A.V.PAREKHTECHNICALINSTITUTE,RAJKOT



COMPUTERENGINEERINGDEPARTMENT**PROFESSIONALPRACTICESUSINGDATABASE(3360702)SEMESTER–6**

|  |  |
| --- | --- |
| **Student Name**  **Enrolment No. Term Date:** | **:VAGHELA KAUSHAL R.** |
| **:186020307112** |
| **:\_10\_\_/\_\_12\_/\_2020\_To\_\_01\_/\_05\_\_/\_\_2021\_\_** |



1. ***V.ParekhTechnicalInstitute,Rajkot***



Certificate

This is to certify that Mr./~~Ms~~.  **VAGHELA KAUSHAL R.**

Enrollment no. **186020307112** Branch **COMPUTER ENGINEERING** Semester 6th has satisfactory completed the course in the subject **PROFESSIONAL PRACTICES USING DATABASE (3360702)**  within the premises of college.

Date of Submission:  **29-04-2021**

**INDEX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SR.NO** | **PRACTICAL** | **CO** | **DATE** | **SIGN** |
| 1 | Install and configure MySQL database | C312.1 | 17/12/2020 |  |
| 2 | Install and use of SQLyog | C312.1 | 24/12/2020 |  |
| 3 | Install and use of php myadmin | C312.1 | 31/12/2020 |  |
| 4 | Install and Use of MySQL Browser | C312.1 | 06/01/2020 |  |
| 5 | Install and use of MySQL Administration | C312.1 | 13/01/2020 |  |
| 6 | Create table and perform various task such as Create a product and product \_price \_history table. The price of product change constantly. Write a trigger for updating product \_ price \_ history table when product price change in product table and such other database can be explored | C312.2 | 16/01/2020 |  |
| 7 | Implement and manipulate trigger such as Create a trigger for deleting all the products of particular product type when that product type is deleted and similar for other databases. | C312.2 | 20/01/2020 |  |
| 8 | Write stored routines such as write a routine for counting all product types and other such routines can be performed | C312.2 | 24/02/2020 |  |
| 9 | Manipulate on routines such as write a routine for updating price of all product by 5% and other such routines can be performed | C312.2 | 06/03/2020 |  |
| 10 | Create cursors such as create a cursor for selecting all product whose price is more than 1000 and other such cursors can be implemented | C312.3 | 17/03/2020 |  |
| 11 | Perform various event handling operations such as create an event that checks the product types having quantity less than 20 in stack at every ten minutes and such other procedure can be done | C312.3 | 20/03/2020 |  |
| 12 | Implement precise events such as create an event that checks the product which has been sold maximum in a day and same exercises can be performed | C312.3 | 24/03/2020 |  |
| 13 | Create and manage user accounts in MySQL | C312.4 | 07/04/2020 |  |
| 14 | Practice with GRANT and Revoke Command | C312.4 | 17/04/2020 |  |
| 15 | Practice with data base backup and recovery operations as well security operations | C312.5 | 24/04/2020 |  |

PROFESSIONAL PRACTICES USING DATABASE (3360702) PRACTICAL 1

**PRACTICAL - 1**

**AIM:** Install and Configure MySQL database

**Basic information:**

MySQL is open-source, cross-platform relational database management server developed by Swedish company “MySQL AB” and later acquired by Oracle Corporation. MySQL is offered as an open-source MySQL community server edition and enterprise server edition.

**Features:**

* MySQL is a database management system.
* MySQL databases are relational.
* MySQL software is Open Source.
* The MySQL Database Server is very fast, reliable, scalable, and easy to use.
* MySQL Server works in client/server or embedded systems.

**How to install Mysql?**

* MySQL installation link for windows operating system:

<https://dev.mysql.com/downloads/installer/>

* MySQL installation link for Linux operating system

<https://dev.mysql.com/doc/refman/5.7/en/linux-installation-yum-repo.html>

* MySQL installation link for other operating system:

<https://dev.mysql.com/doc/refman/5.7/en/installing.html>

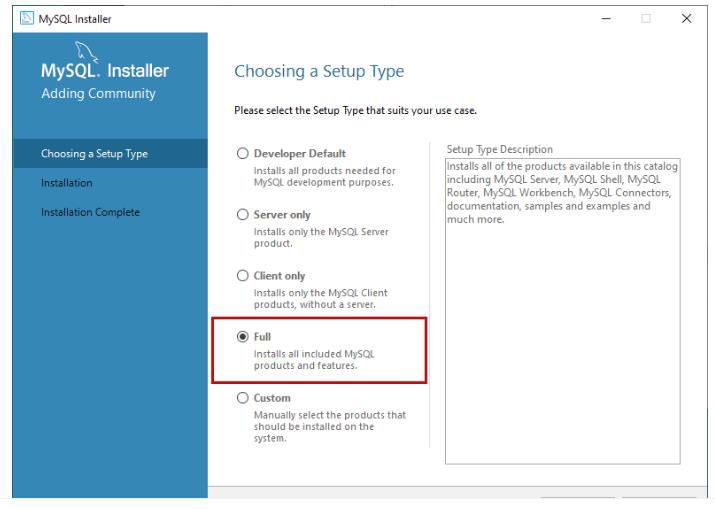
In this practical we will install MySQL community server for windows operating system.

**Step by step installation process of MySQL database server:**

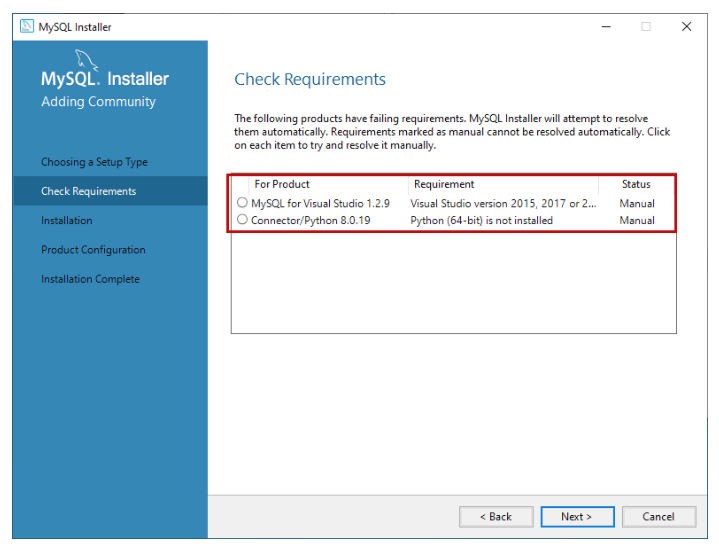
1. Once the installer has been downloaded, double-click the setup file to start the installation process. On the Choosing a Setup Type page, you can see four installation options.

* **Developer default:** If you want to create a development machine, you can use this option. It installs the components which are required for application development, e.g., MySQL Server, MySQL Shell, MySQL connectors, MySQL
* **Server Only:** If you want to create a standalone database server with specific components, you can use this option
* **Full:** If you want to install MySQL Server with its all components, then you can use this option
* **Custom:** If your requirements are limited to the few components, you can use this option

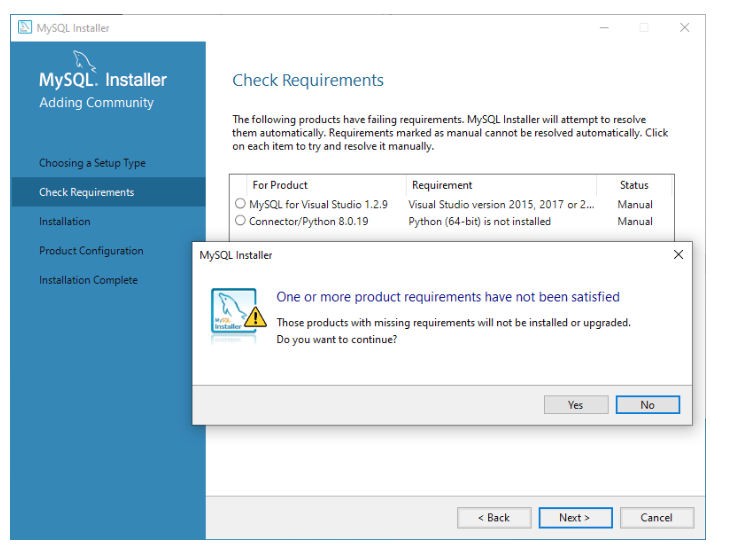
We are going to install MySQL Server with all components; hence, choose “**Full**” and click on **Next**.



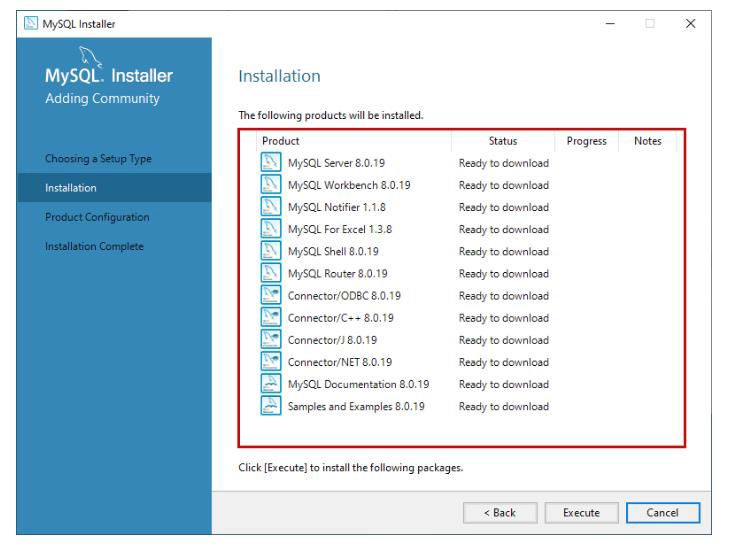
1. Before installation begins, the installer checks all the prerequisites that are required to install all the components of the MySQL database server. If any software prerequisites are missing, then you can see the details of failing requirements on the **“Check Requirements”**screen. It shows the name of the product, required component/software, and its status. As you can see, to install the MySQL database server for visual studio, we must install visual studio 2015 or above. Similarly, to install Python connector, we must install python on the work station. Click on **Next**.



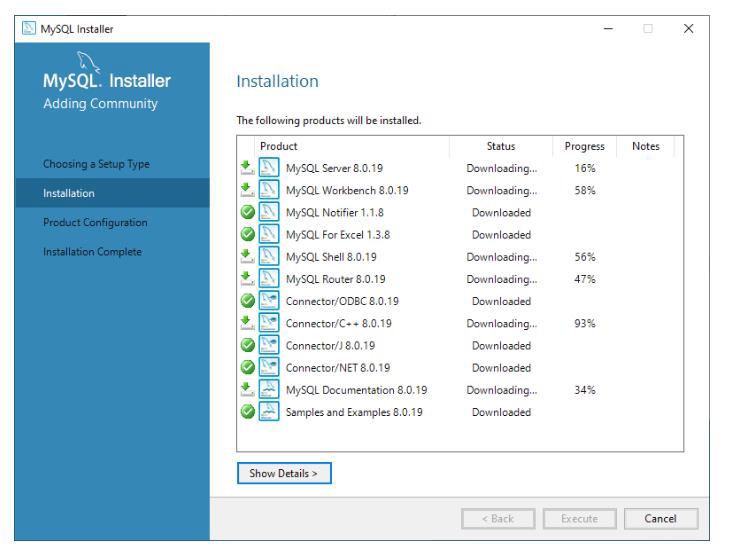
1. An installer gives us a warning. We can continue our installation without installing the visual studio and python. Click on Yes.



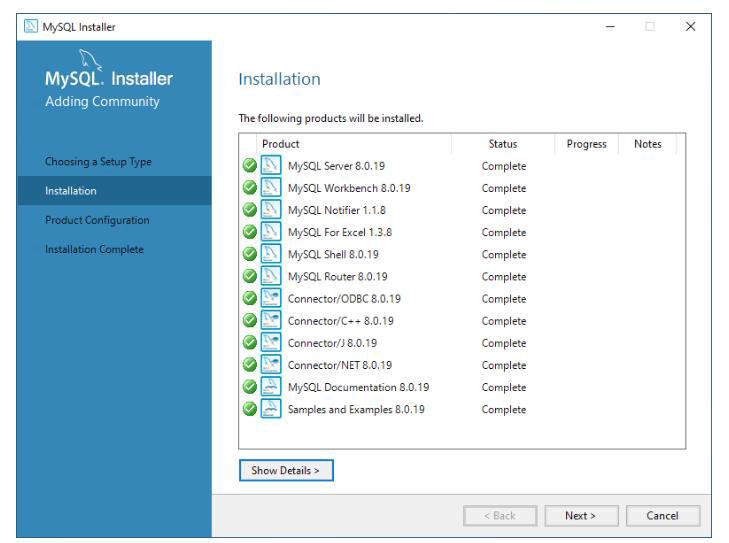
1. On the Installation screen, you can see the list of the MySQL products/software those are going to be installed. Review the list and click on Execute.



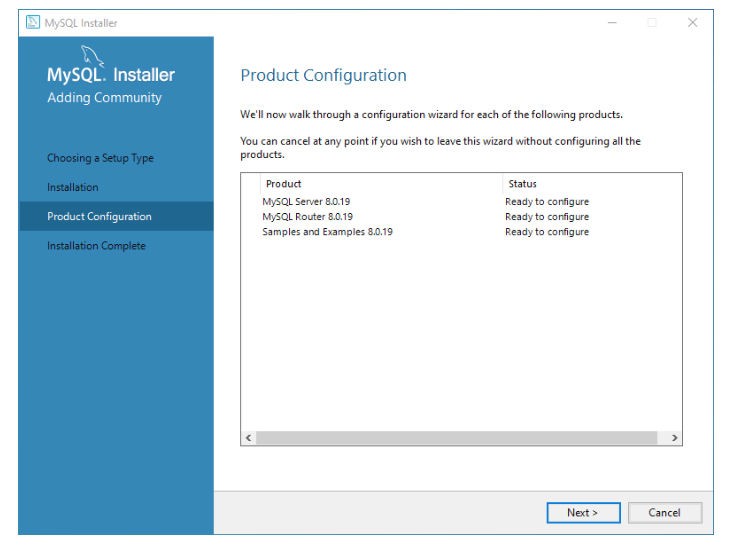
1. The installer downloads all the products/software. After that, it installs all the products.



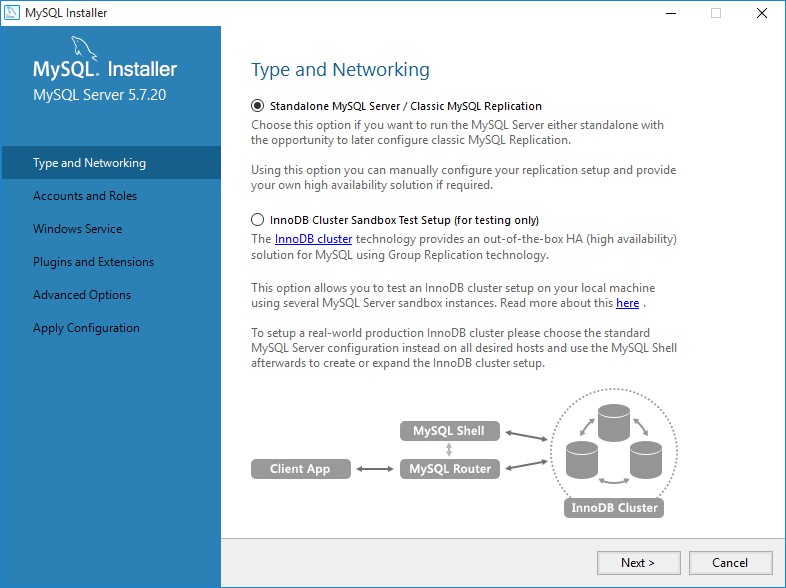
1. Wait for a few mins. Once the installation process completes, we are ready to configure the MySQL database server and other components. Click on Next.



1. On the Product configuration screen, you can see the list of the products that need to be configured. First, let us configure the MySQL Server. Click on **Next**.



1. On the High availability screen, we are going to perform a standalone installation of MySQL Server hence choose “Standalone MySQL Server / Classic MySQL Replication”. Then click Next.

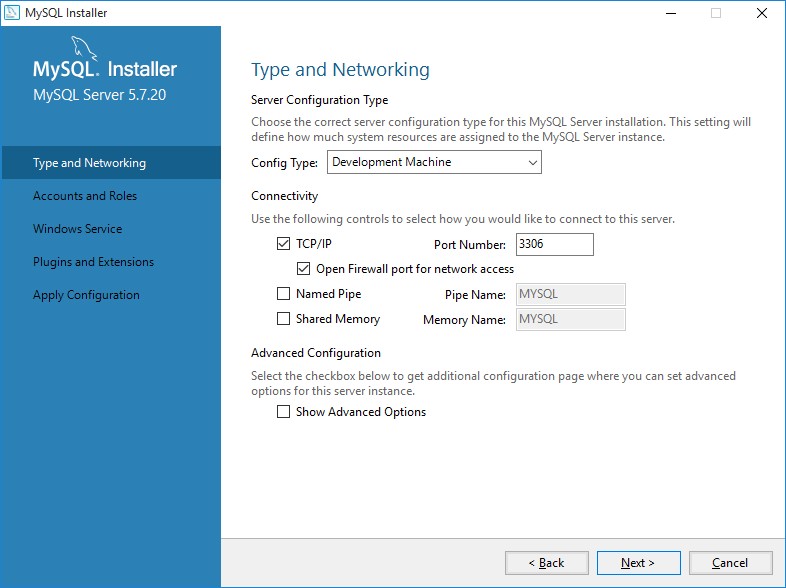


1. On **Type and Networking** screen, we can configure the following:

**The type of MySQL configuration:** This is a predefined set of configuration parameter that determines how much resources should be allocated to the MySQL Services. You have three configuration options:

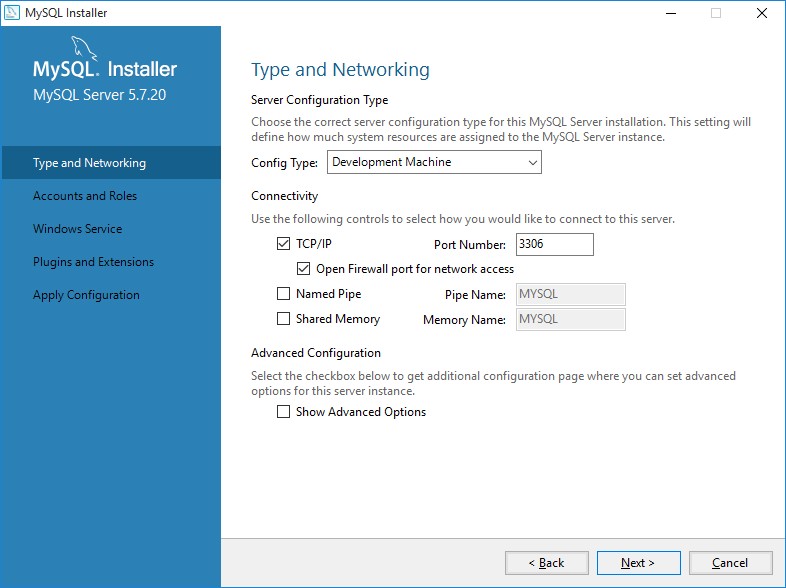
1. **Development Computer:** This configuration uses a minimal amount of the resources to MySQL Service
2. **Server Computer:** This configuration uses a minimal amount of resources. This option is suitable when we are installing database servers and web servers on the same machine. The configuration allocates an average amount of resources to MySQL Service.
3. **Dedicated Computer:** This option is used when we have created a dedicated MySQL Server. The configuration allocates a high amount of resources to MySQL Service.

We would configure the server with minimal resources hence select “Development computer” from the Config Type drop-down box.

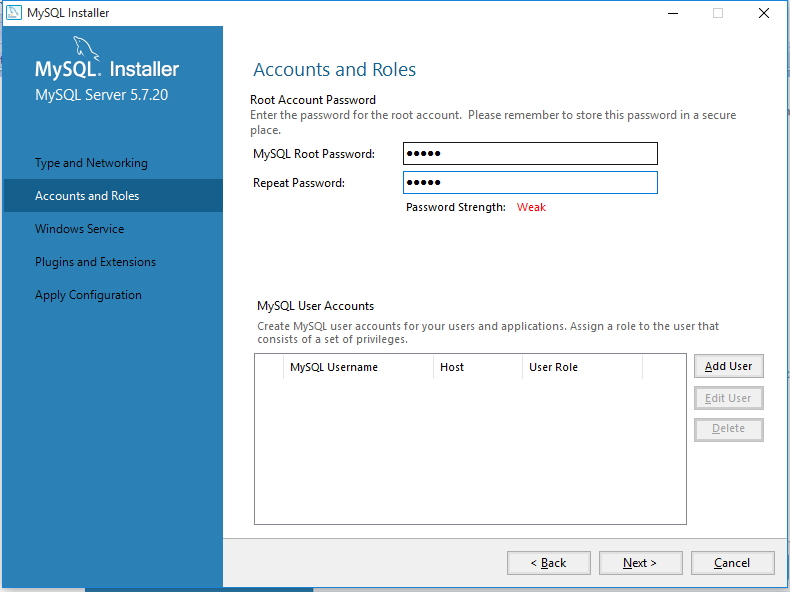


1. Under connectivity, make sure TCP/IP checkbox is selected. Specify the connection port. The default setting is 3306 - leave it unchanged if there is not special reason to change it.

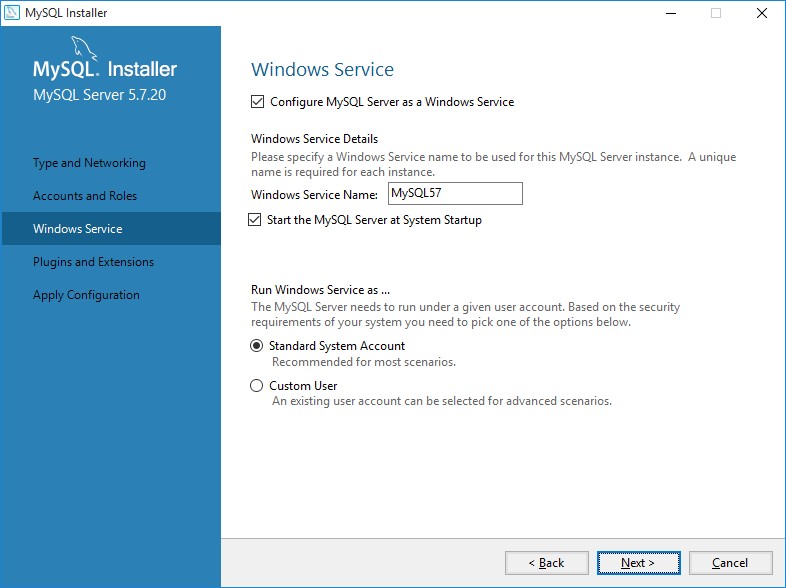
Select Open Firewall port for network access to add firewall exception for the specified port.



