

Experiment 1

Dr. Jyotismita Chaki

Installation: Download MySQL Installer

- If you want to install MySQL on the Windows environment, using MySQL installer is the easiest way.
- To download MySQL installer, go to the following link <http://dev.mysql.com/downloads/installer/>. There are two installer files:
 - If you are connecting to the internet while installing MySQL, you can choose the online installation version `mysql-installer-web-community-<version>.exe`.
 - In case you want to install MySQL offline, you can download the `mysql-installer-community-<version>.exe` file.

Installation

MySQL Community Downloads

MySQL Installer

General Availability (GA) Releases

Archives



MySQL Installer 8.0.26

Select Operating System:

Microsoft Windows

Looking for previous GA versions?

Windows (x86, 32-bit), MSI Installer

(mysql-installer-web-community-8.0.26.0.msi)

8.0.26

2.4M

[Download](#)

MD5: eaddc383a742775a5b33a3783a4890fb | [Signature](#)

Windows (x86, 32-bit), MSI Installer

(mysql-installer-community-8.0.26.0.msi)

8.0.26

450.7M

[Download](#)

MD5: b5b8e6bc39f2b163b817264ae206b815 | [Signature](#)

Installation

MySQL Community Downloads

Login Now or Sign Up for a free account.

An Oracle Web Account provides you with the following advantages:

- Fast access to MySQL software downloads
- Download technical White Papers and Presentations
- Post messages in the MySQL Discussion Forums
- Report and track bugs in the MySQL bug system

Login »

using my Oracle Web account

Sign Up »

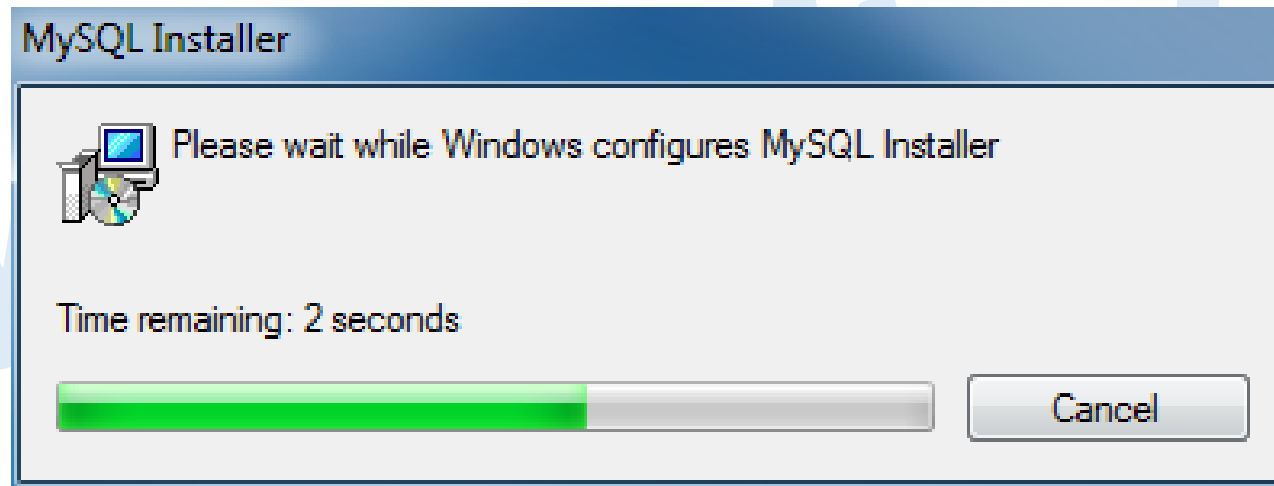
for an Oracle Web account

MySQL.com is using Oracle SSO for authentication. If you already have an Oracle Web account, click the Login link. Otherwise, you can signup for a free account by clicking the Sign Up link and following the instructions.

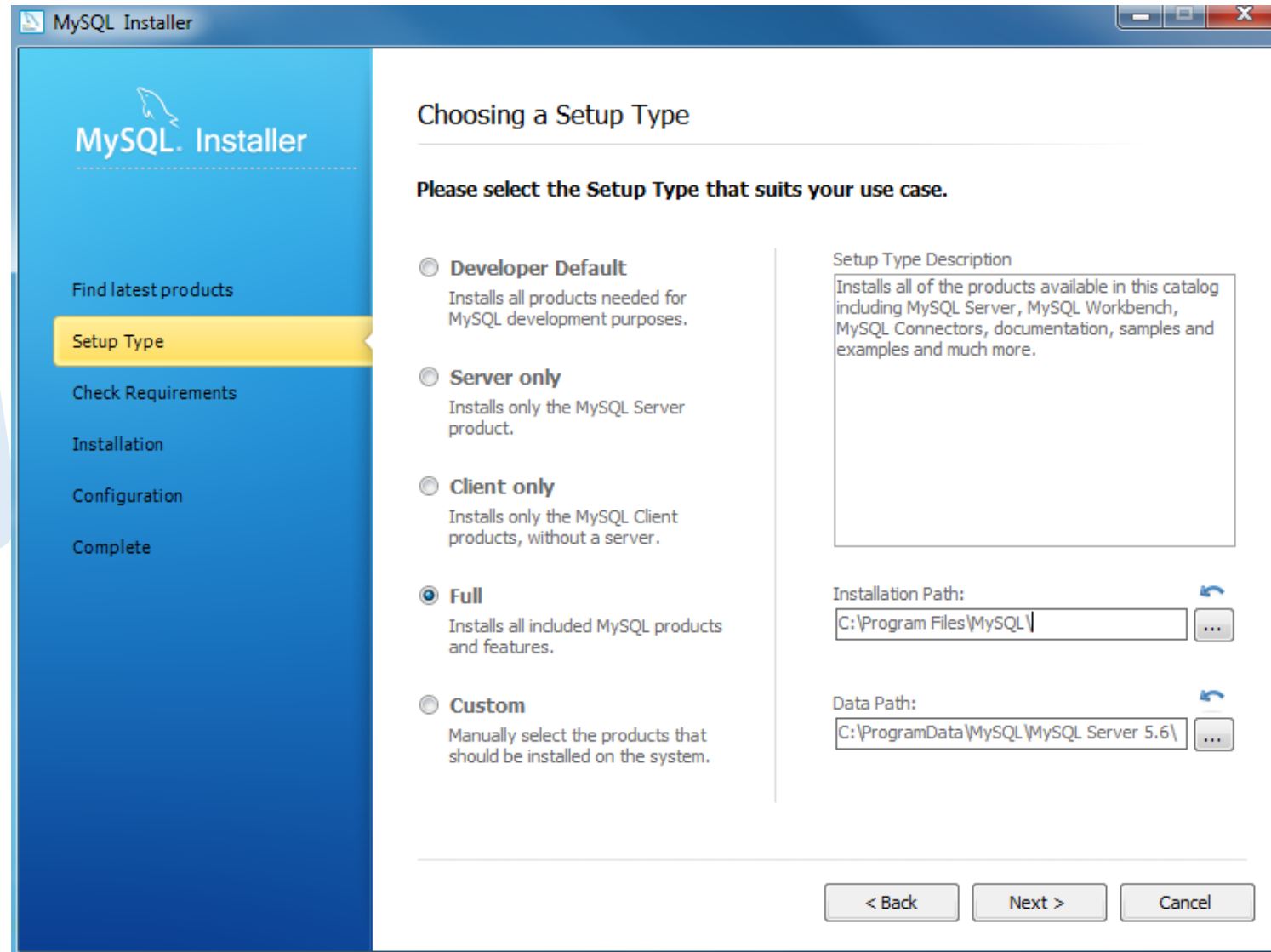
No thanks, just start my download.

Installation: Install MySQL via MySQL Installer

- To install MySQL using the MySQL installer, double-click on the MySQL installer file and follow the steps below:

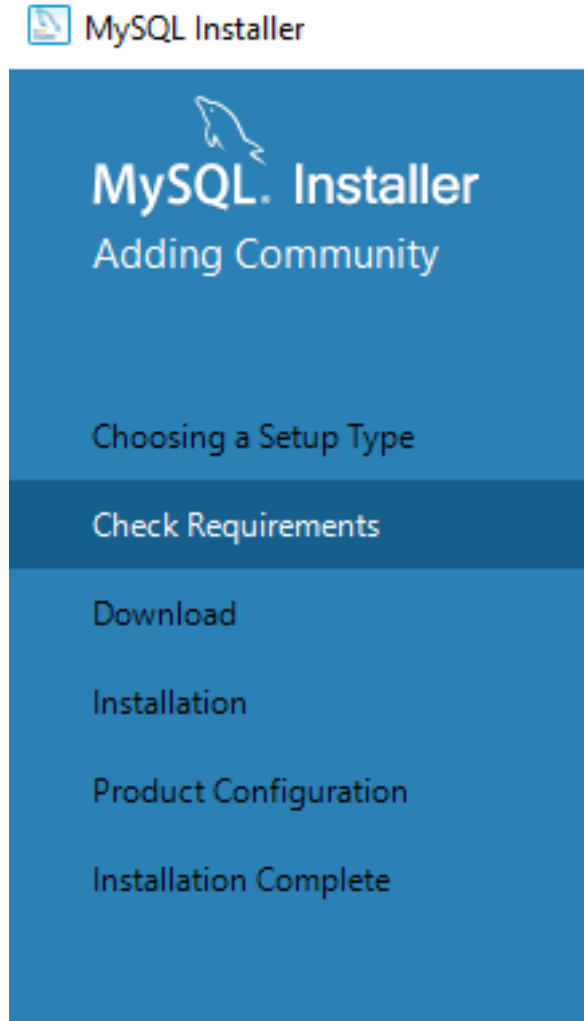


Installation: Setup Type



Installation

Select Execute



Check Requirements

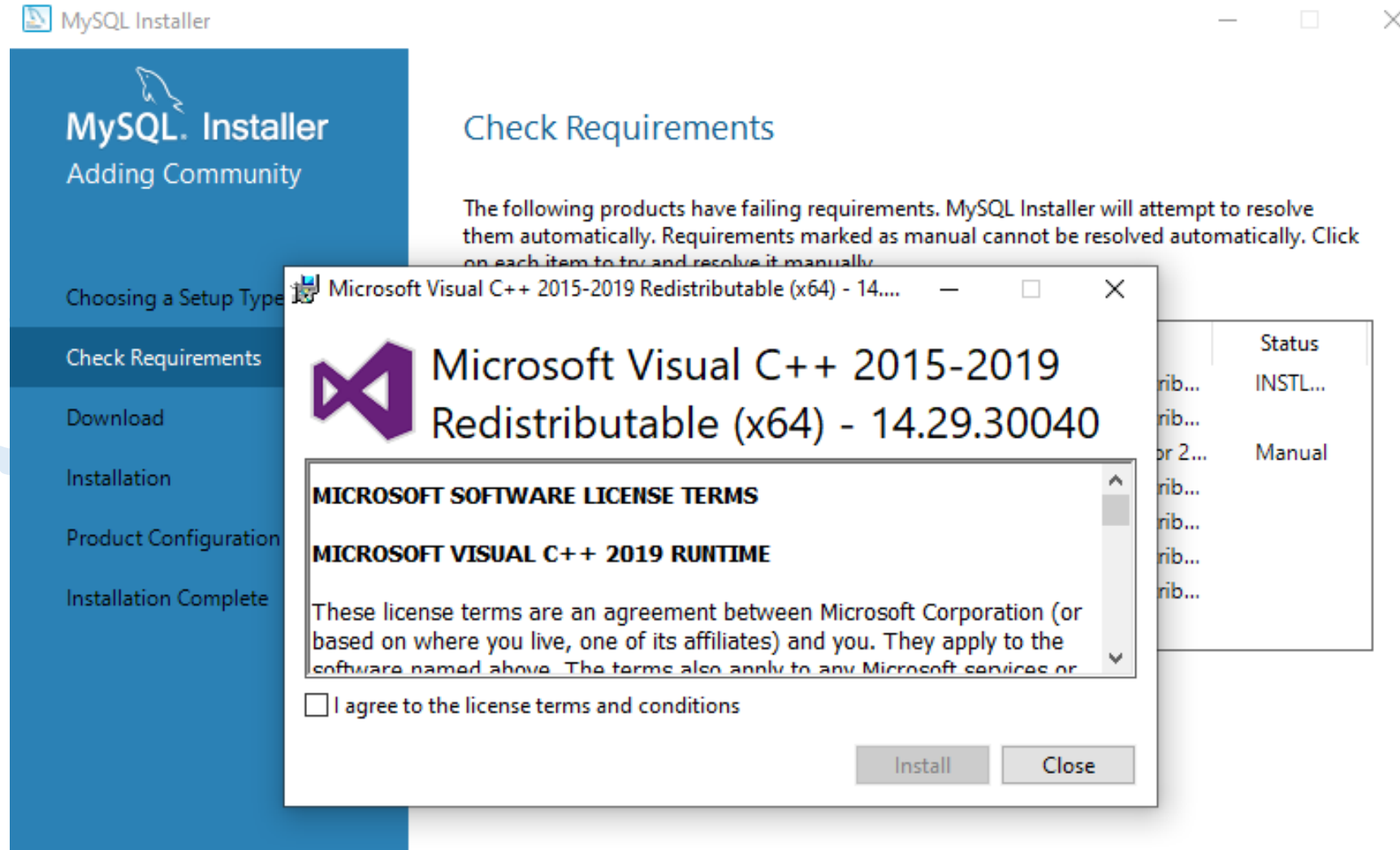
The following products have failing requirements. MySQL Installer will attempt to resolve them automatically. Requirements marked as manual cannot be resolved automatically. Click on each item to try and resolve it manually.

