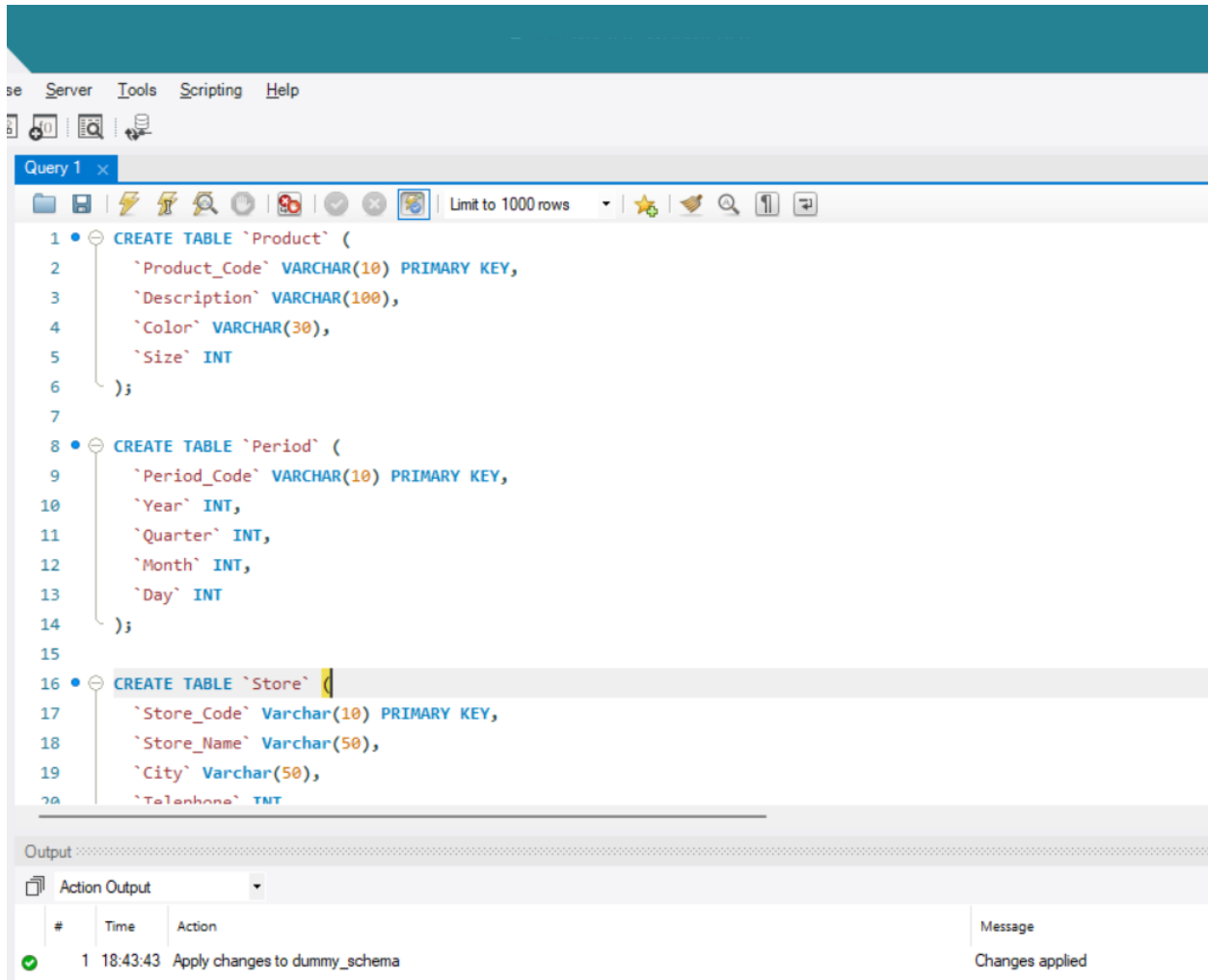
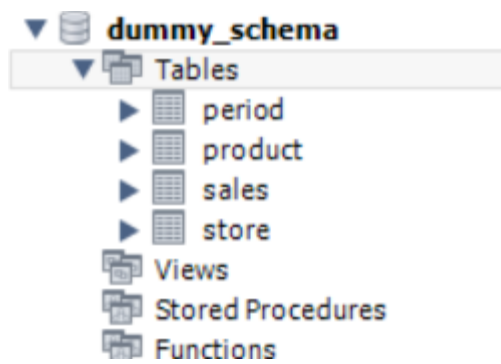


### 3. MySQL WorkBench

#### i. Reverse Engineering

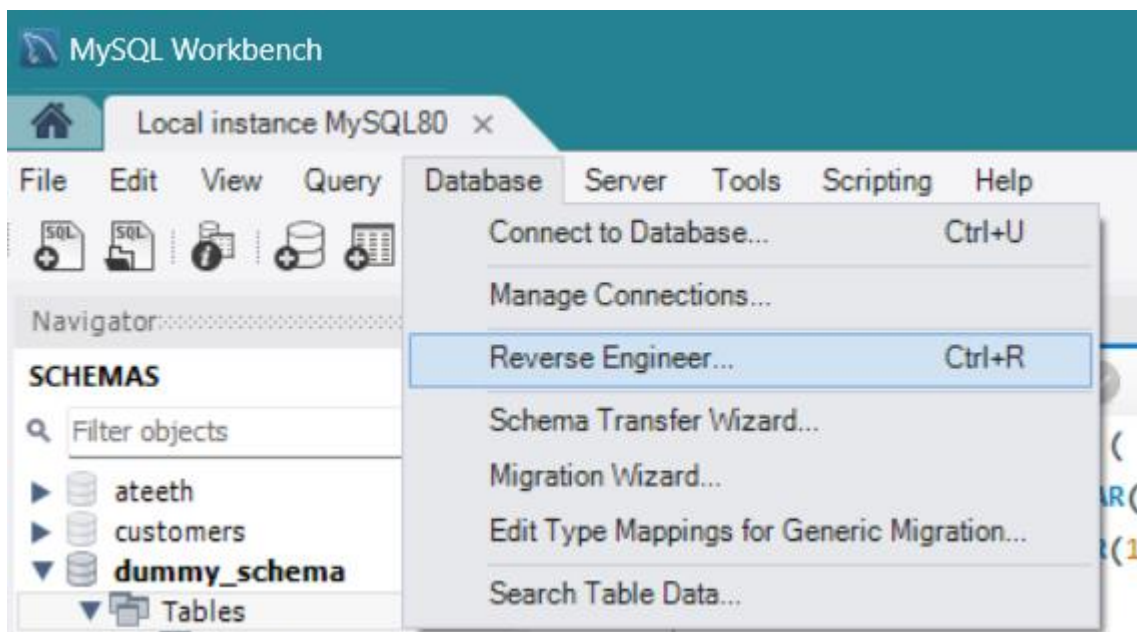
1. Create a new schema(database) and execute the sample code for that database so that all tables are created in that database.



