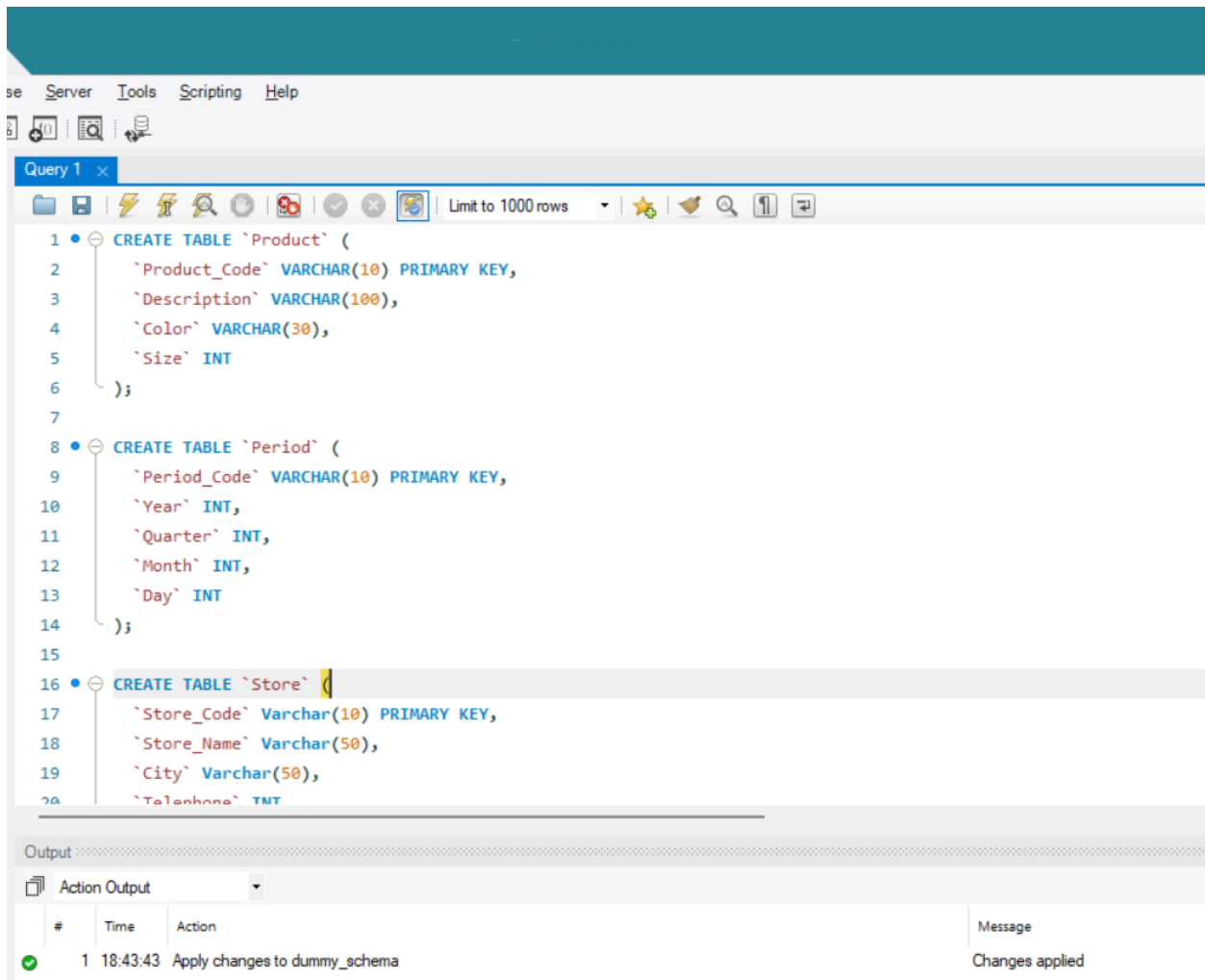
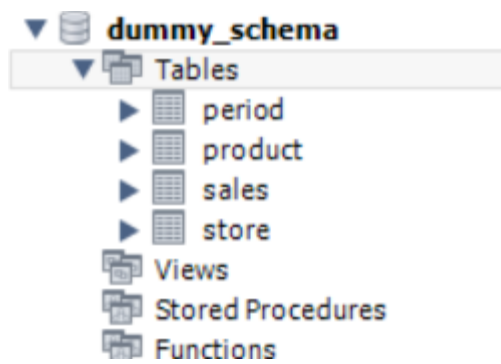


3. MySQL WorkBench

i. Reverse Engineering

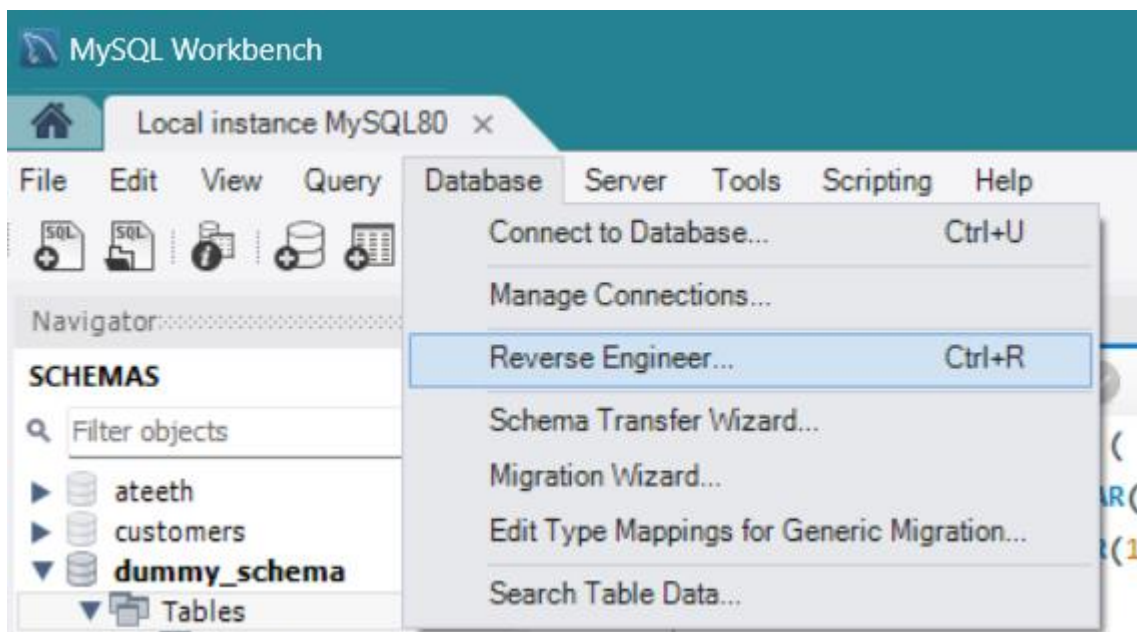
1. Create a new schema(database) and execute the sample code for that database so that all tables are created in that database.



