DBMS PRACTICALS
–
LAB RECORD

BY

Name of the Student: ___________________________________

HALL-TICKET NO:

1	4	0	1	2	3	6	7	2			

AL-QURMOSHI

Institute Of Business Management

(Approved by AICTE & Affiliated to OSMANIA UNIVERSITY)

***Al-Qurmoshi* Institute of Business Management**

(Approved by AICTE, Affiliated to Osmania University)

#18-11-26/7, Jamal Banda, Barkas, Hyderabad -05

Date: 31-01-2025

CERTIFICATE

This is to certify that *Mr./Ms.*____ bearing **HALL-TICKET NO: 1401-23-672-**____studying MBA III-Semester for the academic year 2024-25 has done *Data Base Management System Practicals-Lab work* as a part of MBA curriculum.

Dr. Ali Shabbir
Principal

Signature of The Internal Examiner With Date

Signature of The External Examiner With Date

INDEX

S.No	Topic	Page no	Date	Sign
1	E-R Model: Analyze the problem with the entities which identify data persisted in the database which contains entities, attributes.	9-10		
2	Concept design with E-R Model: Apply cardinalities for each relationship, identify strong entities and weak entities for relationships like generalization, aggregation, specialization.	11		
3	Relation Model: Represent attributes as columns in tables and different types of attributes like Composite, Multi-valued, and Derived.	12-13		
4	Normalization	14-15		
5	Installation of MySql and MONGO DB and practicing DDL commands	16-34		
6	Practicing DML commands SELECT, INSERT, UPDATE, DELETE.	35-38		
7	Querying Queries using ANY, ALL, IN, INTERSECT, UNION	39-43		
8	Querying Using aggregate functions COUNT, SUM using GROUPBY andHAVING.	44-48		
9	TRIGGER	49-53		
10	Procedures and Stored Procedures Creation, Execution and Modification of stored Procedure	54-59		
11	DCL Commands	60-61		
12	CASESTUDY 1	62		
13	CASESTUDY 2	63		

Introduction

Developing databases is a very important task to develop a system. Before going to form exact database tables and establishing relationships between them, we conceptually or logically can model our database using ER diagrams.

In this experiment we will learn how to find the entities, its attributes and how the relationships between the entities can be established for a system.

Entity Relationship Model

Entity-Relationship model is used to represent a logical design of a database to be created. In ER model, real world objects (or concepts) are abstracted as entities, and different possible associations among them are modeled as relationships.
For example, student and school -- they are two entities. Students study in school. So, these two entities are associated with a relationship "Studies in".
As another example, consider a system where some job runs every night, which updates the database. Here, job and database could be two entities. They are associated with the relationship "Updates".

Entity Set and Relationship Set

An entity set is a collection of all similar entities. For example, "Student" is an entity set that abstracts all students. Ram, John are specific entities belonging to this set. Similarly, a "Relationship" set is a set of similar relationships.

Attributes of Entity

Attributes are the characteristics describing any entity belonging to an entity set. Any entity in a set can be described by zero or more attributes.
For example, any student has got a name, age, an address. At any given time a student can study only at one school. In the school he would have a roll number, and of course a grade in which he studies. These data are the attributes of the entity set Student.

Keys

One or more attribute(s) of an entity set can be used to define the following keys:

- **Super key:** One or more attributes, which when taken together, helps to uniquely identify an entity in an entity set. For example, a school can have any number of students. However, if we know grade and roll number, then we can uniquely identify a student in that school.
- **Candidate key:** It is a minimal subset of a super key. In other words, a super key might contain extraneous attributes, which do not help in identifying an object uniquely. When such attributes are removed, the key formed so is called a candidate key.
- **Primary key:** A database might have more than one candidate key. Any candidate key chosen for a particular implementation of the database is called a primary key.
- **Prime attribute:** Any attribute taking part in a super key

Weak Entity

An entity set is said to be weak if it is dependent upon another entity set. A weak entity can't be uniquely identified only by it's attributes. In other words, it doesn't have a super key.

For example, consider a company that allows employees to have travel allowance for their immediate family. So, here we have two entity sets: employee and family, related by "Can claim for". However, family doesn't have a super key. Existence of a family is entirely dependent on the concerned employee. So, it is meaningful only with reference to employee.

Entity Generalization and Specialization

Once we have identified the entity sets, we might find some similarities among them. For example, multiple person interacts with a banking system. Most of them are customers, and rest employees or other service providers. Here, customers, employees are persons, but with certain specializations. Or in other way, person is the generalized form of customer and employee entity sets.

ER model uses the "ISA" hierarchy to depict specialization (and thus, generalization).

Mapping Cardinalities

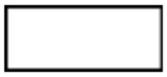
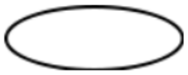
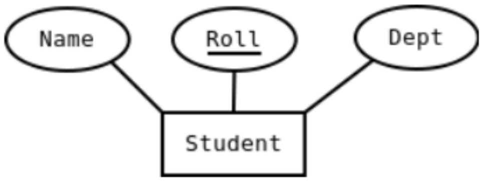
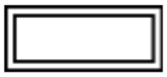
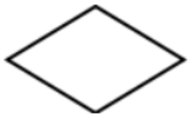
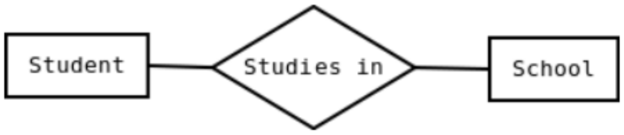
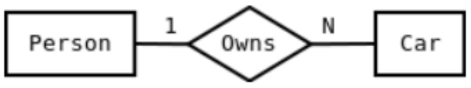
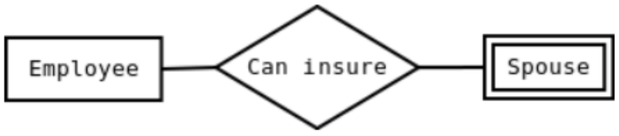
One of the main tasks of ER modeling is to associate different entity sets. Let's consider two entity sets E1 and E2 associated by a relationship set R. Based on the number of entities in E1 and E2 are associated with, we can have the following four type of mappings:

- **One to one:** An entity in E1 is related to at most a single entity in E2, and vice versa
- **One to many:** An entity in E1 could be related to zero or more entities in E2. Any entity in E2 could be related to at most a single entity in E1.
- **Many to one:** Zero or more number of entities in E1 could be associated to a single entity in E2. However, an entity in E2 could be related to at most one entity in E1.
- **Many to many:** Any number of entities could be related to any number of entities in E2, including zero, and vice versa.

ER Diagram

From a given problem statement we identify the possible entity sets, their attributes, and relationships among different entity sets. Once we have these information, we represent them pictorially, called an entity-relationship (ER) diagram.

Graphical Notations for ER Diagram

Term	Notation	Remarks
Entity set		Name of the set is written inside the rectangle
Attribute		Name of the attribute is written inside the ellipse
Entity with attributes		Roll is the primary key; denoted with an underline
Weak entity set		
Relationship set		Name of the relationship is written inside the diamond
Related enity sets		
Relationship cardinality		A person can own zero or more cars but no two persons can own the same car
Relationship with weak entity set		

Importance of ER modeling

Figure - 01 shows the different steps involved in implementation of a (relational) database.

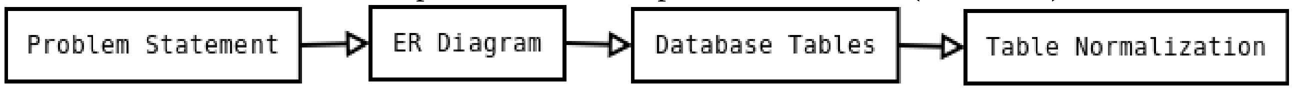


Figure - 01: Steps to implement a RDBMS

Given a problem statement, the first step is to identify the entities, attributes and relationships. We represent them using an ER diagram. Using this ER diagram, table structures are created, along with required constraints. Finally, these tables are normalized in order to remove redundancy and maintain data integrity. Thus, to have data stored efficiently, the ER diagram is to be drawn as much detailed and accurate as possible.

What is MYSQL

MySQL is a relational database management system based on the Structured Query Language, which is the popular language for accessing and managing the records in the database. MySQL is open-source and free software under the GNU license. It is supported by Oracle Company.

MySQL is a Relational Database Management System (RDBMS) software that provides many things, which are as follows:

It allows us to implement database operations on tables, rows, columns, and indexes.

It defines the database relationship in the form of tables (collection of rows and columns), also known as relations.

It provides the Referential Integrity between rows or columns of various tables.

It allows us to updates the table indexes automatically.

It uses many SQL queries and combines useful information from multiple tables for the end-users

MySQL is named after the daughter of co-founder Michael Widenius whose name is "My".

To communicate with Oracle, mysql supports the following categories of commands:

1. **Data Definition Language**

Create, Alter, Drop and Truncate

2. **Data Manipulation Language**

Insert, Update, Delete and Select

3. **Transaction Control Language**

Commit, Rollback and Save point

4. **Data Control Language**

Grant and Revoke

MySQL uses many different data types broken into three categories −

- Numeric
- Date and Time
- String Types.

ROADWAY TRAVELS

Roadway Travels: "Roadway Travels" is in business since 1997 with several buses connecting different places in India. Its main office is located in Hyderabad. The company wants to computerize its operations in the following areas:

- Reservations and Ticketing
- Cancellations

- **Reservations &Cancellation:**

Reservations are directly handled by booking office. Reservations can be made 30 days in advance and tickets issued to passenger. One Passenger/person can book many tickets (to his/her family).

- Cancellations are also directly handed at the booking office.

In the process of computerization of Roadway Travels you have to design and develop a Database which consists the data of Buses, Passengers, Tickets, and Reservation and cancellation details. You should also develop query's using SQL to retrieve the data from the database.

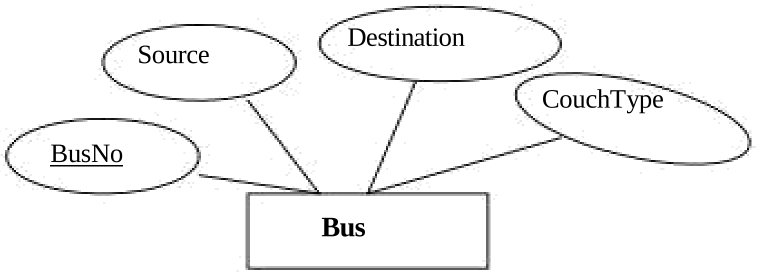
EXPERIMENT-1

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

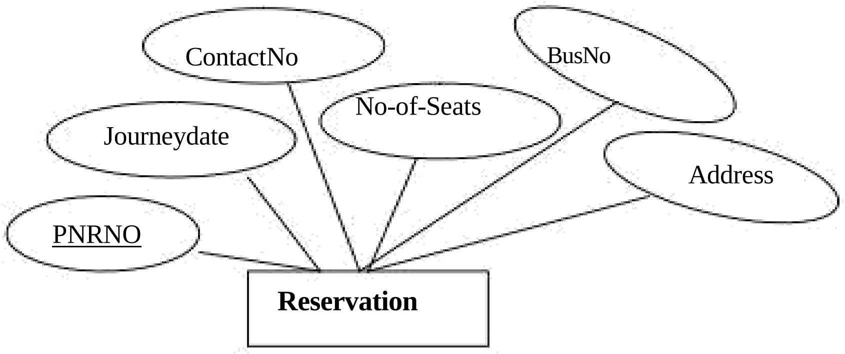
The Following are the entities:

1. **Bus**
2. **Reservation**
3. **Ticket**
4. **Passenger**
5. **Cancellation**

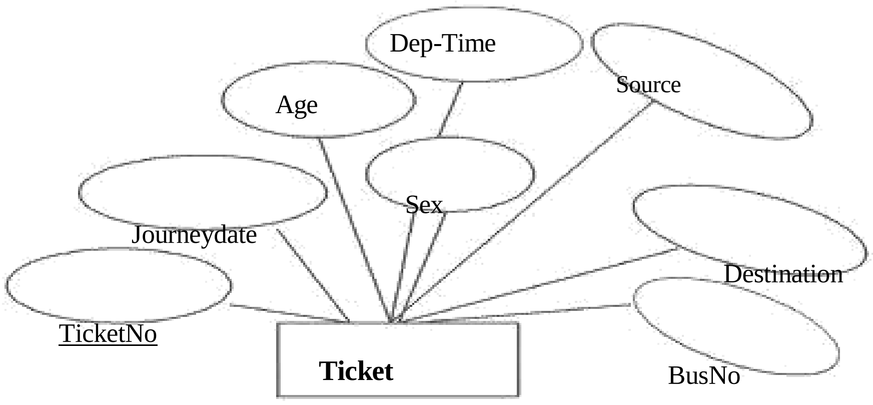
TheattributesintheEntities:Bus:(Entity)



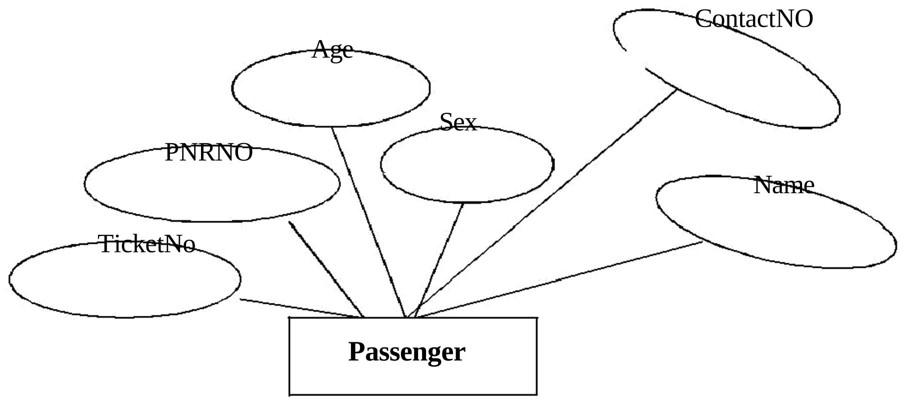
Reservation(Entity)



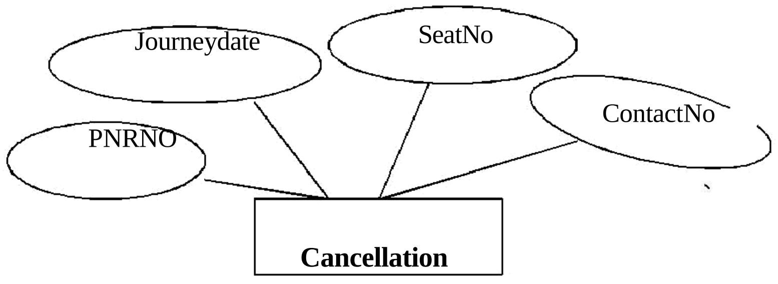
Ticket:(Entity)



Passenger:

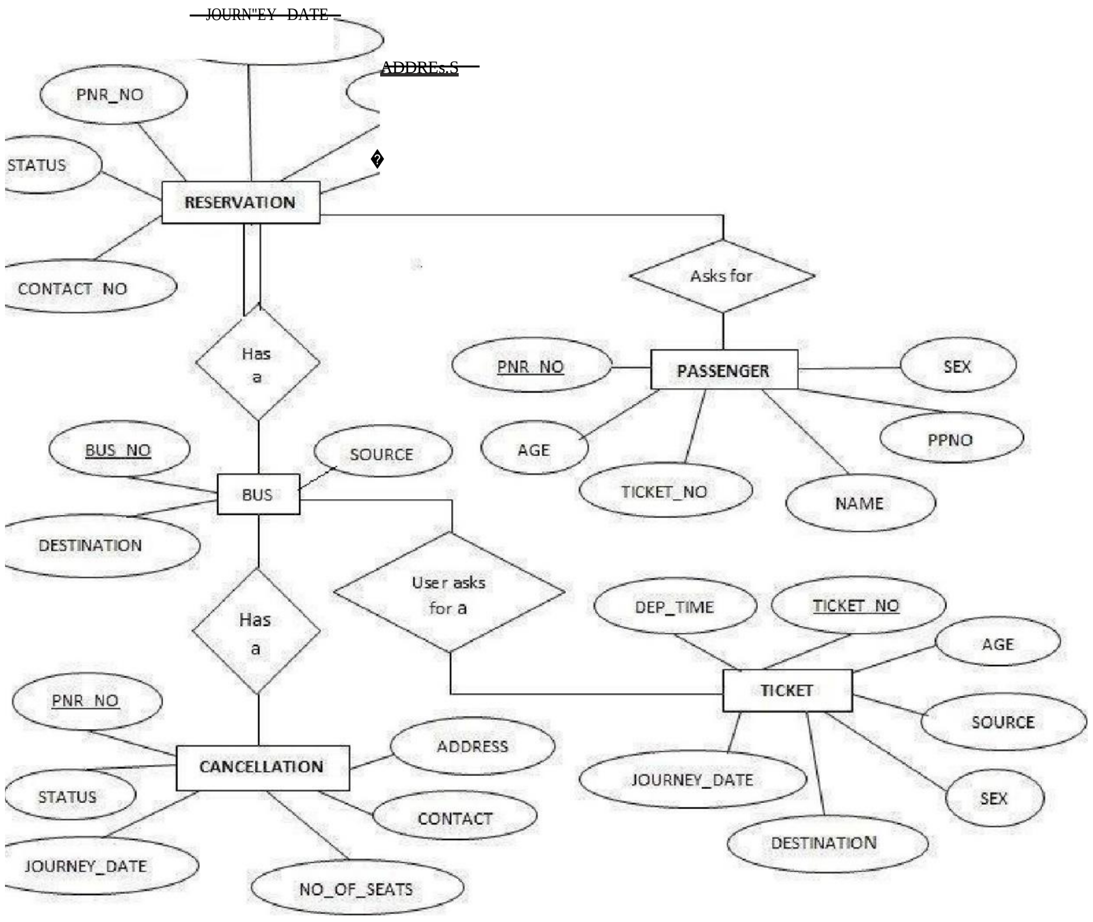


Cancellation(Entity)



EXPERIMENT- 2

Concept design with E-R Model:



EXPERIMENT: 3
Relational Model

Represent all entities and all relationships ina tabular fashion

The following are tabular representation of the above entities and relationships

BUS:

COLUMNNAME	**DATATYPE**	**CONSTRAINT**
BusNo	varchar2(10)	**PrimaryKey**
Source	varchar2(20)	
Destination	varchar2(20)	
CoachType	varchar2(20)	

Reservation:

COLUMNNAME	**DATATYPE**	**CONSTRAINT**
PNRNo	number(9)	**PrimaryKey**
Journeydate	Date	
No-of-seats	integer(8)	
Address	varchar2(50)	
ContactNo	Number(9)	Should be equal to 10 Numbers and not allow Other than numeric
BusNo	varchar2(10)	**Foreign key**
Seatno	Number	

Ticket:

COLUMNNAME	**DATATYPE**	**CONSTRAINT**
Ticket_No	number(9)	**PrimaryKey**
Journeydate	Date	
Age	int(4)	
Sex	Char(10)	
Source	varchar2(10)	
Destination	varchar2(10)	
Dep-time	varchar2(10)	
BusNo	Number2(10)	

Passenger

COLUMNNAME	**DATATYPE**	**CONSTRAINT**
PNRNo	Number(9)	**PrimaryKey**
TicketNo	Number(9)	Foreignkey
Name	varchar2(15)	
Age	integer(4)	
Sex	char(10)	(Male/Female)
Contactno	Number(9)	Should be equal to 10numbers And not allow other than numeric

Cancellation:

COLUMNNAME	**DATATYPE**	**CONSTRAINT**
PNRNo	Number(9)	Foriegn-key
Journey-date	Date	
Seatno	Integer(9)	
Contact_No	Number(9)	Should be equal to 10numbers And not allow other than numeric

EXPERIMENT: 4

Normalization

Database normalization is a technique for designing relational database tables to minimize duplication of information and, in so doing, to safeguard the database against certain types oflogical or structural problems, namely data anomalies.

For example, when multiple instances of a given piece of information occur in a table, the possibility exists that these instances will not be kept consistent when the data within the table is updated, leading to a loss of data integrity.

A table that is sufficiently normalized is less vulnerable to problems of this kind, because its structure reflects the basic assumptions for when multiple instances of the same information should be 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 eg 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 represented by a single instance only.

Normalized tables are:-

```
mysql> create table Bus(BusNo varchar(20) primary key,Source varchar(20),Destination varchar(20));

mysql>Create table passenger(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 Reservation(PNRNO integer Primary key, JourneyDate DateTime,NoofSeats int,Address varchar(20),Contact No Integer);
```

```
mysql> create table Cancellation(PNRNO Integer primary key,JourneyDate DateTime,NoofSeats Integer,Address varchar(20), ContactNo Integer, foreign key(PNRNO) references Reservation(PNRNO));
```

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

VIVA QUESTIONS

1. Explain the need of normalization?
2. What is functional dependency?
3. Explain difference between third normal form and boyce codd normal form?
4. What is PJNF?
5. What is transitivity dependency?

EXPERIMENT-5

AIM: Installation of MySQL and practicing DDL & DML commands.

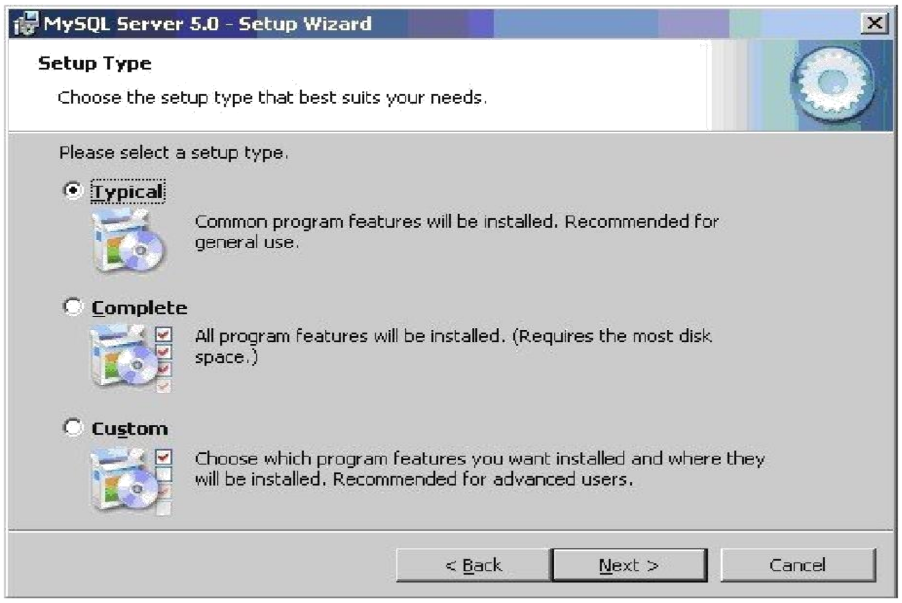
1. Steps for installing MySQL

Step1

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

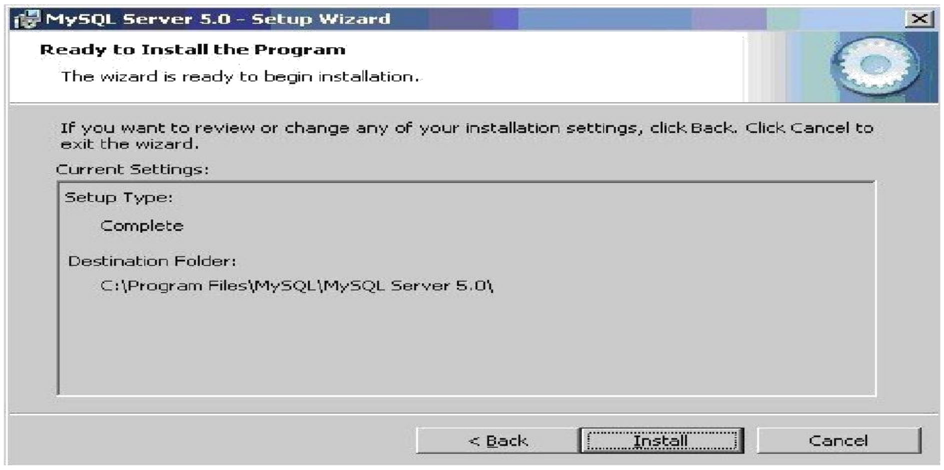
Step2

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
Setup Type
Choose the setup type that best suits your needs.
Please select a setup type.
Typical
Common program features will be installed. Recommended for general use.
Complete
All program features will be installed. (Requires the most disk space.)
Custom
Choose which program features you want installed and where they will be installed. Recommended for advanced users.
< Back
Next >
Cancel

Step 3

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**.


MySQL Server 5.0 - Setup Wizard
Ready to Install the Program
The wizard is ready to begin installation.
< Back
Install
Cancel

If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.

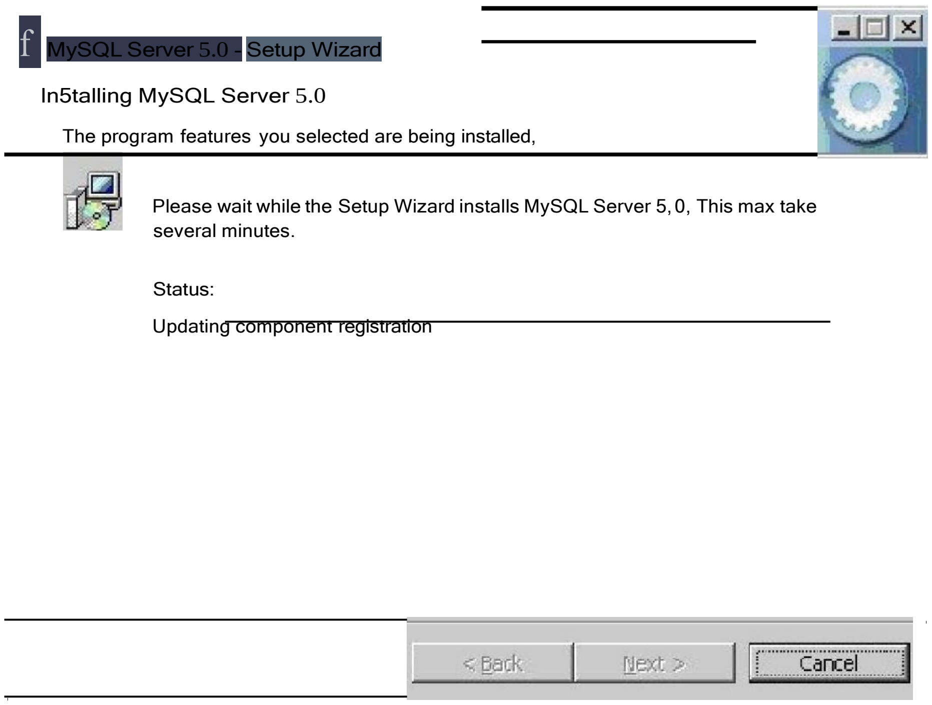
Current Settings:

Setup Type:
Complete

Destination Folder:
C:\Program Files\MySQL\MySQL Server 5.0\

Step 4

This wizard is ready to begin installation. Destination folder will be in **C:\ProgramFiles\MySQL\MySQLServer5.0**\.Tocontinue,click**next**.


MySQL Server 5.0 - Setup Wizard

In5talling MySQL Server 5.0

The program features you selected are being installed,

Please wait while the Setup Wizard installs MySQL Server 5, 0, This max take several minutes.

Status:

Updating component registration

Step 5

5QL Enterprise

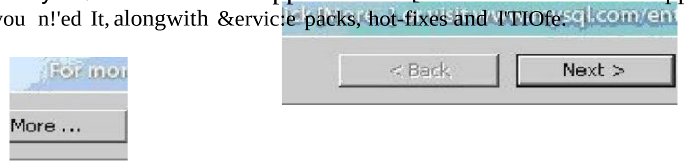
AMySQl Enterpris,esubscription Ls the mos,t comprebe.nsive offering of MySQL darabas,e :software, service., and support to ensure your busln•es;5achieves the h lghest levels of rellablUzy, security and uptime.

An Ente1p1lse SubiScrlptlon Indu:des:

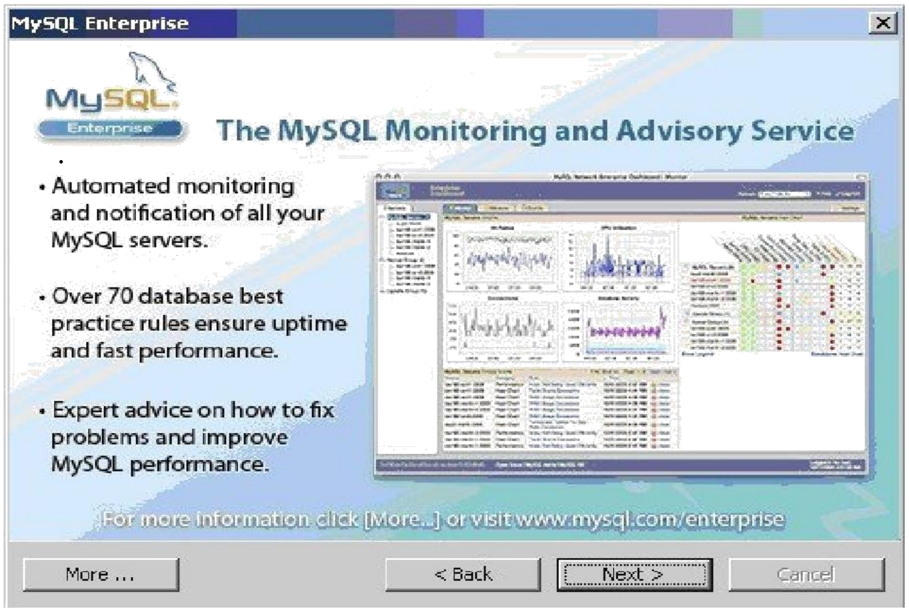
1. The **MySQL !;nterprlse Server** - The most rnliab s, ecure, and up-to--,date versio,nof the,worlds most popular open source database.

2. The **MySQL Monitoring and Advisory S@rvlce-An** automa�d virtual database assistant

3. MySQL Production Support- T&hn[cal and cons1.1ltat esupportwhen you n!'ed It, alongwith &ervic:e packs, hot-fixes and TTIOfe.



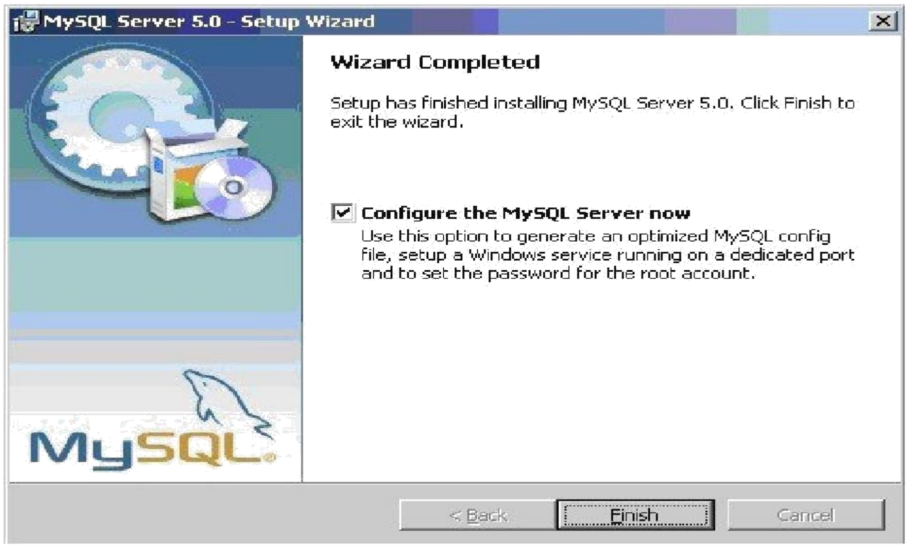
Step 6: To continue, click **next**


MySQL Enterprise

The MySQL Monitoring and Advisory Service

- Automated monitoring and notification of all your MySQL servers.
- Over 70 database best practice rules ensure uptime and fast performance.
- Expert advice on how to fix problems and improve MySQL performance.

For more information click [More...] or visit www.mysql.com/enterprise


MySQL Server 5.0 - Setup Wizard

Wizard Completed

Setup has finished installing MySQL Server 5.0. Click Finish to exit the wizard.