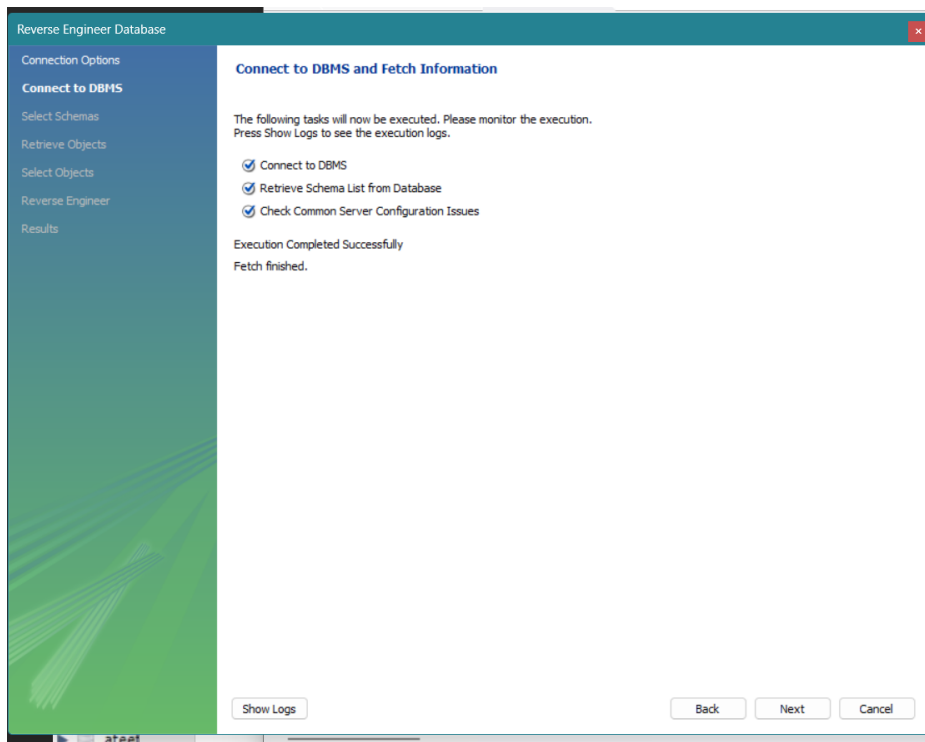


ii. In the top bar click on Database and in the dropdown

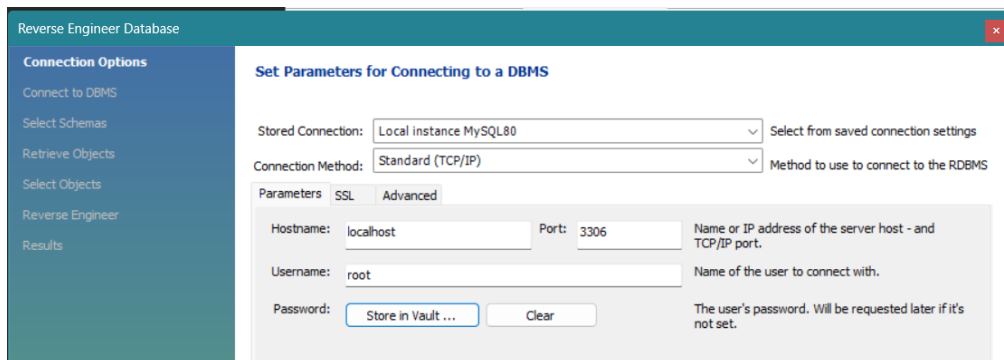
II. click on Reverse Engineer



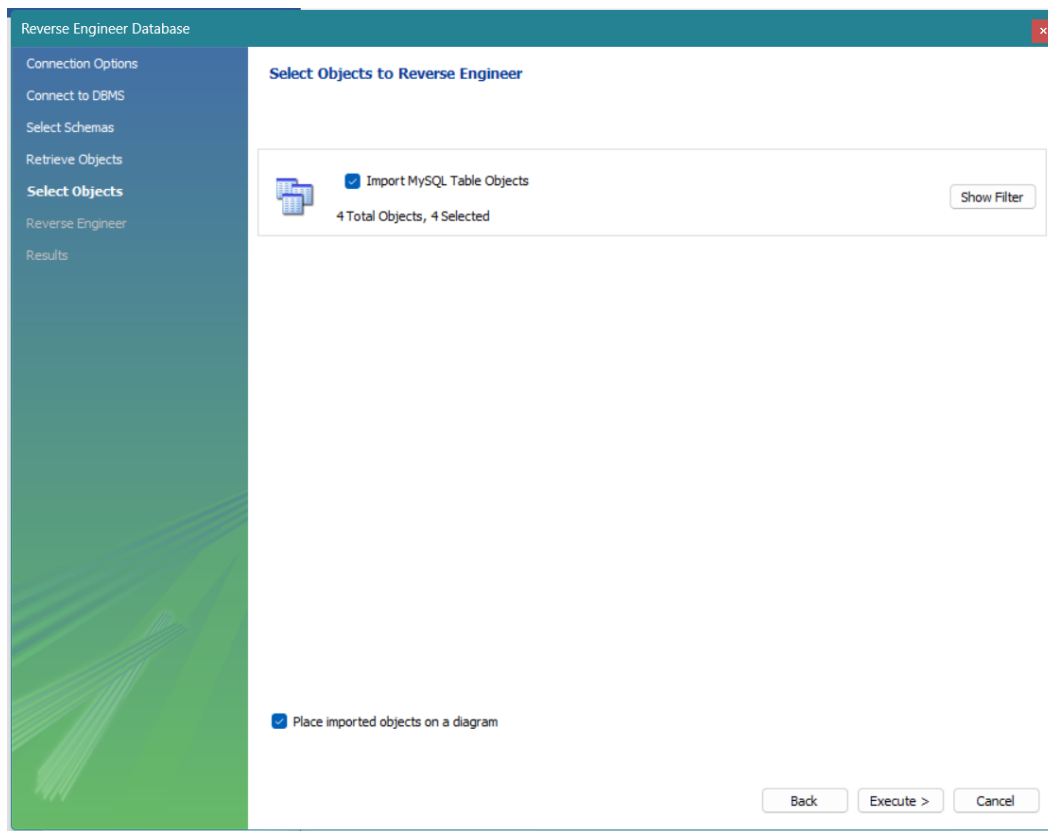
III. Fill in the following details regarding the database