For Product	Requirement	Status
<input type="radio"/> MySQL Server 8.0.26	Microsoft Visual C++ 2019 Redistrib...	
<input type="radio"/> MySQL Workbench 8.0.26	Microsoft Visual C++ 2019 Redistrib...	
<input type="radio"/> MySQL for Visual Studio 1.2.10	Visual Studio version 2015, 2017 or 2...	Manual
<input type="radio"/> MySQL Shell 8.0.26	Microsoft Visual C++ 2019 Redistrib...	
<input type="radio"/> MySQL Router 8.0.26	Microsoft Visual C++ 2019 Redistrib...	
<input type="radio"/> Connector/ODBC 8.0.26	Microsoft Visual C++ 2019 Redistrib...	
<input type="radio"/> Connector/C++ 8.0.26	Microsoft Visual C++ 2017 Redistrib...	

ki

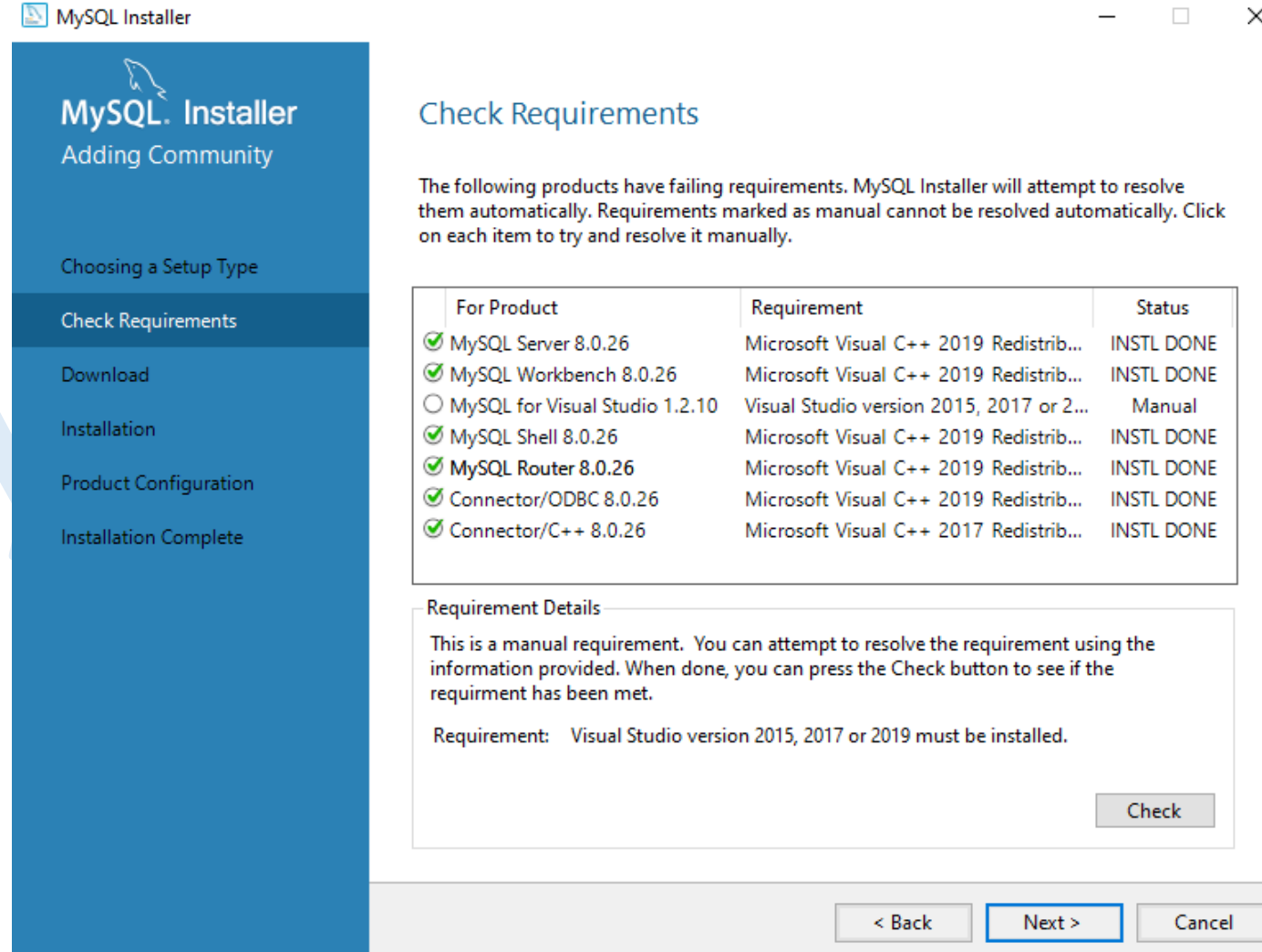
Installation

Select I agree,
then Next



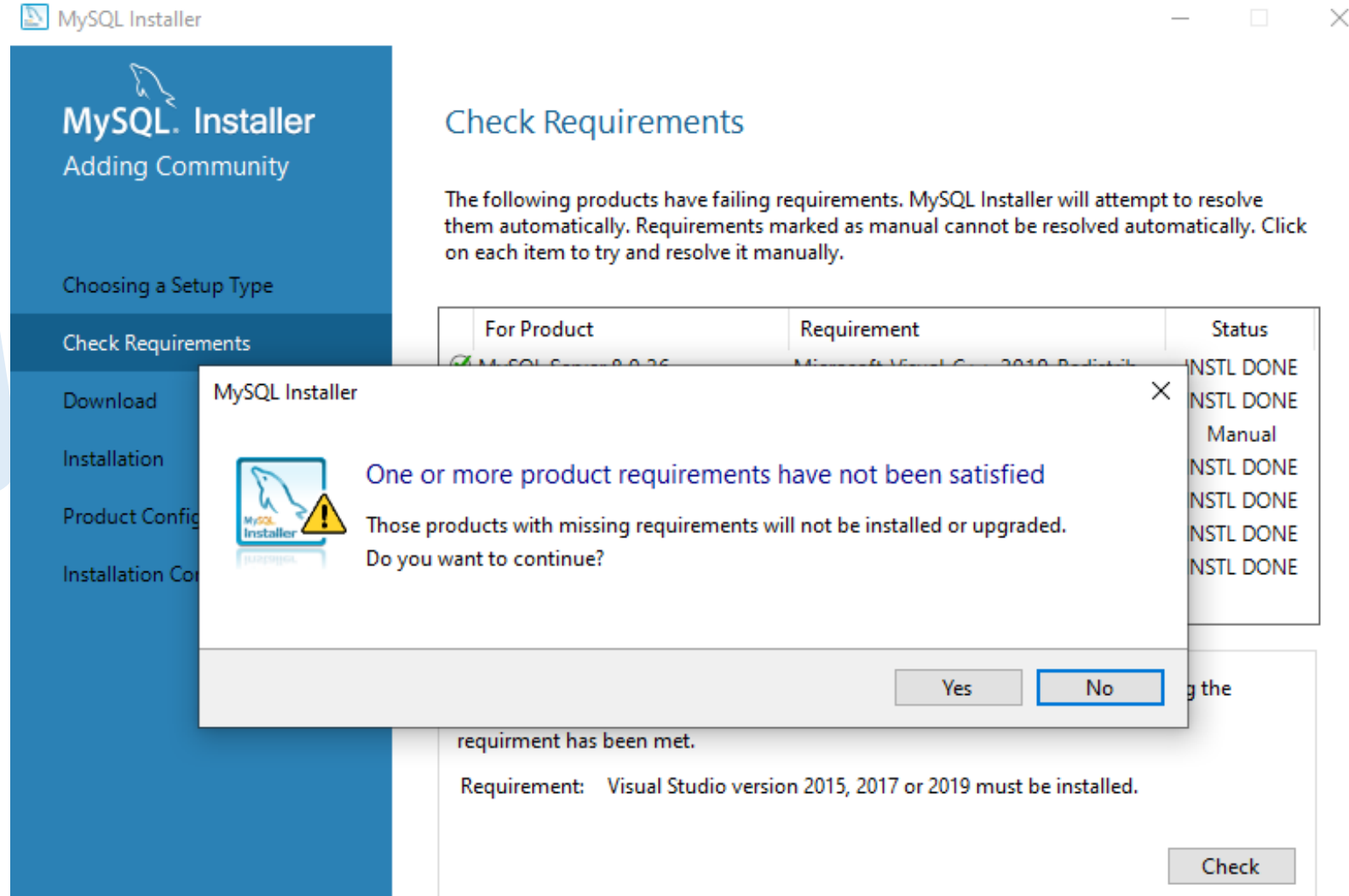
Installation

Select Next



Installation

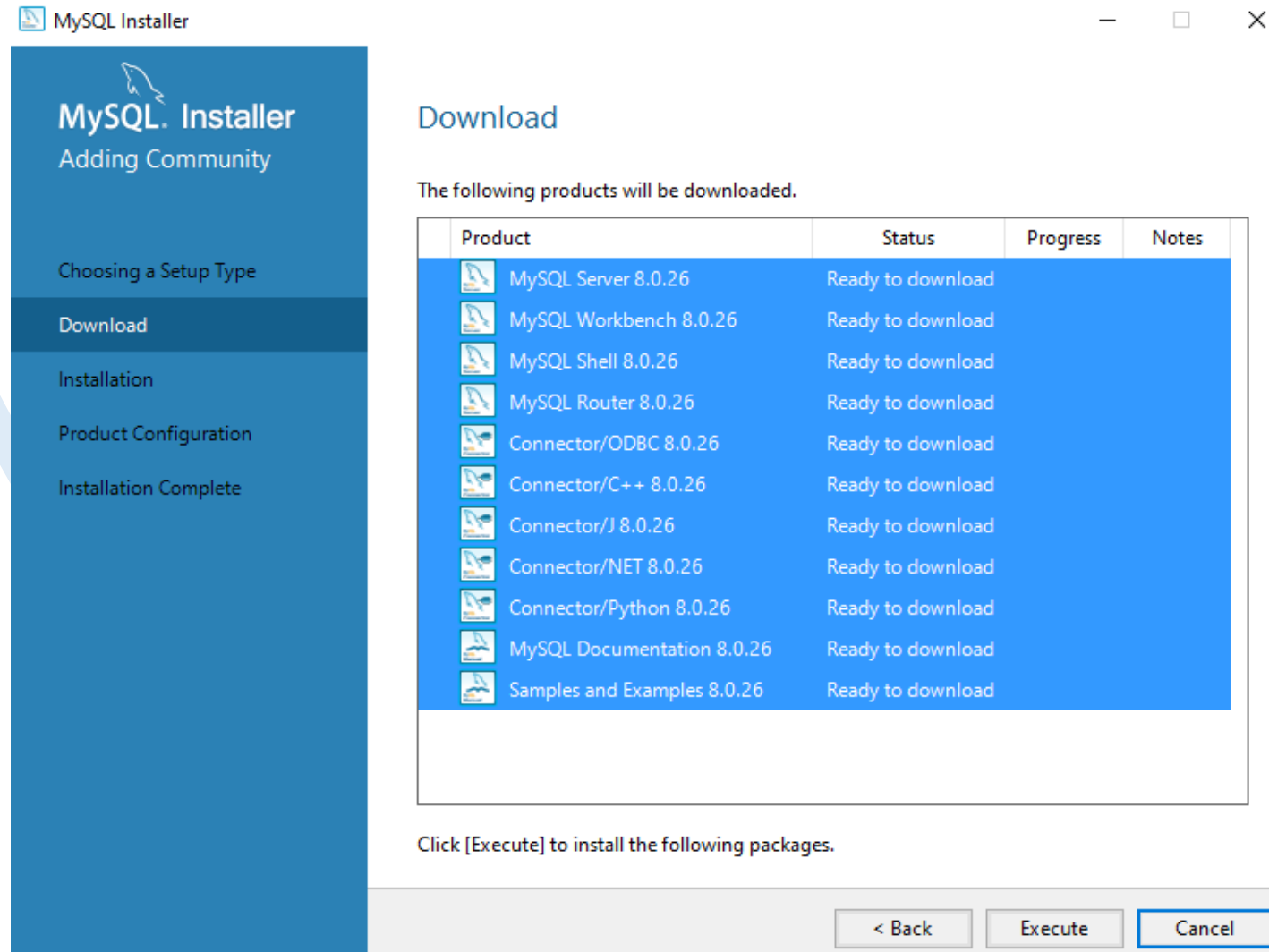
Select Yes



aki

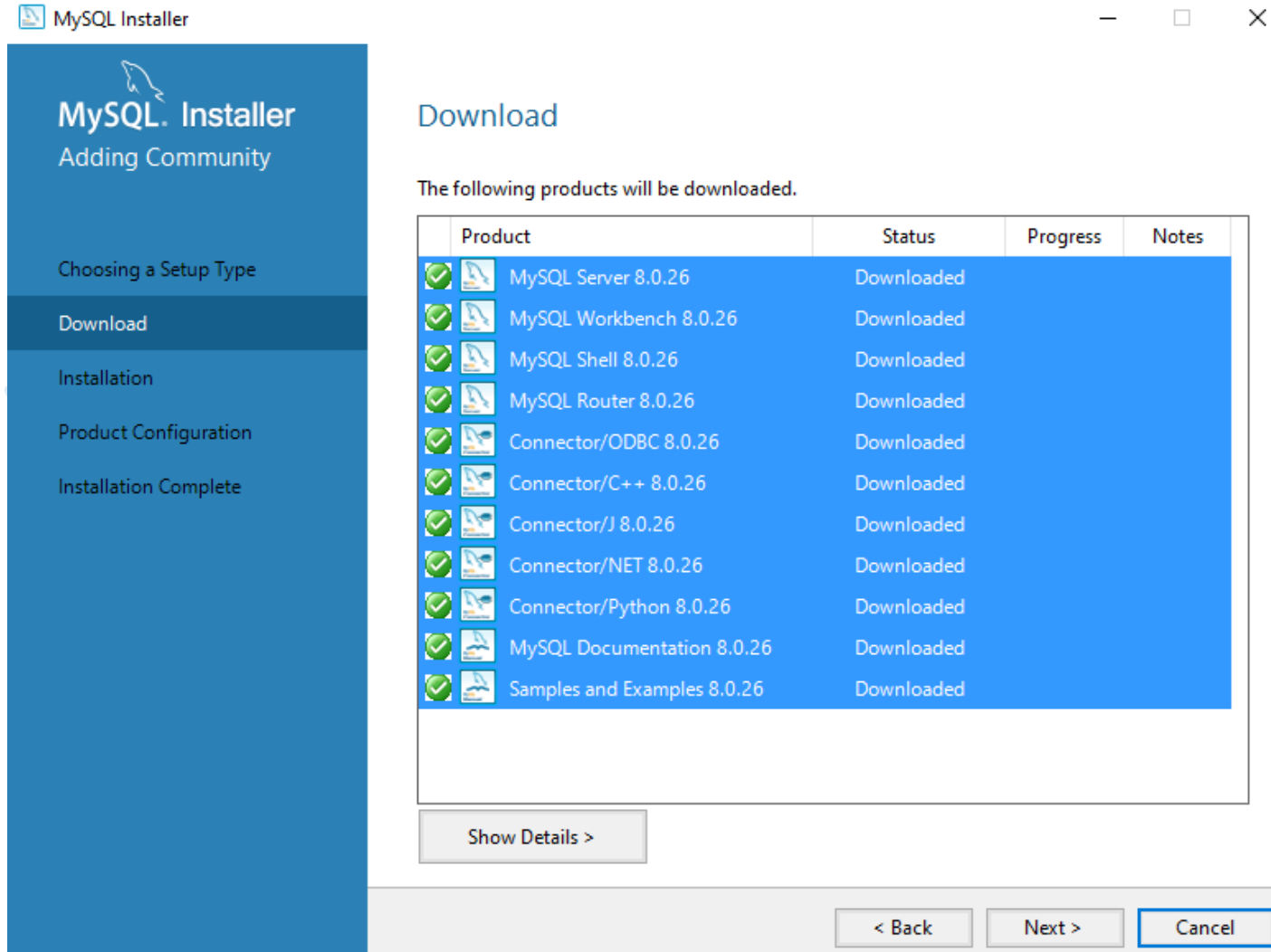
Installation

Select Execute



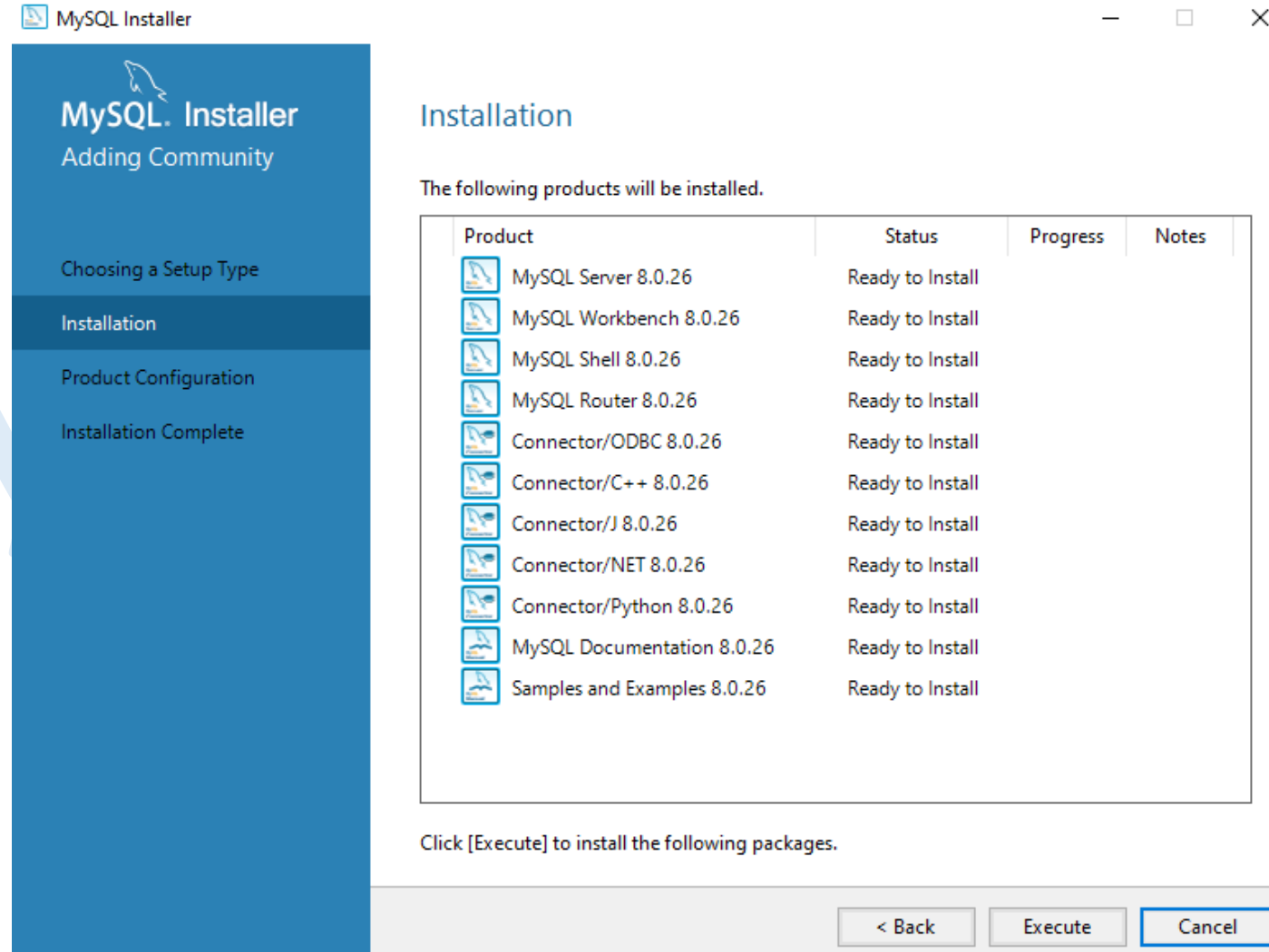
Installation

Select Next



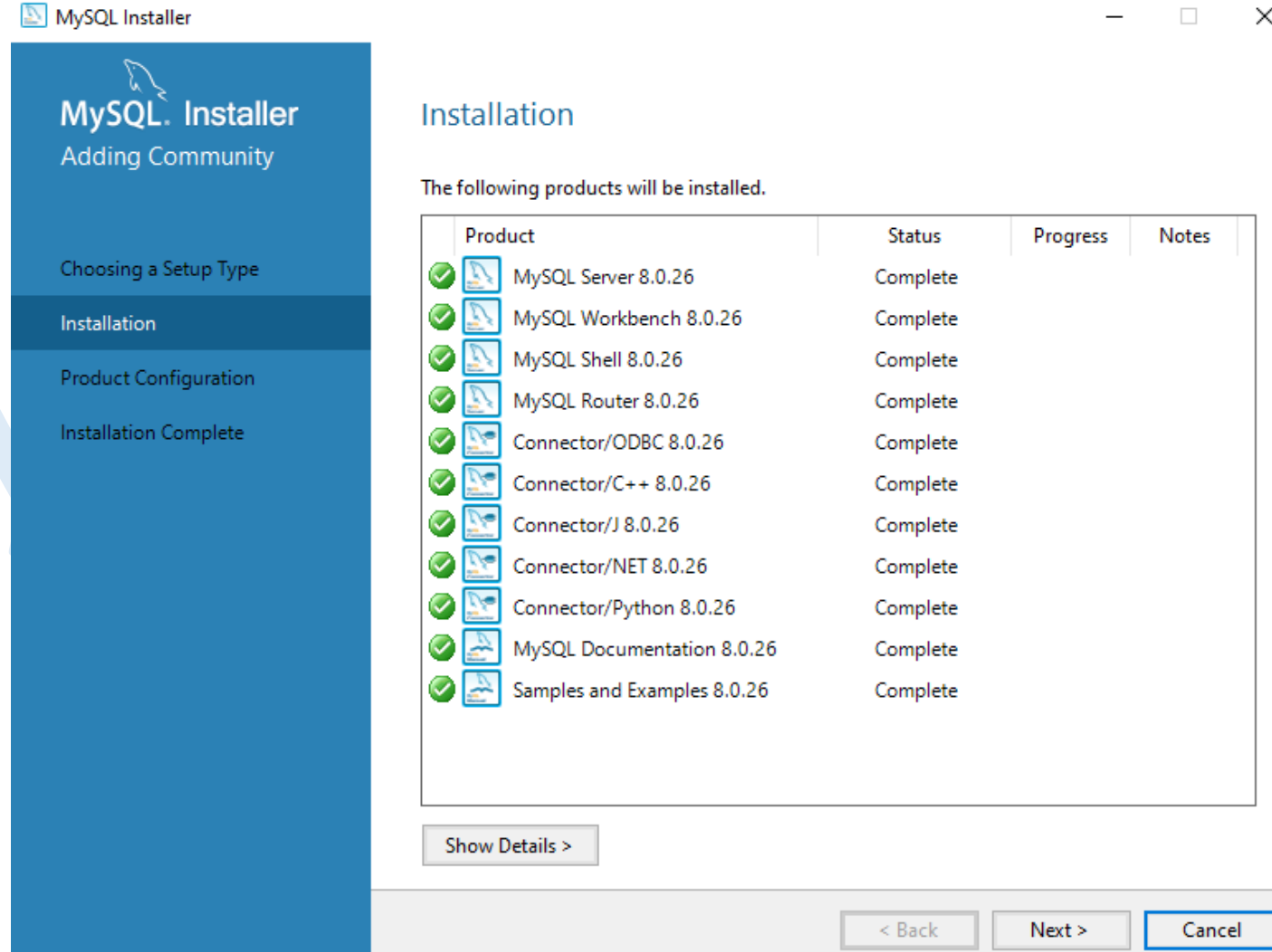
Installation

Select Execute