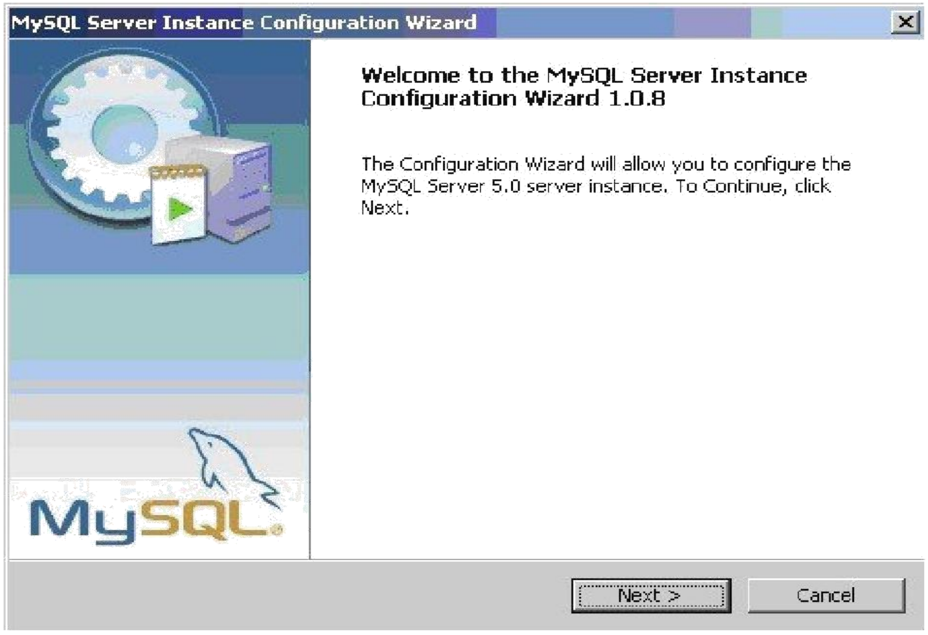
[✓] Configure the MySQL Server now

Use this option to generate an optimized MySQL config file, setup a Windows service running on a dedicated port and to set the password for the root account.

Step 8:

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

Step 9: The configuration wizard will allow you to configure the MySQL Server 5.0 server instance.To continue, click **next**

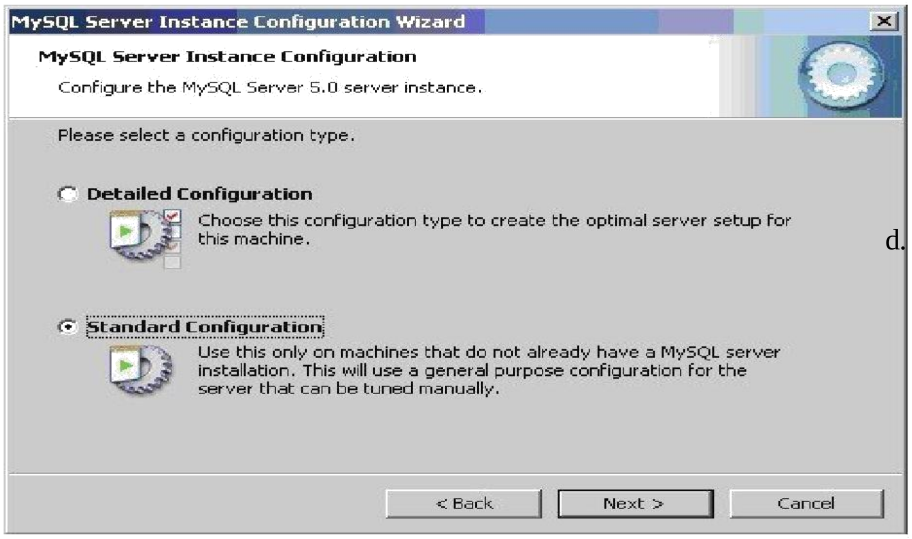


MySQL Server Instance Configuration Wizard

Welcome to the MySQL Server Instance Configuration Wizard 1.0.8

The Configuration Wizard will allow you to configure the MySQL Server 5.0 server instance. To Continue, click Next.

Step 10 :Select a standard configuration and this will use a general purpose configuration for the server that can be tuned manually. To continue, click next



MySQL Server Instance Configuration Wizard

MySQL Server Instance Configuration

Configure the MySQL Server 5.0 server instance.

Please select a configuration type.

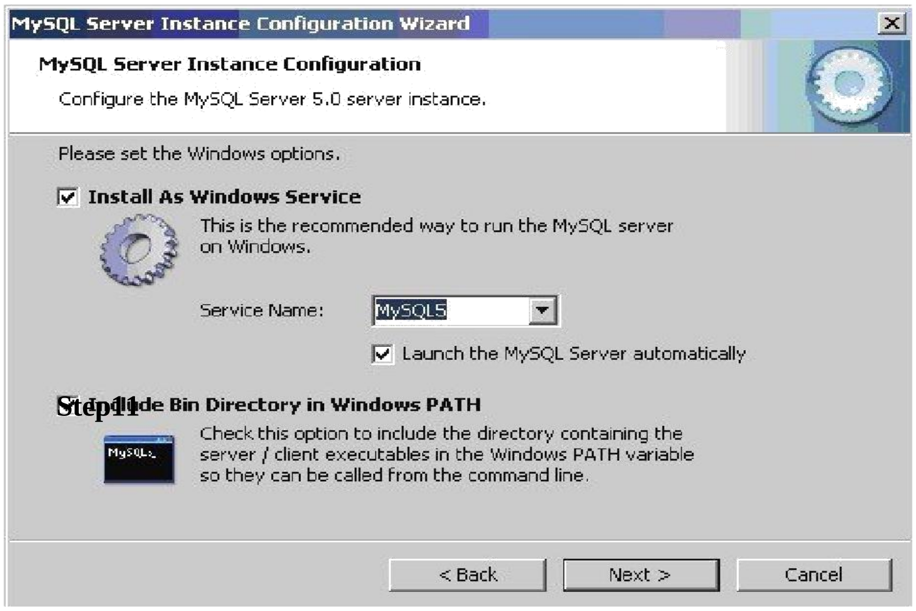
○ **Detailed Configuration**

Choose this configuration type to create the optimal server setup for this machine.

● **Standard Configuration**

Use this only on machines that do not already have a MySQL server installation. This will use a general purpose configuration for the server that can be tuned manually.

d.



MySQL Server Instance Configuration Wizard

MySQL Server Instance Configuration

Configure the MySQL Server 5.0 server instance.

Please set the Windows options.

☑ Install As Windows Service

This is the recommended way to run the MySQL server on Windows.

Service Name: MySQL5

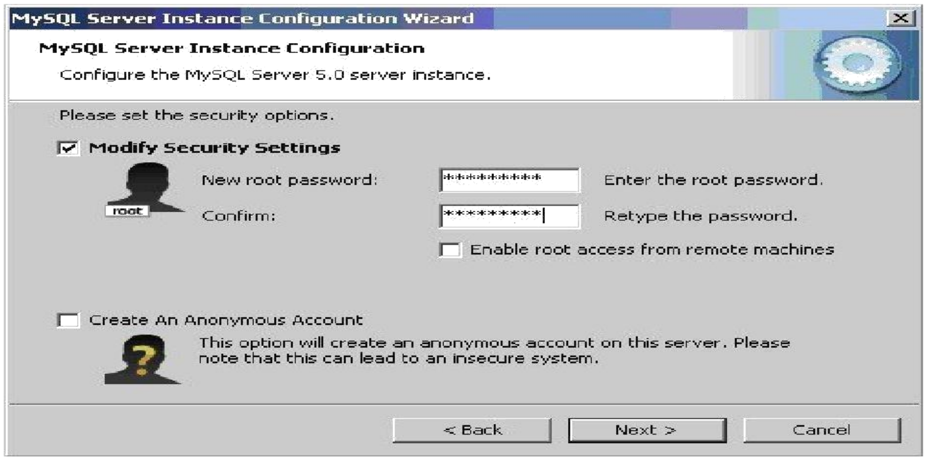
☑ Launch the MySQL Server automatically

Step11

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.

Step:11 Check on the **install as windows service** and **include bin directory in windows path**. Tocontinue,click**next**.



MySQL Server Instance Configuration Wizard

MySQL Server Instance Configuration

Configure the MySQL Server 5.0 server instance.

Please set the security options.

☑ Modify Security Settings

New root password: *********** Enter the root password.

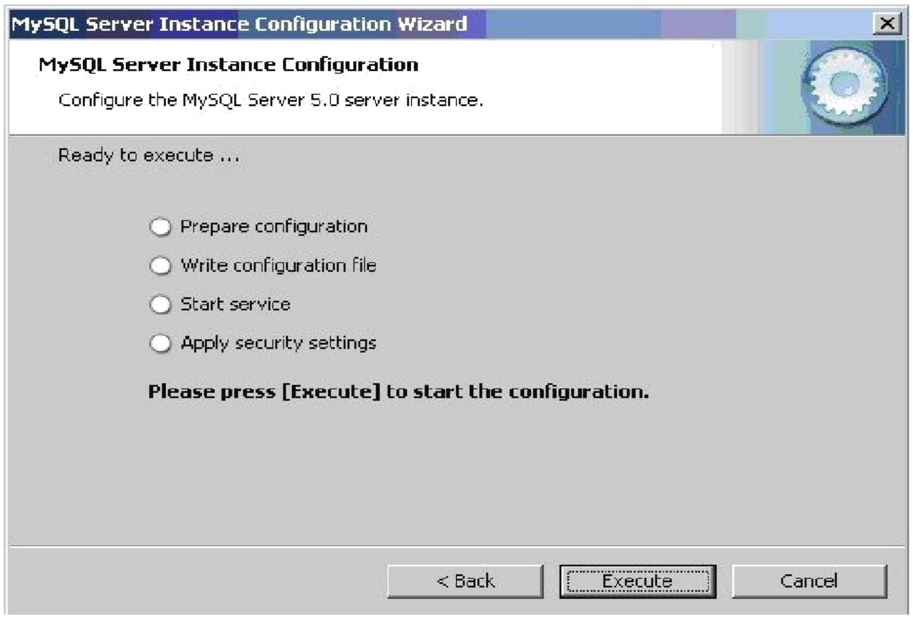
Confirm: ********** Retype the password.

☐ Enable root access from remote machines

☐ Create An Anonymous Account

This option will create an anonymous account on this server. Please note that this can lead to an insecure system.

Step12: Please set the security options by entering the root password and confirm retype the password.continue, click next.


MySQL Server Instance Configuration Wizard

MySQL Server Instance Configuration

Configure the MySQL Server 5.0 server instance.

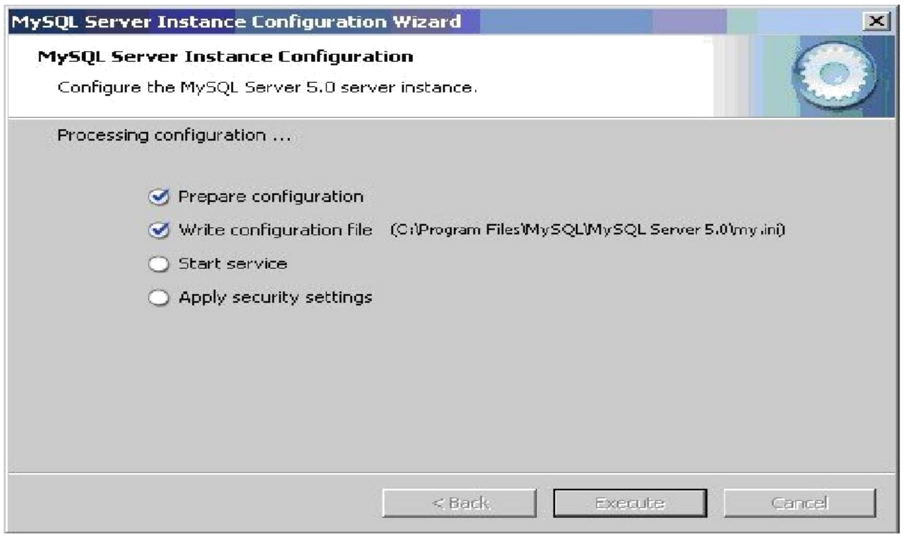
Ready to execute ...

- ○ Prepare configuration
- ○ Write configuration file
- ○ Start service
- ○ Apply security settings

Please press [Execute] to start the configuration.

Step13

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


MySQL Server Instance Configuration Wizard

MySQL Server Instance Configuration

Configure the MySQL Server 5.0 server instance.

Processing configuration ...

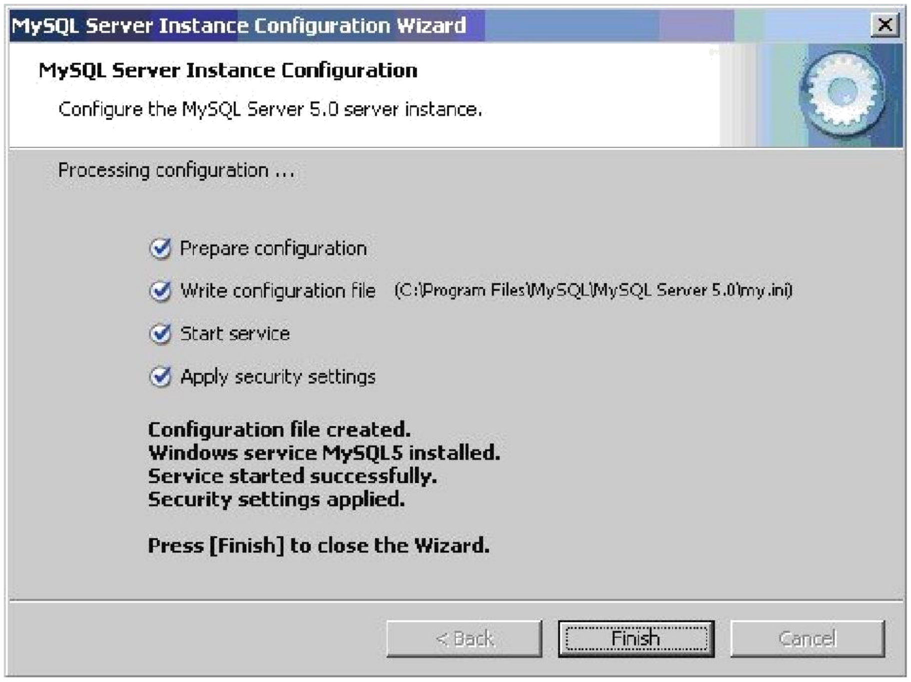
- ✓ Prepare configuration
- ✓ Write configuration file (C:\Program Files\MySQL\MySQL Server 5.0\my.ini)
- ○ Start service
- ○ Apply security settings

Step14

Processing configuration in progress.

Step15

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


MySQL Server Instance Configuration Wizard
MySQL Server Instance Configuration
Configure the MySQL Server 5.0 server instance.
< Back
Finish
Cancel

Processing configuration ...

- ✓ Prepare configuration
- ✓ Write configuration file (C:\Program Files\MySQL\MySQL Server 5.0\my.ini)
- ✓ Start service
- ✓ Apply security settings

Configuration file created.
Windows service MySQL5 installed.
Service started successfully.
Security settings applied.

Press [Finish] to close the Wizard.

