

**NET320-Database Systems**

**SWS213-Database Design**

**Lab 2a**

**Workbench Administration**

## Objective

The objective of this lab is to have hands on experience with MySQL Workbench. After the completion of lab, students should be able to create a model of database.

## Background

MySQL Workbench provides a graphical tool for working with MySQL Servers and databases. MySQL Workbench fully supports MySQL Server versions 5.1 and above.

MySQL Workbench provides three main areas of functionality:

• **SQL Development:** Enables you to create and manage connections to database servers. As well as enabling you to configure connection parameters, MySQL Workbench provides the capability to execute SQL queries on the database connections using the built-in SQL Editor. This functionality replaces that previously provided by the Query Browser standalone application.

• **Data Modeling:** Enables you to create models of your database schema graphically, reverse and forward engineer between a schema and a live database, and edit all aspects of your database using the comprehensive Table Editor. The Table Editor provides easy-to-use facilities for editing Tables, Columns, Indexes, Triggers, Partitioning, Options, Inserts and Privileges, Routines and Views.

• **Server Administration:** Enables you to create and administer server instances.

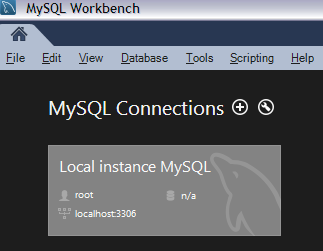
MySQL Workbench is available in two editions, the Community Edition and the Standard Edition. The Community Edition is available free of charge. The Standard Edition provides additional Enterprise features, such as database documentation generation, at low cost.

## Software Required

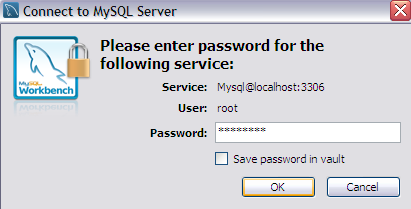
MySQL WorkBench

**Administering the Workbench**

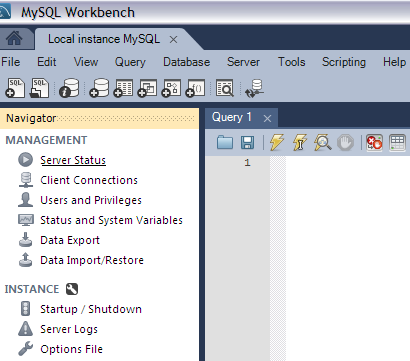
1. Open up the MySQL Workbench from Start, Programs, MySQL.
2. Double click on ‘’ Local instance MySQL”.



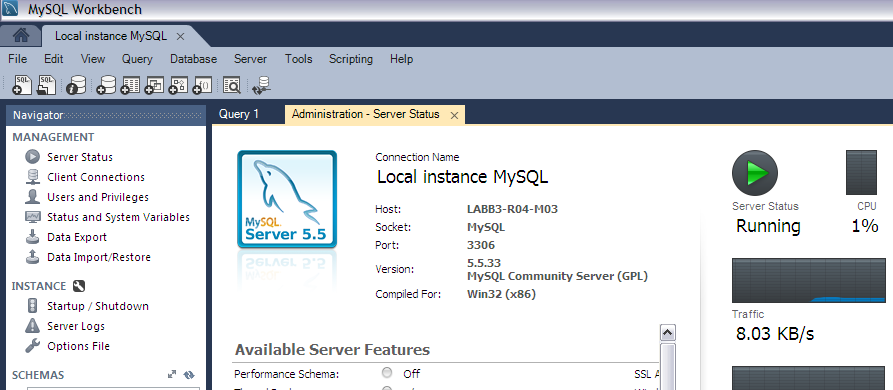
1. Enter the password as ‘password’ and click on ‘OK’.



1. Under the ‘Management’ click on the ‘Server Status’.

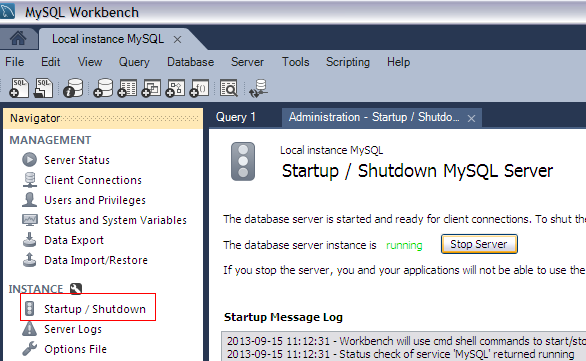


1. If prompted for a password, enter password as ‘password’.
2. Check the server status. Server status should be running.