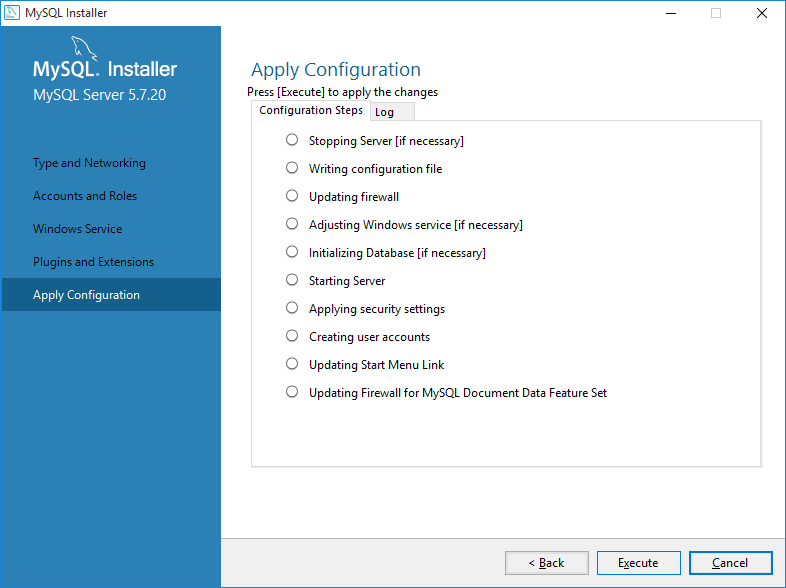
1. On Accounts and Roles screen, specify the MySQL root account password. You can also create other users to do that click on Add user. Provide information for user and then click Next.



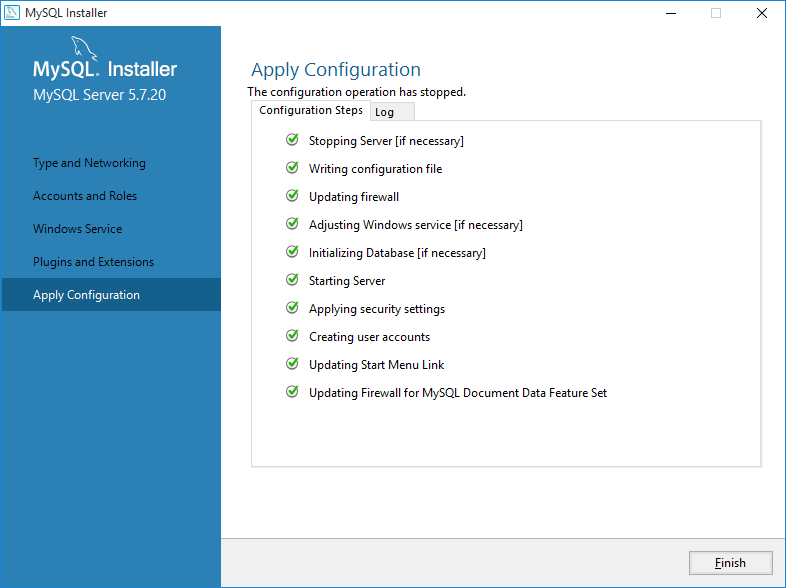
1. On the Windows Service screen, you can configure the MySQL server to run as a windows service. You can provide the desired name and configure it to auto-start the service when the system reboots. You can provide the credentials under which the MySQL Service will run.



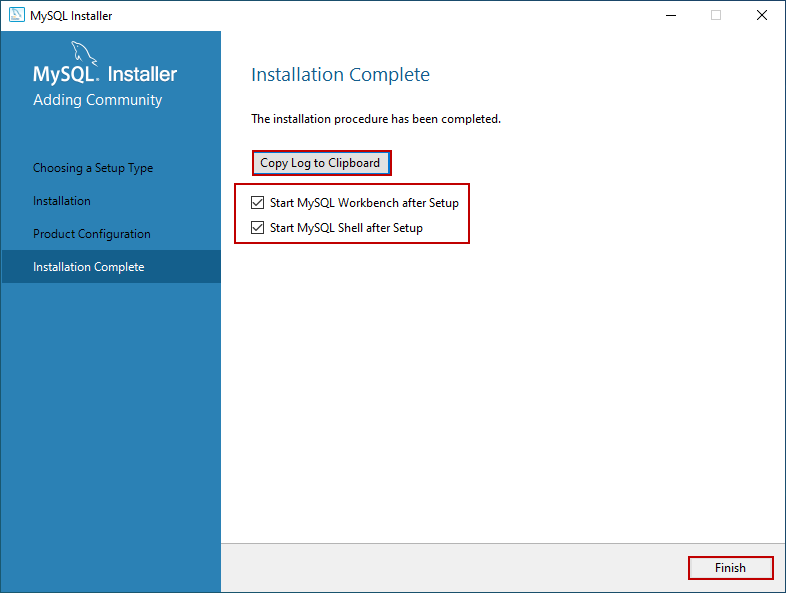
1. On the Apply Configuration screen, you can see the list of confirmation steps. Once all the configuration settings are verified, click on Execute.



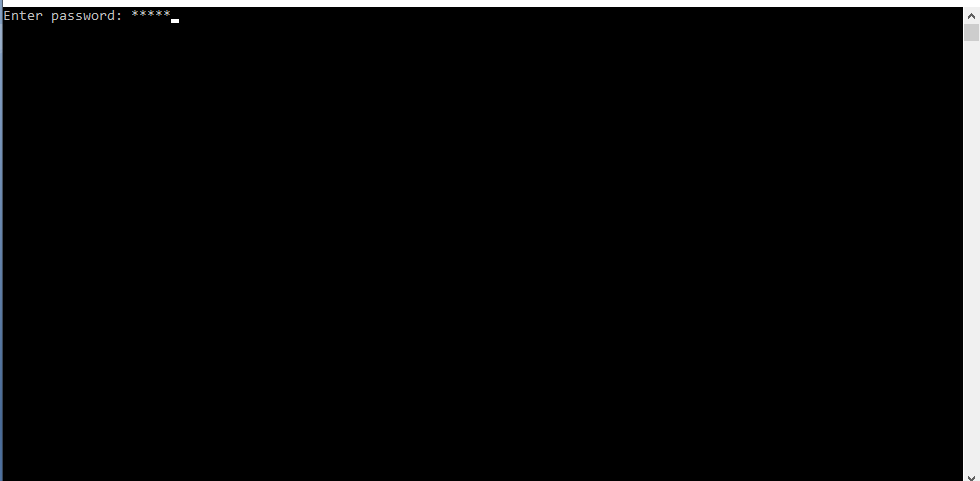
1. The MySQL installation process starts. You can view the installation process in the “Log” tab. Once installation completes successfully, click on “Finish” to close the installer.



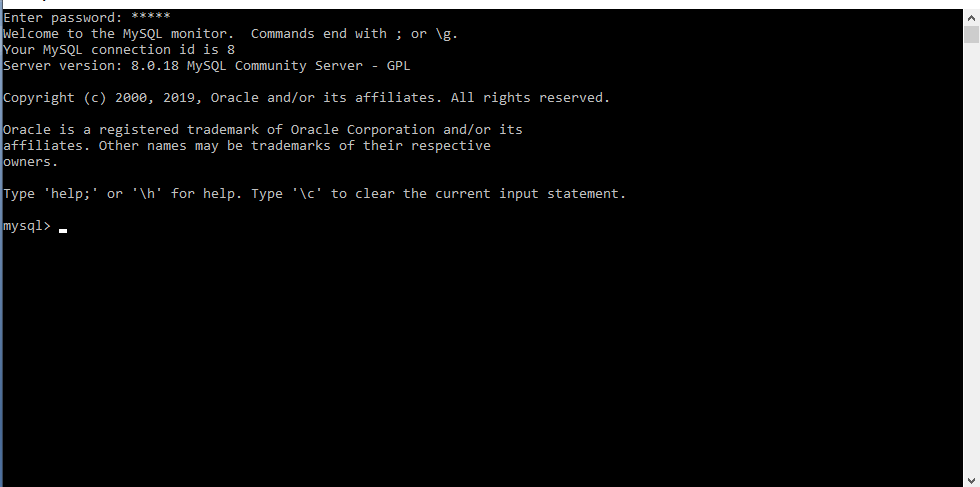
1. The installer continues to the Product Configuration screen. Click next and you can see following image.



After installing MySQL on windows operating system, start MYSQL Command Line Client and enter password to connect with MySQL server.

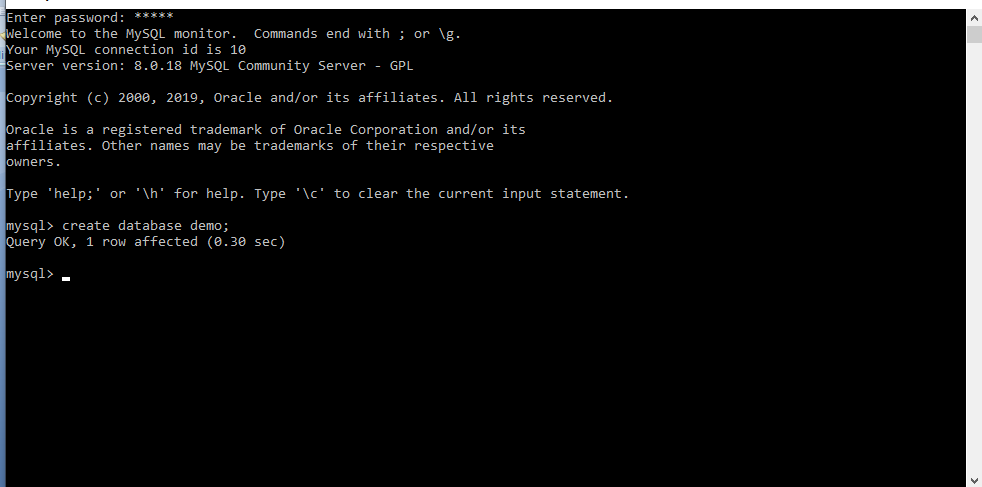


After connecting with MySQL server, you will get following prompt. (mysql>)



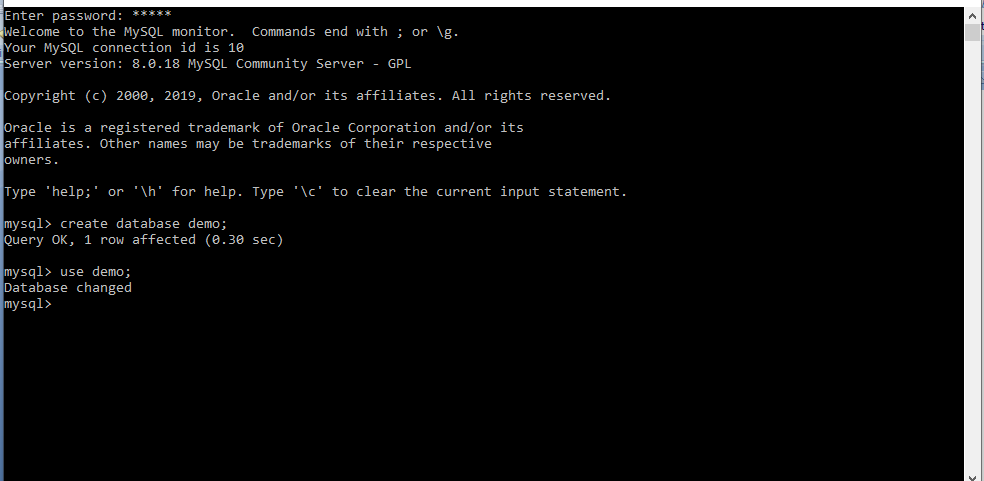
Now create database demo using MySQL query.

create database demo;

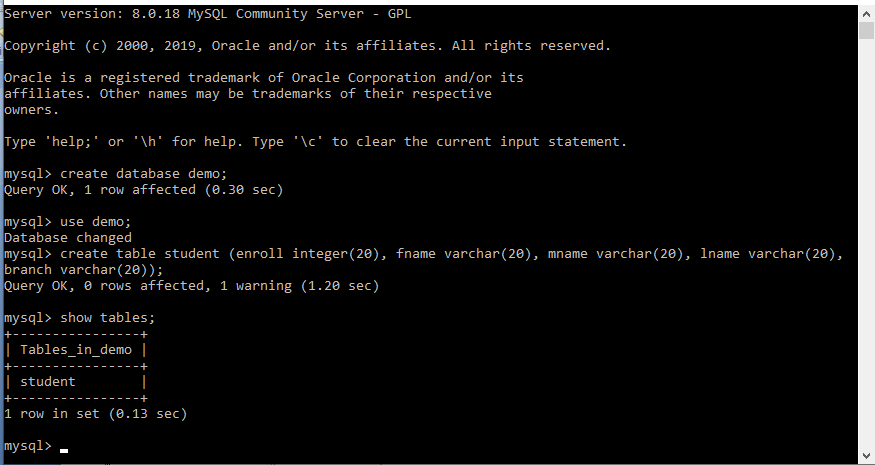


Now use that database that you have created by using “use” query.

use demo;



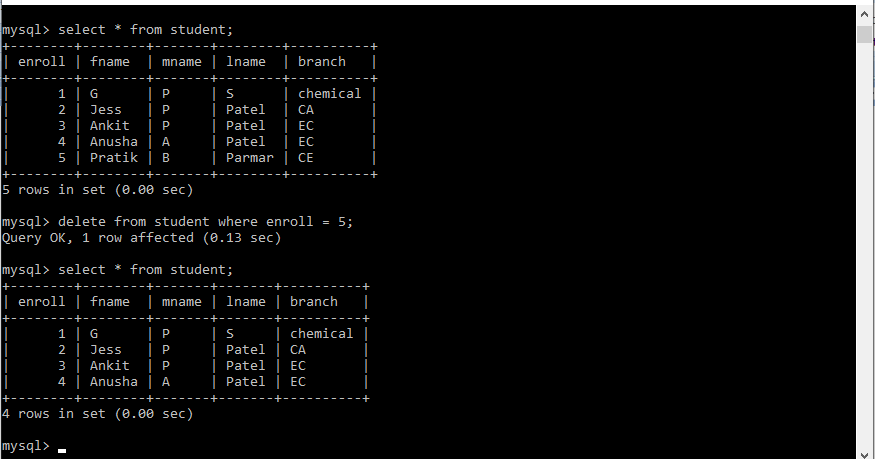
Now create table student as shown in the following snapshot. Where enroll is enrollment no, fname is firstname, mname is middlename, lname is lastname, branch is your stream or branch.



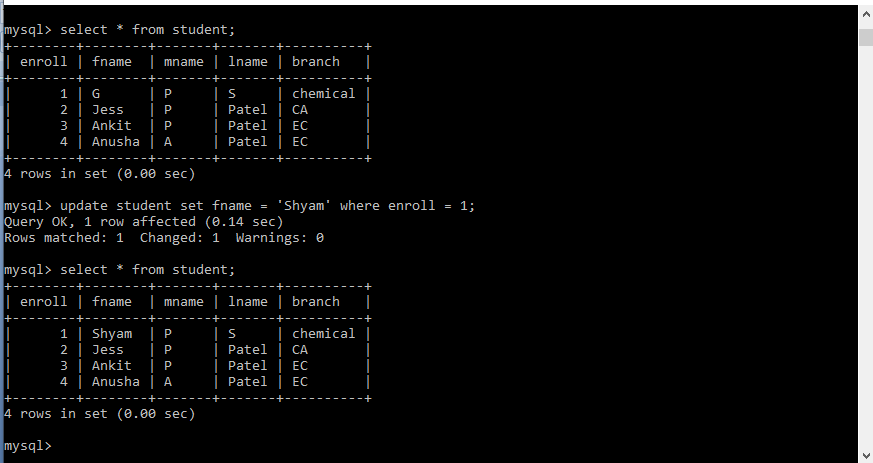
Now insert data using insert query in the table that you have created.



Now delete data using delete query from the table that you have created.



Now update data using update query in the table that you have created.



**PRACTICAL – 2**

**AIM:** Install & Use of SQLyog

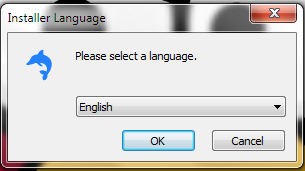
**Basic Information:**

The SQLyog program is probably the most popular GUI tool for MySQL. The program has been around since August 2002 and the maturity of the product is evident when we use the tool.

For people who come from a Windows desktop background, this can be a very useful and practical tool. Though it only runs on Windows desktops, there is both a free community version that is somewhat limited and an enterprise version that currently costs less than 100 USD for a single user license.

**Installation:**

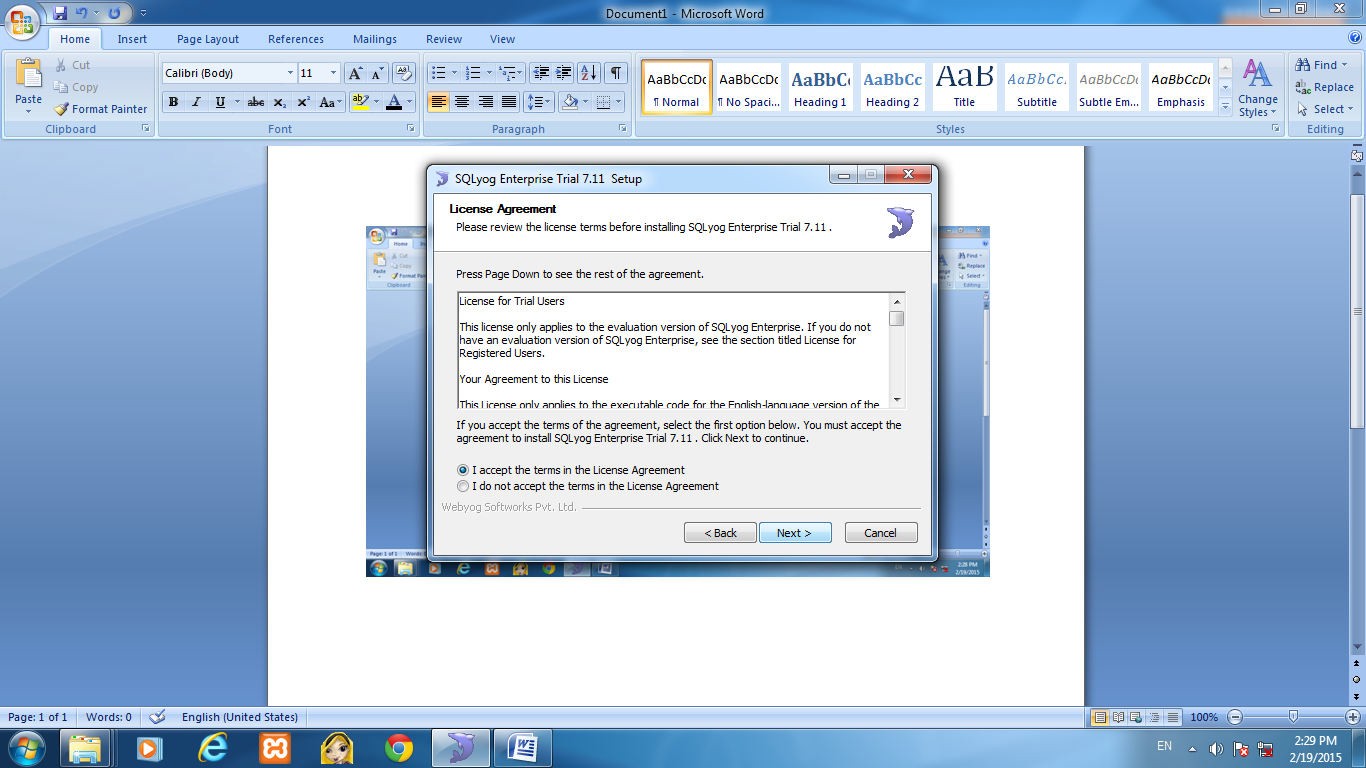
**Step – 1: Click Ok To install with English language.**



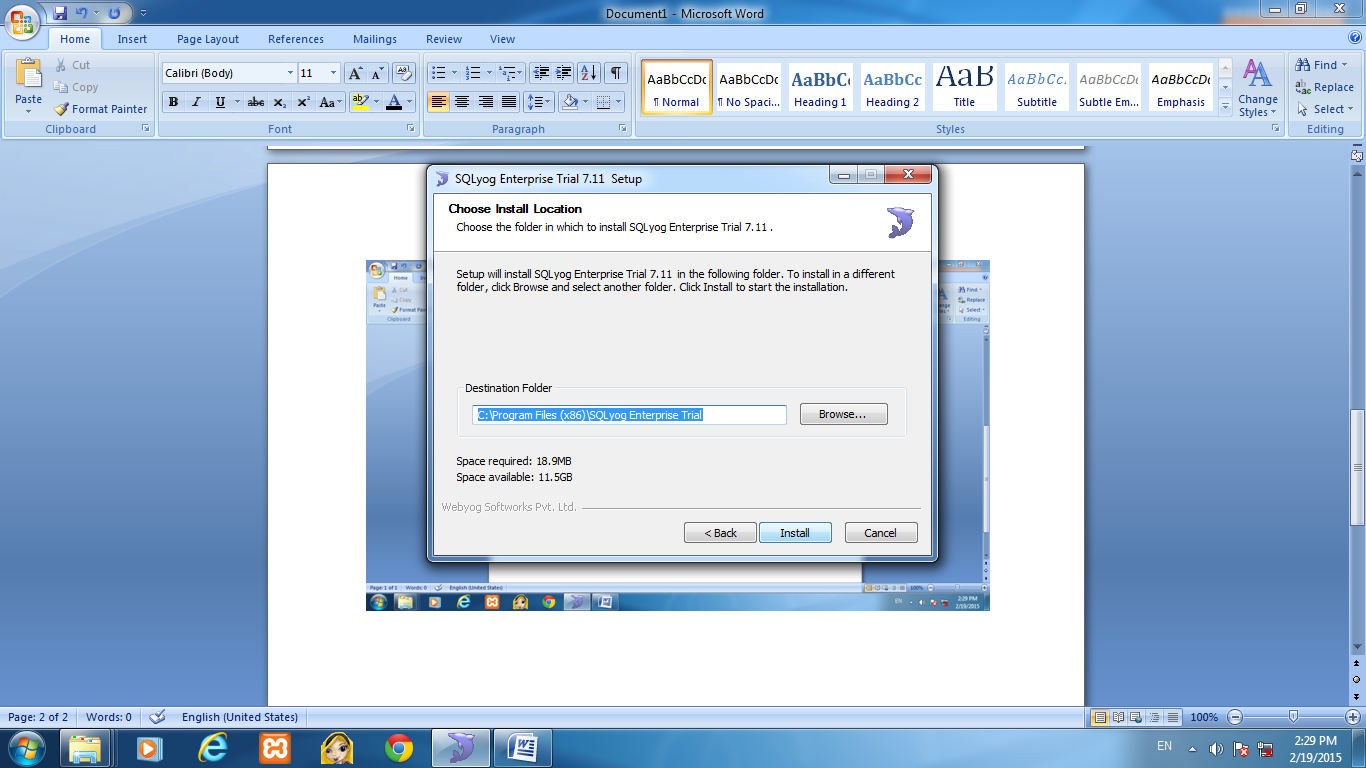
**Step – 2: Click next on the First setup screen.**



