AVN INSTITUTE OF ENGINEERING AND TECHNOLOGY

School of Engineering & Technology

Department of Computer Science & Engineering

Database Management Systems Lab

Master Lab Manual

Subject Code: CS407ES

For the Academic year 2018-19

II B. Tech II Semester



AVN INSTITUTE OF ENGINEERING AND TECHNOLOGY

Ibrahimpatnam, R R District – 501 506 (A. P.)

Department of Computer Science & Engineering

LAB MANUAL FOR THE ACADEMIC YEAR 2018-19

| SUB | : | DATABASE MANAGEMENT SYSTEMS |
|------------------|---|-----------------------------------|
| SUB CODE | : | CS407ES |
| SEMESTER | : | П |
| STREAM | : | CSE |
| Document No | : | AVNIET/CSE/ DBMS/R-16 |
| Date of Issue | : | 24/12/2018 |
| Date of Revision | : | |
| Prepared | : | Mr. P.GANESH, Assistant Professor |

Verified &

Authorized by

Dr. S.K.Abdul Nabi

HOD (CSE)

INDEX

| S.No | Contents |
|------|--|
| | |
| 1 | Lab Objective |
| 2 | Lab outcomes |
| 3 | Introduction About Lab |
| 4 | Guidelines to students |
| 5 | List of experiments as per the university curriculum |
| 6 | List of Additional Experiments |
| 8 | Text Books / Reference Books |

1. LAB OBJECTIVE

This DBMS Lab enables the students to practice the concepts learnt in the DBMS subject by developing a database for an example company named "Roadway Travels"

- The Student is expected to practice the designing, developing and querying a database in the context of "Roadway Travels".
- > The Student are expected to use "MySQL" Database for Practice.

2. LAB OUTCOME

Upon successful completion of this Lab the student will be able to:

- ER Model: The Objective of the lab is to analyze the problem carefully and come up with entities and attributes etc in the Roadway Travels and later buildup the conceptual design with the E-R Model.
- Relational Model: The Objective of the lab exercise is to represent all the entities in tabular fashion, representing relationships as tables based on Cardinality, different types of attributes(composite, Multi-valued and Derived)
- Normalization: The Objective of the lab exercise is to apply the Normalization techniques like INF,2NF,3NF,BCNF to above Roadway travels database
- Installation of MySQL and Practicing DDL & DML commands: The Objective of the lab Exercise is to install the MySQL software and Practice DDL & DML Commands and Practicing the Queries.
- > **Triggers, Cursors and Procedures**: The Objective of the lab exercise is to create triggers and cursors and writing Procedures on Roadway Travels database
- > me system is allotted for students when they do the lab.

| S.No | Name of the Experiment |
|------|---|
| 1 | E-R Model |
| 2 | Concept design with E-R Model |
| 3 | Relational Model |
| 4 | Normalization |
| 5 | Installation of MySQL and Practicing DDL Commands |
| 6 | Practicing DML Commands |
| 7 | Querying using ANY, ALL, IN, EXISTS, NOT EXISTS, UNION, INSERSECT |
| 8 | Querying using Aggregate Functions(COUNT,SUM,AVG,MAX and MIN) |
| 9 | Querying using GROUP BY, HAVING and Creation and Droping of views |
| 10 | Triggers |
| 11 | Procedures |
| 12 | Cursors |

3. LIST OF EXPERIMENTS AS PER THE UNIVERSITY CURRICULUM

4. List of Additional experiments for the semester

| S. No | Name of the experiment |
|-------|---|
| 1 | Queries on employee, department tables |
| 2 | DDL and DML COMMANDS on Sailors, boats & reserves |



E-R Model

Aim: Analyze the problem and come with the entities in it. Identify what Data has to be Persisted in the databases.

Recommended Hardware / Software Requirements:

- Hardware Requirements: Intel Based desktop PC with minimum of 166 MHZ or faster processor with at least 1GB RAM and 500 MB free disk space.
- MySQL 5.6.1

Prerequisites: Student must know the entities and attributes and representation of diagrams The Following are the entities:

Identify the primary keys for all the entities. Identify the other keys like candidate keys, partial keys, if any.

Example:

Entities: 1. BUS

- 2. Ticket
- 3. Passenger

Relationships: 1. Reservation

2. Cancellation

Primary Key Attributes: 1. Ticket ID (Ticket Entity)

2. Passport ID (Passenger Entity)

3. Bus_NO (Bus Entity)

E-R Model

Attributes of the following Entities

Bus

- BusNo
- Source
- Destination
- CoachType

SCHEMA

Bus: Bus(<u>BusNo: String</u>, Source: String, Destination: String, CoachType: String)



Ticket

- TicketNo
- DOJ
- Address
- ContactNo
- BusNo
- SeatNo
- Source
- Destination

SCHEMA

Ticket(<u>TicketNo:</u> string, DOJ: date, Address:string,ContactNo: string, BusNo:String ,SeatNo:Integer,Source: String, Destination: String)



Passenger

- PassportID
- TicketNo
- Name
- ContactNo
- Age
- Sex
- Address

SCHEMA

Passenger(<u>PassportID</u>: <u>String</u>, TicketNo:string,Name: String, ContactNo:string,Age: integer, Sex: character, Address: String)



Reservation

- PNRNo
- DOJ
- No_of_seats
- Address
- ContactNo
- BusNo
- SeatNo

SCHEMA

Reservation(PNRNo: String, DOJ: Date, NoofSeats: integer, Address: String, ContactNo: String, , BusNo: String,SeatNo:Integer)



Cancellation

- PNRNo
- DOJ
- SeatNo
- ContactNo
- Status

SCHEMA

Cancellation(PNRNo: String,DOJ: Date, SeatNo: integer,ContactNo: String,Status: String)The



Conclusion: The Student is able to Identify the Entities and Attributes of the Roadway Travels

Viva-Voce:

- 1. What is SQL?
- 2. what is entity and relationship?
- 3. What is DBMS?
- 4. What is a Database system?
- 5. Advantages of DBMS?
- 6. How many types of database languages are available?

2.CONCEPT DESIGN WITH E-R MODEL

Aim: To Relate the entities appropriately. Apply cardinalities for each relationship. Identify strong and weak entities. Indicate the type of relationships (total/partial). Incorporate generalization, aggregation and specialization etc wherever required.

Recommended Hardware / Software Requirements:

- Hardware Requirements: Intel Based desktop PC with minimum of 166 MHZ or faster processor with at least 1GB RAM and 500 MB free disk space.
- MySQL 5.6.1

Prerequisites: Student must know the entities and attributes and relationship among entities



Conclusion: The Student is able design the concept design of Road-way Travels

Example: E-R diagram for bus



Viva-Vice:

- 1. Describe the three levels of data abstraction?
- 2. Define the "integrity rules"
- 3. What are the usage of SQL?
- 4. What is the use of SELECT?

3. RELATIONAL MODEL

Aim: To Represent all the entities (Strong, Weak) in tabular fashion. Represent relationships in a tabular fashion.

Recommended Hardware / Software Requirements:

- Hardware Requirements: Intel Based desktop PC with minimum of 166 MHZ or faster processor with at least 1GB RAM and 500 MB free disk space.
- MySQL 5.6.1

Prerequisites: Student must know about the Relational Model

1. Bus: Bus(BusNo: String, Source: String, Destination: String, CoachType: String)

| ColumnName | Datatype | Constraints | Type of |
|-------------|-------------|-------------|--------------|
| | | | Attributes |
| BusNo | Varchar(10) | Primary | Single-value |
| | | key | |
| Source | Varchar(20) | | Single-value |
| Destination | Varchar(20) | | Simple |
| CoachType | Varchar(10) | | Simple |

Mysql>create table Bus(BusNo varchar(10),source varchar(20),Destination varchar(20),coachType varchar(10),primary key(BusNo));

Mysql>desc Bus;

mysql> use cse; Database changed

Database cnanged mysql> create table Bus(BusNo varchar(10),source varchar(20),Destination varchar(20),coachType varchar(10),primary key(BusNo)); Query OK, O rows affected (0.06 sec)

nysql> desc Bus;

| Field | Туре | Null | Key | Default | Extra |
|-------------|------------------------------|--------------------|------------|---------|-------------|
| BusNo | varchar(10) | + NO VEC | + PRI | | ++ |
| Destination | varchar(20) varchar(20) | YES | | NULL | |
| coachType | varchar(10) | YES | | NULL | |

rows in set (0.00 sec)

mysql>

Ticket:

Ticket(<u>TicketNo:</u> string, DOJ: date, Address:string,ContactNo: string, BusNo:String ,SeatNo:Integer,Source: String, Destination: String)

| ColumnName | Datatype | Constraints | Type of Attributes |
|-------------|-------------|-------------|--------------------|
| TicketNo | Varchar(20) | Primary Key | Single-valued |
| DOJ | Date | | Single-valued |
| Address | Varchar(20) | | Composite |
| ContactNo | Integer | | Multi-valued |
| BusNo | Varchar(10) | Foreign Key | Single-valued |
| SeatNo | Integer | | Simple |
| Source | Varchar(10) | | Simple |
| Destination | Varchar(10) | | Simple |

Mysql>createtableTicket(TicketNovarchar(20),DOJdate,Addressvarchar(20),ContactNovarchar(15)BusNovarchar(10),SeatNoint,Sourcevarchar(10),primary key(TicketNo,BusNo),foreign key(BusNo) referencesBus(BusNo));

Mysql>desc Ticket;

mysql> create table Ticket (TicketNo varchar(20),DOJ date,Address varchar(20),ContactNo varchar(15),BusNo varc har(10),seatNo int,Source varchar(10),Destination varchar(10),primary key(TicketNo,BusNo),foreign key(BusNo)re ferences Bus(BusNo)); Query OK, O rows affected (0.05 sec)

mysql> desc Ticket;

| Field | Туре | Null | Key | Default | Extra |
|--------------|-------------------|-----------|------------|---------|-------|
| TicketNo | varchar(20) | + NO | + PRI | | |
| DOJ | date | YES | | NULL | 1 |
| Address | varchar(20) | YES | | NULL | |
| ContactNo | varchar(15) | YES | | NULL | |
| BusNo | varchar(10) | NO | PRI | | |
| seatNo | int(11) | YES | | NULL | |
| Source | varchar(10) | YES | | NULL | |
| Destination | varchar(10) | YES | | NULL | |

s rows in set (0.00 sec)

nysql>

Passenger:

Passenger(<u>PassportID: String</u>, TicketNo:string,Name: String, ContactNo:string,Age: integer, Sex: character, Address: String);

| ColumnName | Datatype | Constraints | Type of Attributes |
|------------|-------------|-------------|-----------------------|
| PassportID | Varchar(15) | Primary Key | Single-valued |
| TicketNo | Varchar(20) | Foreign Key | Single-valued |
| Name | Varchar(20) | | Composite |
| ContactNo | Varchar(20) | | Multi-valued |
| Age | Integer | | Single-valued |
| Sex | character | | Simple |
| Address | Varchar(20) | | Composite |

Mysql> Create table passenger(passportID varchar(15), TicketNo varchar(15), Name varchar(15), ContactNo varchar(20), Age integer, sex char(2), address varchar(20), primary key(passportID, TicketNo), foreign key(TicketNo) references Ticket(TicketNo));

Mysql> desc passenger;

mysql> use cse; Database changed mysql> create table passenger(passportid varchar(10),ticketno varchar(15),name varchar(15),contactno varchar(1 5),age integer,sex char(2),address varchar(20),primary key(passportid,ticketno),foreign key(ticketno) referenc es ticket(ticketno)); Query OK, O rows affected (0.08 sec) mysql> desc passenger; Field Null | Key | Default | Extra Type passportid varchar(10) NO PRI varchar(15) ticketno NO PRI varchar(15) YES NULL name varchar(15) contactno YES NULL int(11) YES NULL age char(2)sex YES NULL varchar(20) | YES address NULL rows in set (0.03 sec)

Reservation:

