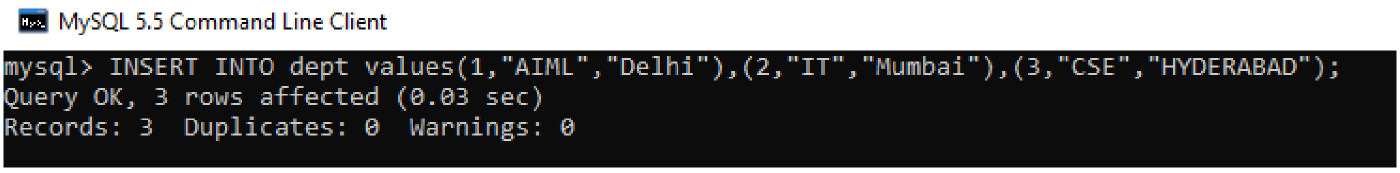
Creating Table Dept:



MySQL 5.5 Command Line Client

```
mysql> CREATE TABLE dept(DEPTNO INT(2) PRIMARY KEY,DNAME VARCHAR(10),LOC VARCHAR(10));
Query OK, 0 rows affected (0.06 sec)

mysql>
```

Inserting the values into the table:

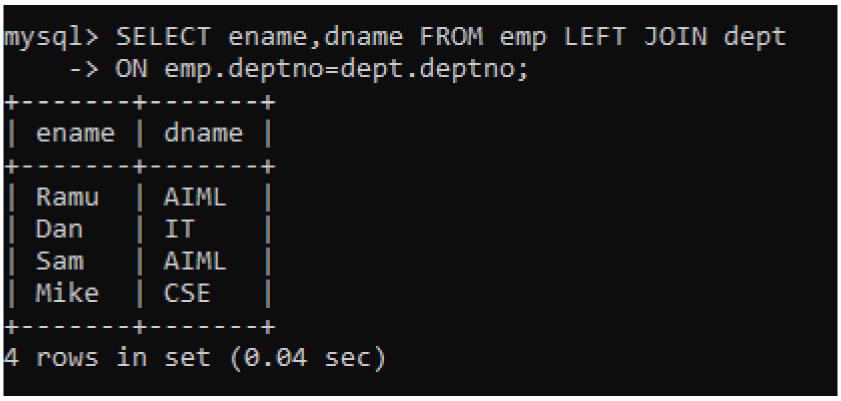


MySQL 5.5 Command Line Client

```
mysql> INSERT INTO dept values(1,"AIML","Delhi"),(2,"IT","Mumbai"),(3,"CSE","HYDERABAD");
Query OK, 3 rows affected (0.03 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

Queries:

5. Display all the employees and the departments implementing a left outer join.



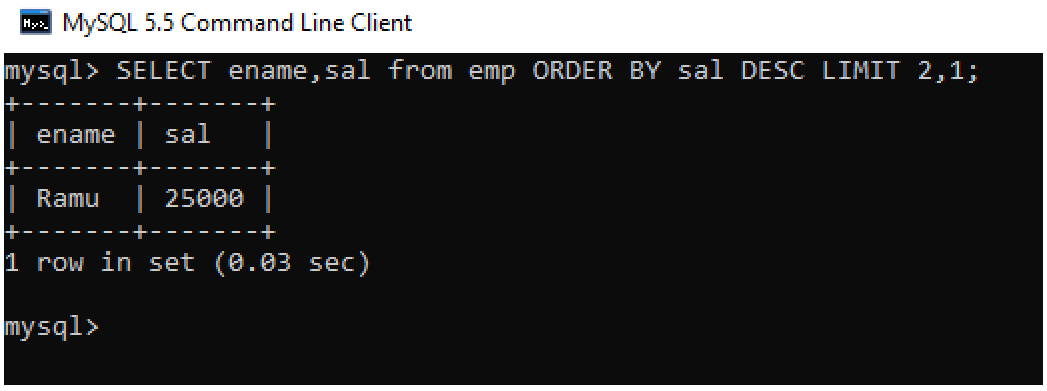
```
mysql> SELECT ename,dname FROM emp LEFT JOIN dept
    -> ON emp.deptno=dept.deptno;
+-------+-------+
| ename | dname |
+-------+-------+
| Ramu  | AIML  |
| Dan   | IT    |
| Sam   | AIML  |
| Mike  | CSE   |
+-------+-------+
4 rows in set (0.04 sec)
```

6. Display the employee name and department name in which they are working implementing a full outer join.

// MySQL does not support full outer join out of the box, unlike other databases.

```
SELECT emame,dname FROM emp

LEFT JOIN dept ON emp.deptno = dept.deptno

UNION ALL

SELECT ename,dname FROM emp

RIGHT JOIN dept ON emp.deptno = dept.deptno

WHERE emp.deptno IS NULL;
```

7. Find the third highest salary of an employee.

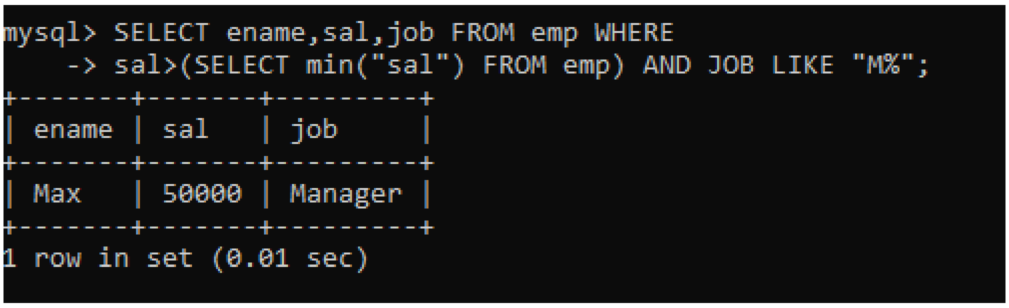


MySQL 5.5 Command Line Client

```
mysql> SELECT ename,sal from emp ORDER BY sal DESC LIMIT 2,1;
+-------+-------+
| ename | sal   |
+-------+-------+
| Ramu  | 25000 |
+-------+-------+
1 row in set (0.03 sec)