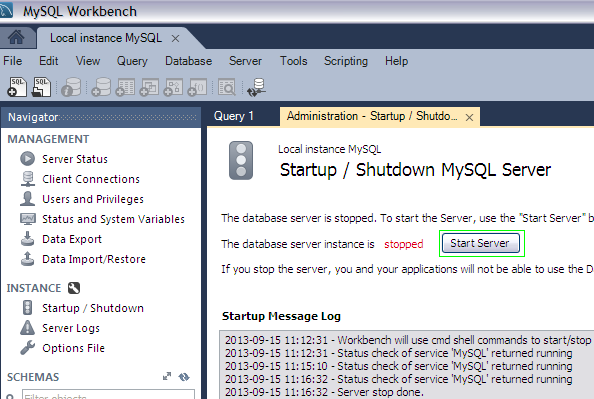


1. **Startup / Shutdown MySQL Server.**

Under Instance click on 'Startup/ Shutdown', check the database server instance status. If you wish to stop the server instance, you can do that by clicking 'Stop Server'.



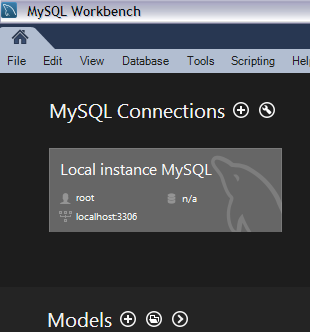
1. If you wish to start the server again, you can do that by clicking ‘Start Server’.



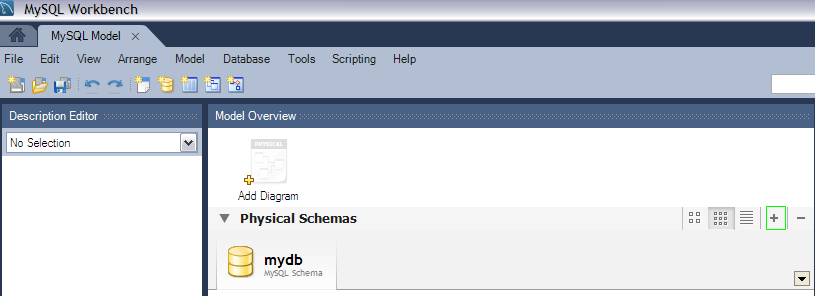
**Creating a Model**

In this section, you will learn how to create a new database model, create a table, create an EER Diagram of your model, and then forward engineer your model to the live database server.

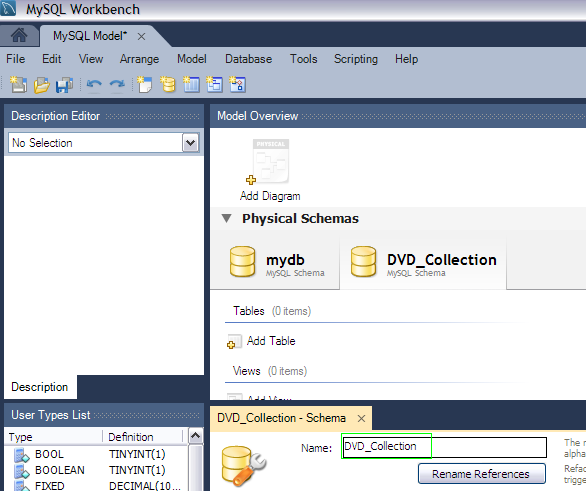
1. From the MySQL home page click on ‘’ for creating a new model.



