CSE 5720-60
Summer 2024
Project 1- MySQL

Acknowledgment: The project was originally designed by Dr. Carey with slight modifications tailored for MySQL.
In this project, we are going to create a database and tables using MySQL and import the data into tables. Then you are required to form SQL queries for problem statements shown in step 5. Due by 11:59PM on **June, 18 (Tuesday)**. Turn in the 2 deliverables to folder "Project 1" on Canvas.
Deliverables
1. An SQL script that contains the queries listed in the same order as shown in step 5.
2. .csv files that contain the results obtained.

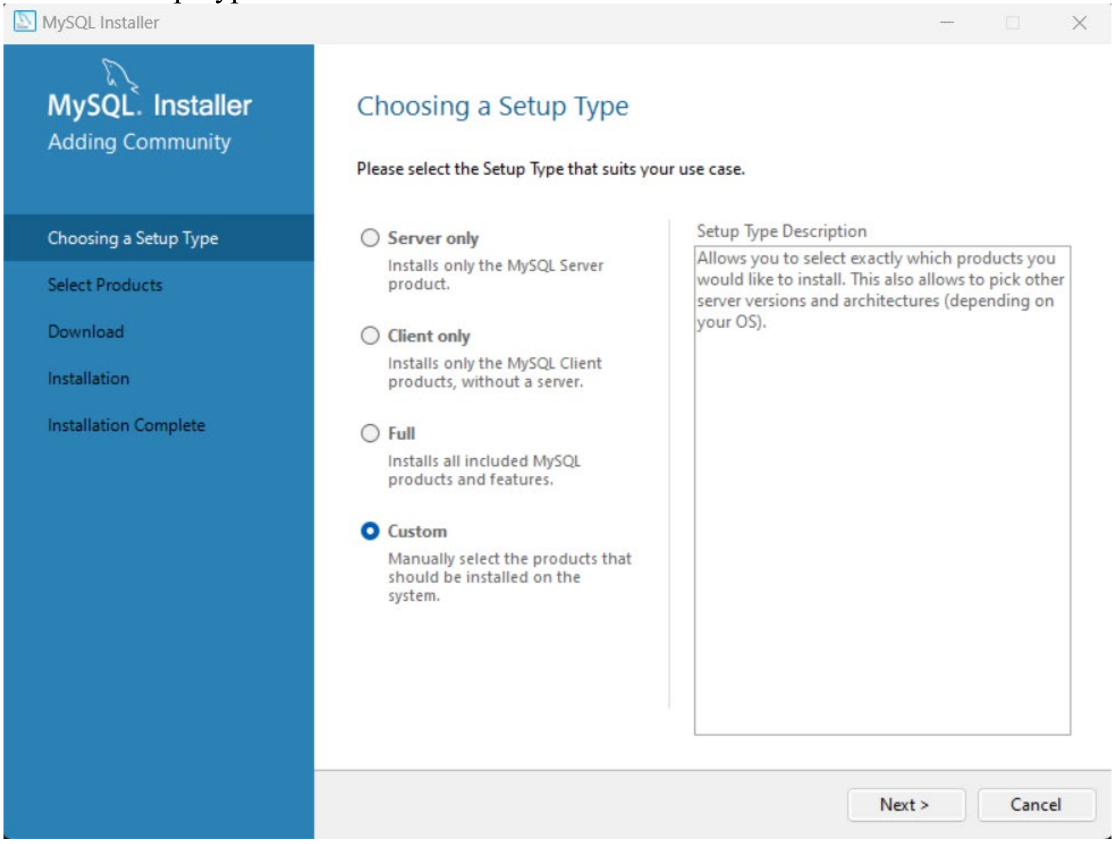
STEP 1 - Install MySQL

Go to MySQL download page (http://dev.mysql.com/downloads/) to download MySQL Installer for Windows. Select the correct installer based on your system.

For Mac OS X, download the DMG files, extract it, and install by the order: mysql-8.0.xx-osx10.6-x86_64.pkg, MySQLStartupItem.pkg, MySQL.prePane

The following description is based on MySQL server 8.0 for Windows.

1. Execute the downloaded file and complete the installation, then it will ask you to choose a setup Type. **Select 'Custom.'**



Choosing a Setup Type

Please select the Setup Type that suits your use case.

( ) Server only
Installs only the MySQL Server product.

( ) Client only
Installs only the MySQL Client products, without a server.

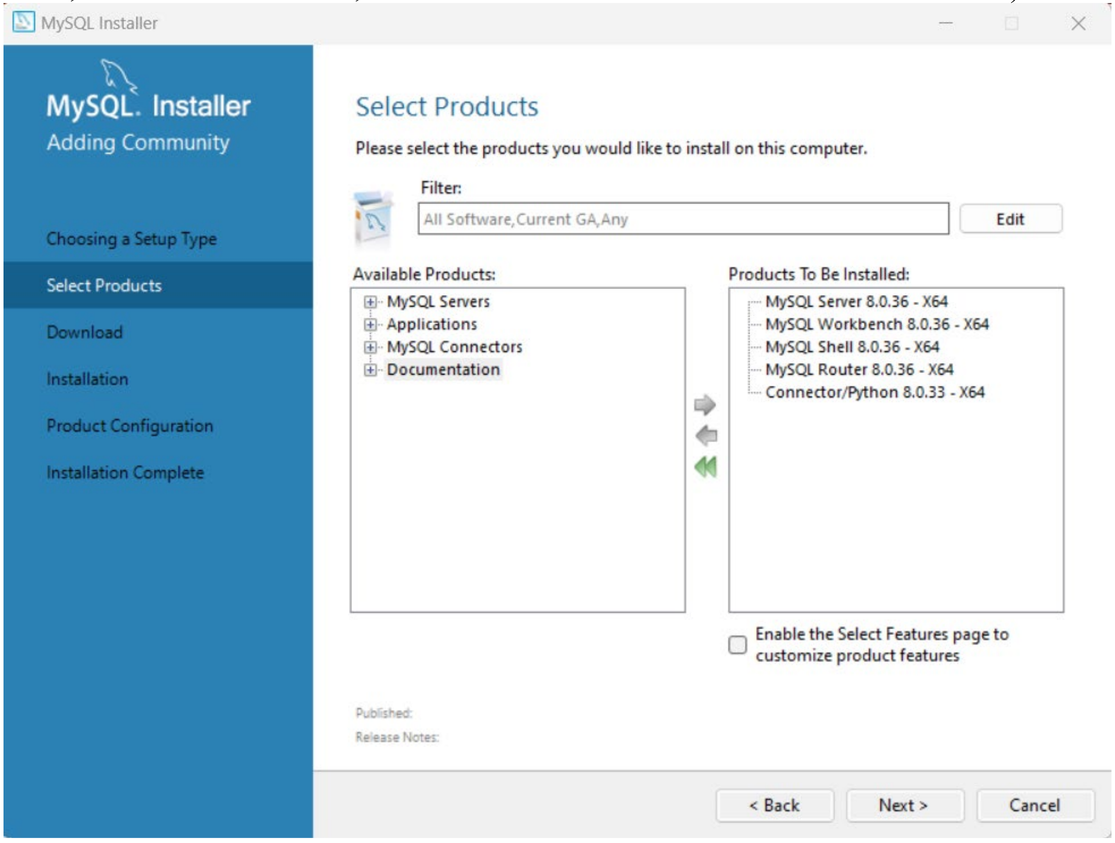
( ) Full
Installs all included MySQL products and features.

(●) Custom
Manually select the products that should be installed on the system.

Setup Type Description
Allows you to select exactly which products you would like to install. This also allows to pick other server versions and architectures (depending on your OS).

2. Add the following product to the install list: MySQL Server 8.0.x, MySQL Workbench 8.0.x, MySQL Shell 8.0.x, MySQL Router 8.0.x, Connector/Python

8.0.x, where x indicates the latest available version.
(NOTE: The Python Connector for MySQL 8.0 requires you to have Python version 3.11, 3.10, 3.9, 3.8, or 3.7 installed on your device. If you have none of these, or a different version, either install or reinstall one of the above versions)


MySQL Installer

MySQL. Installer
Adding Community

Choosing a Setup Type
Select Products
Download
Installation
Product Configuration
Installation Complete

Select Products

Please select the products you would like to install on this computer.

Filter: All Software,Current GA,Any

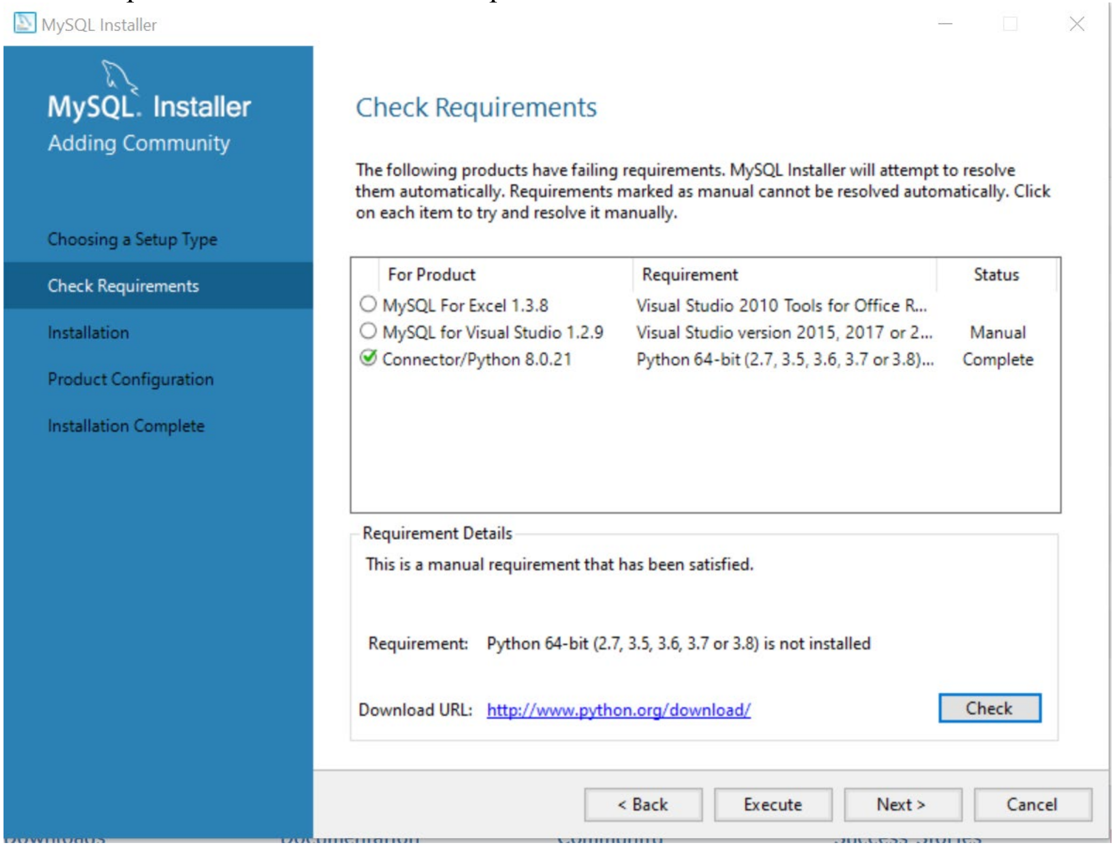
Available Products:
- MySQL Servers
- Applications
- MySQL Connectors
- Documentation

Products To Be Installed:
- MySQL Server 8.0.36 - X64
- MySQL Workbench 8.0.36 - X64
- MySQL Shell 8.0.36 - X64
- MySQL Router 8.0.36 - X64
- Connector/Python 8.0.33 - X64

☐ Enable the Select Features page to customize product features

Published:
Release Notes:

3. Check requirements and install the required software.


MySQL Installer

MySQL. Installer
Adding Community

Choosing a Setup Type
Check Requirements
Installation
Product Configuration
Installation Complete

Check Requirements

The following products have failing requirements. MySQL Installer will attempt to resolve them automatically. Requirements marked as manual cannot be resolved automatically. Click on each item to try and resolve it manually.