Reservation(PNRNo: String, DOJ: Date, NoofSeats: integer, Address: String, ContactNo: String, , BusNo: String,SeatNo:Integer)

| ColumnName | Datatype | Constraints | Type of Attributes |
|-------------|-------------|-------------|--------------------|
| PNRNo | Varchar(20) | Primary | Single-valued |
| | | Key | _ |
| DOJ | date | | Single-valued |
| No_of_Seats | Integer | | Simple |
| Address | Varchar(20) | | Composite |
| ContactNo | Varchar(10) | | Multi-valued |
| BusNo | Varchar(10) | Foreign | Single-valued |
| | | Key | |
| SeatNo | Integer | | Simple |

Mysql> Create table Resevation(PNRNo varchar(20),DOJ date,NoofSeates integer,Address varchar(20),ContactNo varchar(20),BusNo varchar(20),SeatNo integer, primary key(PNRNo,BusNo),foreign key(BusNo) references Bus(BusNo));

Mysql> desc reservation;

mysql> create table reservation(PNRNo varchar(20),DOJ date,NofSeats integer,Address varchar(20),ContactNo varc har(20),BusNo varchar(20),SeatNo integer,primary key(PNRNo,BusNo),foreign key(BusNo) references Bus(BusNo)); Query OK, O rows affected (0.05 sec)

mysql> desc Reservation;

| Field | Туре | Null | Key | Default | Extra |
|-----------|-------------------|-----------|------------|-----------------|-------|
| PNRNo | varchar(20) | + NO | + PRI | + | + |
| DOJ | date | YES | | NULL | |
| NofSeats | int(11) | YES | Ì | NULL | |
| Address | varchar(20) | YES | | NULL | |
| ContactNo | varchar(20) | YES | | NULL | |
| BusNo | varchar(20) | NO | PRI | n standestan ve | |
| SeatNo | int(11) | YES | | NULL | |

Cancellation:

Cancellation(PNRNo: String,DOJ: Date, SeatNo: integer,ContactNo: String,Status: String)

| ColumnName | Datatype | Constraints | Type of Attributes |
|------------|-------------|-------------|--------------------|
| PNRNo | Varchar(10) | Primary Key | Single-valued |
| DOJ | date | | Single-valued |
| SeatNo | Integer | | Simple |
| ContactNo | Varchar(15) | | Multi-valued |
| Status | Varchar(10) | | Simple |

Mysql> create table cancellation(PNRNo varchar(10),DOJ date,SeatNo integer, ContactNo varchar(15),Status varchar(10), primary key(PNRNo), foreign key(PNRNo) references reservation(PNRNo));

Mysql> desc cancellation;

mysql> create table cancellation(PNRNo varchar(10),DOJ date,SeatNo integer,ContactNo varchar(15),Status varcha r(10),primary key(PNRNo),foreign key(PNRNo) references Reservation(PNRNo)); Query OK, 0 rows affected (0.05 sec) mysql> desc cancellation; Null | Key | Default | Extra Field Type PNRNo varchar(10) NO PRI YES DOJ date NULL int(11) YES SeatNo NULL varchar(15) YES ContactNo NULL varchar(10) | YES Status NULL rows in set (0.00 sec)

Conclusion: The Student is able draw the tabular representation of the relations of Roadway travels.

Viva-Voce:

- 1. What is the difference between SUM and COUNT ?
- 2. What is VIEW ? and What will you get when you use VIEW
- 3. What is difference between TRUNCATE and DELETE?

4.NORMALIZATION

Aim: Apply the database Normalization techniques for designing relational database tables to minimize duplication of information like 1NF, 2NF, 3NF, BCNF.

Recommended Hardware / Software Requirements:

- Hardware Requirements: Intel Based desktop PC with minimum of 166 MHZ or faster processor with at least 1GB RAM and 500 MB free disk space.
- MySQL 5.6.1

Prerequisites: Student must know about the Relational database(SQL)

Normalization is a process of converting a relation to be standard form by decomposition a larger relation into smaller efficient relation that depicts a good database design.

1NF: A Relation scheme is said to be in 1NF if the attribute values in the relation are atomic.i.e., Mutli –valued attributes are not permitted.

2NF: A Relation scheme is said to be in 2NF,iff and every Non-key attribute is fully functionally dependent on primary Key.

3NF: A Relation scheme is said to be in 3NF,iff and does not have transitivity dependencies. A Relation is said to be 3NF if every determinant is a key for each & every functional dependency.

BCNF: A Relation scheme is said to be BCNF if the following statements are true for eacg FD P->Q in set F of FDs that holds for each FD. P->Q in set F of FD's that holds over R. Here P is the subset of attributes of R & Q is a single attribute of R.

i)The given FD is a trival

ii) P is a super key.

Normalized tables are:-

Mysql> create table Bus2(BusNo varchar(20) primary key,Source varchar(20),Destination varchar(20));

Mysql>Create table passenger4(PPN varchar(15) Primary key,Name varchar(20),Age integer,Sex char,Address varchar(20));

Mysql> Create table PassengerTicket(PPN varchar(15) Primary key, TicketNo integer);

Mysql> Create table Reservation2(PNRNO integer Primary key, JourneyDate DateTime,NoofSeats int,Address varchar(20),ContactNo Integer);

Mysql> create table Cancellation2(PNRNO Integer primary key,JourneyDate DateTime,NoofSeats Integer,Address varchar(20),ContactNo Integer,foreign key(PNRNO) references Reservation2(PNRNO));

Mysql> Create table Ticket2(TicketNo Integer Primary key,JourneyDate DateTime, Age Int(4),Sex char(2),Source varchar(20),Destination varchar(20),DeptTime varchar(2));

Conclusion: The Student is able to Normalize the tables applying 3NF

Viva-Voce:

- 1. Define Normalization?
- 2. What is 1 NF (Normal Form)?
- 3. What is Fully Functional dependency?
- 4. What is 2NF?
- 5. What is 3NF?
- 6. What is BCNF (Boyce-Codd Normal Form)?

5. INSTALLATION OF MYSQL AND PRACTICING DDL COMMANDS

Aim: Installation of MySQL. Creating database tables, altering the database, dropping tables and truncate commands

Recommended Hardware / Software Requirements:

- Hardware Requirements: Intel Based desktop PC with minimum of 166 MHZ or faster processor with at least 1GB RAM and 500 MB free disk space.
- MySQL 5.6.1

Installation Steps for MySQL:

1. Steps for installing MySQL

Step1

Make sure you already downloaded the **MySQL essential 5.0.45 win32.msi file**. Double click on the .msi file.

Step2

This is MySQL Server 5.0 setup wizard. The setup wizard will install MySQL Server 5.0 release 5.0.45 on your computer. To continue, click **next.**

| 🛃 MySQL Server 5.0 - Setup | Wizard |
|----------------------------|--|
| | Welcome to the Setup Wizard for MySQL Server 5.0 The Setup Wizard will install MySQL Server 5.0 release 5.0.45 on your computer. To continue, click Next. |
| MySQL. | WARNING: This program is protected by copyright law. |
| | < Back Next > Cancel |

Choose the setup type that best suits your needs. For common program features select *Typical* and it's recommended for general use. To continue, click **next**.



Step 4

This wizard is ready to begin installation. Destination folder will be in C:\Program

Files\MySQL\MySQL Server 5.0\. To continue, click next

| 🔂 MySQL Server 5.0 - Setup Wizard | × |
|---|---|
| Ready to Install the Program The wizard is ready to begin installation. | |
| If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard. | |
| Setup Type: Complete Destination Folder: C:\Program Files\MySQL\MySQL Server 5.0\ | |
| < <u>Back</u> Install Cancel | |

Dept of CSE , AVNIET

The program features you selected are being installed. Please wait while the setup wizard installs MySQL 5.0. This may take several minutes.

| 🔂 MySQL So | erver 5.0 - Setup Wizard 📃 🗌 🗙 |
|------------------------|--|
| Installing The prog | MySQL Server 5.0 ram features you selected are being installed. |
| 1 7 | Please wait while the Setup Wizard installs MySQL Server 5.0. This may take several minutes. |
| | Status: |
| | Updating component registration |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | < Back Next > Cancel |

Step6

To continue, click **next**.

| MySQL Enterprise | | × |
|---|---|------------|
| Enterprise I. The MySQL Enterprise Serversion of the worlds most popul | A MySQL Enterprise subscription is the most comprehensive offering of MySQL database software, services and support to ensure your business achieves the highest levels of reliability, security and uptime. An Enterprise Subscription includes: erver - The most reliable, secure, and up-to-date ar open source database. | |
| 2. The MySQL Monitoring a database assistant. | nd Advisory Service - An automated virtual | |
| 3. MySQL Production Supp you need it, along with service pa | ort - Technical and consultative support when acks, hot-fixes and more. | |
| For more information click | More] or visit www.mysql.com/enterprise | b . |
| More | < Back Next > Cancel | |

To continue, click next



Step8

Wizard Completed. Setup has finished installing MySQL 5.0. **Check** the configure the MySQL server now to continue. Click **Finish** to exit the wizard



The configuration wizard will allow you to configure the MySQL Server 5.0 server instance



Step10

Select a **standard configuration** and this will use a general purpose configuration for the server that can be tuned manually. To continue, click **next**



Check on the **install as windows service** and **include bin directory in windows path**. To continue, click **next**.

| MySQL Server Instance Configuration Wizard | × |
|---|------------|
| MySQL Server Instance Configuration Configure the MySQL Server 5.0 server instance. | \bigcirc |
| Please set the Windows options. | |
| ▼ Install As Windows Service | |
| This is the recommended way to run the MySQL server on Windows. | |
| Service Name: MySQL5 | |
| ✓ Launch the MySQL Server automatically | |
| Include Bin Directory in Windows PATH | |
| Check this option to include the directory containing the server / client executables in the Windows PATH variable so they can be called from the command line. | |
| < Back Next > | Cancel |

Step12

Please set the security options by entering the root password and confirm retype the password.

To continue, click next.

| MySQL Server In | <mark>stanc</mark> e Configuration | Wizard | × | | | | | | |
|--------------------------------------|--|--|---------------------------------|--|--|--|--|--|--|
| MySQL Server Configure the | Instance Configurat MySQL Server 5.0 serve | ion er instance, | | | | | | | |
| Please set the | security options. | | | | | | | | |
| Modify Se | curity Settings | | | | | | | | |
| | New root password: | ***** | Enter the root password. | | | | | | |
| root | Confirm: | ***** | Retype the password. | | | | | | |
| | | 🔲 Enable root a | access from remote machines | | | | | | |
| | | | | | | | | | |
| 🔽 Create An | Anonymous Account | | | | | | | | |
| 2 | This option will create note that this can lead | an anonymous accour I to an insecure syster | nt on this server. Please n. | | | | | | |
| | < Back Next > Cancel | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |

Ready to execute? Clicks **execute** to continue.

| MySQL Server Instance Configuration Wizard |
|--|
| MySQL Server Instance Configuration |
| Configure the MySQL Server 5.0 server instance. |
| Ready to execute |
| |
| Prepare configuration |
| Write configuration file |
| Start service |
| Apply security settings |
| Please press [Execute] to start the configuration. |
| |
| |
| |
| |
| < Back Cancel Cancel |

Step14 Processing configuration in progress.

| MySQL Server Instance Configuration | Wizard X | | | | | |
|---|--|--|--|--|--|--|
| MySQL Server Instance Configuration Configure the MySQL Server 5.0 server instance. | | | | | | |
| Processing configuration | | | | | | |
| Prepare configuration Write configuration file Start service Apply security settings | (C:\Program Files\MySQL\MySQL Server 5.0\my.ini) | | | | | |
| | | | | | | |
| I | < Back Execute Cancel | | | | | |
| | | | | | | |
| | | | | | | |

Dept of CSE , AVNIET

Configuration file created. Windows service MySQL5 installed. Press **finish** to close the wizard.