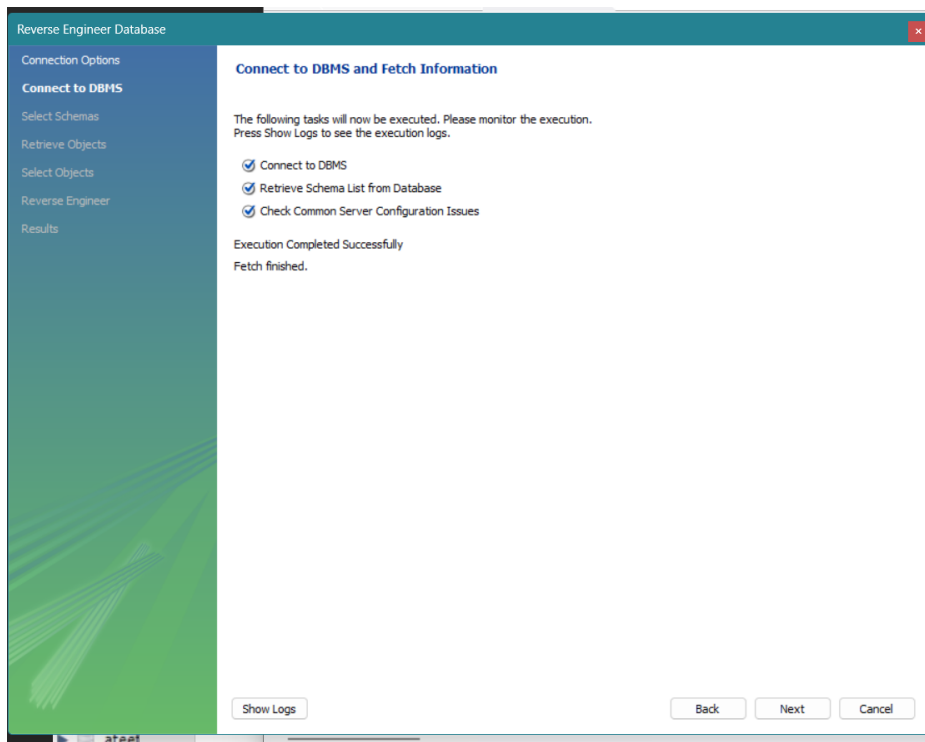


ii. In the top bar click on Database and in the dropdown

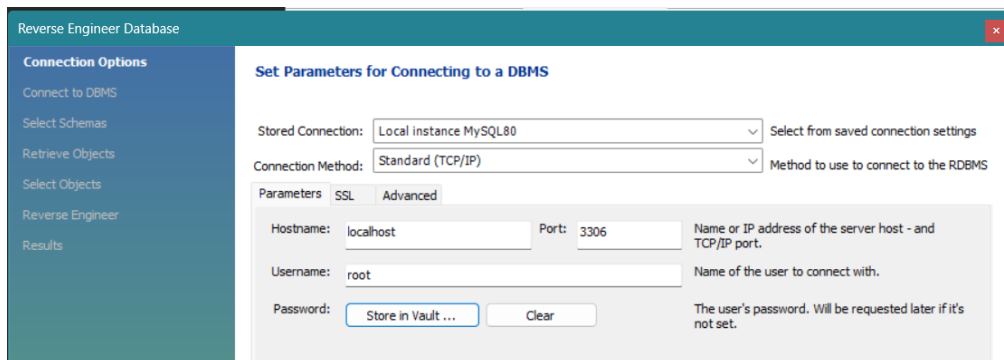
II. click on Reverse Engineer



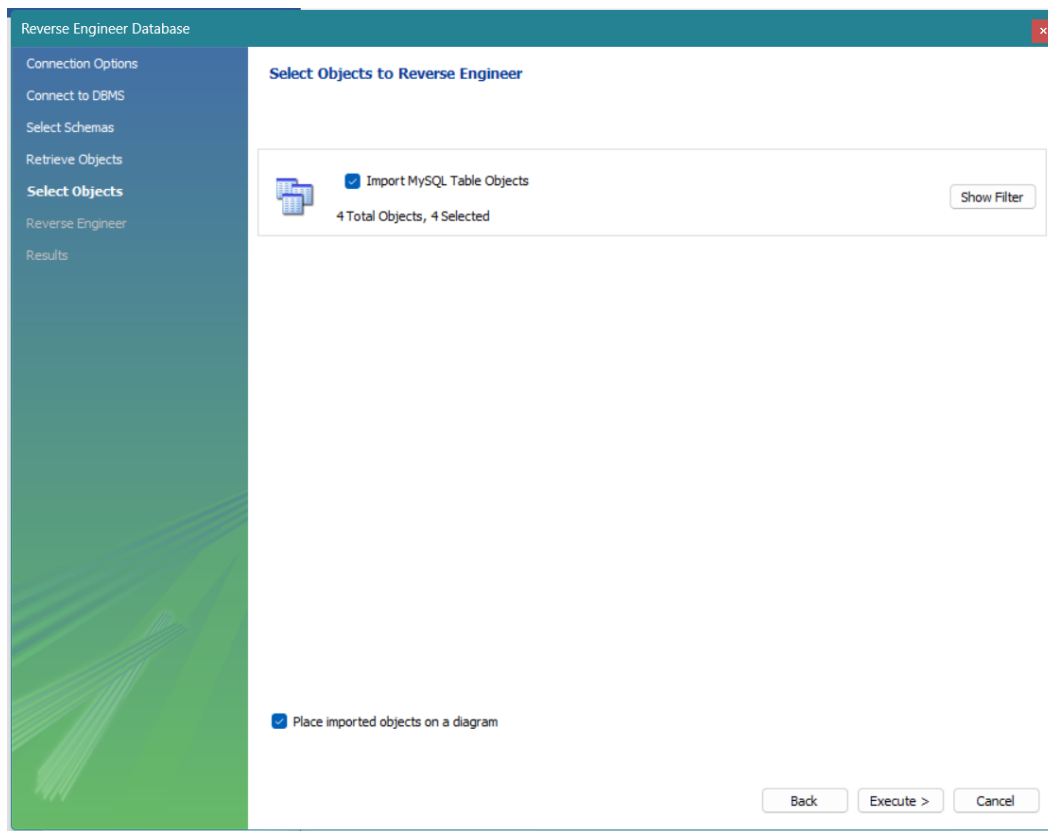
III. Fill in the following details regarding the database