For Product	Requirement	Status
○ MySQL For Excel 1.3.8	Visual Studio 2010 Tools for Office R...	
○ MySQL for Visual Studio 1.2.9	Visual Studio version 2015, 2017 or 2...	Manual
✓ Connector/Python 8.0.21	Python 64-bit (2.7, 3.5, 3.6, 3.7 or 3.8)...	Complete

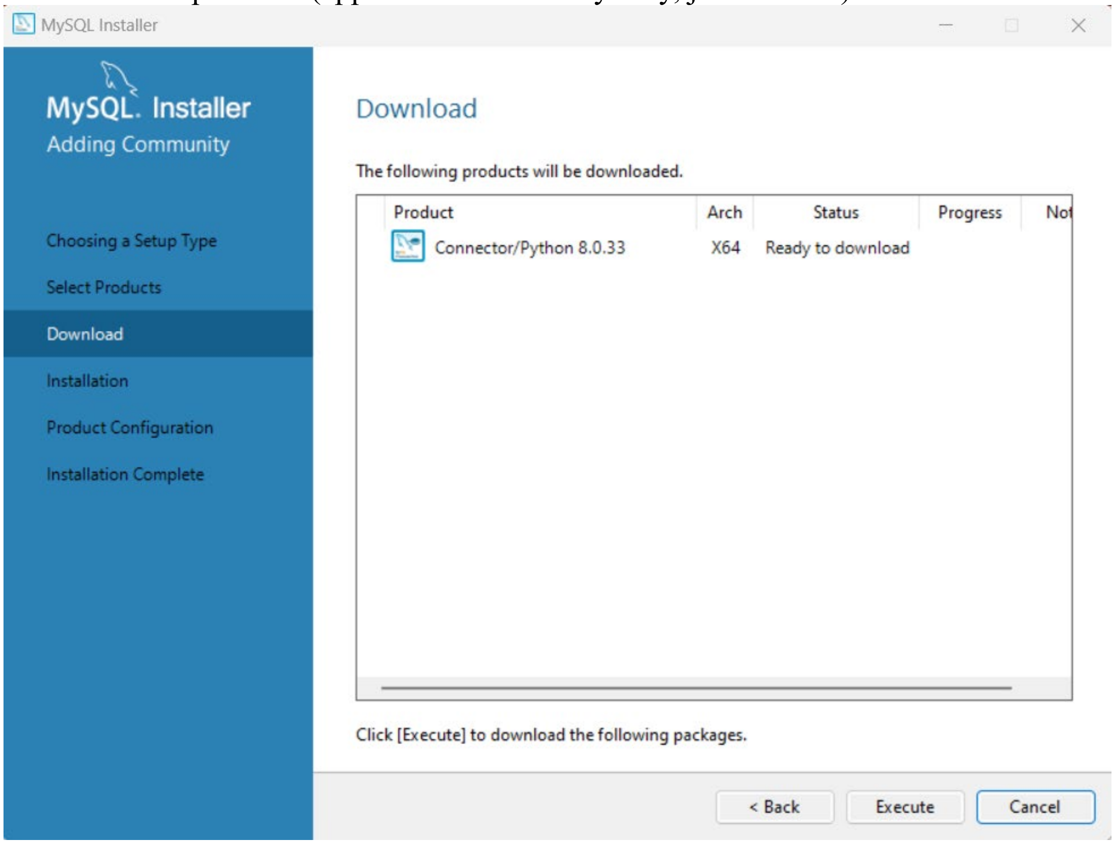
Requirement Details

This is a manual requirement that has been satisfied.

Requirement: Python 64-bit (2.7, 3.5, 3.6, 3.7 or 3.8) is not installed

Download URL: http://www.python.org/download/

4. Download the products (appearance of list may vary, just execute).



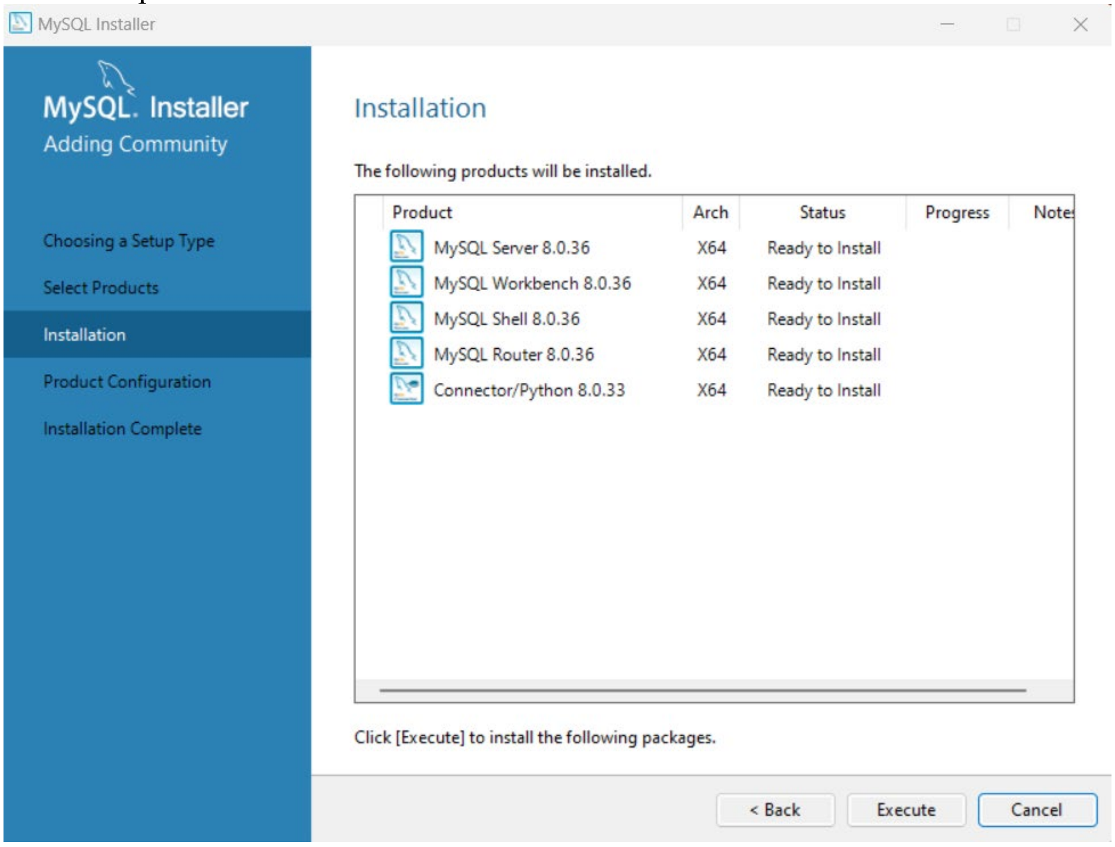
Download

The following products will be downloaded.

Product	Arch	Status	Progress	Not
Connector/Python 8.0.33	X64	Ready to download		

Click [Execute] to download the following packages.

5. Install the products.



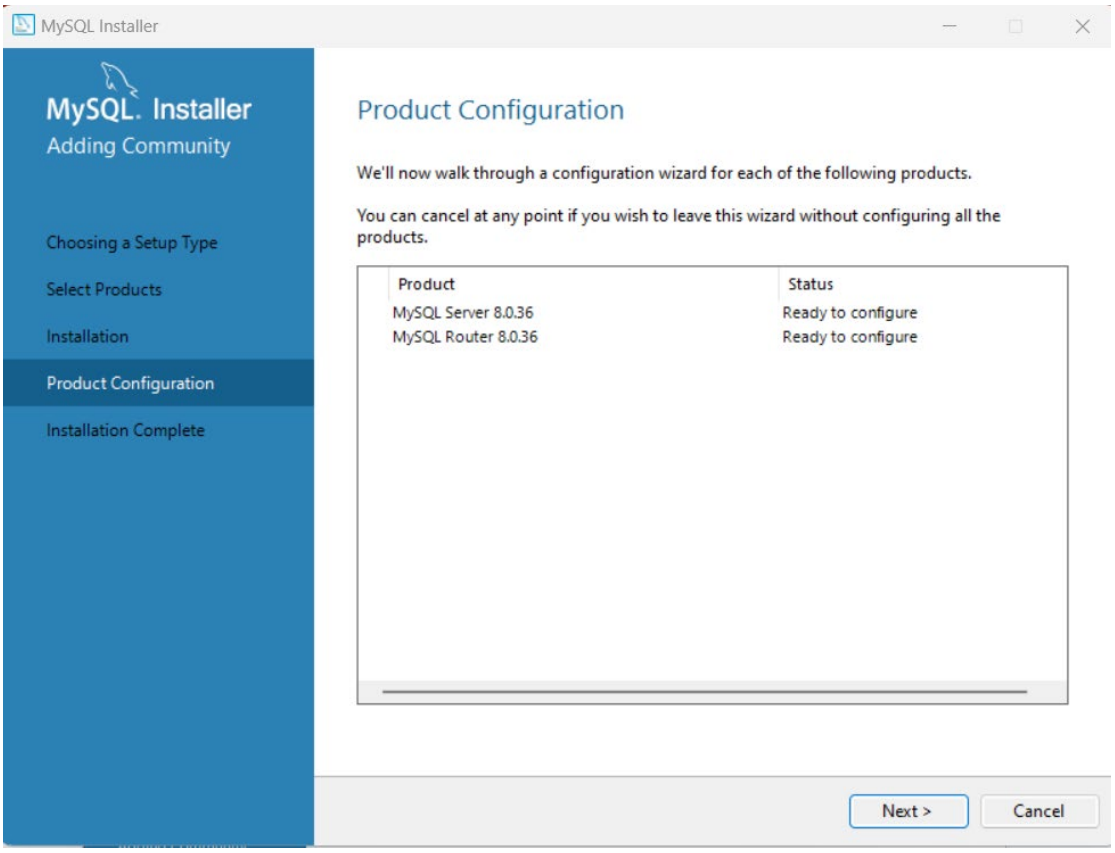
Installation

The following products will be installed.