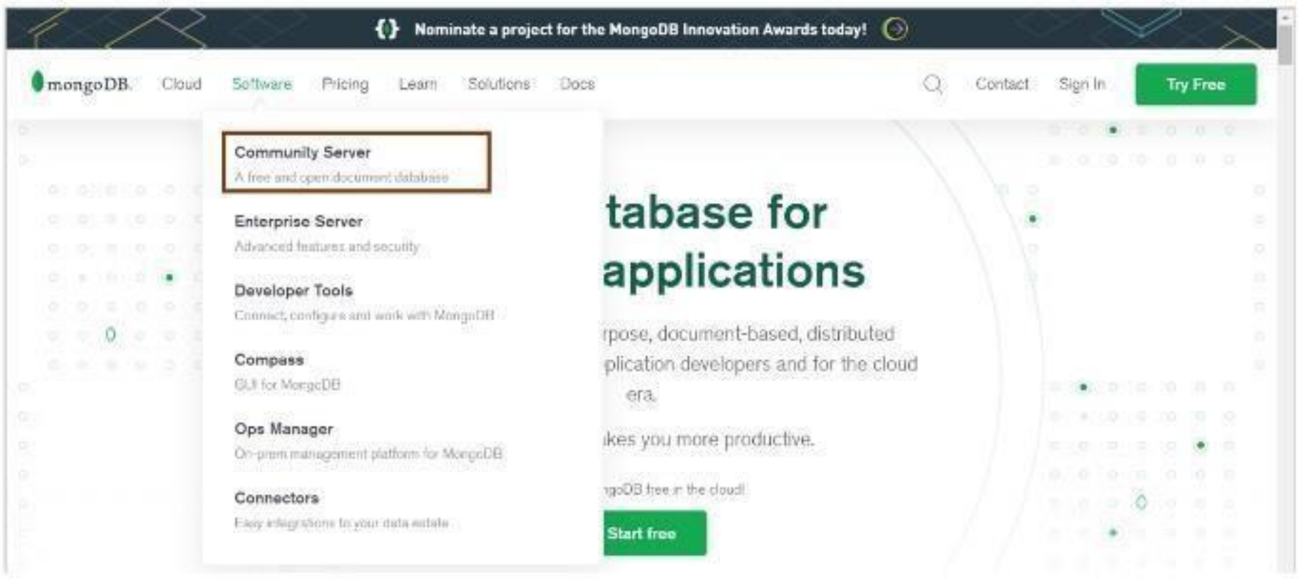
5(b) :Installation of MangoDB

Navigate to the download site:

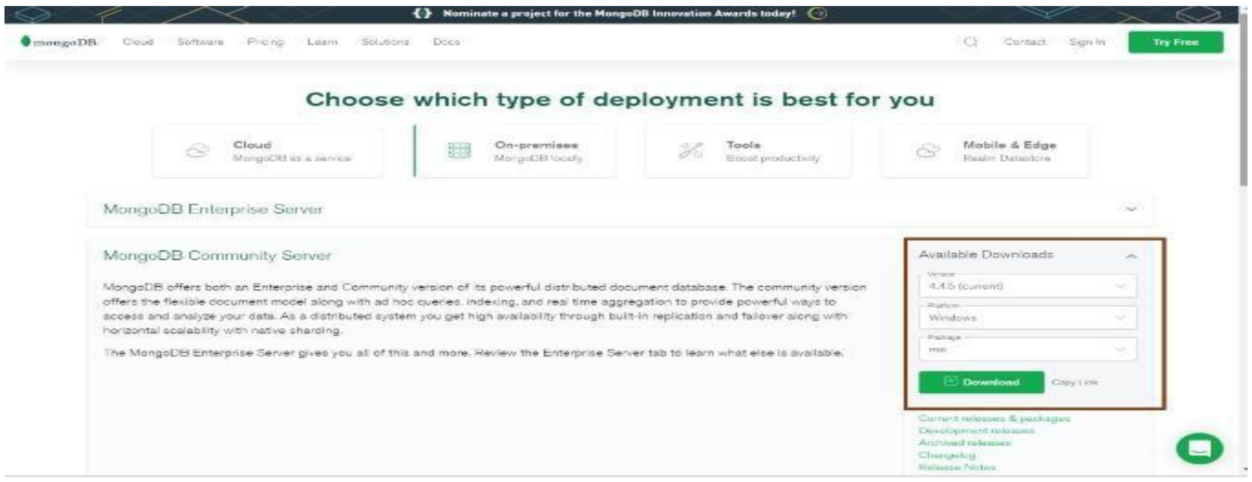
Navigate to the official MongoDB website https://www.mongodb.com/

Cross-check the Specifications and Download MongoDB

Under the Software section, click on the Community server version.


Nominate a project for the MongoDB Innovation Awards today!
mongoDB
Cloud
Software
Pricing
Learn
Solutions
Docs
Contact
Sign In
Try Free
Community Server
A free and open document database
Enterprise Server
Advanced features and security
Developer Tools
Connect, configure and work with MongoDB
Compass
GUI for MongoDB
Ops Manager
On-prem management platform for MongoDB
Connectors
Easy integrations to your data estate
tabase for
applications
rpose, document-based, distributed
plication developers and for the cloud
era.
kes you more productive.
goDB free in the cloud!
Start free

At the time of writing, the latest version is 4.4.5. Ensure that the platform is Windows, and the packageis MSI. Go ahead and click on download.



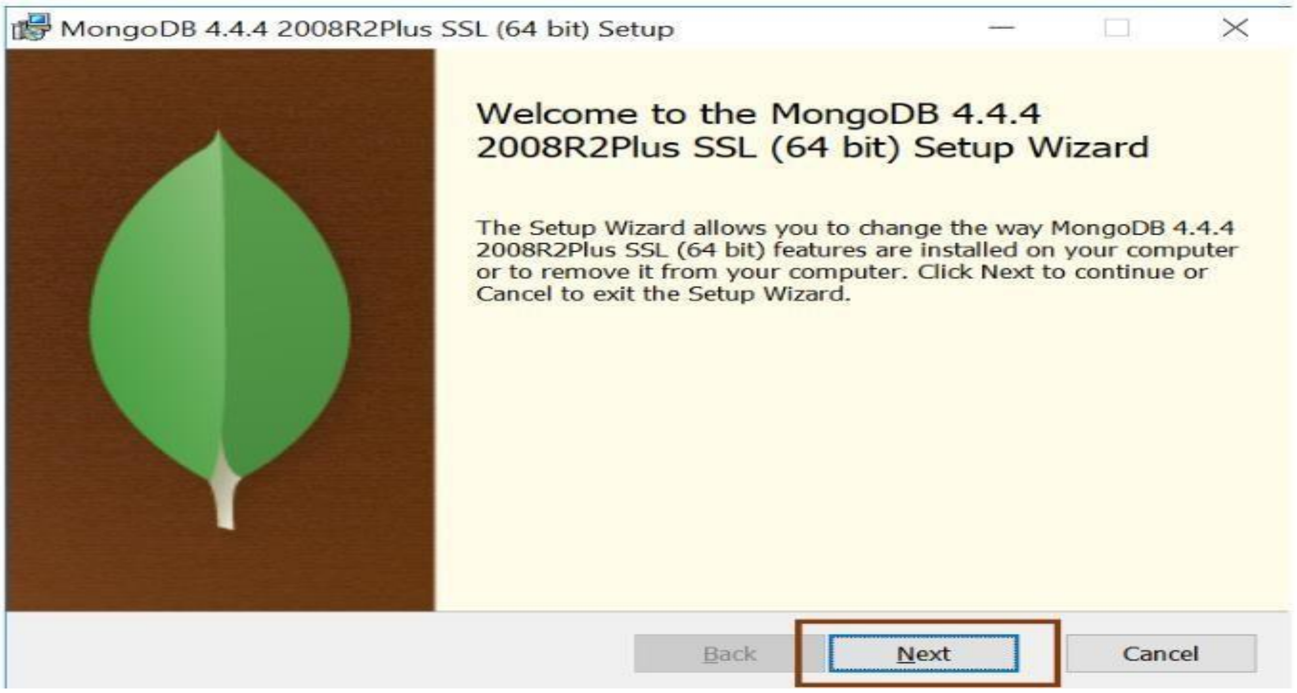
MongoDB offers both an Enterprise and Community version of its powerful distributed document database. The community version offers the flexible document model along with ad hoc queries, indexing, and real time aggregation to provide powerful ways to access and analyze your data. As a distributed system you get high availability through built-in replication and failover along with horizontal scalability with native sharding.

The MongoDB Enterprise Server gives you all of this and more. Review the Enterprise Server tab to learn what else is available.

Version: 4.4.5 (current)
Platform: Windows
Package: msi

Mongo DB Installation:

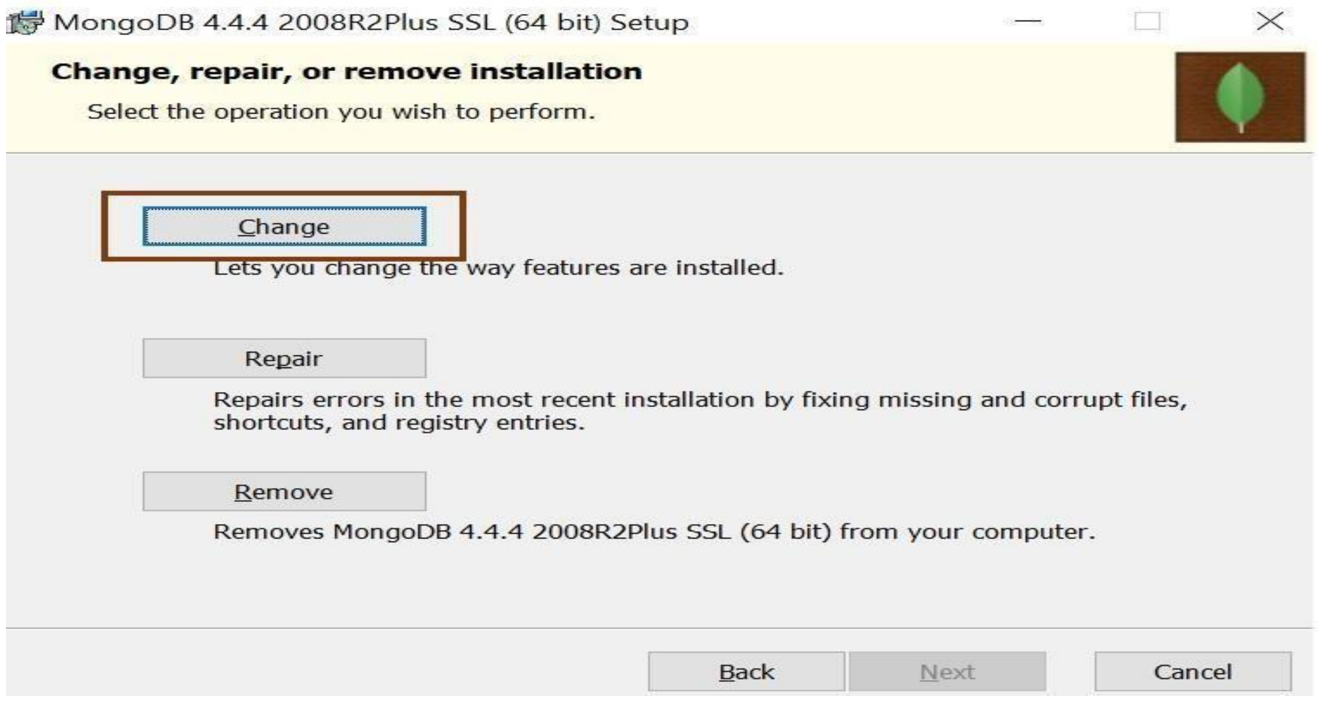
You can find the downloaded file in the downloads directory. You can follow the stepsmentioned there and install the software.



MongoDB 4.4.4 2008R2Plus SSL (64 bit) Setup

Welcome to the MongoDB 4.4.4 2008R2Plus SSL (64 bit) Setup Wizard

The Setup Wizard allows you to change the way MongoDB 4.4.4 2008R2Plus SSL (64 bit) features are installed on your computer or to remove it from your computer. Click Next to continue or Cancel to exit the Setup Wizard.



MongoDB 4.4.4 2008R2Plus SSL (64 bit) Setup

Change, repair, or remove installation

Select the operation you wish to perform.

Change

Lets you change the way features are installed.

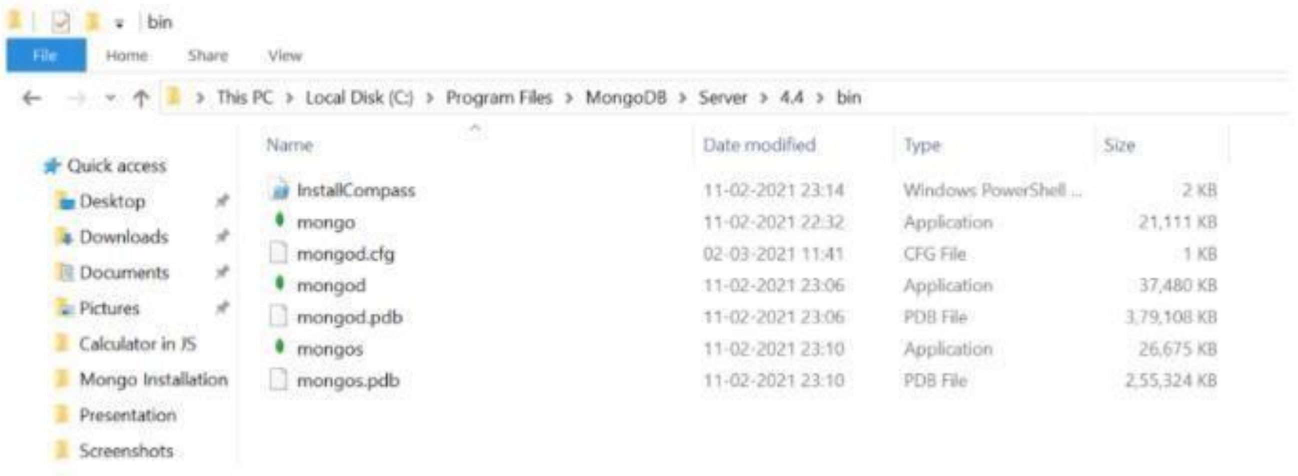
Repair

Repairs errors in the most recent installation by fixing missing and corrupt files, shortcuts, and registry entries.

Remove

Removes MongoDB 4.4.4 2008R2Plus SSL (64 bit) from your computer.

On completing the installation successfully, you will find the software package in your Cdrive. C:\Program Files\MongoDB\Server\4.4\bin.

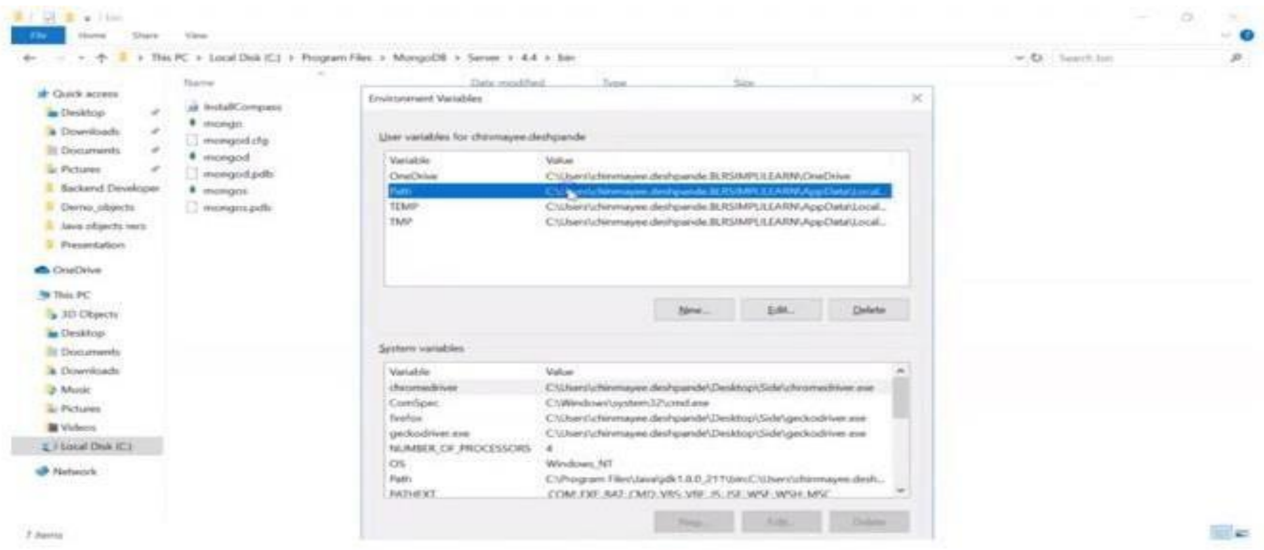


Name	Date modified	Type	Size
InstallCompass	11-02-2021 23:14	Windows PowerShell ...	2 KB
mongo	11-02-2021 22:32	Application	21,111 KB
mongod.cfg	02-03-2021 11:41	CFG File	1 KB
mongod	11-02-2021 23:06	Application	37,480 KB
mongod.pdb	11-02-2021 23:06	PDB File	3,79,108 KB
mongos	11-02-2021 23:10	Application	26,675 KB
mongos.pdb	11-02-2021 23:10	PDB File	2,55,324 KB

You can see that there are mongo and mongod executable files. The mongod file is the daemonprocess that does the background jobs like accessing, retrieving, and updating the database.

Create an Environment Variable:

It's best practice to create an environment variable for the executable file so that youdon't have to change the directory structure every time you want to execute the file.