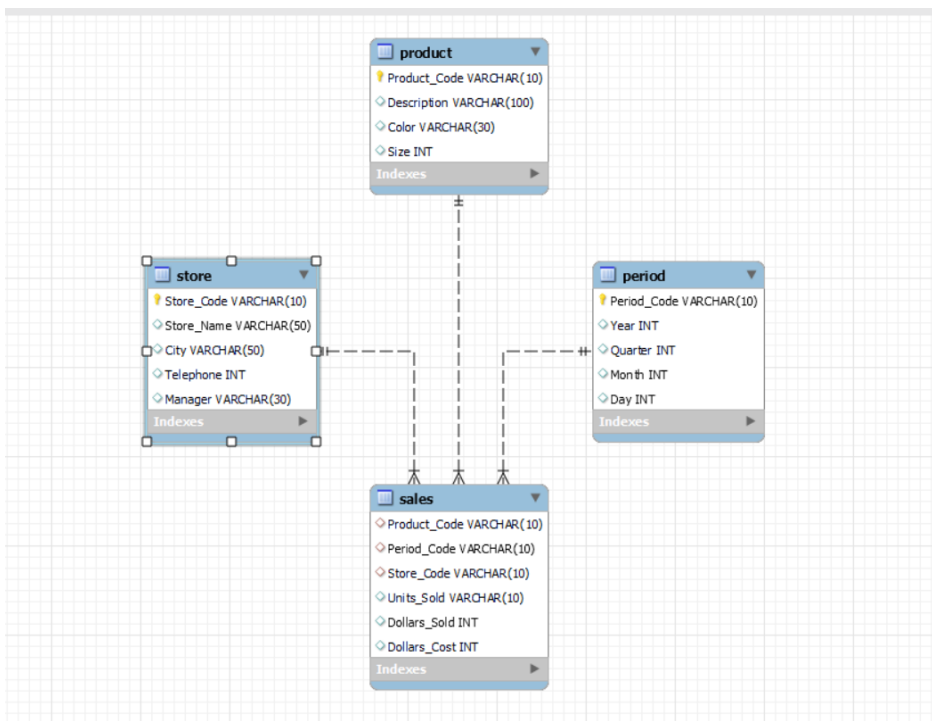
IV. The following message will be visible



V. Select the schema to import and the tables that are to be imported then execute



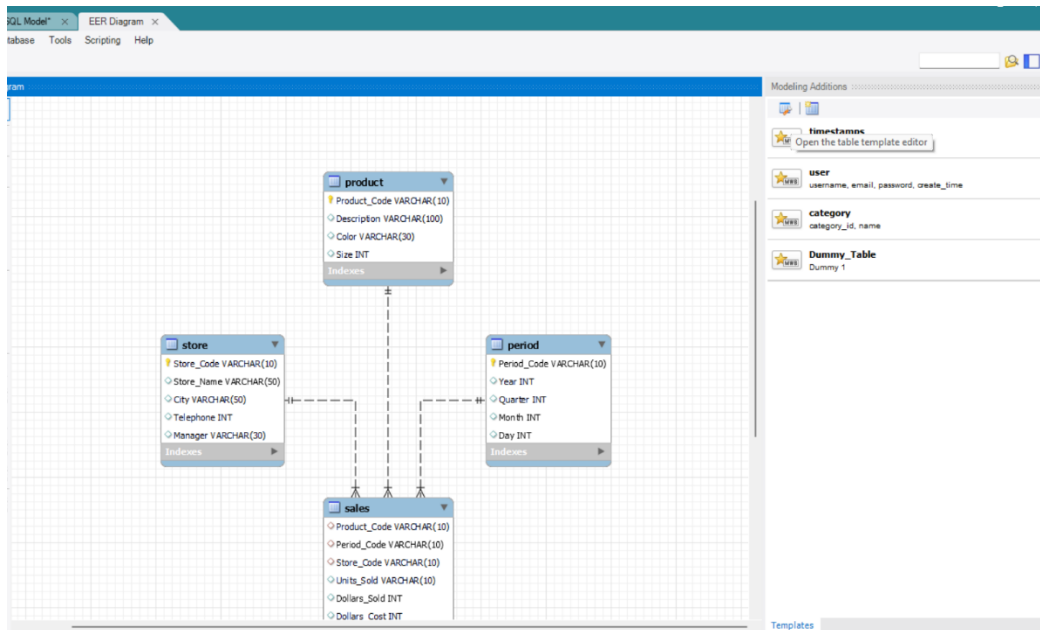
VI. The ER Diagram will be visible when Execute is clicked



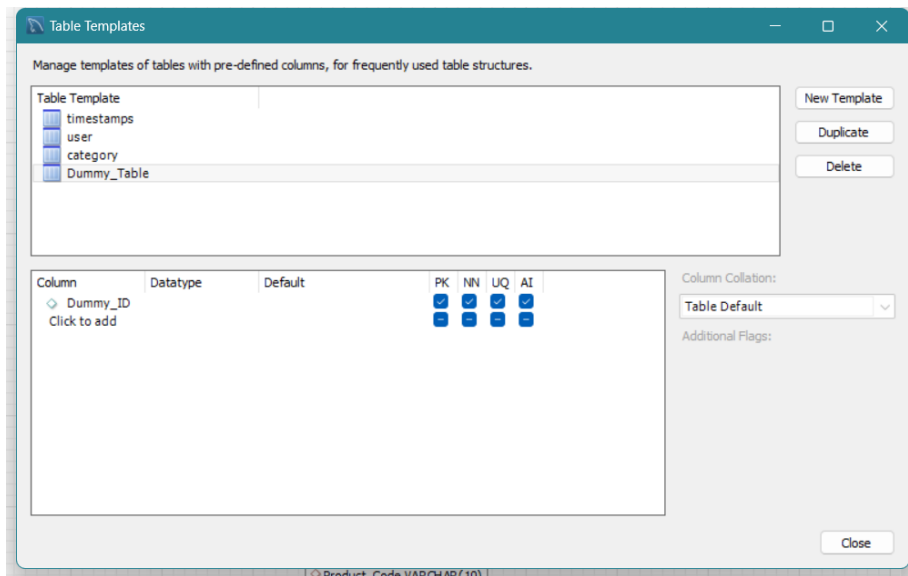
ii. Add a new table in the ER Diagram

Notice on the right side Modeling Additions

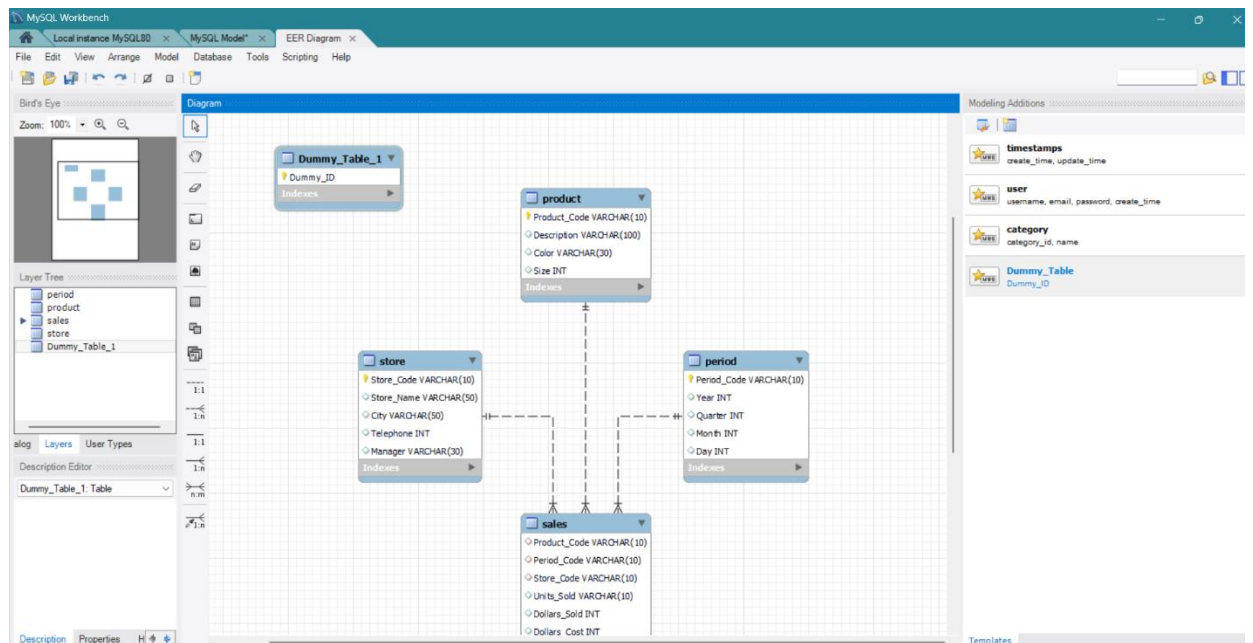
I. Click the first icon that is Open the table editor