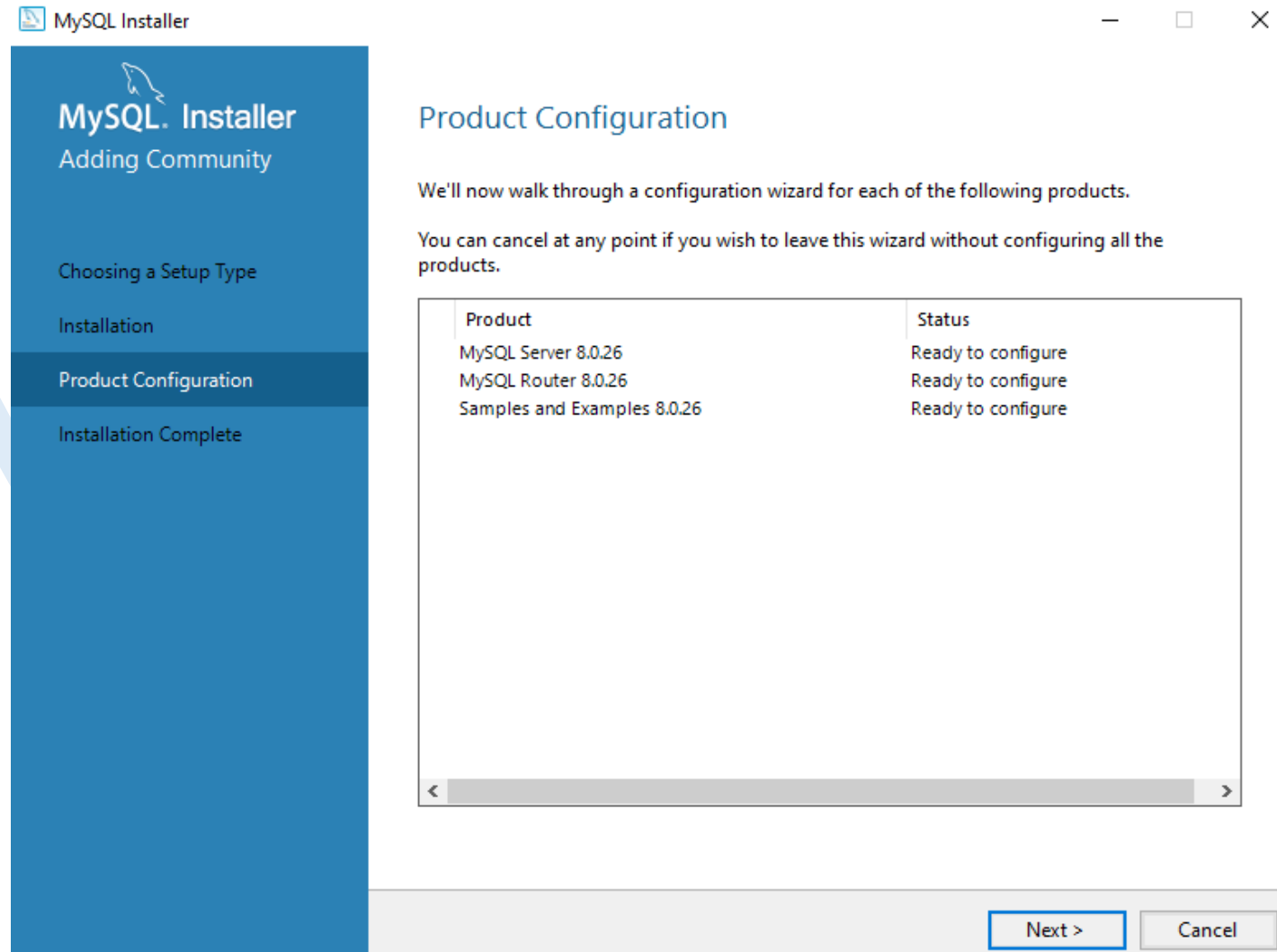
Installation

Select Next



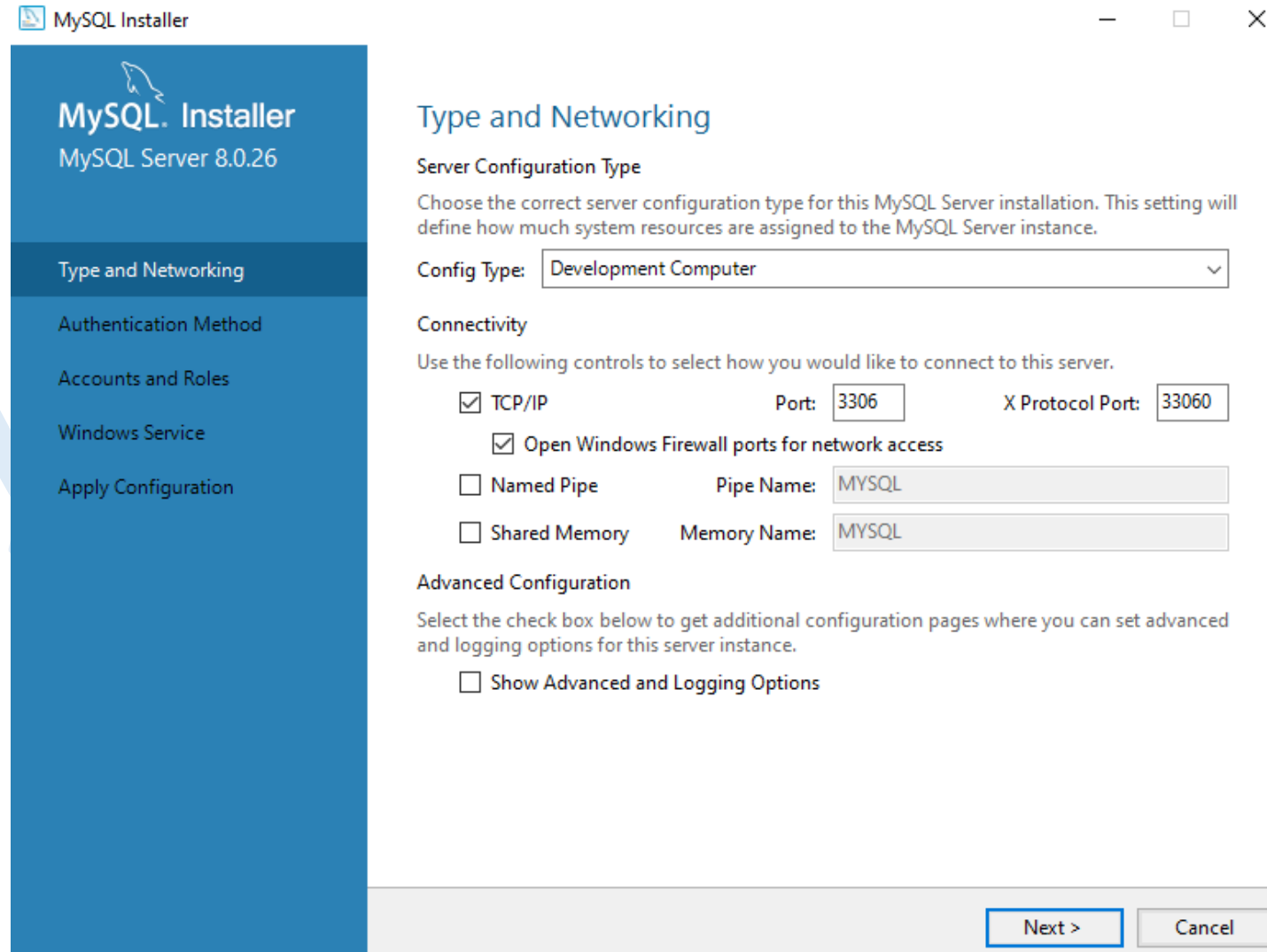
Installation

Select Next



Installation

Select Next



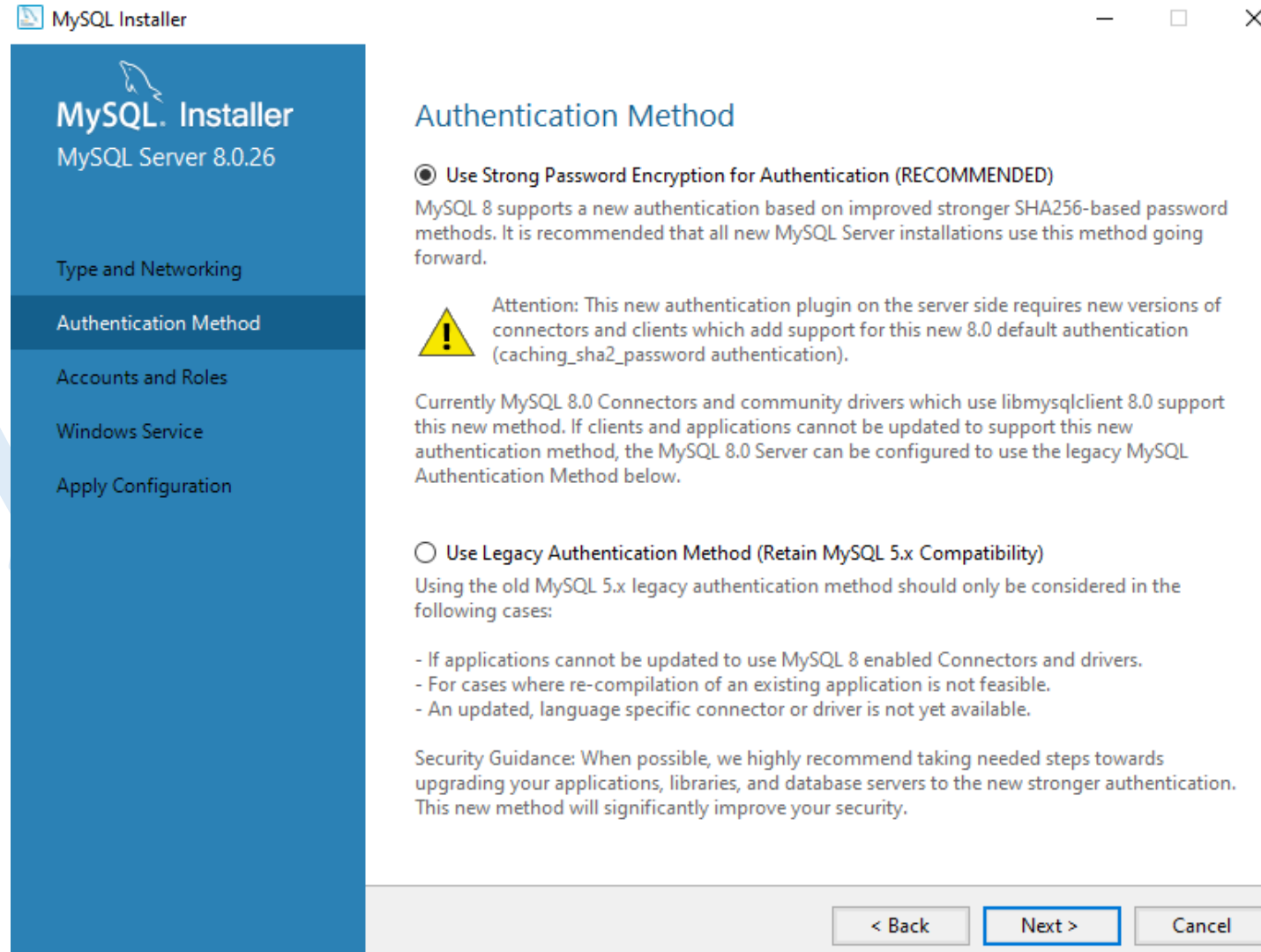
The image shows the MySQL Installer window for MySQL Server 8.0.26. The left sidebar contains the following steps: Type and Networking (selected), Authentication Method, Accounts and Roles, Windows Service, and Apply Configuration. The main area is titled 'Type and Networking' and contains the following sections:

- 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:
- Connectivity**
Use the following controls to select how you would like to connect to this server.
 - ☒ TCP/IP
Port: X Protocol Port:
 - ☒ Open Windows Firewall ports for network access
 - ☐ Named Pipe
Pipe Name:
 - ☐ Shared Memory
Memory Name:
- 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

At the bottom right, there are two buttons: 'Next >' and 'Cancel'.

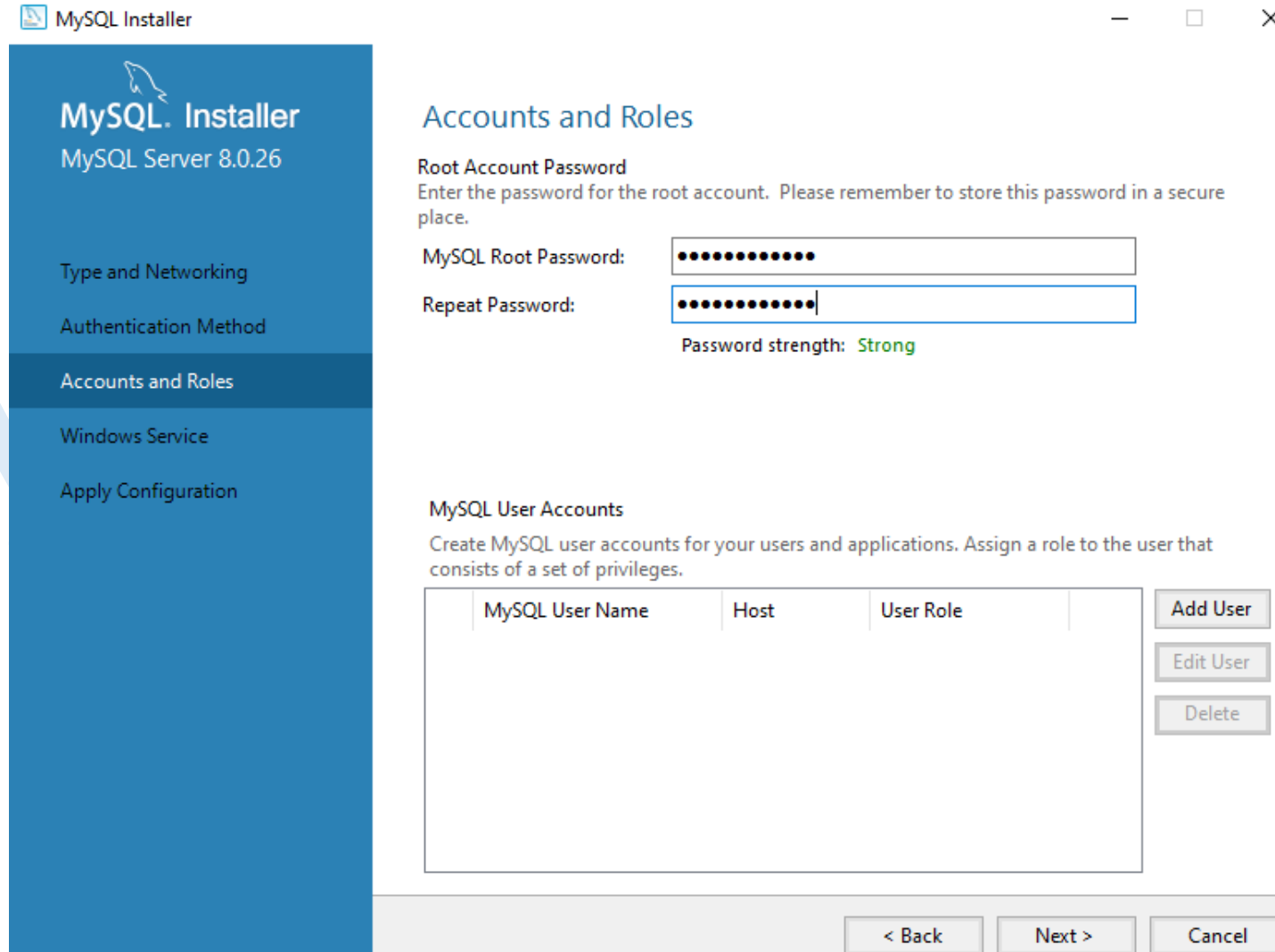
Installation

Select Next



Installation

Select Next