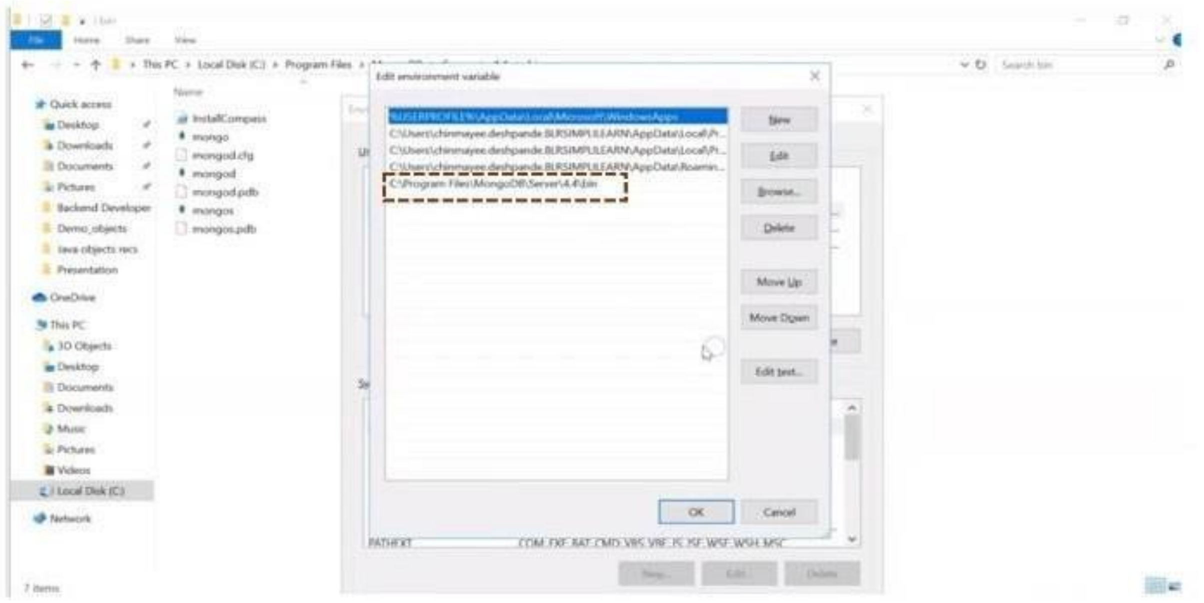


User variables for chinmayee.deshpande

Variable	Value
OneDrive	C:\Users\chinmayee.deshpande.BLRSIMPLILEARN\OneDrive
Path	C:\Users\chinmayee.deshpande.BLRSIMPLILEARN\AppData\Local...
TEMP	C:\Users\chinmayee.deshpande.BLRSIMPLILEARN\AppData\Local...
TMP	C:\Users\chinmayee.deshpande.BLRSIMPLILEARN\AppData\Local...

System variables

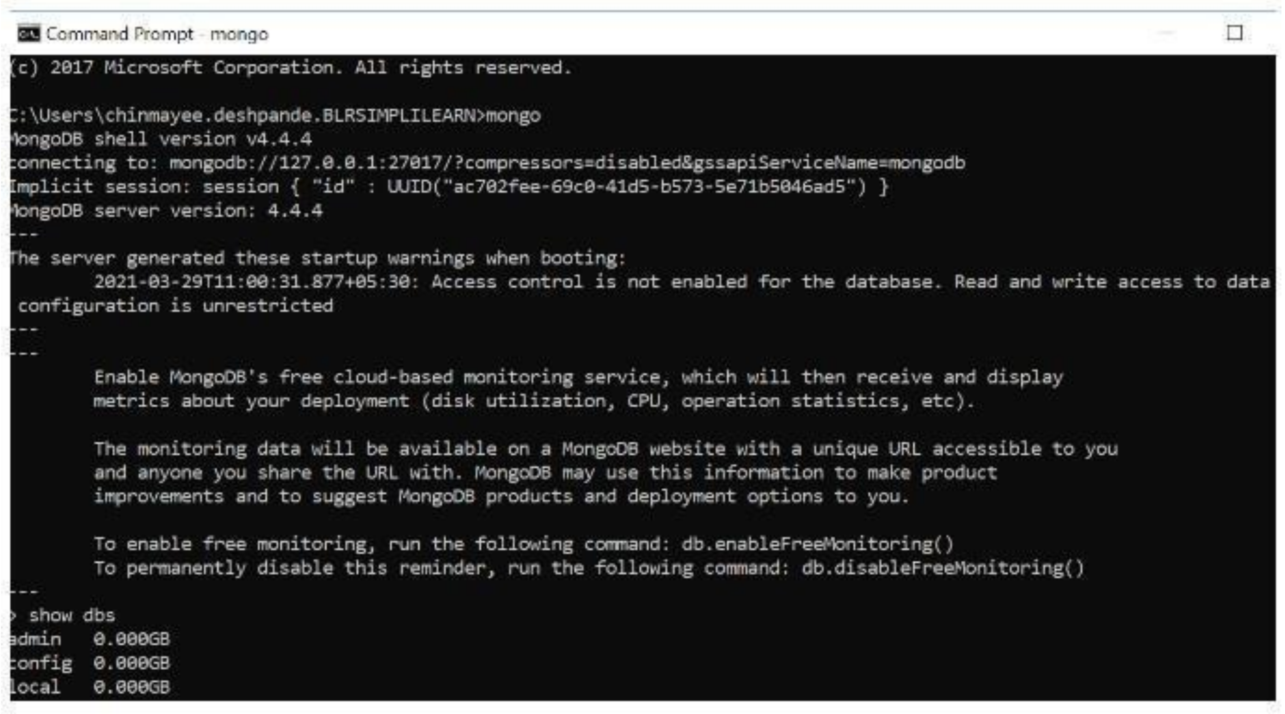
Variable	Value
chromedriver	C:\Users\chinmayee.deshpande\Desktop\Side\chromedriver.exe
ComSpec	C:\Windows\system32\cmd.exe
firefox	C:\Users\chinmayee.deshpande\Desktop\Side\geckodriver.exe
geckodriver.exe	C:\Users\chinmayee.deshpande\Desktop\Side\geckodriver.exe
NUMBER_OF_PROCESSORS	4
OS	Windows_NT
Path	C:\Program Files\Java\jdk1.8.0_211\bin;C:\Users\chinmayee.desh...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC



```
%USERPROFILE%\AppData\Local\Microsoft\WindowsApps
C:\Users\chinmayee.deshpande.BLRSIMPLILEARN\AppData\Local\Pr...
C:\Users\chinmayee.deshpande.BLRSIMPLILEARN\AppData\Local\Pr...
C:\Users\chinmayee.deshpande.BLRSIMPLILEARN\AppData\Roamin...
C:\Program Files\MongoDB\Server\4.4\bin
```

Execute the Mongo App:

After creating an environment path, you can open the command prompt and justtype in mongo and press enter.

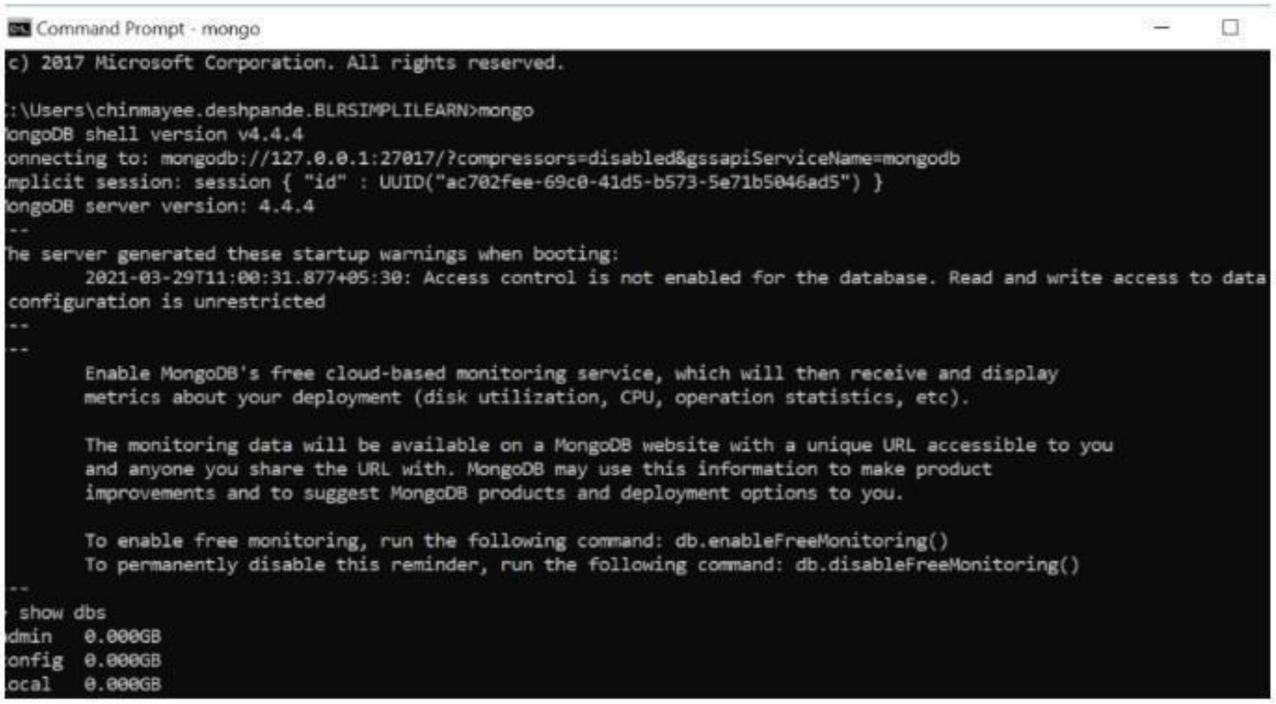


```
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\chinmayee.deshpande.BLRSIMPLILEARN>mongo
MongoDB shell version v4.4.4
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("ac702fee-69c0-41d5-b573-5e71b5046ad5") }
MongoDB server version: 4.4.4
---
The server generated these startup warnings when booting:
        2021-03-29T11:00:31.877+05:30: Access control is not enabled for the database. Read and write access to data
 configuration is unrestricted
---
---
        Enable MongoDB's free cloud-based monitoring service, which will then receive and display
        metrics about your deployment (disk utilization, CPU, operation statistics, etc).

        The monitoring data will be available on a MongoDB website with a unique URL accessible to you
        and anyone you share the URL with. MongoDB may use this information to make product
        improvements and to suggest MongoDB products and deployment options to you.

        To enable free monitoring, run the following command: db.enableFreeMonitoring()
        To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
> show dbs
admin   0.000GB
config  0.000GB
local   0.000GB
```

The mongo server is then generated and is up and running.

Verify the Setup

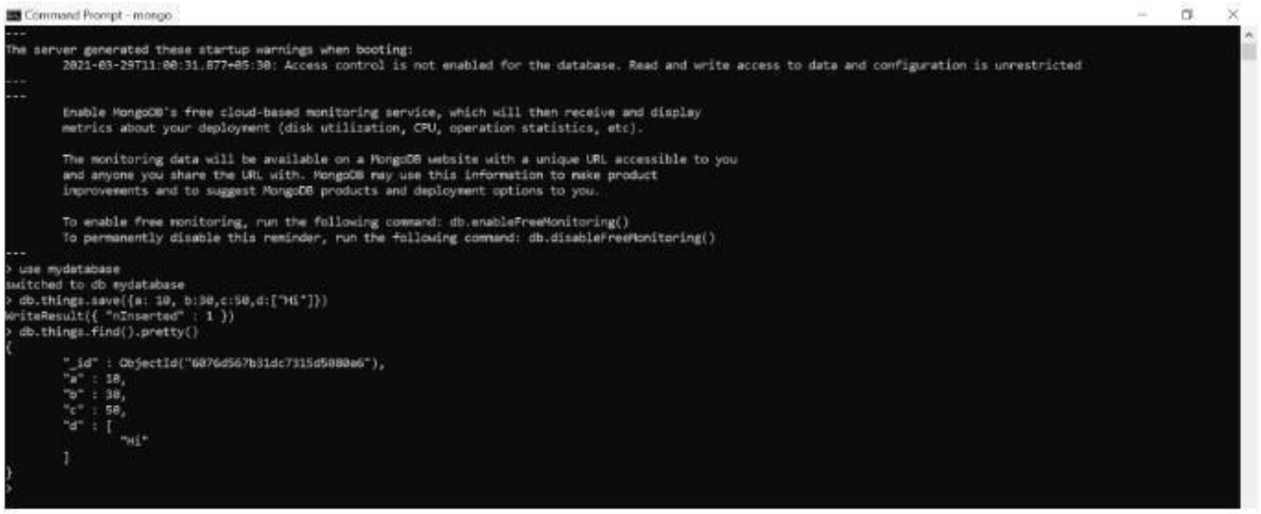
To verify if it did the setup correctly, type in the command show DBS**.**



Command Prompt - mongo

```
c) 2017 Microsoft Corporation. All rights reserved.

:\Users\chinmayee.deshpande.BLRSIMPLILEARN>mongo
ongoDB shell version v4.4.4
onnecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
mplicit session: session { "id" : UUID("ac702fee-69c0-41d5-b573-5e71b5046ad5") }
ongoDB server version: 4.4.4
---
he server generated these startup warnings when booting:
        2021-03-29T11:00:31.877+05:30: Access control is not enabled for the database. Read and write access to data
 configuration is unrestricted
---
---

        Enable MongoDB's free cloud-based monitoring service, which will then receive and display
        metrics about your deployment (disk utilization, CPU, operation statistics, etc).

        The monitoring data will be available on a MongoDB website with a unique URL accessible to you
        and anyone you share the URL with. MongoDB may use this information to make product
        improvements and to suggest MongoDB products and deployment options to you.

        To enable free monitoring, run the following command: db.enableFreeMonitoring()
        To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
 show dbs
dmin   0.000GB
onfig  0.000GB
ocal   0.000GB
```

With that, you have successfully installed and set up MongoDB on your Windows system.



Command Prompt - mongo

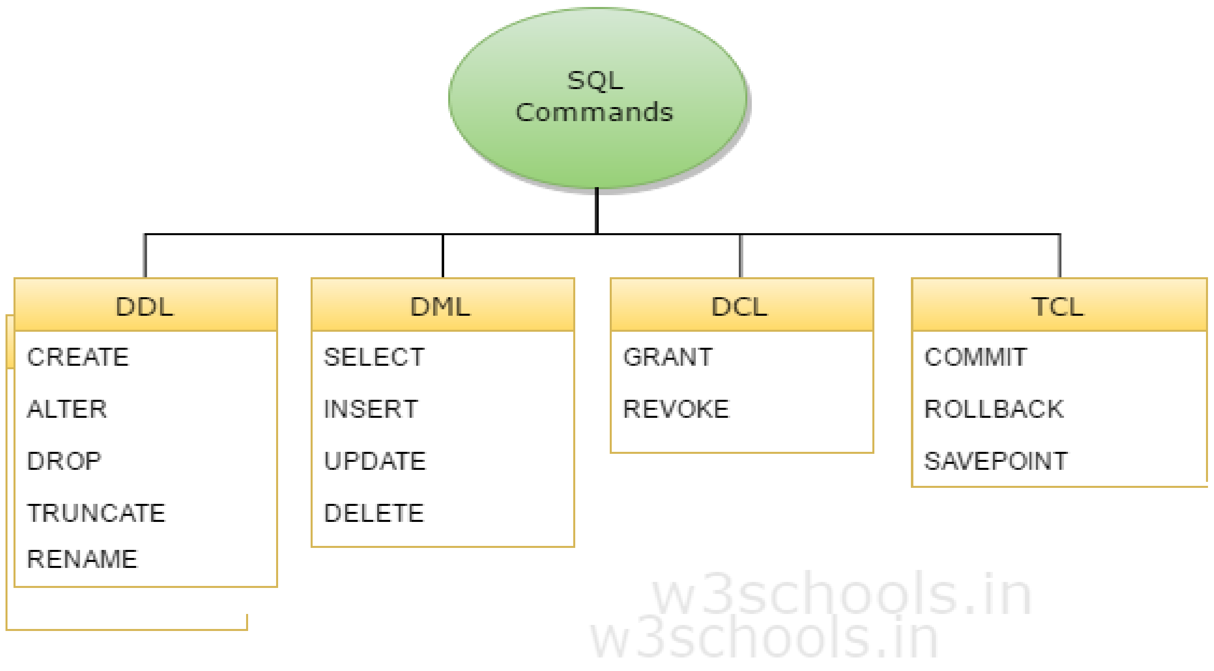
```
---
The server generated these startup warnings when booting:
        2021-03-29T11:00:31.877+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
---
---

        Enable MongoDB's free cloud-based monitoring service, which will then receive and display
        metrics about your deployment (disk utilization, CPU, operation statistics, etc).

        The monitoring data will be available on a MongoDB website with a unique URL accessible to you
        and anyone you share the URL with. MongoDB may use this information to make product
        improvements and to suggest MongoDB products and deployment options to you.

        To enable free monitoring, run the following command: db.enableFreeMonitoring()
        To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
> use mydatabase
switched to db mydatabase
> db.things.save({a: 10, b:30,c:50,d:["Hi"]})
WriteResult({ "nInserted" : 1 })
> db.things.find().pretty()
{
        "_id" : ObjectId("6076d567b31dc7315d5080a6"),
        "a" : 10,
        "b" : 30,
        "c" : 50,
        "d" : [
                "Hi"
        ]
}
>
```

PRACTICING DDL COMMANDS

Data Definition Language

The data definition language is used to create an object, alter the structure of an object and also drop already created object. The Data Definition Languages used for table definition can be classified into following:

- Create table command
- Alter table command
- Truncate table command
- Drop table command



1. CREATION OF TABLES:

CREATE TABLE:

Table is a primary object of database, used to store data in form of rows and columns. It is created using following command:

Syntax: CREATE TABLE tablename (column_name data_ type constraints, …)

CREATING TABLES:

Example:

mysql> create table Bus (Bus_No varchar(5), source varchar(20), destination varchar(20), daysperEXPERIMENT int); Table Created.