II. Click new template rename it by double clicking the row it appears and in the bottom part of the pop up all attributes needed can be added



- III. To add the table to the Schema on the right side modeling Additions just select the table name to be added it will be added in this case Dummy_Table is selected



Note :- Sales table will be edited to add the attribute Dummy_ID as a foreign key referring Dummy_Table

The screenshot displays the MySQL Workbench EER Diagram editor. The main workspace shows an Entity-Relationship Diagram with the following entities and attributes:

- product**: Product_Code VARCHAR(10), Description VARCHAR(100), Color VARCHAR(30), Size INT. Indices: Product_Code.
- store**: Store_Code VARCHAR(10), Store_Name VARCHAR(50), City VARCHAR(50), Telephone INT, Manager VARCHAR(30). Indices: Store_Code.
- period**: Period_Code VARCHAR(10), Year INT, Quarter INT, Month INT, Day INT. Indices: Period_Code.
- sales**: Product_Code V, Period_Code V, Store_Code V, Units_Sold VAR, Dollars_Sold IN, Dollars_Cost IN. Indices: Product_Code, Period_Code, Store_Code.
- Dummy_ID**: A table with an index.
- Dummy_Table**: A table with a column Dummy_ID.

Relationships are indicated by dashed lines with crow's foot notation:

- product** to **sales**: 1:M relationship.
- store** to **sales**: 1:M relationship.
- period** to **sales**: 1:M relationship.

A context menu is open over the **sales** table, showing options such as 'Cut 'sales'', 'Copy 'sales'', 'Edit 'sales'...', 'Delete 'sales'', and 'Remove Figure 'sales''.

The interface includes a menu bar (File, Edit, View, Arrange, Model, Database, Tools, Scripting, Help), a toolbar, and several panels:

- Layer Tree**: Lists layers like period, product, sales, store, and Dummy_Table_1.
- Modeling Additions**: Lists modeling additions like timestamps, user, category, and Dummy_Table.
- Templates**: A panel for templates.

The screenshot shows the MySQL Modeler EER Diagram interface. It displays three tables: **Store**, **sales**, and **Period**.

- Store Table:** Attributes include Store_Code (PK), Store_Name, City, Telephone, and Manager.
- sales Table:** Attributes include Product_Code, Period_Code, and Store_Code (FK).
- Period Table:** Attributes include Year, Quarter, Month, and Day.

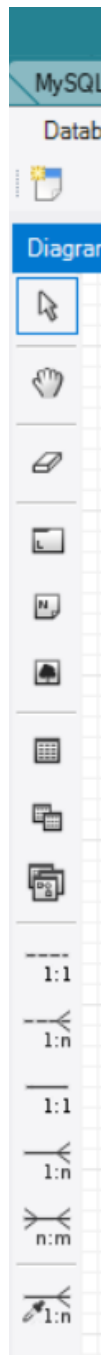
Relationships are indicated by dashed lines with crow's foot notation:

- A one-to-many relationship between **Store** and **sales** on the **Store_Code** attribute.
- A one-to-many relationship between **Period** and **sales** on the **Period_Code** attribute.

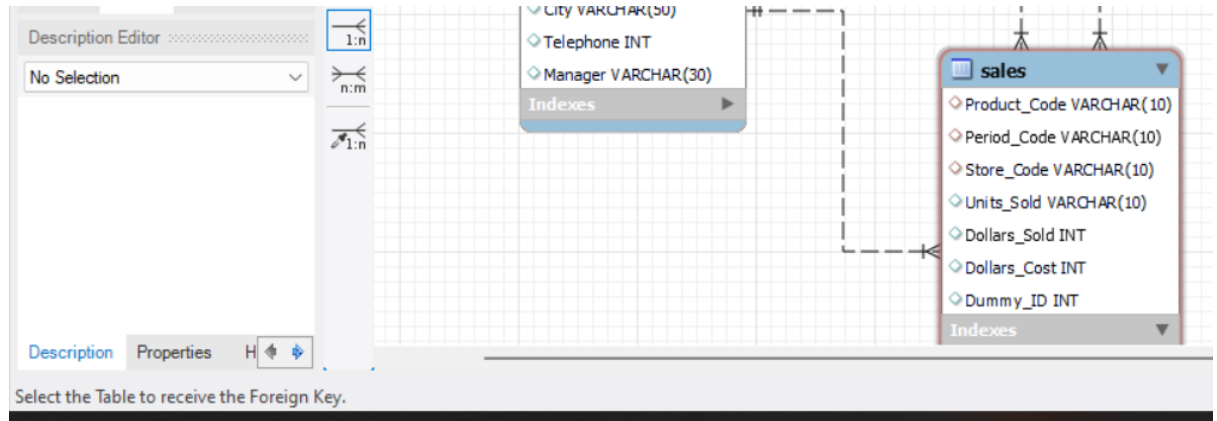
At the bottom, the 'sales - Table' tab is active, showing the table structure for 'sales' in the 'dummy_schema' schema. The table has columns: Dollars_Cost (INT) and Dummy_ID (INT).

iv. Add a foreign key

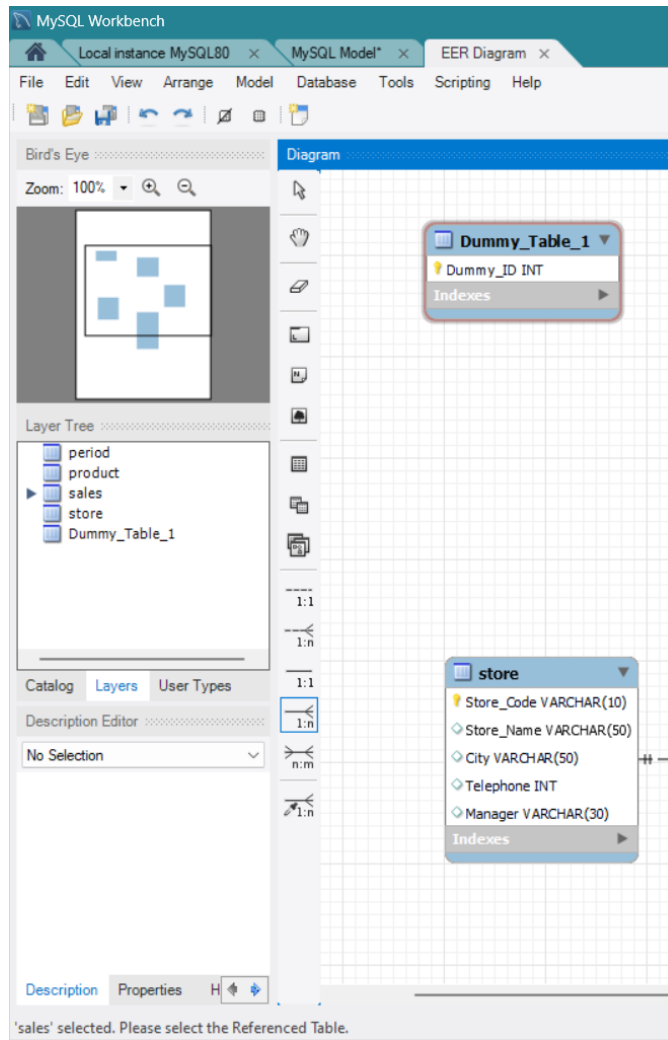
- I. See the bottom 4 options select one of them. In this example 1:n i.e 3rd last icon in above screenshot is selected