| MySQL Server Instance Configuration Wizard | × |
|---|------|
| MySQL Server Instance Configuration Configure the MySQL Server 5.0 server instance. | |
| Processing configuration | |
| Prepare configuration | |
| ♂ Write configuration file (C:\Program Files\MySQL\MySQL Server 5.0\my.inî) | |
| | |
| Apply security settings | |
| Configuration file created. Windows service MySQL5 installed. Service started successfully. Security settings applied. | |
| Press [Finish] to close the Wizard. | |
| < Back Car | ncel |

Creating Tables and altering the Tables

Mysql>Create table passenger2(passportId Integer Primary Key,Name varchar(10) Not Null,Age Integer Not Null,Sex char,Address varchar(20) Not Null);

Mysql> desc passenger2;

```
mysql> create table passenger3(passportId integer primary key,name varchar(10) not null,Age Integer not null,
Sex char, Address varchar(20) not null);
Query OK, 0 rows affected (0.03 sec)
mysql> desc passenger3;
 Field
                             Nu11
                                          Default | Extra
               Type
                                    Key |
  passportId
               int(11)
                             NO
                                    PRI
               varchar(10)
                             NO
  name
               int(11)
  Age
                             NO
               char(1)
                             YES
 Sex
                                           NULL
               varchar(20)
  Address
                             NO
  rows in set (0.02 sec)
```

USING ALTER COMMAND

Adding Extra column to Existing Table

Mysql>Alter table passenger3 add column TicketNo varchar(10);

```
mysql> Alter table passenger3 add column TicketNo
Query OK, O rows affected (0.14 sec)
Records: O Duplicates: O Warnings: O
                                                                         varchar(10);
mysql> desc passenger3;
  Field
                     Type
                                         Nu11
                                                  Key
                                                           Default | Extra
  passportId
                     int(11)
                                         NO
                                                   PRI
                     varchar(10)
                                         NO
  name
                     int(11)
                                         NO
  Age
                     char(1)
varchar(20)
varchar(10)
  Sex
                                         YES
                                                            NULL
  Address
                                         NO
                                         YES
   TicketNo
                                                            NULL
  rows in set (0.00 sec)
```

Mysql>Alter Table passenger3 add Foreign key(TicketNo) references Ticket(TicketNo);

| C:\Program Files (x86 | i)\MySQL\MySQL Server 5. |)\bin\mysql.e | xe | | - | and it is not | And Incident | |
|--|---|--|-----------------------------------|--------------------------------|--------------|---------------|--------------|-----------|
| mysql> alter Query OK, 0 ro Records: 0 D mysql> desc pa | table passenge ows affected (u uplicates: 0 u assenger3; | r3 add 0.08 se Warnings | foreign z) s: O | n key(Tick | etNo) re | Ferences | Ticket(T | icketNo); |
| + Field | + Туре | + Null | + Key | + Default | + Extra | + | | |
| + passportId name Age Sex Address TicketNo | + int(11) varchar(10) int(11) char(1) varchar(20) varchar(10) | NO NO NO YES NO YES | PRI MUL | NULL NULL | | | | |
| + 6 rows in set | + (0.02 sec) | + | + | + | + | F | | |

Mysql>Alter Table passenger3 Modify column Name varchar(20);

| C:\Program Files (x86 |)\MySQL\MySQL Server 5. | 0\bin\mysql.e | xe | | of the second |
|---|--|---|-----------------------|--------------------------|--------------------|
| mysql> Alter ⁻ Query OK, 0 ro Records: 0 Du | Table passenge ows affected (uplicates: 0 | r3 Modi 0.11 se Warning | fy colu c) s: 0 | umn Name va | archar(20); |
| mysql> desc pa | assenger3; | an e richer in ear | | , itselasteria estadosta | para no se a se po |
| Field | Туре | Null | Key | Default | Extra |
| passportId Name Age Sex Address TicketNo | int(11) varchar(20) int(11) char(1) varchar(20) varchar(10) | NO YES NO YES NO YES | PRI | NULL NULL | |
| + 6 rows in set | (0.00 sec) | + | + | + | ++ |

Mysql>Alter table passenger drop foreign key fk1;

| mysql> Alter 1 Query OK, 0 ro Records: 0 Du | table passenge ows affected (0 uplicates: 0 N | r2 add 0.07 se Warning: | column c) s: 0 | n TicketNo | varcha | r(10); | |
|--|--|---|--|----------------------------|---------------|-----------|--------------|
| mysql> alter 1 s Ticket(Ticke Query OK, 0 ro Records: 0 Du mysql> Alter 1 Query OK, 0 ro Records: 0 Du mysgl> desc pa | table passenger etNo); ows affected (0 uplicates: 0 N table passenger ows affected (0 uplicates: 0 N assenger?: | r2 add 0.07 se Warning: r2 drop 0.09 se Warning: | constra c) s: 0 foreig c) s: 0 | aint fk1 fo gn key fk1 | oreign k ; | ey(Ticket | No) referenc |
| + Field | + Type | + Null | + Key | + Default | + Extra | + 1 | |
| + passportId name Age Sex Address TicketNo | int(11) varchar(10) int(11) char(1) varchar(20) varchar(10) | + NO NO YES NO YES | + PRI | + NULL | + | + | |
| + 6 rows in set | (0.00 sec) | + | + | + | + | ÷ | |

Mysql> Alter table passenger2 Drop column TicketNo;

```
mysql> Alter table passenger2 drop column ticketNo;
Query OK, 0 rows affected (0.08 sec)
Records: 0 Duplicates: 0 Warnings: 0
mysql> desc passenger2;
  Field
                                   Nu11
                                                  Default
                  Туре
                                           Key
                                                              Extra
                  int(11)
  passportId
                                   NO
                                           PRI
                  varchar(10)
                                   NO
  name
                  int(11)
                                   NO
  Age
                  char(1)
                                                   NULL
  Sex
                                   YES
                  varchar(20)
  Address
                                   NO
  rows in set (0.01 sec)
```

Viva Voce:

- 1. What is DDL (Data Definition Language)?
- 2. What is VDL (View Definition Language)?
- 3. What is SDL (Storage Definition Language)?
- 4. What is DML (Data Manipulation Language)?
- 5. What is DML Compiler?
- 6. What is PL/SQL?

6. PRACTICING DML COMMANDS

Aim : Create a DML Commands are used to manage data within the scheme objects.

SELECT- retrieve data from the database INSERT- insert data into a table UPDATE- Updates existing data within a table DELETE-delete all records from a table.

Recommended Hardware / Software Requirements:

- Hardware Requirements: Intel Based desktop PC with minimum of 166 MHZ or faster processor with at least 1GB RAM and 500 MB free disk space.
- MySQL 5.6.1

Prerequisites: Student must know about the RDBMS SQL

DML Commands:

INSERT COMMAND ON BUS2 & PASSENGER2 RELATIONS

mysql> select * from Bus2;

Empty set (0.00 sec)

mysql> insert into Bus2 values(1234, 'Hyderabad', 'Tirupathi');

Query OK, 1 row affected (0.03 sec)

mysql> insert into Bus2 values(2345, 'Hyderabad', 'Banglore');

Query OK, 1 row affected (0.01 sec)

mysql> insert into Bus2 values(23,'Hyderabad','Kolkata');

Query OK, 1 row affected (0.03 sec)

mysql> insert into Bus2 values(45, 'Tirupathi', 'Banglore');

Query OK, 1 row affected (0.03 sec)

mysql> insert into Bus2 values(34, 'Hyderabad', 'Chennai');

Query OK, 1 row affected (0.03 sec)

mysql> select * from Bus2;

| mysql> se Empty set | elect * from t (0.00 sec) | Bus2; |
|------------------------|------------------------------|---|
| mysql> in | nsert into Bu | us2 values(1234,'Hyderabad','Tirupathi'); |
| Query OK | , 1 row affe | ted (0.03 sec) |
| mysql> in | nsert into Bu | us2 values(2345,'Hyderabad','Banglore'); |
| Query OK | , 1 row affe | ted (0.01 sec) |
| mysql> in | nsert into Bu | us2 values(23,'Hyderabad','Kolkata'); |
| Query OK | , 1 row affe | ted (0.03 sec) |
| mysql> in | nsert into Bu | us2 values(45,'Tirupathi','Banglore'); |
| Query OK | , 1 row affe | ted (0.03 sec) |
| mysql> in | nsert into Bu | us2 values(34,'Hyderabad','Chennai'); |
| Query OK | , 1 row affe | ted (0.03 sec) |
| mysql> se | elect * from | Bus2; |
| BusNo | Source | Destination |
| 1234 | Hyderabad | Tirupathi |
| 23 | Hyderabad | Kolkata |
| 2345 | Hyderabad | Banglore |
| 34 | Hyderabad | Chennai |
| 45 | Tirupathi | Banglore |
| + 5 rows in | n set (0.01 s | sec) |

mysql> select * from Passenger2;

Empty set (0.00 sec)

mysql> insert into Passenger2 values(145, 'Ramesh', 45, 'M', 'abc123');

Query OK, 1 row affected (0.05 sec)

mysql> insert into Passenger2 values(278,'Geetha',36,'F','abc124');

Query OK, 1 row affected (0.02 sec)

mysql> insert into Passenger2 values(4590, 'Ram', 30, 'M', 'abc12');

Query OK, 1 row affected (0.03 sec)

mysql> insert into Passenger2 values(6789, 'Ravi', 50, 'M', 'abc14');

Query OK, 1 row affected (0.03 sec)

mysql> insert into Passenger2 values(5622, 'Seetha', 32, 'F', 'abc55');

Query OK, 1 row affected (0.03 sec)

mysql> select * from Passenger2;

| mysql> select Empty set (0.(| * from Pa 00 sec) | assenge | er2; | | | | | |
|--|---|----------------------------|--------------------------------|--|---|--|--|--|
| mysql> insert into Passenger2 values(145,'Ramesh',45,'M','abc123'); Query OK, 1 row affected (0.05 sec) | | | | | | | | |
| mysql> insert into Passenger2 values(278,'Geetha',36,'F','abc124'); Query OK, 1 row affected (0.02 sec) | | | | | | | | |
| mysql> insert into Passenger2 values(4590,'Ram',30,'M','abc12'); Query OK, 1 row affected (0.03 sec) | | | | | | | | |
| mysql> insert into Passenger2 values(6789,'Ravi',50,'M','abc14'); Query OK, 1 row affected (0.03 sec) | | | | | | | | |
| mysql> insert into Passenger2 values(5622,'Seetha',32,'F','abc55'); Query OK, 1 row affected (0.03 sec) | | | | | | | | |
| mysql> select * from Passenger2; | | | | | | | | |
| passportId | name | Age | Sex | Address | + | | | |
| 145 278 4590 5622 6789 | Ramesh Geetha Ram Seetha Ravi | 45 36 30 32 50 | М Г Г М Г М | abc123 abc124 abc12 abc12 abc55 abc14 | + | | | |
| ++ 5 rows in set (0.00 sec) | | | | | | | | |

Dept of CSE , AVNIET

UPDATE COMMAND ON BUS2 RELATION

UPDATE Selected Rows & Multiple Rows

mysql> Update Bus2 SET Source='Secundrabad' where BusNo=1234;

Query OK, 1 row affected (0.05 sec)

Rows matched: 1 Changed: 1 Warnings: 0

| C:\Program | Files (x86)\MySQL\MySQ | L Server 5.0\bin\mysql.exe | | | |
|--|---|---|-------------------------------------|--|--|
| mysql> se | elect * from E | Sus2; | | | |
| BusNo | Source | Destination | | | |
| + 1234 23 2345 34 45 | Hyderabad Hyderabad Hyderabad Hyderabad Tirupathi | d Tirupathi d Kolkata d Banglore d Chennai i Banglore | | | |
| mysql> U Query OK Rows mate mysql> se | pdate Bus2 SET , 1 row affect ched: 1 Chang elect * from E | Source='Secu ed (0.05 sec) ed: 1 Warnin Gus2; | ndrabad' where BusNo=1234; gs: O | | |
| BusNo | Source | Destination | | | |
| 1234 23 2345 34 45 | Secundrabad Hyderabad Hyderabad Hyderabad Tirupathi | Tirupathi Kolkata Banglore Chennai Banglore | | | |
| 5 rows in | n set (0.00 se | ec) | | | |
mysql> Update Bus2 SET Source='Secundrabad' where Source='Hyderabad';