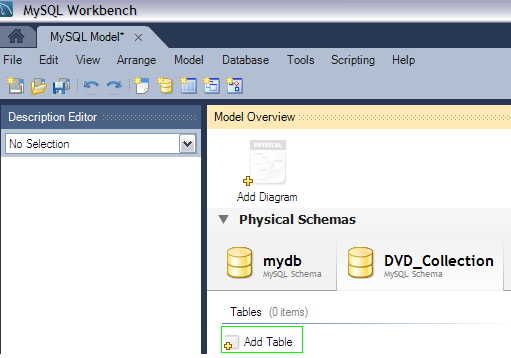
1. On the Physical Schemata toolbar, click the button + to add a new schema. This will create a new schema and display a tabsheet for the schema.



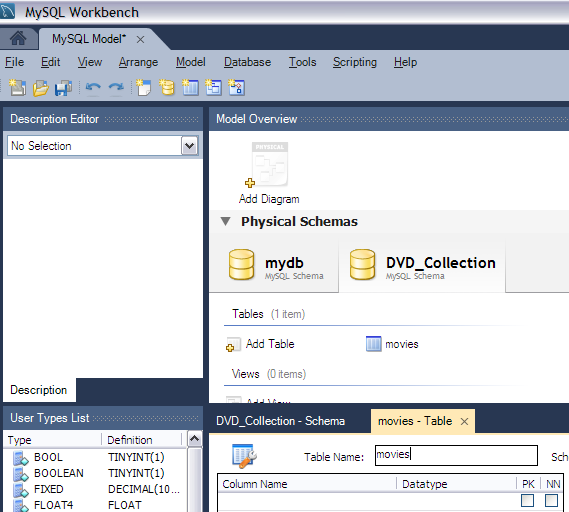
1. In the tabsheet, change the name of the schema to “**dvd\_collection**”, by typing into the field called **Name**. Ensure that this change is reflected on the Physical Schemata tab. Now you are ready to add a table to your schema.



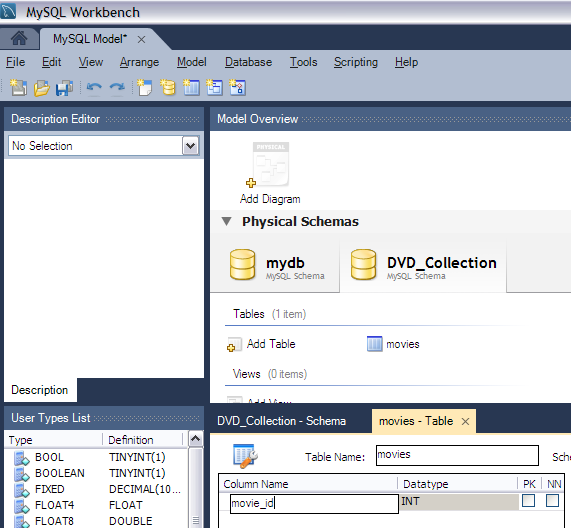
1. In the Physical Schemata section, double-click **Add Table.**



1. This will automatically load the table editor, with the default table name being **table1**. In the table editor, change the name of the table from “table1” to “movies”.



1. Next, add several columns. Double click a cell within the Column Name column, and the first field will default to “idmovies” because MySQL Workbench appends “id” to the table name as the default for the initial field. Change the name to “movie\_id” and keep the Datatype as INT. Then, be sure PK (PRIMARY KEY), NN (NOT NULL), and AI (AUTO\_INCREMENT) are all checked.



The PRIMARY KEY constraint uniquely identifies each record in a database table.

Primary keys must contain unique values.

A primary key column cannot contain NULL values.

Each table should have a primary key, and each table can have only ONE primary key.

The NOT NULL constraint enforces a column to NOT accept NULL values.

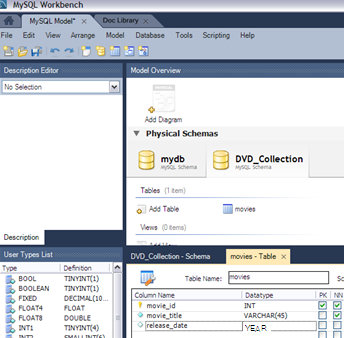
The NOT NULL constraint enforces a field to always contain a value. This means that you cannot insert a new record, or update a record without adding a value to this field.

MySQL uses the AUTO\_INCREMENT keyword to perform an auto-increment feature.