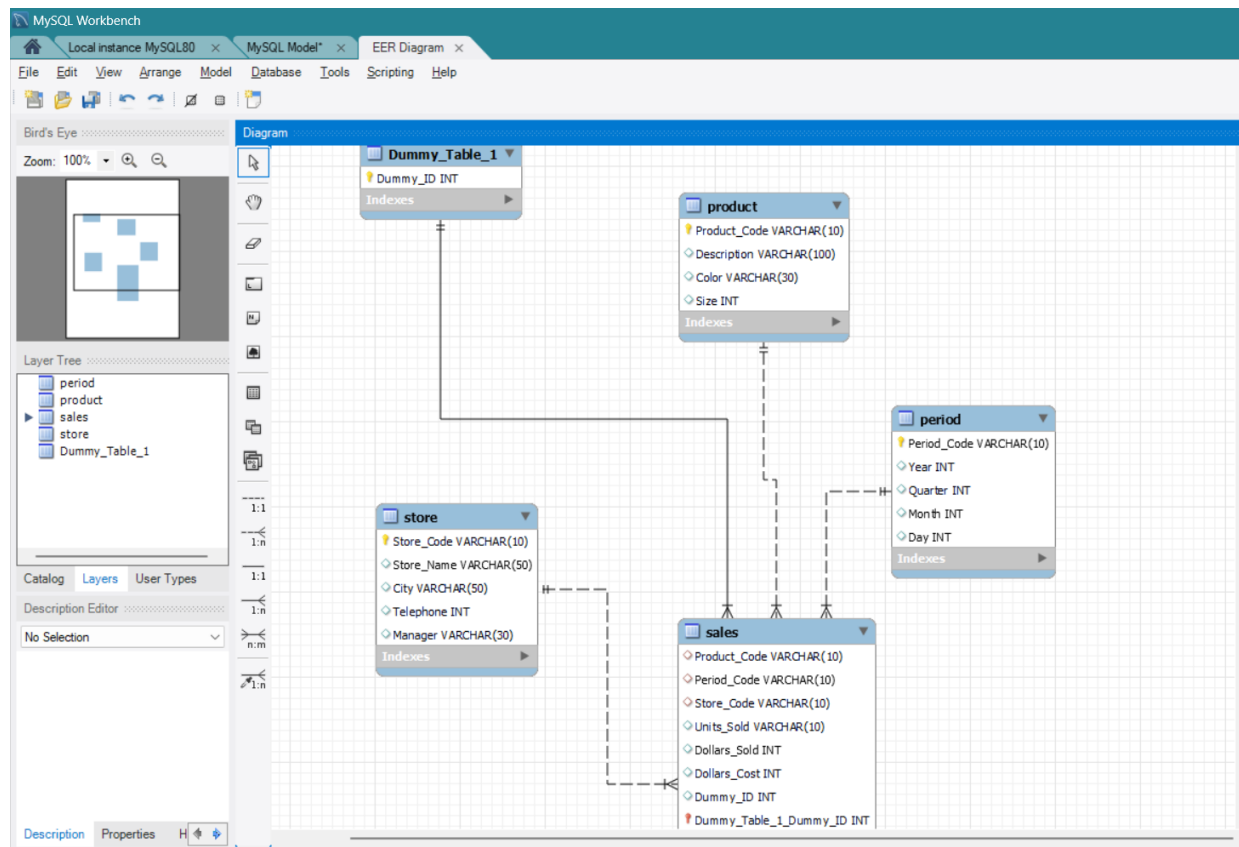
- II. It will prompted to select the table that receives foreign key. In this example Sales table selected



- III. Then select Referenced table. In this example Dummy_Table selected.