Product	Arch	Status	Progress	Note
MySQL Server 8.0.36	X64	Ready to Install		
MySQL Workbench 8.0.36	X64	Ready to Install		
MySQL Shell 8.0.36	X64	Ready to Install		
MySQL Router 8.0.36	X64	Ready to Install		
Connector/Python 8.0.33	X64	Ready to Install		

Click [Execute] to install the following packages.

6. Select next for the product configuration.


MySQL Installer

MySQL. Installer
Adding Community

Choosing a Setup Type
Select Products
Installation
Product Configuration
Installation Complete

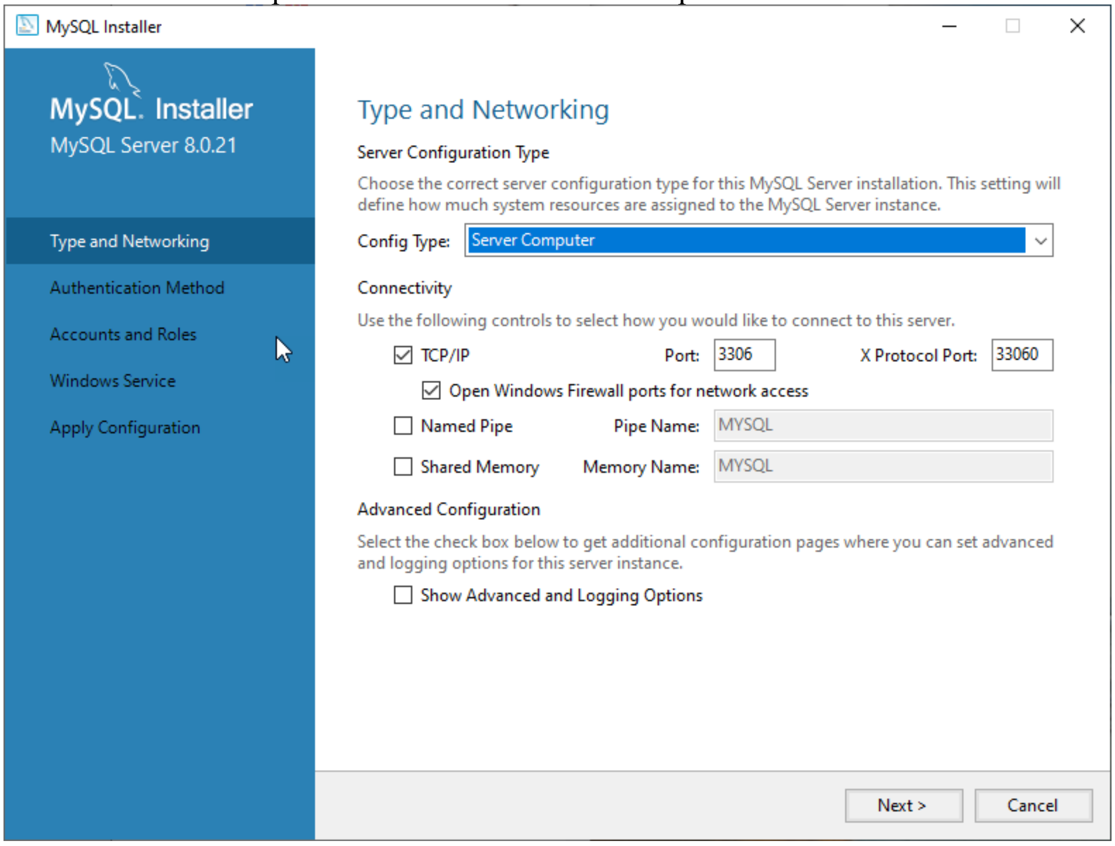
Product Configuration

We'll now walk through a configuration wizard for each of the following products.

You can cancel at any point if you wish to leave this wizard without configuring all the products.

Product	Status
MySQL Server 8.0.36	Ready to configure
MySQL Router 8.0.36	Ready to configure

7. Choose 'Server Computer' and select 'TCP/IP' with port number 3306.


MySQL Installer

MySQL. Installer
MySQL Server 8.0.21

Type and Networking
Authentication Method
Accounts and Roles
Windows Service
Apply Configuration

Type and Networking

Server Configuration Type

Choose the correct server configuration type for this MySQL Server installation. This setting will define how much system resources are assigned to the MySQL Server instance.

Config Type: Server Computer

Connectivity

Use the following controls to select how you would like to connect to this server.

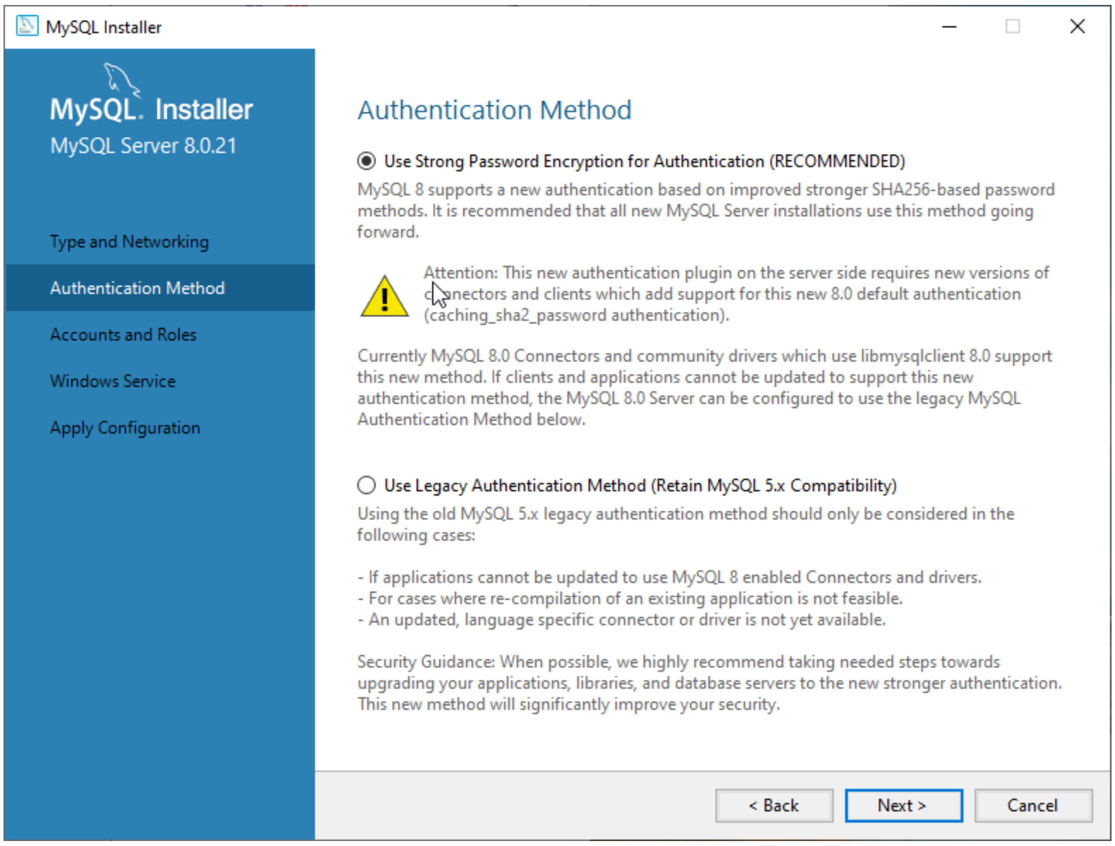
- [✓] TCP/IP Port: 3306 X Protocol Port: 33060
  - [✓] Open Windows Firewall ports for network access
- [ ] Named Pipe Pipe Name: MYSQL
- [ ] Shared Memory Memory Name: MYSQL

Advanced Configuration

Select the check box below to get additional configuration pages where you can set advanced and logging options for this server instance.

- [ ] Show Advanced and Logging Options

8. Select 'Use strong password encryption for authentication.'


MySQL Installer

MySQL. Installer
MySQL Server 8.0.21

- Type and Networking
- Authentication Method
- Accounts and Roles
- Windows Service
- Apply Configuration

Authentication Method

(●) Use Strong Password Encryption for Authentication (RECOMMENDED)

MySQL 8 supports a new authentication based on improved stronger SHA256-based password methods. It is recommended that all new MySQL Server installations use this method going forward.

Attention: This new authentication plugin on the server side requires new versions of connectors and clients which add support for this new 8.0 default authentication (caching_sha2_password authentication).

Currently MySQL 8.0 Connectors and community drivers which use libmysqlclient 8.0 support this new method. If clients and applications cannot be updated to support this new authentication method, the MySQL 8.0 Server can be configured to use the legacy MySQL Authentication Method below.