mysql>
```

8. Display all employee names and salary whose salary is greater than minimum salary of the company and job title starts with ‘M’.

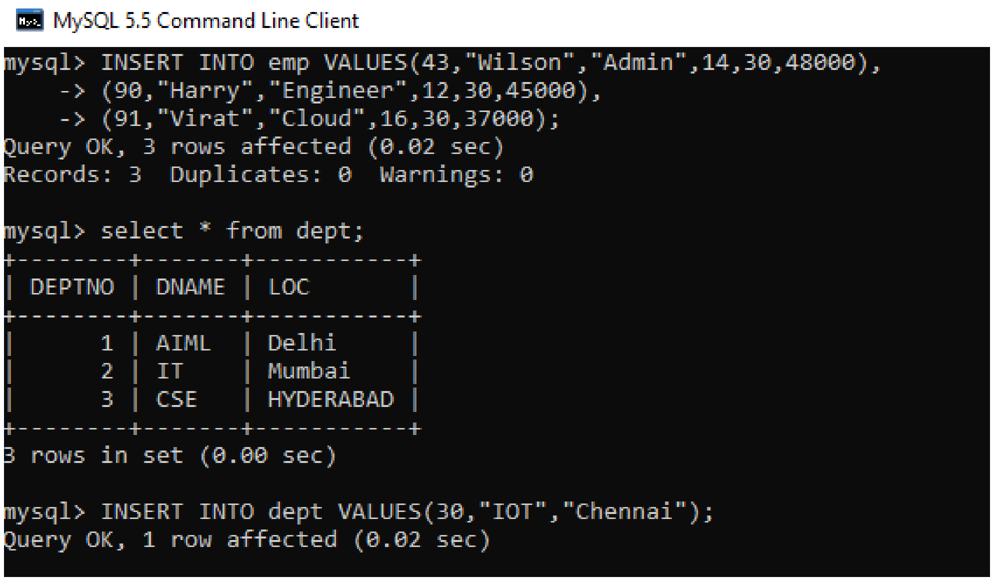
Insert the Manger record and display the details:



```
mysql> SELECT ename,sal,job FROM emp WHERE
    -> sal>(SELECT min("sal") FROM emp) AND JOB LIKE "M%";
+-------+-------+---------+
| ename | sal   | job     |
+-------+-------+---------+
| Max   | 50000 | Manager |
+-------+-------+---------+
1 row in set (0.01 sec)
```

5. Write a query to display information about employees who earn more than any employee in dept 30.

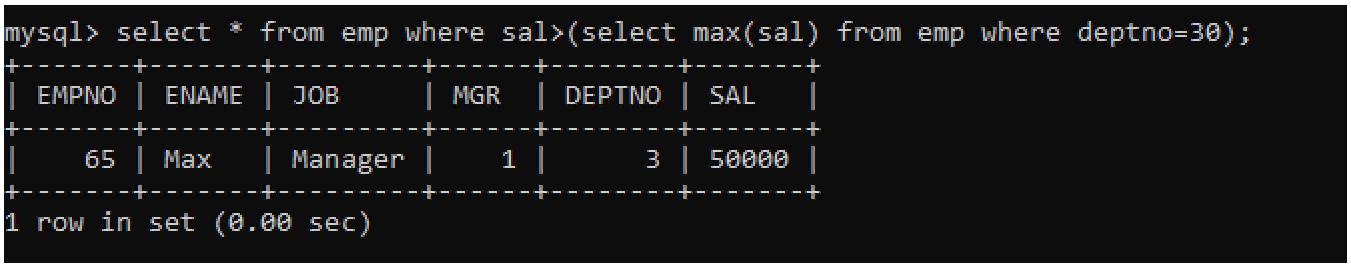
Inserting more values:



MySQL 5.5 Command Line Client

```
mysql> INSERT INTO emp VALUES(43,"Wilson","Admin",14,30,48000),
    -> (90,"Harry","Engineer",12,30,45000),
    -> (91,"Virat","Cloud",16,30,37000);
Query OK, 3 rows affected (0.02 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> select * from dept;
+--------+-------+-----------+
| DEPTNO | DNAME | LOC       |
+--------+-------+-----------+
|      1 | AIML  | Delhi     |
|      2 | IT    | Mumbai    |
|      3 | CSE   | HYDERABAD |
+--------+-------+-----------+
3 rows in set (0.00 sec)

mysql> INSERT INTO dept VALUES(30,"IOT","Chennai");
Query OK, 1 row affected (0.02 sec)
```

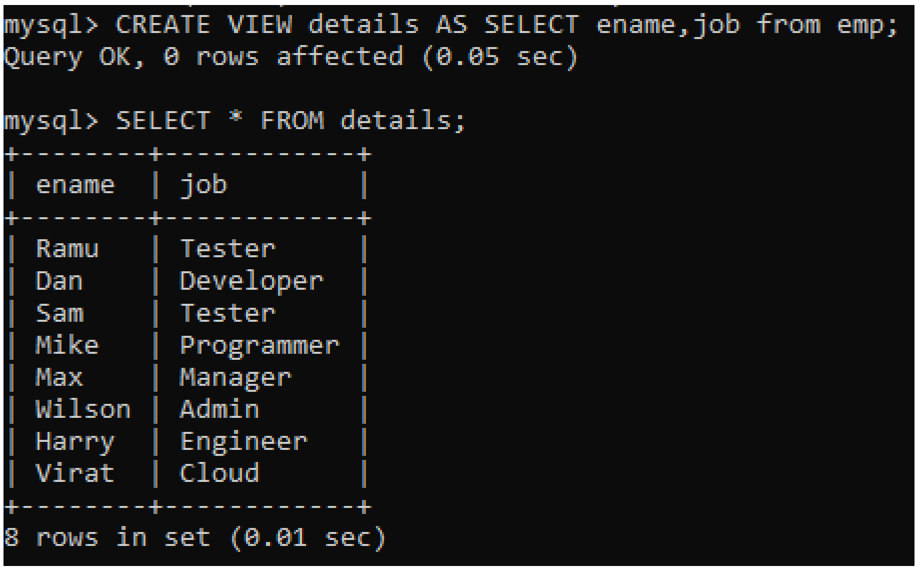
Query:



```
mysql> select * from emp where sal>(select max(sal) from emp where deptno=30);
+-------+-------+---------+------+--------+-------+
| EMPNO | ENAME | JOB     | MGR  | DEPTNO | SAL   |
+-------+-------+---------+------+--------+-------+
|    65 | Max   | Manager |    1 |      3 | 50000 |
+-------+-------+---------+------+--------+-------+
1 row in set (0.00 sec)
```

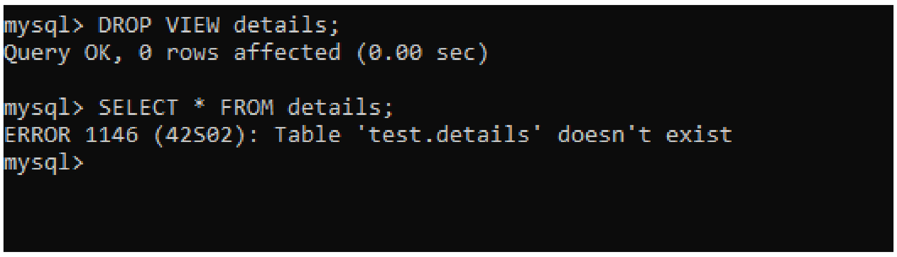
6. Write a query to create and drop View

To create a view:



```
mysql> CREATE VIEW details AS SELECT ename,job from emp;
Query OK, 0 rows affected (0.05 sec)

mysql> SELECT * FROM details;
+--------+------------+
| ename  | job        |
+--------+------------+
| Ramu   | Tester     |
| Dan    | Developer  |
| Sam    | Tester     |
| Mike   | Programmer |
| Max    | Manager    |
| Wilson | Admin      |
| Harry  | Engineer   |
| Virat  | Cloud      |
+--------+------------+
8 rows in set (0.01 sec)
```

To drop a view:



```
mysql> DROP VIEW details;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT * FROM details;
ERROR 1146 (42S02): Table 'test.details' doesn't exist
mysql>
```

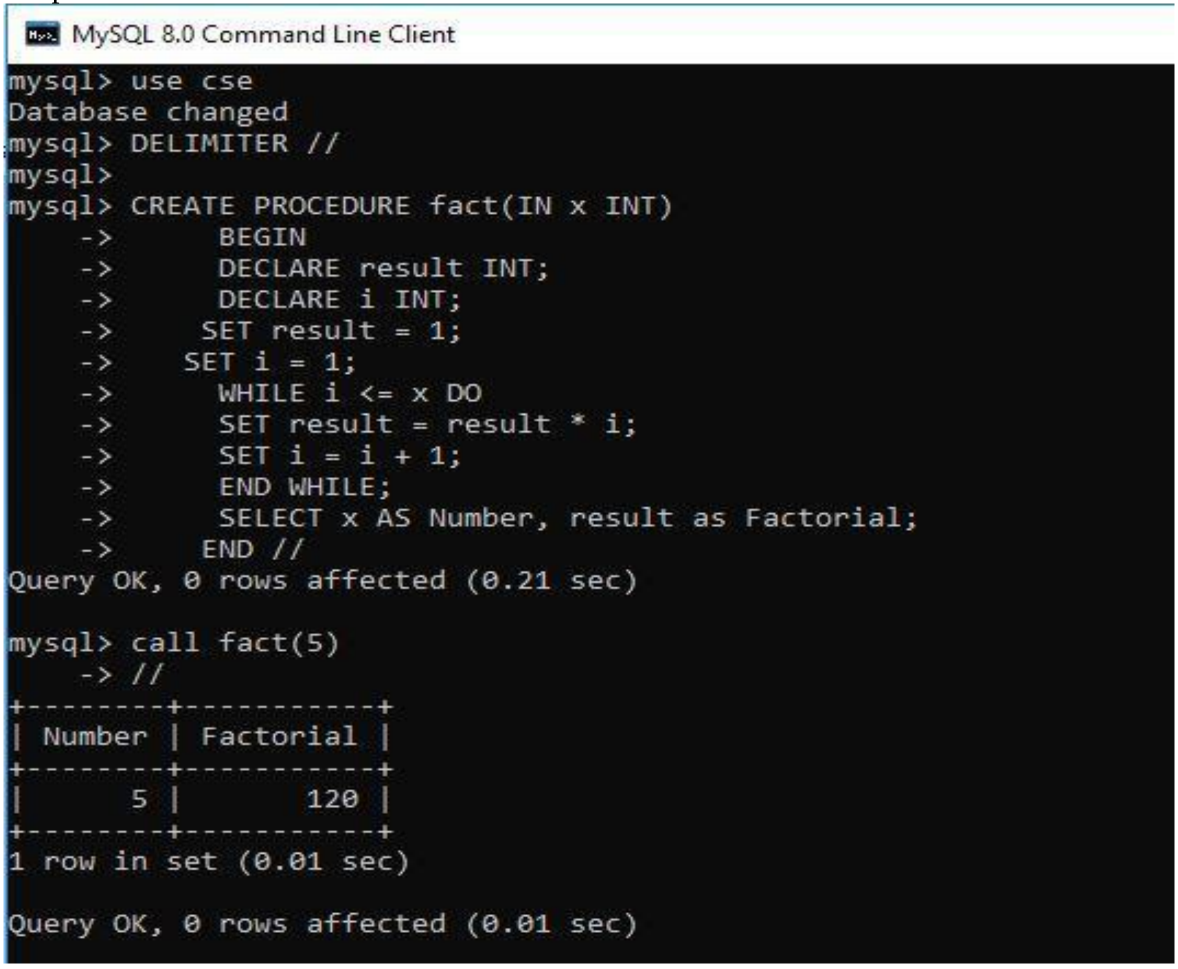
TASK 8:

8.Write a simple PL/SQL block to.
1. Print the factorial of a given number.

Queries:

```
DELIMITER //
CREATE PROCEDURE fact(IN x INT)
BEGIN
DECLARE result INT;
DECLARE i INT;
ET result = 1;
SET i = 1;
WHILE i <= x DO
SET result = result * i;
SET i = i + 1;
END WHILE;
SELECT x AS Number, result as Factorial;
END //
```

Output:



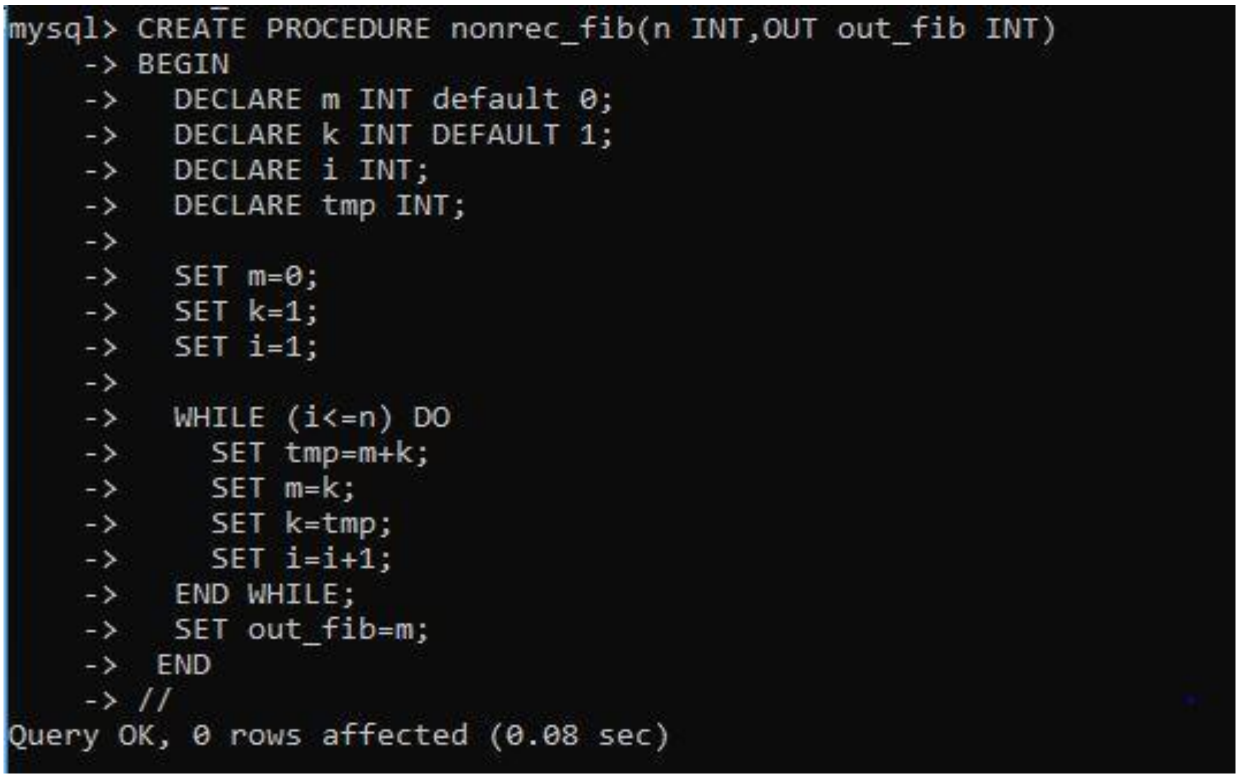
```
mysql> use cse
Database changed
mysql> DELIMITER //
mysql>
mysql> CREATE PROCEDURE fact(IN x INT)
    ->      BEGIN
    ->      DECLARE result INT;
    ->      DECLARE i INT;
    ->     SET result = 1;
    ->    SET i = 1;
    ->      WHILE i <= x DO
    ->      SET result = result * i;
    ->      SET i = i + 1;
    ->      END WHILE;
    ->      SELECT x AS Number, result as Factorial;
    ->     END //
Query OK, 0 rows affected (0.21 sec)