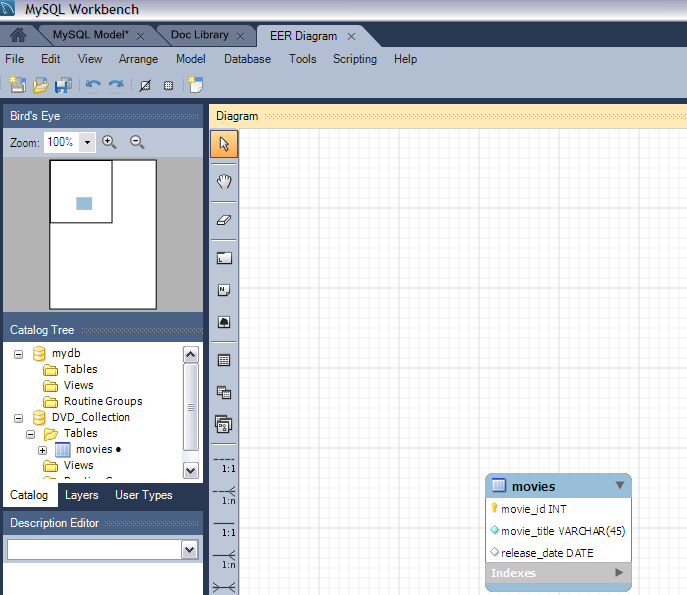
By default, the starting value for AUTO\_INCREMENT is 1, and it will increment by 1 for each new record.

1. Add two additional columns using the same method as described above:

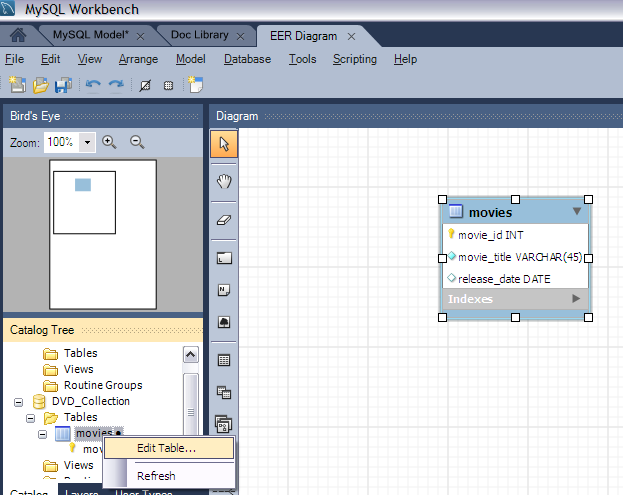
|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Column Properties** |
| movie\_title | VARCHAR(45) | NN |
| release\_date | YEAR | None |



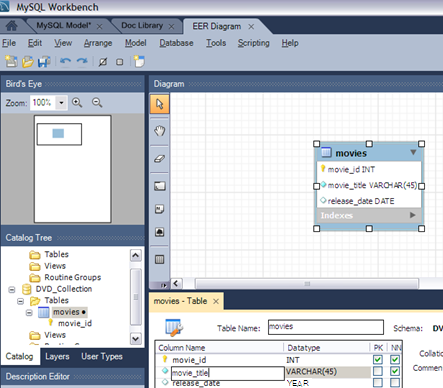
1. Now you can obtain a visual representation of this schema so far. From the main menu, select Model, Create Diagram from Catalog Objects. The EER Diagram will be created and displayed.



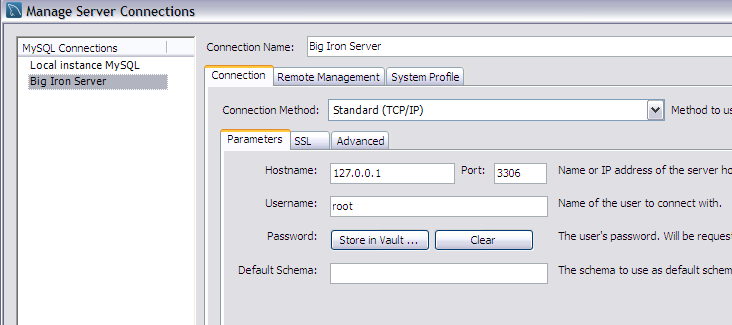
1. Under the ‘Catalog tree’, right click the ‘movies’ and select the option ‘Edit Table’.