Query OK, 1 row affected (0.05 sec)

Rows matched: 1 Changed: 1 Warnings: 0

| C:\Program | Files (x86)\MySQL\MySQL | Server 5.0\bin\mysql.exe | |
|---|--|--|---|
| mysql> se | elect * from Bu | us2; | |
| BusNo | Source | Destination | |
| 1234 23 2345 34 45 | Secundrabad Hyderabad Hyderabad Hyderabad Tirupathi | Tirupathi Kolkata Banglore Chennai Banglore | |
| 5 rows ir mysql> Up Query OK, Rows matc mysql> se | n set (0.00 sed odate Bus2 SET 3 rows affect hed: 3 Change elect * from Bu | c) Source='Secund ted (0.02 sec) ed: 3 Warnings us2; | drabad' where Source='Hyderabad'; ;: 0 |
| BusNo | Source | Destination | |
| 1234 23 2345 34 45 + | Secundrabad Secundrabad Secundrabad Secundrabad Tirupathi | Tirupathi Kolkata Banglore Chennai Banglore | |
| 5 rows ir | n set (0.00 sec | z) | |

DELETE COMMAND ON BUS2 RELATION

DELETES Selected Rows and Multiple Rows

mysql> Delete from Bus2 where BusNo=1234;

Query OK, 1 row affected (0.05 sec)

mysql> select * from Bus2;

| BusNo Source | | Destination | | |
|--------------------------------|---|---|--|--|
| 1234 23 2345 34 45 | Secundrabad Secundrabad Secundrabad Secundrabad Tirupathi | Tirupathi Kolkata Banglore Chennai Banglore | | |
| rows 11 | elete from Bus |) where BusNo | | |
| Jery OK | . 1 row affecte | ed (0.05 sec) | | |
| very OK | , 1 row affecte | ≥d (0.05 sec) µs2; | | |
| ysql> OK ysql> so BusNo | , 1 row affecte elect * from Bu | us2; Destination | | |

mysql> Delete from Bus2 where Source='Secundrabad';

Query OK, 1 row affected (0.05 sec)

mysql> select * from Bus2;

| mysql> se | elect * from B | 152; | |
|---|--|---|------|
| BusNo | Source | Destination | |
| 23 2345 34 45 | Secundrabad Secundrabad Secundrabad Tirupathi | Kolkata Banglore Chennai Banglore | |
| 4 rows 11 mysql> De Query OK mysql> se | n set (0.00 se elete from Bus , 3 rows affec elect * from B | :) 2 where Source='Secundrab ted (0.03 sec) 1s2; | ad'; |
| + BusNo | ++- Source | Destination | |
| 45 | Tirupathi | Banglore | |
| 1 row in | set (0.00 sec |) | |

Conclusion: The Student is able perform DML Commands like Insert, Update, Delete and Select

Viva-Voce

- 1. What are DML commands?
- 2. Write the syntax for insert command?
- 3. What is the syntax for update command?
- 4. Write the syntax for delete command?
- 5. Write the syntax for select command?
- 6. Difference between DCL and TCL

7. QUERYING using ANY, ALL, IN, EXISTS, NOT EXISTS, UNION, INSERSECT

Aim: Practice the following Queries:

- 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.

Recommended Hardware / Software Requirements:

- Hardware Requirements: Intel Based desktop PC with minimum of 166 MHZ or faster processor with at least 64MB RAM and 100 MB free disk space.
- MySQL 5.6.1

Prerequisites: Student must know about the RDBMS

| mysql> DESC | RESERVATION2; | | | | | | | |
|---|---|--------------------------------|---------------------|--------------------------------------|--|--------------------------------|---------|----------|
| Field | Туре | Null | Key | Default | Extra | | | |
| PNRNO Journeyda NoofSeats Address CONTACTNO | int(11) te datetime int(11) varchar(20) varchar(15) | NO YES YES YES YES | PRI | NULL NULL NULL NULL | | | | |
| 5 rows in s | set (0.00 sec) | | ++ | | + | | | |
| mysql> inse 235242); Query OK, 1 | ert into reservatio row affected (0.0 | on2 valu)3 sec) | ues (102 | 01,'2012-0 | 02-20 10:2 | 20:25 | ,05,'H | YD',9654 |
| mysql≻ inse 232451); Query OK, 1 | ert into reservatio . row affected (0.0 | on2 valu)2 sec) | ues (102 | 02,'2012-0 | 02-22 10:2 | 2:25' | ,05,'н | YD',9654 |
| mysql≻ inse 54587960); Query OK, 1 | ert into reservatio row affected (0.0 | on2 valu)1 sec) | ues (102 | 03,'2012-0 | 3-22 10:3 | 0:25' | ,05,'DI | ELHI',90 |
| mysql> inse 9845761254) Query ОК, 1 | ert into reservatio ; row affected (0.0 | on2 valu)2 sec) | ues (102 | 04,'2013-0 | 3-22 11:3 | 0:25' | ,05,'ci | HENNAI' |
| mysql> SELE ++ | CT * FROM RESERVAT | TION2; | | + | + | + | | |
| PNRNO J | lourneydate | Noo | fSeats | Address | CONTACT | NO | | |
| 10201 2 10202 2 10203 2 10204 2 | 2012-02-20 10:20:21 2012-02-22 10:22:21 2012-03-22 10:30:21 2013-03-22 11:30:21 | | 5 5 5 5 | HYD HYD DELHI CHENNAI | 9654235 9654232 9654587 9845761 | 242 451 960 254 | | |
| 4 rows in s | set (0.01 sec) | -+ | | + | + | + | | |

mysql> insert into passenger2 values(82302,'Smith',23,'M','Hyderabad'); Query OK, 1 row affected (0.02 sec)

mysql> insert into passenger2 values(82303,'Neha',23,'F','Hyderabad'); Query OK, 1 row affected (0.01 sec)

mysql> insert into passenger2 values(82304,'Neha',35,'F','Hyderabad'); Query OK, 1 row affected (0.03 sec)

mysql> insert into passenger2 values(82306, 'Ramu', 40, 'M', 'Hyderabad'); Query OK, 1 row affected (0.02 sec)

mysql> insert into passenger2 values(82308,'Aakash',40,'M','Hyderabad'); Query OK, 1 row affected (0.02 sec)

mysql> insert into passenger2 values(82402,'Aravind',42,'M','Hyderabad'); Query OK, 1 row affected (0.02 sec)

mysql> insert into passenger2 values(82403,'Avinash',42,'M','Hyderabad'); Query OK, 1 row affected (0.02 sec)

mysql> insert into passenger2 values(82502, 'Ramesh', 23, 'M', 'Hyderabad'); Query OK, 1 row affected (0.02 sec)

mysql> insert into passenger2 values(82602, 'Rajesh', 23, 'M', 'Hyderabad'); Query OK, 1 row affected (0.02 sec)

RESERVATION2

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) 1. Display unique PNR_NO of all reservation

Mysql>Select DISTINCT PNR_NO from Reservation;