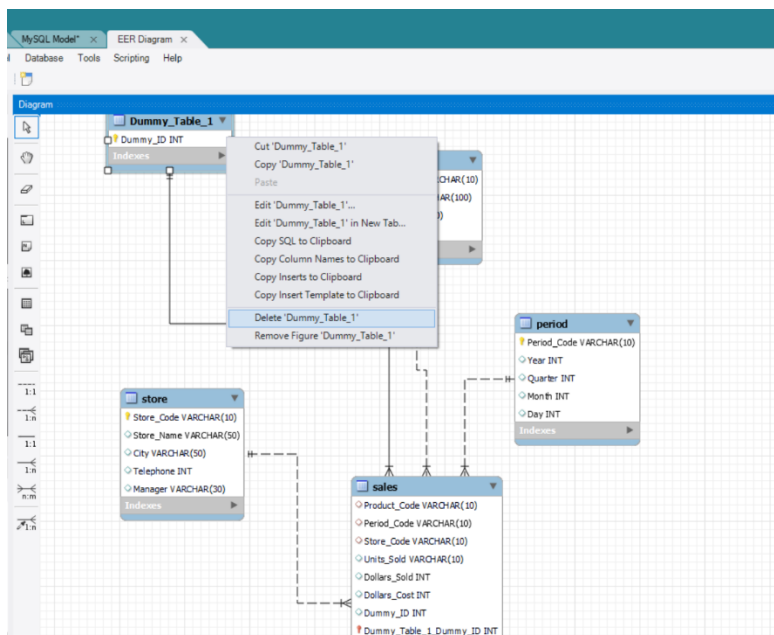


IV. Then the relationship is drawn , in this case between Dummy_Table and Sales

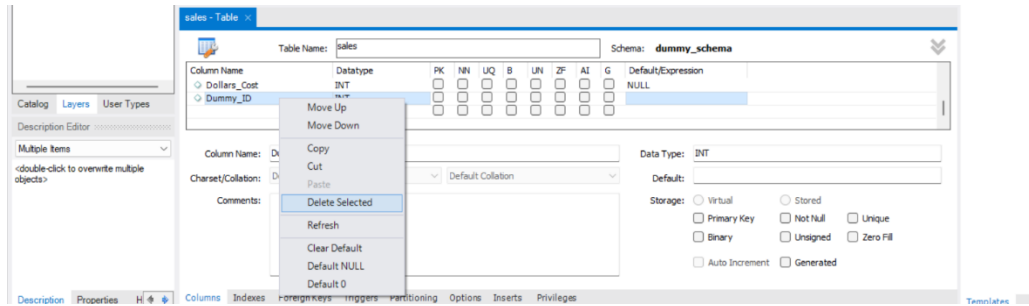


v. Delete a table or attribute

I. Right click the table and click option delete {Table_Name}



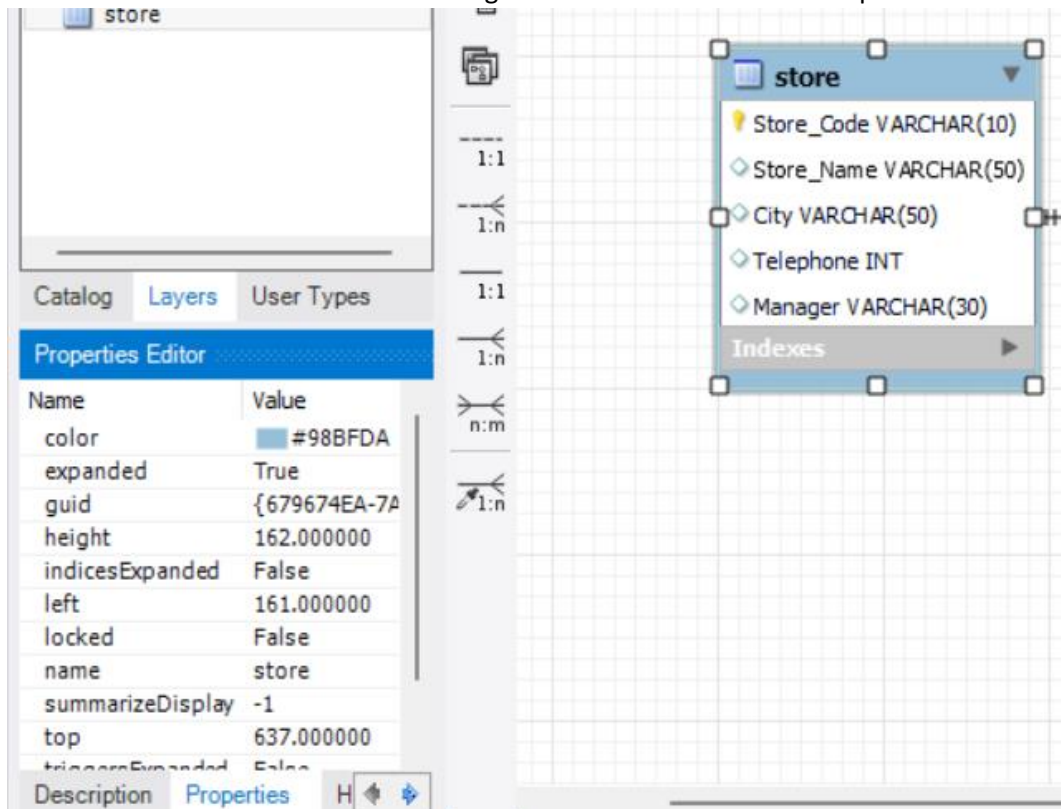
- II. To delete attribute just how attribute was added double click table / right click table and click edit , then just click the attribute row in the tab that opens at the bottom , right click and select delete selected.