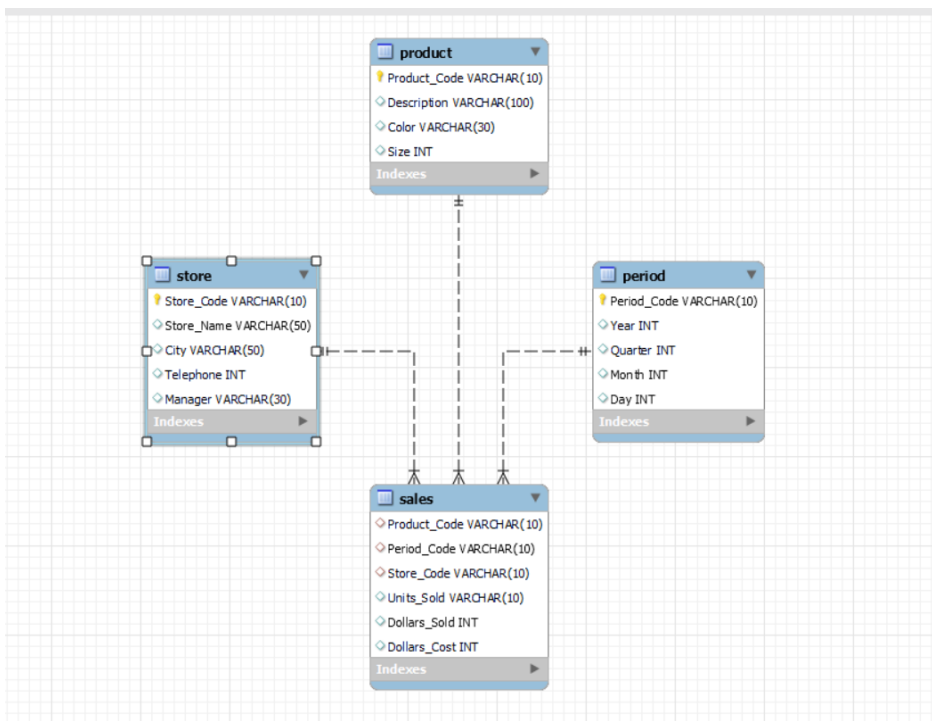
IV. The following message will be visible



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



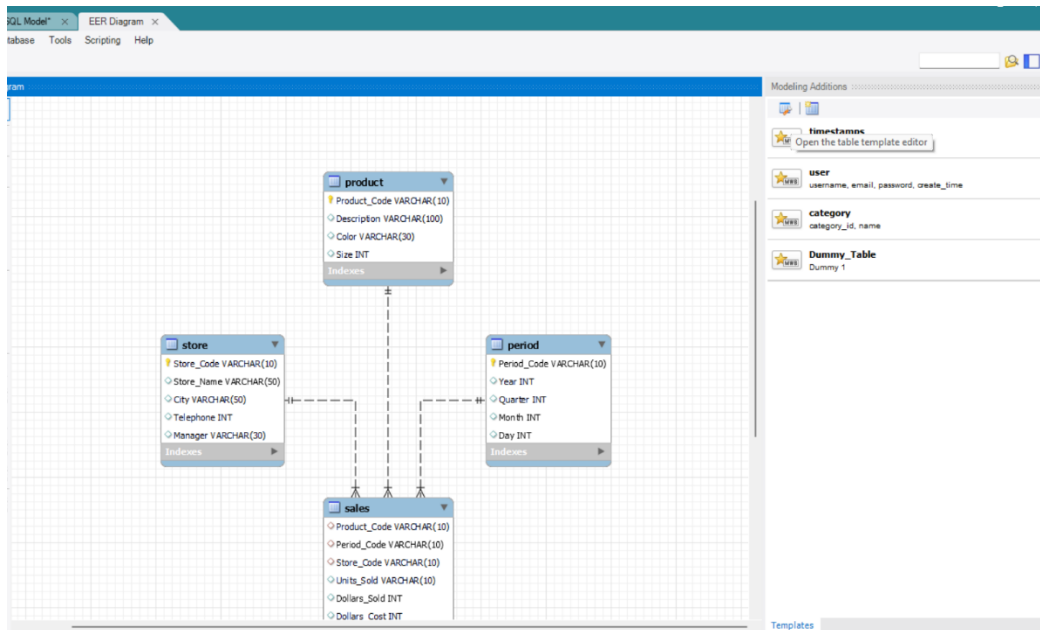
VI. The ER Diagram will be visible when Execute is clicked