The image shows the MySQL Installer window for MySQL Server 8.0.26. The left sidebar contains the following steps: Type and Networking, Authentication Method, Accounts and Roles (selected), Windows Service, and Apply Configuration. The main area is titled 'Accounts and Roles' and contains two sections: 'Root Account Password' and 'MySQL User Accounts'.

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:

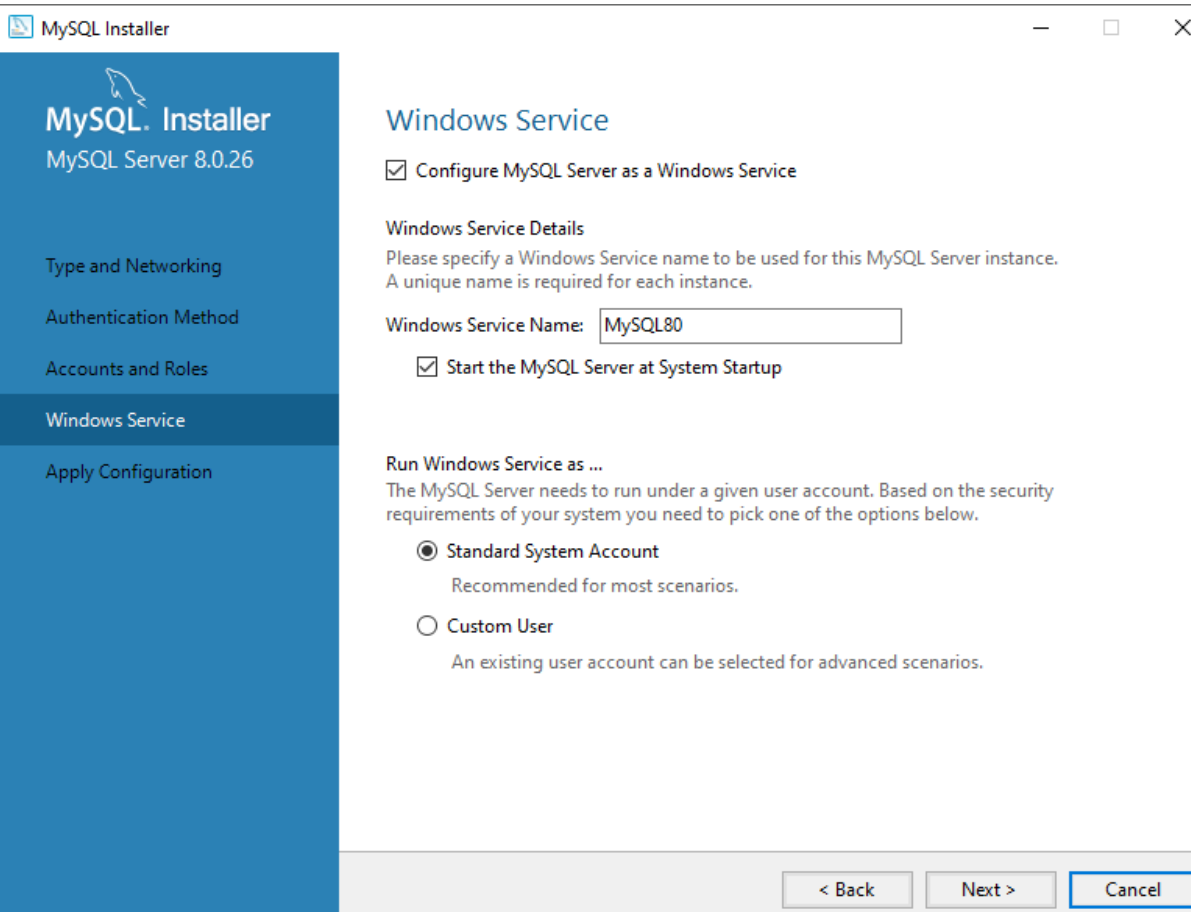
Password strength: **Strong**

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
-----------------	------	-----------

Installation

Select Next



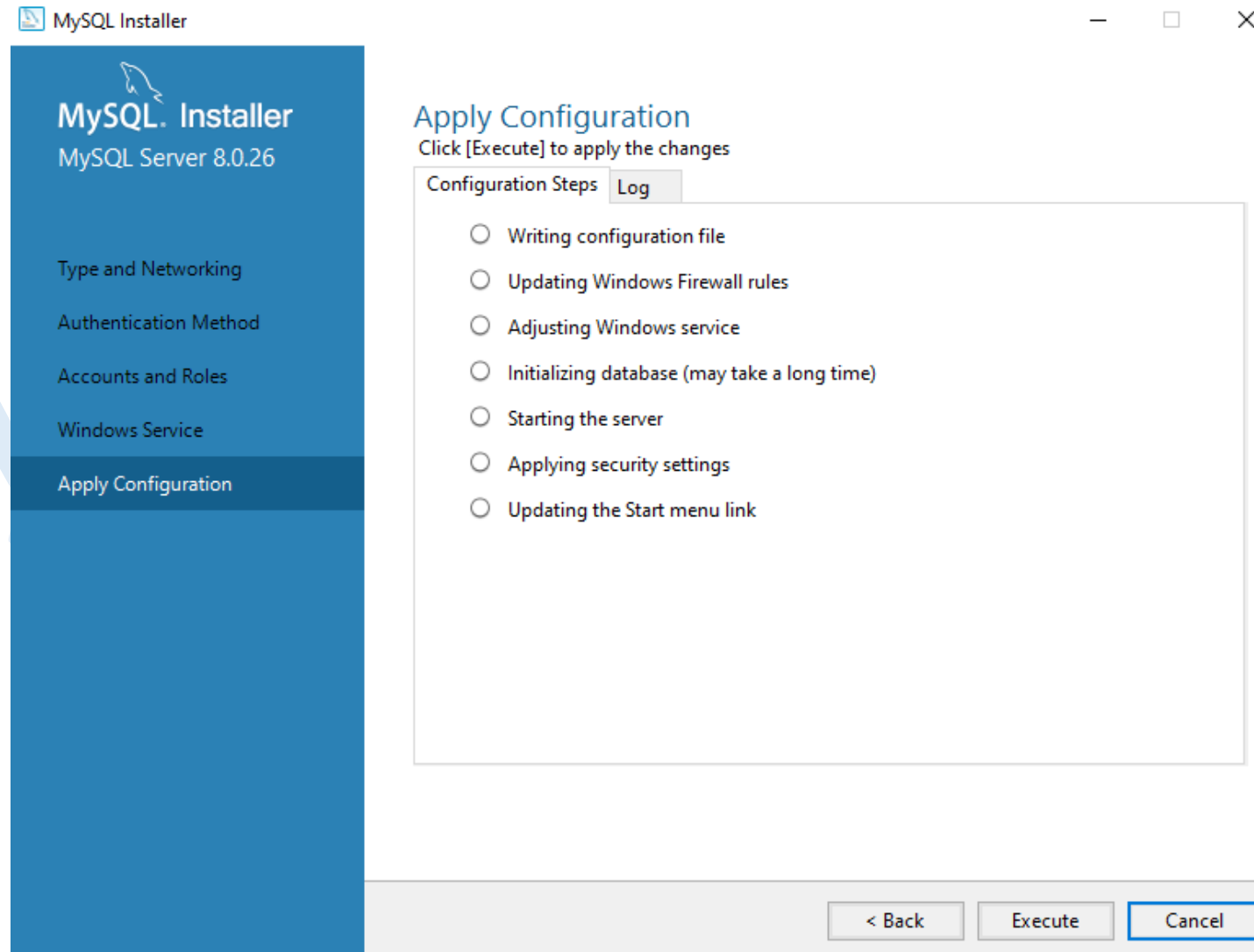
The image shows the MySQL Installer window for MySQL Server 8.0.26. The left sidebar contains a list of configuration steps: Type and Networking, Authentication Method, Accounts and Roles, Windows Service (which is currently selected and highlighted in a darker blue), and Apply Configuration. The main area of the window is titled 'Windows Service' and contains the following options:

- ☒ 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:
- ☒ 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.