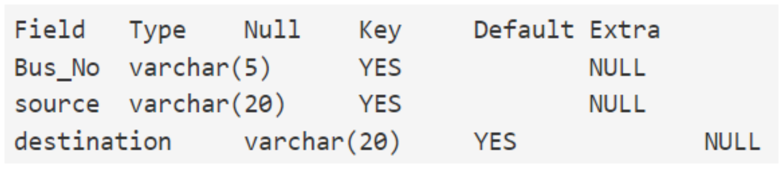
Above definition will create simple table. Still there are more additional option related with create table for the object-relation feature we will discuss it afterwards.

Desc command:

Describe command is external command of Oracle. The describe command is used to view thestructure of table as follows.

Desc <table name>

mysql> desc Bus**;**

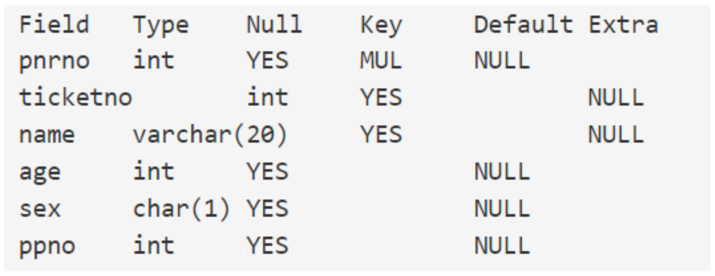


```
Field   Type    Null    Key     Default Extra
Bus_No  varchar(5)      YES             NULL
source  varchar(20)     YES             NULL
destination     varchar(20)     YES             NULL
```

Creating Passenger table:

Mysql>create table passenger(pnrno integer,ticketno integer,name varchar(20),age int,sex char,ppno integer);

Mysql>desc passenger;

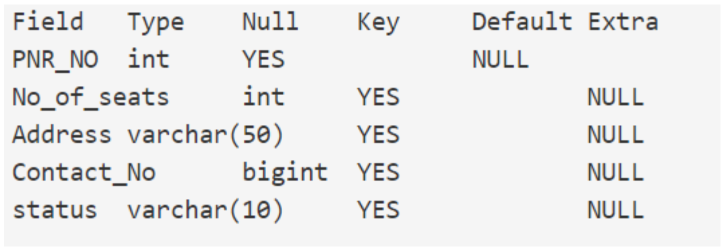


```
Field   Type    Null    Key     Default Extra
pnrno   int     YES     MUL     NULL
ticketno        int     YES             NULL
name    varchar(20)     YES             NULL
age     int     YES             NULL
sex     char(1) YES             NULL
ppno    int     YES             NULL
```

<u>Reservation Table:</u>

mysql > create table Reservation (PNR_NO integer(9), No_of_seats integer(8), Address varchar(50), Contact_No Bigint(12), Status varchar(10));

desc Reservation;

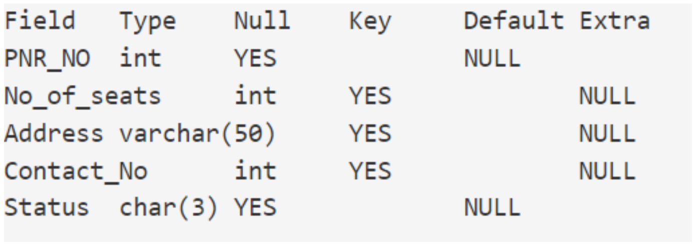


```
Field   Type    Null    Key     Default Extra
PNR_NO  int     YES             NULL
No_of_seats     int     YES             NULL
Address varchar(50)     YES             NULL
Contact_No      bigint  YES             NULL
status  varchar(10)     YES             NULL
```

<u>Cancellation Table:</u>

mysql > create table Cancellation (PNR_NO integer (9), No_of_seats integer (8), Address varchar (50), Contact_No integer (12), Status char (3));
Table created.

SQL> desc Cancellation;

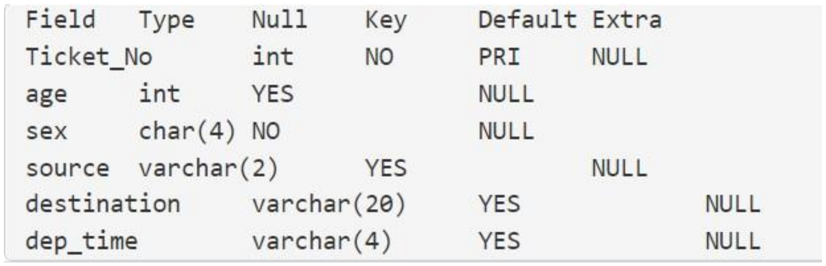


```
Field   Type    Null    Key     Default Extra
PNR_NO  int     YES             NULL
No_of_seats     int     YES             NULL
Address varchar(50)     YES             NULL
Contact_No      int     YES             NULL
Status  char(3) YES             NULL
```

<u>Ticket Table:</u>

mysql >create table Ticket(Ticket_No integer(9) primary key, age int, sex char(4) Not null, source varchar(2), destination varchar(20), dep_time varchar(4));
Table created;

mysql > desc Ticket;



```
Field   Type    Null    Key     Default Extra
Ticket_No       int     NO      PRI     NULL
age     int     YES             NULL
sex     char(4) NO              NULL
source  varchar(2)      YES             NULL
destination     varchar(20)     YES             NULL
dep_time        varchar(4)      YES             NULL
```

ALTER TABLE :

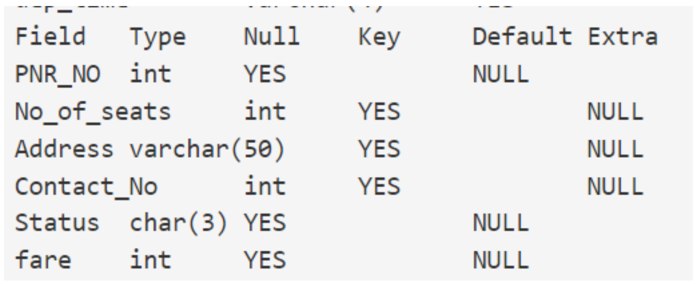
To ADD a column:

SYNTAX: ALTER TABLE <TABLE NAME>ADD (<NEW COLUMN
NAME><DATA TYPE>(<SIZE>), <NEW COLUMNNAME><DATA
TYPE>(<SIZE>) ....);

Example:

mysql > alter table Reservation add column fare integer;

mysql > desc Reservation;



```
Field   Type    Null    Key     Default Extra
PNR_NO  int     YES             NULL
No_of_seats     int     YES             NULL
Address varchar(50)     YES             NULL
Contact_No      int     YES             NULL
Status  char(3) YES             NULL
fare    int     YES             NULL
```

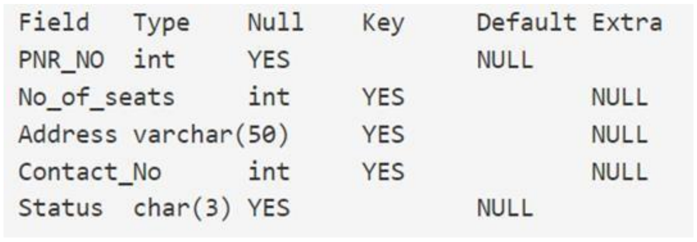
To DROP a column:

SYNTAX: ALTER TABLE <TABLE NAME>DROP COLUMN <COLUMN NAME>;

Example:

mysql >alter table Reservation drop column fare ;

mysql > desc Reservation;



```
Field   Type    Null    Key     Default Extra
PNR_NO  int     YES             NULL
No_of_seats     int     YES             NULL
Address varchar(50)     YES             NULL
Contact_No      int     YES             NULL
Status  char(3) YES             NULL
```

To MODIFY a column:

SYNTAX: ALTER TABLE <TABLE NAME>MODIFY COLUMN <COLUMN NAME>
<NEW DATATYPE>(<NEW SIZE>);

Example:

mysql >alter table Reservation modify column status varchar(10);

mysql >desc Reservation;

```
Field   Type    Null    Key     Default Extra
PNR_NO  int     YES             NULL
No_of_seats     int     YES             NULL
Address varchar(50)     YES             NULL
Contact_No      int     YES             NULL
status  varchar(10)     YES             NULL
```

TO ADD FOREIGN KEY TO THE EXISTING TABLE

mysql > ALTER TABLE passenger ADD FOREIGN KEY (pnrno) REFERENCES Reservation (PNR_NO);
 Table altered.
desc passenger;

```
Field   Type    Null    Key     Default Extra
pnrno   int     YES     MUL     NULL
ticketno        int     YES             NULL
name    varchar(20)     YES             NULL
age     int     YES             NULL
sex     char(1) YES             NULL
ppno    int     YES             NULL
```

mysql > ALTER TABLE Cancellation ADD FOREIGN KEY (PNR_NO) REFERENCES **Reservation (PNR_NO);**

Table altered.

```
Field   Type    Null    Key     Default Extra
PNR_NO  int     YES     MUL     NULL
No_of_seats     int     YES             NULL
Address varchar(50)     YES             NULL
Contact_No      int     YES             NULL
Status  char(3) YES             NULL
```

TRUNCATE TABLE:

If there is no further use of records stored in a table and the structure is required then only data can be deleted using truncate command. Truncate command will delete all the records permanently of specified table as follows.

Truncate table <table name>
EXAMPLE: try your own Query

RENAME A TABLE

Rename command is used to give new names for existing tables.

```
RENAME table old tablename TO new tablename;
```

Example:

```
MYSQL>RENAME table passenger  TO   Passenger;
```

VIVA QUESTIONS

1. Define data and information.
2. Define Data base management system.
3. What is SQL?
4. What is the syntax for creating a table?
5. List the components of SQL.
6. Define DDL? What are the DDL commands?
7. List out the uses of alter command.
8. What is Syntax for truncate a table?
9. What is the use drop table command?

EXPERIMENT- 6

AIM: Applying DML commands on Road Way Travels Tables.

DML COMMANDS

1. INSERTING DATA IN TO TABLE

Insert command is used to insert rows into the table.

SYNTAX:
INSERT INTO tablename values (columnname1, columnname2,….columnname n)

INSERTION of Data can also be done by the following Syntax:

SYNTAX

```
INSERT IN TO tablename (columnname1, columnname2,….columnname n)
VALUES(Value1,Value2,..Value n);              -
```

Inserting values into "Bus" table:

```
mysql > insert into Bus values('w1234','hyderabad', 'tirupathi',4);
mysql >insert into Bus values ('p2345','hyderabad', 'Banglore',3);
mysql >insert into Bus values ('9w01','hyderabad','Kolkata',4);
```

Inserting values into " RESERVATION" table:

```
mysql >insert into Reservation values(1,2,'masabtank',9009897812,'confirm');
mysql>insert into Reservation values(1,2,'masabtank',9009897812,'confirm');
```

Inserting values into “PASSENGER" table:

```
 mysql >insert into Passenger values (1, 1,'SACHIN', 12,'m', 1234);
mysql >insert into Passenger values (2, 2,'rahul', 34,'m', 3456);
mysql >insert into Passenger values(3,3,'swetha',24,'f',8734);
mysql >insert into Passenger values(5,5,'Arun',24,'m',7387);
mysql >insert into Passenger values(6,6,'Aruna',25,'f',7389);
mysql >insert into Passenger values(4,3,'rohith',24,'m',734);
```

UPDATE

This SQL command is used to modify the values in an existing table.

mysql >**UPDATE** tablename

SET column1= expression1, column2= expression 2,...

WHERE somecolumn=somevalue;

An expression consists of either a constant (new value), an arithmetic or string operation or an SQL query. Note that the new value to assign to <column> mustmatching data type.

An update statement used without a where clause results in changing respective attributes of all tuples in the specified table.

EXAMPLE:

```
mysql >update Passenger set age='43' where pnrno='2';
TEST OUTPUT:
```

DELETE

In order to delete rows from a table we use this command

mysql >**DELETE** FROM tablename WHERE condition;

```
EX: delete from Passenger where pnrno='3';
1 row deleted.
```

```
TEST OUTPUT:
SQL> select * from Passenger;
```

TO RETRIEVE / DISPLAY DATA FROM TABLES:

a. Select command is used to select values or data from table

SYNTAX

SELECT * FROM TABLENAME;

Example:

```
SQL> select * from Passenger;
```

TESTOUTPUT:

b. Elimination of duplicates from the select statement

SQL> **SELECT DISTINCT** columnname 1, columnname 2,…. columnname n FROM tablename;

EXAMPLE QUERY:

TEST OUTPUT:

c. The retrieving of specific columns from a table

Mysql > **SELECT** columnname 1, columnname 2,…. columnname n FROM tablename;

Mysql>select name,age,sex from Passenger;

TEST OUTPUT:

Example1:

Display Data From BUS Table

Example2: Display Data From Reservation Table

VIVA QUESTIONS

1. What are the DML commands?
2. How the data or values to be entered into a table?
3. What is the use of DELETE command?
4. How the data or values to be updated on a table?
5. List out the uses of SELECT command?
6. How the data or values are retrieved from a table?
7. Define DML? What are the DML commands?

EXPERIMENT -7

AIM : QUERIES USING ANY, ALL, IN, INTERSECT, UNION

UNION

Union is used to combine the results of two queries into a single result set of all matching rows. Both the queries must result in the same number of columns and compatible data types in order to unite. All duplicate records are removed automatically unless UNION ALL is used.

INTERSECT
It is used to take the result of two queries and returns the only those rows which are common in both result sets. It removes duplicate records from the final result set.

EXCEPT
It is used to take the distinct records of two one query and returns the only those rows which do not appear in the second result set.

