CIND110 - Data Organization for Data Analysts
Lab 02 - Design and Manage Relational Models

Contents

1 Running MySQL Workbench 2

2 Creating a New Relational Model 4

3 Generating an Entity-Relationship Diagram 6

4 Forward Engineer an Existing Database 9

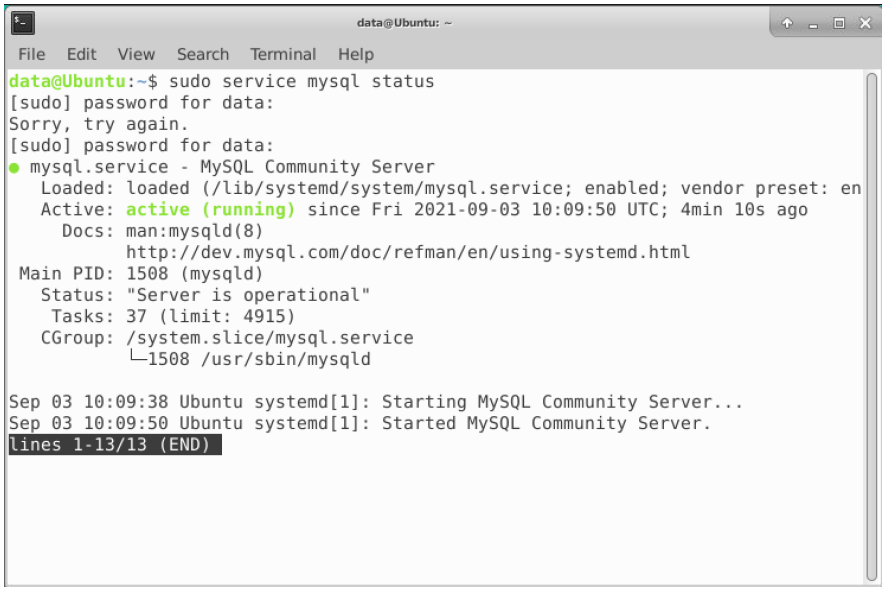
5 Reverse Engineer an Existing Database 10

6 Example 12

1 Running MySQL Workbench

- After installing the MySQL Workbench tool, you might need to ensure that the MySQL service is active. You can check the status of the MySQL service using the following Linux command from the Terminal application.

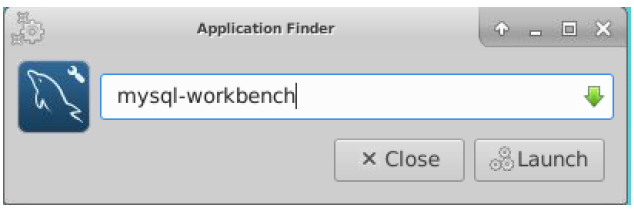
```
sudo service mysql status
```



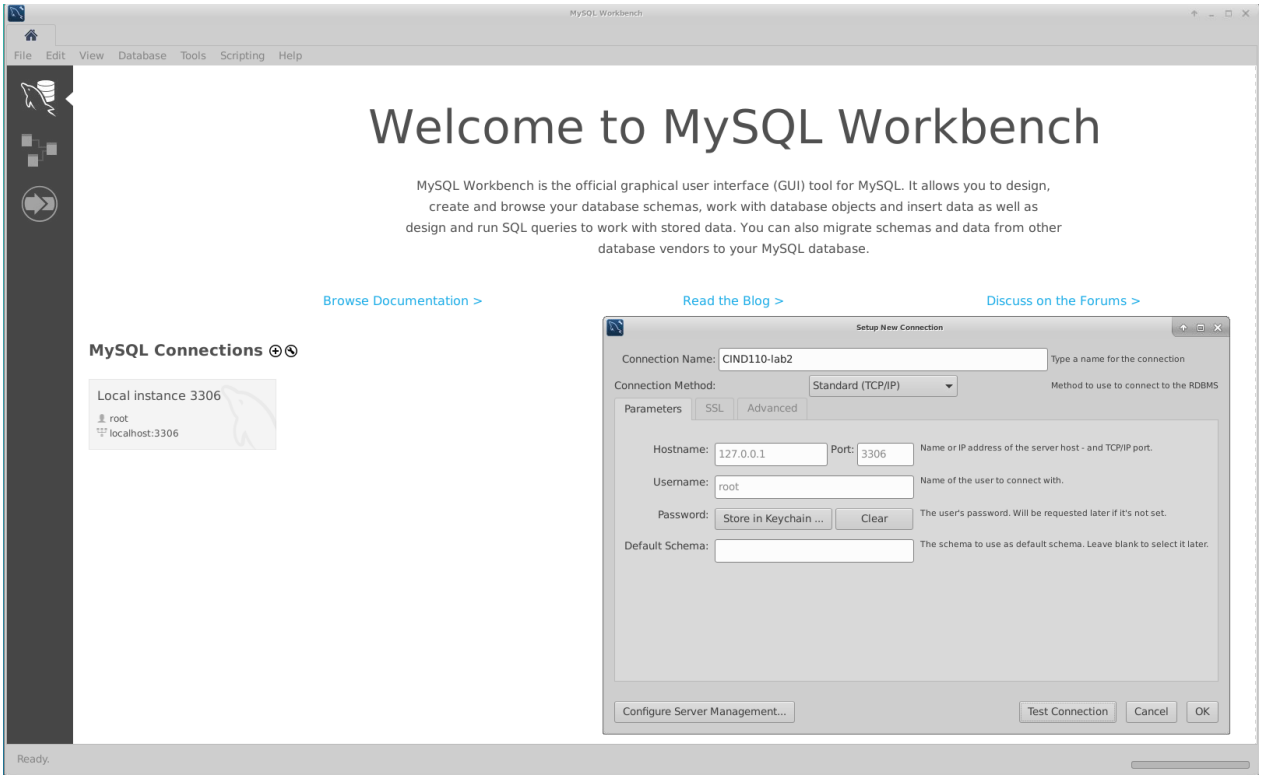
- In case finding the MySQL service inactive, consider (re)starting the service using the following Linux command from the Terminal application.

```
sudo service mysql restart
```

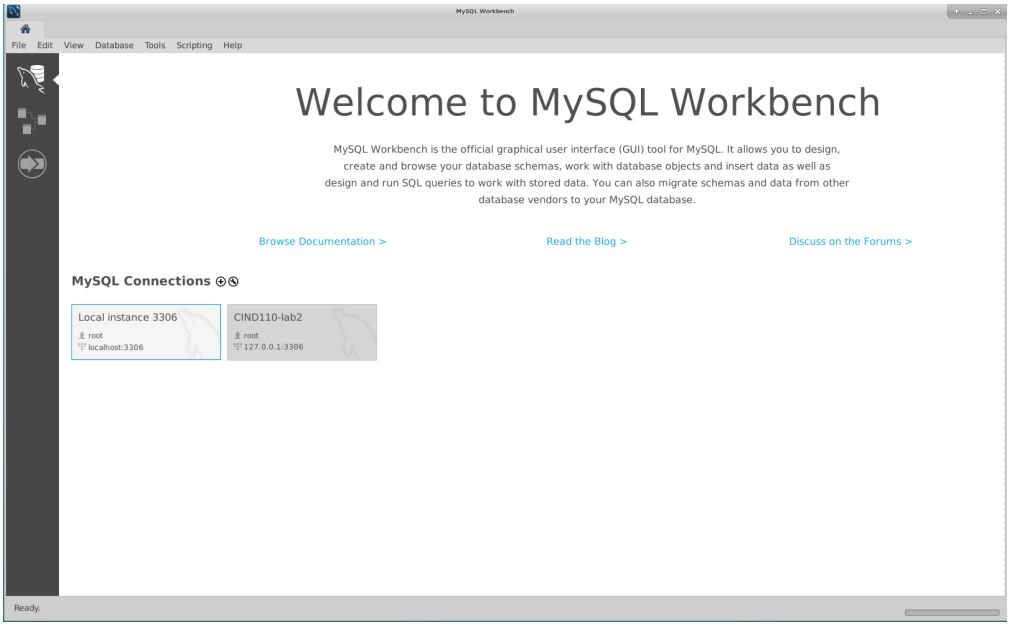
- Now, launch the MySQL Workbench tool either from the Application Finder or by double-clicking the icon labelled `MySQL Workbench` on the desktop, as shown below.