At the bottom right of the window, there are three buttons: '< Back', 'Next >', and 'Cancel'.

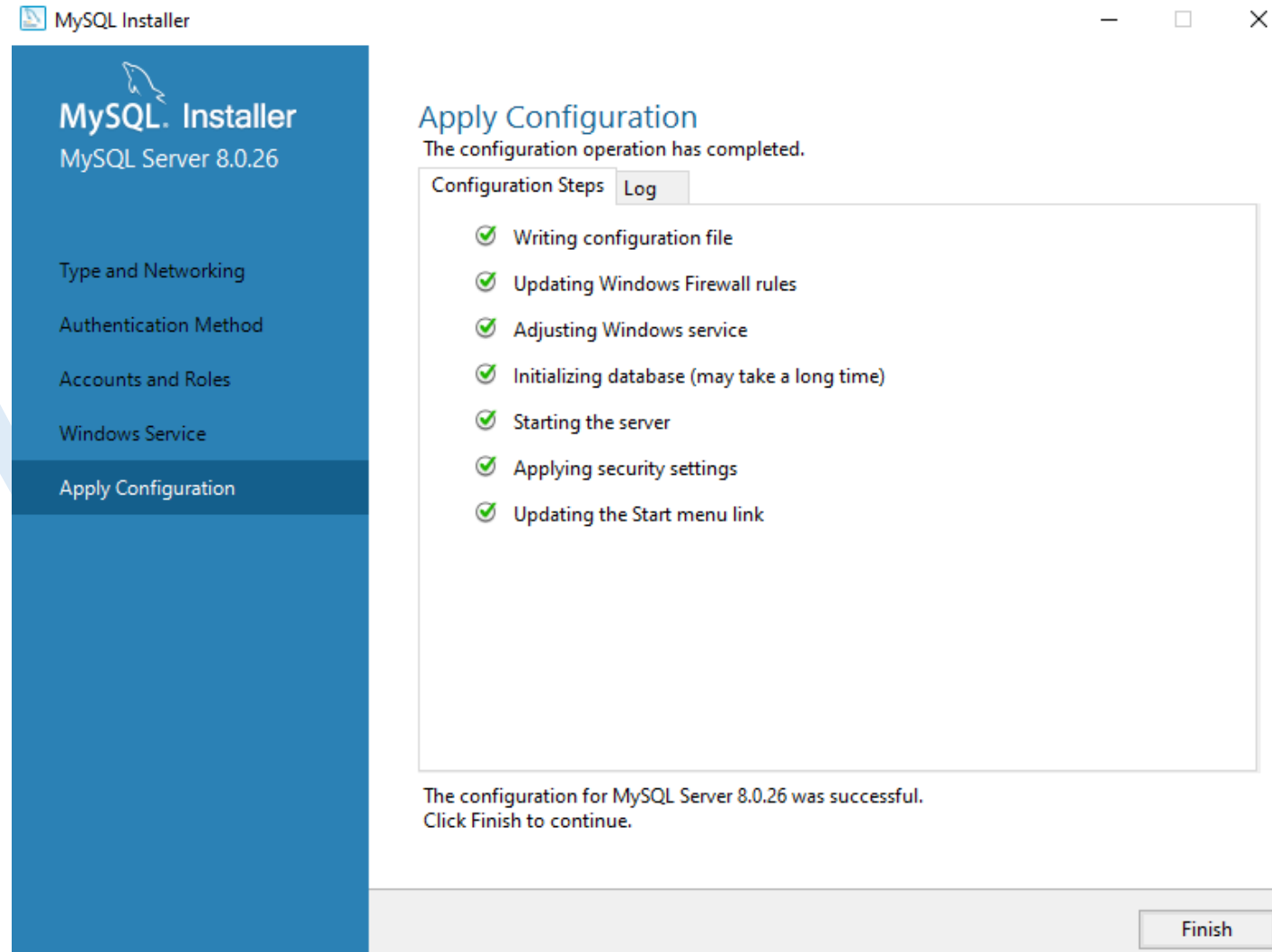
Installation

Select Execute



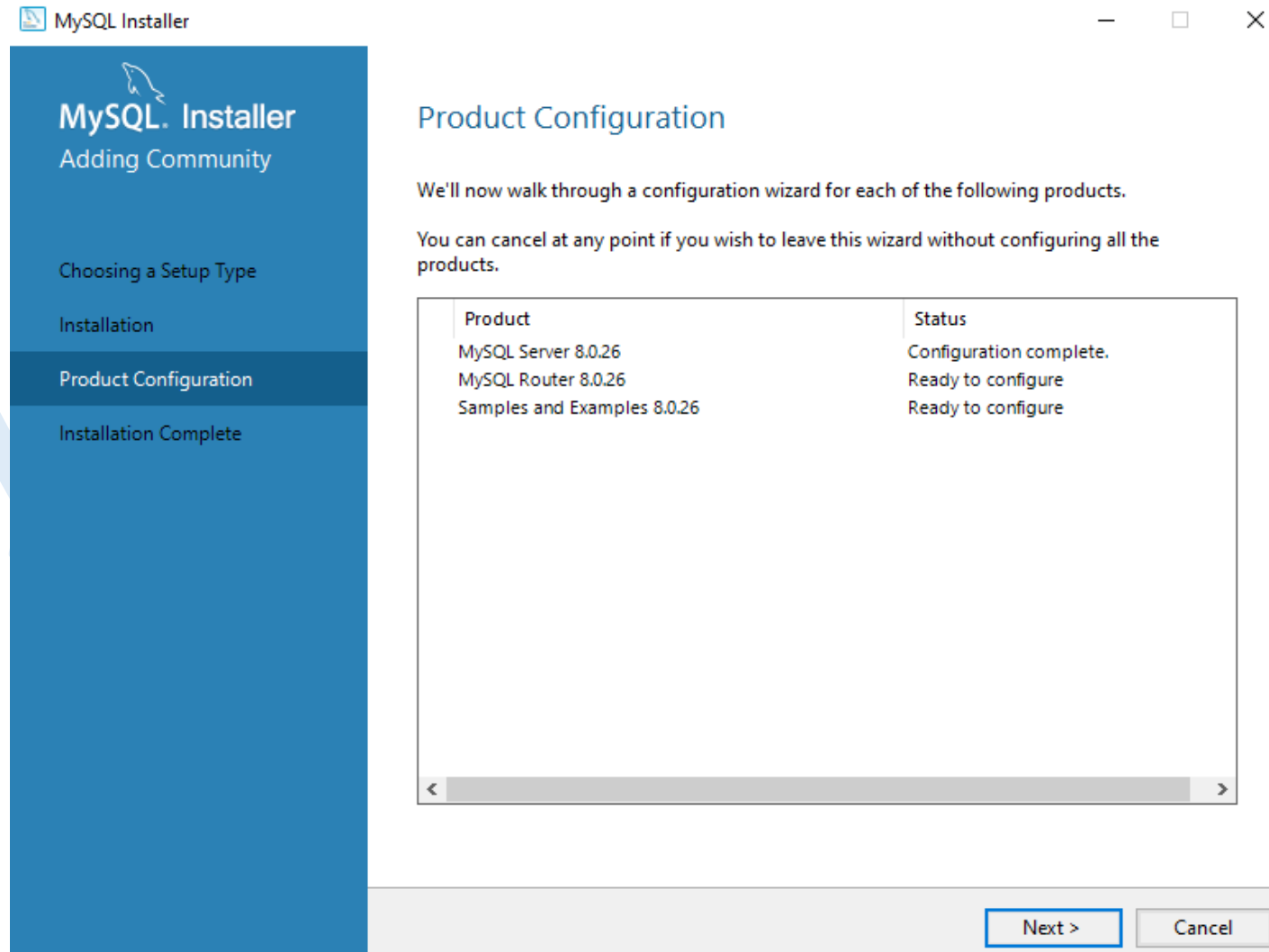
Installation

Select Finish



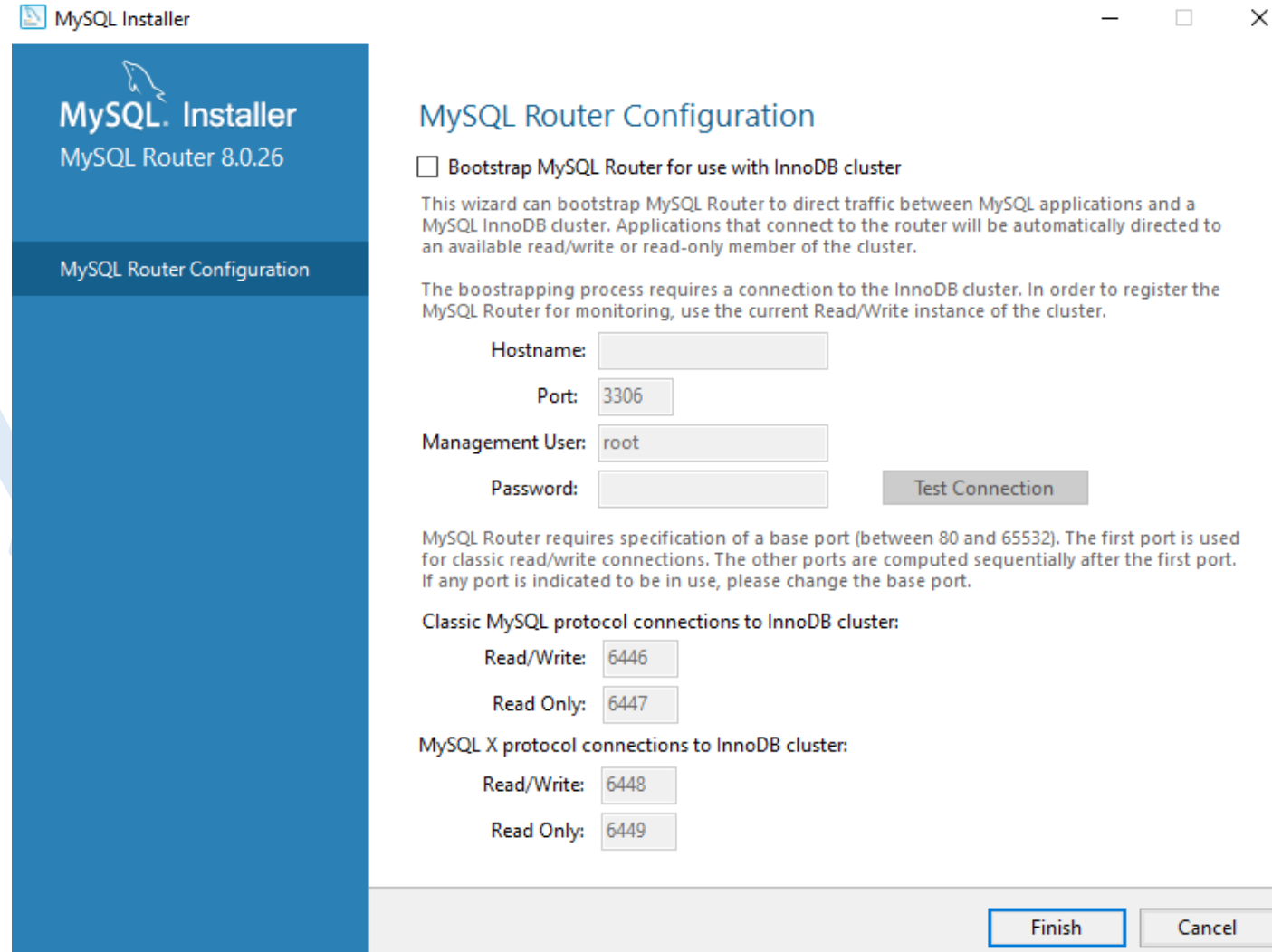
Installation

Select Next



Installation

Select Finish

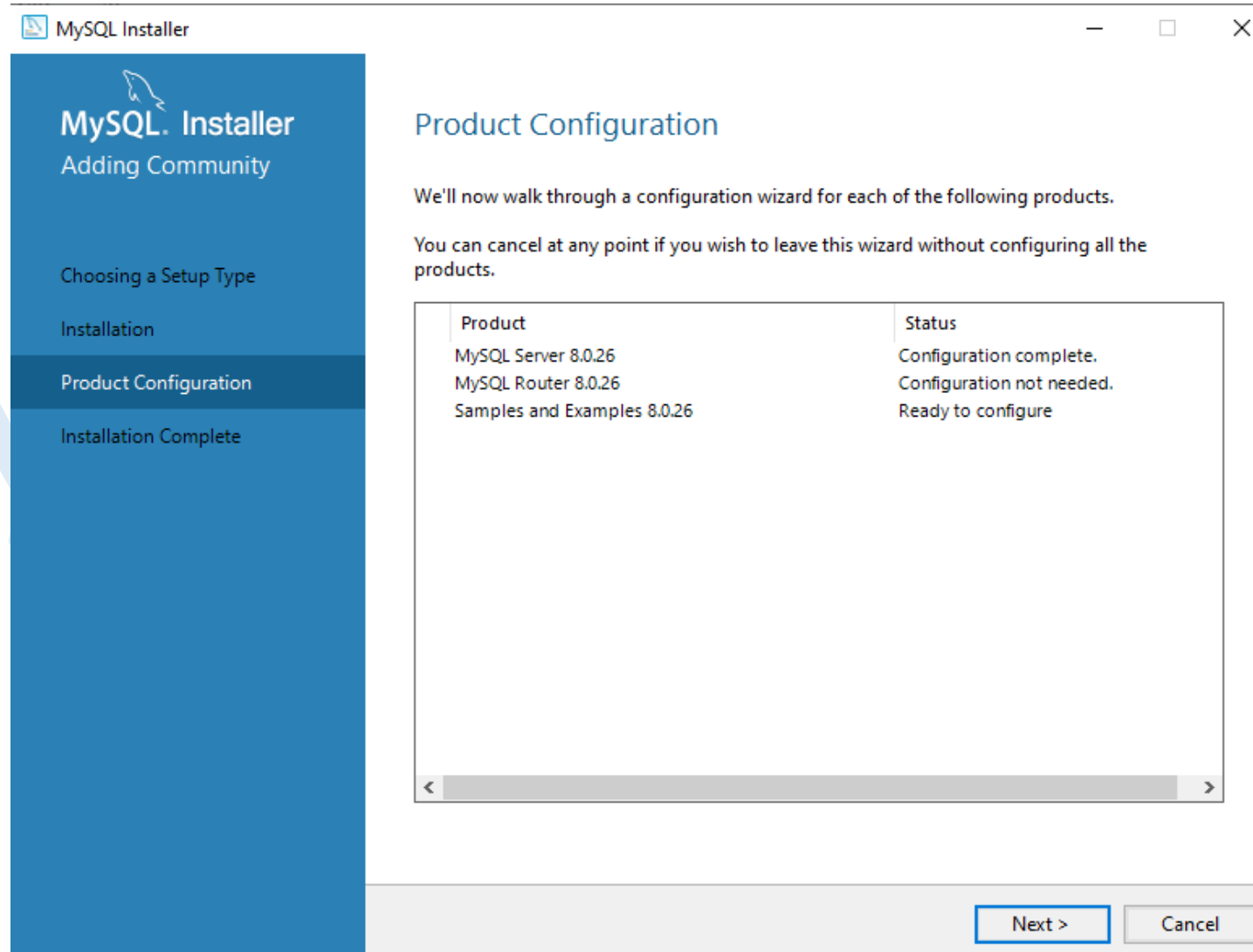


The image shows a screenshot of the MySQL Installer application window. The window title is "MySQL Installer". The main content area is titled "MySQL Router Configuration". On the left side, there is a blue sidebar with the MySQL logo and the text "MySQL Installer" and "MySQL Router 8.0.26". Below this, the text "MySQL Router Configuration" is visible. The main configuration area on the right contains the following elements:

- A checkbox labeled "Bootstrap MySQL Router for use with InnoDB cluster" which is currently unchecked.
- A paragraph of text: "This wizard can bootstrap MySQL Router to direct traffic between MySQL applications and a MySQL InnoDB cluster. Applications that connect to the router will be automatically directed to an available read/write or read-only member of the cluster."
- Another paragraph of text: "The bootstrapping process requires a connection to the InnoDB cluster. In order to register the MySQL Router for monitoring, use the current Read/Write instance of the cluster."
- Input fields for "Hostname:", "Port:" (with the value "3306"), "Management User:" (with the value "root"), and "Password:".
- A "Test Connection" button.
- A paragraph of text: "MySQL Router requires specification of a base port (between 80 and 65532). The first port is used for classic read/write connections. The other ports are computed sequentially after the first port. If any port is indicated to be in use, please change the base port."
- A section titled "Classic MySQL protocol connections to InnoDB cluster:" with input fields for "Read/Write:" (6446) and "Read Only:" (6447).
- A section titled "MySQL X protocol connections to InnoDB cluster:" with input fields for "Read/Write:" (6448) and "Read Only:" (6449).
- At the bottom right, there are "Finish" and "Cancel" buttons.