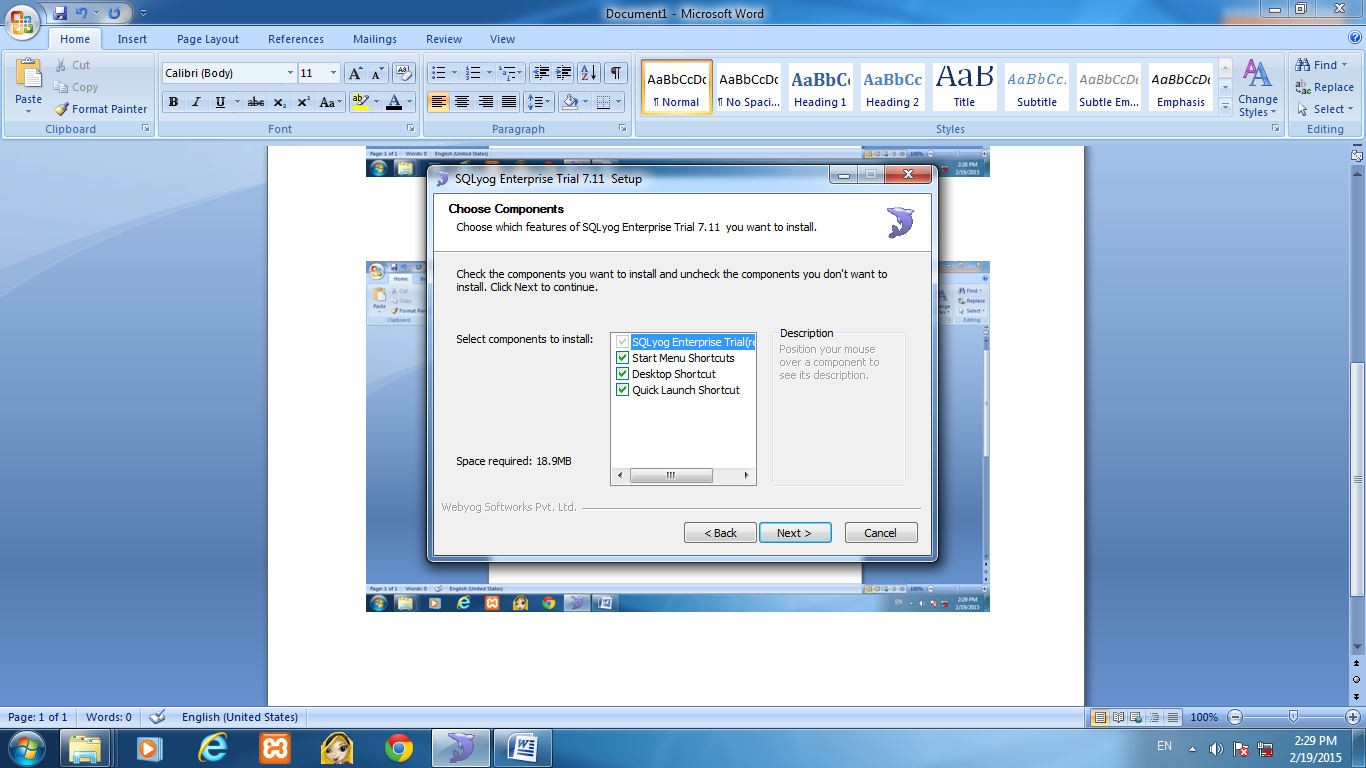
**Step – 3: Tick the I accept the terms of the License Agreement and Click next on the setup screen.**



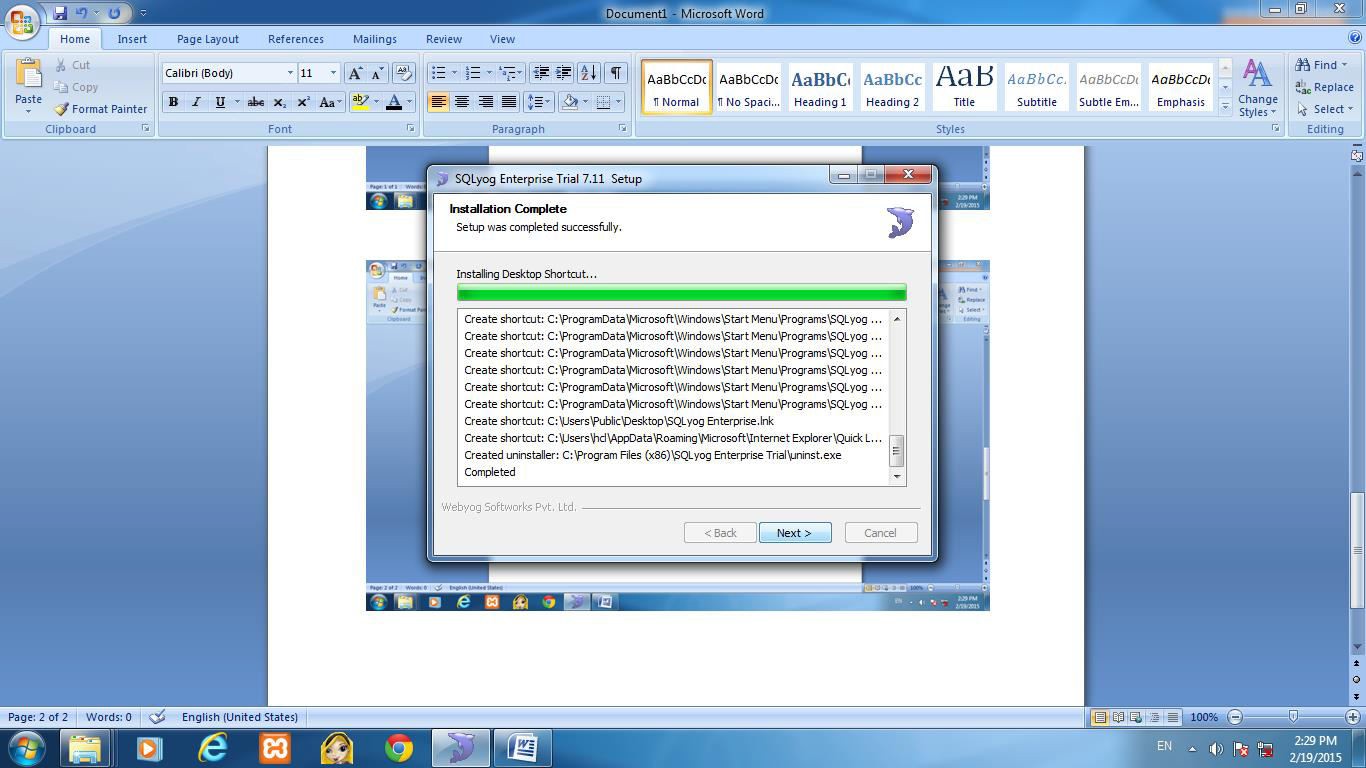
**Step - 4: Browse the path where you want to install and hit the Install button. Then following dialog box appear.**



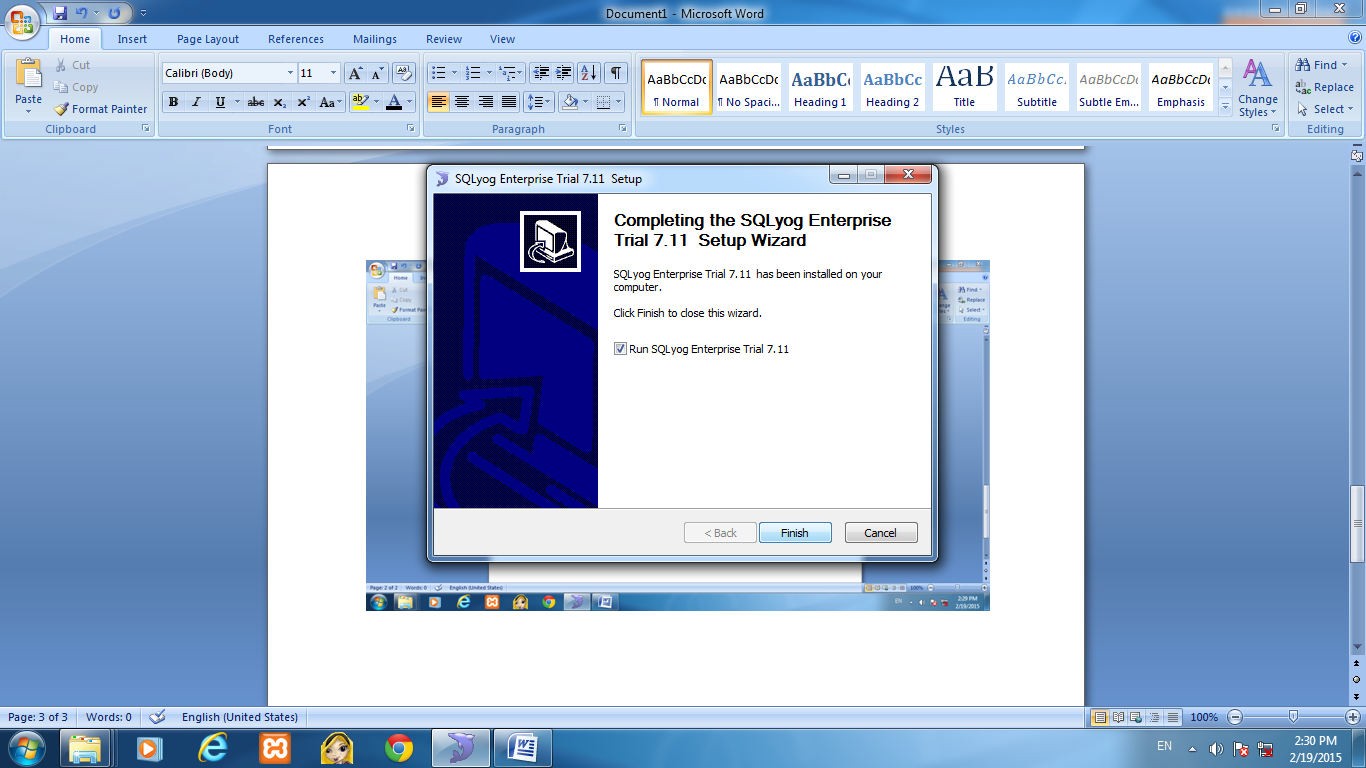
**Step – 5: Click on the Next Button.**



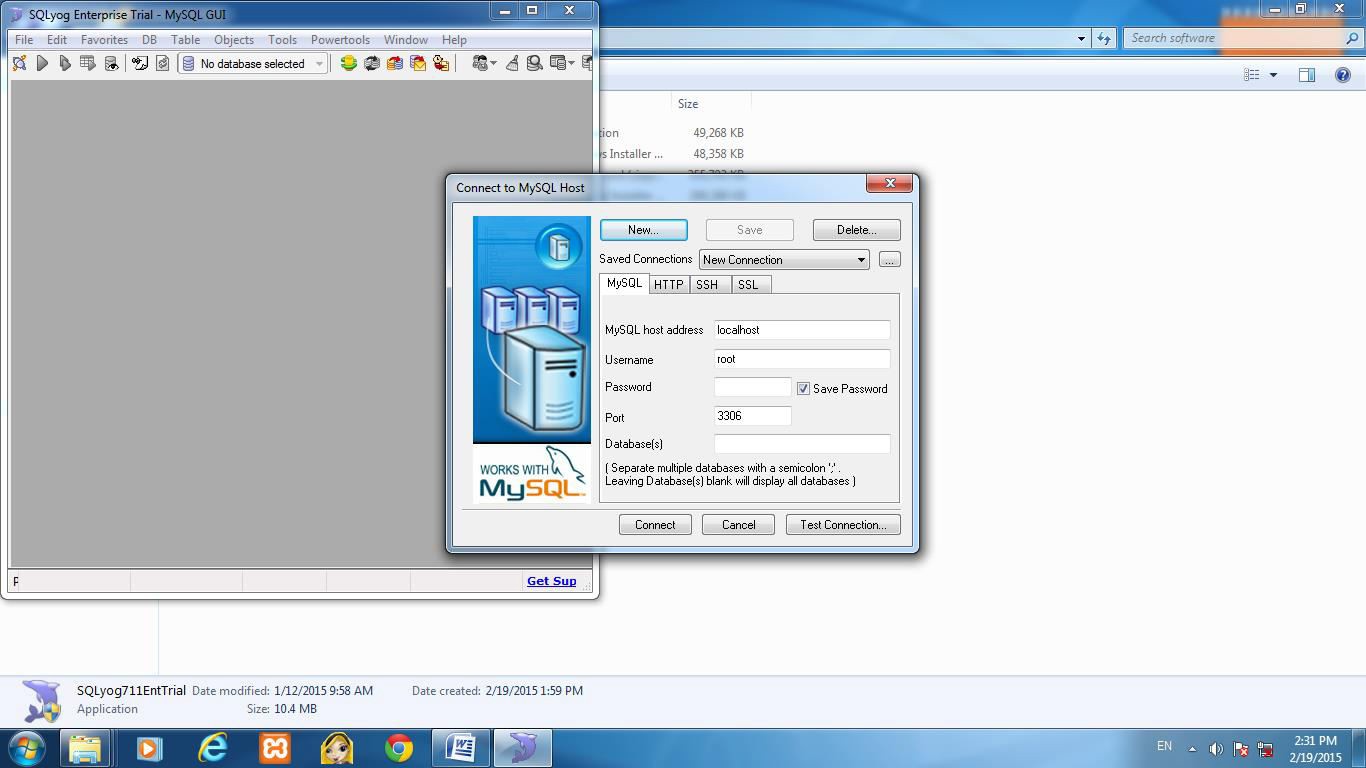
**Step-6: Wait until it completes the process.**



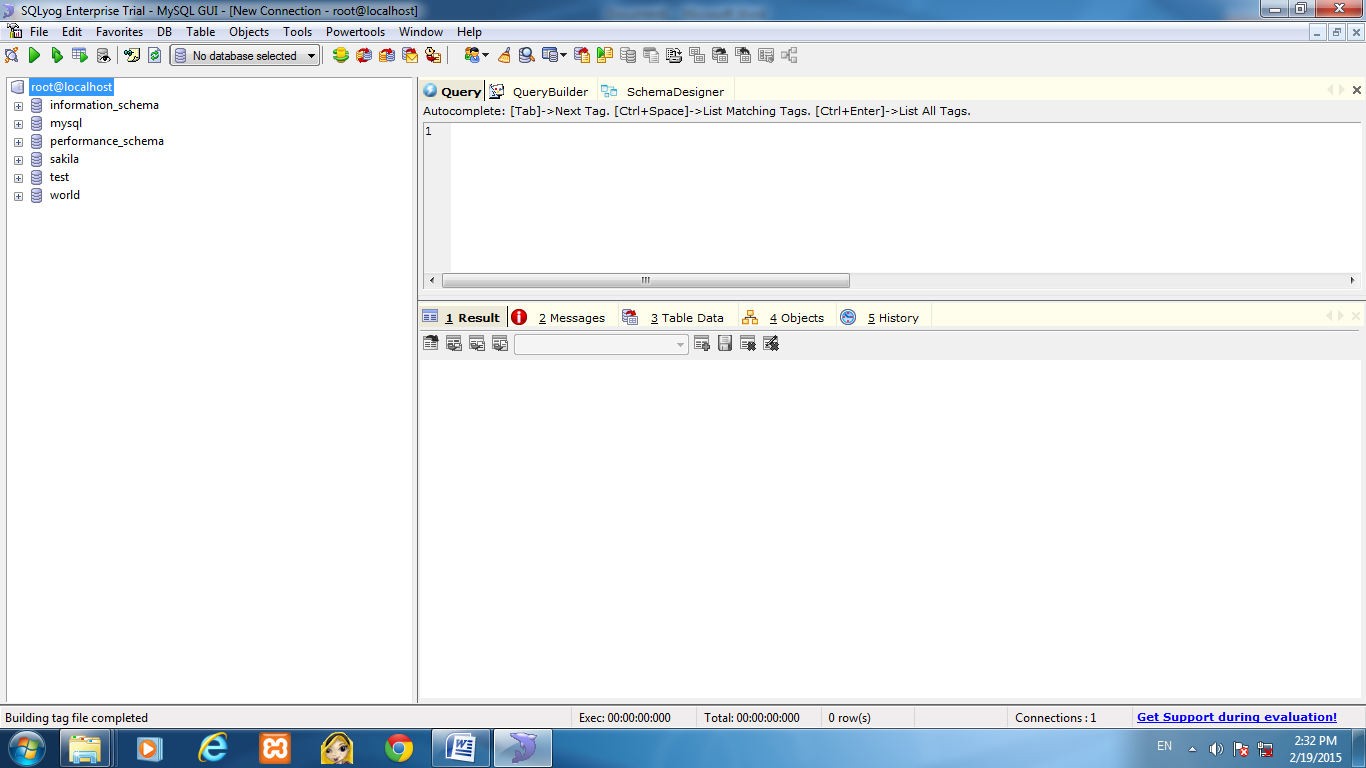
**Step-7: Click on the Finish button.**



**Step-8: Click on new button to make new connection. Enter details as shown in below. Click the connect button to make a connection.**

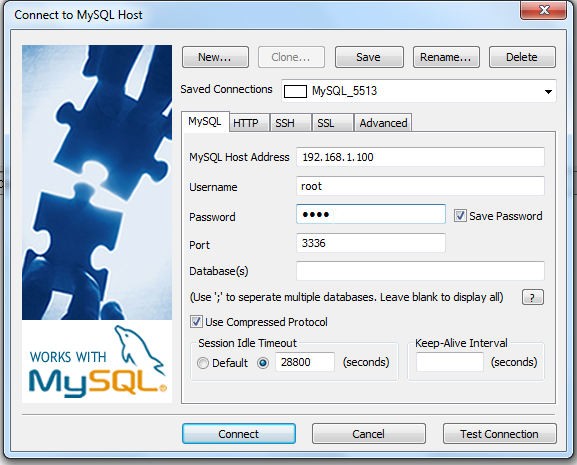


**Step-9: Expand the database to create and alter tables, views, stored procedure, functions, triggers, events.**



**Server connections**

With any of these GUI tools, first we will need to make a connection to mysql. It is common to use the ssh protocol to establish secure connections. One of the reasons SQLyog stands out among other tools is that it is easy to create an ssh connection using what is called ssh tunneling. Creating an ssh tunnel is as easy as checking a box and filling in the information about the server host and the mysql instance. If we experience any problems SQLyog has very thorough, context-sensitive help.



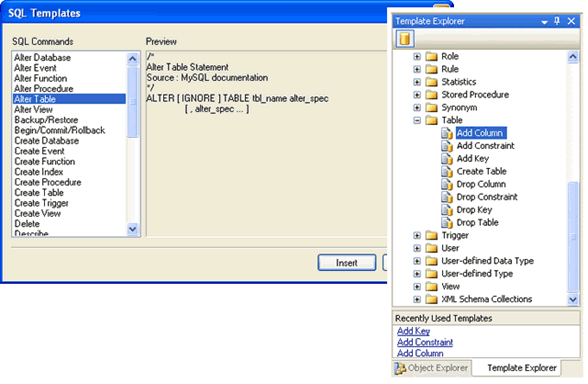
**SQL query creation**

SQLyog has two methods of SQL query creation.

With the first method we type the query in a query building window. SQLyog provides two tools to assist us — the auto-completion feature and SQL templates. mysql does not have tab completion on Windows. With SQLyog we can use the Tab key to complete table or column names or even functions, no matter what operating system we are using. SQL templates help with the proper syntax for the various SQL commands.

First, right-click in the query building window. This will bring up a menu where we left-click Insert Templates to use a template. Figure shows the list of available templates with the CREATE TRIGGER statement template selected. At this point we just click the Insert button and SQLyog will create the template for us to fill in with our specific data.

The second method of query writing is using the query builder to create the query. This is a more graphical method of query building where we drag the appropriate tables from the schema objects to a blank window. We can then do the common query actions of joining tables, sorting, filtering, and grouping based on our chosen criteria. SQLyog will then build the SQL command for us to execute without actually having to type any SQL code.

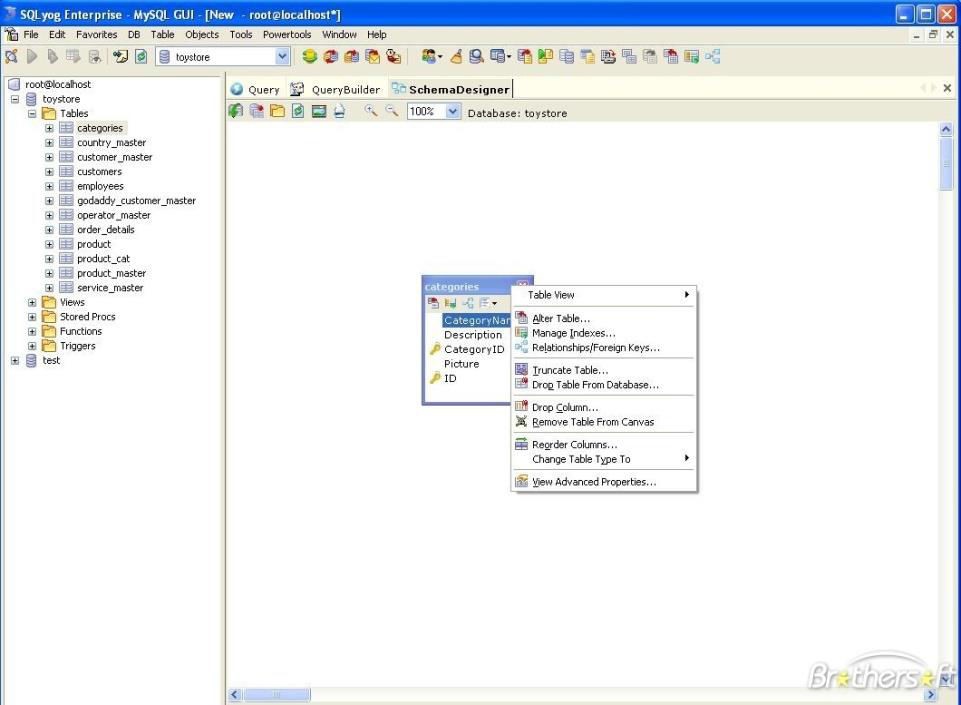


After building the query we can copy the query over to the Query tab and then execute it with just a couple of clicks.