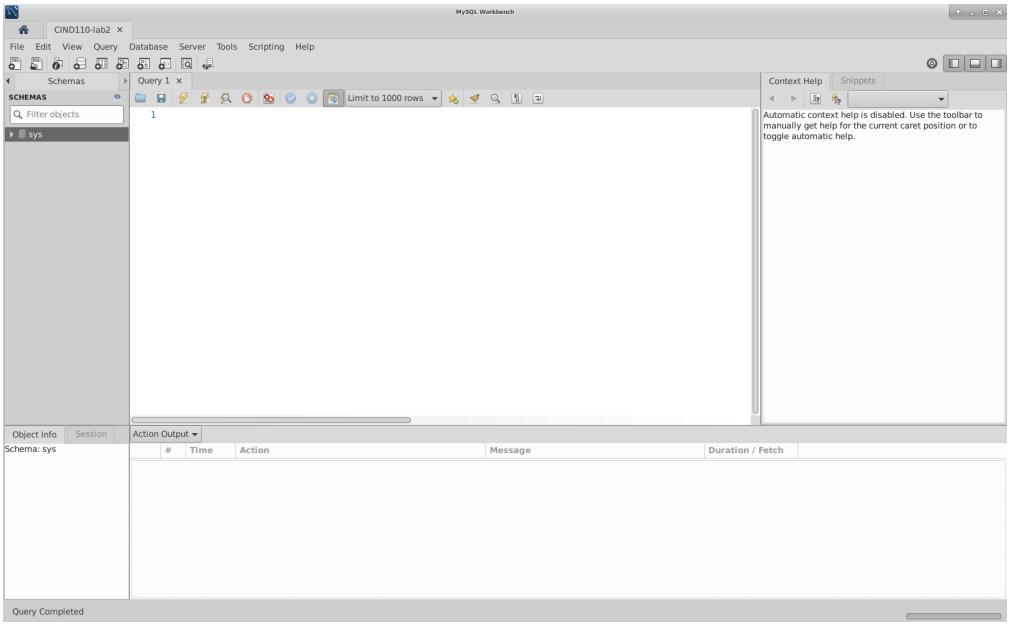


- You will see a window with your connections and models. This is your MySQL Integrated Development Environment.
- Establish a new MySQL connection by clicking on the plus icon, then enter the connection's name. In this example, it is `CIND110-lab2`.



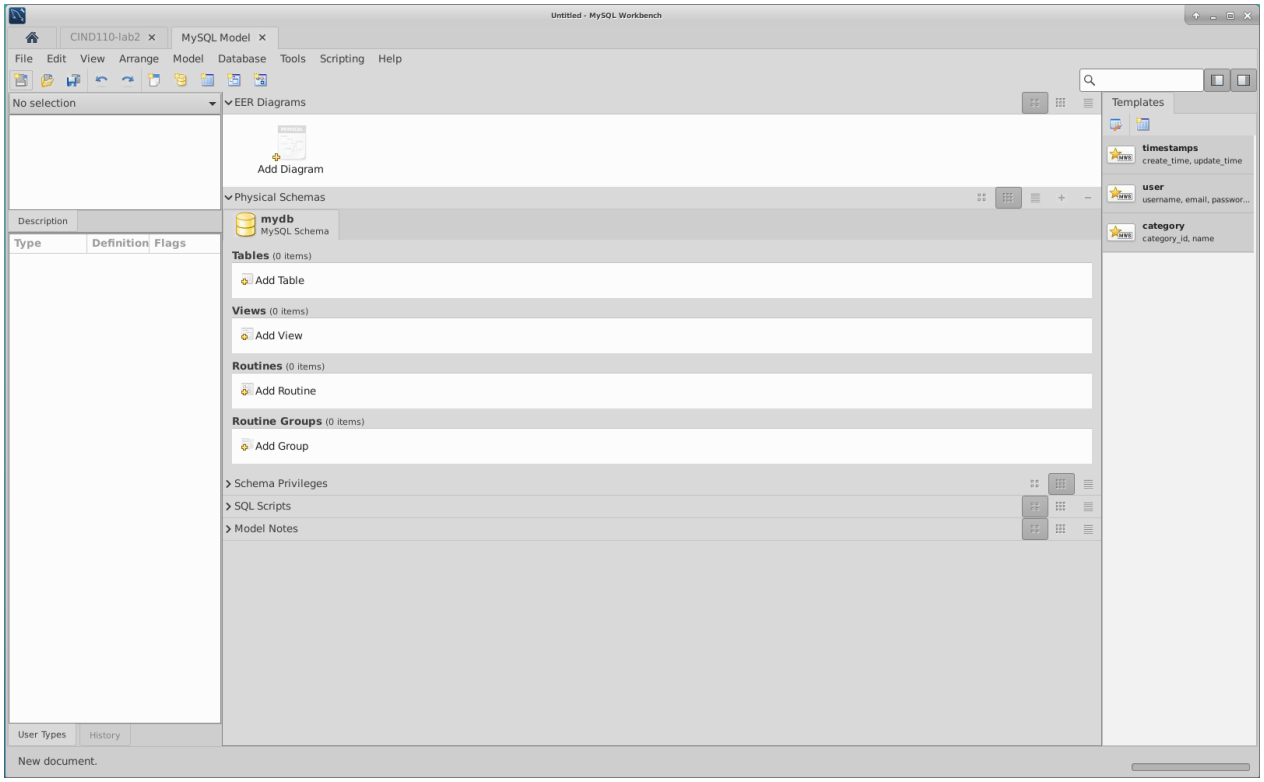
- You can use the Hostname as `localhost` or the IP address `127.0.0.1`, to connect to your local machine. If you would like to link to another machine/server, you need to enter the IP address of that machine/server.
- When it prompts for a password, please enter the given user's password. In this example, the user is `root`, and the password is `root`.
- You might need to check the establishment of the connection using the `Test Connection` button.
- After clicking the `OK` button, you will see that the new connection has been listed as a new connection and is ready to be activated.
- Clicking on the new connection, `CIND110-lab2` box will open a new window listing all the databases created on that connection and the SQL editor where you can start writing your queries.