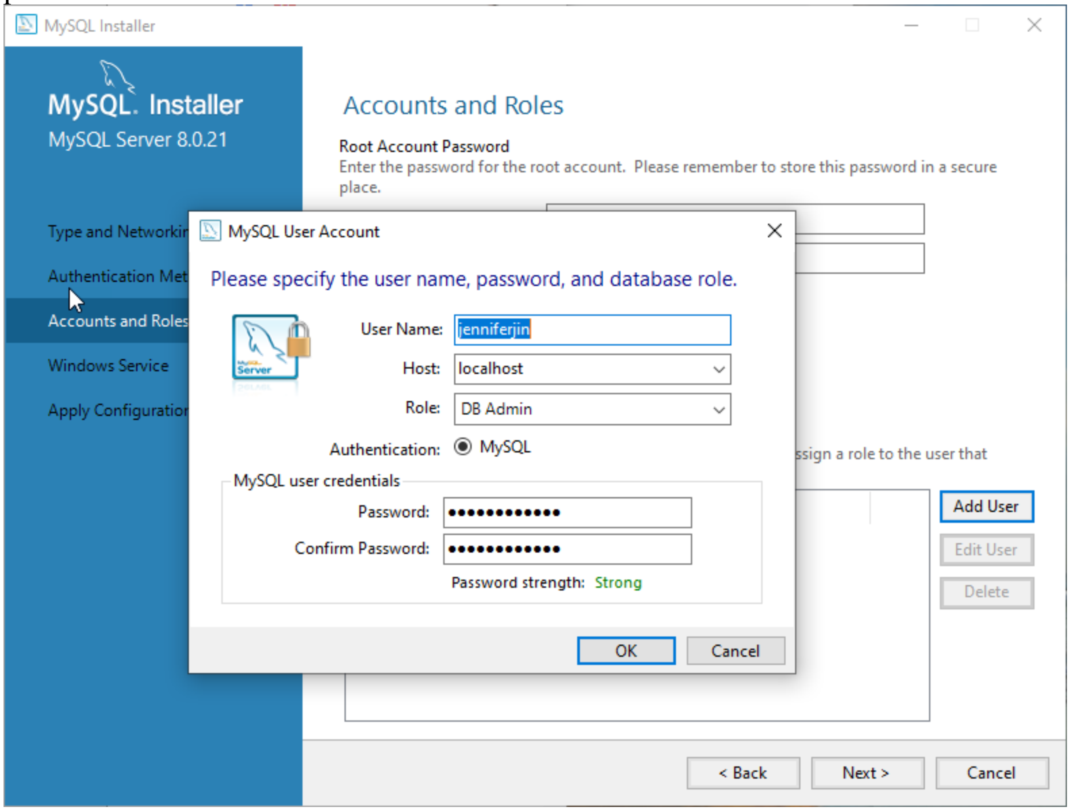
( ) Use Legacy Authentication Method (Retain MySQL 5.x Compatibility)

Using the old MySQL 5.x legacy authentication method should only be considered in the following cases:

- If applications cannot be updated to use MySQL 8 enabled Connectors and drivers.
- For cases where re-compilation of an existing application is not feasible.
- An updated, language specific connector or driver is not yet available.

Security Guidance: When possible, we highly recommend taking needed steps towards upgrading your applications, libraries, and database servers to the new stronger authentication. This new method will significantly improve your security.

9. Select a user name, choose *localhost* for host, *DB Admin* for Role and type your password.


MySQL Installer

MySQL. Installer
MySQL Server 8.0.21

Accounts and Roles

Root Account Password
Enter the password for the root account. Please remember to store this password in a secure place.

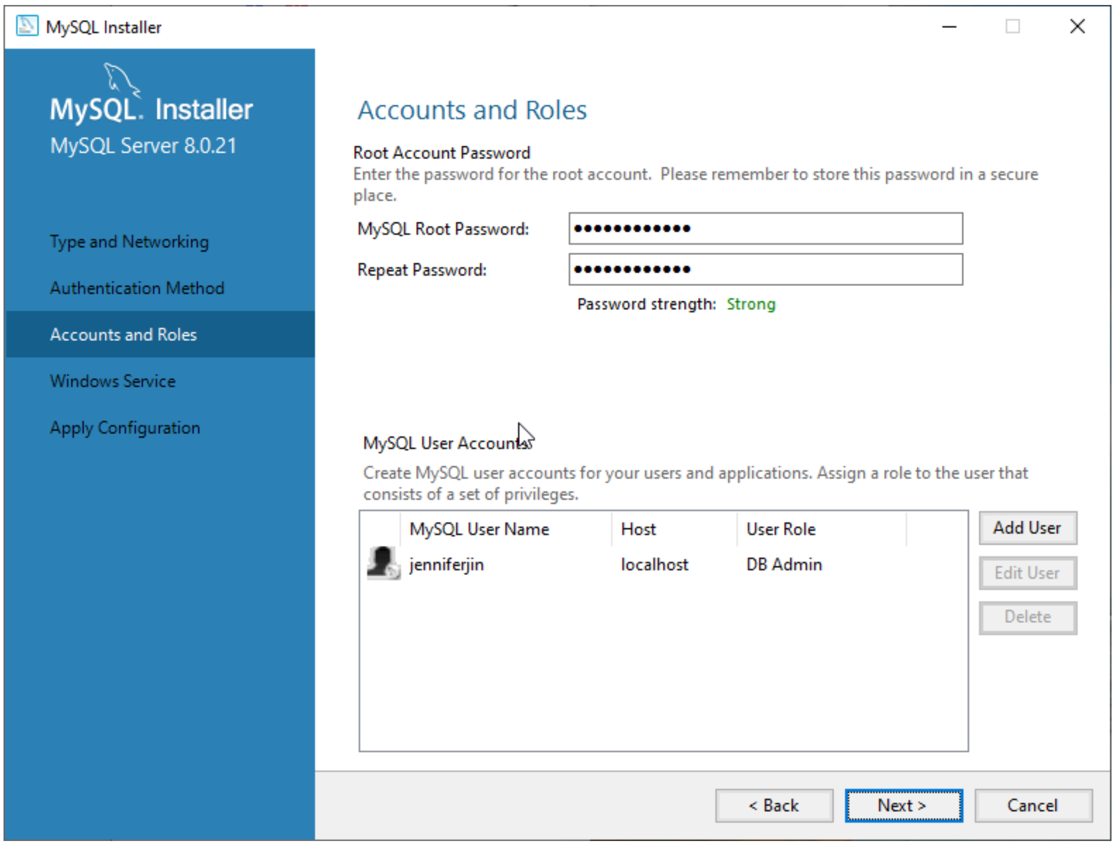
MySQL User Account

Please specify the user name, password, and database role.

User Name: jenniferjin
Host: localhost
Role: DB Admin
Authentication: (●) MySQL

MySQL user credentials
Password: ●●●●●●●●●●●●
Confirm Password: ●●●●●●●●●●●●
Password strength: Strong

10. Type your password and click next.


MySQL Installer

MySQL. Installer
MySQL Server 8.0.21

- Type and Networking
- Authentication Method
- Accounts and Roles
- Windows Service
- Apply Configuration

Accounts and Roles

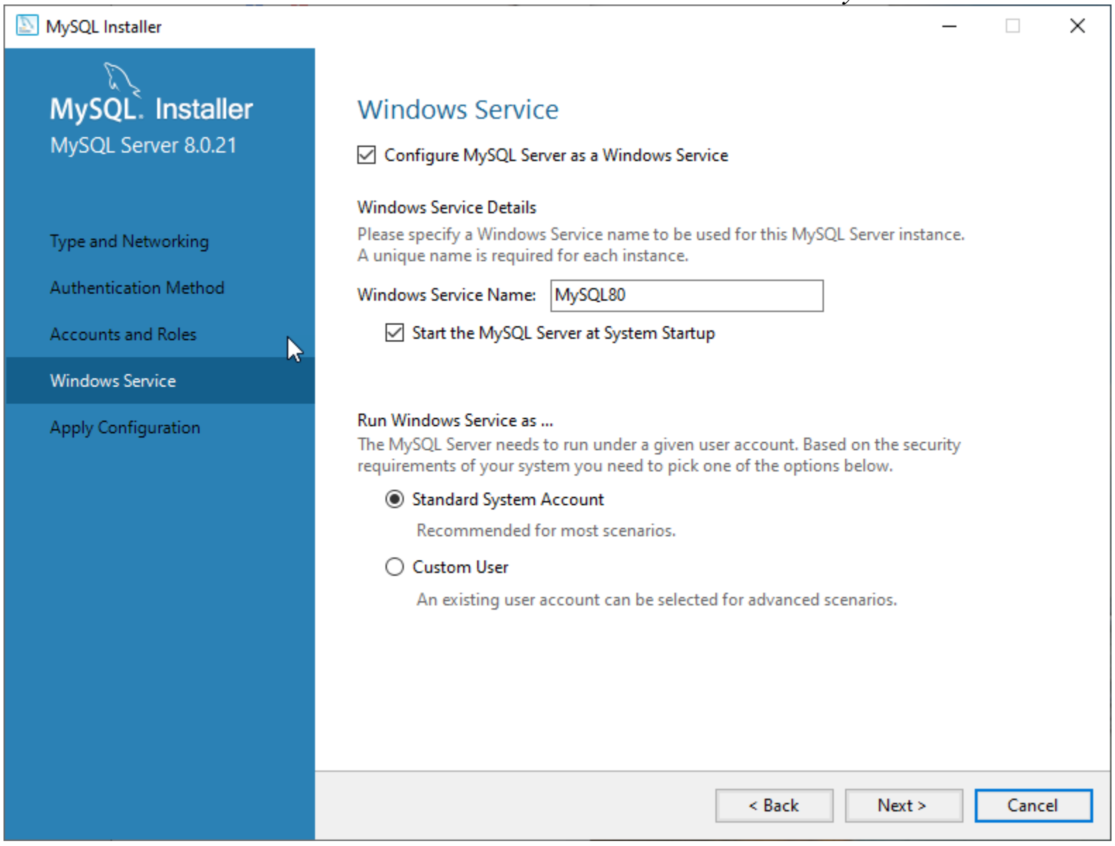
Root Account Password
Enter the password for the root account. Please remember to store this password in a secure place.

MySQL Root Password: ●●●●●●●●●●●●
Repeat Password: ●●●●●●●●●●●●
Password strength: Strong

MySQL User Accounts
Create MySQL user accounts for your users and applications. Assign a role to the user that consists of a set of privileges.

MySQL User Name	Host	User Role
jenniferjin	localhost	DB Admin

11. Choose a name for Windows Service Name and select *Standard System Account*.


MySQL Installer

MySQL. Installer
MySQL Server 8.0.21

- Type and Networking
- Authentication Method
- Accounts and Roles
- Windows Service
- Apply Configuration

Windows Service

☑ Configure MySQL Server as a Windows Service

Windows Service Details
Please specify a Windows Service name to be used for this MySQL Server instance.
A unique name is required for each instance.