mysql> call fact(5)
    -> //
+--------+-----------+
| Number | Factorial |
+--------+-----------+
|      5 |       120 |
+--------+-----------+
1 row in set (0.01 sec)

Query OK, 0 rows affected (0.01 sec)
```

2. Print the Fibonacci series.

Queries:

```
DELIMITER //
CREATE PROCEDURE nonrec_fib(n INT,OUT out_fib INT)
BEGIN
DECLARE m INT default 0;
DECLARE k INT DEFAULT 1;
DECLARE i INT;
DECLARE tmp INT;
SET m=0;
SET k=1;
SET i=1;
WHILE (i<=n) DO
SET tmp=m+k;
SET m=k;
SET k=tmp;
SET i=i+1;
END WHILE;
SET out_fib=m;
END //
```

OUTPUT:



```
mysql> CREATE PROCEDURE nonrec_fib(n INT,OUT out_fib INT)
    -> BEGIN
    ->   DECLARE m INT default 0;
    ->   DECLARE k INT DEFAULT 1;
    ->   DECLARE i INT;
    ->   DECLARE tmp INT;
    ->
    ->   SET m=0;
    ->   SET k=1;
    ->   SET i=1;
    ->
    ->   WHILE (i<=n) DO
    ->     SET tmp=m+k;
    ->     SET m=k;
    ->     SET k=tmp;
    ->     SET i=i+1;
    ->   END WHILE;
    ->   SET out_fib=m;
    ->  END
    -> //
Query OK, 0 rows affected (0.08 sec)
```

TASK 9:

9. Write a cursor for the following: Declare a cursor that defines a result set. Open the cursor to establish the result set. Fetch the data into local variables as needed from the cursor, one row at a time. Close the cursor when done.

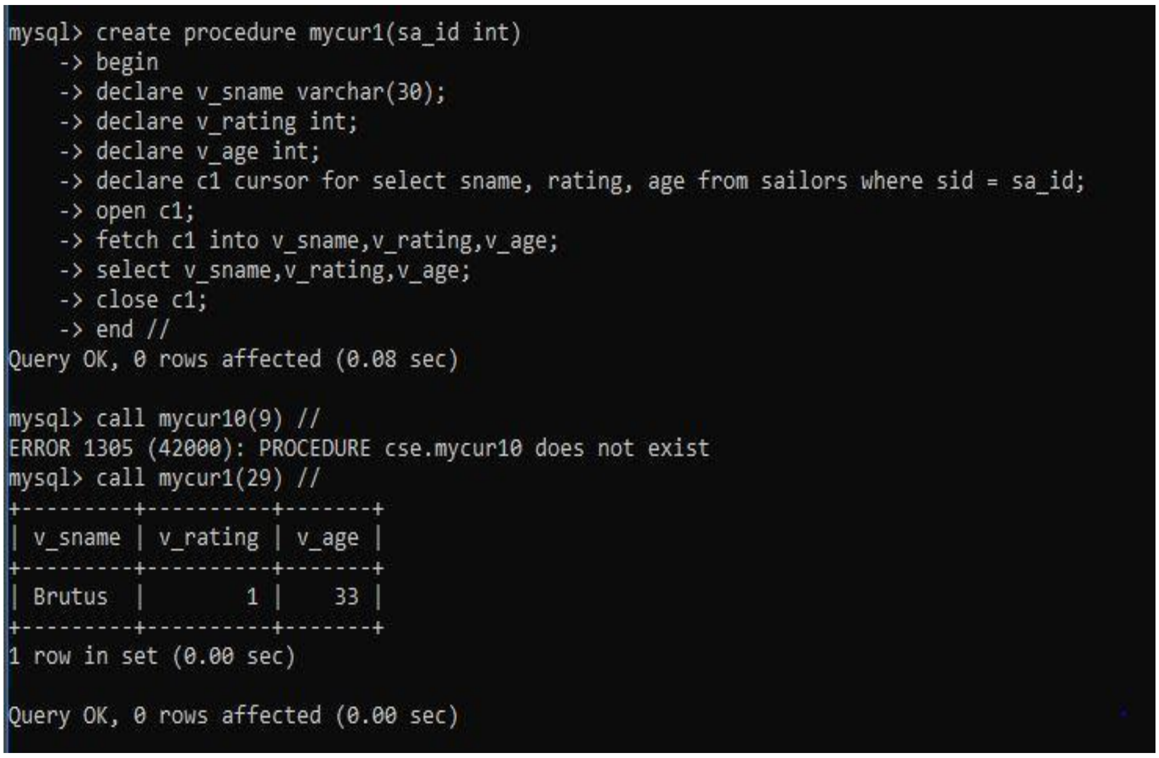
Example 1:

Queries:

```
CREATE TABLE Sailors( sid INT, sname VARCHAR(20), rating INT, age FLOAT,
PRIMARY KEY(sid) );

INSERT INTO Sailors VALUES(22,'Dustin',7,45);

INSERT INTO Sailors VALUES(29,'Brutus',1,33);

INSERT INTO Sailors VALUES(31,'Lubber',8,56);

 INSERT INTO Sailors VALUES(32,'Andy',8,26);

 INSERT INTO Sailors VALUES(58,'Rusty',10,35);

INSERT INTO Sailors VALUES(74,'Horatio',9,35);

INSERT INTO Sailors VALUES(64,'Horatio',7,35);

INSERT INTO Sailors VALUES(95,'Bob',3,64);

INSERT INTO Sailors VALUES(85,'Art',3,26);

INSERT INTO Sailors VALUES(71,'Zorba',10,16);


DELIMITER //

create procedure mycur1(sa_id int)

 begin

declare v_sname varchar(30);

declare v_rating int;

declare v_age int;

declare c1 cursor for select sname, rating, age from sailors where sid = sa_id;

open c1;

fetch c1 into v_sname,v_rating,v_age;

select v_sname,v_rating,v_age;

close c1;
```

end //

OUTPUT:



```
mysql> create procedure mycur1(sa_id int)
    -> begin
    -> declare v_sname varchar(30);
    -> declare v_rating int;
    -> declare v_age int;
    -> declare c1 cursor for select sname, rating, age from sailors where sid = sa_id;
    -> open c1;
    -> fetch c1 into v_sname,v_rating,v_age;
    -> select v_sname,v_rating,v_age;
    -> close c1;
    -> end //
Query OK, 0 rows affected (0.08 sec)

mysql> call mycur10(9) //
ERROR 1305 (42000): PROCEDURE cse.mycur10 does not exist
mysql> call mycur1(29) //
+---------+----------+-------+
| v_sname | v_rating | v_age |
+---------+----------+-------+
| Brutus  |        1 |    33 |
+---------+----------+-------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

Example 2

Queries:

DELIMITER //

create procedure mycur2(sa_rating int)

begin

declare v_sname varchar(30);

declare v_sid int;

declare v_age int;

declare c1 cursor for select sid,sname,age from sailors where rating=sa_rating;

open c1;

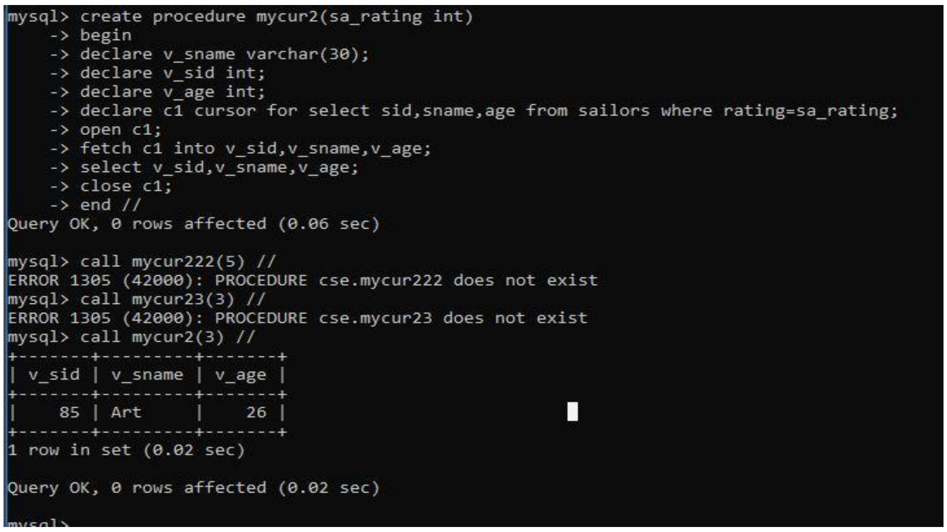
fetch c1 into v_sid,v_sname,v_age;

select v_sid,v_sname,v_age;

close c1;

end //

OUTPUT:



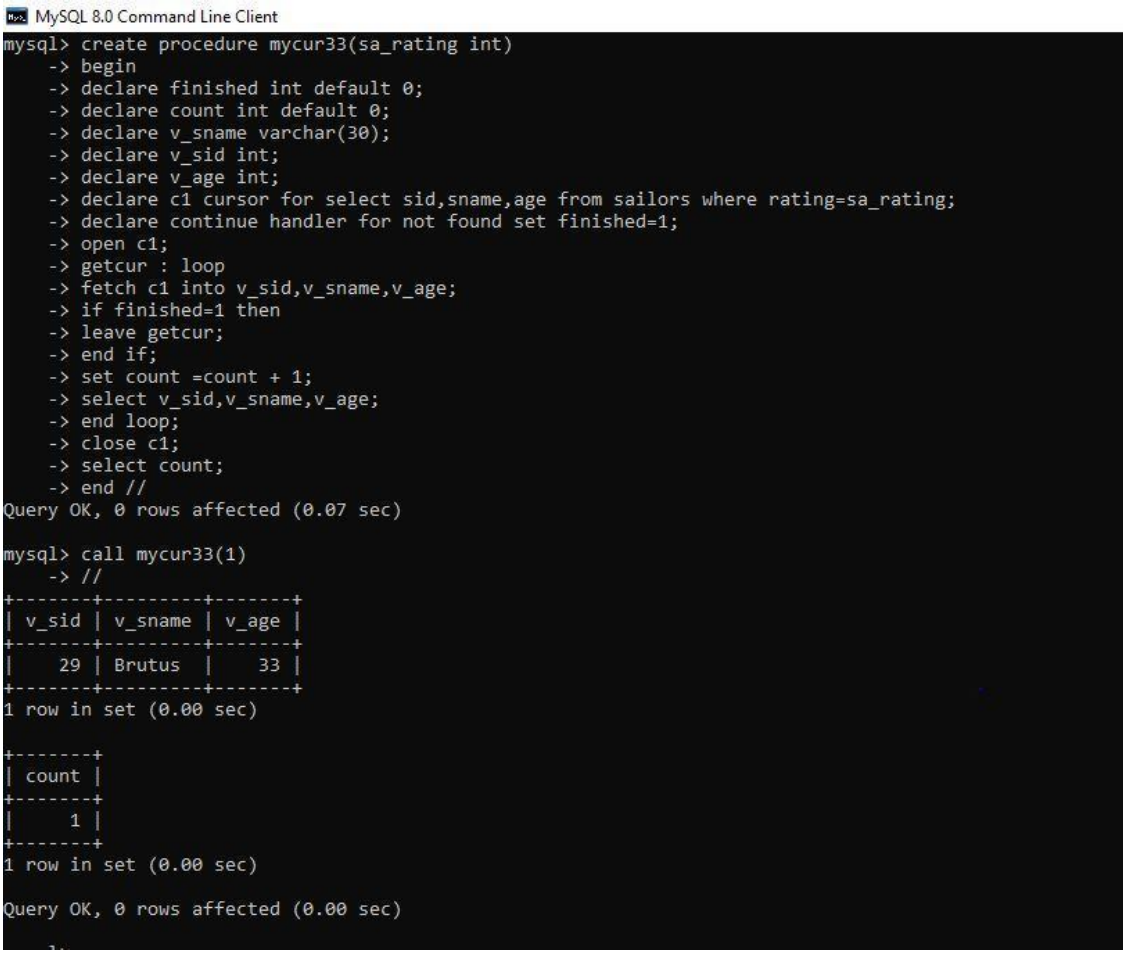
```
mysql> create procedure mycur2(sa_rating int)
    -> begin
    -> declare v_sname varchar(30);
    -> declare v_sid int;
    -> declare v_age int;
    -> declare c1 cursor for select sid,sname,age from sailors where rating=sa_rating;
    -> open c1;
    -> fetch c1 into v_sid,v_sname,v_age;
    -> select v_sid,v_sname,v_age;
    -> close c1;
    -> end //
Query OK, 0 rows affected (0.06 sec)

mysql> call mycur222(5) //
ERROR 1305 (42000): PROCEDURE cse.mycur222 does not exist
mysql> call mycur23(3) //
ERROR 1305 (42000): PROCEDURE cse.mycur23 does not exist
mysql> call mycur2(3) //
+-------+---------+-------+
| v_sid | v_sname | v_age |
+-------+---------+-------+
|    85 | Art     |    26 |
+-------+---------+-------+
1 row in set (0.02 sec)

Query OK, 0 rows affected (0.02 sec)

mysql>
```