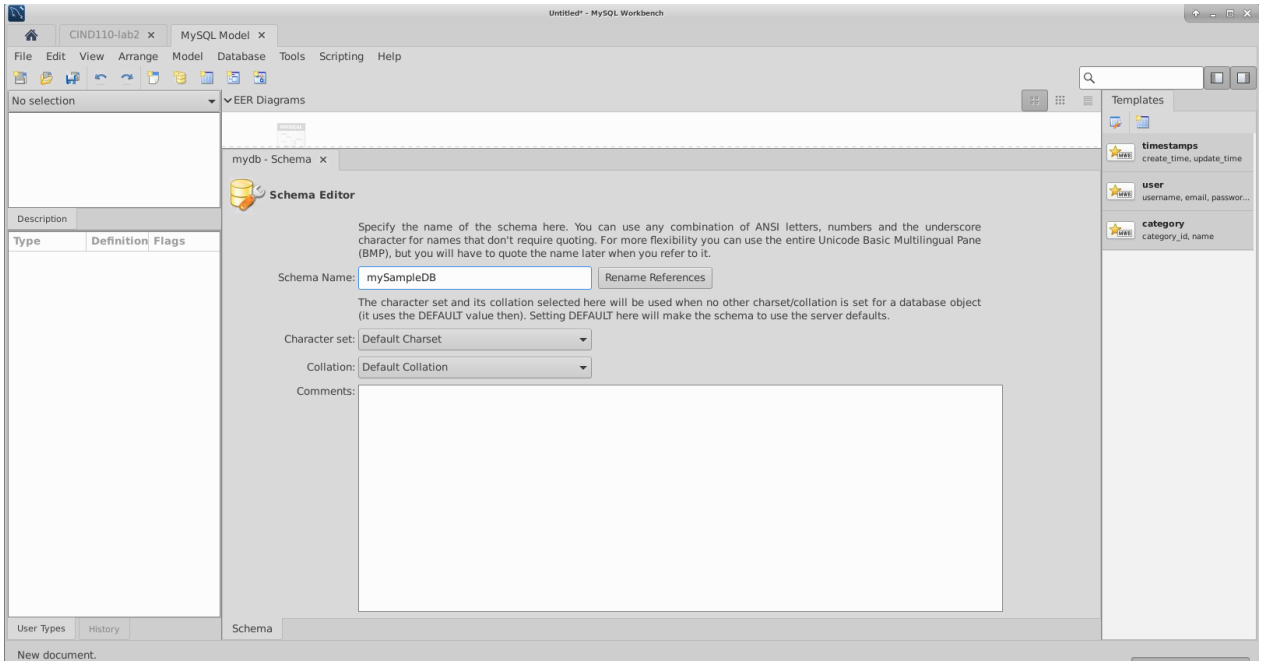




2 Creating a New Relational Model

- To create a new relational model, start the MySQL Workbench tool, establish a connection, then click on the `File` tab from the top navigation bar, and then select `New Model`. This will open a new panel and add it to the workbench environment as `MySQL Model`.
- Open the Schema Editor by double-clicking the tab labelled `mydb MySQL Schema`.
- In the resulting properties panel, change the name of the scheme to `mySampleDB`.

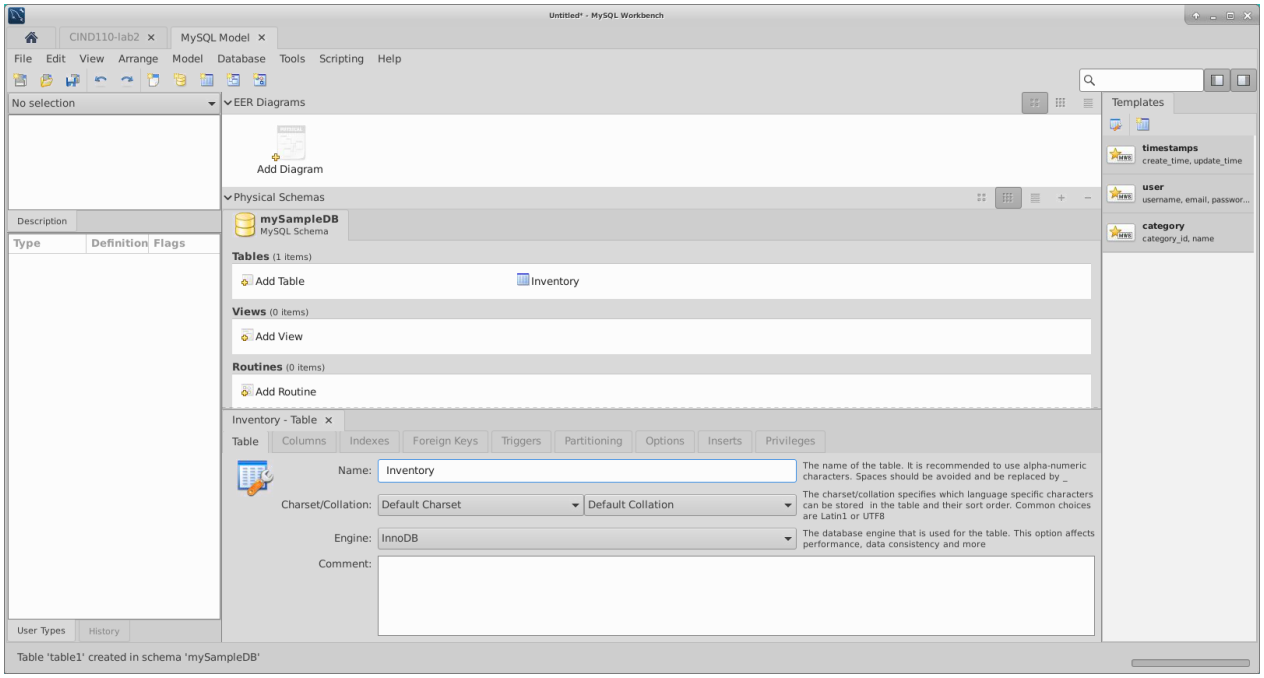




- Close the Schema Editor by clicking on the `x` button next to the schema tab.
- Having given our schema a name: `mySampleDB`, we can now start adding tables to our relational model. This can be achieved by double-clicking the `Add Table` button in the `Tables`

tab.

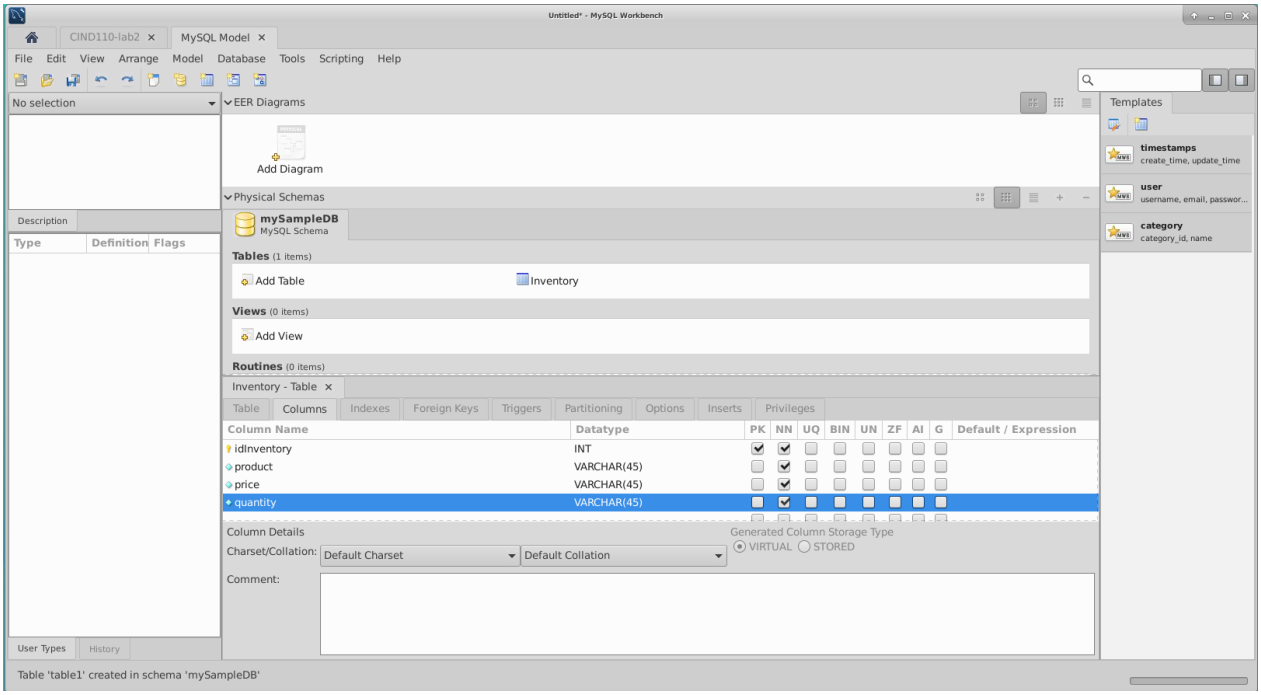
- Change the table name to `Inventory`, then click the `Columns` tab to begin adding columns/attributes to the `Inventory` table/entity.