**Database management**

SQLyog has numerous features designed for database management. Among the tasks we can perform are:

* Create new tables
* Manage table indexes
* Manage users
* Synchronize databases
* Backup and restore databases
* Create triggers, views, events, and stored procedures



**Database synchronization**

One of the best features of SQLyog is the database synchronization with its power tool. This tool allows us to synchronize two different databases by creating the corresponding SQL commands needed to match both the structure and data of the two database tables. There is a wizard that will guide through each step of the process. The process is simple — just select the databases and tables to synchronize and SQLyog takes care of everything else. Resolving a data synchronization problem by hand is incredibly difficult, and SQLyog reduces the complexity to a few simple clicks.

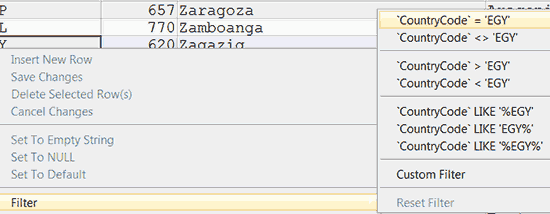
**Switching between Spreadsheet-Style, Form-Style and Plain Text**

Both in "Table Data" tabs and in Query Result tabs, by default, the data is displayed in a grid, which is very handy. But we can also see results in a form (letting we modify records), or as plaintext, which is convenient for copying and pasting. To switch between the Grid and Text styles, I use the default shortcut (Ctrl + L). We can also use the three icons above the grid.

**Quickly Filtering Table Data**

Sometimes, instead of typing a WHERE, we can get quick results using SQLyog's Filter tools. These are accessible both in Result tabs and in the Table Data tab. If we right-click on a cell, we get all kinds of filtering options, including options where the filter is pre-filled with the cell's value.

9

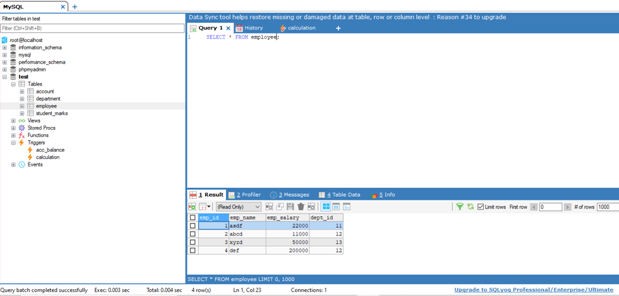


**Exercise:**

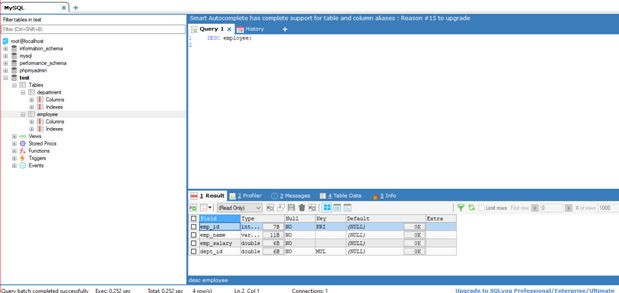
**Create table named employee and department.**

**Inserts the records. Display the content of employee table.**

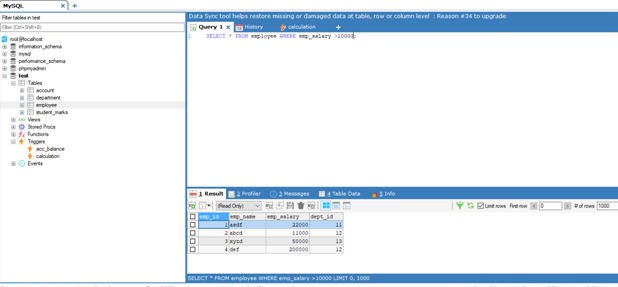
.



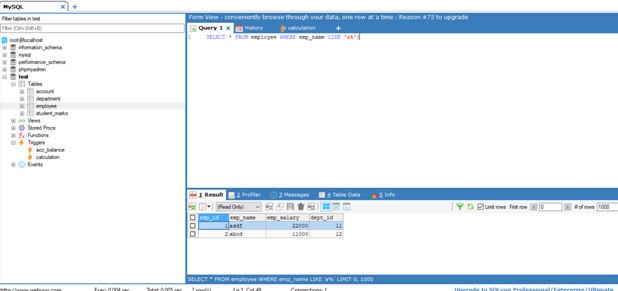
**Describe the structure of employee table.**



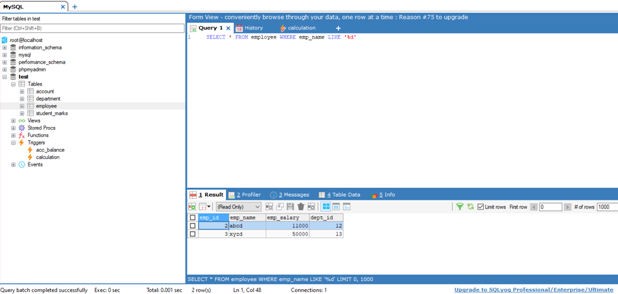
**Select the employees whose salary is greater than 10000.**



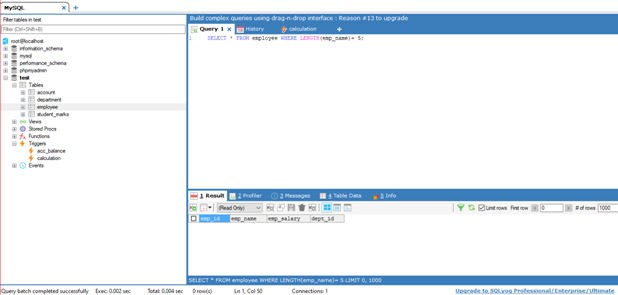
**Display employee whose name starts with “A”.**



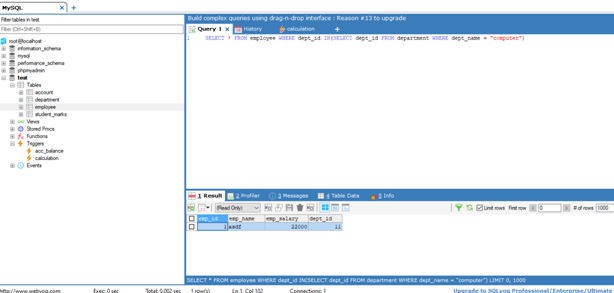
**Display employee whose name ends with “D”.**



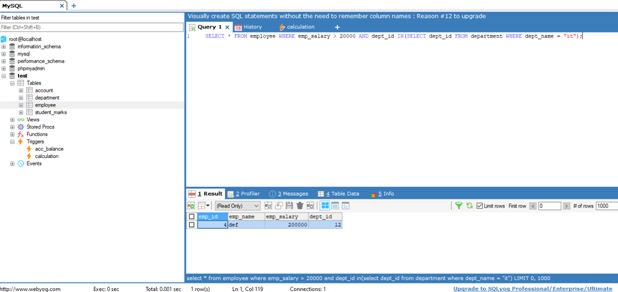
**Display the employee whose name length is 5.**



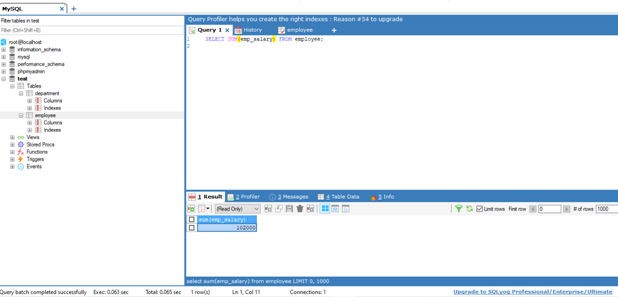
**Show employees from ‘computer’ department.**



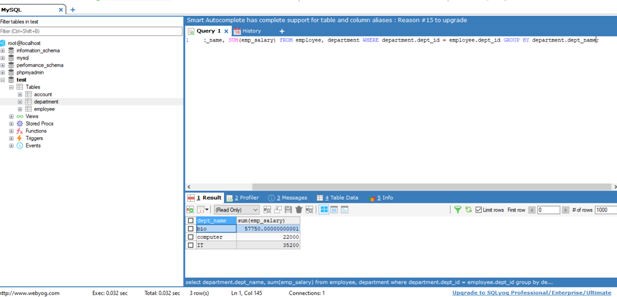
**Find all employees from IT department and salary greater than 20000.**



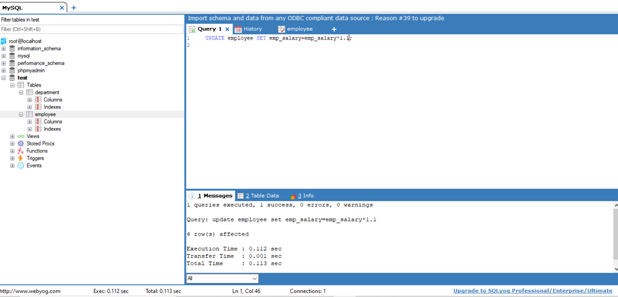
**List total salary of all the departments.**



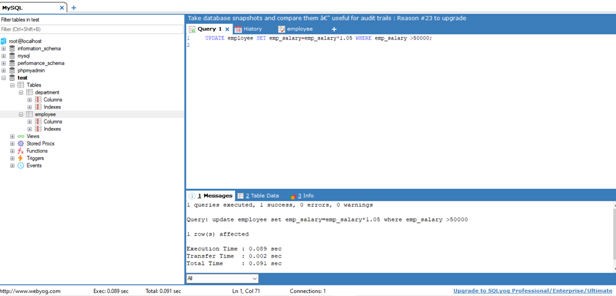
**Get total salary of each department.**



**Give 10% increment to all employees.**



**Give 5% employees to all the employees whose total salary is greater than 50000.**



PROFESSIONALPRACTICESUSINGDATABASE(3360702) PRACTICAL 3

**PRACTICAL – 3**

**AIM:** Install and Use of Phpmyadmin.

**Basic Information:**

PHP (PHP: Hypertext Pre-processor) is a widely-used Open Source general-purpose scripting language that is especially suited for Web development and can be embedded into HTML.

PHP is one of the most popular server side scripting languages running today. It is used for creating dynamic WebPages that interact with the user offering customized information. PHP offers many advantages; it is fast, stable, secure, easy to use and open source (free).

[PHPMyAdmin](http://www.phpmyadmin.net/home_page/index.php) (PMA) is an excellent free, open source web-based database client which can be used to interact more easily with MySQL and application databases.

**Installation:**

Download the latest stable version of the [PhpMyAdmin software](http://www.phpmyadmin.net/home_page/downloads.php). Extract the archive file on computer. Then upload the extracted files and folders through an [FTP Client](https://www.siteground.com/tutorials/ftp/ftp_client.htm).

We will Place them at the exact location where we would like to access the software. If we want to open it directly through http://www.yourdomainname.com/PhpMyAdmin we should place the content of the extracted directory straight under the public\_html/PhpMyAdmin folder of our account. Substitute www.yourdomainname.com with our actual domain name.

Add a database user with full privileges to the databases which we want to manage through our cPanel->MySQL Databases.

Load the PhpMyAdmin page using the corresponding URL.



1

Enter the database username and the corresponding password. In this way we will be able to manage all the databases for which the database user has privileges. Click on the Go button to log in. Now we can proceed with the databases management.

**Database Management:**

The main functionality of the PhpMyAdmin tool is to manage the databases.

Click on the Databases link. Pick the preferred database which we want to manage and click on its name.

.



In the new screen you will see a list with the database tables, the allowed actions with them, the number of the records, the storage engine, the collation, the tables' sizes and the overhead.The possible actions which we can perform to a chosen table are:

**1)Browse**

Only the tables with existing records can be browsed. Once we click on the Browse icon a new window with the records list will be opened. By clicking on the Pen icon we can edit the chosen record. we will see the record structure and you can alter the values of the records.

**2)Structure**

The next option is named Structure. In the Structure screen we will see the database's table structure.

we will see the fields' names, their types, collations, attributes, additional extra information,  the default values and whether the fields' values can be NULL. we can browse for distinct values by clicking on the corresponding action icon. Also, we can edit a field's structure or delete a field. we can define different indexes: Primary, Unique, Index and Full text. More about the indexes can be found in the [MySQL Indexes](http://dev.mysql.com/doc/refman/5.0/en/mysql-indexes.html) documentation.

In the Indexes area we will find the indexes assigned for the table and the fields for which they are set. we can edit and delete them.

**3)Search**

Through the Search action we can generate a search query for the chosen table.

we can either write the WHERE clause or we can use the "query by example" functionality. we should click on the Go button to execute the query.

For example, if you want to visualize all the records with a field value that starts with a we should select the fields which we want to show. Pick the LIKE operator from the drop-down menu and enter in the corresponding field value a% (% stands for a wildcard string). Click on the Go button to see the result.

**4)Insert**

Using the Insert action we can insert records in your database table.

Once we fill in the corresponding values click on the Go button and the new record will be inserted.

**5)Empty**

The Empty action allows us to empty your database table, removing the data and keeping the empty table

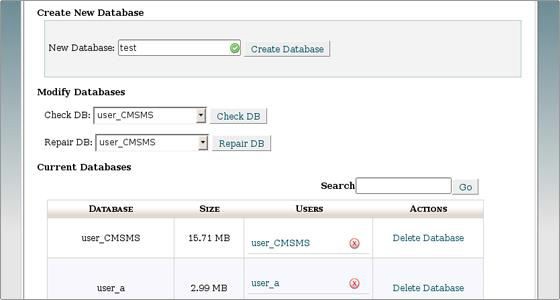
**6)Drop**

Through the Drop action we can delete the whole table and all the records stored in it.

Page Break

**How to Create a MySQL Database?**

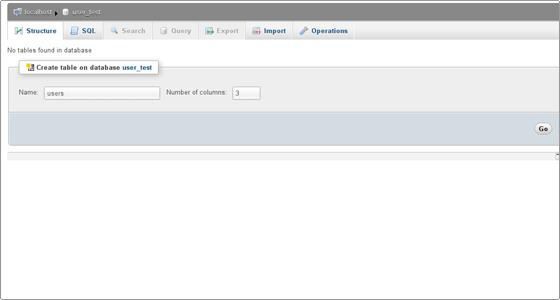
We cannot create a database directly through cPanel->PhpMyAdmin due to the lack of user privileges. However, we can easily create a new database from our cPanel->MySQL Databases. Navigate to the Create New Database box. Enter the database name in the New Database text field and click on the Create Database button.



The database name will be preceded by the cPanel username.

**How to Add MySQL Database Tables?**

Navigate to our cPanel->PhpMyAdmin tool and open the newly create database. It is empty and there are no tables.



The database name will be preceded by the cPanel username.

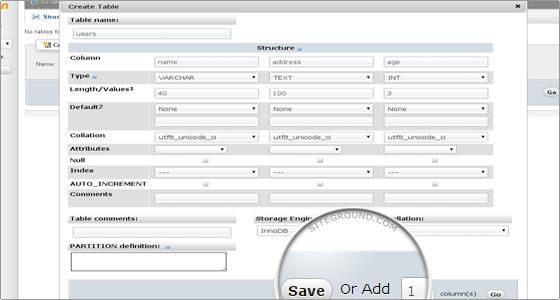
**How to Add MySQL Database Tables?**

Navigate to our cPanel->PhpMyAdmin tool and open the newly create database. It is empty and there are no tables.

Enter the table name and the number of fields. Click on the Go button to create the table.

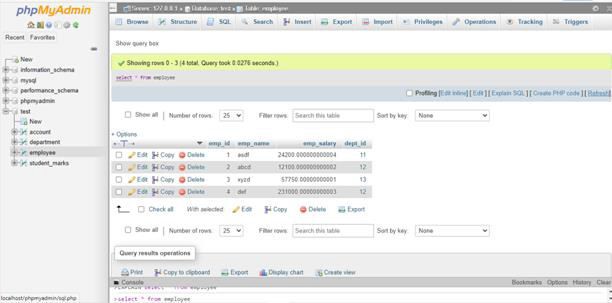
On the next screen we should enter the fields' names and the corresponding properties. The properties are:

Type, Length, Values, Collation, Attributes, Null, Default, Extra, Comments. Then click save button.

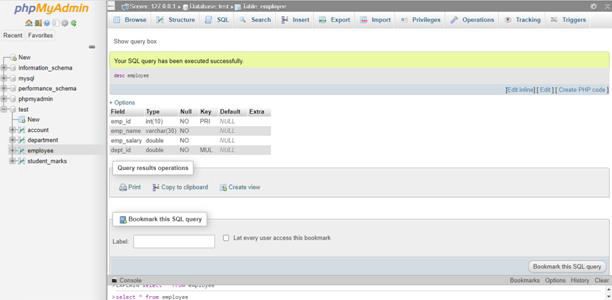


**Create table named employee and department.**

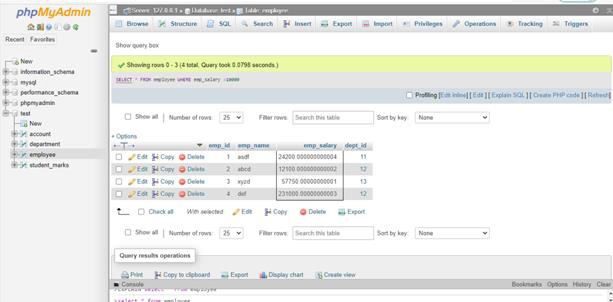
**Inserts the records. Display the content of employee table.**



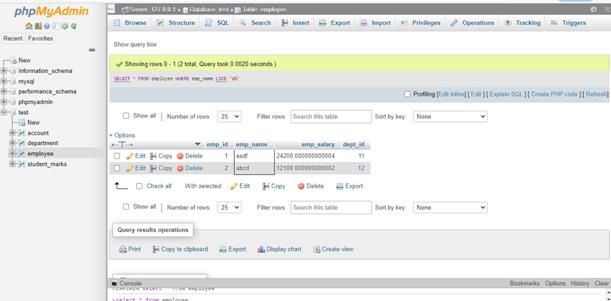
**Describe the structure of employee table.**



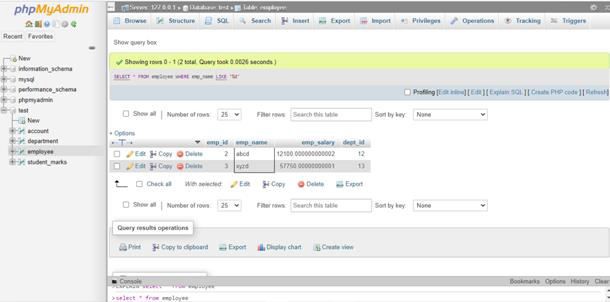
**Select the employees whose salary is greater than 10000.**



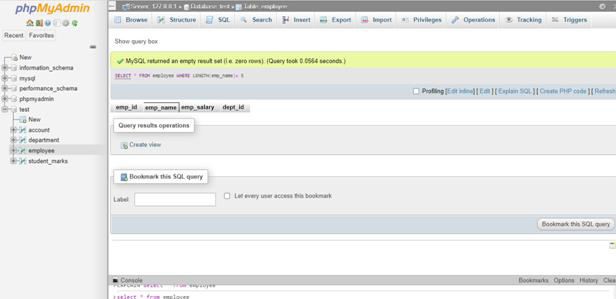
**Display employee whose name starts with “A”.**



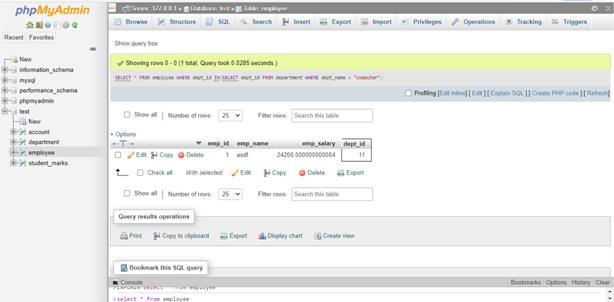
**Display employee whose name ends with “D”.**



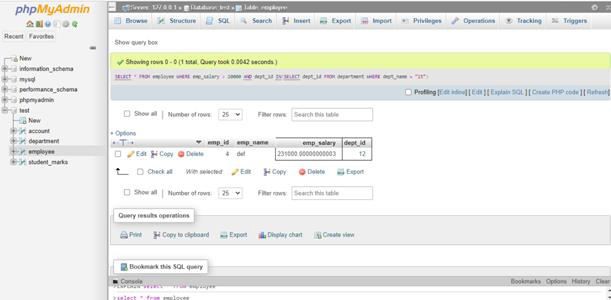
**Display the employee whose name length is 5.**



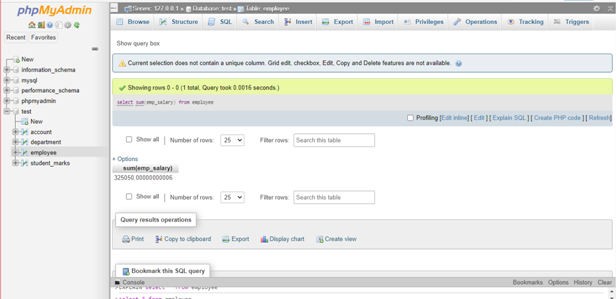
**Show employees from ‘computer’ department.**



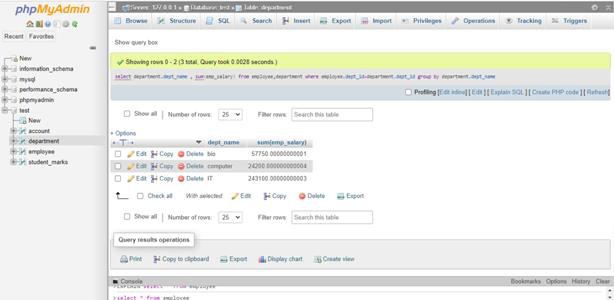
**Find all employees from IT department and salary greater than 20000.**



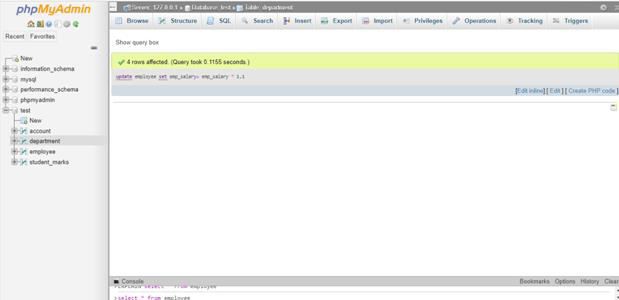
**List total salary of all the departments.**



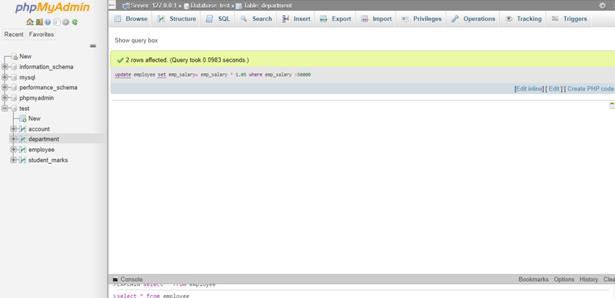
**Get total salary of each department.**



**Give 10% increment to all employees.**



**Give 5% employees to all the employees whose total salary is greater than 50000.**



**PRACTICAL – 4**

**AIM:** Install and Use of MySQL Browser (Query Browser)

**Basic Information:**

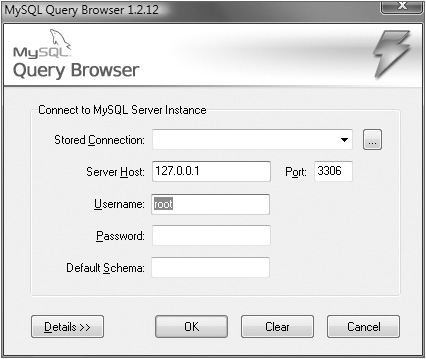
The Query Browser is part of the GUI bundle that MySQL has available for free download from http://dev.mysql.com/downloads/gui-tools. Also included in this bundle are the MySQL Administrator tool and the MySQL Migration Tool. The Query Browser is a GUI tool for creating, executing, and optimizing queries.