| PNR_No | |
|--------|--|
| 10201 | |
| 10202 | |
| 10203 | |
| 10204 | |

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

2. Display all the names of male passengers.

mysql> Select p.name from passenger2 p where p.passportid IN (select p2.passportid from passenger2 p2

```
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 |
| |
| |
| Aakach |
| Aravind |
| L Avinash L |
| Ramesh |
| Rajesh |
| ++ |
| 10 rows in set (0.00 sec) |
| |

| mysql> SELECT | * FROM PAS | SENGER | R2; | a a. |
|--|-------------|----------|-----|-----------|
| passportId | name | Age | Sex | Address |
| 145 | Ramesh | 45 | | 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 | I M | Hyderabad |
| -> FROM P/ -> WHERE I NAME Ramesh Ram Ravi Smith Ramu Aakash Aravind Avinash | ASSENGER2 F | >2); | | |
| Rajesh ++ | | | | |
| 10 rows in set | t (0.00 sec | z) | | |

| mysql> desc pa | assengerticket; | | | | |
|---------------------------------|-----------------------------------|--------------------|--------|--------------|-------|
| Field | Туре | Null | Key | Default | Extra |
| passportid TicketNo | varchar(15) int(11) | NO YES | PRI | NULL | |
| 2 rows in set | (0.00 sec) | + | | | + |
| mysql> insert Query OK, 1 ro | into passenger ow affected (0. | rticket 03 sec) | values | ;(145,100); | |
| mysql> insert Query OK, 1 ro | into passenger ow affected (0. | rticket 03 sec) | values | ;(278,200); | |
| mysql> insert Query OK, 1 ro | into passenger ow affected (0. | rticket 03 sec) | values | 6789,300) | ; |
| mysql> insert Query OK, 1 ro | into passenger ow affected (0. | rticket 03 sec) | values | (82302,400 |)); |
| mysql> insert Query OK, 1 ro | into passenger ow affected (0. | rticket 03 sec) | values | s(82403,500 |)); |
| mysql> insert Query OK, 1 ro | into passenger ow affected (0. | rticket 02 sec) | values | s (82502,600 |)); |

3. Display the ticket numbers and names of all the passengers.

mysql> select t.ticketno,p.name from passengerticket t,passenger2 p where t.passportid = p.passportid;

| mysql> SELEC | T 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 |
| f rows in se | et (0.00 sec) |

4. Find the ticket numbers of the passengers whose name start with 'r' and ends with 'h'.

MySQL> SELECT Name FROM Passenger WHERE name LIKE 'R%H'

| Name | |
|--------|--|
| Rajesh | |
| Ramesh | |
| Ramesh | |

| ssportId | 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 | IM | Hyderabad |
| ows in se l> SELECT | t (0.00 sed | E) PASSEI | NGER2 W | HERE NAME LIK |

5. Find the names of Passengers whose age is between 30 and 45.

MySQL> SELECT Name FROM PASSENGER WHERE AGE BETWEEN 30 AND 45

| mysql> SELECT | * FROM PAS | SSENGER | R2; | | |
|--|--|--|---|---|---------|
| passportId | name | Age | Sex | Address | |
| + 1 145 278 4590 5622 6789 82302 82303 82304 82306 82308 82308 82402 82402 82403 82402 82403 82502 82602 | Ramesh Geetha Ram Seetha Ravi Smith Neha Neha Ramu Aakash Aravind Avinash Ramesh Ramesh Rajesh | 45 36 30 50 23 23 23 40 40 40 42 42 42 23 23 | M F M F M F M M M M M M M M M | abc123 abc124 abc12 abc55 abc55 Hyderabad Hyderabad | |
| 14 rows in set mysql> SELECT | t (0.00 sed Name FROM | E) PASSEN | NGER2 W | HERE AGE BETWEEN 30 | AND 45; |
| ++ Name | | | | | |
| <pre>++ Ramesh Geetha Ram Seetha Neha Ramu Ramu Aakash Aravind Avinash ++ 9 rows in set</pre> | (0.00 sec) | | | | |

6. Display all the passengers names beginning with 'A'.

MySQL> SELECT * FROM PASSENGER WHERE NAME LIKE 'A%';

| Name | |
|---------|--|
| Akash | |
| Arivind | |
| Avinash | |
| | |

| mysql> SELECT | * FROM PAS | SENGER | R2; | 1 |
|--|--|--|--------|---|
| passportId | name | Age | Sex | Address |
| 145 278 4590 5622 6789 82302 82303 82304 82304 82306 82308 82308 82402 82402 82403 82502 82602 | Ramesh Geetha Ram Seetha Ravi Smith Neha Neha Ramu Aakash Aravind Avinash Ramesh Rajesh | 45 36 30 23 23 23 40 40 42 42 23 23 | | abc123 abc124 abc12 abc55 abc14 Hyderabad Hyderabad Hyderabad Hyderabad Hyderabad Hyderabad Hyderabad Hyderabad Hyderabad Hyderabad |
| 82002 + | NAME FROM | PASSEN | M + | Hyderabad ++ |

7. Display the sorted list of Passengers names

MySQL> SELECT NAME FROM PASSENGER ORDER BY NAME;

| 145 Ramesh 45 M abc123 278 Geetha 36 F abc124 4590 Ram 30 M abc12 5622 Seetha 32 F abc14 82302 Smith 23 M Hyderabad 82302 Smith 23 F Hyderabad 82303 Neha 23 F Hyderabad 82304 Neha 23 F Hyderabad 82306 Ramu 40 M Hyderabad 82308 Aakash 40 M Hyderabad 82403 Avinash 42 M Hyderabad 82403 Avinash 42 M Hyderabad 82602 Ramesh 23 M Hyderabad 82602 Rajesh 23 M Hyderabad 82602 Rajesh 23 M Hyderabad 82602 Rajesh 23 M Hyderabad Avinash Setton Seetha | passportId | name | Age | Sex | Address |
|--|---|-----------|--------|---------|-------------|
| 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 82402 Rajesh 23 M Hyderabad 82502 Ramesh 23 M Hyderabad 82602 Rajesh 23 M Hyderabad 82603 NAME + rows in set (0.00 sec) sql> SELECT NAME FROM PASSENGER2 ORDER BY NAM + NAME + NAME Aravind Avinash Geetha Ramesh Ramesh Ramesh Ramesh Ramesh Ramu Ravi Seetha Smith + | 145 | Ramesh | 45 | | abc123 |
| 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 82304 Neha 35 F Hyderabad 82304 Neha 35 F Hyderabad 82304 Neha 40 M Hyderabad 82308 Aakash 40 M Hyderabad 82308 Aakash 40 M Hyderabad 82403 Avinash 42 M Hyderabad 82402 Aravind 42 M Hyderabad 82602 Ramesh 23 M Hyderabad 82602 Rajesh 23 M Hyderabad Aravind SELECT NAME F NAME + Akash I Aravind I | 278 | Geetha | 36 | F | abc124 |
| 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 82304 Neha 35 F Hyderabad 82306 Ramu 40 M Hyderabad 82306 Ramu 40 M Hyderabad 82308 Aakash 40 M Hyderabad 82402 Aravind 42 M Hyderabad 82403 Avinash 42 M Hyderabad 82602 Ramesh 23 M Hyderabad 82602 Rajesh 23 M Hyderabad rows in set (0.00 sec) sql> SELECT NAME FROM PASSENGER2 ORDER BY NAM Aravind Axash Aravind Aravind Aravind Ramesh Ramesh Ramesh Ramesh Ramesh Ramesh Ramesh Ramesh Rames | 4590 | Ram | 30 | M | abc12 |
| 6789 Ravi 50 M abcl4 82302 Smith 23 M Hyderabad 82303 Neha 23 F Hyderabad 82306 Ramu 40 M 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 26 27 27 27 27 27 27 27 | 5622 | Seetha | 32 | F | abc55 |
| 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 82308 Aakash 40 M Hyderabad 82402 Aravind 42 M Hyderabad 82402 Aravind 42 M Hyderabad 82402 Ramesh 23 M Hyderabad 82402 Ramesh 23 M Hyderabad 82402 Ramesh 23 M Hyderabad 82502 Ramesh 23 M Hyderabad 82602 Rajesh 23 M Hyderabad rows in set (0.00 sec) Sql> Sql> NAM + Aakash Aravind Hyderabad Aravind Avinash Sql> Sql> Sql> Sql> Seetha Seetha Seetha | 6789 | Ravi | 50 | M | abc14 |
| 82303 Neha 23 F Hyderabad 82304 Neha 35 F Hyderabad 82306 Ramu 40 M Hyderabad 82308 Aakash 40 M Hyderabad 82308 Aakash 40 M Hyderabad 82402 Aravind 42 M Hyderabad 82402 Aravind 42 M Hyderabad 82402 Ramesh 23 M Hyderabad 82502 Ramesh 23 M Hyderabad 82602 Rajesh 23 M Hyderabad 82602 Rajesh 23 M Hyderabad rows in set (0.00 sec) seq1> SELECT NAME FROM PASSENGER2 ORDER BY NAM Seq1> SELECT NAME FROM PASSENGER2 ORDER BY NAM Avinash Avinash Avinash Geetha Neha Ramesh Ramesh Ramesh Ramesh Ramesh Ramesh Ramu Ravi Seetha Seetha Seetha Smith Imagerin seq Ima | 82302 | Smith | 23 | M | Hyderabad |
| 82304 Neha 35 F Hyderabad 82306 Ramu 40 M Hyderabad 82308 Aakash 40 M Hyderabad 82402 Aaravind 42 M Hyderabad 82403 Avinash 42 M Hyderabad 82403 Avinash 42 M Hyderabad 82502 Ramesh 23 M Hyderabad 82602 Rajesh 23 M Hyderabad 82602 Rajesh 23 M Hyderabad 82602 Rajesh 23 M Hyderabad rows in set (0.00 sec) seq1> Seq1> SELECT NAME FROM PASSENGER2 ORDER BY NAM + Aakash Aaravind Aaravind Aavinash Geetha Seetha Seq1> Ramesh Ramesh Ramesh Ramesh Ramesh Seetha Seetha Seetha Smith I I Seq2 Seq2 | 82303 | Neha | 23 | F | Hyderabad |
| 82306 Ramu 40 M Hyderabad 82308 Aakash 40 M Hyderabad 82402 Aravind 42 M Hyderabad 82403 Avinash 42 M Hyderabad 82402 Ramesh 23 M Hyderabad 82502 Ramesh 23 M Hyderabad 82602 Rajesh 23 M Hyderabad 82602 Rajesh 23 M Hyderabad secore Rajesh 23 M Hyderabad rows in set (0.00 sec) Seql> SELECT NAME FROM PASSENGER2 ORDER BY NAM seql> SELECT NAME FROM PASSENGER2 ORDER BY NAM + Aakash Aakash Aravind Avinash Geetha Seetha Ramesh Ramesh Ramesh Ramesh Ramesh Ramu Ramesh Ramesh Seetha Smith Hernov + Power in set (0.02 sec) Hernov | 82304 | Neha | 35 | F | 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 82602 Rajesh 23 M Hyderabad 82602 Rajesh 23 M Hyderabad rows in set (0.00 sec) Sql> SELECT NAME FROM PASSENGER2 ORDER BY NAM + NAME I + Aakash I Aravind I I Avinash I I Geetha I I Neha I I Ramesh I I Ramesh I I Ramesh I I Ramu I I Seetha I I Smith I I I I I I I I | 82306 | Ramu | 40 | M | Hyderabad |
| 82402 Aravind 42 M Hyderabad 82403 Avinash 42 M Hyderabad 82502 Ramesh 23 M Hyderabad 82602 Rajesh 23 M Hyderabad 82602 Rajesh 23 M Hyderabad 82602 Rajesh 23 M Hyderabad rows in set (0.00 sec) sec seq1> SELECT NAME FROM PASSENGER2 ORDER BY NAM + NAME I + Aakash I Aravind Avinash I Geetha I I Neha I I Rajesh I I Ramesh I I Ramesh I I Ramesh I I Ramesh I I Seetha I I Seetha I I Smith I I I I I I I I I <t< td=""><td>82308</td><td>Aakash</td><td>40</td><td>M</td><td>Hyderabad</td></t<> | 82308 | Aakash | 40 | M | Hyderabad |
| 82403 Avinash 42 M Hyderabad 82502 Ramesh 23 M Hyderabad 82602 Rajesh 23 M Hyderabad rows in set (0.00 sec) ************************************ | 82402 | Aravind | 42 | M | Hyderabad |
| 82502 Ramesh 23 M Hyderabad 82602 Rajesh 23 M Hyderabad rows in set (0.00 sec) sql> SELECT NAME FROM PASSENGER2 ORDER BY NAM + NAME + NAME Aravind Avinash Geetha Neha Neha Ramesh Ramesh Ramesh Ramesh Ramu Ramu Seetha Smith + Pows in set (0.02 sec) | 82403 | Avinash | 42 | M | Hyderabad |
| 82602 Rajesh 23 M Hyderabad rows in set (0.00 sec) sql> SELECT NAME FROM PASSENGER2 ORDER BY NAM + NAME + Aakash Aravind Avinash Geetha Neha Ram Ramesh Ramesh Ramesh Ramu Ramu Ramu Seetha Seetha Seetha Smith + | 82502 | Ramesh | 23 | M | Hyderabad |
| rows in set (0.00 sec) sql> SELECT NAME FROM PASSENGER2 ORDER BY NAM + NAME + Aakash Aravind Avinash Geetha Neha Rahesh Ramesh Ramesh Ramesh Ramu Ramu Ramu Ramu Seetha Smith + pows in set (0.02 sec) | 82602 | Rajesh | 23 | M | Hyderabad |
| Aakash Aravind Avinash Geetha Neha Neha Rajesh Ram Ramesh Ramesh Ramu Ramu Ravi Seetha Smith | ysql> SELECT | NAME FROM | PASSEN | NGER2 O | RDER BY NAM |