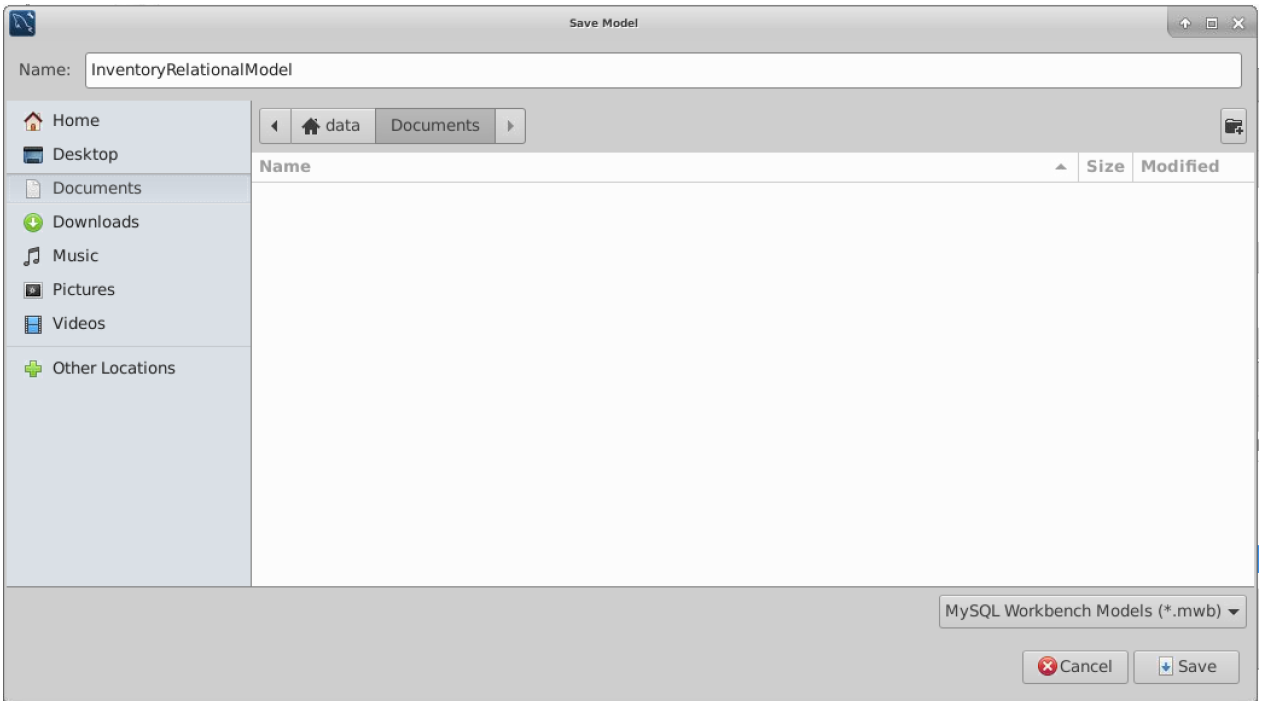


- By default, the workbench IDE will create a non-null primary key named `idinventory`. For this exercise, we will keep this column name as is; however, this can be changed in practice.
- Select the second line, double click the `ColumnName` field and create another column/attribute called `product`.
- Accept the default value of `VARCHAR(45)` as the data type for the `product` attribute.
- Repeat the same steps to add a `price` column of type `FLOAT` and a `quantity` attribute of type `INTEGER`.
- Each of these attributes should be declared as non-null by checking the boxes in the `NN` column.
- Save the relational model as `InventoryRelationalModel`, by selecting the `File -> Save Model` menu item. Notably the extension of the file will be `.mwb` as it has been created by the MySQL workbench tool.

3 Generating an Entity-Relationship Diagram

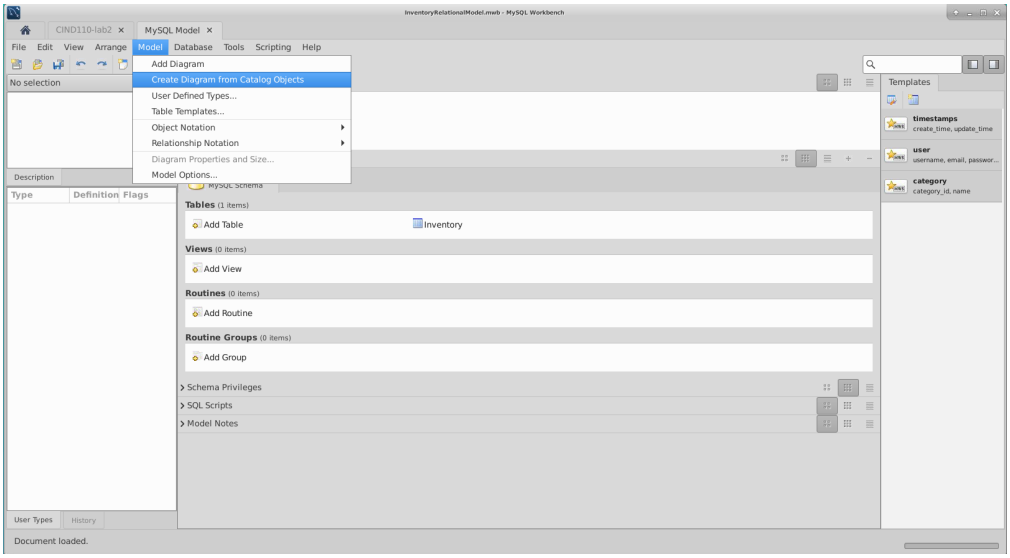
- To generate an EER diagram of the `mySampleDB` database, select the `Model -> Create Diagram from Catalog Objects` menu option.

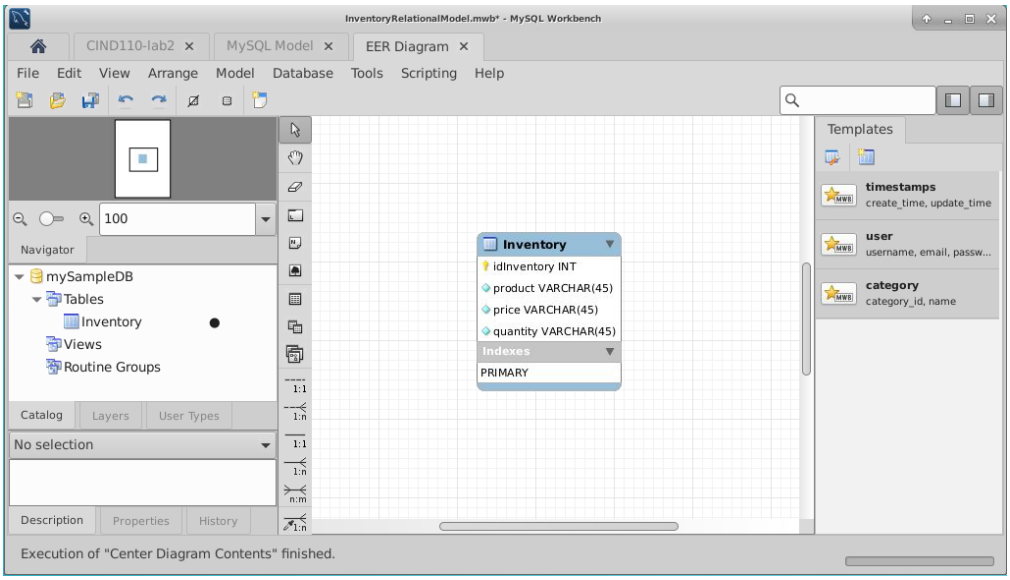




- The EER diagram illustrated in the following figure will then be generated automatically based on the `mySampleDB` database and its including tables, which is in our case, only one table, being the `Inventory` table.

- Although we are not creating any relationships in this example, it is essential to note that the