The Query Browser includes the following components:

1. Query Toolbar: Create small queries and execute them. Browse old queries.
2. Script Editor: Useful for editing more complex queries.
3. ResultsWindow: Displays query results.
4. Database Explorer: Navigate, create, and drop tables and fields.

**Installation:**

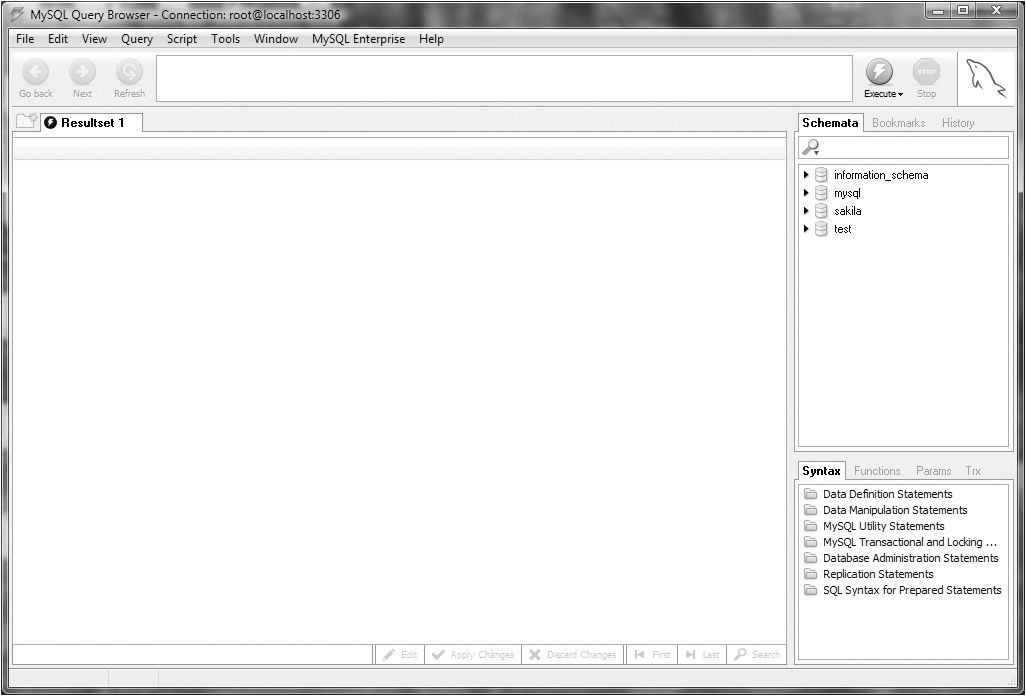
* **Extract MySQL Query Browser and open MYSQL Query Browser. Following Window will be appeared.**



* **Then give name to the Stored Connection, Server Host and Username and then click on OK Button.**

One of the biggest shortcomings of Query Browser is that it does not allow for the direct connection to a server using an ssh session. In order to use Query Browser we must establish an ssh tunnel manually.

Looking on the right-hand side of the screen we can see the databases on the connected server.

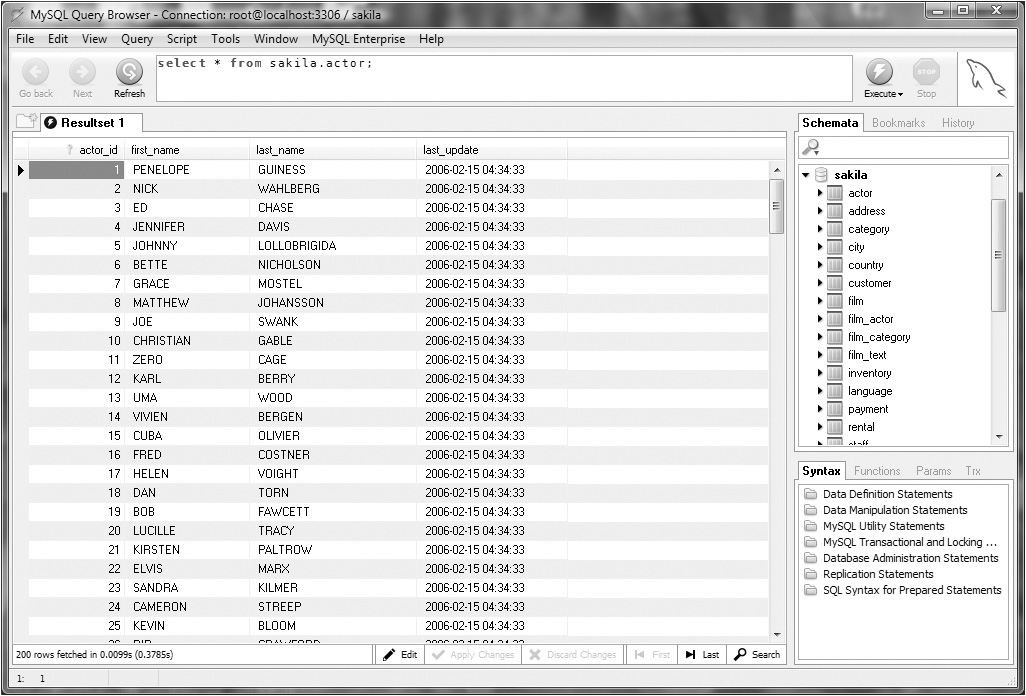


To work with the database a single mouse-click on the will display all the tables. Just enter the query to execute and click the Execute button to the right of query entry screen. Once we enter the query and execute it the result shown in the results pane on the bottom left of the screen. After we have entered multiple queries you can use the Go Back and Next buttons to browse through the history of the already executed queries.

**Features of mysql query browser:**

Under the Tools menu we can start MySQL Administrator, the mysql command-line client, the Windows command line, a text importer and the connections manager, where we can save connection information.

 We can also **manage stored routines very easily** with Query Browser.

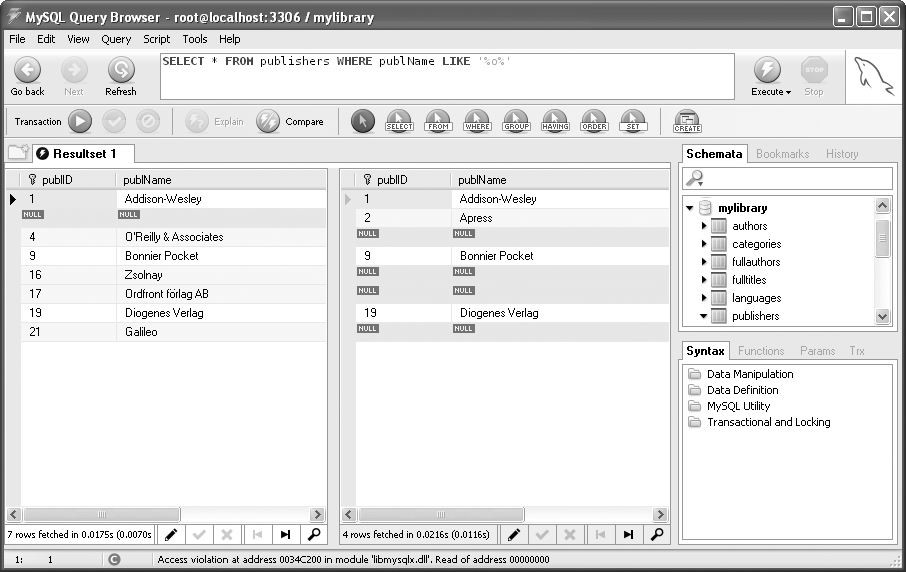


To begin a new stored procedure, make sure we have a database selected and then just click

Script and then Create Stored Procedure/Function.

**SQL commands with a mouse click:**

We can directly click on buttons to get the query we want.



**Exporting SELECT results:**

With FILE | EXPORT RESULTSET you can export the most recent result of SELECT commands in the formats CSV, HTML, XML, and Excel.

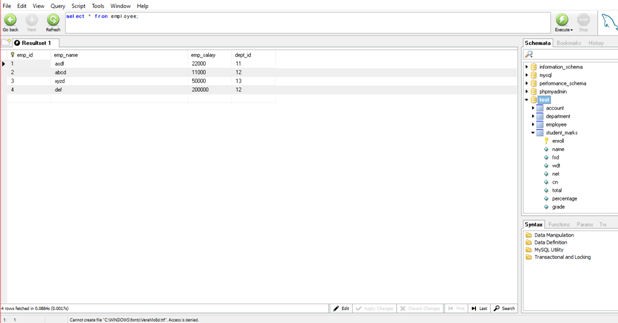
**Comparing SELECT results:**

A special function of the MySQL Query Browser is that it can compare the results of two SELECT commands. To do this, first we must divide the result table with the context menu command SPLIT TAB VERTICALLY. Then execute an SQL command for each of the two domains. Both SQL commands must refer to the same column. Finally, execute QUERY | COMPARE RESULTSETS. The program marks records in white, green, and red depending on whether they are present in the second result, lacking in the second result, or additional records exist.

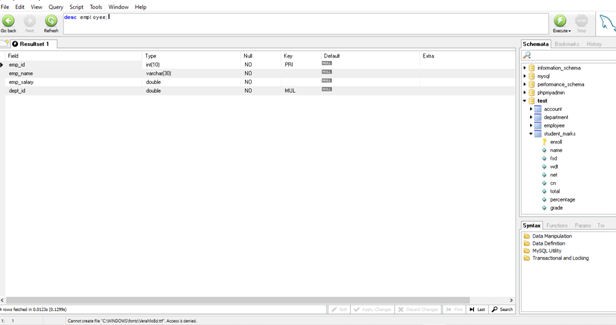
**Exercise:**

**Create table named employee and department.**

**Inserts the records. Display the content of employee table.**

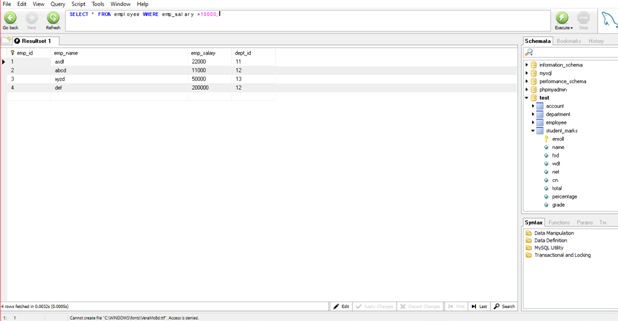


**Describe the structure of employee table.**

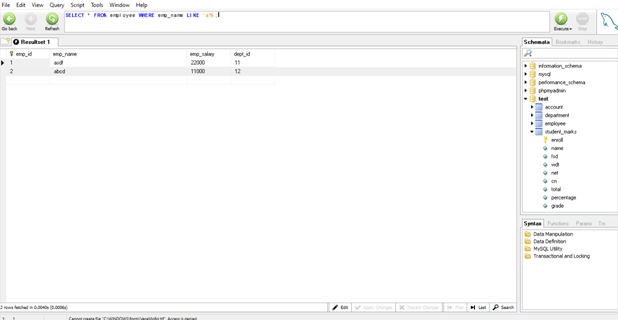


**Select the employees whose salary is greater than 10000.**

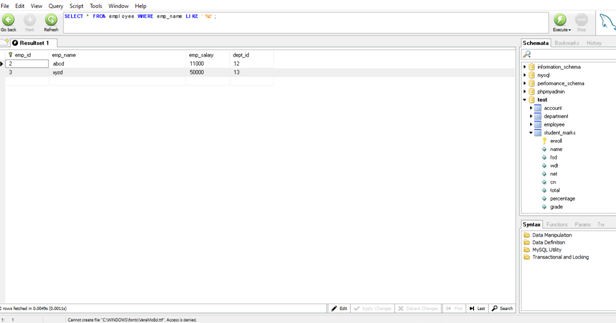
**.**



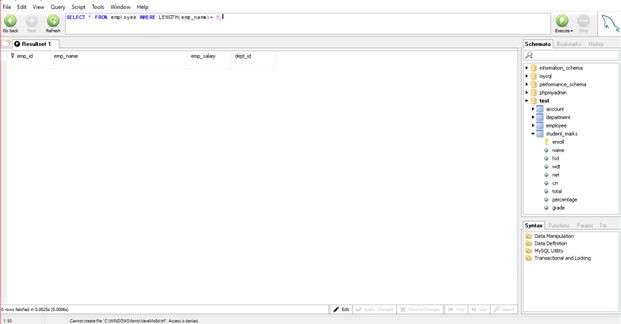
**Display employee whose name starts with “A”.**



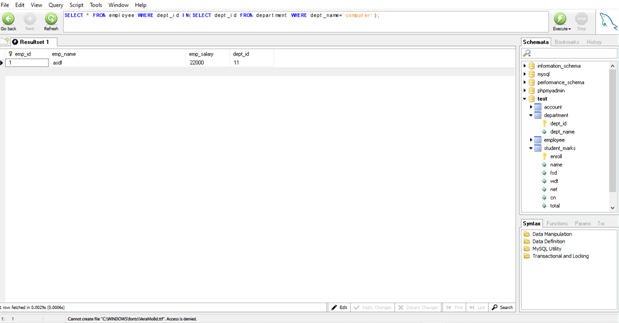
**Display employee whose name ends with “D”.**



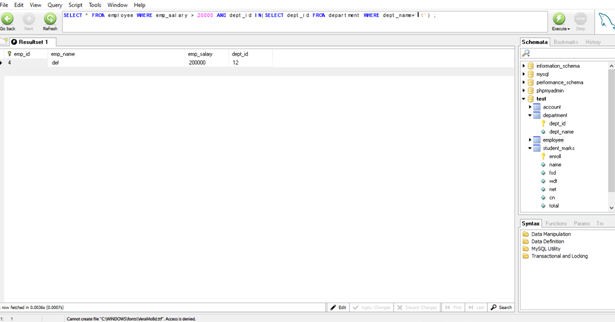
**Display the employee whose name length is 5.**



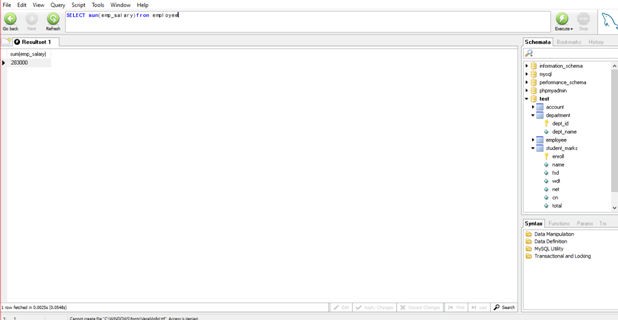
**Show employees from ‘computer’ department.**



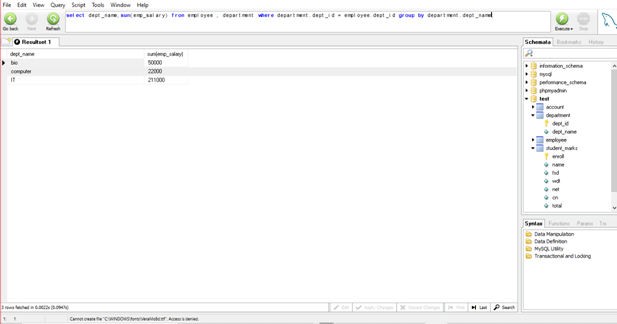
**Find all employees from IT department and salary greater than 20000.**



**List total salary of all the departments.**



**Get total salary of each department.**



**Give 10% increment to all employees.**



**Give 5% employees to all the employees whose total salary is greater than 50000**



**PRACTICAL – 5**

**AIM:** Install and Use of MySQL Administrator.

**Basic Information:**

The MySQL Administrator is a powerful graphical client tool designed to ease the administrator and monitoring of the MySQL database server.

MySQL Administrator helps with management of a MySQL server as well as carrying out a number of management tasks. The program can be used for administration both on a local server and over a network.

Once installed, the MYSQL administrator connects to the specified database server and allows tasks such as adding users, monitoring the server performance and server connections and the backup and restoration of database to be performed quickly and easily.

**Installation:**

**Step-1 This is the first screen of installation. Click next.**

**.**

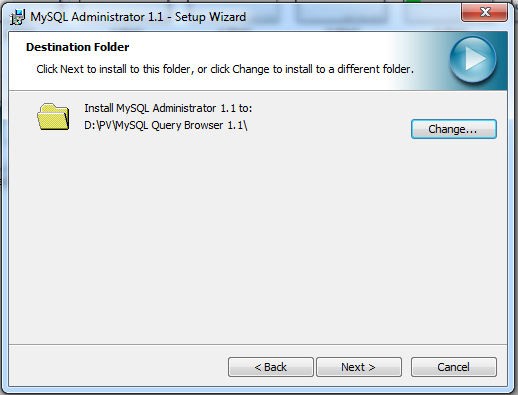


**Step-2 Select I accept the terms in license agreement and click next.**

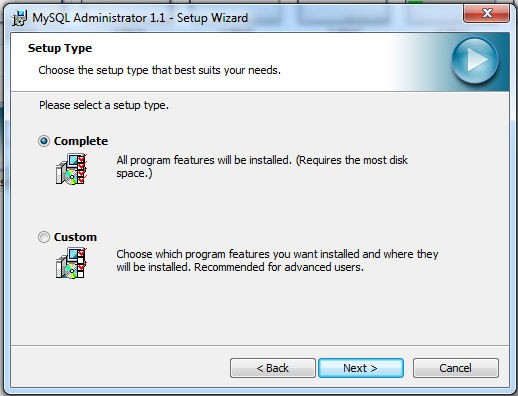


1

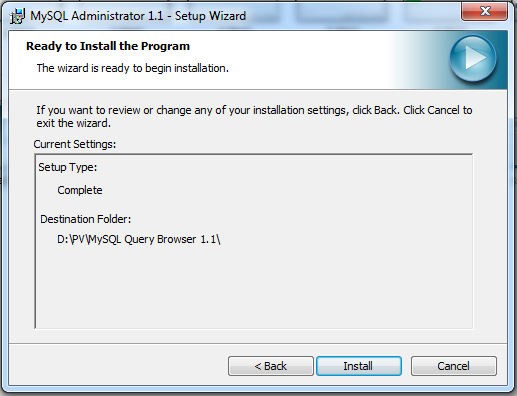
**Step-3 Select destination folder and click next.**



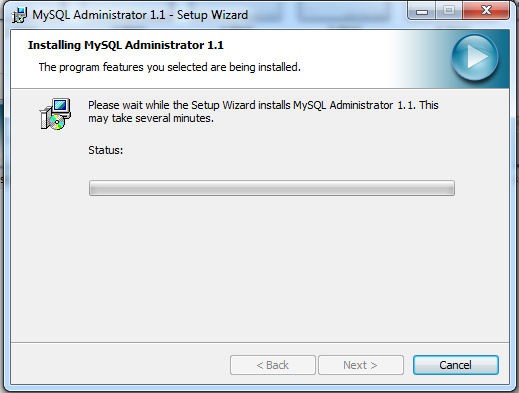
**Step-4 Select setup type as complete and click next.**



**Step-5 Click on install.**

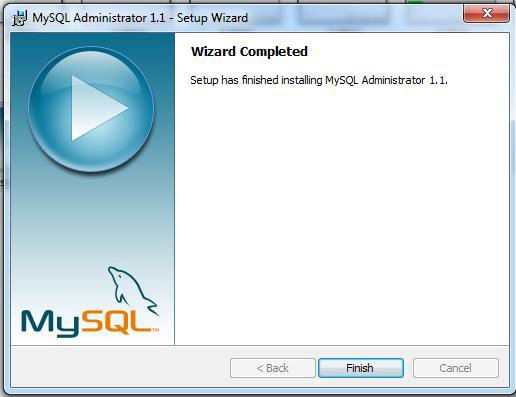


**Step-6 After clicking install this screen will appear which show the status.**



3

**Step-7 Click finish.**



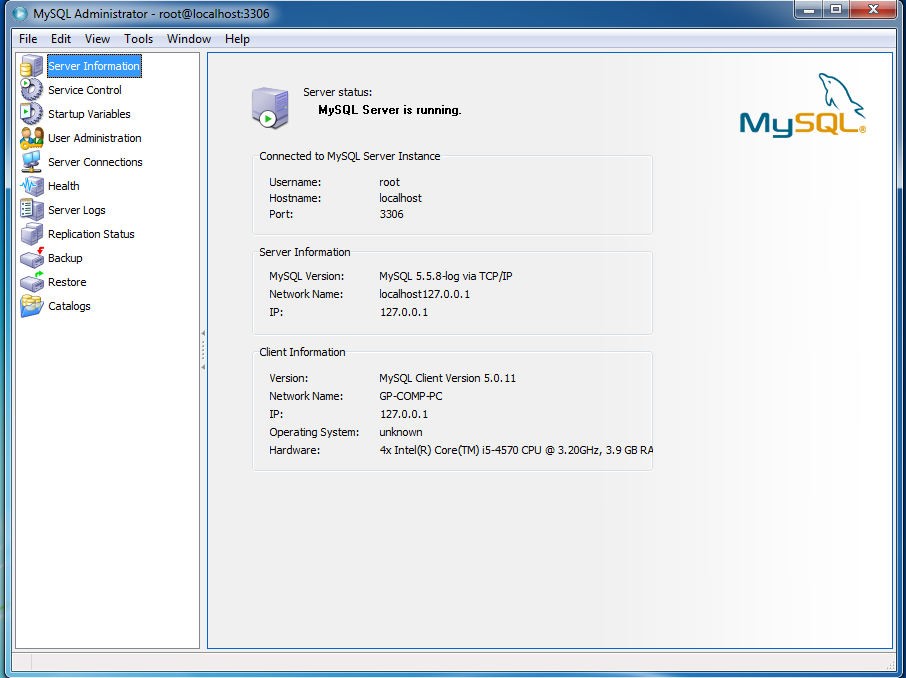
**Step-8 Enter stored connection, server host, username, password and click ok.**



4

**Server Information**

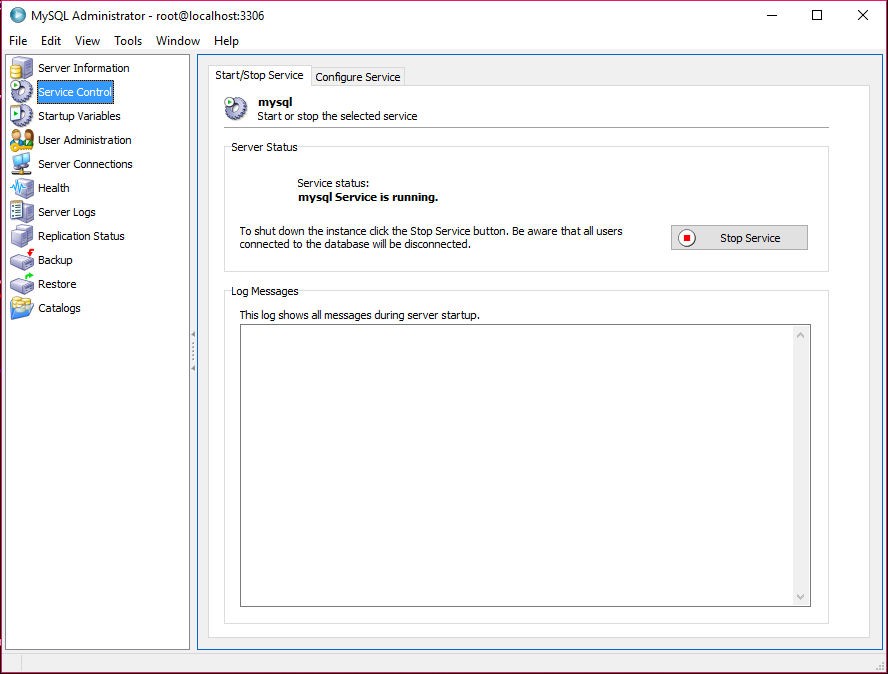
The server information screen shows basic server and client information, including status, version number, and host information.