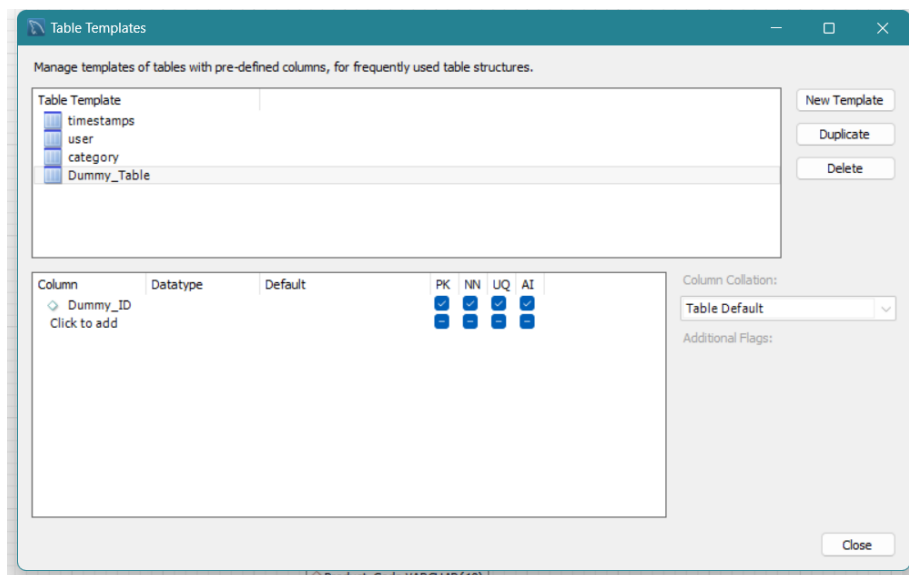
ii. Add a new table in the ER Diagram

Notice on the right side Modeling Additions

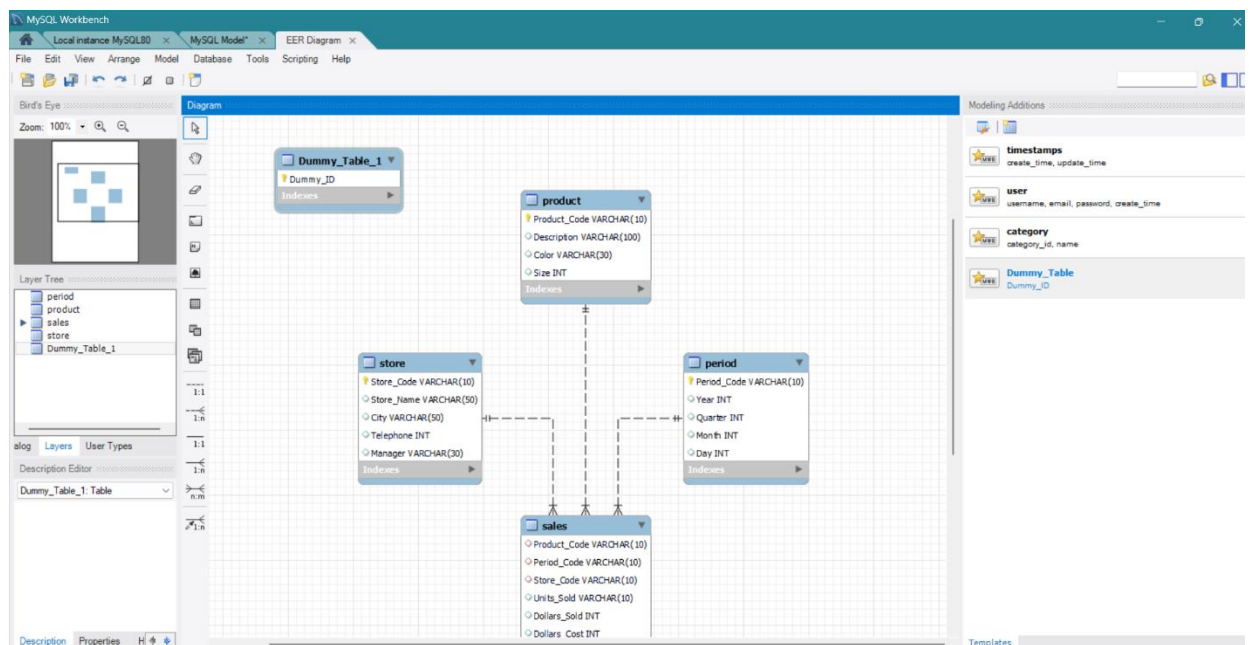
I. Click the first icon that is Open the table editor



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 shows the MySQL Workbench EER Diagram interface. The main workspace displays a database schema with the following tables and attributes:

- product**: Product\_Code VARCHAR(10), Description VARCHAR(100), Color VARCHAR(30), Size INT. (Primary Key: Product\_Code)
- store**: Store\_Code VARCHAR(10), Store\_Name VARCHAR(50), City VARCHAR(50), Telephone INT, Manager VARCHAR(30). (Primary Key: Store\_Code)
- period**: Period\_Code VARCHAR(10), Year INT, Quarter INT, Month INT, Day INT. (Primary Key: Period\_Code)
- sales**: Product\_Code VARCHAR(10), Period\_Code VARCHAR(10), Store\_Code VARCHAR(10), Units\_Sold INT, Dollars\_Sold INT, Dollars\_Cost INT. (Foreign Keys: Product\_Code to product, Period\_Code to period, Store\_Code to store)
- Dummy\_ID**: (Primary Key)
- Dummy\_Table**: Dummy\_ID (Primary Key)

Relationships are shown with dashed lines and crow's foot notation. The **sales** table is highlighted, and a context menu is open over it, showing options like 'Cut 'sales'', 'Copy 'sales'', 'Edit 'sales'...', etc.

The right sidebar shows the 'Modeling Additions' panel with the following items:

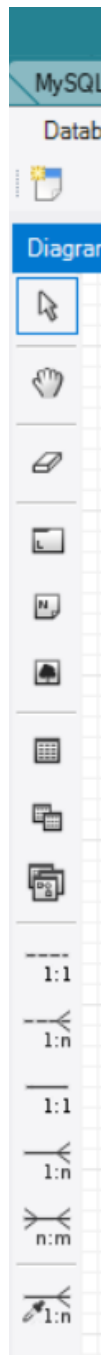
- timestamps**: create\_time, update\_time
- user**: username, email, password, create\_time
- category**: category\_id, name
- Dummy\_Table**: Dummy\_ID

The bottom status bar indicates the current table is 'Placed Dummy\_Table\_1'.