EXAMPLES:

Let us create tables for sailors, Reserves and Boats

```
CREATE TABLE sailors ( sid integer, sname varchar(20),rating integer,age integer);

insert into sailors values(22,'dustin',7,45);
insert into sailors values(29,'brutus',1,33);
insert into sailors values(31,'lubber',79,55);
insert into sailors values(32,'andy',8,25);
insert into sailors values(58,'rusty',10,35);
insert into sailors values(58,'buplb',10,35);
insert into sailors values(58,'buplerb',10,35);

CREATE TABLE boats(  bid integer, bname varchar(20),color varchar(20));
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');

CREATE TABLE reserves( sid integer, bid integer, day  date);
insert into reserves values(22,101,'2004-01-01');
insert into reserves values(22,102,'2004-01-01');
insert into reserves values(22,103,'2004-02-01');
insert into reserves values(22,105,'2004-02-01');
insert into reserves values(31,103,'2005-05-05');
insert into reserves values(32,104,'2005-04-07');
```

QUERIES

1. **Find all sailor id's of sailors who have a rating of at least 8 or reserved boat 103.**

```
mysql >(SELECT sid  FROM sailors WHERE rating>=8)
UNION
(SELECT sid  FROM reserves WHERE bid=103);
```

TEST OUTPUT:

2. **Find all sailor id's of sailors who have a rating of at least 8 and reserved boat 103.**

```
mysql >( (SELECT sid  FROM sailors WHERE rating>=8)
intersect
(SELECT sid  FROM reserves WHERE bid=103);
```

TEST OUTPUT:

3. **Find the names of sailors who have reserved boat number 103.**

```
mysql >(select s.sname from sailors s  where s.sid  in (select r.sid from reserves r where r.bid=103);
```

TEST OUTPUT:

4. **Find the names of sailors who have never reserved boat number 103.**

```
mysql >(select s.sname from sailors s where s.sid not in (select r.sid from reserves r where r.bid=103);
```

TEST OUTPUT:

5. **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');
```

TEST OUTPUT:

6. **Find the sailors with the highest rating**

   mysql >(select s.sid from sailors s where s.rating >= all (select s2.rating from sailors s2);

TEST OUTPUT:

QUERIES ON ROADWAY TRAVELS DATABASE

1. Display unique PNR_no of all Passengers..

   . Mysql>select distinct(pnrno) from Passenger;

   TEST OUTPUT:

2. Display all the names of male passengers

   Mysql >select Name from Passenger where Sex='m';

   TEST OUTPUT:

3. Display Ticket numbers and names of all Passengers.

   Mysql>select ticketno,Name from Passenger;

TEST OUTPUT:

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

```
Mysql>select  ticketno from Passenger where Name like'r%h';
```

TEST OUTPUT:

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

```
Mysql>select Name from Passenger where age between 30 and 45;
```

TEST OUTPUT:

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

```
Mysql>select  Name from Passenger where Name like 'A%';
```

TEST OUTPUT:

7. Display the sorted list of passengers names

```
Mysql>select name from Passenger order by Name;
```

TEST OUTPUT:

VIVA QUESTIONS

1) Explain the flow of execution for a Nested query

2) Differentiate between flow of execution in Nested Query and Correlated Query?

3) What happens if we eliminate HAVING clause in a query which is having both GROUP BY and HAVING clauses.

4) What are the different types of Nested Queries?

5) What are the different types of Correlated Queries?

EXPERIMENT 8 & 9

Querying Using Aggregate functions (COUNT, SUM, AVERAGE using GROUPBY and HAVING) Creation and dropping of Views.

Aggregate operators: In addition to simply retrieving data, we often want to perform some computation or summarization. SQL allows the use of arithmetic expressions.

1. COUNT:

SYNTAX:

Select count ([<distinct>/<ALL]<expr>)

2. SUM:

SYNTAX:

Select SUM ([<distinct>/<ALL]<column name>)

3. AVG:

SYNTAX:

Select AVG ([<distinct>/<ALL]<column name>)

4. MINIMUM(MIN):

SYNTAX:

Select MIN ([<distinct>/<ALL]<expr>)

5. _

MAXIMUM(MAX):

SYNTAX:

Select MAX ([<distinct>/<ALL]<expr>)

GROUP BY and HAVING Clause

SYNTAX

Select [DISTINCT] select list FROM from list WHERE qualification
Group by Groupinglist having group-qualification

1. **Write a Query to display the Information present in the Reservation and cancellation tables.**

```
mysql>select * from Reservation
        union
       select * from Cancellation;
```

TEST OUTPUT:

2. **Display the number of days in a EXPERIMENT on which the 9W01 bus is available.**

```
mysql> select daysperEXPERIMENT from Bus where Bus_No='9w01';
```

TEST OUTPUT:

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

```
mysql>select  count(No_of_seats),PNR_NO  from Reservation group by PNR_NO;
```

TEST OUTPUT:

4. **Find the number of tickets booked by a passenger where the number of seats is greater than 1**

```
mysql>select count(No_of_seats) from Reservation group by PNR_NO having
sum(No_of_seats)>1;
```

TEST OUTPUT:

5. **Find the distinct PNR numbers that are present.**

   mysql>select distinct(PNR_NO) from Reservation;

   TEST OUTPUT:

6. **Find the total number of cancelled seats.**

mysql > select sum(No_of_seats) AS Cancelled_seats from Cancellation;

   TEST OUTPUT:

VIEWS

After a table is created and populated with data, it may become necessary to prevent all users from accessing all columns of a table, for data security reasons. This would mean creating several tables having the appropriate number of columns and assigning specific users to each table as required. This will achieve the security requirements but will rise to a great deal of redundant data being resident in tables, in the database. To reduce redundant data to the minimum possible, oracle allows the creation ofan object called a view.

A view is a virtual table or logical representation of another table or combination of tables. A view consists of rows and columns just like a table. The difference between a view and a table is that views are definitions built on top of other tables (or views), and do not hold data themselves. If data is changing in the underlying table, the same change is reflected in the view. A view can be built on top of a single table or multiple tables. It can also be built on top of another view. A view derives its data from the tables on which it is based. These tables are called base tables. Base tables might in turn be actual tables or might be views themselves. All operations performed on a view actually affect the base table of the view. We can use views in almost the same way as tables. Also can query, update, insert into and delete from views, just as in standard tables

AIM : Implement Views:

Syntax: Create View <View_Name> As Select statement;
Example:
SQL>Create View EmpView As Select * from Emp_master where job='clerk';

View created.

Syntax: Select columnname,columnname from <View_Name>;

Example:
SQL>Select Empno,Ename,Salary from EmpView where salary in (10000,20000);

TEST OUTPUT:

UPDATABLE VIEWS:
Syntax for creating an Updatable View:

Create View Emp_vw As
Select Empno,Ename,Deptno from Employee;

View created.

SQL>Insert into Emp_vw values(1126,'Brijesh',20);

SQL>Update Emp_vw set Deptno=30 where

Empno=1125;

1 row updated.

SQL>Delete from Emp_vw where

Empno=1122;

TEST OUTPUT:

mysql >Update EmpDept_Vw set salary=4300 where Empno=1125;

TEST OUTPUT:

mysql >Delete From EmpDept_Vw where Empno=1123;

TEST OUTPUT

DESTROYING A VIEW:

Syntax: Drop View <View_Name>;
Example:
mysql >Drop View Emp_Vw;

TEST OUTPUT:

VIVA QUESTIONS:

1. Define view.
2. What is the need of a view?
3. List out the advantages of views.
4. What is the syntax for creating a view?
5. How can you insert data into a view?
6. How can you update data into from a view?
7. What is the syntax for deleting a view?
8. List out the criteria for updatable views.
9. What is the syntax for renaming the columns of a view

EXPERIMENT 10

Triggers

In MySQL, a trigger is a set of SQL statements that is invoked automatically when a change is made tothe data on the associated table. A trigger can be defined to be invoked either before or after the data ischanged by INSERT, UPDATE or DELETE statement.

- BEFORE INSERT – activated before data is inserted into the table.
- AFTER INSERT – activated after data is inserted into the table.
- BEFORE UPDATE – activated before data in the table is updated.
- AFTER UPDAT – activated after data in the table is updated.
- BEFORE DELETE – activated before data is removed from the table.
- AFTER DELETE – activated after data is removed from the table.

A database trigger is procedural code that is automatically executed in response to certain events on a particular table or view in a database. The trigger is mostly used for maintaining the integrity of the information on the database.

The events that fire a trigger include the following:
1) DML statements that modify data in a table (INSERT , UPDATE , or DELETE)
2) DDL statements.
3) System events such as startup, shutdown, and error messages.
4) User events such as logon and logoff. Note: Oracle Forms can define, store, and run triggersofa different sort.

To View list of triggers;

Show triggers;

To remove a trigger for Database

drop trigger trigger_name;

ex: drop trigger ins_sal;

<u>Types of Triggers:-</u>

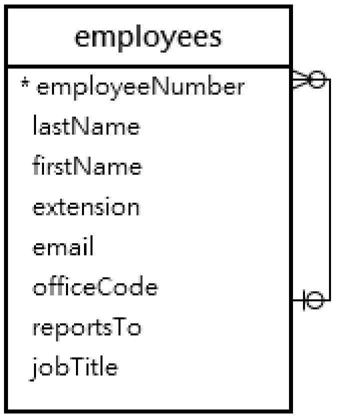
1. **Row Triggers** :-A row trigger is fired each time the table is affected by the triggering statement. For example, if an UPDATE statement updates multiple rows of a table, a row trigger is fired once for each row affected by the UPDATE statement. If a triggering statement affects no rows, a row trigger is not executed at all.

Row triggers are useful if the code in the trigger action depends on data provided by the triggering statement or rows that are affected. For example, Figure 15 - 3 illustrates a row trigger thatuses the values of each row affected by the triggering statement.

2. **Statement Triggers** :A statement trigger is fired once on behalf of the triggering statement, regardless of the number of rows in the table that the triggering statement affects (even if no rows are affected). For example, if a DELETE statement deletes several rows from a table, a statement-level DELETE trigger is fired only once, regardless of how many rows are deleted from the table.

Statement triggers are useful if the code in the trigger action does not depend on the data provided by the triggering statement or the rows affected. For example, if a trigger makes a complex security check on the current time or user, or if a trigger generates a single audit record based on the type of triggering statement, a statement trigger is used.

When defining a trigger, specify the trigger timing. That is, specify whether the trigger action isto be executed before or after the triggering statement. BEFORE and AFTER apply to both statement and row triggers



Example:

```
CREATE TRIGGER trigger_name      trigger_time      trigger_event
ON table_name
FOR EACH ROW
BEGIN
END;

trigger_time=before/after
trigger_event=insert/delete/update
```

Example:

```
CREATE TRIGGER sal_sum after insert ON emp
FOR EACH ROW SET @sal = @sal + NEW.sal;
```

FIRING A TRIGGER:

creating a trigger in MySQL to log the changes of the employeestable.

The CREATE TRIGGER statement creates a new trigger. Here is the basic syntax of the CREATE TRIGGER statement:

```
CREATE TRIGGER trigger_name
{BEFORE | AFTER} {INSERT | UPDATE| DELETE }
ON table_name FOR EACH ROW
trigger_body;
```

First, create a new table named employees_audit to keep the changes to the employees table:

```
 CREATE TABLE employees_audit (
id INT AUTO_INCREMENT PRIMARY KEY,
employeeNumber    INT    NOT    NULL,
lastname VARCHAR(50) NOT NULL,
changedat DATETIME DEFAULT NULL,
action VARCHAR(50) DEFAULT NULL
 );
```

create a BEFORE UPDATE trigger that is invoked before a change is made to the employees table.

```
CREATE TRIGGER before_employee_update
   BEFORE UPDATE ON employees
   FOR EACH ROW
INSERT INTO employees_audit
SET action = 'update',
    employeeNumber = OLD.employeeNumber,
    lastname = OLD.lastname,
    changedat = NOW();
```

Inside the body of the trigger, we used the OLD keyword to access values of the columns employeeNumber
and lastname of the row affected by the trigger.

show all triggers in the current database by using the SHOW TRIGGERS statement:

mysql> show triggers;

mysql> **update a row in the employees table:**

```
UPDATE employeesSET
   lastName = 'Phan'WHERE
employeeNumber = 1056;
```

query the employees_audit table to check if the trigger was fired by the UPDATE statement:

```
SELECT * FROM employees_audit;
```

OUTPUT:

MySQL AFTER INSERTtrigger

create a new table called members:

```
DROP TABLE IF EXISTS members;

CREATE TABLE members ( id INT
AUTO_INCREMENT,
name VARCHAR(100) NOT NULL, email
VARCHAR(255),
birthDate DATE, PRIMARY KEY (id)
);
```

create another table called reminders that stores reminder messages to members.

```
DROP TABLE IF EXISTS reminders;

CREATE TABLE reminders ( id INT
AUTO_INCREMENT,
memberId INT,
message VARCHAR(255) NOT NULL,
PRIMARY KEY (id , memberId)
);
```

VIVA QUESTIONS:

1. Define database triggers.
2. List out the uses of database triggers.
3. What are the pars of triggers and it uses?
4. List out the types of trigger.
5. What is the use of row trigger?
6. What is the use of statement trigger?
7. What do you meant by trigger time?
8. Compare before trigger and after trigger.
9. What is the syntax for DROP a trigger?
10. List out the some situations to apply before and after triggers.

EXPERIMENT 11 &12

Procedures: Creation of stored procedures, Execution of procedure and modification of procedures.

PROCEDURES

Procedure (often called a stored procedure) is a collection of pre-compiled SQL statements stored inside the database. It is a subroutine or a subprogram in the regular computing language. A procedure always contains a name, parameter lists, and SQL statements. We can invoke the procedures by using triggers, other procedures and applications such as Java
, Python
, PHP, etc.

It was first introduced in MySQL version 5. Presently, it can be supported by almost all relational database systems.

Creating a procedure:

The following syntax is used for creating a stored procedure in MySQL. It can return one or more value through parameters or sometimes may not return at all. By default, a procedure is associated with our current database. But we can also create it into another database from the current database by specifying the name as **database_name.procedure_name**.