**Service Control**

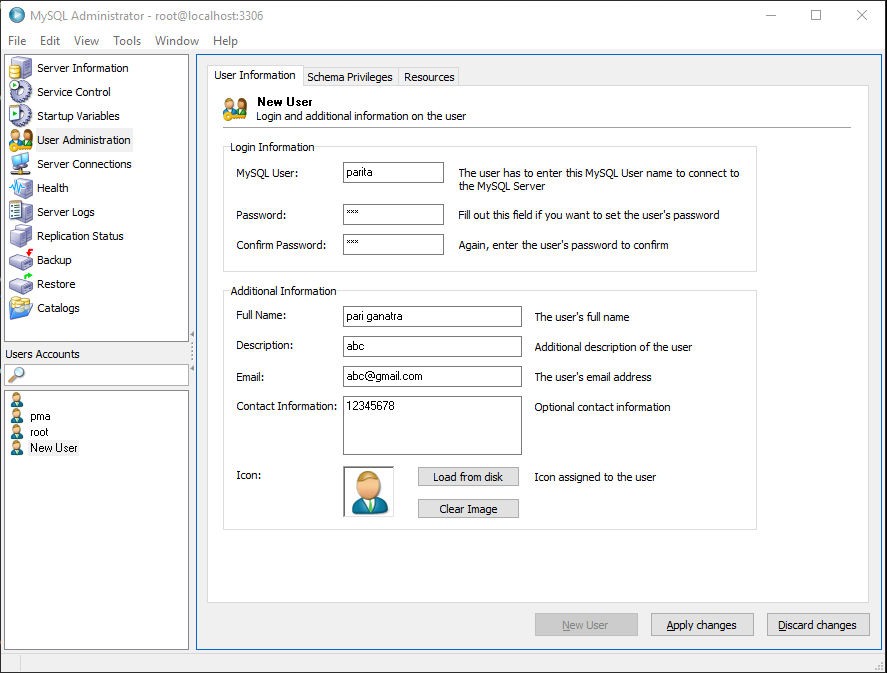
With this module we can start and stop the MySQL server. This works only for local MySQL servers (not for those running on another computer over a network) and only if MySQL Administrator was launched by someone with system administrator privileges.

.



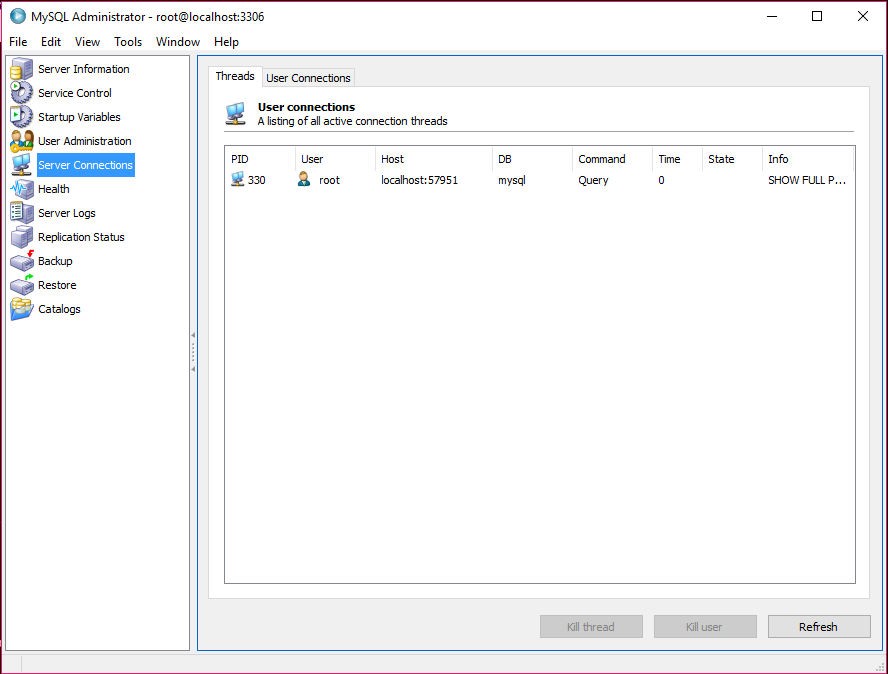
**User Administration**

This module helps in setting up new MySQL users and in changing the access privileges of existing users. The MySQL access system makes it possible to grant a user different access rights depending on the location from which the user establishes a connection to the MySQL server. Therefore, MySQL Administrator shows a list of permitted host names for each user.



**Server Connections**

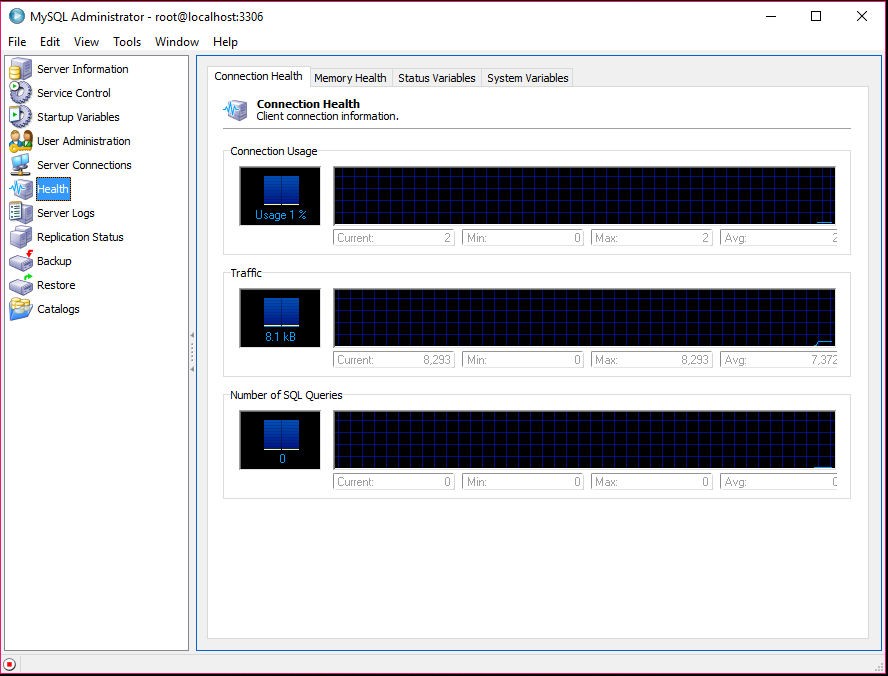
Click on server connection to see user connection information.



**Health monitoring**

The health monitoring screen gives you a view into the activity of your MySQL server. This provides both an immediate view of system activity, plus a more long-term view as it keeps

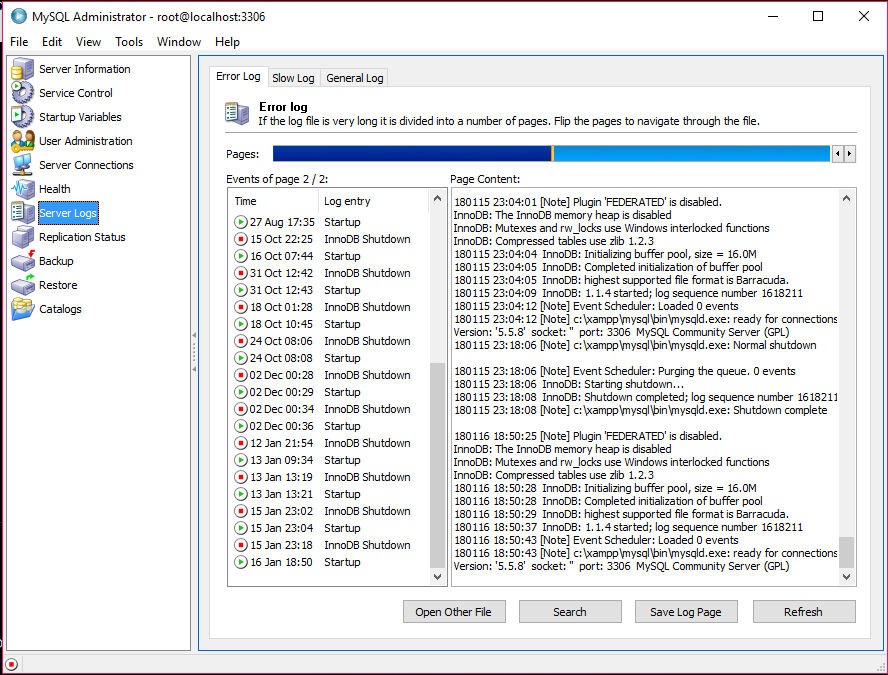
track of statistics as long the Administrator is running.



**Server log**

On the server log management screen, we can view all the log files — error logs, binary logs,

and slow query logs. This can help you when troubleshooting server issues.

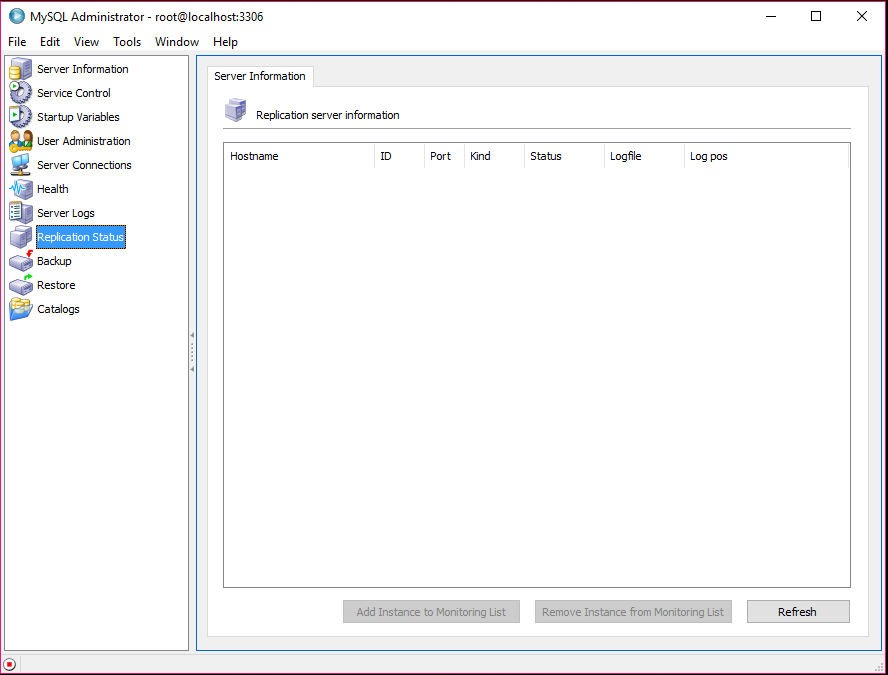


7

**Replication status**

The replication status screen allows us to view the network topology of our replication master

and slaves if the slaves are configured with the report-host option in the MySQL server configuration file and we are connected to the master server with the MySQL Administrator program.



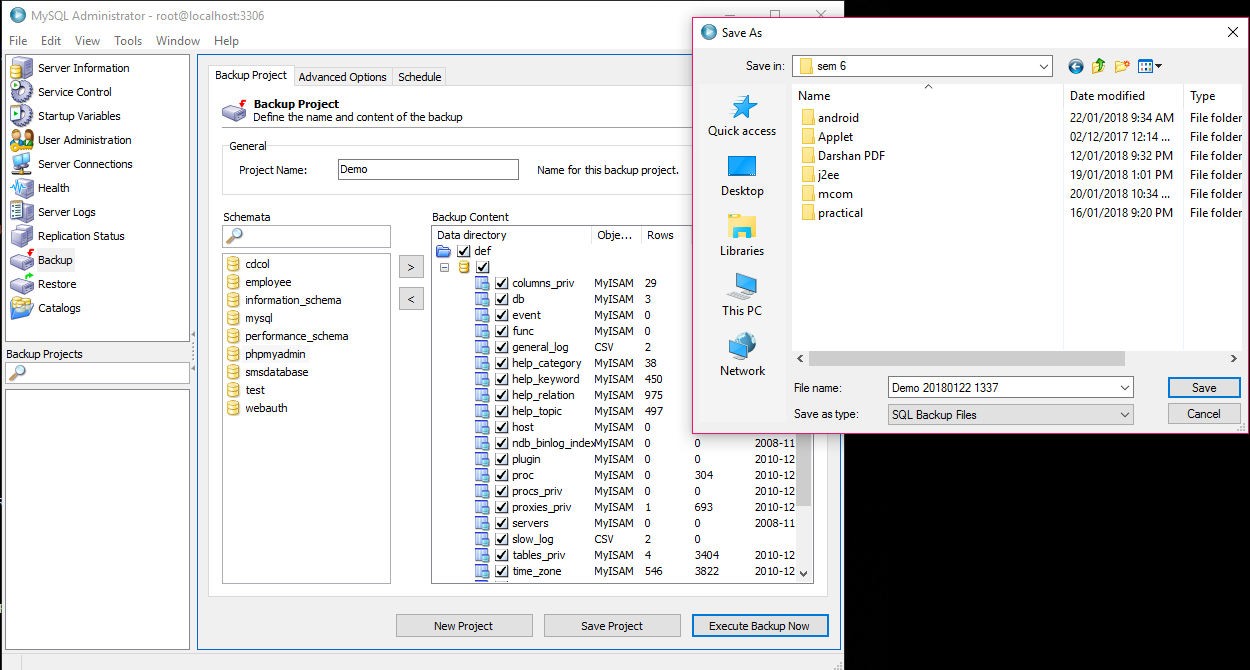
**Backup and recovery**

With the backup and recovery screen, we can create backups and restore data using a graphical

interface. You can choose to back up an entire database or one or more tables by clicking on

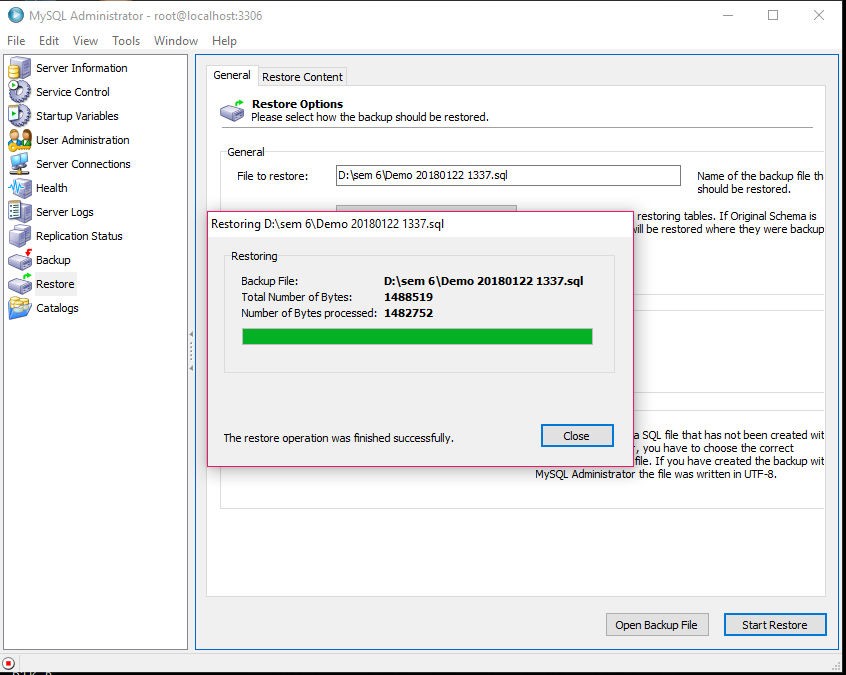
check boxes.

Enter project name and click on execute backup now. In save as screen dialog select folder as per your choice and click save.



**Restore**

To restore your backup, select the backup file and click on Start Restore.



**PRACTICAL – 6**

**AIM:** MySQL Trigger.

**Basic Information:**

A trigger is a set of actions that are run automatically when a specified change operation (SQL INSERT, UPDATE, or DELETE statement) is performed on a specified table. Triggers are useful for tasks such as enforcing business rules, validating input data, and keeping an audit trail.

The trigger is mostly used for maintaining the integrity of the information on the database. For example, when a new record is added to the employee table, new record should also be created in the tables of the taxes, vacation and salary. Trigger can also be used to log historical data.

Uses for triggers

:

1. Enforce business rules
2. Validate input data
3. Generate a unique value for a newly inserted row on a different file.
4. Write to other files for audit trail purposes
5. Query from other files for cross-referencing purposes
6. Access system functions
7. Replicate data to different files to achieve data consistency

**Syntax for creating triggers:**

CREATE

[DEFINER = {user | CURRENT\_USER}]

TRIGGER trigger\_name trigger\_time trigger\_event

ON tbl\_name FOR EACH ROW

trigger\_body

trigger\_time: {BEFORE | AFTER}

trigger\_event: {INSERT | UPDATE | DELETE}

**Explanation:**

**DEFINER clause**:

The DEFINER clause specifies the MySQL account to be used when checking access privileges at trigger activation time. If a user value is given, it should be a MySQL account specified as 'user\_name'@'host\_name' or CURRENT\_USER or CURRENT\_USER ().

The default DEFINER value is the user who executes the CREATE TRIGGER statement. This is the same as specifying DEFINER = CURRENT\_USER explicitly.

If we specify the DEFINER clause, these rules determine the valid DEFINER user values:

If we do not have the SUPER privilege, the only permitted user value is our own account, either specified literally or by using CURRENT\_USER. We cannot set the definer to some other account.

If we have the SUPER privilege, we can specify any syntactically valid account name. If the

account does not actually exist, a warning is generated.

**trigger\_name**:

All triggers must have unique names within a schema. Triggers in different schemas can have the same name.

**trigger\_time**:

trigger\_time is the trigger action time. It can be BEFORE or AFTER to indicate that the trigger activates before or after each row to be modified.

**trigger\_event**:

trigger\_event indicates the kind of operation that activates the trigger. These trigger\_event values are permitted:

INSERT: The trigger activates whenever a new row is inserted into the table.

UPDATE: The trigger activates whenever a row is modified.

DELETE: The trigger activates whenever a row is deleted from the table. DROP TABLE and TRUNCATE TABLE statements on the table do not activate this trigger, because they do not use DELETE. Dropping a partition does not activate DELETE triggers, either.

**tbl\_name**:

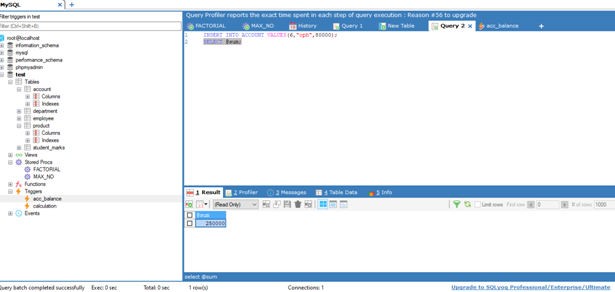
The trigger becomes associated with the table named tbl\_name, which must refer to a permanent table. We cannot associate a trigger with a TEMPORARY table or a view.

**trigger\_body**:

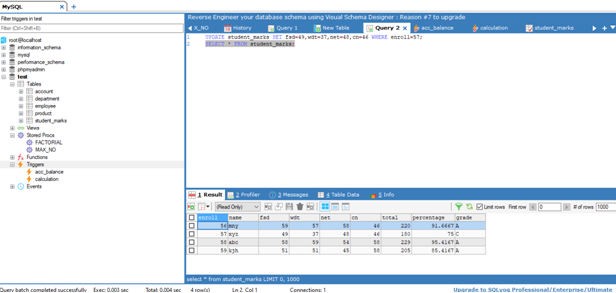
trigger\_body is the statement to execute when the trigger activates. To execute multiple statements, use the BEGIN ... END compound statement construct.

**EXERCISE:**

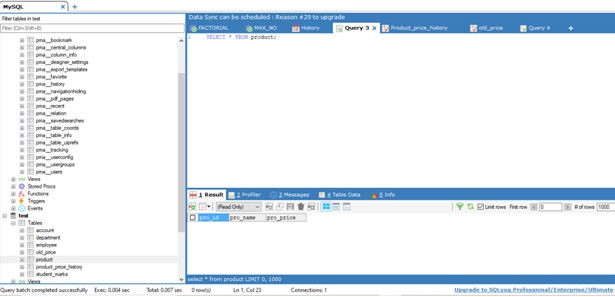
1. Create an account table and an INSERT trigger. The trigger sums the values inserted into balance column.