The screenshot displays the MySQL Modeler EER Diagram interface. The diagram shows three tables: 'Store', 'Period', and 'sales'. 'Store' has attributes: Store\_Code (PK), Store\_Name, City, Telephone, and Manager. 'Period' has attributes: Period\_Code (PK), Year, Quarter, Month, and Day. 'sales' has attributes: Product\_Code, Period\_Code (FK), Store\_Code (FK), and an implicit PK. Relationships are shown with dashed lines and crow's foot notation. The bottom panel shows the 'sales' table structure with columns: Dollars\_Cost and Dummy\_ID.

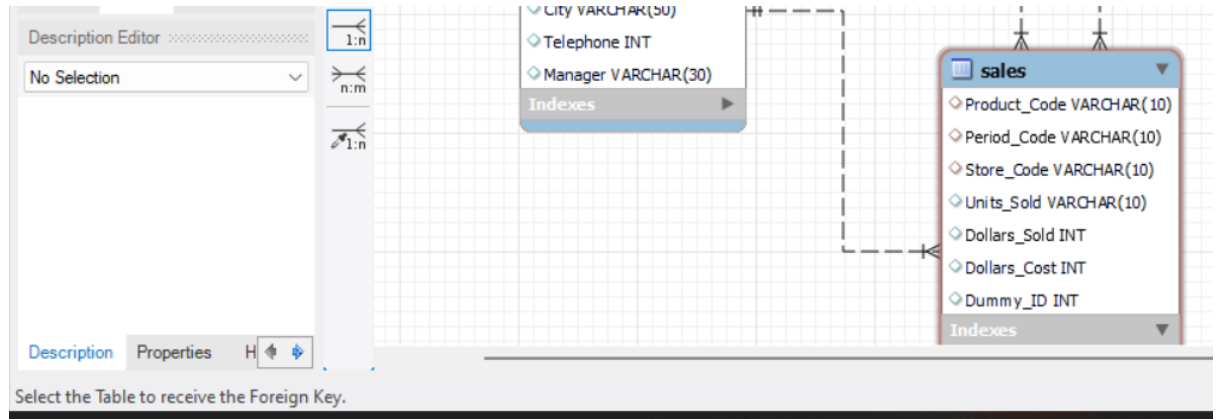
iv. Add a foreign key

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

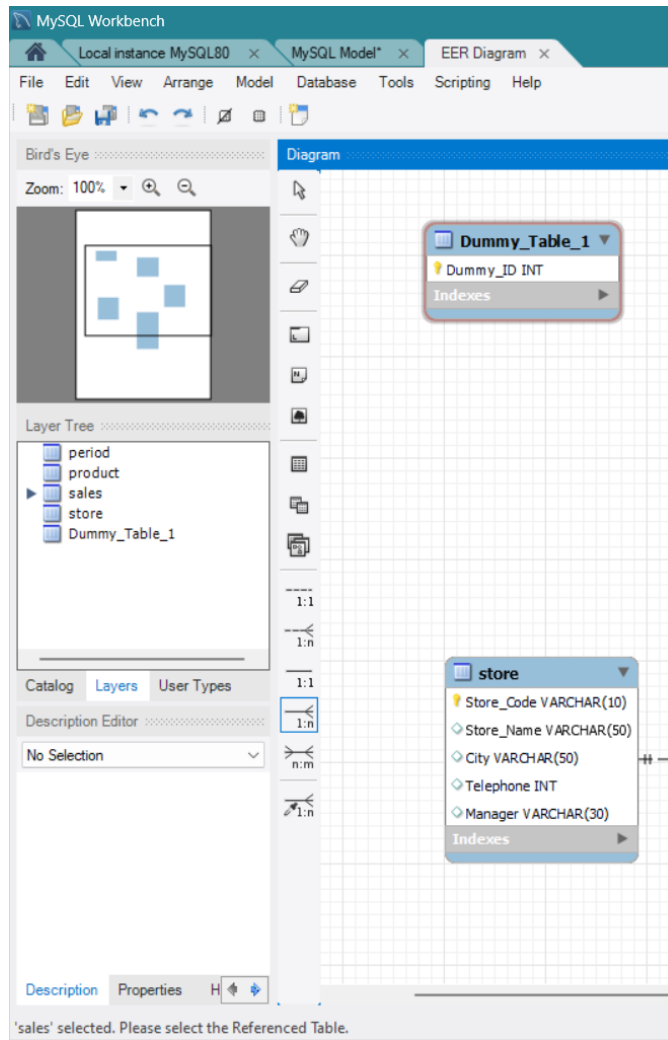




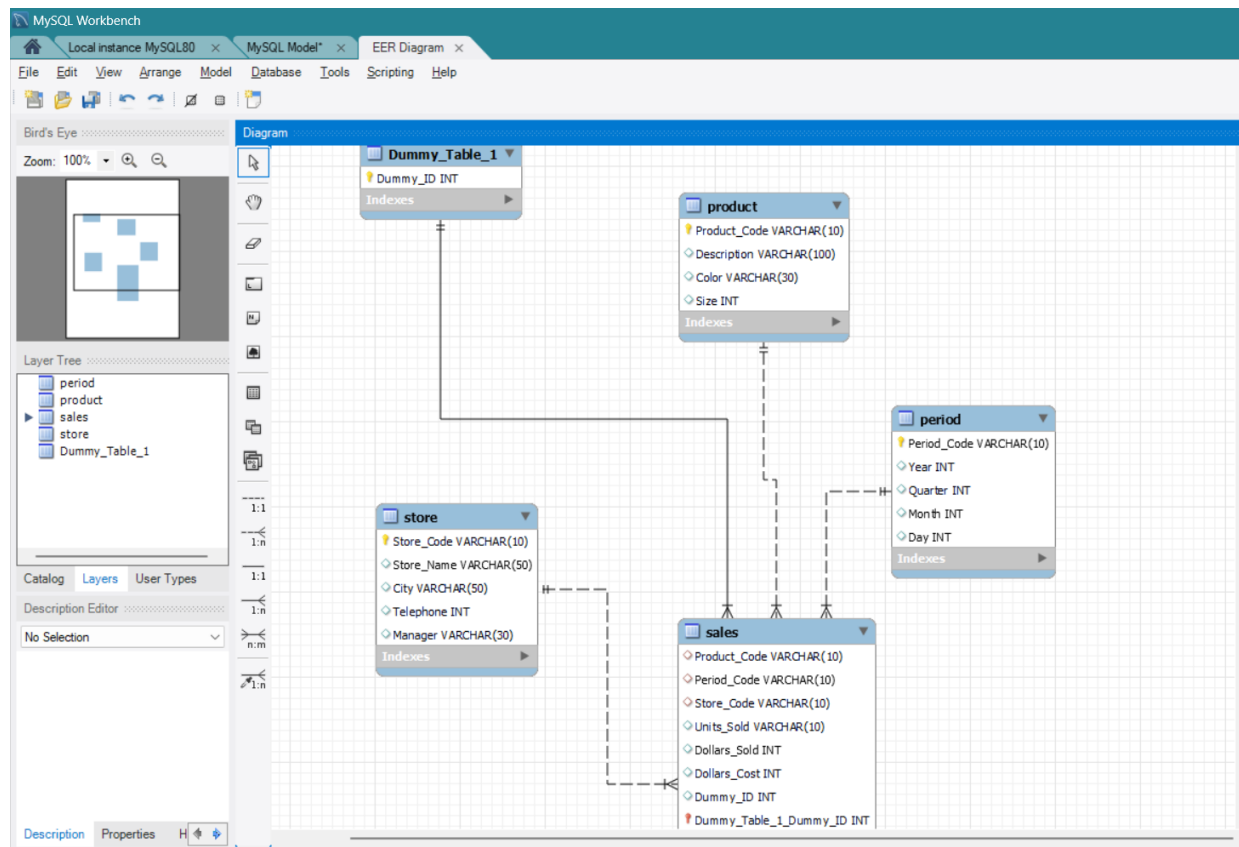
- II. It will prompted to select the table that receives foreign key. In this example Sales table selected