Windows Service Name: MySQL80

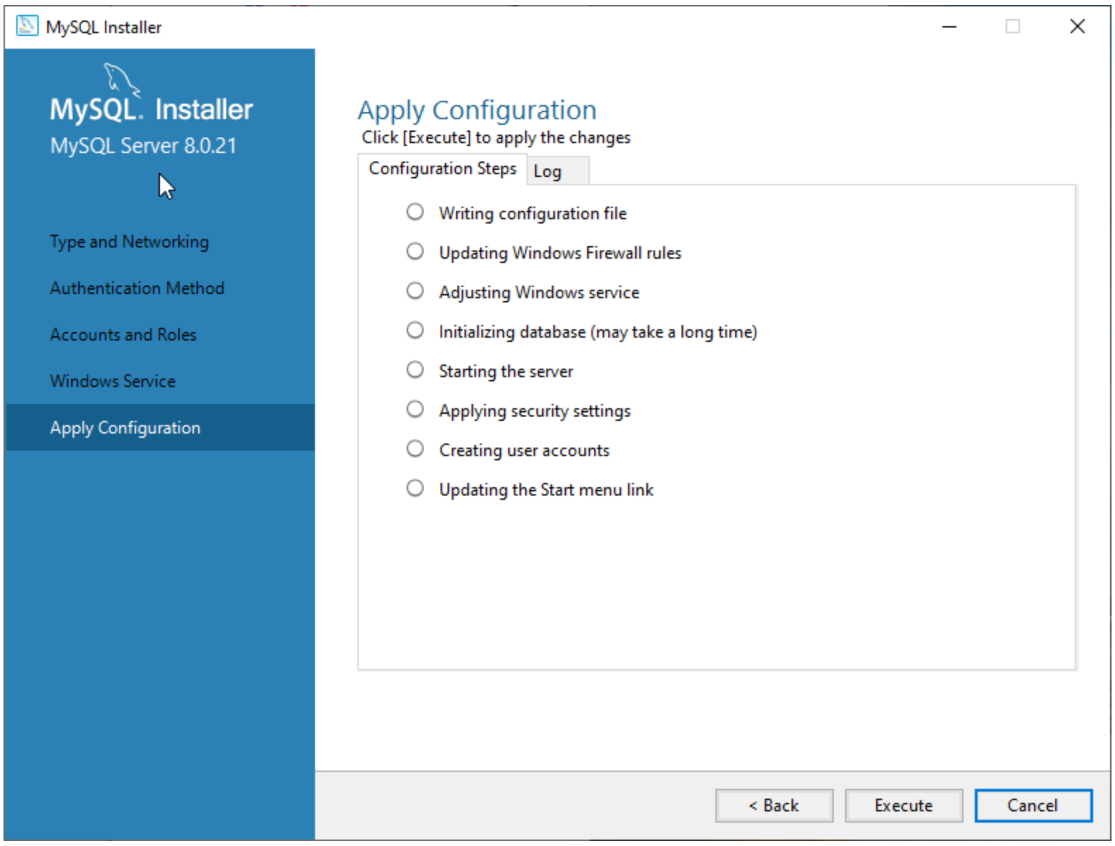
☑ Start the MySQL Server at System Startup

Run Windows Service as ...
The MySQL Server needs to run under a given user account. Based on the security requirements of your system you need to pick one of the options below.

● Standard System Account
Recommended for most scenarios.

○ Custom User
An existing user account can be selected for advanced scenarios.

12. Apply configuration.

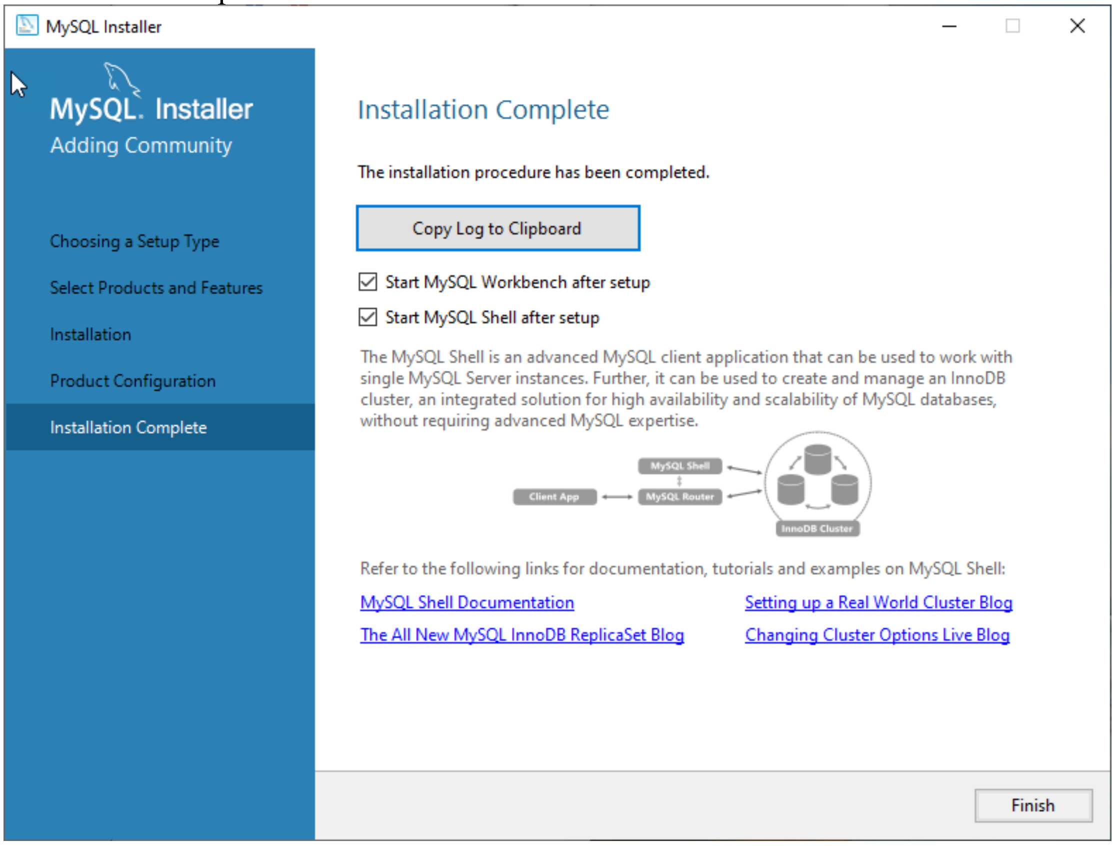


Apply Configuration

Click [Execute] to apply the changes

- Writing configuration file
- Updating Windows Firewall rules
- Adjusting Windows service
- Initializing database (may take a long time)
- Starting the server
- Applying security settings
- Creating user accounts
- Updating the Start menu link

13. Installation Complete!



Installation Complete

The installation procedure has been completed.

[✓] Start MySQL Workbench after setup

[✓] Start MySQL Shell after setup

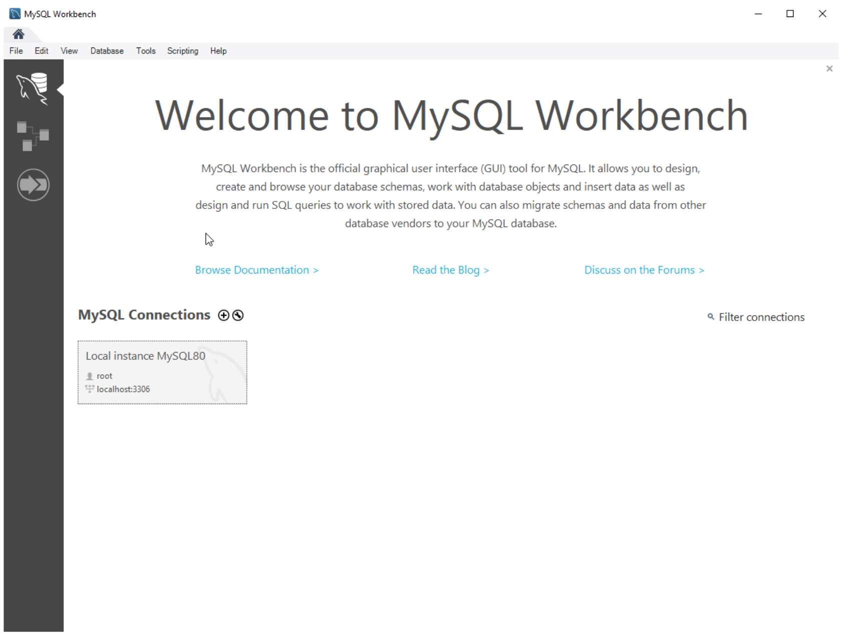
The MySQL Shell is an advanced MySQL client application that can be used to work with single MySQL Server instances. Further, it can be used to create and manage an InnoDB cluster, an integrated solution for high availability and scalability of MySQL databases, without requiring advanced MySQL expertise.

Refer to the following links for documentation, tutorials and examples on MySQL Shell:

MySQL Shell Documentation | Setting up a Real World Cluster Blog
The All New MySQL InnoDB ReplicaSet Blog | Changing Cluster Options Live Blog

STEP 2 - A GUI Tool for MySQL – MySQL Workbench

Download the MySQL Workbench from MySQL download page. After the installation, run it and you should see the following:


MySQL Workbench

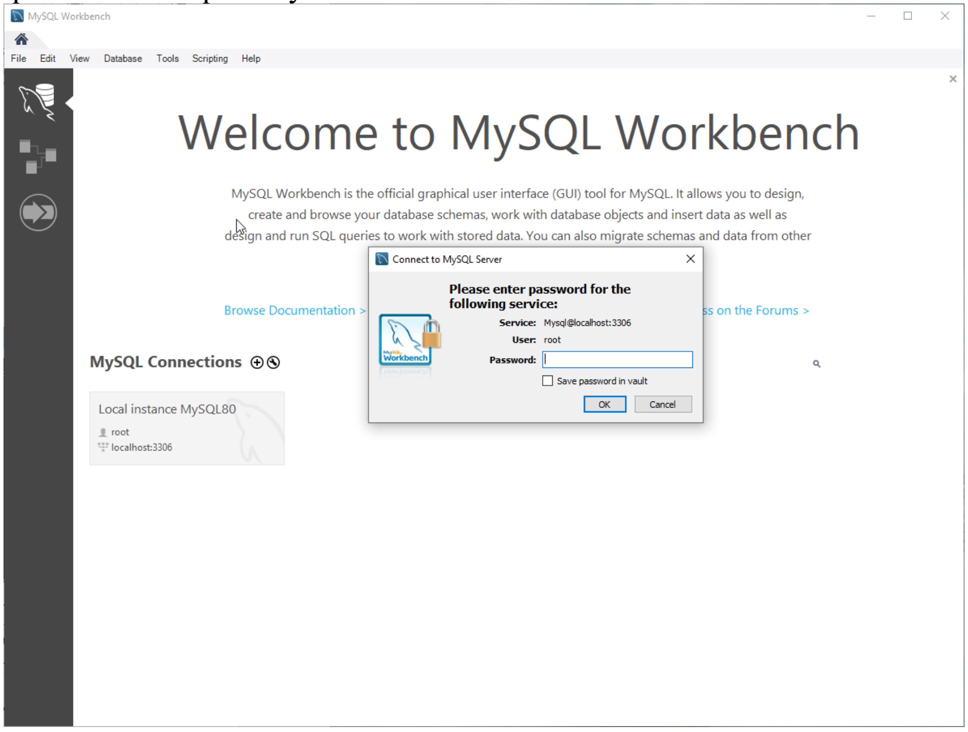
Welcome to MySQL Workbench

MySQL Workbench is the official graphical user interface (GUI) tool for MySQL. It allows you to design, create and browse your database schemas, work with database objects and insert data as well as design and run SQL queries to work with stored data. You can also migrate schemas and data from other database vendors to your MySQL database.