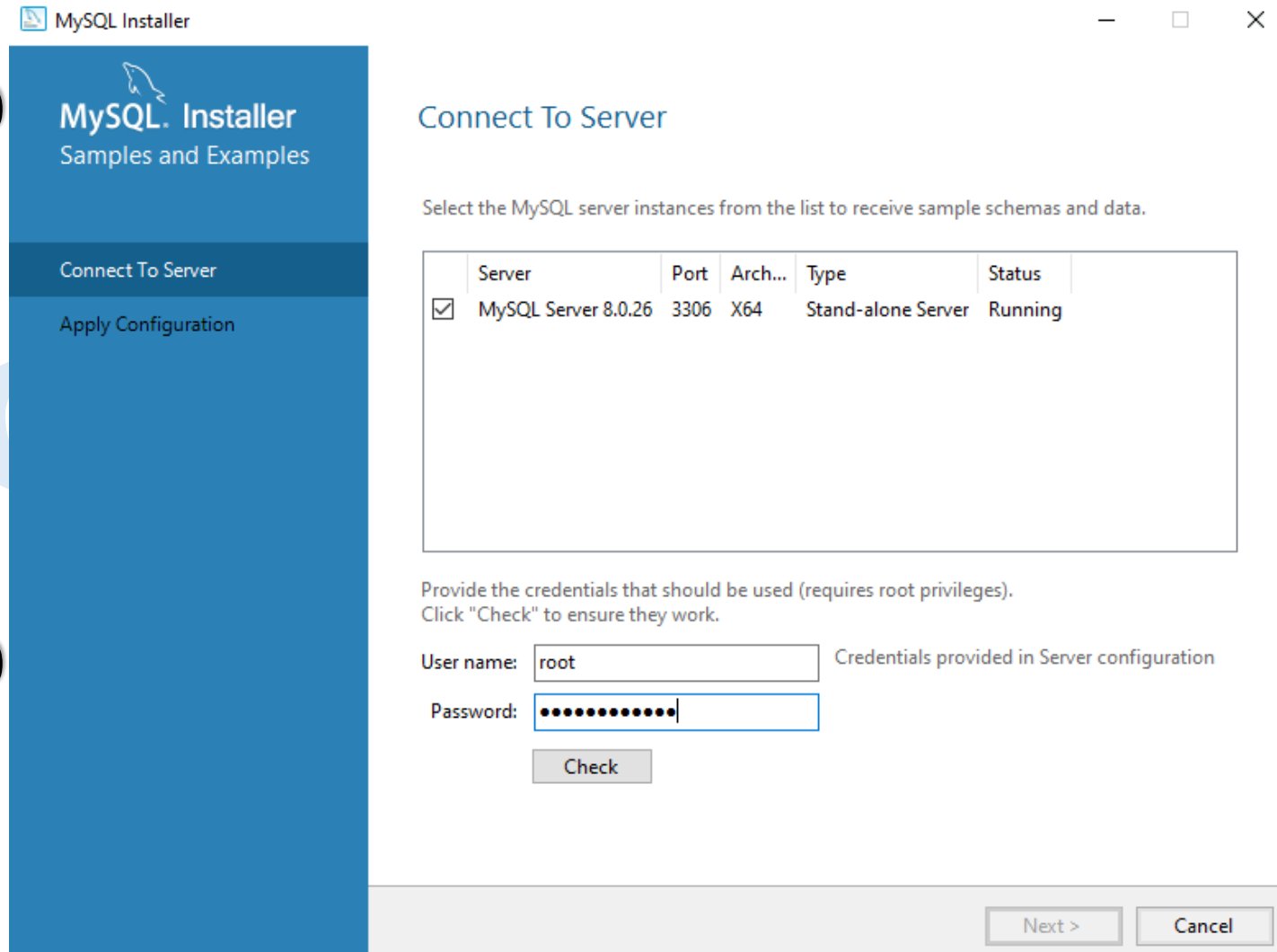
Installation

Select Next



Installation

Type the same
password you
already entered in
your Server
Configuration stage



The image shows the 'MySQL Installer' window, specifically the 'Connect To Server' tab. The window has a blue header with the MySQL logo and the text 'MySQL. Installer Samples and Examples'. Below the header, there are two buttons: 'Connect To Server' (highlighted) and 'Apply Configuration'. The main content area is titled 'Connect To Server' and contains the instruction: 'Select the MySQL server instances from the list to receive sample schemas and data.' Below this is a table with columns: 'Server', 'Port', 'Arch...', 'Type', and 'Status'. There is one row in the table with a checked checkbox, 'MySQL Server 8.0.26', '3306', 'X64', 'Stand-alone Server', and 'Running'. Below the table, there is a text box for 'User name' containing 'root' and a 'Check' button. To the right of the 'User name' field is the text 'Credentials provided in Server configuration'. Below the 'User name' field is a 'Password' field with masked characters (dots) and a 'Check' button. At the bottom right of the window are 'Next >' and 'Cancel' buttons.

MySQL. Installer
Samples and Examples

Connect To Server

Apply Configuration

Connect To Server

Select the MySQL server instances from the list to receive sample schemas and data.

Server	Port	Arch...	Type	Status
<input checked="" type="checkbox"/> MySQL Server 8.0.26	3306	X64	Stand-alone Server	Running

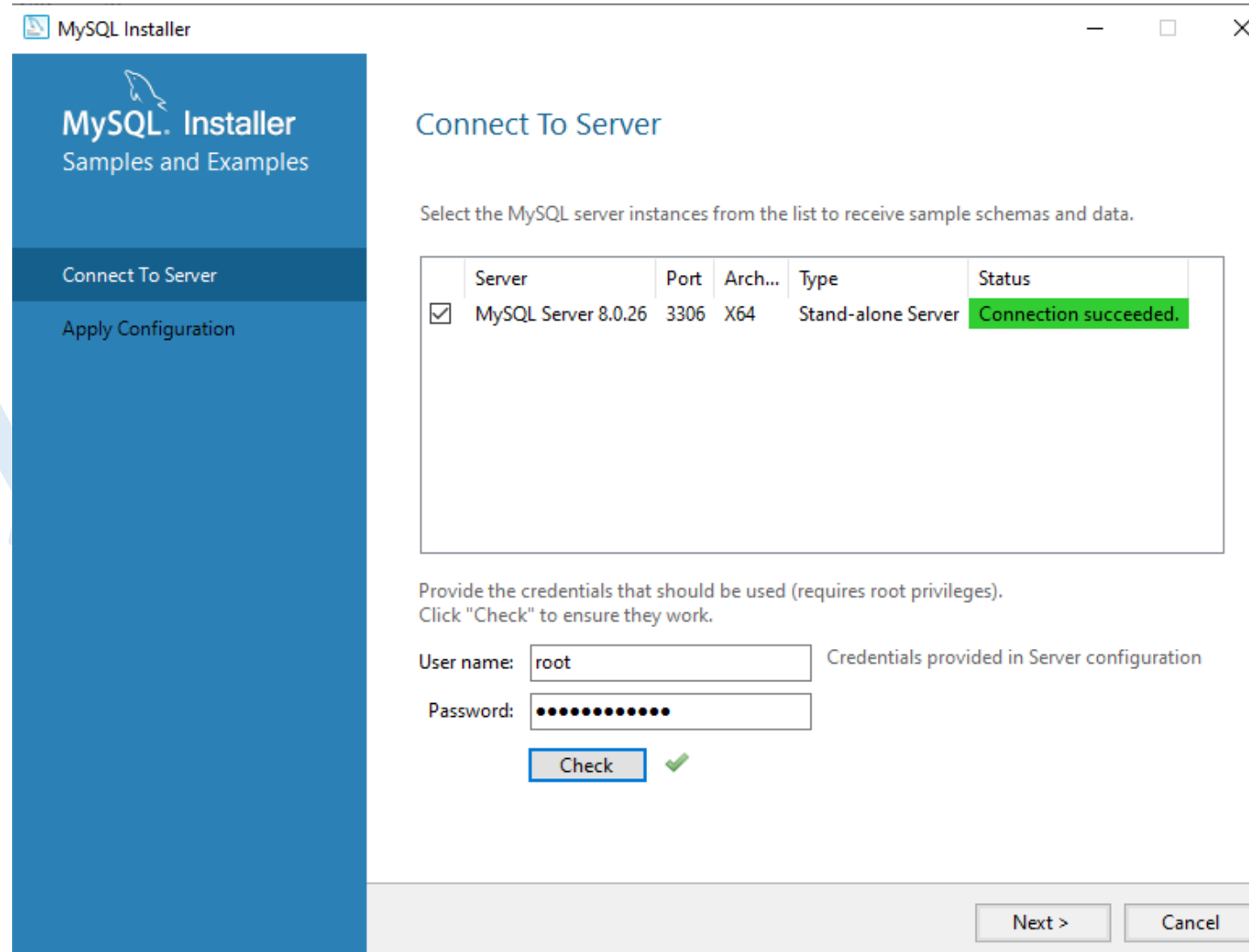
Provide the credentials that should be used (requires root privileges).
Click "Check" to ensure they work.

User name: Credentials provided in Server configuration

Password:

Installation

Select Check and
then Next



The image shows the 'Connect To Server' window of the MySQL Installer. On the left is a blue sidebar with the MySQL logo and the text 'MySQL. Installer Samples and Examples'. Below this are two buttons: 'Connect To Server' (highlighted) and 'Apply Configuration'. The main area is titled 'Connect To Server' and contains instructions to select MySQL server instances. A table lists one instance: 'MySQL Server 8.0.26' on port '3306', 'X64' architecture, 'Stand-alone Server' type, with a status of 'Connection succeeded.' (highlighted in green). Below the table, there is a section for credentials with the text 'Provide the credentials that should be used (requires root privileges). Click "Check" to ensure they work.' The 'User name' field contains 'root' and the 'Password' field is masked with dots. A 'Check' button with a green checkmark icon is next to the password field. At the bottom right are 'Next >' and 'Cancel' buttons.

MySQL. Installer
Samples and Examples

Connect To Server

Apply Configuration

Connect To Server


Select the MySQL server instances from the list to receive sample schemas and data.

	Server	Port	Arch...	Type	Status
<input checked="" type="checkbox"/>	MySQL Server 8.0.26	3306	X64	Stand-alone Server	Connection succeeded.

Provide the credentials that should be used (requires root privileges).
Click "Check" to ensure they work.

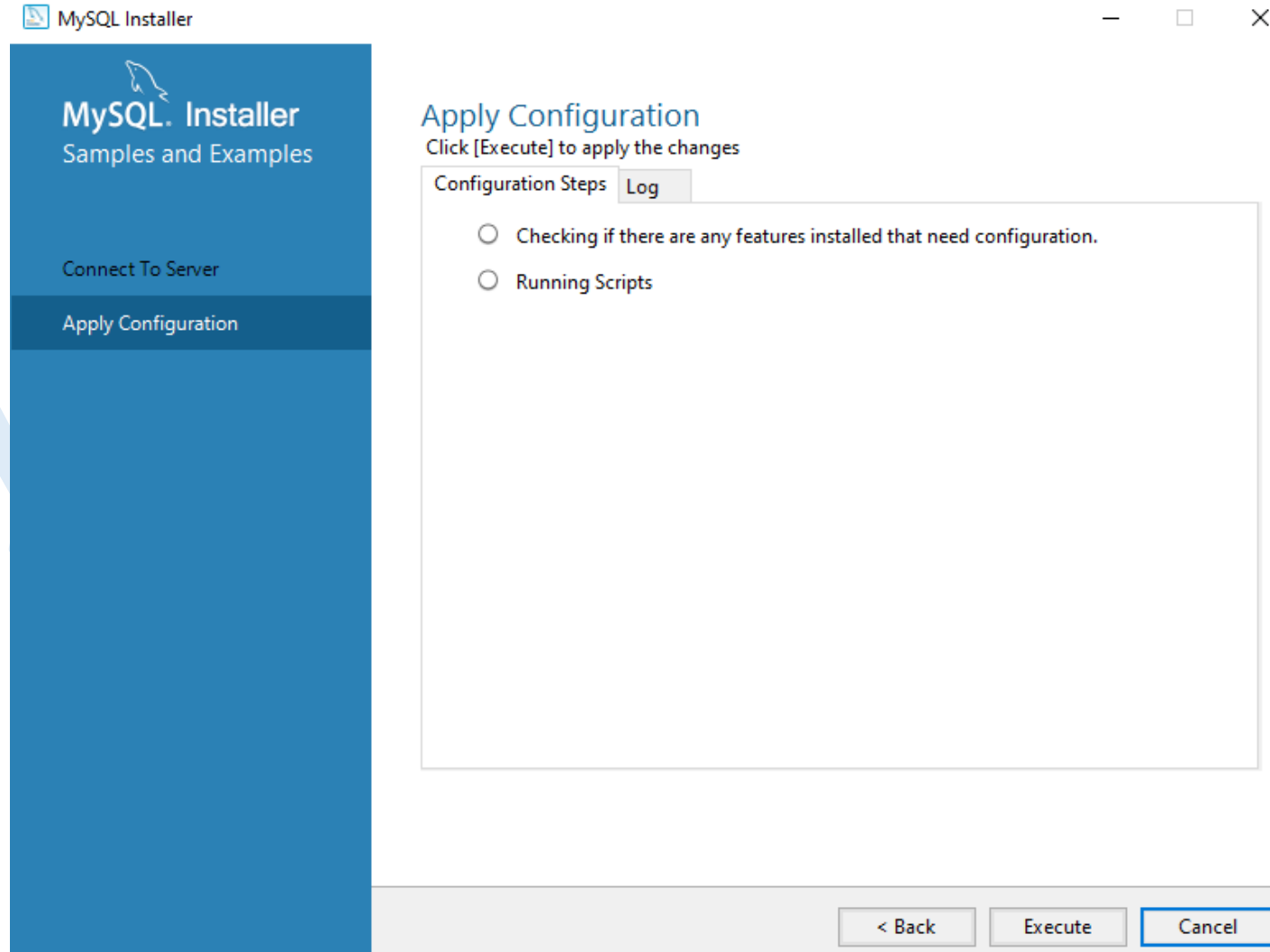
User name: Credentials provided in Server configuration