| Aravind Avinash Geetha Neha Neha Rajesh Ramesh Ramesh Ramesh Ramu Ravi Seetha Smith + | ysql> SELECT + NAME + | NAME FROM | PASSEN | NGER2 O | RDER BY NAM |
| Avinash Geetha Neha Neha Rajesh Ram Ramesh Ramesh Ramu Ravi Seetha Smith + | ysql> SELECT + NAME + Aakash | NAME FROM | PASSEN | NGER2 O | RDER BY NAM |
| Geetha Neha Neha Rapesh Ram Ramesh Ramu Ravi Seetha Smith + | ysql> SELECT NAME Aakash Aravind | NAME FROM | PASSEN | NGER2 O | RDER BY NAM |
| Neha Neha Rajesh Ram Ramesh Ramu Ravi Seetha Smith + | ysql> SELECT NAME Aakash Aravind Avinash | NAME FROM | PASSEN | NGER2 O | RDER BY NAM |
| Neha Rajesh Ram Ramesh Ramu Ravi Seetha Smith + | ysql> SELECT NAME Aakash Aravind Avinash Geetha | NAME FROM | PASSEN | NGER2 O | RDER BY NAM |
| Rajesh Ram Ramesh Ramu Ravi Seetha Smith + | ysql> SELECT NAME Aakash Aravind Avinash Geetha Neha | NAME FROM | PASSE | NGER2 O | RDER BY NAM |
| Ram Ramesh Ramesh Ramu Ravi Seetha Smith + | ysql> SELECT NAME Aakash Aravind Avinash Geetha Neha Neha | NAME FROM | PASSE | NGER2 O | RDER BY NAM |
| Ramesh Ramesh Ramu Ravi Seetha Smith + | ysql> SELECT NAME Aakash Aravind Avinash Geetha Neha Neha Rajesh | NAME FROM | PASSE | NGER2 O | RDER BY NAM |
| Ramesh Ramu Ravi Seetha Smith + | ysql> SELECT NAME Aakash Aravind Avinash Geetha Neha Neha Rajesh Ram | NAME FROM | PASSE | NGER2 O | RDER BY NAM |
| Ramu Ravi Seetha Smith + | ysql> SELECT NAME Aakash Aravind Avinash Geetha Neha Neha Rajesh Ram Ramesh | NAME FROM | PASSE | NGER2 O | RDER BY NAM |
| Ravi Seetha Smith + | ysql> SELECT NAME Aakash Aravind Avinash Geetha Neha Neha Rajesh Ram Ramesh Ramesh | NAME FROM | PASSE | NGER2 O | RDER BY NAM |
| Seetha Smith + | ysql> SELECT NAME Aakash Aravind Avinash Geetha Neha Rajesh Ram Ramesh Ramesh Ramesh Ramu | NAME FROM | PASSE | NGER2 O | RDER BY NAM |
| $r_{r_{r_{r_{r_{r_{r_{r_{r_{r_{r_{r_{r_{r$ | ysql> SELECT NAME Aakash Aravind Avinash Geetha Neha Neha Rajesh Ram Ramesh Ramesh Ramu Ramu Ramu | NAME FROM | PASSE | NGER2 O | RDER BY NAM |
| r_{0} r_{0 | ysql> SELECT NAME Aakash Aravind Avinash Geetha Neha Neha Rajesh Ramesh Ramesh Ramesh Ramu Ramu Ravi Seetha | NAME FROM | PASSE | NGER2 O | RDER BY NAM |
| | ysql> SELECT NAME Aakash Aravind Avinash Geetha Neha Raha Rajesh Ramesh Ramesh Ramesh Ramu Ravi Seetha Smith | NAME FROM | PASSE | NGER2 O | RDER BY NAM |

Conclusion: The Student is able execute the Queries from above database.

Viva-Vice:

- 1. What is the result of String functions?
- 2. What is the result of Date functions?
- 3. What is the result of conversion function?
- 4. What is Concatenation?
- 5. What is the difference between LTRIM and RTRIM?

8. Querying Aggregate Functions(COUNT,SUM,AVG,MAX and MIN)

Aim: To Practice Queries using Aggregate functions for the following

- 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.

Recommended Hardware / Software Requirements:

- Hardware Requirements: Intel Based desktop PC with minimum of 166 MHZ or faster processor with at least 1GB RAM and 500 MB free disk space.
- MySQL 5.6.1

Prerequisites: Student must know about the RDBMS-SQL

1. Write a Query to display the information present in the passenger and cancellation tables

MYSQL> CREATE TABLE CANCELLATION2(PNRNO INT PRIMARY KEY,JOURNEYDATE DATETIME, NOOFSEATS INT,ADDRESS VARCHAR(20),CONTACTNO INT,STATUS VARCHAR(10),FOREIGN KEY(PNRNO) REFERENCES RESERVATION2(PNRNO));

mysql> INSERT INTO CANCELLATION2 VALUES(10201,'2012-02-20 10:20:25',2,'HYD',9654235242,'CONFIRM');

mysql> INSERT INTO CANCELLATION2 VALUES(10202,'2012-02-22 10:22:25',2,'HYD',9654232451,'CONFIRM');

mysql> INSERT INTO CANCELLATION2 VALUES(10203,'2012-03-22 10:30:25',2,'DELHI',9654587960,'CONFIRM');

MySQL> SELECT * FROM RESERVATION

UNION

SELECT * FROM CANCELLATION;

| mysql> SE -> UN -> SE | LECT * FROM RESERVATI(IION LECT * FROM CANCELLAT | DN2 ION2; | | | |
|---|---|----------------------------|---|--|---|
| ++ PNRNO | Journeydate | + NoofSeats | + Address | + | + STATUS |
| 10201 10202 10203 10204 10201 10202 10203 | 2012-02-20 10:20:25 2012-02-22 10:22:25 2012-03-22 10:30:25 2013-03-22 11:30:25 2012-02-20 10:20:25 2012-02-22 10:22:25 2012-03-22 10:30:25 | 5 5 5 2 2 2 | HYD HYD DELHI CHENNAI HYD HYD DELHI | 9654235242 9654232451 9654587960 9845761254 9654235242 9654235242 9654232451 9654587960 | NULL NULL NULL NULL CONFIRM CONFIRM CONFIRM |

2. Display the Minimum age of the Passenger

MySQL> SELECT MIN(AGE) as MINAGE FROM PASSENGER;

| mysql> SELECT | * FROM PAS | SENGER | 2; | | | |
|--|--|--|-----|---|--|--|
| passportId | name | Age | Sex | Address | | |
| 145 278 4590 5622 6789 82302 82303 82304 82304 82306 82308 82308 82402 82402 82403 82502 82602 | Ramesh Geetha Ram Seetha Ravi Smith Neha Neha Ramu Aakash Aravind Avinash Ramesh Ramesh Rajesh | 45 36 30 23 23 23 40 40 42 42 23 23 | М | abc123 abc124 abc12 abc55 abc14 Hyderabad Hyderabad Hyderabad Hyderabad Hyderabad Hyderabad Hyderabad Hyderabad Hyderabad Hyderabad | | |
| <pre>+++++++ 14 rows in set (0.00 sec) mysql> SELECT MIN(AGE) as MINAGE FROM PASSENGER2; ++ MINAGE ++ 23 ++ 1 row in set (0.03 sec)</pre> | | | | | | |

3. Find number of tickets booked for each PNR_No using GROUP BY CLAUSE

MySQL> SELECT PNRNO,SUM(No_of_SEATS) AS SUM_OF_SEATS FROM RESERVATION2 GROUP BY PNRNO;

| mysql> SE | LECT * FROM RESERVATIO | DN2; | | | |
|--|--|-----------------------|--------------------------------|--|------------------------------------|
| PNRNO | Journeydate | NoofSeats | Address | CONTACTNO | STATUS |
| 10201 10202 10203 10204 | 2012-02-20 10:20:25 2012-02-22 10:22:25 2012-03-22 10:30:25 2013-03-22 11:30:25 | 5 5 5 5 5 | HYD HYD DELHI CHENNAI | 9654235242 9654232451 9654587960 9845761254 | NULL NULL NULL NULL |
| 4 rows in mysql> SE PNRNO; | set (0.00 sec) | ATS) AS SUM_(| DF_SEATS FF | ROM RESERVATIO | DN2 GROUP BY |
| PNRNO | SUM_OF_SEATS | | | | |
| 10201 10202 10203 10204 | 5 5 5 5 | | | | |
| ++ 4 rows in | set (0.00 sec) | | | | |

4. Find the distinct PNR Numbers that are present.

MySQL> SELECT DISTINCT PNR_NO FROM RESERVATION2;

| mysql> SELECT * FROM RESERVATIO | DN2; | | | |
|--|-------------------------------|--------------------------------|--|------------------------------------|
| PNRNO Journeydate | NoofSeats | Address | CONTACTNO | STATUS |
| 10201 2012-02-20 10:20:25 10202 2012-02-22 10:22:25 10203 2012-03-22 10:30:25 10204 2013-03-22 11:30:25 | 5 5 5 5 5 | HYD HYD DELHI CHENNAI | 9654235242 9654232451 9654587960 9845761254 | NULL NULL NULL NULL |
| 4 rows in set (0.00 sec) mysql> SELECT DISTINCT PNRNO FF | ROM RESERVATI | :on2; | | + |
| PNRNO | | | | |
| 10201 10202 10203 10204 | | | | |
| 4 rows in set (0.00 sec) | | | | |

| 10201 2 10202 2 10203 2 | 2012-02-20 10:20:25 | 2 | + | | |
|-------------------------------------|---|----------|-------------------------|--|-------------------------------|
| 10202 1 4 | 2012-02-22 10:22:23 | 22 | HYD HYD DELHI | 9654235242 9654232451 9654587960 | CONFIRM CONFIRM CONFIRM |
| 3 rows in s nysql> SELE | set (0.00 sec) ECT SUM(NOOFSEATS) FR + SEATS) + | CANCELLA | rion2; | | |
| | 6 + | | | | |

5.Mysql> select sum(Noofseats) from Cancellation2;

Viva-Voce:

- 1. What is the difference between SUM and COUNT ?
- 2. What will you get when you use MIN ?
- 3. What will you get when you use MAX ?
- 4. What is VIEW? and What will you get when you use VIEW
- 5. What is difference between DROP table and DELETE?
- 6. What will you get when you use AVG?

9. Querying using GROUP BY ,HAVING and Creation and Droping of views

Aim: To Practice Queries using Aggregate functions for following

- 5. Find the number of tickets booked by a passenger where the number of seats is greater than 1.
- 6. Find the total number of cancelled seats.

Recommended Hardware / Software Requirements:

- Hardware Requirements: Intel Based desktop PC with minimum of 166 MHZ or faster processor with at least 1GB RAM and 500 MB free disk space.
- MySQL 5.6.1

Prerequisites: Student must know about the RDBMS-SQL

- 5. Find the number of tickets booked by a passenger where the number of seats is greater than 1.
- MySQL> select sum(noofseats) from reservation2 GROUP BY PNRNO HAVING Noofseats >1;

| mysql> SE | ELECT * FROM RESERVATIO | DN2; | r fan in trouwer in trouwer | | |
|--|--|-----------------------|--------------------------------------|--|--|
| PNRNO | Journeydate | NoofSeats | Address | CONTACTNO | STATUS |
| 10201 10202 10203 10204 | 2012-02-20 10:20:25 2012-02-22 10:22:25 2012-03-22 10:30:25 2013-03-22 11:30:25 | 5 5 5 5 5 | HYD HYD DELHI CHENNAI | 9654235242 9654232451 9654587960 9845761254 | ++ NULL NULL NULL NULL |
| 4 rows in mysql> SE HAVING NG + NOOFSE/ + | n set (0.00 sec) ELECT SUM(NOOFSEATS) AS DOFSEATS>1; + ATS + | 5 NOOFSEATS I | ROM RE | SERVATION2 | GROUP BY PNRNO |
| + | 5 5 5 5 | | | | |

6. Find the total number of cancelled seats.

MySQL> select sum(noofseats) as canceled_seats from cancellation2;

| mysql> SE | LECT * FROM CANCELLAT | ton2; | | | |
|---|--|---------------|-------------------------|--|-----------------------------------|
| PNRNO | JOURNEYDATE | NOOFSEATS | ADDRESS | CONTACTNO | STATUS |
| 10201 10202 10203 | 2012-02-20 10:20:25 2012-02-22 10:22:25 2012-03-22 10:30:25 | 2 2 2 | HYD HYD DELHI | 9654235242 9654232451 9654587960 | CONFIRM CONFIRM CONFIRM |
| 3 rows in mysql> se + cancele + | set (0.00 sec) elect sum(noofseats) as + ed_seats + 6 | s canceled_se | eats from | cancellation | 2; |
| + 1 row in | set (0.00 sec) | | | | |

Creation and Droping of Views

mysql> create table students(sid int primary key,name varchar(15),login varchar(15), age int,gpa real);

mysql> create table Enrolled(sid int,cid int,grade varchar(5),primary key(sid,cid),

foreign key(sid) references students(sid));

| mysql> se | elect * fro | om students; | L | L |
|---|---|---|--|---|
| sid | name | login | age | gpa |
| 5000 53650 53666 53688 53831 53832 | Dave Smith jones Smith Madayan Guldu | Dave@cs smith@math jones@cs smith@ee Madayan@music guldu@music | 19 19 18 18 18 11 12 | $\begin{array}{c ccccccccccccccccccccccccccccccccccc$ |
| 6 rows in mysql> se + sid | n set (0.00 elect * fro cid gra | 0 sec) om enrolled; ade | | |
| 53650 53666 53831 53832 | 4 A 3 B 5 C 2 B | + | | |
| + 4 rows in | h set (0.00 | +) sec) | | |

mysql>create view BStudents(name,sid,course) AS SELECT s.name,s.sid,E.cid from

students s,enrolled E where s.sid=e.sid AND E.grade='B';