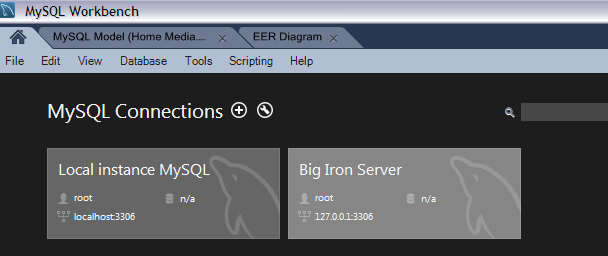
1. Change the ‘movie\_title’ to ‘title’.



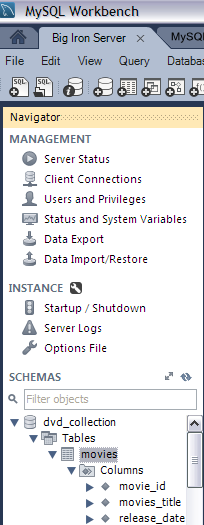
1. At this point, you can save your model. Click the main toolbar button Save Model to Current File. You have not yet saved this file so you will be prompted to enter a model file name. For this tutorial, enter “Home\_Media”. The Home\_Media model may contain further schemata in addition to dvd\_collection, such as cd\_collection. Click Save to save the model.
2. You can synchronize your model with the live database server. First, you must tell MySQL Workbench how to connect to the live server. From the main menu, select Database, Manage Connections.
3. In the Manage DB Connections dialog, click New.
4. Enter “Big Iron Server” for the connection name. This enables you to identify the server to which this connection corresponds, although it is possible to create multiple connections to the same server.
5. Enter the user name for the account you will use to connect to the server.
6. Click on the Store in Vault... button and enter the password for the user name you entered in the previous step. You can optionally ignore this step, and you will be prompted for this password whenever MySQL Workbench connects to the server.



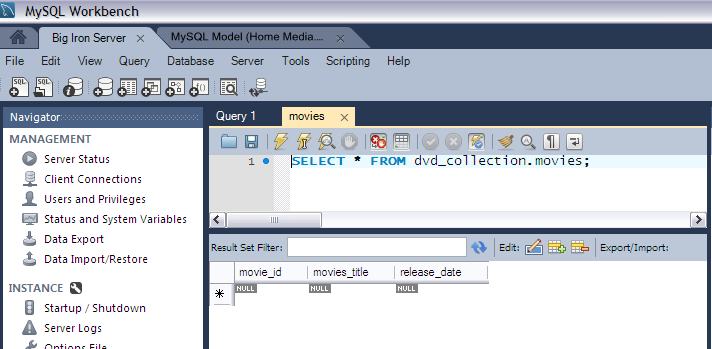
1. Click Test Connection to test your connection parameters. If everything is okay at this point, you can click Close.
2. You are now ready to forward engineer your model to the live server. From the main menu, select Database, Forward Engineer.... The Forward Engineer to Database wizard will be displayed.
3. The Options page of the wizard shows various advanced options. For this tutorial, you can ignore these and simply click Next.
4. On the next page, you can select the object you want to export to the live server. In this case, you only have a table, so no other objects need be selected. Click Next.
5. The next page, Review SQL Script, displays the script that will be run on the live server to create your schema. Review the script to make sure that you understand the operations that will be carried out. Click Next.
6. Double click on ‘Big Iron Server’.



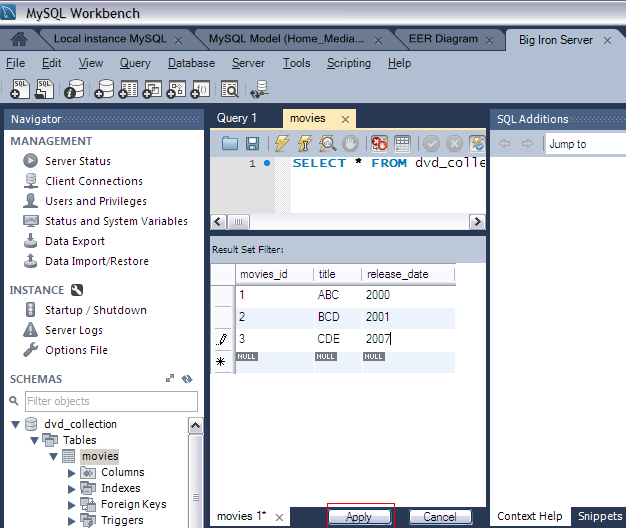
1. Select ‘movies’ under ‘Schemas, dvd\_collection, tables’.