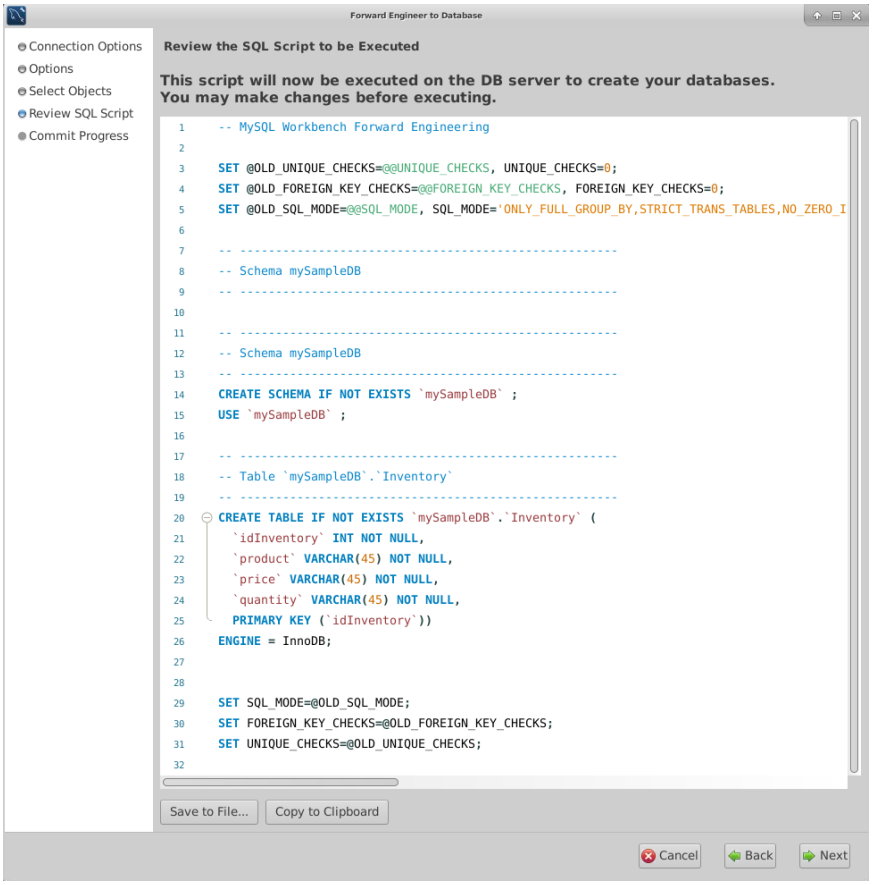




tools on the left side of the EER diagram, which are in the form of a vertical array of buttons, allow such relationships to be created between entities.

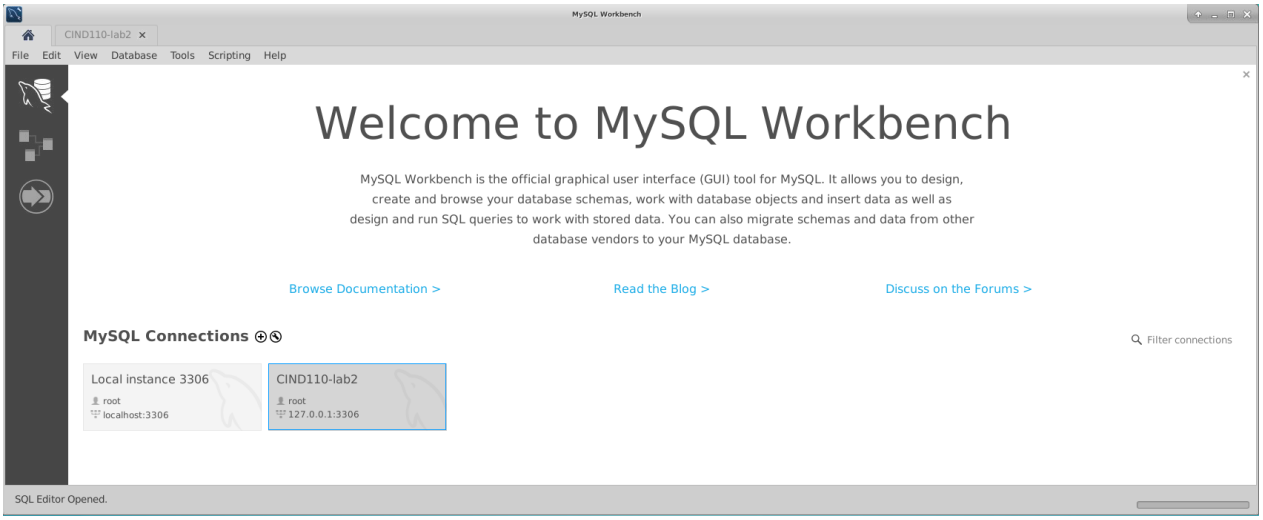
4 Forward Engineer an Existing Database

- To migrate a database from one machine to another, we need to convert it to an SQL script that can be executed on the new server.
- To achieve this, select the `Database -> Forward Engineer` menu option. It will pop up a new dialogue window, select the database connection and click the `Next` button. You will see a set of options available to filter and modify the script that the system will ultimately generate.
- The output of the forward engineering process is an executable SQL script that creates the structure of the `mySampleDB` database, along with the included tables.
- Click on the `Save to file ...` button to store the file with an `.sql` extension.
- Click the `Next` button to ensure that the forward engineering step has been completed successfully.

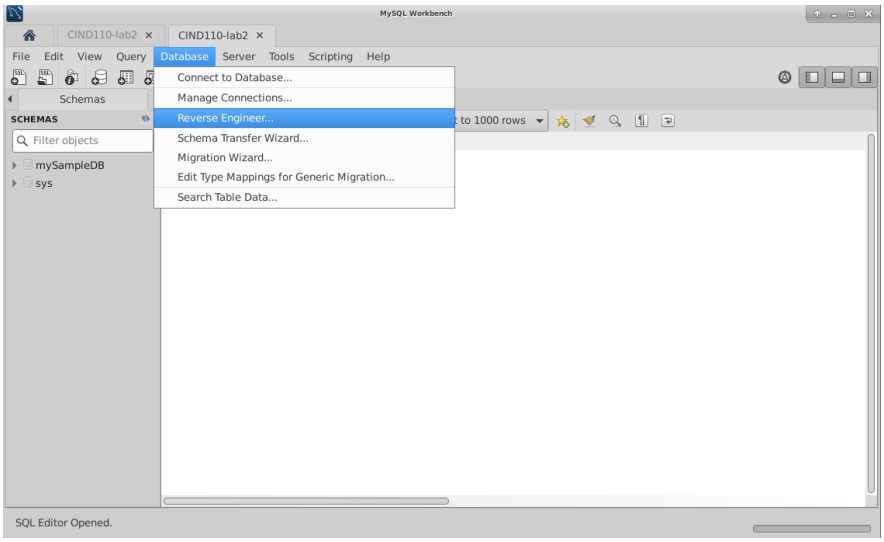


5 Reverse Engineer an Existing Database

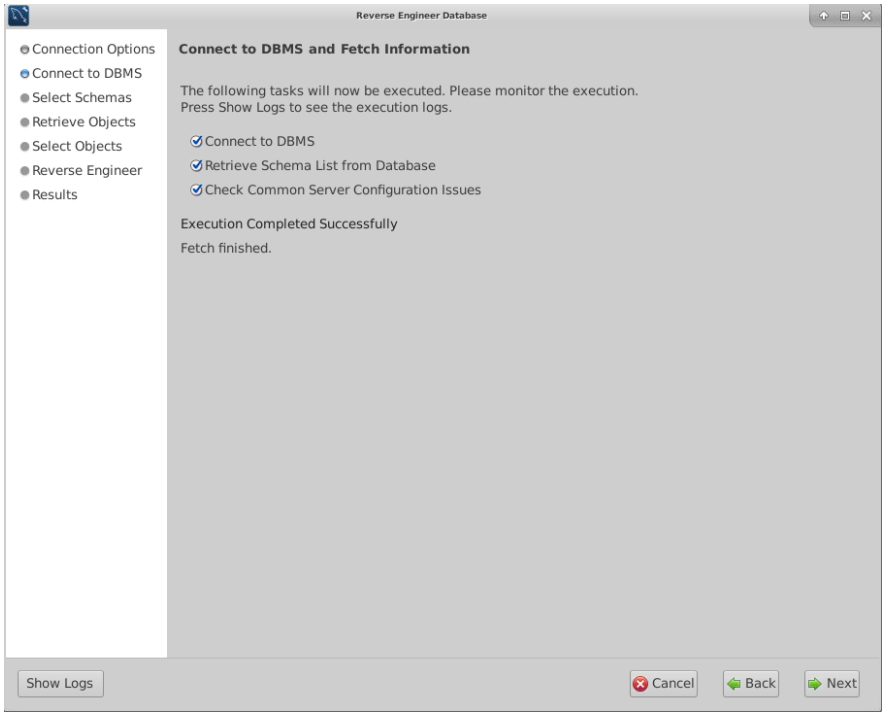
- As the database evolves, database designers need to update the respective EER diagram or the logical model, so the end-users can see the database map, write queries, and generate reports.
- To reverse engineer an existing database, first, go to the `Home screen` then select the desired database connection. In this example, the database connection is `CIND110-lab2`.