Local instance MySQL80
root
localhost:3306

Double click on the local instance. Note that the port should be the same as when you installed MySQL (3306 in the case). If not, edit the connection by:
Windows and Linux: hover over the right side of a connection title and click the title.
OS X: hover over a connection title and click the little (i) in appears in the bottom right corner

Type in your password and connect to the database. Now you can perform queries or manipulate your database:


MySQL Workbench

Welcome to MySQL Workbench

MySQL Workbench is the official graphical user interface (GUI) tool for MySQL. It allows you to design, create and browse your database schemas, work with database objects and insert data as well as design and run SQL queries to work with stored data. You can also migrate schemas and data from other

Connect to MySQL Server

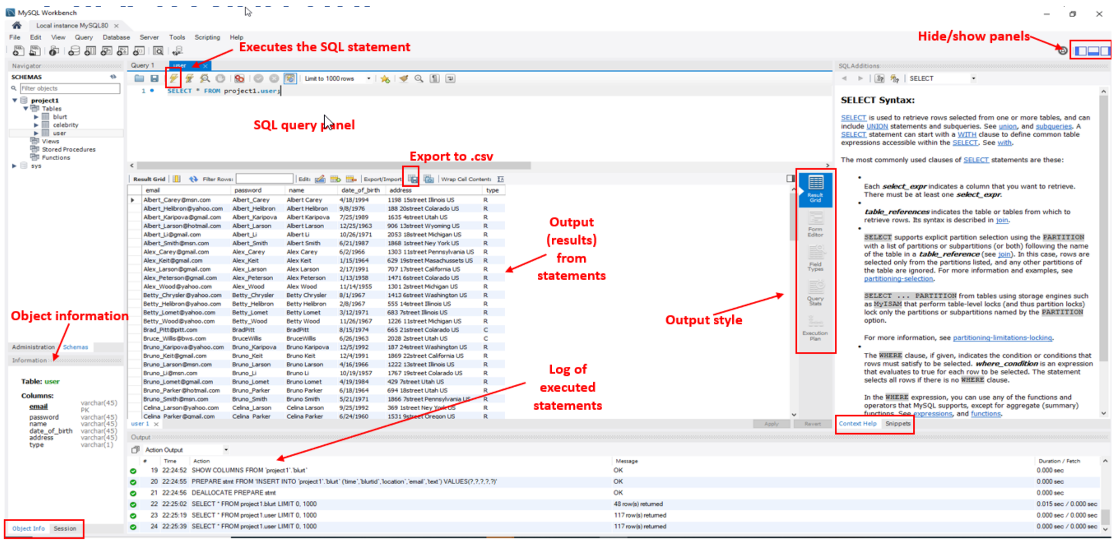
Please enter password for the following service:

Service: Mysql@localhost:3306
User: root
Password:

Save password in vault

Local instance MySQL80
root
localhost:3306

This is the MySQL workbench overview.



```
SELECT * FROM project1.user;
```

email	password	name	date_of_birth	address	type
Albert_Carey@msn.com	Albert_Carey	Albert Carey	4/18/1994	1198 15street Illinois US	R
Albert_Helbron@yahoo.com	Albert_Helbron	Albert Helbron	9/8/1976	188 20street Colarado US	R
Albert_Karipova@gmail.com	Albert_Karipova	Albert Karipova	7/25/1989	1635 4street Utah US	R
Albert_Larson@hotmail.com	Albert_Larson	Albert Larson	12/25/1963	906 13street Wyoming US	R
Albert_Li@gmail.com	Albert_Li	Albert Li	10/26/1971	2053 18street Michigan US	R
Albert_Smith@msn.com	Albert_Smith	Albert Smith	6/21/1987	1868 1street Ney York US	R
Alex_Carey@gmail.com	Alex_Carey	Alex Carey	6/2/1966	1303 11street Pennsylvania US	R
Alex_Keit@gmail.com	Alex_Keit	Alex Keit	1/15/1964	629 19street Masachussets US	R
Alex_Larson@gmail.com	Alex_Larson	Alex Larson	2/17/1991	707 17street California US	R
Alex_Peterson@gmail.com	Alex_Peterson	Alex Peterson	1/13/1958	1471 6street Colarado US	R
Alex_Wood@yahoo.com	Alex_Wood	Alex Wood	11/14/1955	1301 2street Michigan US	R
Betty_Chrysler@yahoo.com	Betty_Chrysler	Betty Chrysler	8/1/1967	1413 6street Washington US	R
Betty_Helbron@yahoo.com	Betty_Helbron	Betty Helbron	2/8/1967	555 14street Illinois US	R
Betty_Lomet@yahoo.com	Betty_Lomet	Betty Lomet	3/12/1971	683 7street Illinois US	R
Betty_Wood@yahoo.com	Betty_Wood	Betty Wood	11/26/1967	1226 11street Michigan US	R
Brad_Pitt@pitt.com	BradPitt	BradPitt	8/15/1974	665 21street Colarado US	C
Bruce_Willis@bws.com	BruceWillis	BruceWillis	6/26/1963	2028 2street Utah US	C
Bruno_Karipova@yahoo.com	Bruno_Karipova	Bruno Karipova	12/5/1992	187 24street Washington US	R
Bruno_Keit@gmail.com	Bruno_Keit	Bruno Keit	12/4/1991	1869 22street California US	R
Bruno_Larson@msn.com	Bruno_Larson	Bruno Larson	4/16/1966	1222 13street Illinois US	R
Bruno_Li@msn.com	Bruno_Li	Bruno Li	10/19/1957	1767 19street Colarado US	R
Bruno_Lomet@gmail.com	Bruno_Lomet	Bruno Lomet	4/19/1984	429 7street Utah US	R
Bruno_Parker@hotmail.com	Bruno_Parker	Bruno Parker	6/18/1964	694 18street Utah US	R
Bruno_Smith@msn.com	Bruno_Smith	Bruno Smith	5/21/1971	1866 7street Pennsylvania US	R
Celina_Larson@yahoo.com	Celina_Larson	Celina Larson	9/25/1992	369 1street Ney York US	R
Celina_Parker@gmail.com	Celina_Parker	Celina Parker	6/24/1960	1531 9street Oregon US	R

Table: user

Columns:

email	varchar(45) PK
password	varchar(45)
name	varchar(45)
date_of_birth	varchar(45)
address	varchar(45)
type	varchar(1)

```
#   Time      Action                                                                                          Message            Duration / Fetch
19  22:24:52  SHOW COLUMNS FROM `project1`.`blurt`                                                            OK                 0.000 sec
20  22:24:55  PREPARE stmt FROM 'INSERT INTO `project1`.`blurt` (`time`,`blurtid`,`location`,`email`,`text`) VALUES(?,?,?,?,?)'   OK   0.000 sec
21  22:24:56  DEALLOCATE PREPARE stmt                                                                         OK                 0.000 sec
22  22:25:02  SELECT * FROM project1.blurt LIMIT 0, 1000                                                      48 row(s) returned   0.015 sec / 0.000 sec
23  22:25:19  SELECT * FROM project1.user LIMIT 0, 1000                                                       117 row(s) returned  0.000 sec / 0.000 sec
24  22:25:39  SELECT * FROM project1.user LIMIT 0, 1000                                                       117 row(s) returned  0.000 sec / 0.000 sec
```

SELECT Syntax:

SELECT is used to retrieve rows selected from one or more tables, and can include UNION statements and subqueries. See union, and subqueries. A SELECT statement can start with a WITH clause to define common table expressions accessible within the SELECT. See with.

The most commonly used clauses of SELECT statements are these:

- Each ***select_expr*** indicates a column that you want to retrieve. There must be at least one ***select_expr***.
- ***table_references*** indicates the table or tables from which to retrieve rows. Its syntax is described in join.
- `SELECT` supports explicit partition selection using the `PARTITION` with a list of partitions or subpartitions (or both) following the name of the table in a ***table_reference*** (see join). In this case, rows are selected only from the partitions listed, and any other partitions of the table are ignored. For more information and examples, see partitioning-selection.

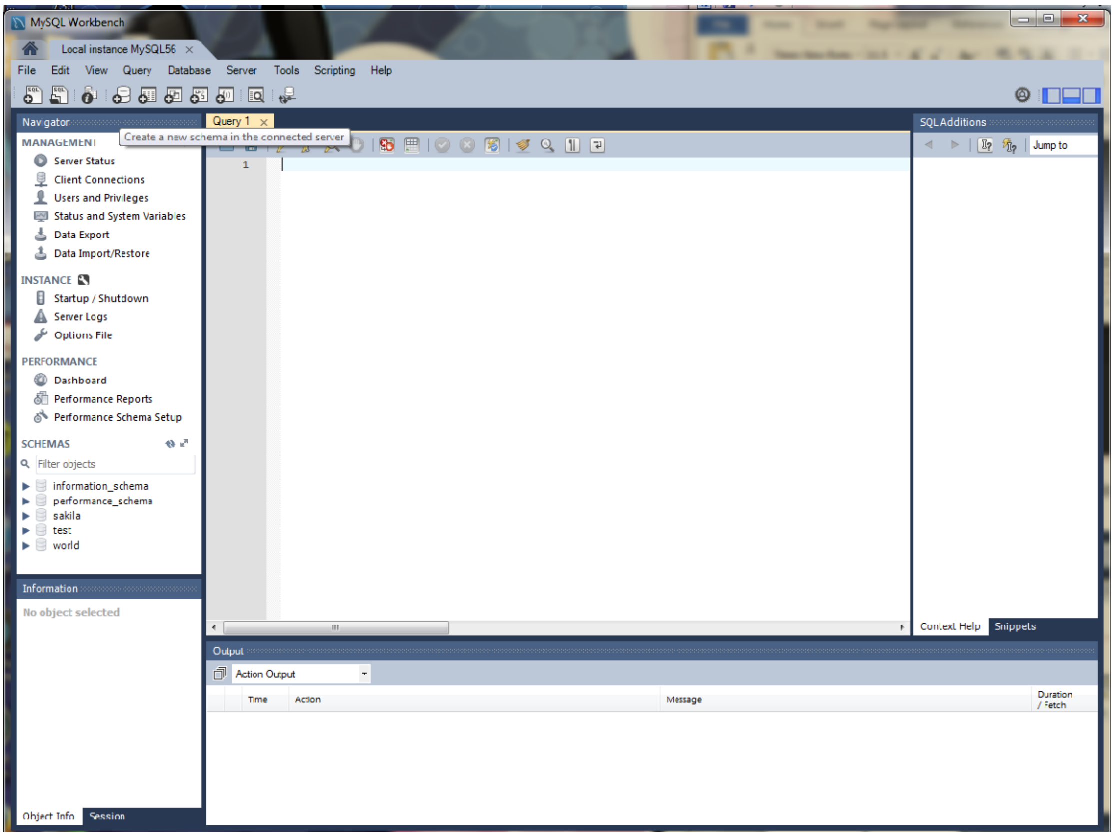
  `SELECT ... PARTITION` from tables using storage engines such as `MyISAM` that perform table-level locks (and thus partition locks) lock only the partitions or subpartitions named by the `PARTITION` option.