```
mysql> create view BStudents(name,sid,course) AS SELECT s.name,s.sid,E.cid from
students s,enrolled E where s.sid=e.sid AND E.grade='B';
Query OK, 0 rows affected (0.00 sec)
mysql> select * from Bstudents;
+-----+
| name | sid | course |
+-----+
| jones | 53666 | 3 |
| Guldu | 53832 | 2 |
+----+
2 rows in set (0.03 sec)
```

mysql> create view Goodstudents(sid,gpa) AS select s.sid,s.gpa from students s where gpa > 3.0;

```
mysql> create view Goodstudents(sid,gpa) As select s.sid,s.gpa from students s
where s.gpa > 3.0;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from Goodstudents;
+-----+
| sid | gpa |
+----++
| 5000 | 3.3 |
| 53650 | 3.8 |
| 53666 | 3.4 |
| 53688 | 3.2 |
+----++
4 rows in set (0.00 sec)
```

Syntax: Drop view viewname;

Mysql> Drop view Bstudents;

Mysql> Drop view Goodstudents;

mysql> Drop view Bstudents; Query OK, O rows affected (0.00 sec)

mysql> Drop view Goodstudents; Query OK, 0 rows affected (0.00 sec)

10.Triggers

Aim: Creation of insert trigger, delete trigger and update trigger.

Recommended Hardware / Software Requirements:

- Hardware Requirements: Intel Based desktop PC with minimum of 166 MHZ or faster processor with at least 64MB RAM and 100 MB free disk space.
- MySQL 5.6.1

Prerequisites: Student must know about the Relational Database SQL-Triggers.

MySQL>CREATE TABLE BUS(BUSNO VARCHAR(10) NOT NULL,

SOURCE VARCHAR(10), DESTINATION VARCHAR(10),

CAPACITY INT(2), PRIMARY KEY(BUSNO));

MySQL>INSERT INTO BUS VALUES('AP123','HYD','CHENNAI','40');

| C:\MySQL\bin\mysql.exe | - • • |
|--|----------|
| mysql> CREATE TABLE BUS(BUSNO VARCHAR(10) NOT NULL, -> SOURCE VARCHAR(10), DESTINATION VARCHAR(10), -> CAPACITY INT(2), PRIMARY KEY(BUSNO)); Query OK, 0 rows affected (0.06 sec) | <u>^</u> |
| mysql> INSERT INTO BUS VALUES('AP123','HYD','CHENNAI','40'); Query OK, 1 row affected (0.02 sec) | |
| mysql> | |
| | - |

i) CREATE TABLE BUS_AUDIT1(ID INT NOT NULL AUTO_INCREMENT, SOURCE

VARCHAR(10) NOT NULL, CHANGEDON DATETIME DEFAULT NULL, ACTION

VARCHAR(10) DEFAULT NULL, PRIMARY KEY(ID));



DELIMITER \$\$

CREATE TRIGGER BEFORE_BUS_UPDATE

BEFORE UPDATE ON BUS

FOR EACH ROW

BEGIN

INSERT INTO BUS_AUDIT1

SET action='update',

source=OLD.source,

changedon=NOW();

END\$\$



i)UPDATE :

MySQL>UPDATE BUS SET SOURCE='KERALA' WHERE BUSNO='AP123'\$\$

| C:\MySQL\bin\mysql.exe | ۲. |
|---|----|
| mysql> DELIMITER \$\$ mysql> CREATE TRIGGER BEFORE_BUS_UPDATE -> BEFORE UPDATE ON BUS -> FOR EACH ROW -> BEGIN -> INSERT INTO BUS_AUDIT1 | • |
| -> SET action='update', -> source=OLD.source, -> changedon=NOW<>; -> END\$\$ Query OK, Ø rows affected <0.00 sec> | |
| mysql> UPDATE BUS SET SOURCE='KERALA' WHERE BUSNO='AP123'\$\$ Query OK, 1 row affected (0.03 sec) Rows matched: 1 Changed: 1 Warnings: 0 | |
| mysql> _ | |
| | Ŧ |

| SNo | Source | Changedon | Action |
|-----|----------|---------------------|--------|
| 1 | Banglore | 2014:03:23 12:51:00 | Insert |
| 2 | Kerela | 2014:03:25:12:56:00 | Update |
| 3 | Mumbai | 2014:04:26:12:59:02 | Delete |

ii) INSERT:

CREATE TRIGGER BEFORE_BUS_INSERT

BEFORE INSERT ON BUS

FOR EACH ROW

BEGIN

INSERT INTO BUS_AUDIT1

SET action='Insert',

source=NEW.source,

changedon=NOW();

END\$\$

MYSQL>INSERT INTO BUS VALUES('AP789', 'VIZAG', 'HYDERABAD', 30)\$\$



| SNo | Source | Changedon | Action |
|-----|----------|---------------------|--------|
| 1 | Banglore | 2014:03:23 12:51:00 | Insert |
| 2 | Kerela | 2014:03:25:12:56:00 | Update |
| 3 | Mumbai | 2014:04:26:12:59:02 | Delete |

iii)

CREATE TRIGGER BEFORE_BUS_DELETE

BEFORE DELETE ON BUS

FOR EACH ROW

BEGIN

DELETE FROM BUS_AUDIT1

SET action='Insert',

source=NEW.source,

changedon=NOW();

END\$\$

DELETE FROM BUS WHERE SOURCE='HYDERABAD'\$\$

| SNo | Source | Changedon | Action |
|-----|----------|---------------------|--------|
| 1 | Banglore | 2014:03:23 12:51:00 | Insert |
| 2 | Kerela | 2014:03:25:12:56:00 | Update |
| 3 | Mumbai | 2014:04:26:12:59:02 | Delete |

Examples

CREATE TRIGGER updcheck1 BEFORE UPDATE ON passengerticket

FOR EACH ROW

BEGIN

IF NEW.TicketNO > 60 THEN

SET New.TicketNo = New.TicketNo;

ELSE

SET New.TicketNo = 0;

END IF;

END;

| mysql> select | * from passen | gertick | et;\$\$ | | |
|--|--|-------------|------------|-----------------|--------------|
| passportid | TicketNo | | | | |
| + 145 278 6789 82302 82403 82502 + 6 rows in set mysql> desc pa | 100 200 300 400 500 600 (0.00 sec) | ; \$ \$ | | | |
| + Field | + Туре | + Null | + Key | + Default | + Extra |
| passportid TicketNo | varchar(15) int(11) | NO YES | PRI | NULL | |
| 2 rows in set | (0.00 sec) | + | + | + | + |