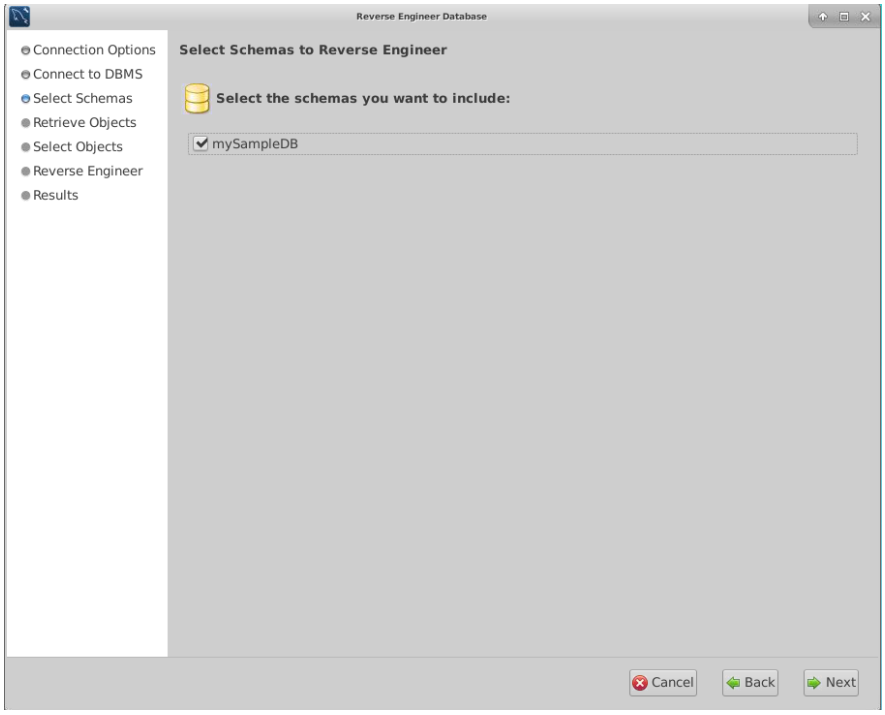


- This can be achieved by selecting the `Database -> Reverse Engineer ...` menu item, then adjusting the parameters of the stored connection, hostname, username and password.



- Before selecting the working database, ensure that the system connected successfully to the DBMS and retrieved the schema list from the specified database connection.





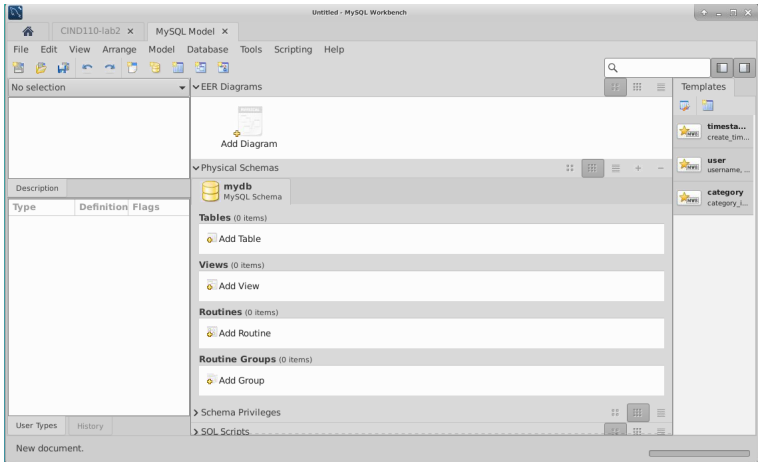
- The output of the reverse engineering step is an EER diagram or a logical model representing the relationships between the database entities.

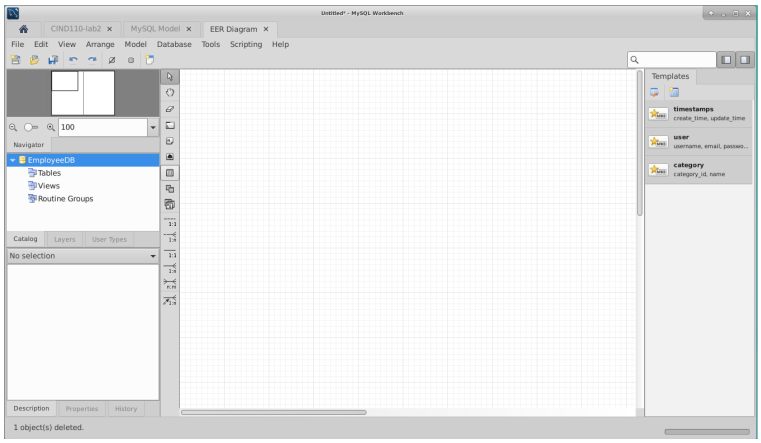
6 Example

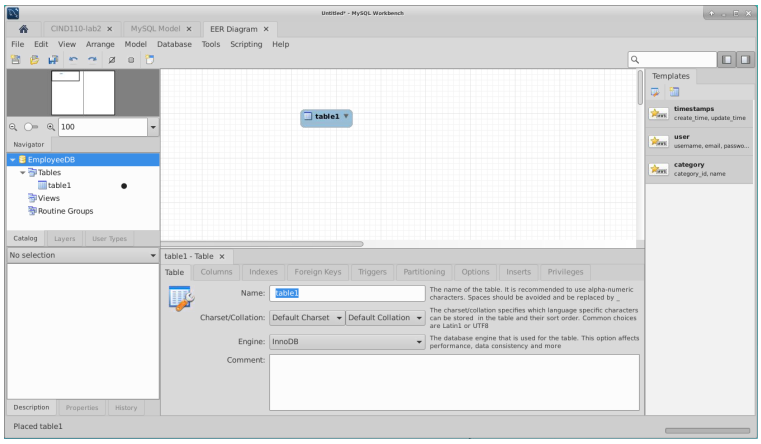
Given below are the details of the Employee database schema.

Table	Attribute	Data Type	Primary Key	Foreign Key	Not Null
Employee	Emp_No	INT(11)	Yes		Yes
	Birth_Date	DATE			
	First_Name	VARCHAR(14)			Yes
	Last_Name	VARCHAR(16)			Yes
	Gender	ENUM('M', 'F')			Yes
	Hire_Date	DATE			
Department	Dept_No	CHAR(4)	Yes		Yes
	Dept_Name	VARCHAR(40)			
Dept_Emp	Emp_No	INT(11)		Yes	
	Dept_No	CHAR(4)		Yes	
	From_Date	DATE			
	To_Date	DATE			
Dept_Manager	Dept_No	CHAR(4)		Yes	
	Emp_No	INT(11)		Yes	
	From_Date	DATE			
	To_Date	DATE			
Title	Emp_No	INT(11)		Yes	
	Title_attr	VARCHAR(50)			
	From_Date	DATE			
	To_Date	DATE			
Salary	Emp_No	INT(11)		Yes	
	Salary_attr	INT(11)			
	From_Date	DATE			
	To_Date	DATE			