  For more information, see partitioning-limitations-locking.
- The `WHERE` clause, if given, indicates the condition or conditions that rows must satisfy to be selected. ***where_condition*** is an expression that evaluates to true for each row to be selected. The statement selects all rows if there is no `WHERE` clause.

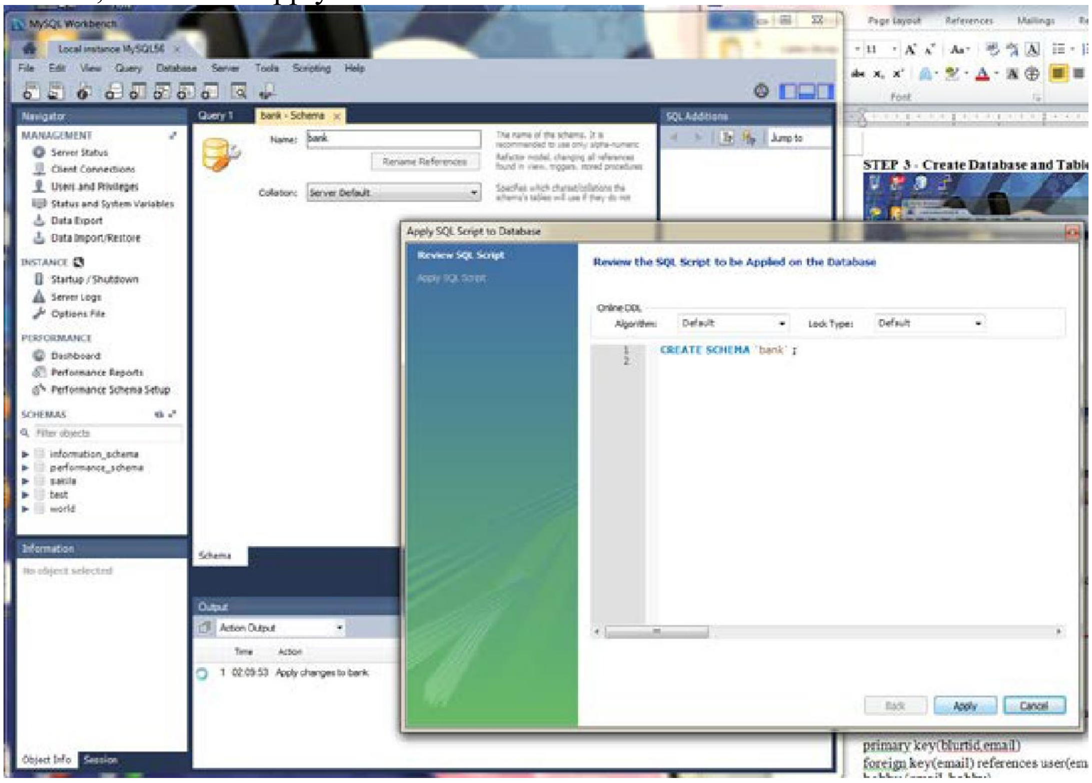
  In the `WHERE` expression, you can use any of the functions and operators that MySQL supports, except for aggregate (summary) functions. See expressions, and functions.

STEP 3 - Create Database and Tables

1. Click on "Create a new schema in the connected server" icon on the upper left corner.



2. Under the schema click on “create a new table in the active schema in connected server”, then click “apply.”



Name: bank

Collation: Server Default

Apply SQL Script to Database

Review the SQL Script to be Applied on the Database

Online DDL — Algorithm: Default — Lock Type: Default

```
CREATE SCHEMA `bank` ;
```

```
1  02:09:53  Apply changes to bank
```

STEP 3 - Create Database and Table

primary key(blurtid,email)
foreign key(email) references user(em

3. Fill in the columns with column name, data type and indicate the primary key(s).

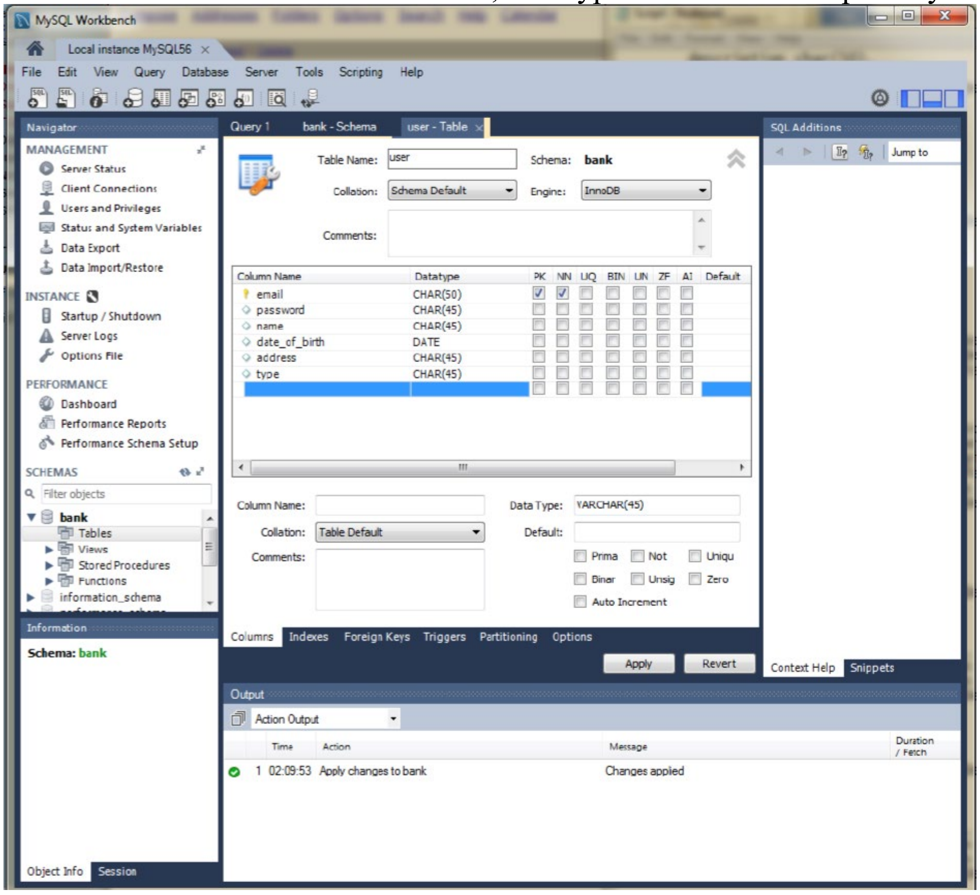


Table Name: user — Schema: bank

Collation: Schema Default — Engine: InnoDB

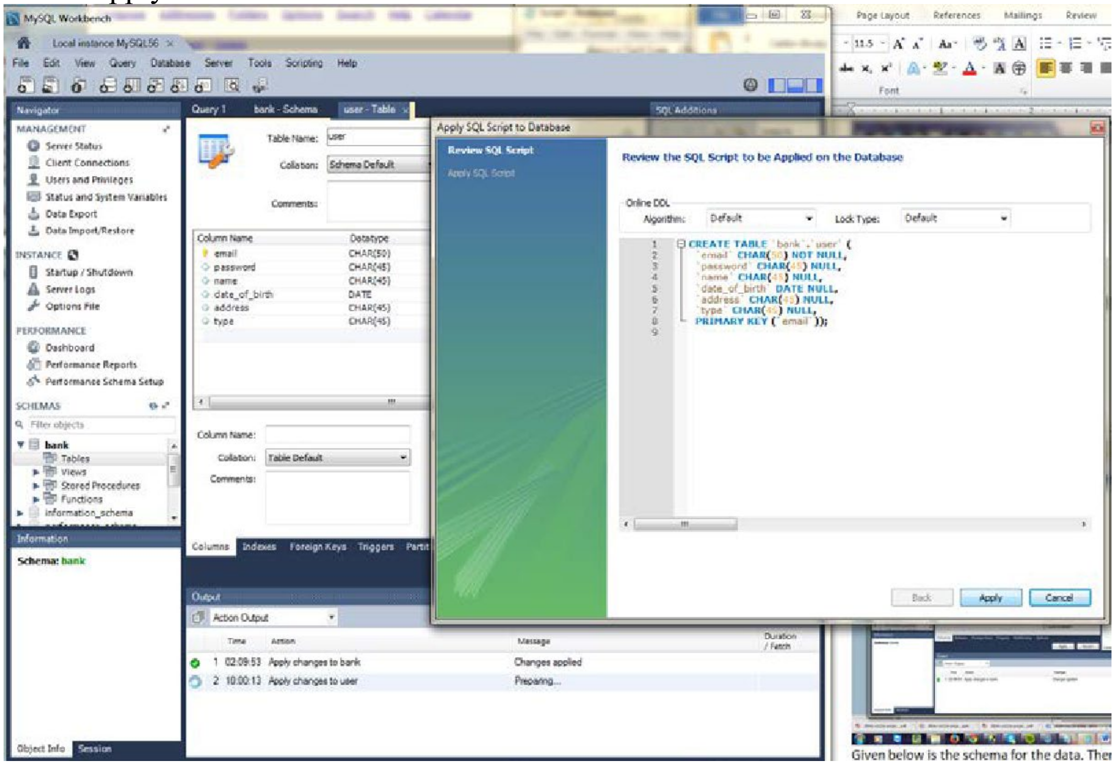
Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
email	CHAR(50)	✓	✓						
password	CHAR(45)								
name	CHAR(45)								
date_of_birth	DATE								
address	CHAR(45)								
type	CHAR(45)								