- III. Then select Referenced table. In this example Dummy\_Table selected.

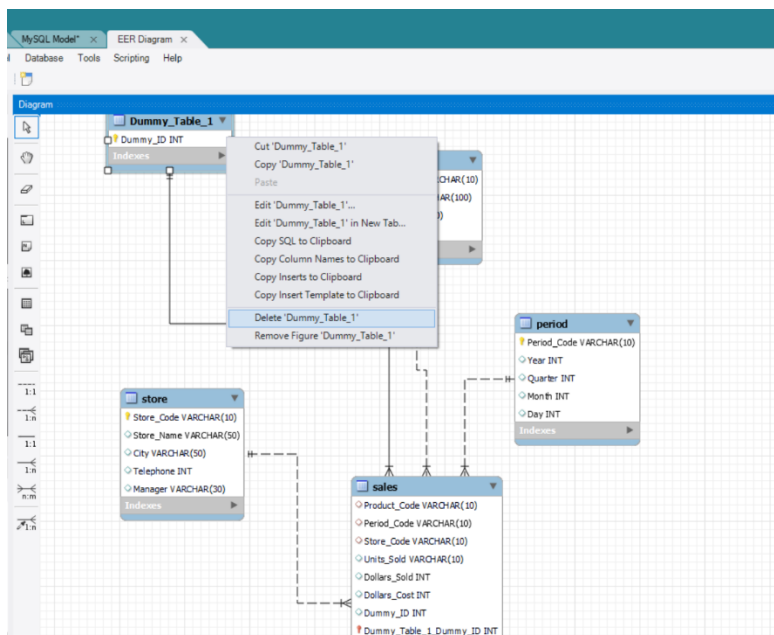


IV. Then the relationship is drawn , in this case between Dummy\_Table and Sales

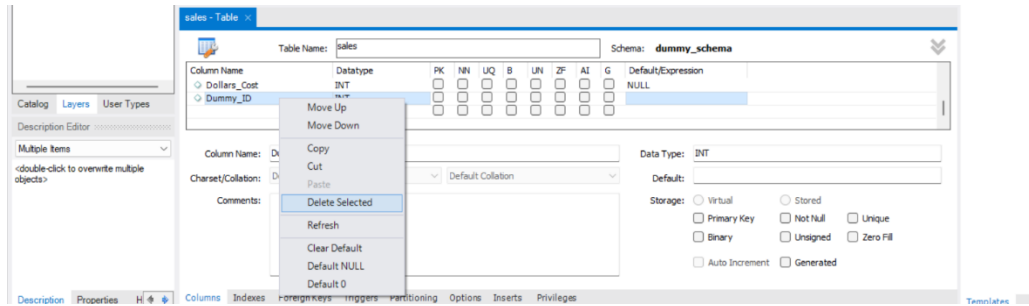


v. Delete a table or attribute

I. Right click the table and click option delete {Table\_Name}



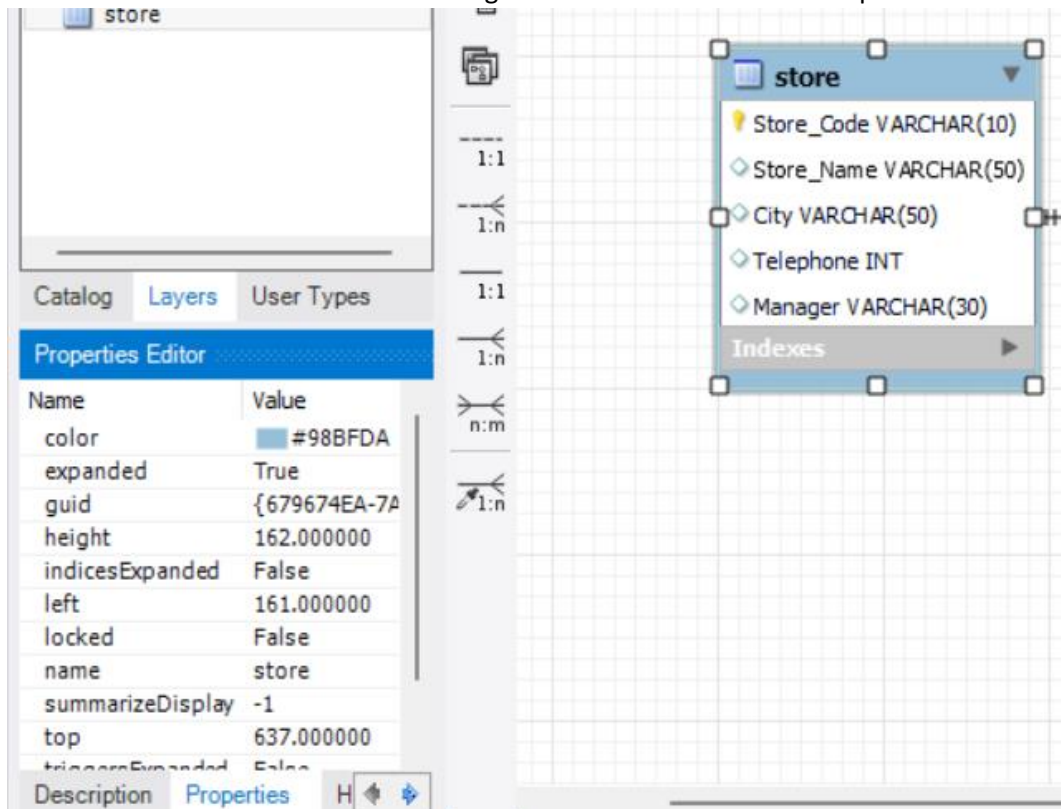
- II. To delete attribute just how attribute was added double click table / right click table and click edit , then just click the attribute row in the tab that opens at the bottom , right click and select delete selected.