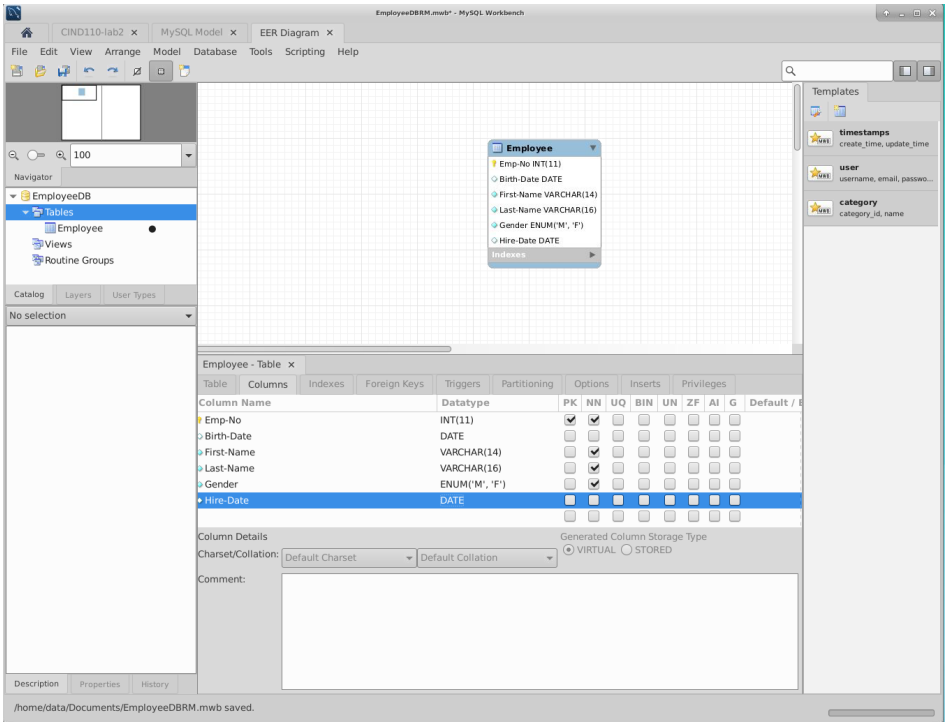
- Click on the `File -> New Model` menu item, then select `Add Diagram` from the `EER Diagrams` panel. You will see the EER diagram window with the tools required to start designing the respective logical model.
- Change the name of the database to `EmployeeDB`, then create the entities/tables by clicking the grid and choose the tool that places a new table inside the grid.
- After placing the table/entity on the grid, configure the properties of that table by double-clicking its representing icon.

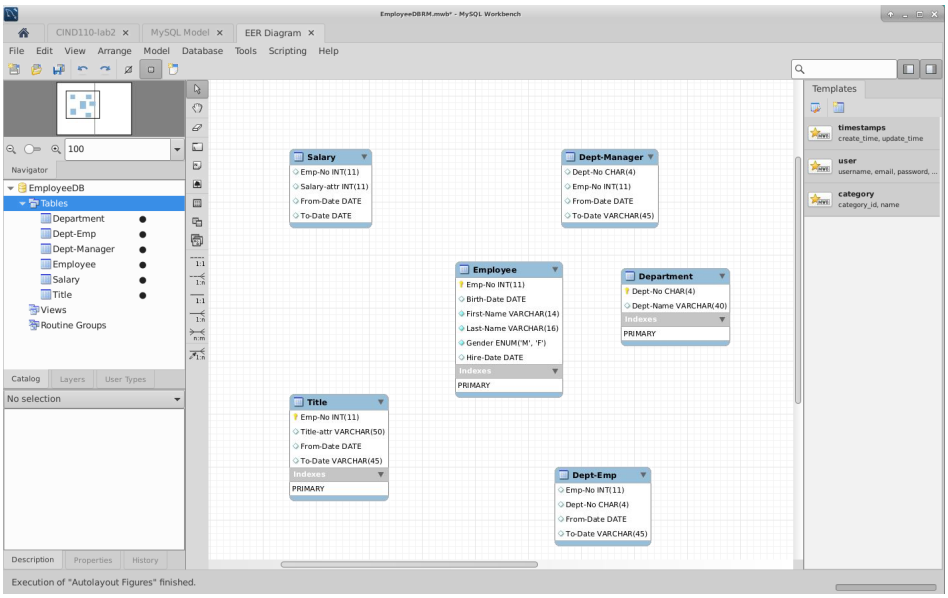




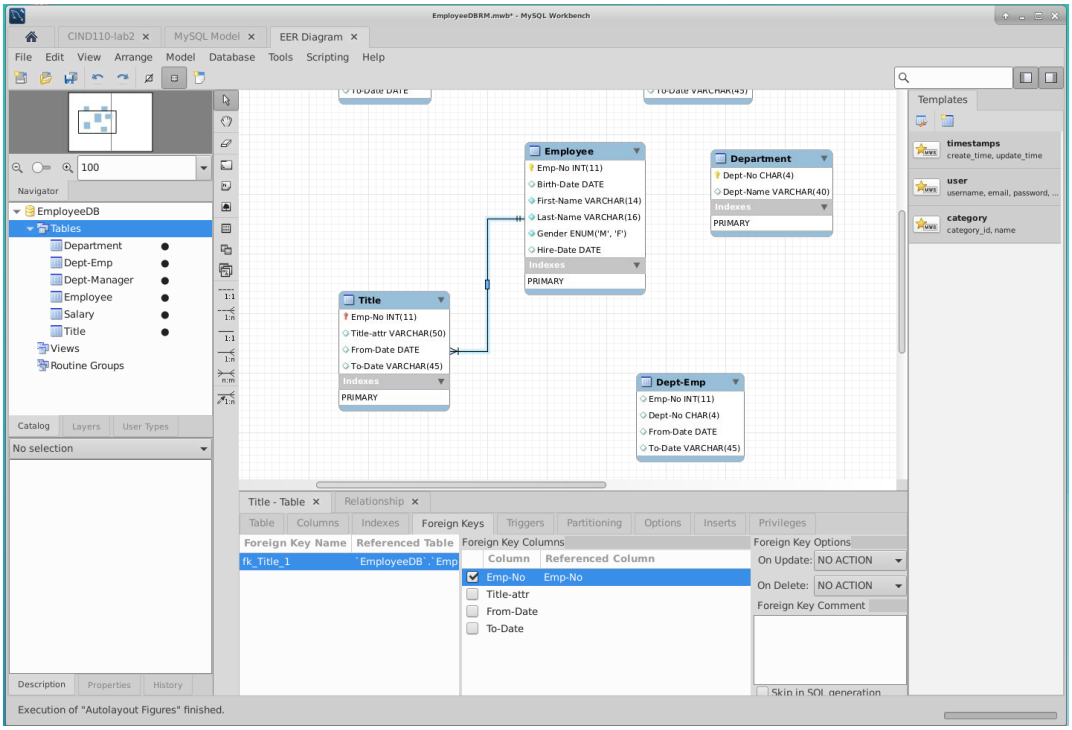


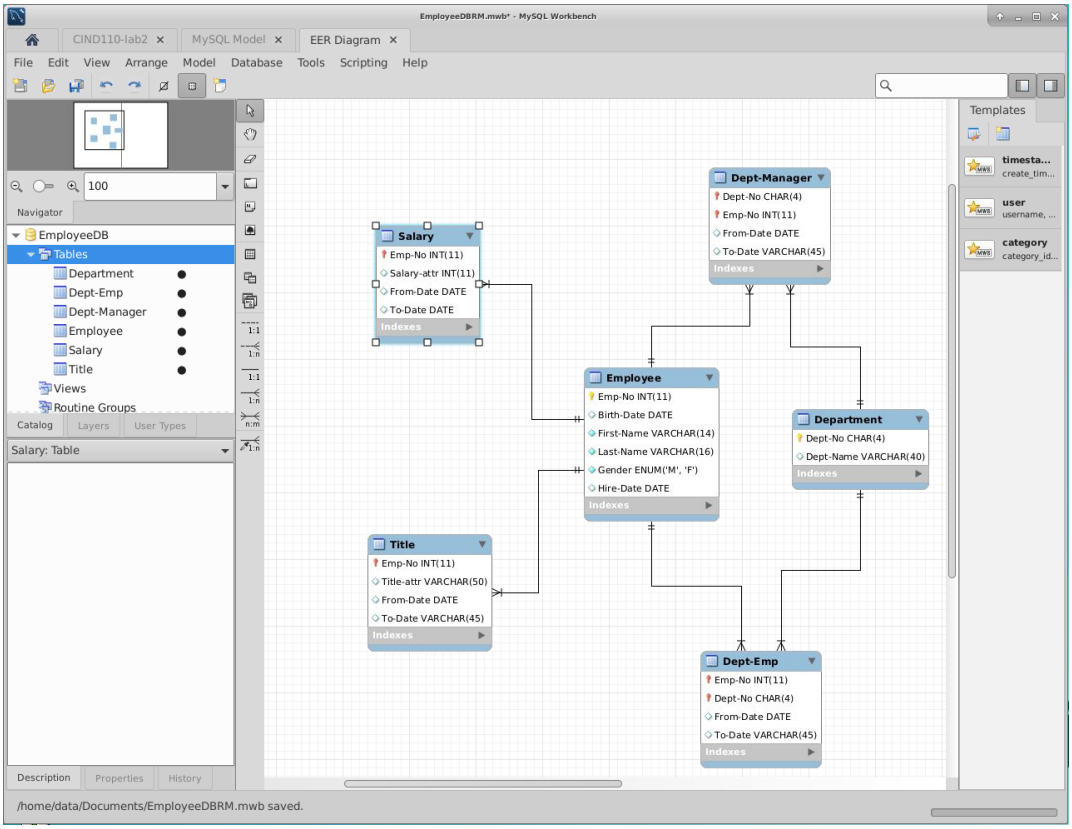
- After configuring the properties of the first table, create the other tables/entities as mentioned earlier in the `EmployeeDB` schema.