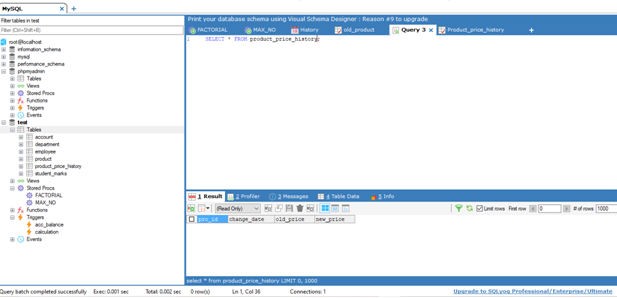
1. Create a table student\_marks(enrollment\_no, name, fsd, wdt, .net, cn, total, percentage, grade). Now the exam is over and we have received all subject marks which update the table. Create a trigger to calculate total marks of all subjects, percentage of total marks and grade.



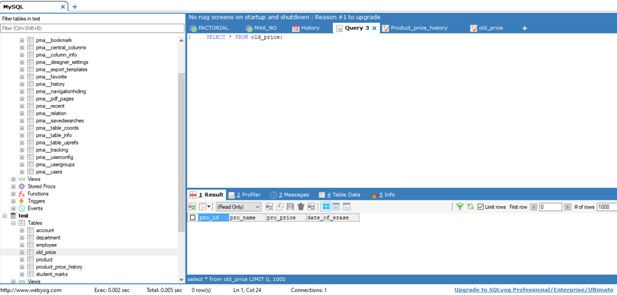
1. Create a product, product\_price\_history and old\_products table.



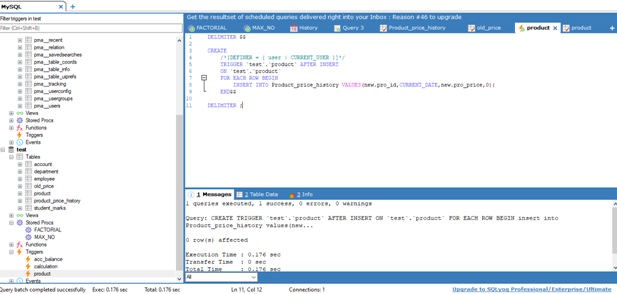
1. Create row level Trigger for product that would fire after INSERT operation performed on product table. Trigger will insert a new record in product\_price\_history table with details about product\_id, new\_price, old\_price and date\_of\_change. Old price should be zero.

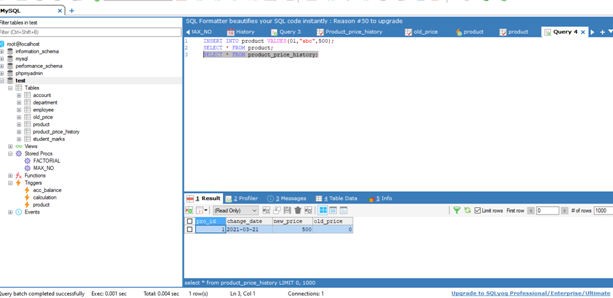


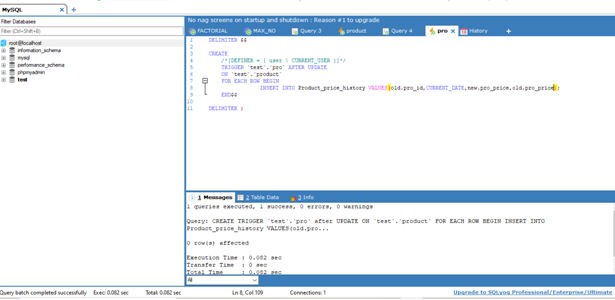
OLD PRICE:-

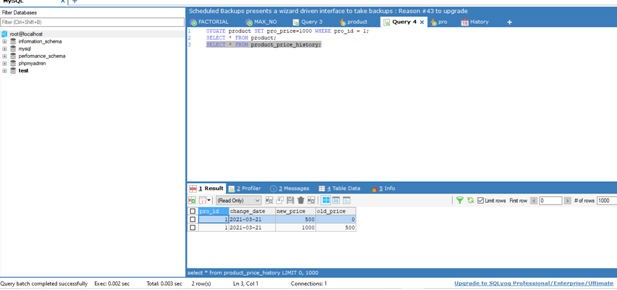


1. Create row level Trigger for product table that would fire after UPDATE operation performed on product table. Trigger will insert a new record in product\_price\_history table with details about product\_id, new price, old price and date of change.

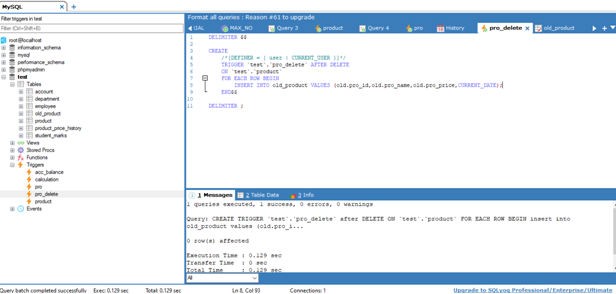


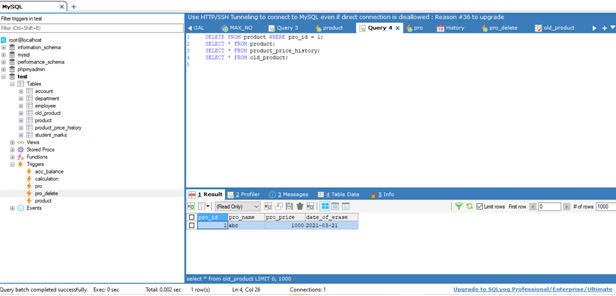




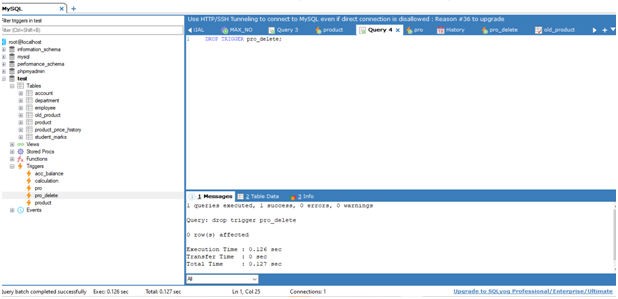


1. Create trigger for product table that would fire after DELETE operation performed on product table. Trigger will insert a new record in old\_products table with details about product\_id, product\_name and date\_of\_deletion.





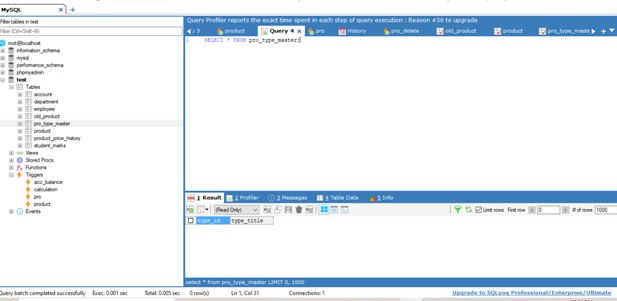
1. Write down query to drop a Trigger.



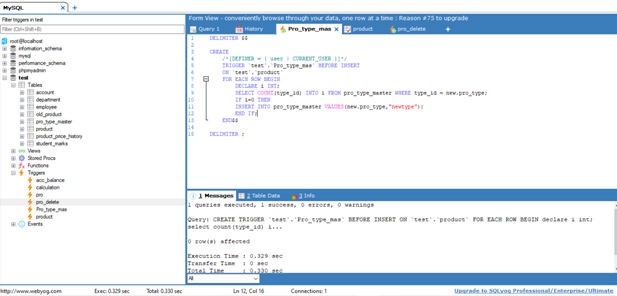
**PRACTICAL – 7**

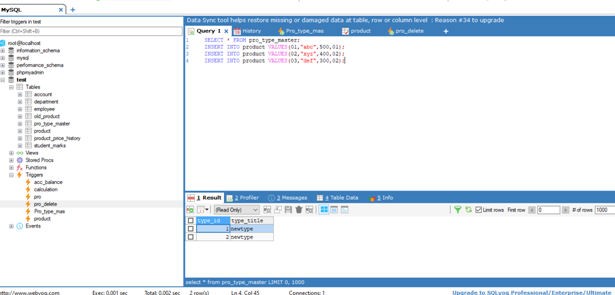
**AIM:** Implement Trigger.

1. Create product\_type\_master table.

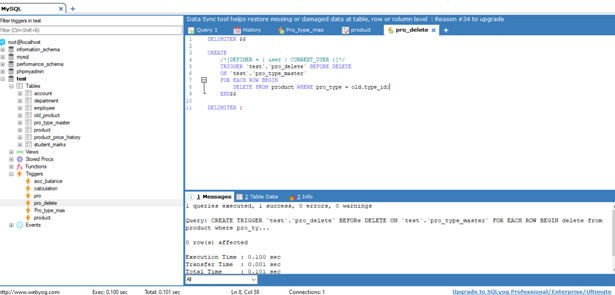


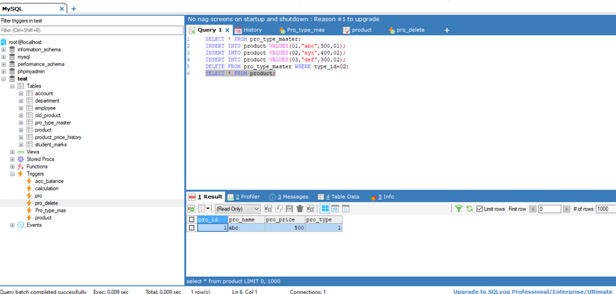
1. Creating Row level trigger for product table that would fire before Insert operation performed on product. This trigger will insert a new record in product\_type\_master table.



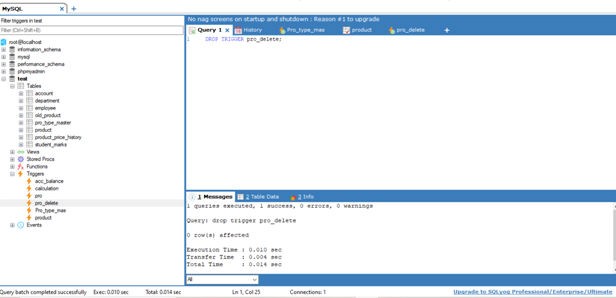


1. Create a trigger for deleting all the product of particular product type when product type in product\_type\_master is deleted.





1. Drop Trigger.



PROFESSIONALPRACTICESUSINGDATABASE(3360702) PRACTICAL 8,9

**PRACTICAL  8-9**

**AIM:** MySQL Stored Procedures (Routines).

**Basic Information:**

A stored procedure is a set of Structured Query Language (SQL) statements with an assigned name, which are stored in a relational database management system as a group, so it can be reused and shared by multiple programs.

A procedure has a name, a parameter list, and SQL statement(s). A stored procedure can be invoked by triggers, other stored procedures, and applications such as Java, Python, PHP.

**Syntax for creating procedure:**

DELIMITER $$

CREATE

    /\*[DEFINER = { user | CURRENT\_USER }]\*/

    PROCEDURE `database\_name`.`procedure\_name`()

    /\*LANGUAGE SQL

    | [NOT] DETERMINISTIC

    | { CONTAINS SQL | NO SQL | READS SQL DATA | MODIFIES SQL DATA }

    | SQL SECURITY { DEFINER | INVOKER }

    | COMMENT 'string'\*/

    BEGIN

END$$

The four characteristics of a procedure are:

1. Language: For portability purposes; the default value is SQL.
2. Deterministic: If the procedure always returns the same results, given the same input. This is for replication and logging purposes. The default value is NOT DETERMINISTIC.
3. SQL Security: At call time, check privileges of the user. INVOKER is the user who calls the procedure. DEFINER is the creator of the procedure. The default value is DEFINER.
4. Comment: For documentation purposes; the default value is "".

**Calling a Stored Procedure:**

CALL procedure\_name(parameter1,parameter2,…);

**Modify a Stored Procedure:**

MySQL provides an ALTER PROCEDURE statement to modify a routine, but only allows for the ability to change certain characteristics. If we need to alter the body or the parameters, we must drop and recreate the procedure.

**Delete a Stored Procedure:**

DROP PROCEDURE IF EXISTS procedure\_name;

**Procedure with Parameters:**

A procedure can contain three types of parameters: IN, OUT, INOUT

1. CREATE PROCEDURE `database\_name`.`procedure\_name`(IN varname DATA-TYPE): One input parameter. The word IN is optional because parameters are IN (input) by default.

2. CREATE PROCEDURE `database\_name`.`procedure\_name`(OUT varname DATA-TYPE): One output parameter.

3. CREATE PROCEDURE `database\_name`.`procedure\_name`(INOUT varname DATA-TYPE): One parameter which is both input and output.

**IN example:**

DELIMITER //

CREATE PROCEDURE `test`.`proc\_IN`(IN var1 INT)

BEGIN

    SELECT var1 + 2 AS result;

END//

Call proc\_IN(20);

**OUT example:**

DELIMITER //

CREATE PROCEDURE `test`.`proc\_OUT`(OUT var1 VARCHAR(100))

BEGIN

    SET var1 = 'This is a test';

END //

Call proc\_OUT(@test);

Select @test;

**INOUT example:**

DELIMITER //

CREATE PROCEDURE `test`.`proc\_INOUT`(INOUT var1 INT)

BEGIN

    SET var1 = var1 \* 2;

END //

Set @sum=3;

Call proc\_INOUT(@sum);

Select @sum;

**Variables:**

Declare a variable using the following syntax:

DECLARE varname DATA-TYPE DEFAULT defaultvalue;

**Working with variables:**

Once the variables have been declared, you can assign them values using the SET or SELECT command:

DELIMITER //

CREATE PROCEDURE `test`.`var\_proc`(IN paramstr VARCHAR(20))

BEGIN

    DECLARE a, b INT DEFAULT 5;

    DECLARE str VARCHAR(50);

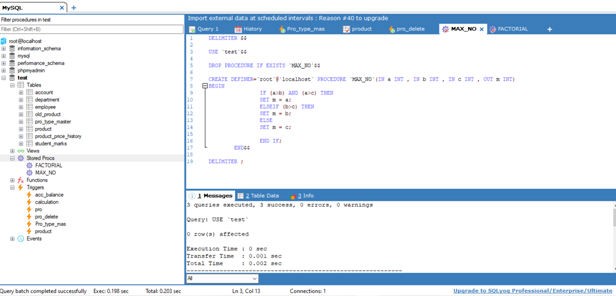
    DECLARE today TIMESTAMP DEFAULT CURRENT\_DATE;

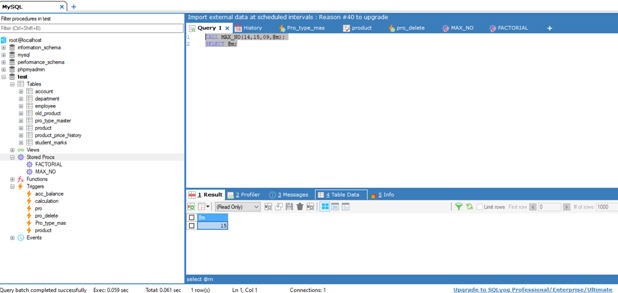
    INSERT INTO table1 VALUES(a);

    SET str = 'I am a string';

    SELECT CONCAT(str, paramstr), today FROM table2 WHERE b >=5;

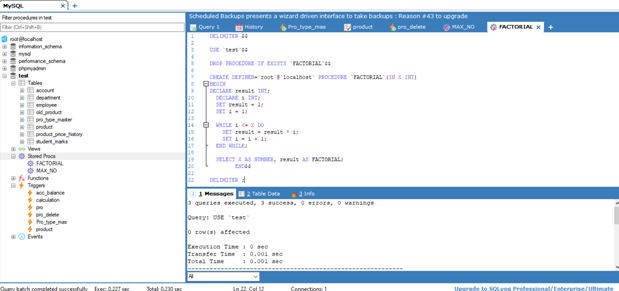
END //

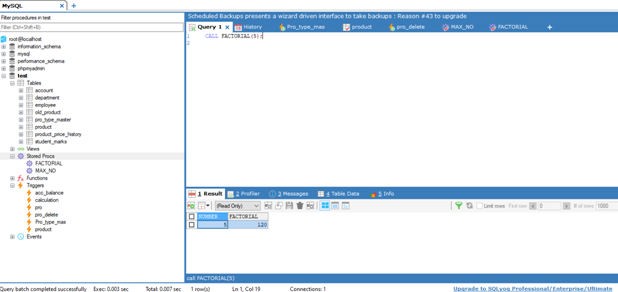




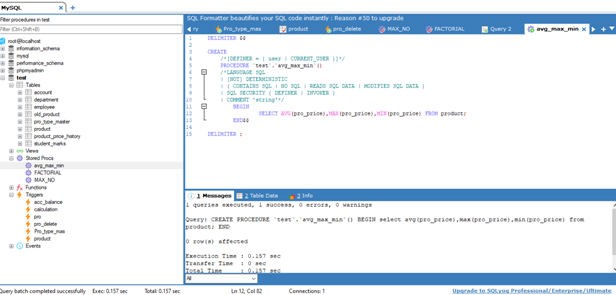
**EXERCISE:**

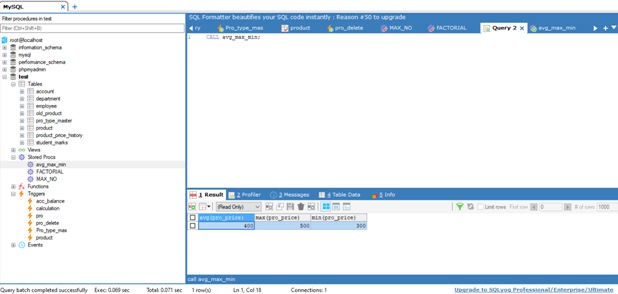
1. Create a stored procedure to find maximum of 3 numbers.



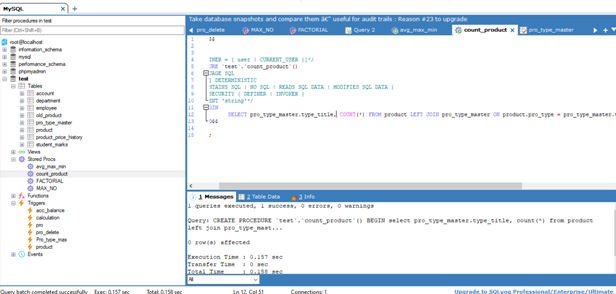


1. Create a stored procedure to calculate factorial of any number.

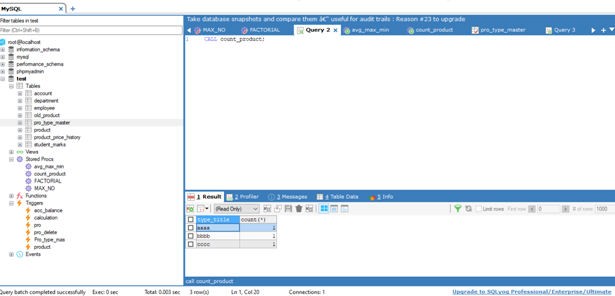




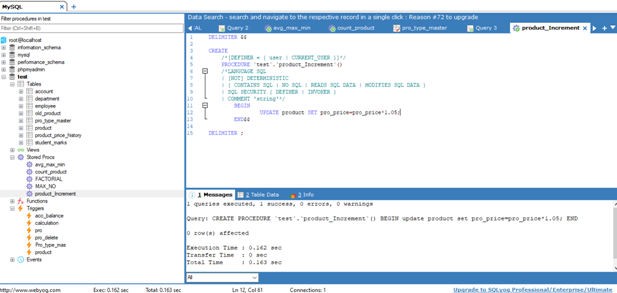
1. Create a procedure that returns average, minimum, maximum price of the products.

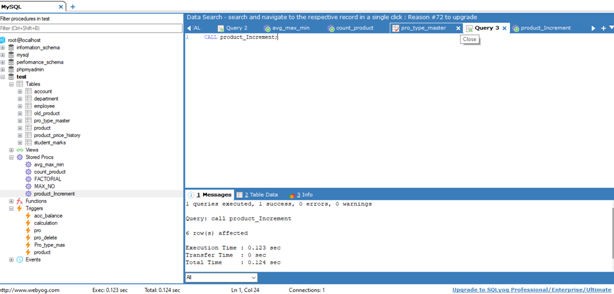


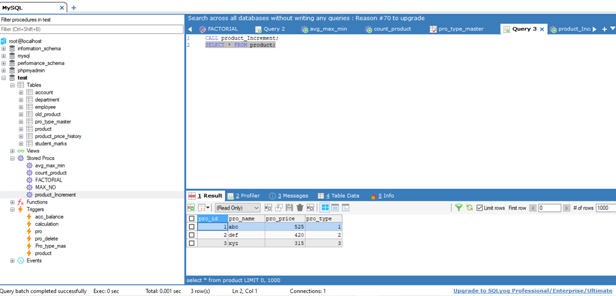
1. Write a stored routine for counting all product types. (Create a procedure that list out category wise count of the product.)



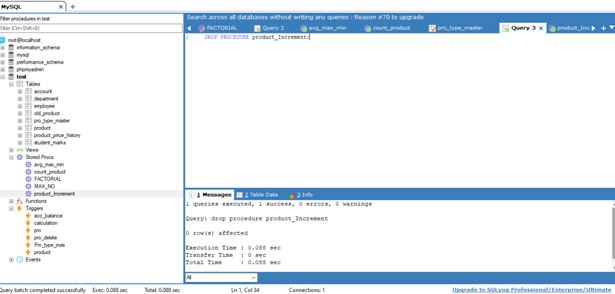
1. Write a routine for updating price of all product by 5%.







1. Delete stored routine.



**PRACTICAL  10**

**AIM:** Understanding and implementation of Cursors.

**Basic Information:**

To handle a result set inside a stored procedure, we use a cursor. A cursor allows us to iterate a set of rows returned by a query and process each row accordingly. A database cursor is a control structure that enables traversal over the record in a database.

MySQL cursor is read only, non-scrollable and asensitive.

1. Read only: Data cannot be updated through the cursor.
2. Non-scrollable: Rows can only be fetched in the order specified by the SELECT statement. Rows cannot be skipped, you cannot jump to a specific row, and you cannot fetch rows in reverse order.
3. Asensitive: There are two kinds of cursors: asensitive cursor and insensitive cursor. An asensitive cursor points to the actual data, whereas an insensitive cursor uses a temporary copy of the data. An asensitive cursor performs faster than an insensitive cursor because it does not have to make a temporary copy of data.

However, any change that made to the data from other connections will affect the data that is being used by an asensitive cursor, therefore it is safer if we don’t update the data that is being used by an asensitive cursor. MySQL cursor is asensitive. We can use MySQL cursors in stored procedures, stored functions and triggers.

**Working with MySQL cursor:**

1. Declare a cursor: The cursor declaration must be after any variable declaration. A cursor must always be associated with a SELECT statement.

DECLARE cursor\_name CURSOR FOR SELECT statement;

1. Open the cursor: The OPEN statement initializes the result set for the cursor therefore you must call the OPEN statement before fetching rows from the result set.

OPEN cursor\_name;

1. Fetch the data: FETCH statement is used to retrieve the row pointed by the cursor and move the cursor to the next row in the result set.