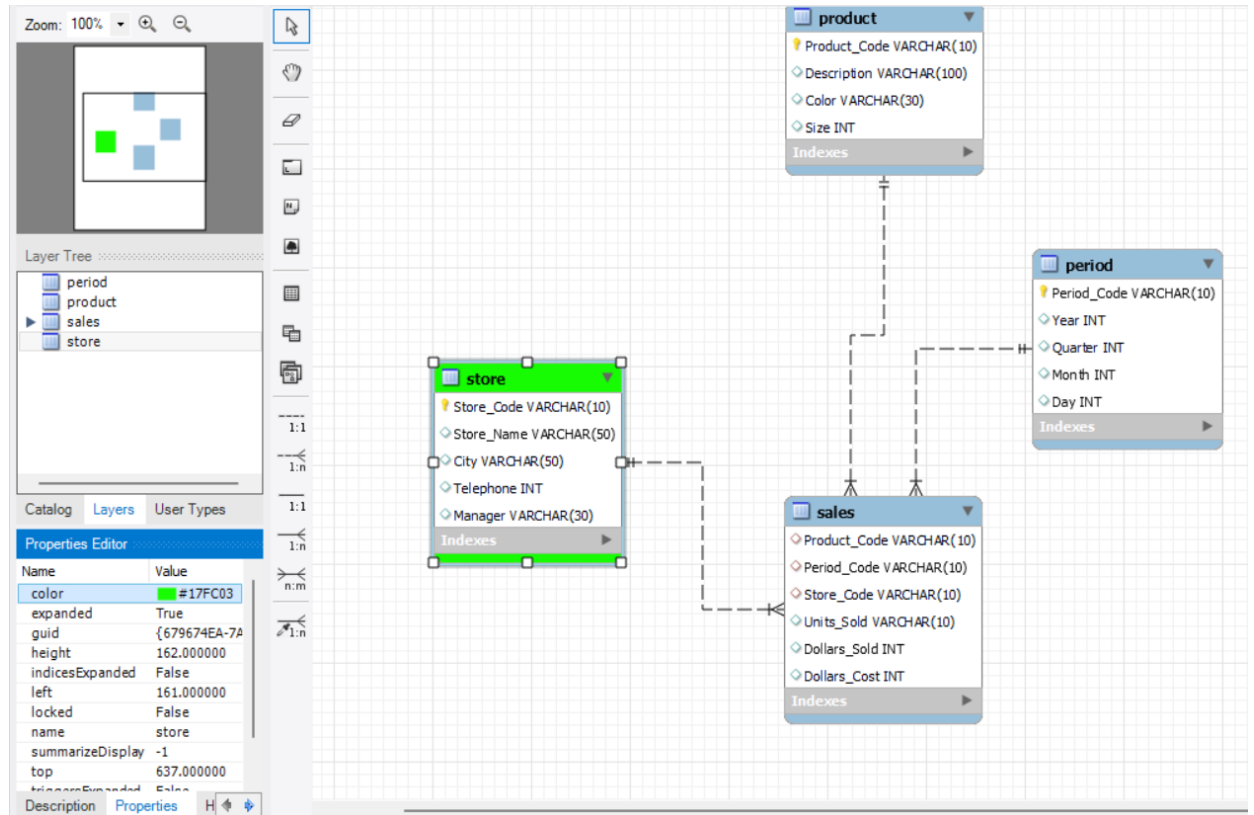
vi. Change the color of a table

Note : store table is considered at first for this example

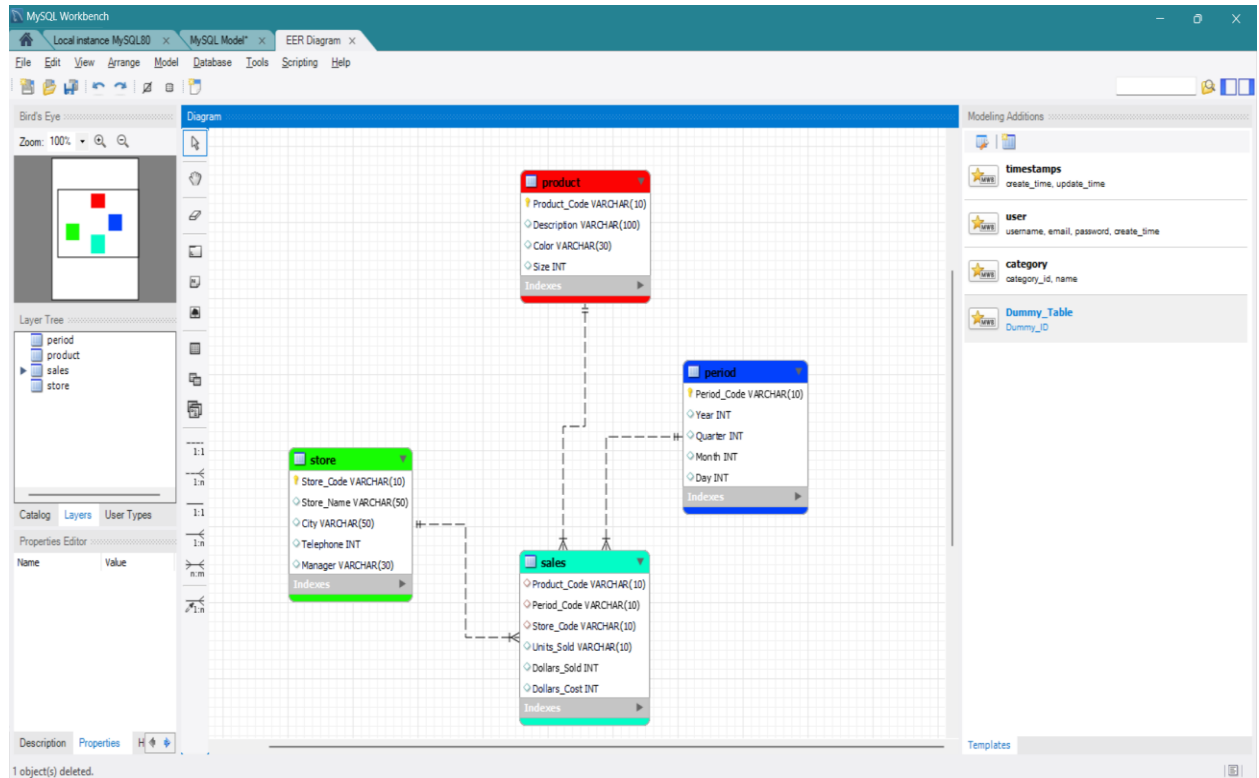
- I. Click the table whose color is to be changed and at bottom left click Properties



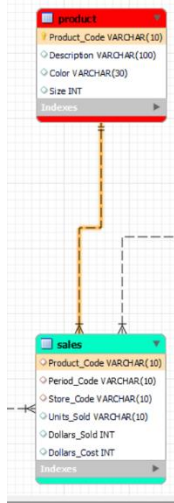
- II. Change the color column value to any hexadecimal color. Note that store table has a new color which is light green after changing the color value. Note the color is now #17fc03