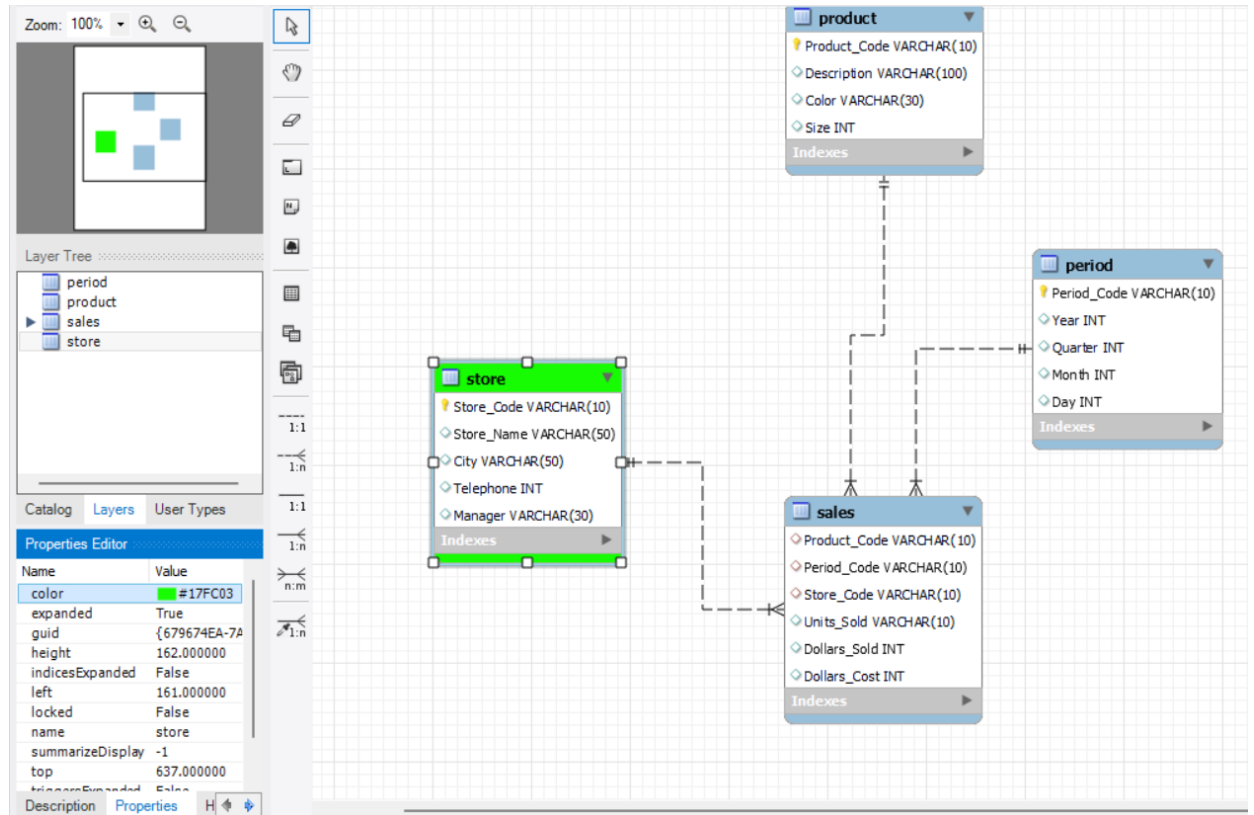
vi. Change the color of a table

Note : store table is considered at first for this example

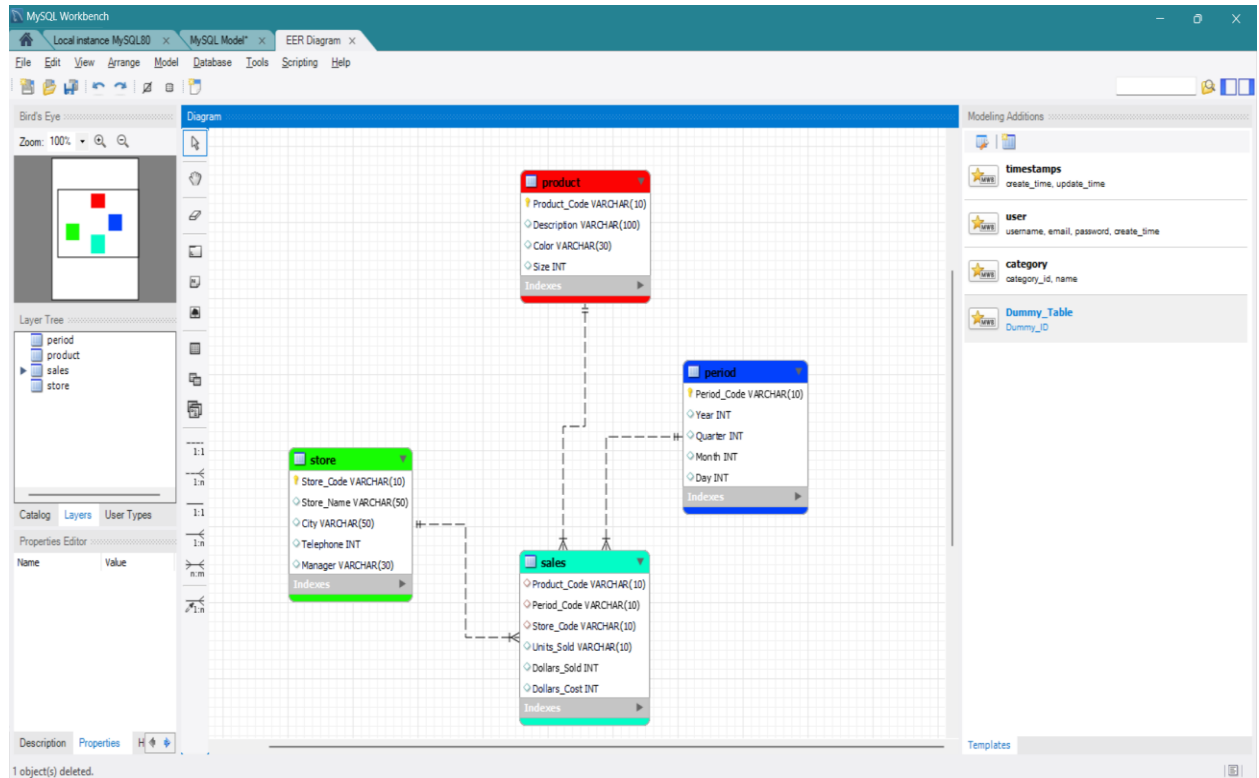
- I. Click the table whose color is to be changed and at bottom left click Properties



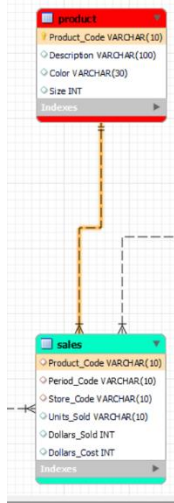
- II. Change the color column value to any hexadecimal color. Note that store table has a new color which is light green after changing the color value. Note the color is now #17fc03



- III. Now the color of all tables can be changed similarly
- Product - #FC0303
 - Store - #17FC03
 - Sales – #03FCC6
 - Period - #0341FC

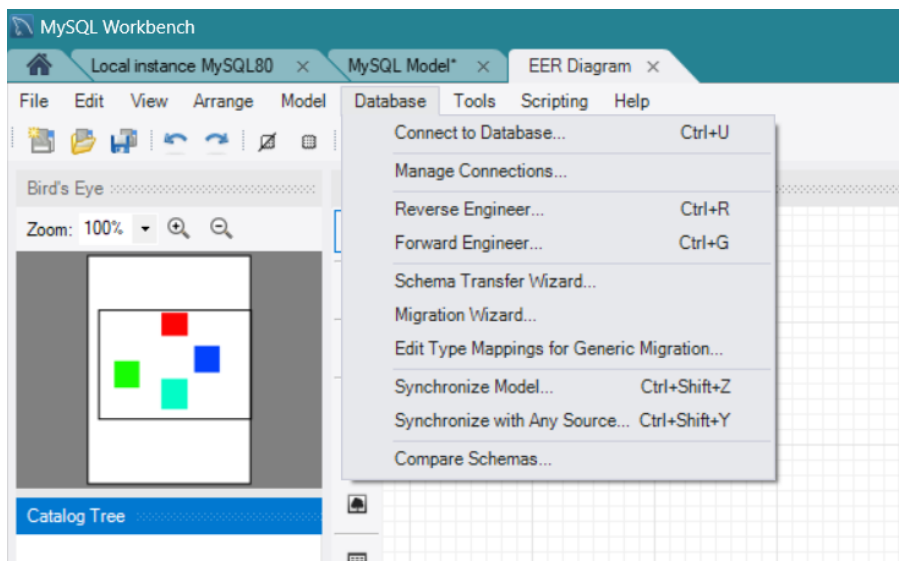


vii. Note when hovering over a relationship line , the color of relationship line and the attributes involved in the relationship change dynamically as seen in the below example.