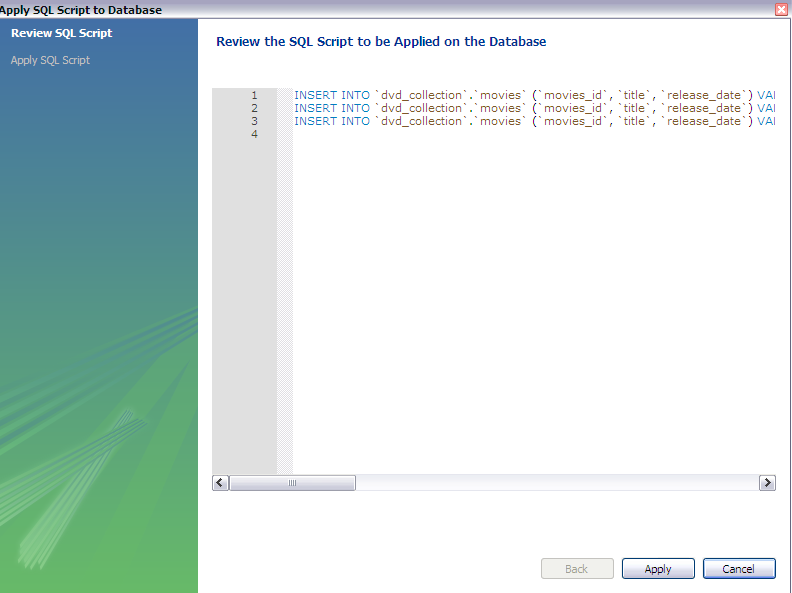
1. Right click on ’movies’ and select ‘Select Rows- Limit 1000’.



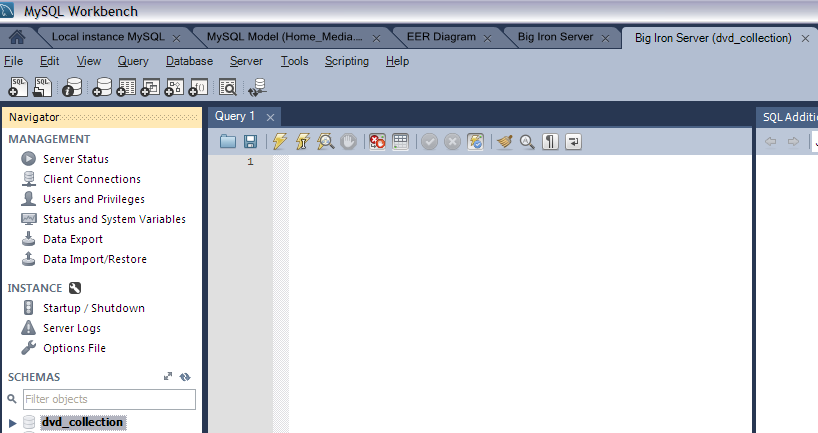
1. Enter ‘movie id, title and release date’. Click on ‘Apply’.



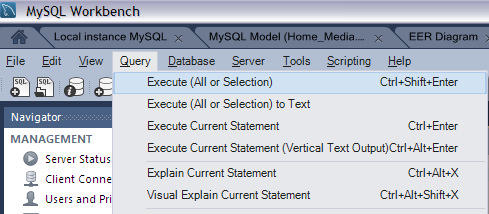
1. A list of SQL statements will be displayed. Confirm that you understand the operations to be carried out. Click Apply to apply these changes to the live server.



1. Confirm that the script was executed correctly, then click Finish.
2. Double click on ‘Big Iron Server’.



1. Now you will check that the data really has been applied to the live server.. Enter ‘SELECT \* FROM movies;’ to see the data just entered.
2. Click on ‘Execute’ under Query.



1. The result should be like this.