Example 3

Queries:

```
DELIMITER //
create procedure mycur3(sa_rating int)
begin
declare finished int default 0;
declare count int default 0;
declare v_sname varchar(30);
declare v_sid int;
declare v_age int;
declare c1 cursor for select sid,sname,age from sailors where rating=sa_rating;
declare continue handler for not found set finished=1;
open c1; getcur : loop fetch c1 into v_sid,v_sname,v_age;
if finished=1 then leave getcur; end if;
set count =count + 1;
select v_sid,v_sname,v_age;
end loop;
close c1;
select count;
 end //
```

OUTPUT:



MySQL 8.0 Command Line Client

```
mysql> create procedure mycur33(sa_rating int)
    -> begin
    -> declare finished int default 0;
    -> declare count int default 0;
    -> declare v_sname varchar(30);
    -> declare v_sid int;
    -> declare v_age int;
    -> declare c1 cursor for select sid,sname,age from sailors where rating=sa_rating;
    -> declare continue handler for not found set finished=1;
    -> open c1;
    -> getcur : loop
    -> fetch c1 into v_sid,v_sname,v_age;
    -> if finished=1 then
    -> leave getcur;
    -> end if;
    -> set count =count + 1;
    -> select v_sid,v_sname,v_age;
    -> end loop;
    -> close c1;
    -> select count;
    -> end //
Query OK, 0 rows affected (0.07 sec)

mysql> call mycur33(1)
    -> //
+-------+---------+-------+
| v_sid | v_sname | v_age |
+-------+---------+-------+
|    29 | Brutus  |    33 |
+-------+---------+-------+
1 row in set (0.00 sec)

+-------+
| count |
+-------+
|     1 |
+-------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

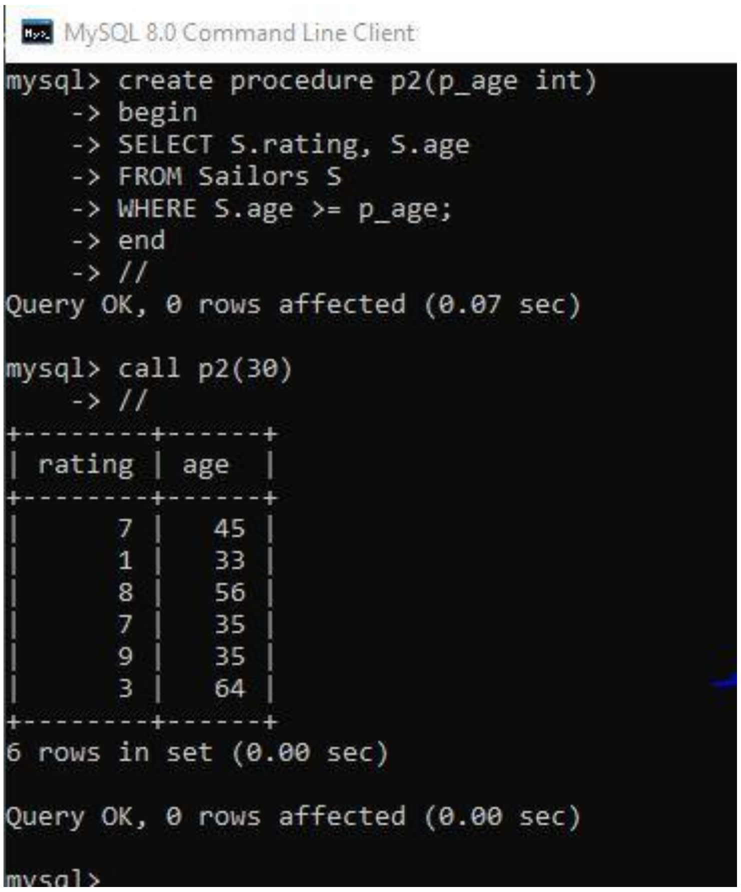
TASK 10:

10. Write a PL/SQL procedure to: Creation of stored procedure, Execution of procedure and modification of procedure.

Queries:

```
CREATE TABLE Sailors( sid INT, sname VARCHAR(20), rating INT, age FLOAT,
PRIMARY KEY(sid) );

INSERT INTO Sailors VALUES(22,'Dustin',7,45);

INSERT INTO Sailors VALUES(29,'Brutus',1,33);

INSERT INTO Sailors VALUES(31,'Lubber',8,56);

 INSERT INTO Sailors VALUES(32,'Andy',8,26);

 INSERT INTO Sailors VALUES(58,'Rusty',10,35);

INSERT INTO Sailors VALUES(74,'Horatio',9,35);

INSERT INTO Sailors VALUES(64,'Horatio',7,35);

INSERT INTO Sailors VALUES(95,'Bob',3,64);

INSERT INTO Sailors VALUES(85,'Art',3,26);

INSERT INTO Sailors VALUES(71,'Zorba',10,16);

DELIMITER //

create procedure p1(p_age int)

begin

SELECT S.rating, S.age

FROM Sailors S

WHERE S.age >= p_age;

End

call p1(30) //
```

OUTPUT:



```
mysql> create procedure p2(p_age int)
    -> begin
    -> SELECT S.rating, S.age
    -> FROM Sailors S
    -> WHERE S.age >= p_age;
    -> end
    -> //
Query OK, 0 rows affected (0.07 sec)

mysql> call p2(30)
    -> //
+--------+------+
| rating | age  |
+--------+------+
|      7 |   45 |
|      1 |   33 |
|      8 |   56 |
|      7 |   35 |
|      9 |   35 |
|      3 |   64 |
+--------+------+
6 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql>
```

TASK 11:

11. Write a Trigger for the following:

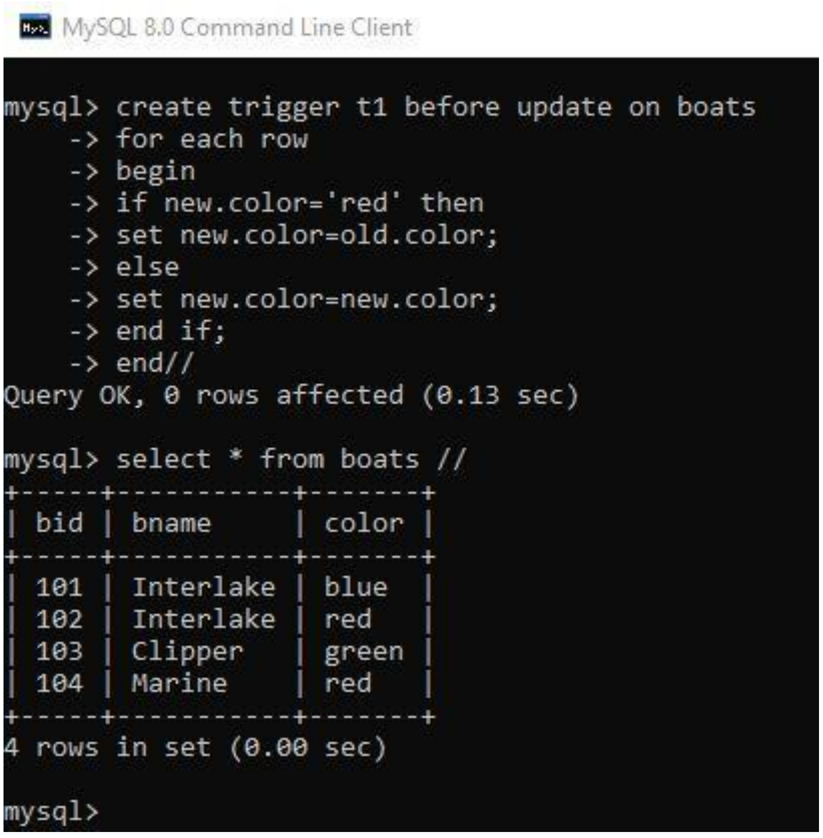
Creation of insert trigger, delete trigger, update trigger.

Update Trigger:

Queries:

```
CREATE TABLE Boats( bid INT, bname VARCHAR(10), color VARCHAR(10),
PRIMARY KEY(bid));

DESC Boats;

INSERT INTO Boats VALUES(101,'Interlake','blue');

INSERT INTO Boats VALUES(102,'Interlake','red');

INSERT INTO Boats VALUES(103,'Clipper','green');

INSERT INTO Boats VALUES(104,'Marine','red');

DELIMITER //

create trigger t1 before update on boats

 for each row

begin

 if new.color='red' then

set new.color=old.color;

else

set new.color=new.color;

end if;

end//
```

OUTPUT:



MySQL 8.0 Command Line Client

```
mysql> create trigger t1 before update on boats
    -> for each row
    -> begin
    -> if new.color='red' then
    -> set new.color=old.color;
    -> else
    -> set new.color=new.color;
    -> end if;
    -> end//
Query OK, 0 rows affected (0.13 sec)

mysql> select * from boats //
+-----+-----------+-------+
| bid | bname     | color |
+-----+-----------+-------+
| 101 | Interlake | blue  |
| 102 | Interlake | red   |
| 103 | Clipper   | green |
| 104 | Marine    | red   |
+-----+-----------+-------+
4 rows in set (0.00 sec)

mysql>
```

Insert Trigger:

Queries:

CREATE TABLE Sailors(sid INT, sname VARCHAR(20), rating INT, age FLOAT, PRIMARY KEY(sid));

INSERT INTO Sailors VALUES(22,'Dustin',7,45);

INSERT INTO Sailors VALUES(29,'Brutus',1,33);

INSERT INTO Sailors VALUES(31,'Lubber',8,56);

INSERT INTO Sailors VALUES(32,'Andy',8,26);

INSERT INTO Sailors VALUES(58,'Rusty',10,35);

INSERT INTO Sailors VALUES(74,'Horatio',9,35);

INSERT INTO Sailors VALUES(64,'Horatio',7,35);

INSERT INTO Sailors VALUES(95,'Bob',3,64);

INSERT INTO Sailors VALUES(85,'Art',3,26);

INSERT INTO Sailors VALUES(71,'Zorba',10,16);

DELIMITER //

create trigger t2

before insert on sailors

for each row

begin

if new.age>40 then

set new.rating='10';

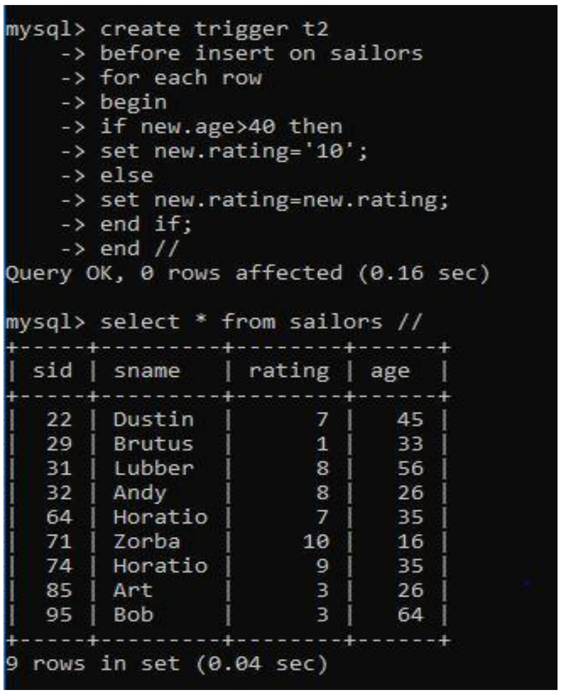
else

set new.rating=new.rating;

end if;

end //

OUTPUT:



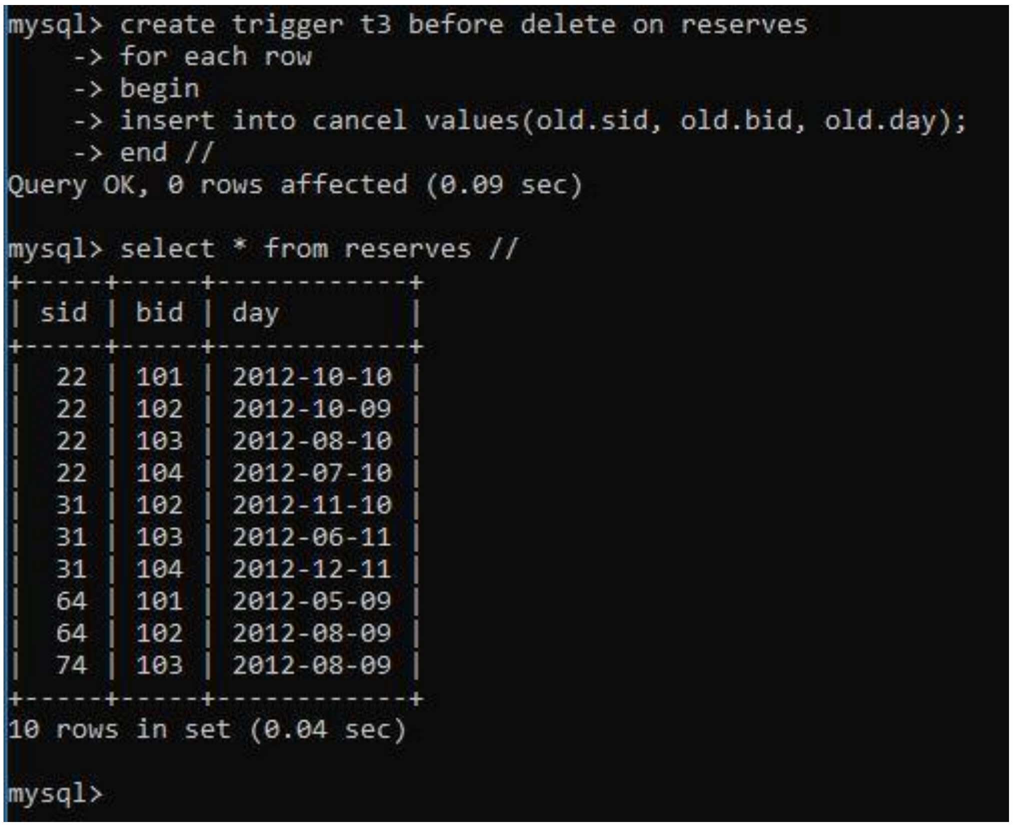
```
mysql> create trigger t2
    -> before insert on sailors
    -> for each row
    -> begin
    -> if new.age>40 then
    -> set new.rating='10';
    -> else
    -> set new.rating=new.rating;
    -> end if;
    -> end //
Query OK, 0 rows affected (0.16 sec)