```
mysql> CREATE TRIGGER updcheck BEFORE UPDATE ON passengerticket
     -> FOR EACH ROW
     -> BEGIN
     -> IF NEW.TicketNO > 60 THEN
     -> SET New.TicketNo = TicketNo;
     -> ELSE
     -> SET New.TicketNo = 0;
-> END IF;
     -> END;
     -> $$
Query OK, 0 rows affected (0.00 sec)
mysql> update passengerticket set TicketNo=TicketNo-50 where passportid=145;$$
Query OK, 1 row affected (0.03 sec)
Rows matched: 1 Changed: 1 Warnings: 0
mysql> select * from passengerticket;$$
  passportid | TicketNo |
  145
278
6789
                          0
200
300
  82302
82403
                         400
                          500
  82502
                          600
  rows in set (0.00 sec)
6
```

| mysql> select | * from pass | engerticket;\$\$ |
|---|--|--|
| passportid | TicketNo | |
| + 145 278 6789 82302 82403 82502 | 0 200 300 400 500 600 | |
| 6 rows in set | (0.00 sec) | |
| <pre>mysql> CREATE -> FOR EAC -> BEGIN -> IF NEW -> ELSE -> SET Nev -> END IF; -> END; -> \$\$ Query OK, 0 ro </pre> | TRIGGER upo TH ROW TicketNO>60 v.TicketNo=N v.TicketNo=0 ows affected | lcheck BEFORE UPDATE ON passengerticket) THEN lew.TicketNo;); (0.00 sec) |
| Query OK, 1 ro | passengerti ow affected | (0.03 sec) |
| Rows matched: | I Changed: | 1 Warnings: 0 |
| mysql> select | * from pass | engerticket;\$\$ |
| passportid + | TicketNo | |
| 145 278 6789 82302 82403 82502 | 80 200 300 400 500 600 | |
| 6 rows in set | (0.00 sec) | |

Conclusion: The Student is able to work on Triggers and create active database.

Viva-Vice:

- 1. What is TRIGGER?
- 2. What is BEFORE Trigger?
- 3. What is AFTER Trigger?
- 4. What is the difference between BEFORE and AFTER?
- 5. What is ROW and Statement Triggers?
- 6. What is INSTEAD Triggers?
- 7. What are the types on Triggers?
- 8. It is possible to create Trigger on Views?

11.Procedures

Aim: Creation of stored Procedures and Execution of Procedures and Modification of Procedures.

Recommended Hardware / Software Requirements:

- Hardware Requirements: Intel Based desktop PC with minimum of 166 MHZ or faster processor with at least 64MB RAM and 100 MB free disk space.
- MySQL 5.6.1

Prerequisites: Student must know about the Relational Database SQL-Procedures

Ex1:

CREATE PROCEDURE BUS_PROC1()

BEGIN

SELECT * FROM BUS;

END\$\$

CALL BUS_PROC1()\$\$

| C:\MySQ | L\bin\mysql.e | exe | | |
|--|--|--|--------------|---|
| mysq1> mysq1> Cl -> 1 -> 1 Query OK mysq1> Cl | REATE PROG BEGIN SELECT END\$\$, Ø rows a ALL BUS_PI | CEDURE BUS_PROC * FROM BUS; affected <0.00 ROC1<>\$\$ | c1() sec) | • |
| BUSNO | SOURCE | DESTINATION | CAPACITY | |
| AP123 AP789 | KERALA VIZAG | CHENNAI Hyderabad | 40 30 | |
| 2 rows in | n set (0.0 | 00 sec) | | |
| Query OK, | , O rows a | affected (0.00 | sec) | |
| mysql> | | | | |
| | | | | |
| | | | | Ŧ |

Ex2:

CREATE PROCEDURE SAMPLE2()

BEGIN

DECLARE X INT(3);

SET X=10;

SELECT X;

END\$\$

Mysql> CALL SAMPLE2()\$\$

| C:\MySQL\bin\mysql.exe | 3 |
|---|---|
| mysql> CREATE PROCEDURE SAMPLE2<> -> BEGIN -> DECLARE X INT<3>; -> SET X=10; -> SELECT X; -> END\$\$ Query OK, 0 rows affected <0.00 sec> | ^ |
| mysql> mysql> CALL SAMPLE2<>\$\$ ++ ! X | |
| Query OK, 0 rows affected (0.00 sec) | |
| mysql> _ | Ŧ |

Ex3: CREATE PROCEDURE SIMPLE_PROC(OUT PARAM1 INT)

BEGIN

SELECT COUNT(*) INTO PARAM1 FROM BUS;

END\$\$

Mysql> CALL SIMPLE_PROC(@a)\$\$

Mysql> select @a;

| mysql> SE | ELECT * FROM | BUS2; | | |
|---|---|---|----------------------------------|------|
| BusNo | Source | Destination | ĺ | |
| 35 45 55 65 | HYD Tirupathi HYD DELHI | CHENNAI Banglore MUMBAI KOLKATHA | | |
| 4 rows in | n set (0.00 s | sec) | - | |
| mysql> DE mysql> CF -> BE -> SE -> EN Query OK Query OK | ELIMITER \$\$ REATE PROCEDU EGIN ELECT COUNT(* ND \$\$, 0 rows affe ALL SIMPLE_PF , 0 rows affe | URE SIMPLE_PROG) INTO PARAM1 ected (0.00 sec ROC(@a)\$\$ ected (0.03 sec | C(OUT PARAM1 FROM BUS2; C) | INT) |
| mysql> SE | ELECT @a\$\$ | | | |
| @a ++ | | | | |
| 4 ++ | | | | |
| 1 row in | set (0.00 se | ec) | | |

Viva Voce:

- 1. What is a stored procedure?
- 2. When would you use stored procedure or functions ?
- 3. What are external procedures?
- 4. What is input parameter?
- 5. How to use Stored Procedures.

12.Cursors

Aim: 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.

Recommended Hardware / Software Requirements:

- Hardware Requirements: Intel Based desktop PC with minimum of 166 MHZ or faster processor with at least 64MB RAM and 100 MB free disk space.
- MySQL 5.6.1

Prerequisites: Student must know about the Relational SQL-Cursors

Cursors

In MySQL, a cursor allows row-by-row processing of the result sets. A cursor is used for the result set and returned from a query. By using a cursor, you can iterate, or by step through the results of a query and perform certain operations on each row. The cursor allows you to iterate through the result set and then perform the additional processing only on the rows that require it.

In a cursor contains the data in a loop. Cursors may be different from SQL commands that operate on all the rows in the returned by a query at one time.

There are some steps we have to follow, given below :

- Declare a cursor
- Open a cursor statement
- Fetch the cursor
- Close the cursor

1. **Declaration of Cursor** : To declare a cursor you must use the DECLARE statement. With the help of the variables, conditions and handlers we need to declare a cursor before we can use it. first of all we will give the cursor a name, this is how we will refer to it later in the procedure. We can have more than one cursor in a single procedure so its necessary to give it a name that will in some way tell us what its doing. We then need to specify the select statement we want to associate with the cursor. The SQL statement can be any valid SQL statement and it is possible to use a dynamic where clause using variable or parameters as we have seen previously.

Syntax : DECLARE cursor_name CURSOR FOR select_statement;

2. Open a cursor statement : For open a cursor we must use the open statement. If we want to fetch rows from it you must open the cursor.

Syntax : OPEN cursor_name;

3. Cursor fetch statement : When we have to retrieve the next row from the cursor and move the cursor to next row then you need to fetch the cursor.

Synatx : FETCH cursor_name INTO var_name;

If any row exists, then the above statement fetches the next row and cursor pointer moves ahead to the next row.

4 . Cursor close statement : By this statement closed the open cursor.

Syntax: CLOSE_name;

By this statement we can close the previously opened cursor. If it is not closed explicitly then a cursor is closed at the end of compound statement in which that was declared.

Delimiter \$\$ Create procedure p1(in_customer_id int) begin declare v_id int; declare v_name varchar(20); declare v_finished integer default 0; declare c1 cursor for select sid, sname from students where sid=in_customer_id; declare continue handler for NOT FOUND set v_finished=1; open c1; std:LOOP fetch c1 into v_id,v_name; if v_finished=1 then leave std; end if; select concat(v_id,v_name); end LOOP std; close c1; end;

| sid | sname | age | marks |
|-----|--------|-----|-------|
| 1 | ravi | 15 | 25 |
| 2 | ramu | 20 | 30 |
| 2 | rahul | 18 | 26 |
| 5 | kiran | 19 | 28 |
| 6 | varun | 21 | 32 |
| 8 | ramesh | 22 | 33 |
| 8 | xyz | 10 | 20 |




Conclusion: The Student is able to work on Cursors.

Viva Voce:

- 1. What is a cursor?
- 2. What are the types of cursor?
- 3. What is the use of parameterized cursor?
- 4. What is the use of cursor variable?
- 5. What is a normal cursor?
- 6. What are Explicit cursor attributes?

ADDITIONAL PROGRAMMS

EMPLOYEES TABLE

mysql> create table Employees(ssn varchar(15),name varchar(20),lot int,PRIMARY KEY(ssn));

mysql> insert into Employees values('123-22-3666','Attishoo',48);

mysql> insert into Employees values('321-31-5368', 'Smiley', 22);

mysql> insert into Employees values('131-24-3650','Smethurst',35);

| mysql> d | esc Employee | s; | | - | 8 18 |
|--------------------------------|--|----------------------------|---------------------|---------------------------|-------|
| Field | + Туре | Nu] | 1 Key | Default | Extra |
| ssn name lot | varchar(15 varchar(20 int(11) |) NO) YES YES | + PRI | NULL NULL | |
| 3 rows i mysql> s | n set (0.00 elect * from | sec) Employ | ees; | | |
| ssn | name | Ī | lot | | |
| 123-22 131-24 321-31 | -3666 Atti -3650 Smet -5368 Smil | shoo hurst ey | 48 35 22 | | |
| 3 rows i | n set (0.02 | sec) | + | | |

DEPARTMENT TABLE

mysql> create table Departments(did int,dname varchar(10),budget real, PRIMARY KEY(did));

mysql> insert into Departments values(05,'CSE',500000);

mysql> insert into Departments values(04, 'ECE', 400000);

mysql> insert into Departments values(03,'ME',300000);

mysql> insert into Departments values(01,'CE',100000);

| mysql> de | esc Departments | , | | | |
|----------------------------|--|----------------------|----------------|-------------------|---------|
| Field | Туре | Null | + Key | Default | Extra |
| did dname budget | int(11) varchar(10) double | NO YES YES | + PRI | 0 NULL NULL | |
| 3 rows in | n set (0.00 sec | +) | + | +======= | H=====+ |
| mysql> s | elect * from De | partmen | ts; | | |
| did 0 | dname budget | + | | | |
| | CE 100000 ME 300000 ECE 400000 CSE 500000 | | | | |
| 4 rows i | n set (0.00 sec | +) | | | |

Sailors, Reserves, Boats Tables

Mysql> Create table Sailors(Sid integer PRIMARY KEY, sname varchar(15), rating int, age real);

Mysql>Create table Reserves(Sid int,Bid int,Day Date);

Mysql>Create table Boats(Bid int,Bname varchar(15),Color varchar(15);

| mysql> | select * | from | sailo | ors; |
|--|---|--|--|---|
| sid | sname | ra | ting | age |
| ++ 22 29 31 32 58 58 64 71 74 85 95 | Dustin Brutus Lubber Andy Rusty Horatio Zorba Horatio Art Bob | | 7 1 8 10 7 10 9 3 3 | $\begin{array}{c ccccccccccccccccccccccccccccccccccc$ |
| 10 rows | in set | (0.00 | sec) | |
| mysql> | select * | from | reser | ves; |
| sid | bid | day | | |
| 22 22 22 22 22 22 31 31 31 64 64 64 44 44 10 rows | 101 102 103 104 104 102 103 104 101 102 103 ++ in set | 1998 1998 1998 1998 1998 1998 1998 1998 1998 0.00 | -10-10 -10-10 -08-10 -07-10 -10-11 -06-11 -12-11 -05-09 -08-09 -08-09 -08-09 -08-09 -08-09 | |
| mysql> | select * | from | boats | ; |
| bid | bname | 1 | color | . |
| 101 102 103 103 | Interla Interla Clippe Marine | ake ake r | blue red greer red | |

mysql> select S.sname from sailors S, reserves R where S.sid=R.sid AND R.bid=103 ; +-----+ | sname | +-----+ | Dustin | | Lubber | +-----+ 2 rows in set (0.00 sec)

mysql> select S.sname from sailors S, reserves R where S.sid=R.sid AND R.bid=103;

mysql> select sname from sailors s,Reserves R where S.sid=R.sid AND bid=103;

mysql> select R.sid from Boats B,Reserves R where B.bid=R.bid AND B.color='red';

| mysql> select | sname from | sailors s,Reser | ves R where S.sid= | R.sid AND bid=103; |
|--|------------|-----------------|--------------------|------------------------|
| sname | | | | |
| Dustin Lubber | | | | |
| 2 rows in set | (0.00 sec) | | | |
| mysql≻ select | R.sid from | Boats B,Reserve | s R where B.bid=R. | bid AND B.color='red'; |
| ++ sid | | | | |
| 22 22 31 31 64 44 | | | | |
| ++ 6 rows in set | (0.00 sec) | | | |

mysql> select S.sname from sailors S,reserves R,Boats B where S.sid=R.sid AND R.bid=B.bid

AND B.color='red';

mysql> select B.color from Sailors S,Reserves R,Boats B where S.sid=R.sid AND R.bid=B.bid

AND S.sname='Lubber';

mysql> select S.sname from sailors S,reserves R,Boats B where S.sid=R.sid AND R.bid=B.bid AND B.color='red'; sname Dustin Dustin Lubber Lubber Horatio rows in set (0.00 sec) mysql> select B.color from Sailors S,Reserves R,Boats B where S.sid=R.sid AND R.bid=B.bid AND S.sname='Lubber'; color red green red rows in set (0.00 sec)

mysql> select S.sname,S.rating+1 AS rating from Sailors S,Reserves R1,Reserves R2 where S.sid=R1.sid AND S.sid=R2.sid AND R1.day=R2.day AND R1.bid<>R2.bid;

mysql> select S1.sname AS name1,S2.sname AS name2 from sailors S1,sailors S2

where 2*S1.rating=S2.rating-1;

mysql> select S.sname,S.rating+1 AS rating from Sailors S,Reserves R1,Reserves R2 where S.sid=R1.sid AND S.sid=R2.sid AND R1.day=R2.day AND R1.bid<>R2.bid; | rating sname 8 Dustin | 8 Dustin rows in set (0.00 sec) mysql> select S1.sname AS name1,S2.sname AS name2 from sailors S1,sailors S2 where 2*S1.rating=S2.rating-1; name1 | name2 Art Dustin Bob Dustin Horatio Art Horatio Bob Brutus Art Bob Brutus rows in set (0.02 sec) mysql> select S.age from sailors S where S.sname LIKE 'B_%B'; age 63.5 I 1 row in set (0.00 sec) mysql> select S.sname from sailors S where S.sname LIKE 'B_%B'; sname Bob row in set (0.00 sec)

USING UNION, INTERSECT, AND EXCEPT

1). Find the names of sailors who have reserved a red or a green boat.

| <pre>mysql> SELECT S.SNAME FROM SAILORS S,RESERVES R,BOATS B -> WHERE S.SID=R.SID AND R.BID=B.BID -> AND(B.COLOR='red' OR B.COLOR='green');</pre> | |
|--|--|
| SNAME | |
| Dustin Dustin Dustin Lubber Lubber Lubber | |
| 7 rows in set (0.01 sec) | |

OR



2). Find the names of sailors who have reserved both a red and a green boat.

SELECT S.SNAME FROM SAILORS S,RESERVES R,BOATS B WHERE S.SID=R.SID AND R.BID=B.BID AND B.COLOR='red' INTERSECT SELECT S2.SNAME FROM SAILORS S2,RESERVES R2,BOATS B2 WHERE S2.SID=R2.SID AND R2.BID=B2.BID AND B2.COLOR='green';

NESTED QUERIES

Find the Names of sailors who have reserved boat 103



Find the names of Sailors who have reserved a red Boat

| mysgl> SELECT S.SNAME FROM SAILORS S |
|---|
| -> WHERE S.SID IN (SELECT R.SID FROM RESERVES R |
| -> WHERE R.BID IN (SELECT B.BID FROM BOATS B |
| -> WHERE B.COLOR='RED')); |
| ++ |
| SNAME |
| ++ Dustin Lubber Horatio ++ 3 rows in set (0.00 sec) |

Find the names of Sailors who have NOT reserved a red Boat

| <pre>mysql> select s.sname from sailors s -> where s.sid NOT IN (select r.sid from reserves -> where r.bid IN (select b.bid from boats b -> where b color='red'));</pre> | r |
|--|---|
| ++ | |
| ++ | |
| Brutus Andv | |
| Rusty | |
| Horatio | |
| Art Bob | |
| ++ 7 rows in set (0.00 sec) | |

Correlated Nested Queries:

Find the names of Sailors who have reserved a red Boat



Set Comparison Operators:

Find sailors whose rating is better than some sailor called Horatio

```
mysql> select s.sid from sailors s
    -> where s.rating > ANY ( select s2.rating from sailors s2
    -> where s2.sname='Horatio');
+----+
| sid |
+----+
| 31 |
| 32 |
| 58 |
| 71 |
| 74 |
+----+
5 rows in set (0.00 sec)
```

Find the sailors with the highest rating.

mysql> SELECT S.sid FORM Sailors WHERE S.rating>=ALL(SELECT S2.rating FROM

Sailors S2);

The GROUP BY and HAVING Clauses:

Find the age of the youngest sailor for each rating level.



Find the age of the youngest sailor who is eligible to vote for each rating level with at least two such sailors



For each red boat, find the number of reservations for this boat

| mysql> | SELECT B.BID,COUNT(*) AS SAILORCOUNT |
|--------|--------------------------------------|
| -> | FROM BOATS B, RESERVES R |
| -> | WHERE R.BID=B.BID AND B.COLOR='RED' |
| -> | GROUP BY B.BID; |
| + | -++ |
| BID | SAILORCOUNT |
| + | -++ |
| 102 | 3 |
| 103 | 3 |
| + | -++ |
| 2 rows | in set (0.00 sec) |
| | |

Find the average age of sailors for each rating level that has at least two sailors