FETCH cursor\_name INTO variables list;

1. Close the cursor: Cursor is closed using CLOSE statement. It deactivates the cursor and release the memory associated with it.

CLOSE cursor\_name;

**NOT FOUND handler:**

Each time we call the FETCH statement, the cursor attempts to read the next row in the result set. When the cursor reaches the end of the result set, it will not be able to get the data, and a No Data condition occurs with SQLSTATE value '02000'. To detect this condition, the handler is set up to handle this condition (or for a NOT FOUND condition).

Syntax to declare NOT FOUND handler:

DECLARE CONTINUE HANDLER FOR NOT FOUND SET finished = 1;

Where finished is a variable to indicate that the cursor has reached the end of the result set. The handler declaration must appear after variable and cursor declaration inside the stored procedures.

**Example:**Insert student data into low\_attendance table.

  CREATE PROCEDURE `test`.`curdemo`()

  BEGIN

  DECLARE done INT DEFAULT FALSE;

  DECLARE stud\_id INT(20);   DECLARE stud\_nm VARCHAR(30) ;

DECLARE stud\_attendance CURSOR FOR SELECT sid, sname FROM student WHERE    lec\_attend <= 4;

  DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

  OPEN stud\_attendance;

  read\_loop: LOOP

    FETCH stud\_attendance INTO stud\_id, stud\_nm;

    IF done THEN

      LEAVE read\_loop;

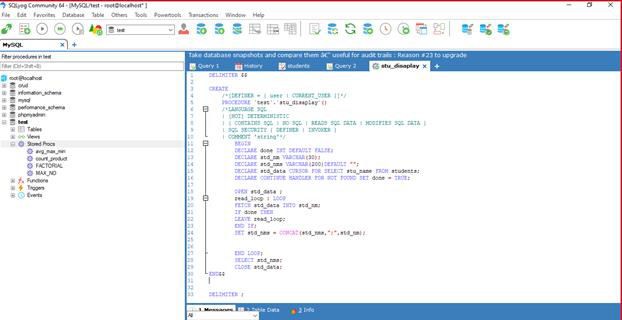
    END IF;

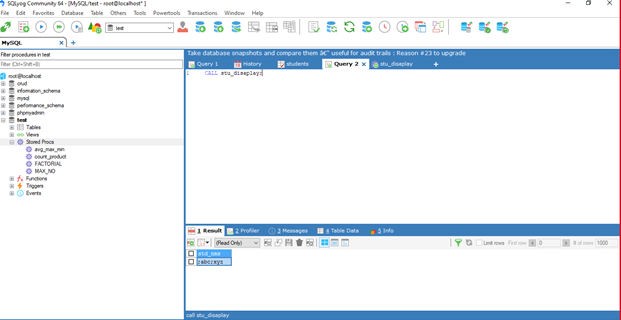
   INSERT INTO low\_attendance VALUES (stud\_id, stud\_nm);

   END LOOP;  CLOSE stud\_attendance; END$$

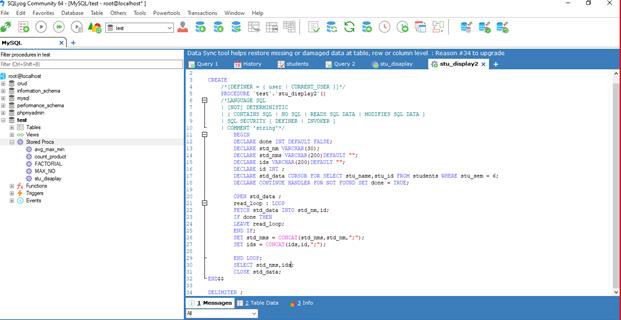
**EXERCISE:**

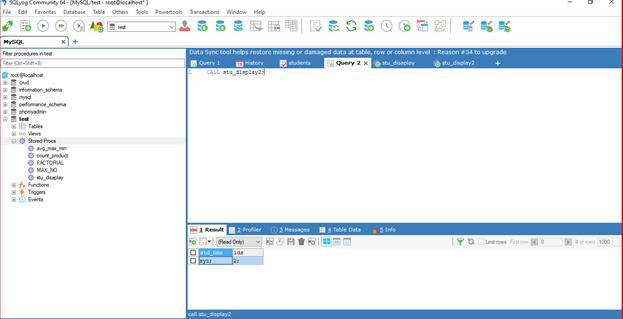
1. Create a cursor that builds name list of all students in the student table.



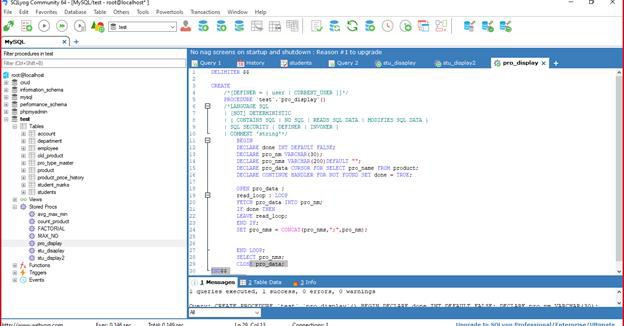


Create a cursor to select 6th semester students from student table.



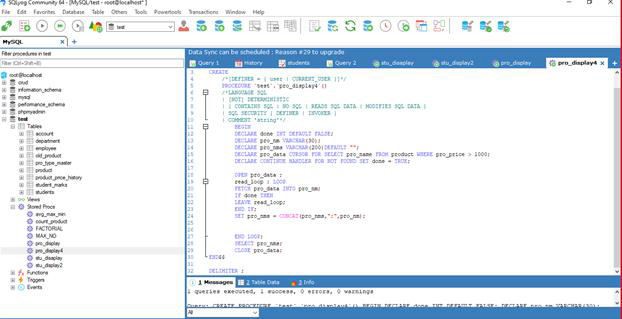


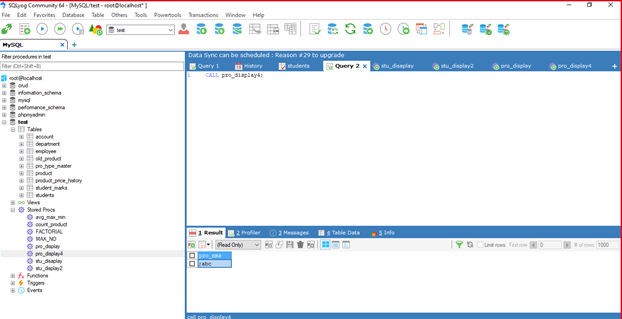
1. Create a cursor which selects all products from product table.



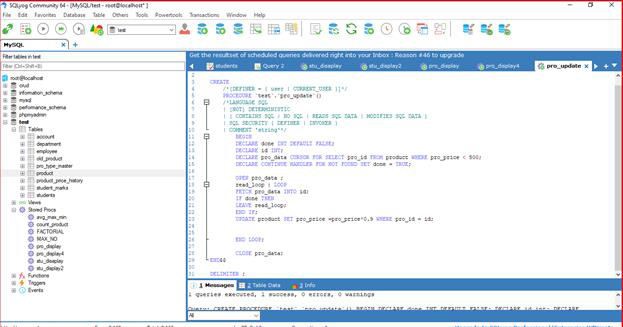


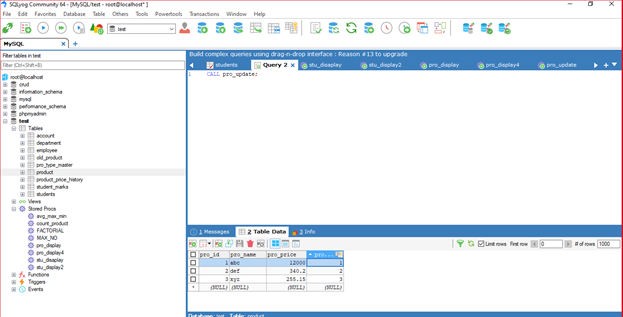
Create a cursor for selecting all product whose price is more than 1000.





Create a cursor that check price of all product and if price is less than 500 than give 10% discount on that product.





**PRACTICAL  11-12**

**AIM:** Understanding and implementation of Events

**Basic Information:**

MySQL Events are tasks that run according to a schedule. Therefore, we sometimes refer to them as scheduled events.

When you create an event, you are creating a named database object containing one or more SQL statements to be executed at one or more regular intervals, beginning and ending at a specific date and time.

Scheduled tasks of this type are also sometimes known as “temporal triggers”, implying that these are objects that are triggered by the passage of time.

**Syntax for creating event:**

CREATE

    [DEFINER = {user | CURRENT\_USER}]

    EVENT

    [IF NOT EXISTS]

    event\_name

    ON SCHEDULE schedule

    [ON COMPLETION [NOT] PRESERVE]

    [ENABLE | DISABLE | DISABLE ON SLAVE]

    [COMMENT 'comment']

    DO event\_body;

schedule:

    AT timestamp [+ INTERVAL interval] ...

    | EVERY interval

    [STARTS timestamp [+ INTERVAL interval] ...]

    [ENDS timestamp [+ INTERVAL interval] ...]

interval:

    quantity {YEAR | QUARTER | MONTH | DAY | HOUR | MINUTE |

              WEEK | SECOND | YEAR\_MONTH | DAY\_HOUR | DAY\_MINUTE |

              DAY\_SECOND | HOUR\_MINUTE | HOUR\_SECOND | MINUTE\_SECOND}

This statement creates and schedules a new event. The event will not run unless the Event Scheduler is enabled.

The minimum requirements for a valid CREATE EVENT statement are as follows:

* The keywords CREATE EVENT plus an event name, which uniquely identifies the event within a database schema.
* An ON SCHEDULE clause, which determines when and how often the event executes.
* A DO clause, which contains the SQL statement to be executed by an event.

**Example:**

CREATE EVENT myevent ON SCHEDULE AT CURRENT\_TIMESTAMP + INTERVAL 1 HOUR DO   UPDATE myschema.mytable SET mycol = mycol + 1;

The statement creates an event named myevent. This event executes once—one hour following its creation—by running an SQL statement that increments the value of the myschema.mytable table's mycol column by 1.

The event\_name must be a valid MySQL identifier with a maximum length of 64 characters. Event names are not case sensitive, so we cannot have two events named myevent and MyEvent in the same schema

**Drop the event:**

DROP EVENT mark\_insert;

**Suspend execution for the event:**

ALTER EVENT mark\_insert DISABLE;

**EXERCISE:**

1. Create an event that deletes rows that are more than three years old from student table.
2. Create an event that check the product having quantity less than 20 in stock at every ten minutes and if stock is less than 20 then insert the detail regarding that product into product\_def table.
3. Create an event that check the product which has been sold maximum in a day and insert the detail regarding this product into discount table.

**PRACTICAL  13**

**AIM:** Creating and Managing User Accounts In MySQL.

**My SQL Create User:**

In MySQL, you can specify not only who can connect to a database server but also from where. Therefore, an account in MySQL consists of username and host name where user connects from, that separated by @. For example, if the root user connects from the mysqltutorial.org host to the database server the account name would be [root@mysqlavpti.org](mailto:root@mysqlavpti.org).

This allows you to setup multiple accounts with the same name but connects from different hosts and have different privileges. The username and host are stored in the table named user.

Two ways to create new user:

1. Using CREATE USER statement

Syntax: CREATE USER user IDENTIFIED BY password;

User account in the format 'username'@'hostname' is followed by the CREATE USER.

The password is specified after the IDENTIFIED BY clause. The password must be in plain text. MySQL will encrypt the password for you when it saves the user into the user table.

Example: CREATE USER 'demouser'@'localhost' IDENTIFIED BY 'Demouser';

To allow user to connect from any host you use the % wildcard, which means any host.

CREATE USER superadmin@'%' IDENTIFIED BY 'Secured'

You can also use the underscrore wildcard \_  in the CREATE USER statement.

If you omit the hostname part of the account, MySQL still accepts it and allows user to connect from any host. The quote is very important especially when the account contains special characters such as- or %.

1. Using INSERT statement

A less common way to create user is using INSERT statement to insert a new record into the grant table named user. You need to use the PASSWORD function to encrypt the password before inserting the user record into the table.

Example: INSERT INTO user (host, user, password) VALUES('localhost',' demouser', PASSWORD('Demouser’));

**MySQL Changing Password:**

Three ways to change password.

1. Using UPDATE statement

The first way to change the password is to use the UPDATE statement to update the User and Host column of the user table in the mysql database.

Suppose you want to change the password for user named demouser that connect from host the localhost to Secret0406, you need to execute the following statements:

USE mysql;

UPDATE USER SET PASSWORD = PASSWORD('Secret0406') WHERE USER = 'demouser' AND HOST = 'localhost';

FLUSH PRIVILEGES;

After executing the UPDATE statement, you need to execute the FLUSH PRIVILEGES statement to reload privileges from the grant table in the mysql database.

1. Using SET PASSWORD statement

Here, we use the MySQL account in user@host format to update the password. If you need to change password for other accounts, you need to have at least UPDATE privilege. You don’t need to execute the FLUSH PRVILILEGES statement to reload privileges from grant table.

The following statement change password of demouser account by using the SET PASSWORD statement.

SET PASSWORD FOR 'demouser'@'localhost'=PASSWORD('Secret0406');

1. Using GRANT USAGE statement

The third way to change the password is by using the GRANT USAGE statement with the IDENTIFIED BY clause. In this way, specify the password in plain text instead of the encrypted from.

The following statement changes the password of mysqltutorial account by using the GRANT USAGE statement:

GRANT USAGE ON \*.\* TO 'demouser'@'localhost' IDENTIFIED BY 'Secret0406';

**MySQL Rename the user:**

Syntax: RENAME USER old name TO new name;

Example: RENAME USER me@localhost TO demo@localhost;

**MySQL Delete a user:**

Syntax: DROP USER username@hostname;

Example: DROP USER demo@localhost;

**PRACTICAL  14**

**AIM:** Creating and Managing User Accounts in MySQL.

**MySQL GRANT Statement Syntax:**

MySQL provides you with the MySQL GRANTstatement that allows you to grant access privileges to database accounts.

|  |  |
| --- | --- |
|  | **GRANT privileges [(column\_list)]**  **ON [object\_type] privilege\_level**  **TO account [IDENTIFIED BY 'password']**  **[REQUIRE encryption]**  **WITH with\_options** |

* **privileges** indicate the privileges that you assign to the account. For example, the CREATE privilege allows an account to [create databases](http://www.mysqltutorial.org/mysql-create-drop-database.aspx) and [create tables](http://www.mysqltutorial.org/mysql-create-table/). You can grant multiple privileges using single GRANT statement; the privileges are separated by commas.
* **column\_list** specifies the columns to which a privilege applies. The columns are separated by commas and listed within parentheses. The column list is optional element.
* **privilege\_level** specifies the level at which the privileges apply. You can use global privileges, database-specific privileges, table-specific privileges, column-specific privileges, etc.
* **account** specifies which account is being granted the privileges.
* **password** specifies the password to assign to the account*.* If the account exists, the GRANT statement replaces the old password by the new one. The IDENTIFIED BY clause is optional.
* After the **REQUIRE** clause, you specify whether the account has to connect to the database server over secure connection using SSL.
* If you want the account to have the privilege that can grant its own privileges to other accounts, you need to use the WITH clause with GRANT OPTION clause. In addition, you can use the WITH clause to allocate My SQL database server’s resource e.g., to set how many connections or statements that an account can use per hour. This is very helpful in shared environments such as My SQL shared hosting.
* If the account that you specify in the **GRANT** statement after the TO clause already exists, the GRANT statement modifies its privileges, otherwise, the GRANT statement creates a new account with the specified privileges.

**MySQL GRANT examples:**

If you want to create a super account that can do anything including being able to grant privileges to other users, you can use the following statements:

|  |  |
| --- | --- |
|  | **CREATE USER 'super'@'localhost' IDENTIFIED BY 'SecurePass1';**  **GRANT ALL ON \*.\* TO 'super'@'localhost' WITH GRANT OPTION;** |

Here, ON \*.\* clause means all databases and all objects in the databases. The only limitation of the super user is that it can only connect to the database server from the local host, which makes the My SQL server more secure.

You can grant multiple privileges using a single GRANT statement. For example, you can create a user that can execute the [SELECT](http://www.mysqltutorial.org/mysql-select-statement-query-data.aspx), [INSERT](http://www.mysqltutorial.org/mysql-insert-statement.aspx)and [UPDATE](http://www.mysqltutorial.org/mysql-update-data.aspx)statements on the test database using the following statements:

|  |  |
| --- | --- |
|  | **CREATE USER 'dbuser'@'localhost' IDENTIFIED BY 'Secret0406';**  **GRANT SELECT, UPDATE, DELETE ON test.\* TO 'dbuser'@'localhost';** |

**MySQL REVOKE Syntax:**

In order to revoke privileges from an account, you use the MySQL REVOKE statement.

|  |  |
| --- | --- |
|  | **REVOKE   privileges [(column\_list)]  [, priv\_type [(column\_list)]]...**  **ON [object\_type] privilege\_level**  **FROM user [, user]...** |

* You specify a list of privileges that you want to revoke from an account right after the REVOKE keyword. You need to separate privileges by comma.
* ON clause specifies the privilege level means privileges those are to be revoked.
* After FROM keyword, you specify the account that you want to revoke the privileges.
* You can specify multiple accounts in the FROM clause. You separate the accounts by comma.

In order to revoke privileges from an account, you must have GRANT OPTION privilege and privileges that you are revoking. To revoke all privileges, you use the following MySQL REVOKE syntax:

|  |  |
| --- | --- |
|  | **REVOKE ALL PRIVILEGES, GRANT OPTION FROM user [, user]…** |

To execute the above command, you must have the global CREATE USER privilege or the UPDATE privilege for the *mysql* database.

To revoke proxy user, you use the REVOKE PROXY command as follows:

|  |  |
| --- | --- |
|  | **REVOKE PROXY ON user FROM user [, user]...** |

A proxy user is a valid user in My SQL who can impersonate as another user therefore the proxy user has all privileges of the user that it impersonates.

Before revoking privileges of a user, it is good practice to check if the user has the privileges by using the SHOW GRANTS statement as follows:

|  |  |
| --- | --- |
|  | **SHOW GRANTS FOR user;** |

**MySQL REVOKE examples:**

Suppose *dbuser* account has privileges SELECT, UPDATE and DELETE on the *test*database. If you want to revoke UPDATE and DELETE privileges from the *dbuser* account, you can do so as follows:

First, you check the privileges of *dbuser* account using SHOW GRANTS statement:

|  |  |
| --- | --- |
|  | **SHOW GRANTS FOR 'dbuser'@'localhost';** |
|  | **GRANT SELECT, UPDATE, DELETE ON 'test'.\* TO 'dbuser'@'localhost'** |

Second, you can only revoke the UPDATE and DELETE privileges from the *dbuser* account:

|  |  |
| --- | --- |
|  | **REVOKE UPDATE, DELETE ON 'test'.\*  FROM 'dbuser'@'localhost';** |

Third, you can check the privileges of the *dbuser* account again using the SHOW GRANTS command

|  |  |
| --- | --- |
|  | **SHOW GRANTS FOR 'dbuser'@'localhost';** |

If you want to revoke all privileges of the *dbuser* account, use following command:

|  |  |
| --- | --- |
|  | **REVOKE ALL PRIVILEGES, GRANT OPTION FROM 'dbuser'@'localhost';** |

If you check the privileges of the *dbuser* account again, you will see the *dbuser* account has no privilege.

|  |  |
| --- | --- |
|  | **SHOW GRANTS FOR 'dbuser'@'localhost';** |
|  | **GRANT USAGE ON \*.\* TO 'dbuser'@'localhost'** |

**PRACTICAL  15**

**AIM:** Practicing with Database Backup and Recovery Operations.

**Backup Databases Using mysqldump Tool**

MySQL GUI tools such as php My Admin, SQLyog and etc often provide features for backup MySQL databases with ease. However, if your database is big, the backup process could be very slow because the backup file needs to be transferred across the network to your client PC. As the result, the backup process increases locking time therefore MySQL unavailability.

MySQL provides a very useful tool for backup or dump MySQL databases locally on server very fast. The backup file is stored in the file system in the server so you just need to download it when needed. The tool to backup MySQL databases is mysqldump. It is located in the root/bin folder of MySQL installation folder. The mysqldump is a program provided by MySQL that can be used to dump databases for backup or transfer database to another database server. The dump file contains a set of SQL statements to create and populate tables. In addition, the mysqldump can be used to generate CSV, delimited or XML files. In this practical, we will focus only on how to backup MySQL database by using mysqldump tool.

**Back up a MySQL Database**

To back up a MySQL database, the database first has to exist in the database server and you have access to that server as well. You can use SSH or Telnet to login to the remote server if you do not have remote desktop to it. The command to back up a MySQL database as follows:

|  |  |
| --- | --- |
|  | **mysqldump -u [username] –p [password] [database\_name] > [dump\_file.sql]** |

The parameter of the command above as follows:

* [username]: valid MySQL username.
* [password]: valid password for the user. Note that there is no space between –p and the password.
* [database\_name]: database name you want to backup.
* [dump\_file.sql]: dump file you want to generate.

By executing the above command all database structure and data will be exported into a single [dump\_file.sql] dump file. For example, in order to back our sample database test, we use the following

command:

|  |  |
| --- | --- |
|  | **mysqldump -u dbuser –p Secret0406 test > c:\temp\backup001.sql** |

**Backup MySQL Database Structure Only**

If you only want to backup database structure only you just need to add an option –no-data to tell mysqldump that only database structure needs to export as follows:

|  |  |
| --- | --- |
|  | **mysqldump -u [username] –p [password] –no-data [database\_name] > dump\_file.sql]** |

For example, to back up test database with structure only, you use the following command

:

|  |  |
| --- | --- |
|  | **mysqldump -u dbuser –p Secret0406 -no-data test > c:\temp\backup002.sql** |

**Backup MySQL Database Data Only**

There is a case that you want to refresh data in staging and development system so the data in those systems are the same as production system. In this case you just need to export data only from production system and import it to staging and development system. In order to backup data only, you use option –no-create-info of mysqldump as follows:

|  |  |
| --- | --- |
|  | **mysqldump -u [username] –p [password] –no-create-info [database\_name] > [dump\_file.sql]** |

For example, to back up our sample database with data only, you use the following command:

|  |  |
| --- | --- |
|  | **mysqldump –u dbuser –p Secret –no-create-info test > c:\temp\backup003.sql** |

**Backup Multiple MySQL Databases into a Single File**

If you want to backup multiple databases just separate database name by command in the [database\_name]. If you want to back up all databases in the database server, use the option **–all-**

**database.**

|  |  |
| --- | --- |
|  | **mysqldump -u [username] –p [password]  [dbname1, dbname2,…>[dump\_file.sql]**    **mysqldump -u [username] –p [password] –all-database > [dump\_file.sql]** |