mysql> select * from sailors //
+-----+---------+--------+------+
| sid | sname   | rating | age  |
+-----+---------+--------+------+
|  22 | Dustin  |      7 |   45 |
|  29 | Brutus  |      1 |   33 |
|  31 | Lubber  |      8 |   56 |
|  32 | Andy    |      8 |   26 |
|  64 | Horatio |      7 |   35 |
|  71 | Zorba   |     10 |   16 |
|  74 | Horatio |      9 |   35 |
|  85 | Art     |      3 |   26 |
|  95 | Bob     |      3 |   64 |
+-----+---------+--------+------+
9 rows in set (0.04 sec)
```

Delete Trigger:

Queries:

```
CREATE TABLE Reserves( sid INT, bid INT, day DATE NOT NULL, PRIMARY
KEY(sid,bid), FOREIGN KEY(sid) REFERENCES Sailors(sid) ON DELETE CASCADE,
FOREIGN KEY(bid) REFERENCES Boats(bid) ON DELETE CASCADE);

DESC Reserves;

INSERT INTO Reserves VALUES(22,101,'2012/10/10');

INSERT INTO Reserves VALUES(22,102,'2012/10/9');

INSERT INTO Reserves VALUES(22,103,'2012/08/10');

INSERT INTO Reserves VALUES(22,104,'2012/07/10');

INSERT INTO Reserves VALUES(31,102,'2012/11/10');

INSERT INTO Reserves VALUES(31,103,'2012/06/11');

INSERT INTO Reserves VALUES(31,104,'2012/12/11');

INSERT INTO Reserves VALUES(64,101,'2012/05/09');

INSERT INTO Reserves VALUES(64,102,'2012/08/09');

INSERT INTO Reserves VALUES(74,103,'2012/08/09');

DELIMITER //

create trigger t3 before delete on reserves

for each row

begin

insert into cancel values(old.sid, old.bid, old.day);

end //
```

OUTPUT:



```
mysql> create trigger t3 before delete on reserves
    -> for each row
    -> begin
    -> insert into cancel values(old.sid, old.bid, old.day);
    -> end //
Query OK, 0 rows affected (0.09 sec)

mysql> select * from reserves //
+-----+-----+------------+
| sid | bid | day        |
+-----+-----+------------+
|  22 | 101 | 2012-10-10 |
|  22 | 102 | 2012-10-09 |
|  22 | 103 | 2012-08-10 |
|  22 | 104 | 2012-07-10 |
|  31 | 102 | 2012-11-10 |
|  31 | 103 | 2012-06-11 |
|  31 | 104 | 2012-12-11 |
|  64 | 101 | 2012-05-09 |
|  64 | 102 | 2012-08-09 |
|  74 | 103 | 2012-08-09 |
+-----+-----+------------+
10 rows in set (0.04 sec)

mysql>
```

TASK 12:

12. Use TCL commands for your transactions.

1.commit

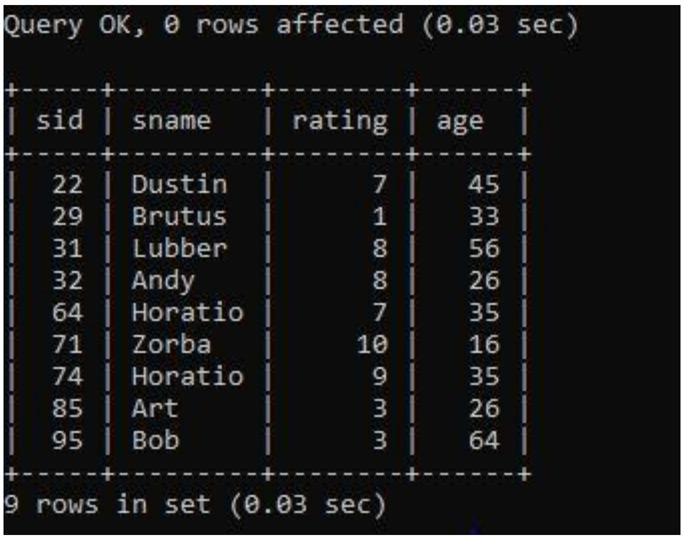
2.rollback

3.savepoint

Queries:

```
CREATE TABLE Sailors( sid INT, sname VARCHAR(20), rating INT, age FLOAT,
PRIMARY KEY(sid) );

INSERT INTO Sailors VALUES(22,'Dustin',7,45);

INSERT INTO Sailors VALUES(29,'Brutus',1,33);

INSERT INTO Sailors VALUES(31,'Lubber',8,56);

 INSERT INTO Sailors VALUES(32,'Andy',8,26);

 INSERT INTO Sailors VALUES(58,'Rusty',10,35);

INSERT INTO Sailors VALUES(74,'Horatio',9,35);

INSERT INTO Sailors VALUES(64,'Horatio',7,35);

INSERT INTO Sailors VALUES(95,'Bob',3,64);

INSERT INTO Sailors VALUES(85,'Art',3,26);

INSERT INTO Sailors VALUES(71,'Zorba',10,16);

SELECT *FROM sailors;
START TRANSACTION;
COMMIT;
SET autocommit = 0;
SAVEPOINT Insertion;
UPDATE sailors SET rating= 10 WHERE age = 35;
SAVEPOINT Updation;
ROLLBACK TO Insertion;
SELECT *FROM sailors;
```

OUTPUT:



```
Query OK, 0 rows affected (0.03 sec)

+-----+---------+--------+------+
| sid | sname   | rating | age  |
+-----+---------+--------+------+
|  22 | Dustin  |      7 |   45 |
|  29 | Brutus  |      1 |   33 |
|  31 | Lubber  |      8 |   56 |
|  32 | Andy    |      8 |   26 |
|  64 | Horatio |      7 |   35 |
|  71 | Zorba   |     10 |   16 |
|  74 | Horatio |      9 |   35 |
|  85 | Art     |      3 |   26 |
|  95 | Bob     |      3 |   64 |
+-----+---------+--------+------+
9 rows in set (0.03 sec)
```