Password:



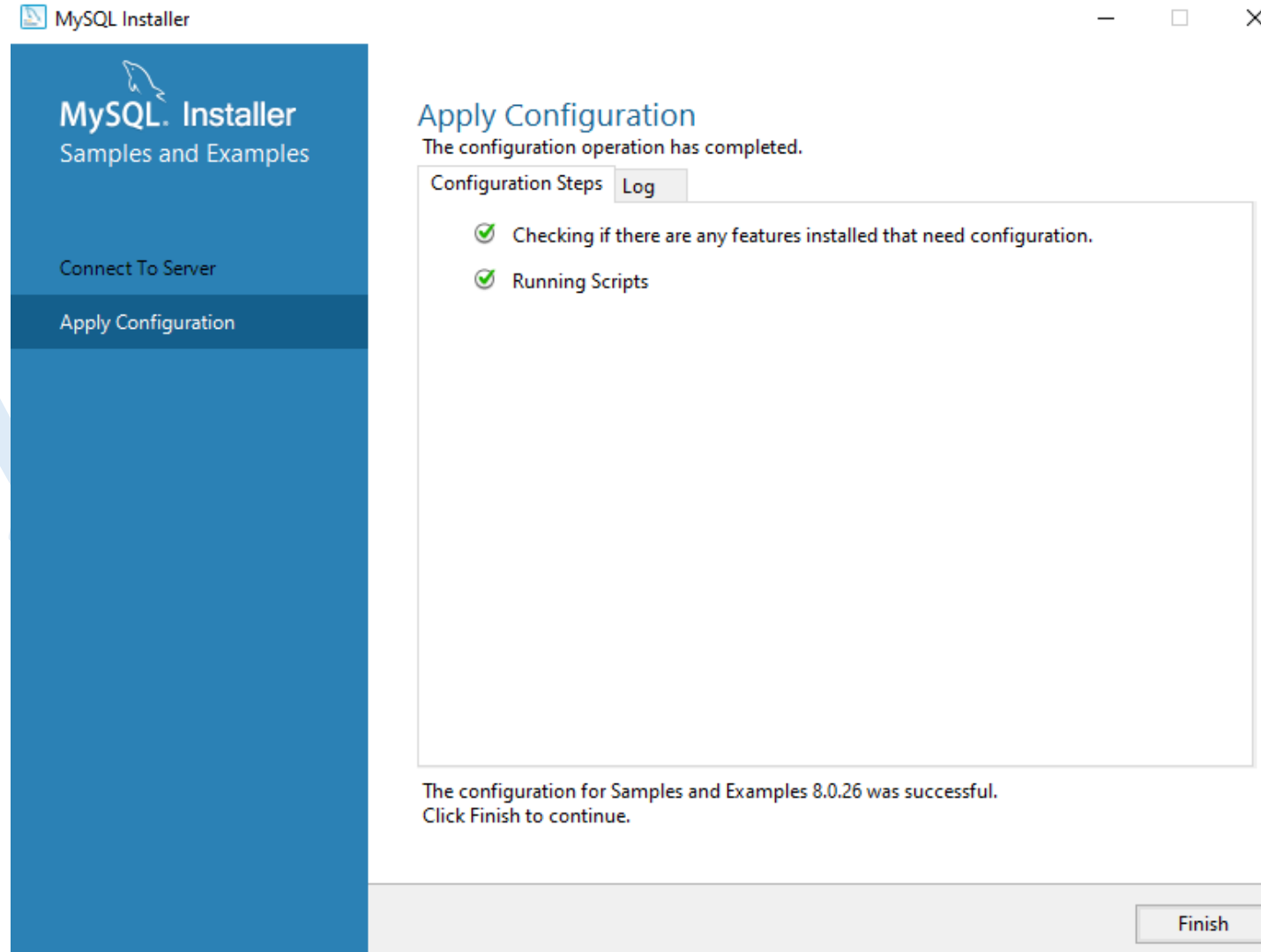
Installation

Select Execute



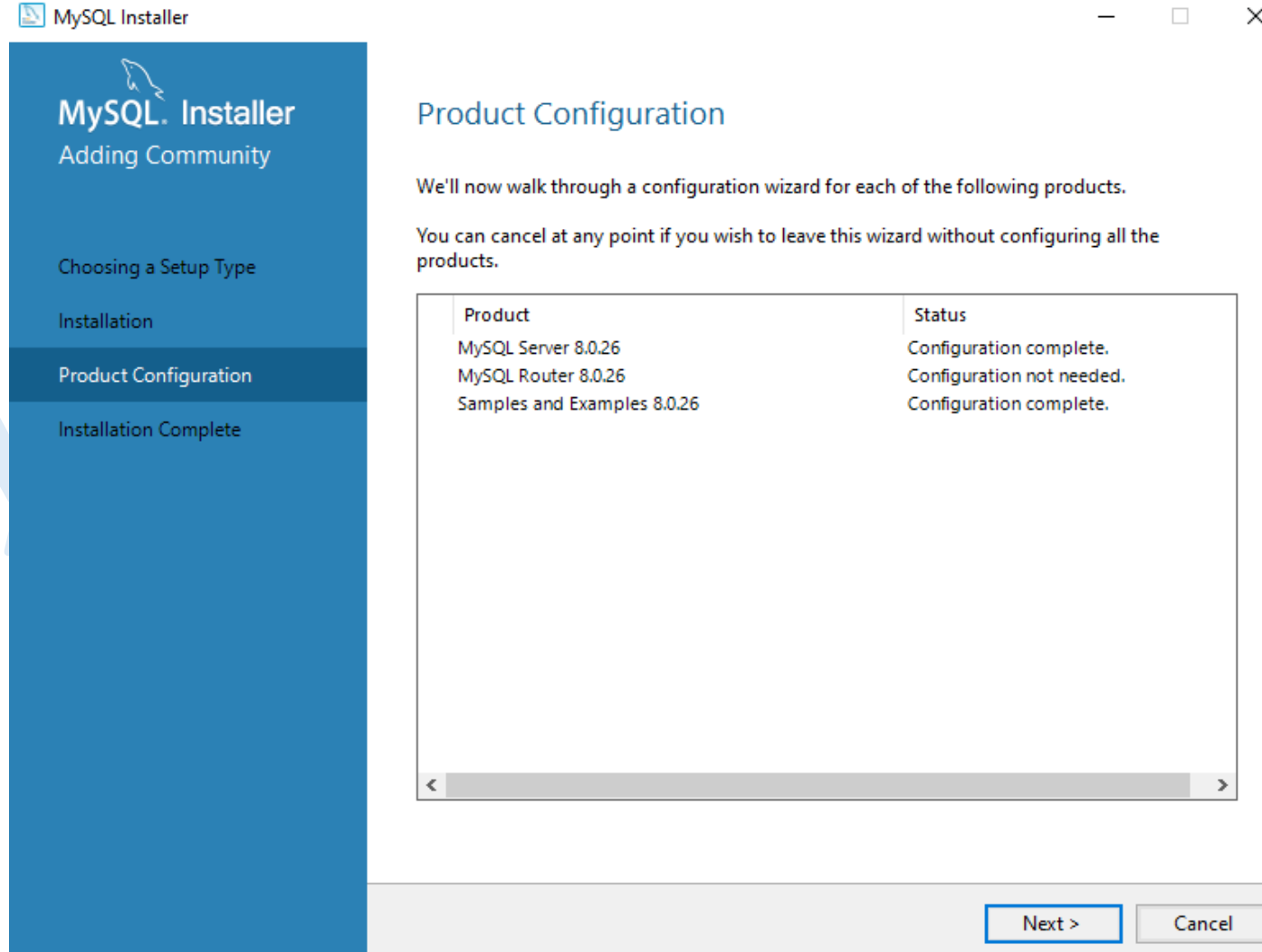
Installation

Select Finish



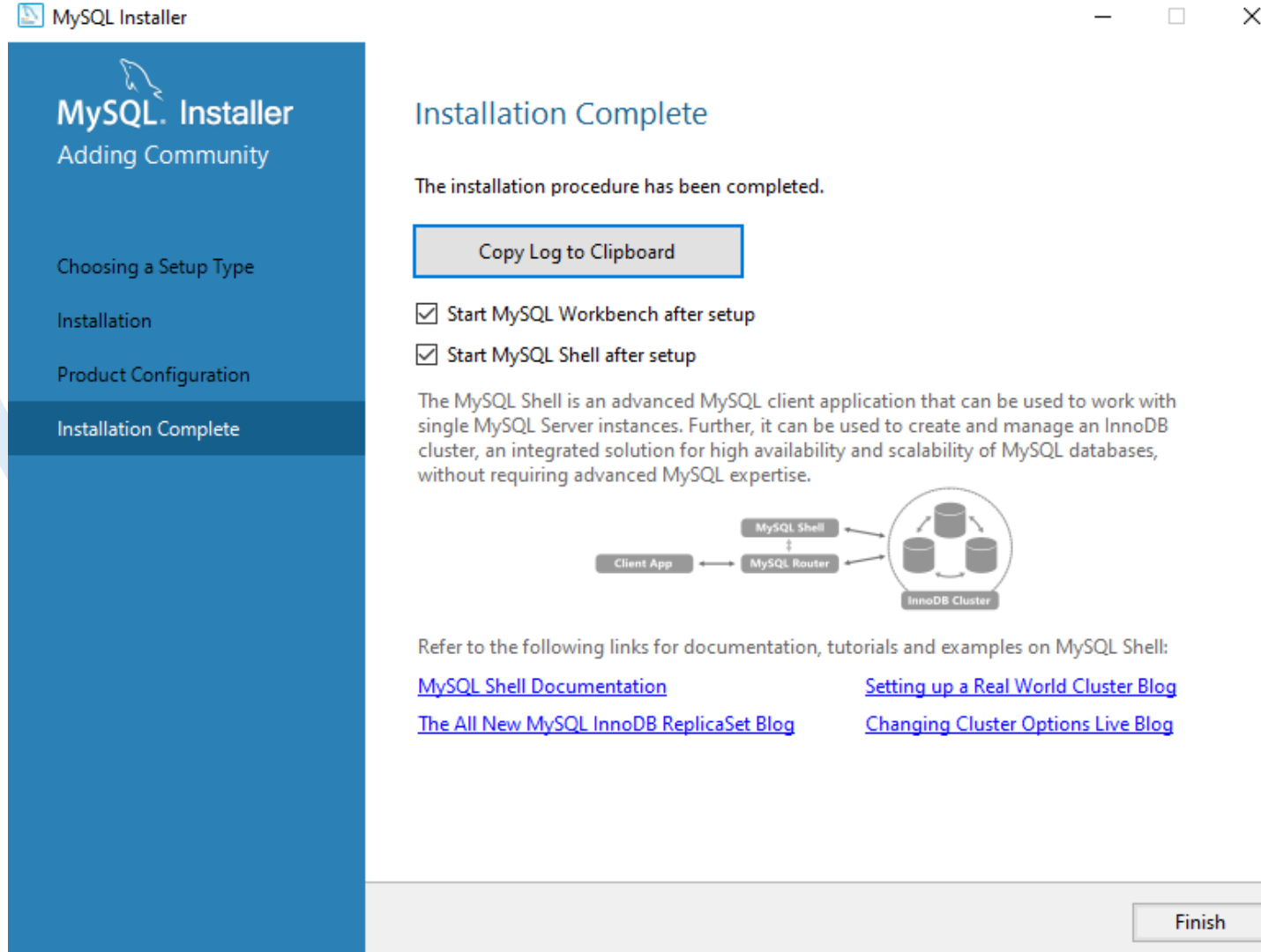
Installation

Select Next



Installation

Select Finish



Installation: View of Shell

 C:\Program Files\MySQL\MySQL Shell 8.0\bin\mysqlsh.exe

MySQL Shell 8.0.26

Copyright (c) 2016, 2021, Oracle and/or its affiliates.

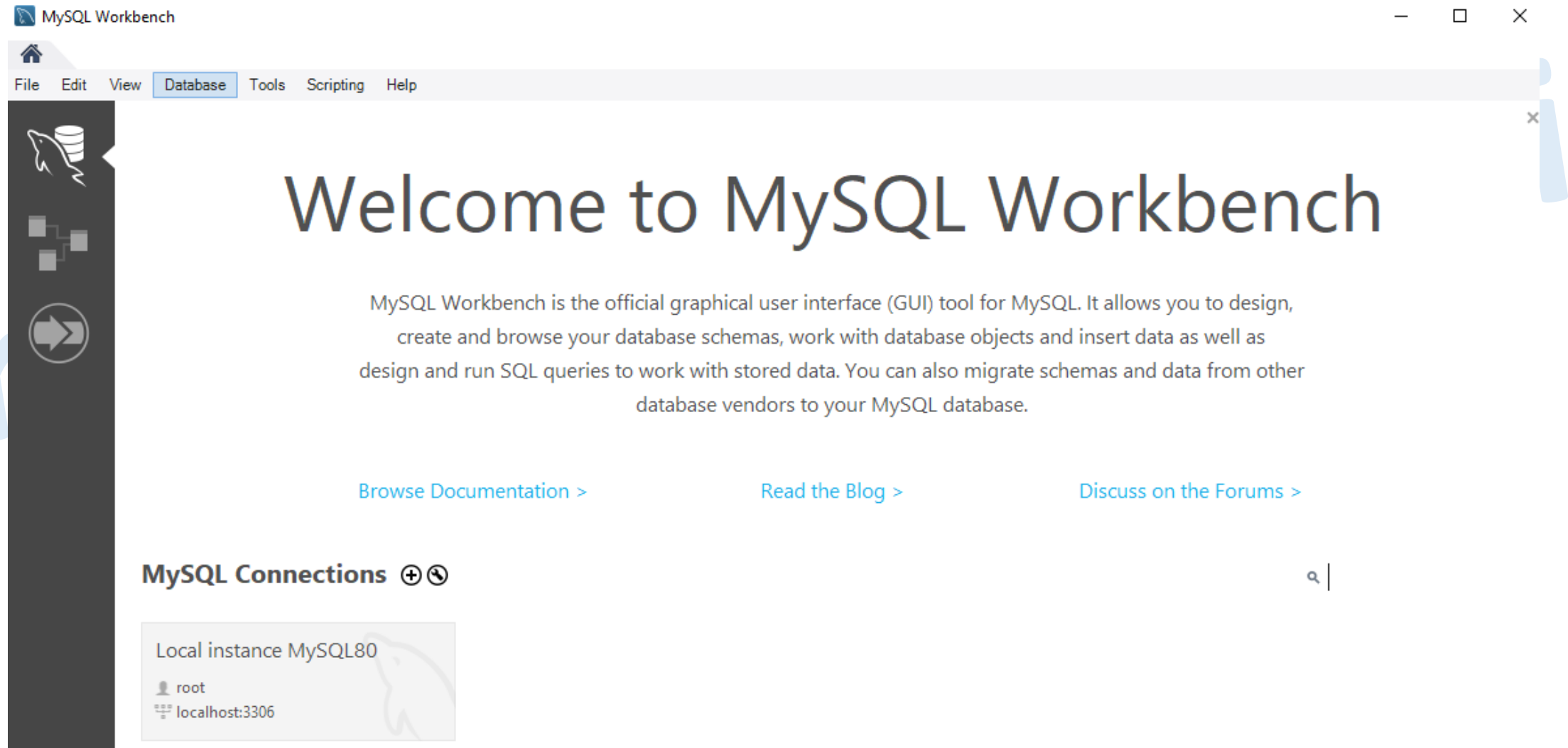
Oracle is a registered trademark of Oracle Corporation and/or its affiliates.