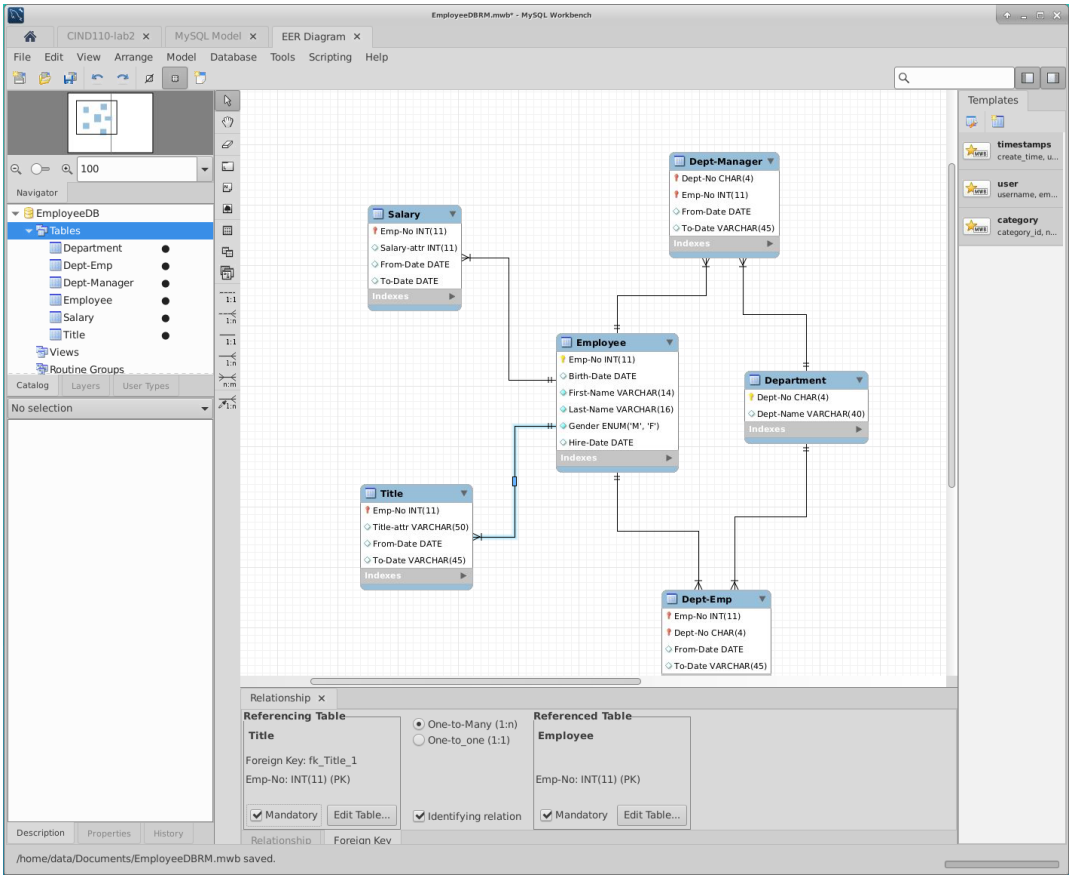
- Ensure adding the required relationships by specifying the foreign key before selecting the referenced table from the `Referenced Table` tab. For example, the `Emp_No` attribute is a foreign key in the `Title` table referencing the `Emp_No` attribute from the `Employee` relation/entity. Similarly, the `Dept_No` attribute is a foreign key in the `Dept_Manager` table referencing the `Dept_No` attribute from the `Department` relation/entity. Notably, Both foreign keys and referenced attributes should be of the same datatype.


EmployeeDBRM.mwb* - MySQL Workbench
CIND110-lab2
MySQL Model
EER Diagram
File Edit View Arrange Model Database Tools Scripting Help
Navigator
EmployeeDB
Tables
Department
Dept-Emp
Dept-Manager
Employee
Salary
Title
Views
Routine Groups
Catalog Layers User Types
No selection
Description Properties History
Employee
Emp-No INT(11)
Birth-Date DATE
First-Name VARCHAR(14)
Last-Name VARCHAR(16)
Gender ENUM('M', 'F')
Hire-Date DATE
Indexes
PRIMARY
Department
Dept-No CHAR(4)
Dept-Name VARCHAR(40)
Indexes
PRIMARY
Title
Emp-No INT(11)
Title-attr VARCHAR(50)
From-Date DATE
To-Date VARCHAR(45)
Indexes
PRIMARY
Dept-Emp
Emp-No INT(11)
Dept-No CHAR(4)
From-Date DATE
To-Date VARCHAR(45)
Templates
timestamps
create_time, update_time
user
username, email, password, ...
category
category_id, name
Title - Table
Relationship
Table Columns Indexes Foreign Keys Triggers Partitioning Options Inserts Privileges
Foreign Key Name
Referenced Table
Foreign Key Columns
Column
Referenced Column
fk_Title_1
`EmployeeDB`.`Emp
Emp-No
Emp-No
Title-attr
From-Date
To-Date
Foreign Key Options
On Update: NO ACTION
On Delete: NO ACTION
Foreign Key Comment
Skip in SQL generation
Execution of "Autolayout Figures" finished.


EmployeeDBRM.mwb* - MySQL Workbench
CIND110-lab2
MySQL Model
EER Diagram
File Edit View Arrange Model Database Tools Scripting Help
Navigator
EmployeeDB
Tables
Department
Dept-Emp
Dept-Manager
Employee
Salary
Title
Views
Routine Groups
Catalog Layers User Types
Salary: Table
Description Properties History
Salary
Emp-No INT(11)
Salary-attr INT(11)
From-Date DATE
To-Date DATE
Indexes
Dept-Manager
Dept-No CHAR(4)
Emp-No INT(11)
From-Date DATE
To-Date VARCHAR(45)
Indexes
Employee
Emp-No INT(11)
Birth-Date DATE
First-Name VARCHAR(14)
Last-Name VARCHAR(16)
Gender ENUM('M', 'F')
Hire-Date DATE
Indexes
Department
Dept-No CHAR(4)
Dept-Name VARCHAR(40)
Indexes
Title
Emp-No INT(11)
Title-attr VARCHAR(50)
From-Date DATE
To-Date VARCHAR(45)
Indexes
Dept-Emp
Emp-No INT(11)
Dept-No CHAR(4)
From-Date DATE
To-Date VARCHAR(45)
Indexes
Templates
timesta...
create_tim...
user
username, ...
category
category_id...
/home/data/Documents/EmployeeDBRM.mwb saved.

- There are two types of entities: regular/dominant/strong and subordinate/weak. While weak entities do not have their own key attributes, the regular ones have their own key attributes. The relationship type that relates a weak entity to its owner/parent is called the identifying relationship of the weak entity. Notably, weak entities have a total existence dependency for their identifying relationship because a weak entity cannot be identified without an owner entity. To adjust the type of relationship between entities, double click the line connecting two entities and choose whether the relationship is `identifying` or not. In the EER diagram, the dotted line refers to non-identifying relationships, while the solid ones refer to the identifying relationships.

- You can also adjust the carnality ratios between entities, whether they are `one-to-one`, `one-to-many`, or `many-to-many`, by double-clicking the relationship line and adjust the carnality ratio. For example, the relationship between `Employee` and `Title` is a one-to-many relationship, as one employee can hold more than one title.



This is the end of lab2
Tamer Abdou, PhD