Column Name: — Data Type: VARCHAR(45)

Collation: Table Default — Default:

```
Time      Action                 Message
1  02:09:53  Apply changes to bank   Changes applied
```

4. Click “apply.”



Review the SQL Script to be Applied on the Database

Online DDL
Algorithm: Default   Lock Type: Default

```sql
CREATE TABLE `bank`.`user` (
  `email` CHAR(50) NOT NULL,
  `password` CHAR(45) NULL,
  `name` CHAR(45) NULL,
  `date_of_birth` DATE NULL,
  `address` CHAR(45) NULL,
  `type` CHAR(45) NULL,
  PRIMARY KEY (`email`));
```

```
   Time      Action                    Message
1  02:09:53  Apply changes to bank     Changes applied
2  10:00:13  Apply changes to user     Preparing...
```

5. You can right click on the “tables” under the schema to create new tables.

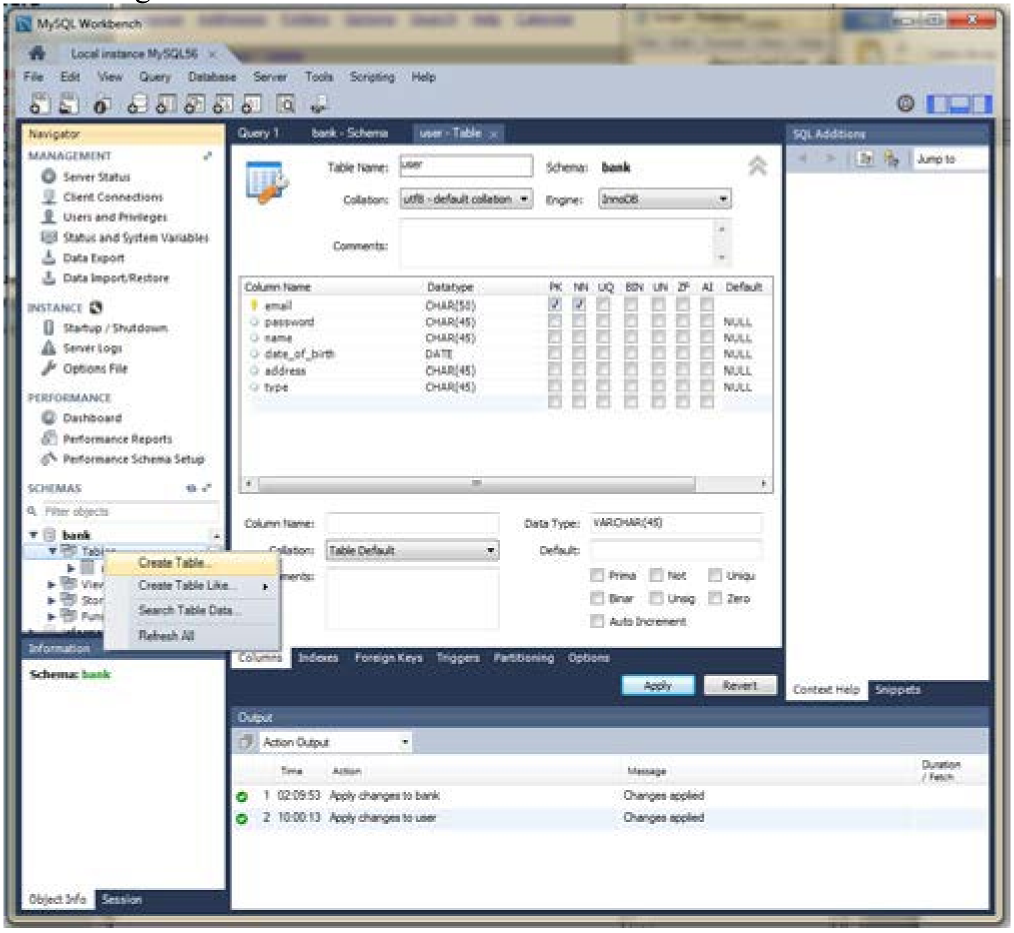


Table Name: user   Schema: bank
Collation: utf8 - default collation   Engine: InnoDB

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
email	CHAR(50)	✓	✓						
password	CHAR(45)								NULL
name	CHAR(45)								NULL
date_of_birth	DATE								NULL
address	CHAR(45)								NULL
type	CHAR(45)								NULL

```
   Time      Action                    Message
1  02:09:53  Apply changes to bank     Changes applied
2  10:00:13  Apply changes to user     Changes applied
```

Given below is the schema for the data. There are a total of 12 tables, thus 12 CSV files, each corresponding to a relational table.

user (email, password, name, date_of_birth, address, type)
primary key(email)

celebrity (email, website, kind)
primary key(email)

blurt (blurtid, email, text, location, time)
primary key(blurtid,email)
foreign key(email) references user(email)

hobby (email, hobby)
primary key(email,hobby)
foreign key(email) references user(email))

follow (follower,followee)
primary key(follower,followee)
foreign key(follower) references user(email)
foreign key(followee) references user(email))

vendor (id, name)
primary key(id)

vendor_ambassador (vendorid, email)
primary key(vendorid)
foreign key(email) references user(email)
foreign key(vendorid) references vendor(id))

topic (id, description)
primary key(id)

vendor_topics (vendorid,topicid)
primary key(vendorid, topicid)
foreign key(vendorid) references vendor(id)
foreign key(topicid) references topic(id))

blurt_analysis (email,blurtid,topicid,confidence,sentiment)
primary key(email, blurtid, topicid)
foreign key(email,blurtid) references blurt(email,blurtid)
foreign key(topicid) references topic(id)
constraint confidence >= 0 and confidence <=10
constraint sentiment >= -5 and sentiment <=5

advertisement (id, content, vendorid)
primary key(id)
foreign key(vendorid) references vendor(id))

user_ad (email,adid)
primary key(email,adid)
foreign key(email) references user(email)

foreign key(adid) references advertisement(id))

The design model:
Users can post their thoughts in form of short messages that we call “blurts”. When signing up, users need to provide their email and a password of their choice. In addition, they need to enter some basic information – name, date of birth, address, email ID and hobbies. Once signed up, they can (besides blurting) “follow” other users. To “follow” a user means subscribing to his/her “blurts”. Users are categorized into “regular” users and “celebrities”. A celebrity has an associated website url and an attribute called “kind” indicating whether he is a politician, actor, singer, etc. Each blurt by a user (regular or celebrity) is assigned an id. Blurt ids are serial and unique to the “blurts” by a given user; the first blurt by a given user would have blurt id 1 and ids are incremented for each successive blurt by the user. Note that blurt ids are unique only to a user, so blurts by two different users may have the same blurt id. Besides an id, each blurt also has its text, timestamp, and user location as additional attributes. The system should have a pre-defined notion of “topics” that are simply subjects that people may blurt about. Examples of topics might include music, pollution, disease, disaster, sports, weather, etc. A topic has a unique id and description (the name of the topic). Each blurt by a user is analyzed to associate with it zero or more topics. Related blurt-topic pairs are stored in blurt_analysis table. To account for the possible ambiguity arising from the choice of words or language used by a user, an association with a topic has a corresponding confidence level (an integer ranging from 1 – 10 indicating the strength of the association). For example consider the following blurt: “I absolutely hate the rainy weather, can’t go out, listening to the Beatles, just love them” is analyzed to be associated with two topics, weather and music (Beatles). For each topic, the associated sentiment is evaluated and quantified as an integral value ranging between -5 and 5, with higher values indicating a more positive sentiment. Considering the example blurt used above, the topic weather would have an associated sentiment of -5 (hate) while for music the corresponding value is 4 (love). **Note: You don’t need to implement the value constraint as MySQL doesn’t support it.** A vendor has interest in one or more topics and is interested in tracking all users who are blurting about a topic of interest. A vendor may also have a celebrity as its brand ambassador. Vendors create advertisements that have an associated unique id and a textual content. These advertisements are stored in the system and are available to be shown to the regular set of users (that is, not to the celebrities, just to the other “regular” users). Careful matching is done based upon a historical analysis of all blurts by a user. Based upon the analysis, a user may be shown zero or more advertisements.

STEP 4 – Import CSV files
Script Template:
LOAD DATA LOCAL INFILE "[CSV file name]" INTO TABLE [table name] COLUMNS TERMINATED BY ',' LINES TERMINATED BY '\n'

For each CSV file, replace [CSV file name] and [table name] with actual CSV file name and corresponding table name, e.g.:
LOAD DATA LOCAL INFILE "d:\\csvdata\\advertisement.csv" INTO TABLE advertisement COLUMNS TERMINATED BY ',' LINES TERMINATED BY '\n'

Execute 12 scripts using the GUI client

STEP 5 - Form SQL Queries

For the following statements, you are required to form SQL queries and execute them using the GUI client. Then export the result using the name "Query x.csv", x being the label of each query. Put all the SQL you formed into a file named "Script.txt" in the same order. Then archive the file as "5720project1-xxxxxxxxx.zip", xxxxxxxxx being your student id, and turn it in on Canvas. The filenames of your result has to follow the instructions exactly or you may get a deduction in your credit.

1) Print every hobby, along with the total number of users that included that hobby in their profile.
2) Print the name and address of all celebrities who are not movie stars.
3) Print a list of every vendor and the number of different advertisements they're running, in descending order.
4) Print email, password, name, date_of_birth, kind of all celebrities who are Movie Stars and are following someone.
5) Print the name and email of every celebrity along with their total number of followers if the amount is over 55. The contents should be in descending order based on the number of followers.6) Find all pair of users (A,B) such that A and B have not blurted on a common topic, but user A is following user B. The query should print the names of A and B in that order.
7) Print every location, along with the total amount of blurts, where users made more than 40 collective blurts. The query should print the name of the location and total amount of user blurts in descending order of the amount of blurts.
8) For each vendor, print the email of its brand ambassador along with the number of users the brand ambassador is following. Your query should print the vendor name, ambassador email, and number of users the ambassador is following. The output should be in descending order according to the numbers of users being followed.
9) Print the following: vendor id, vendor name, vendor ambassador's email, and the total number of users that have been shown the ad(s) from the vendor. Do this for every vendor.