Other names may be trademarks of their respective owners.

Type '\help' or '\?' for help; '\quit' to exit.

MySQL JS >

Installation: View of Workbench



SQL: Data Definition Language (DDL)

- The DDL commands in SQL are used to create database schema and to define the type and structure of the data that will be stored in a database.
- SQL DDL commands are further divided into the following major categories:
 - CREATE: The CREATE query is used to create a database or objects such as tables, views, stored procedures, etc.
 - ALTER: alters the structure of the existing database
 - DROP: delete objects from the database
 - TRUNCATE: remove all records from a table, including all spaces allocated for the records are removed

SQL: DDL: CREATE

- Database

- `CREATE DATABASE LibraryDB;`

- Table

- `CREATE TABLE Books`

- `(`

- `Id INT (1),`

- `Name VARCHAR (50),`

- `Price INT (10)`

- `);`

Data type

- A Data Type in SQL server is defined as the type of data that any column or variable can store.
- It is a type of data that an object holds like integer, character, string, etc.
- An SQL developer must decide what type of data that will be stored inside each column when creating a table.
- While creating any table or variable, in addition to specifying the name, you also set the Type of Data it will store.
- The data type is a guideline for SQL to understand what type of data is expected inside of each column, and it also identifies how SQL will interact with the stored data.
- In MySQL there are three main data types: string, numeric, and date and time.

Data Type: String

Data type	Description
CHAR(size)	A FIXED length string (can contain letters, numbers, and special characters). The <i>size</i> parameter specifies the column length in characters - can be from 0 to 255. Default is 1
VARCHAR(size)	A VARIABLE length string (can contain letters, numbers, and special characters). The <i>size</i> parameter specifies the maximum column length in characters - can be from 0 to 65535
BINARY(size)	Equal to CHAR(), but stores binary byte strings. The <i>size</i> parameter specifies the column length in bytes. Default is 1
VARBINARY(size)	Equal to VARCHAR(), but stores binary byte strings. The <i>size</i> parameter specifies the maximum column length in bytes.
TINYBLOB	For BLOBs (Binary Large OBjects). Max length: 255 bytes
TINYTEXT	Holds a string with a maximum length of 255 characters

Data Type: String

TEXT(size)	Holds a string with a maximum length of 65,535 bytes
BLOB(size)	For BLOBs (Binary Large Objects). Holds up to 65,535 bytes of data
MEDIUMTEXT	Holds a string with a maximum length of 16,777,215 characters
MEDIUMBLOB	For BLOBs (Binary Large Objects). Holds up to 16,777,215 bytes of data
LONGTEXT	Holds a string with a maximum length of 4,294,967,295 characters
LOB	For BLOBs (Binary Large Objects). Holds up to 4,294,967,295 bytes of data
ENUM(val1, val2, val3, ...)	A string object that can have only one value, chosen from a list of possible values. You can list up to 65535 values in an ENUM list. If a value is inserted that is not in the list, a blank value will be inserted. The values are sorted in the order you enter them
SET(val1, val2, val3, ...)	A string object that can have 0 or more values, chosen from a list of possible values. You can list up to 64 values in a SET list

Data Type: Numeric

Data type	Description
BIT(<i>size</i>)	A bit-value type. The number of bits per value is specified in <i>size</i> . The <i>size</i> parameter can hold a value from 1 to 64. The default value for <i>size</i> is 1.
TINYINT(<i>size</i>)	A very small integer. Signed range is from -128 to 127. Unsigned range is from 0 to 255. The <i>size</i> parameter specifies the maximum display width (which is 255)
BOOL	Zero is considered as false, nonzero values are considered as true.
BOOLEAN	Equal to BOOL
SMALLINT(<i>size</i>)	A small integer. Signed range is from -32768 to 32767. Unsigned range is from 0 to 65535. The <i>size</i> parameter specifies the maximum display width (which is 255)
MEDIUMINT(<i>size</i>)	A medium integer. Signed range is from -8388608 to 8388607. Unsigned range is from 0 to 16777215. The <i>size</i> parameter specifies the maximum display width (which is 255)
INT(<i>size</i>)	A medium integer. Signed range is from -2147483648 to 2147483647. Unsigned range is from 0 to 4294967295. The <i>size</i> parameter specifies the maximum display width (which is 255)

Data Type: Numeric

INTEGER(<i>size</i>)	Equal to INT(<i>size</i>)
BIGINT(<i>size</i>)	A large integer. Signed range is from -9223372036854775808 to 9223372036854775807. Unsigned range is from 0 to 18446744073709551615. The <i>size</i> parameter specifies the maximum display width (which is 255)
FLOAT(<i>size</i> , <i>d</i>)	A floating point number. The total number of digits is specified in <i>size</i> . The number of digits after the decimal point is specified in the <i>d</i> parameter. This syntax is deprecated in MySQL 8.0.17, and it will be removed in future MySQL versions
FLOAT(<i>p</i>)	A floating point number. MySQL uses the <i>p</i> value to determine whether to use FLOAT or DOUBLE for the resulting data type. If <i>p</i> is from 0 to 24, the data type becomes FLOAT(). If <i>p</i> is from 25 to 53, the data type becomes DOUBLE()
DOUBLE(<i>size</i> , <i>d</i>)	A normal-size floating point number. The total number of digits is specified in <i>size</i> . The number of digits after the decimal point is specified in the <i>d</i> parameter
DOUBLE PRECISION(<i>size</i> , <i>d</i>)	
DECIMAL(<i>size</i> , <i>d</i>)	An exact fixed-point number. The total number of digits is specified in <i>size</i> . The number of digits after the decimal point is specified in the <i>d</i> parameter. The maximum number for <i>size</i> is 65. The maximum number for <i>d</i> is 30. The default value for <i>size</i> is 10. The default value for <i>d</i> is 0.
DEC(<i>size</i> , <i>d</i>)	Equal to DECIMAL(<i>size</i> , <i>d</i>)

Data Type: Date and Time

Data type	Description
DATE	A date. Format: YYYY-MM-DD. The supported range is from '1000-01-01' to '9999-12-31'
DATETIME(<i>fsp</i>)	A date and time combination. Format: YYYY-MM-DD hh:mm:ss. The supported range is from '1000-01-01 00:00:00' to '9999-12-31 23:59:59'. Adding DEFAULT and ON UPDATE in the column definition to get automatic initialization and updating to the current date and time
TIMESTAMP(<i>fsp</i>)	A timestamp. TIMESTAMP values are stored as the number of seconds since the Unix epoch ('1970-01-01 00:00:00' UTC). Format: YYYY-MM-DD hh:mm:ss. The supported range is from '1970-01-01 00:00:01' UTC to '2038-01-09 03:14:07' UTC. Automatic initialization and updating to the current date and time can be specified using DEFAULT CURRENT_TIMESTAMP and ON UPDATE CURRENT_TIMESTAMP in the column definition
TIME(<i>fsp</i>)	A time. Format: hh:mm:ss. The supported range is from '-838:59:59' to '838:59:59'
YEAR	A year in four-digit format. Values allowed in four-digit format: 1901 to 2155, and 0000. MySQL 8.0 does not support year in two-digit format.

Primary Key Constraint

- The **PRIMARY KEY** constraint uniquely identifies each record in a table.
- Primary keys must contain UNIQUE values, and cannot contain NULL values.
- A table can have only ONE primary key; and in the table, this primary key can consist of single or multiple columns (fields).

SQL: DDL: ALTER

- The ALTER command in SQL DDL is used to modify the structure of an already existing table.
- Add primary key:
 - ALTER TABLE Books
ADD PRIMARY KEY (Id);
- Add new column:
 - ALTER TABLE Books
ADD Publisher varchar(50),
ADD Year year;
ALTER TABLE Books
ADD AuthorName varchar(50);
- Modify the data type of a column:
 - ALTER TABLE Books
MODIFY COLUMN Price float(10,2);

SQL: DDL: ALTER

- Modify the column name:
 - `ALTER TABLE Books`
`RENAME COLUMN AuthorName TO FirstName,`
`ADD LastName varchar(50);`
- Modify table name:
 - `ALTER TABLE Books RENAME Book_Details;`
- Drop a column:
 - `ALTER TABLE Book_Details`
`DROP COLUMN Publisher;`
- Add NOT NULL constraint:
 - `ALTER TABLE Book_Details`
`MODIFY Name varchar(50) NOT NULL;`

SQL: DDL: DROP and TRUNCATE

- Drop a column:
 - `ALTER TABLE Book_Details`
`DROP COLUMN Publisher;`
- The DROP TABLE statement is used to drop an existing table in a database.
 - `DROP TABLE Book_Details;`
- Drop the database:
 - `DROP DATABASE libraryDB;`
- The TRUNCATE TABLE statement is used to delete the data inside a table, but not the table itself.
 - `TRUNCATE TABLE Book_Details;`

SQL: DML

- DML is short name of **Data Manipulation Language** which deals with data manipulation and includes most common SQL statements such as
 - SELECT: Used to query or fetch selected fields or columns from a database table.
 - INSERT: Used to insert new data records or rows in the database table
 - UPDATE: Used to set the value of a field or column for a particular record to a new value
 - DELETE: Used to remove one or more rows from the database table

SQL: DML: INSERT

- Specify both the column names and the values to be inserted:
 - **INSERT INTO** *table_name* (*column1, column2, column3, ...*)
VALUES (*value1, value2, value3, ...*);
- If you are adding values for all the columns of the table, you do not need to specify the column names in the SQL query.
 - **INSERT INTO** *table_name*
VALUES (*value1, value2, value3, ...*);
- Insert data in specific columns.
 - **INSERT INTO** *table_name* (*column2, column5, column6*)
VALUES (*value1, value2, value3*);

SQL: DML: SELECT and SELECT DISTINCT

- The field names of the table you want to select data from:
 - `SELECT column1, column2, ...
FROM table_name;`
- If you want to select all the fields available in the table, use the following syntax:
 - `SELECT * FROM table_name;`
- We can also populate one table using another table with the help of select statement. The only condition is that the table must have the same sets of attributes.
 - `Insert into table_no_first [(column1, column 2...column n)]
select column1, column 2...column n from table_no_two;`
- Used to return only distinct (different) values.
 - `SELECT DISTINCT column1, column2, ...
FROM table_name;`

SQL: DML: SELECT with where

- used to extract only those records that fulfill a specified condition.
 - `SELECT column1, column2, ...`
`FROM table_name`
`WHERE condition;`
- SQL requires single quotes around text values (most database systems will also allow double quotes).
- However, numeric fields should not be enclosed in quotes.
- Operators → =, >, <, <>, >=, <=, BETWEEN, LIKE, IN

SQL: DML: SELECT with AND, OR, NOT

- The **AND** operator displays a record if all the conditions separated by **AND** are TRUE.
 - `SELECT column1, column2, ...
FROM table_name
WHERE condition1 AND condition2 AND condition3 ...;`
- The **OR** operator displays a record if any of the conditions separated by **OR** is TRUE.
 - `SELECT column1, column2, ...
FROM table_name
WHERE condition1 OR condition2 OR condition3 ...;`
- The **NOT** operator displays a record if the condition(s) is NOT TRUE.
 - `SELECT column1, column2, ...
FROM table_name
WHERE NOT condition;`
- Combine:
 - `SELECT * FROM students WHERE student_name LIKE 'r%' AND (course='C' OR roll_no between 2 and 4);`

SQL: DML: SELECT with ORDER BY

- Used to sort the result-set in ascending or descending order.
- Sorts the records in ascending order by default. To sort the records in descending order, use the **DESC** keyword.
 - **SELECT** *column1, column2, ...*
FROM *table_name*
ORDER BY *column1, column2, ...* **ASC|DESC**;
- ORDER BY Several Columns: Means that it orders by column1, but if some rows have the same column1 value, it orders them by column2:
 - **SELECT** * **FROM** *table_name*
ORDER BY *column1, column2*;
 - **SELECT** * **FROM** *table_name*
ORDER BY *column1* **ASC**, *column2* **DESC**;

SQL: DML: SELECT with GROUP BY

- groups rows that have the same values into summary rows

- `SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
ORDER BY column_name(s);`

- `SELECT COUNT(roll_no), course
FROM students
GROUP BY course; #lists the number of students in each course`

- `SELECT COUNT(roll_no), course
FROM students
GROUP BY course
ORDER BY COUNT(roll_no) DESC; #number of students in each course, sorted high
to low`

SQL: DML: SELECT with NULL and NOT NULL

- `SELECT column_names`
`FROM table_name`
`WHERE column_name IS NULL;`
- `SELECT column_names`
`FROM table_name`
`WHERE column_name IS NOT NULL;`

SQL: DML: SELECT with HAVING

- The **HAVING** clause was added to SQL because the **WHERE** keyword cannot be used with aggregate functions.

- **SELECT** *column_name(s)*
FROM *table_name*
WHERE *condition*
GROUP BY *column_name(s)*
HAVING *condition*
ORDER BY *column_name(s);*

- **SELECT COUNT**(roll_no), course
FROM students
GROUP BY course
HAVING COUNT(roll_no) > 1; #lists the number of students in each course with more than 1 student

SQL: DML: SELECT with Aggregate Functions

- MIN() Syntax: returns the smallest value of the selected column
 - `SELECT MIN(column_name) FROM table_name WHERE condition;`
- MAX() Syntax: returns the largest value of the selected column
 - `SELECT MAX(column_name) FROM table_name WHERE condition;`
- COUNT() Syntax: returns the number of rows that matches a specified criterion
 - `SELECT COUNT(column_name) FROM table_name WHERE condition;`
- AVG() Syntax: returns the average value of a numeric column
 - `SELECT AVG(column_name) FROM table_name WHERE condition;`
- SUM() Syntax: returns the total sum of a numeric column
 - `SELECT SUM(column_name) FROM table_name WHERE condition;`

SQL: DML: UPDATE

- **UPDATE** *table_name*
SET *column1 = value1, column2 = value2, ...*
WHERE *condition*;
- UPDATE Multiple Records: update the column1 to value1 for all records where the condition is true
 - **UPDATE** *table_name*
SET *column1 = value1*
WHERE *condition*;
- Warning
 - Be careful when updating records. If you omit the WHERE clause, ALL records will be updated!
 - **UPDATE** *table_name*
SET *column1 = value1*;

SQL: DML: DELETE

- `DELETE FROM table_name WHERE condition;`
- `DELETE FROM table_name;` [to delete all records]

Dr. Jyotismita Chaki