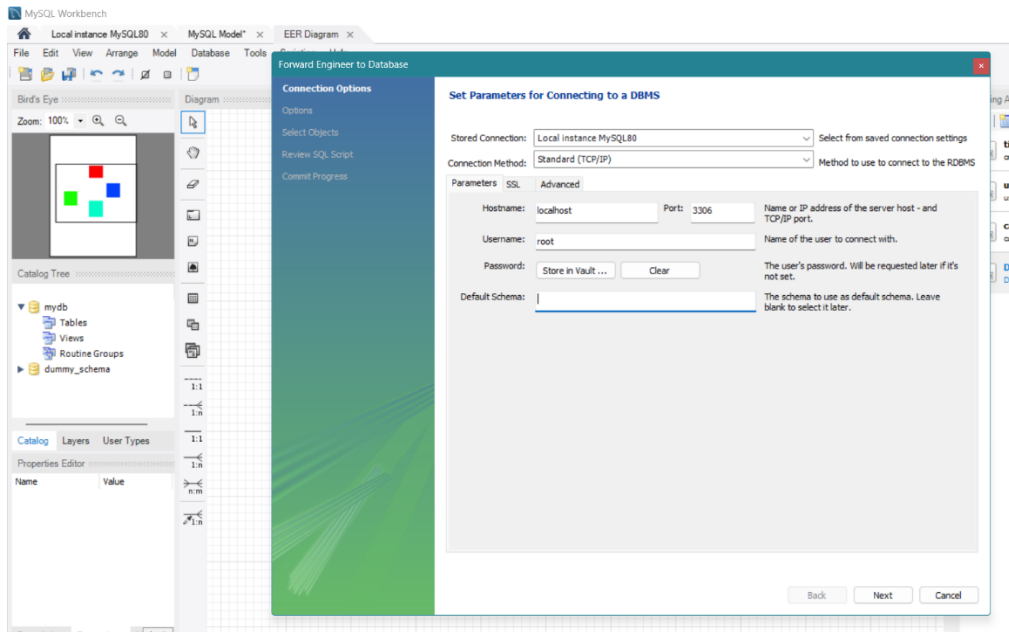


viii. Export the schema diagram i.e Forward Engineering

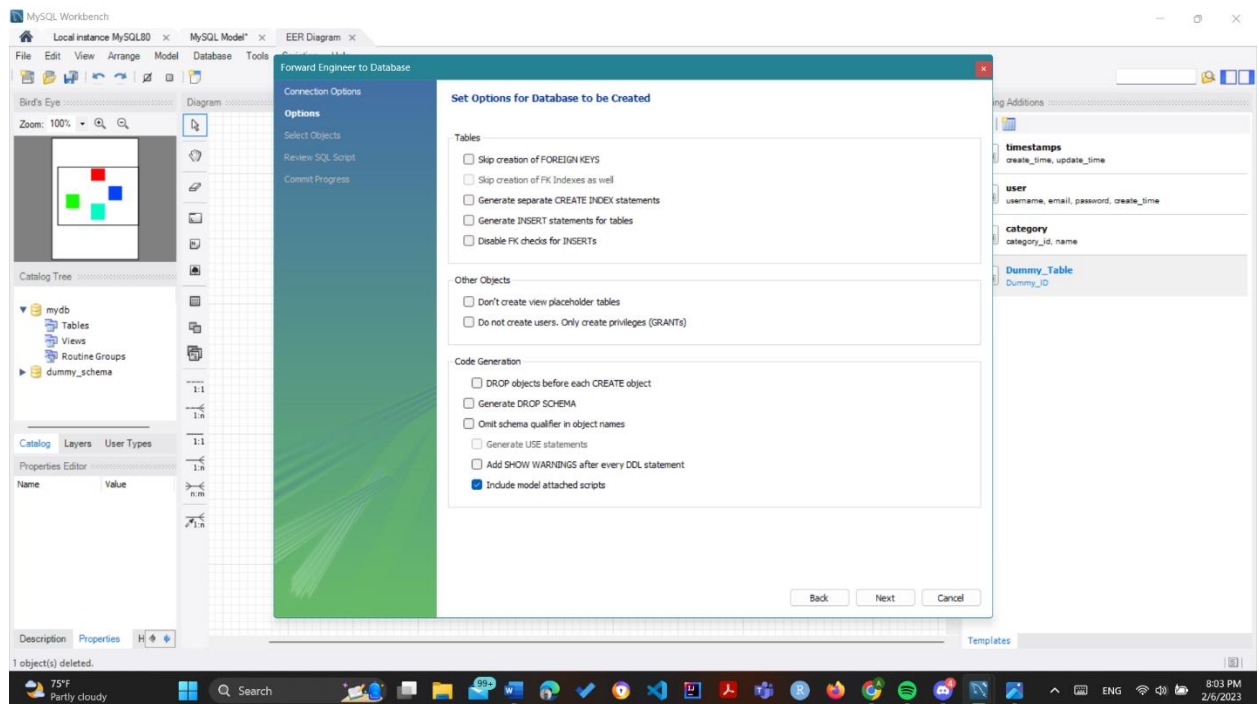
I. Click Database in top menu then Forward Engineering



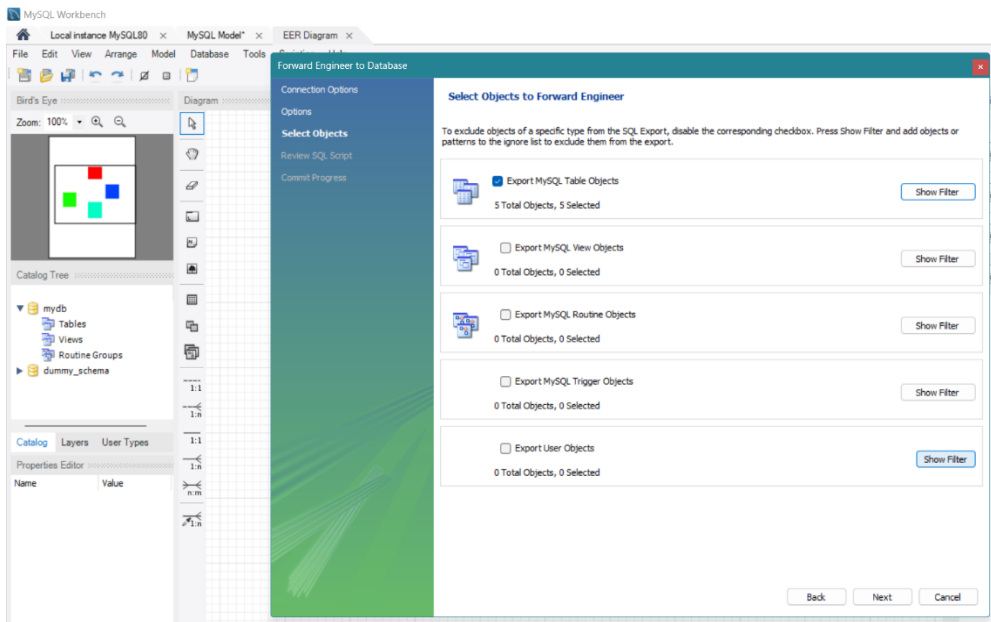
II. Fill Required details



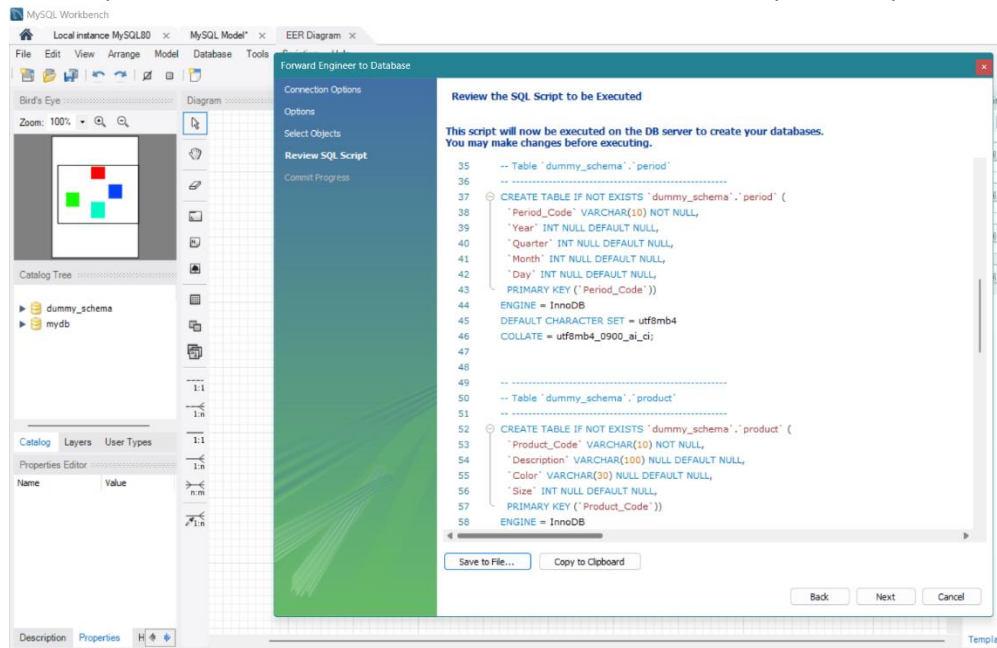
III. Select any options if needed



IV. Select any options if needed



V. SQL Script is visible and can be saved to a file or else it can be copied to clipboard



Then the code is executed and the code can be saved in clipboard or in a file