```
DELIMITER &&
CREATE PROCEDURE procedure_name [[IN | OUT | INOUT] parameter_name datatype [, parameter
datatype]) ]
BEGIN
   Declaration_section
   Executable_section
END &&
DELIMITER ;
```

Parameter Explanations

The procedure syntax has the following parameters:

Parameter Name	Descriptions
procedure_name	It represents the name of the stored procedure.
parameter	It represents the number of parameters. It can be one or more than one.
Declaration_section	It represents the declarations of all variables.

Executable_section	It represents the code for the function execution.

MySQL procedure parameter has one of three modes:

IN parameter

It is the default mode. It takes a parameter as input, such as an attribute. When we define it, the calling program has to pass an argument to the stored procedure. This parameter's value is always protected.

OUT parameters

It is used to pass a parameter as output. Its value can be changed inside the stored procedure, and the changed (new) value is passed back to the calling program. It is noted that a procedure cannot access the OUT parameter's initial value when it starts.

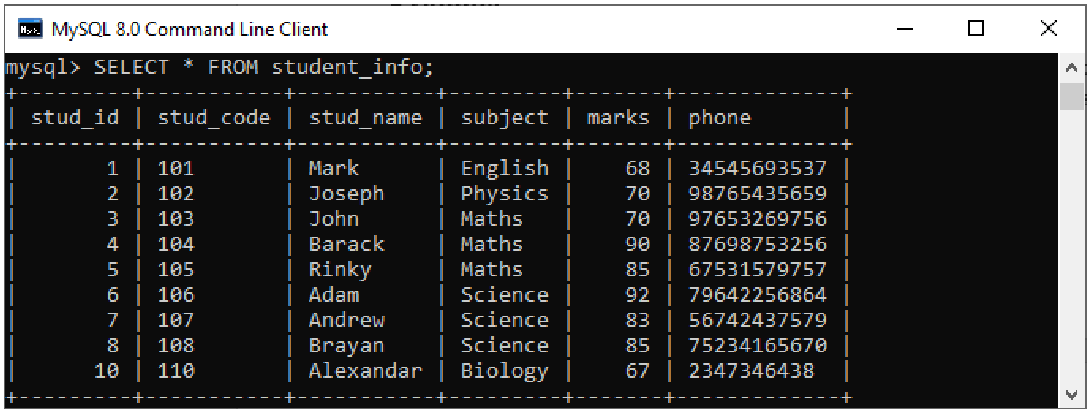
INOUT parameters

It is a combination of IN and OUT parameters. It means the calling program can pass the argument, and the procedure can modify the INOUT parameter, and then passes the new value back to the calling program.

How to call a stored procedure?

CALL procedure_name (parameter(s))

Suppose this database has a table named **student_info** that contains the following data:



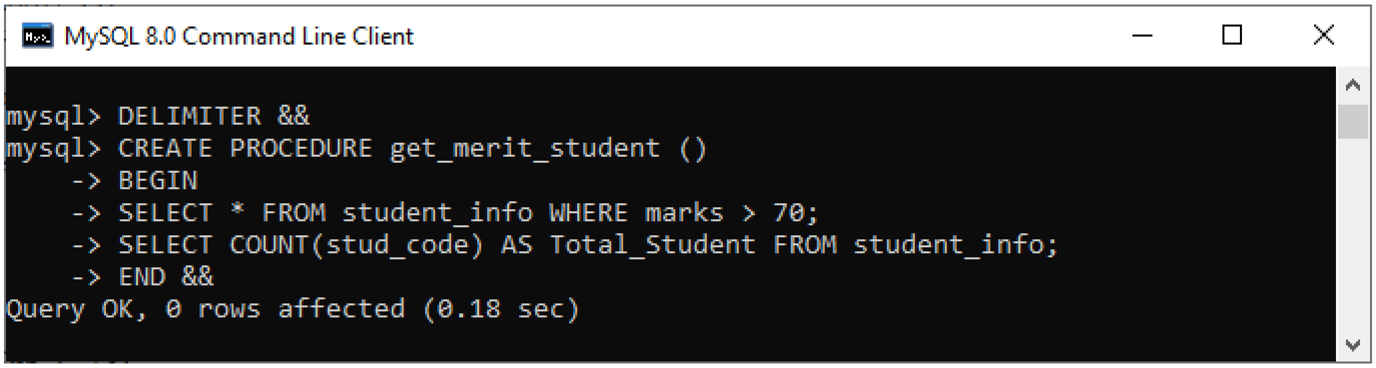
```
mysql> SELECT * FROM student_info;
+---------+-----------+-----------+---------+-------+-------------+
| stud_id | stud_code | stud_name | subject | marks | phone       |
+---------+-----------+-----------+---------+-------+-------------+
|       1 | 101       | Mark      | English |    68 | 34545693537 |
|       2 | 102       | Joseph    | Physics |    70 | 98765435659 |
|       3 | 103       | John      | Maths   |    70 | 97653269756 |
|       4 | 104       | Barack    | Maths   |    90 | 87698753256 |
|       5 | 105       | Rinky     | Maths   |    85 | 67531579757 |
|       6 | 106       | Adam      | Science |    92 | 79642256864 |
|       7 | 107       | Andrew    | Science |    83 | 56742437579 |
|       8 | 108       | Brayan    | Science |    85 | 75234165670 |
|      10 | 110       | Alexandar | Biology |    67 | 2347346438  |
+---------+-----------+-----------+---------+-------+-------------+
```

Procedure without Parameter
Suppose we want to display all records of this table whose marks are greater than 70 and count all the

table rows. The following code creates a procedure named get_merit_students**:**

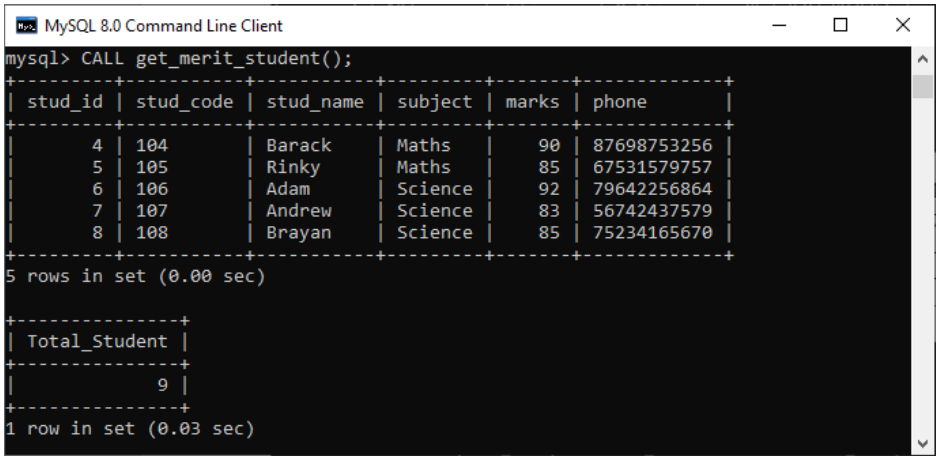
```
DELIMITER &&
CREATE PROCEDURE get_merit_student ()
BEGIN
    SELECT * FROM student_info WHERE marks > 70;
    SELECT COUNT(stud_code) AS Total_Student FROM student_info;
END &&
DELIMITER ;
```

If this code executed successfully, we would get the below output:



```
mysql> DELIMITER &&
mysql> CREATE PROCEDURE get_merit_student ()
    -> BEGIN
    -> SELECT * FROM student_info WHERE marks > 70;
    -> SELECT COUNT(stud_code) AS Total_Student FROM student_info;
    -> END &&
Query OK, 0 rows affected (0.18 sec)
```

mysql> CALL get_merit_student();



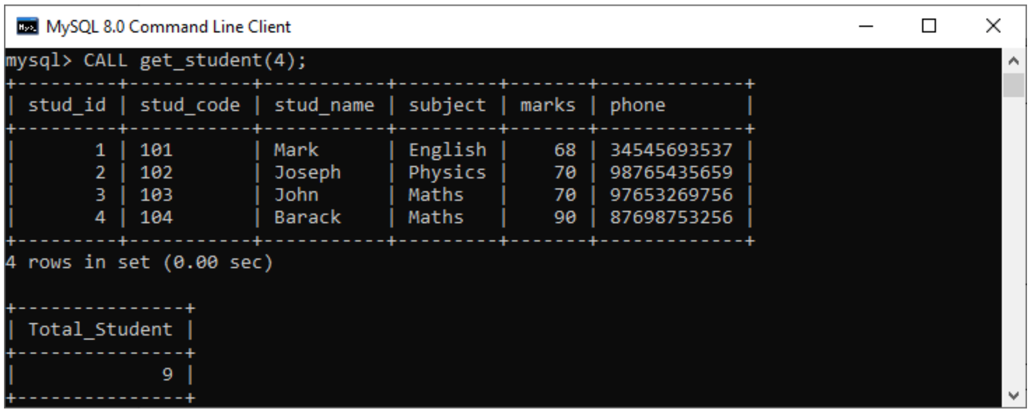
```
mysql> CALL get_merit_student();
+---------+-----------+-----------+---------+-------+-------------+
| stud_id | stud_code | stud_name | subject | marks | phone       |
+---------+-----------+-----------+---------+-------+-------------+
|       4 | 104       | Barack    | Maths   |    90 | 87698753256 |
|       5 | 105       | Rinky     | Maths   |    85 | 67531579757 |
|       6 | 106       | Adam      | Science |    92 | 79642256864 |
|       7 | 107       | Andrew    | Science |    83 | 56742437579 |
|       8 | 108       | Brayan    | Science |    85 | 75234165670 |
+---------+-----------+-----------+---------+-------+-------------+
5 rows in set (0.00 sec)

+---------------+
| Total_Student |
+---------------+
|             9 |
+---------------+
1 row in set (0.03 sec)
```

Procedures with IN Parameter

In this procedure, we have used the IN parameter as 'var1' of integer type to accept a number from users. Its body part fetches the records from the table using a SELECT statement
and returns only those rows that will be supplied by the user. It also returns the total number of rows of the specified table. See the procedure code**:**

```
DELIMITER &&
CREATE PROCEDURE get_student (IN var1 INT)
BEGIN
    SELECT * FROM student_info LIMIT var1;
    SELECT COUNT(stud_code) AS Total_Student FROM student_info;
END &&
DELIMITER ;
```

mysql> CALL get_student(4);

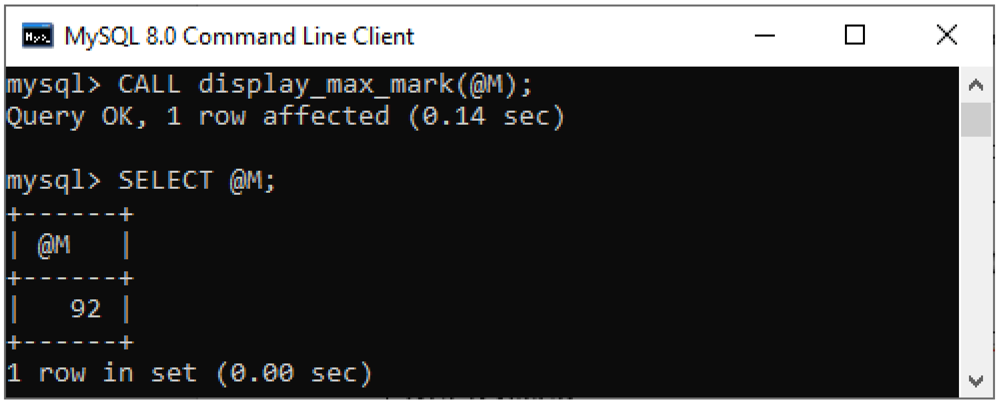


```
mysql> CALL get_student(4);
+---------+-----------+-----------+---------+-------+-------------+
| stud_id | stud_code | stud_name | subject | marks | phone       |
+---------+-----------+-----------+---------+-------+-------------+
|       1 | 101       | Mark      | English |    68 | 34545693537 |
|       2 | 102       | Joseph    | Physics |    70 | 98765435659 |
|       3 | 103       | John      | Maths   |    70 | 97653269756 |
|       4 | 104       | Barack    | Maths   |    90 | 87698753256 |
+---------+-----------+-----------+---------+-------+-------------+
4 rows in set (0.00 sec)

+---------------+
| Total_Student |
+---------------+
|             9 |
+---------------+
```

Procedures with OUT Parameter:
In this procedure, we have used the OUT parameter as the **'highestmark'** of integer type. Its body part fetches the maximum marks from the table using a **MAX() function.** See the procedure code

```
DELIMITER &&
CREATE PROCEDURE display_max_mark (OUT highestmark INT)
BEGIN
    SELECT MAX(marks) INTO highestmark FROM student_info;
END &&
DELIMITER ;
```

mysql> CALL display_max_mark(@M);
mysql> SELECT @M;



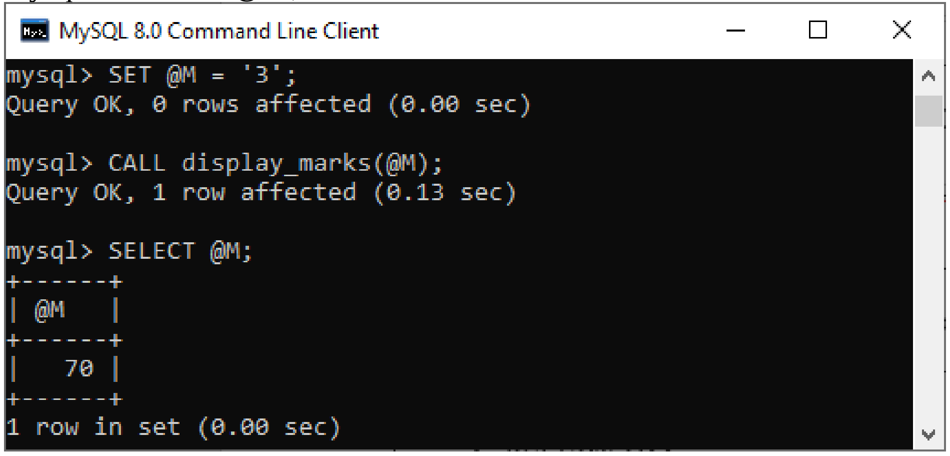
```
mysql> CALL display_max_mark(@M);
Query OK, 1 row affected (0.14 sec)

mysql> SELECT @M;
+------+
| @M   |
+------+
|   92 |
+------+
1 row in set (0.00 sec)
```

Procedures with INOUT Parameter:

In this procedure, we have used the INOUT parameter as 'var1' of integer type. Its body part first fetches the marks from the table with the specified id and then stores it into the same variable var1. The var1 first acts as the IN parameter and then OUT parameter. Therefore, we can call it the INOUT parameter mode. See the procedure code:

```
DELIMITER &&
CREATE PROCEDURE display_marks (INOUT var1 INT)
BEGIN
   SELECT marks INTO var1 FROM student_info WHERE stud_id = var1;
END &&
DELIMITER ;
```

```
mysql> SET @M = '3';
mysql> CALL display_marks(@M);
mysql> SELECT @M;
```



```
mysql> SET @M = '3';
Query OK, 0 rows affected (0.00 sec)

mysql> CALL display_marks(@M);
Query OK, 1 row affected (0.13 sec)

mysql> SELECT @M;
+------+
| @M   |
+------+
|   70 |
+------+
1 row in set (0.00 sec)
```

CREATING PROCEDURE ON FOR ROADWAY TRAVELS

```
CREATE PROCEDURE  my Proc() BEGIN

SELECT COUNT (Tickets) FROM Ticket WHERE age>=40;End;

Procedures created
```

;

VIVA QUESTIONS:

1. What is Stored Procedure?
2. What is difference between Function and Stored Procedure?
3. What are the various types of parameters in procedures?

EXPERIMENT 13

DCL (DATA CONTROL LANGUAGE): Data Control Language statements are used to create roles, permissions, and referential integrity as well it is used to control access to database by securing it. DCL Commands are Grant and Revoke

GRANT - gives user's access privileges to database

REVOKE - withdraw access privileges given with the GRANT command

Checking of User Privileges, Grants etc

mysql> create user mrcet_cse;

Query OK, 0 rows affected (0.30 sec)

*To Check where is the created user i.e location in our database

mysql> select user();

TEST OUTPUT

To check what are the grants that the location is having/

mysql> show grants;

TEST OUTPUT

***To Check what are the GRANTS having for created user**

mysql> show grants for mrcet_IT;

TEST OUTPUT

mysql> show tables;

TEST OUTPUT

***To Flush (RE-FRESH) the privileges**

mysql> flush privileges;

Query OK, 0 rows affected (0.08 sec)

* **Explanation:** *To check where is the user i.e in case if we created user (Ex: mrcet_it) it will be displayed as “%”. Root user is by default so it will be available in “Localhost”*

```
mysql> select host, user from mysql.user;
```

TEST OUTPUT

VIVA QUESTIONS

1. What are DCL commands?
2. List out the uses of various DCL commands?
3. What are the different types of Commands in SQL.
4. What is the difference between TCL & DCLcommands.
5. Who has the privilegeto access the DCLcommands.

CASE STUDY 1

Consider the following relations containing airline flight information:

Flights(flno: integer, from: string, to: string,

distance: integer, departs: time, arrives: time)

Aircraft(aid: integer, aname: string, cruisingrange: integer)

Certified(eid: integer, aid: integer)

Employees(eid: integer, ename: string, salary: integer)

Note that the Employees relation describes pilots and other kinds of employees as well; every pilot is certified for some aircraft (otherwise, he or she would not qualify as a pilot), and only pilots are certified to fly.

Write the following queries in relational algebra, tuple relational calculus, and domain relational

calculus.

1. Find the eids of pilots certified for some Boeing aircraft.
2. Find the names of pilots certified for some Boeing aircraft.
3. Find the aids of all aircraft that can be used on non-stop flights from Bonn to Madras.
4. Identify the flights that can be piloted by every pilot whose salary is more than $100,000.

(Hint: The pilot must be certified for at least one plane with a sufficiently large cruising

range.)

5. Find the names of pilots who can operate some plane with a range greater than 3,000

miles but are not certified on any Boeing aircraft.

6. Find the eids of employees who make the highest salary.
7. Find the eids of employees who make the second highest salary.
8. Find the eids of pilots who are certified for the largest number of aircraft.
9. Find the eids of employees who are certified for exactly three aircraft.
10. Find the total amount paid to employees as salaries.

CASESTUDY 2

Normalization Exercise

HEALTH HISTORY REPORT

PET ID	PET NAME	PET TYPE	PETAGE	OWNER	VISIT DATE	PROCEDURE
246	ROVER	DOG	12	SAM COOK	JAN 13/2002	01 - RABIES VACCINATION
					MAR 27/2002	10 - EXAMINE and TREAT WOUND
					APR 02/2002	05 - HEART WORM TEST
298	SPOT	DOG	2	TERRY KIM	JAN 21/2002	08 - TETANUS VACCINATION
					MAR 10/2002	05 - HEART WORM TEST
341	MORRIS	CAT	4	SAM COOK	JAN 23/2001	01 - RABIESVACCINATION
					JAN 13/2002	01 - RABIESVACCINATION
519	TWEEDY	BIRD	2	TERRY KIM	APR 30/2002	20 - ANNUAL CHECK UP
					APR 30/2002	12 - EYE WASH

UNF:

Pet [pet_id, pet_name, pet_type, pet_age, owner, (visitdate, procedure_no,procedure_name)]

1NF:

Pet [pet_id, pet_name, pet_type, pet_age, owner]
Pet_Visit [pet_id, visitdate, procedure_no, procedure_name]

2NF:

Pet [pet_id, pet_name, pet_type, pet_age, owner] Pet_Visit [
pet_id, visitdate, procedure_no] Procedure [
procedure_no, procedure_name]

3NF:

same as 2NF