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

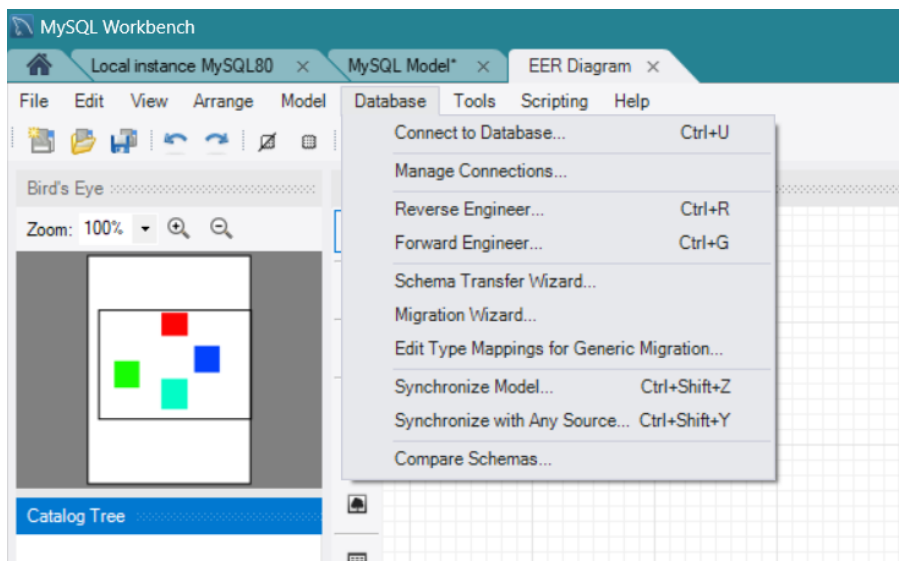


vii. Note when hovering over a relationship line , the color of relationship line and the attributes involved in the relationship change dynamically as seen in the below example.

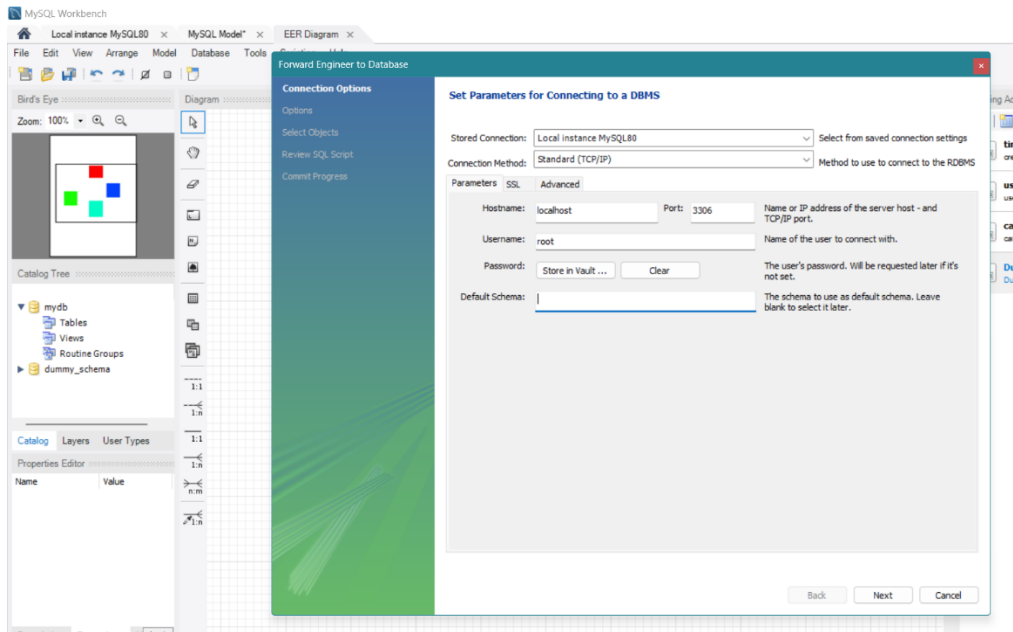


viii. Export the schema diagram i.e Forward Engineering

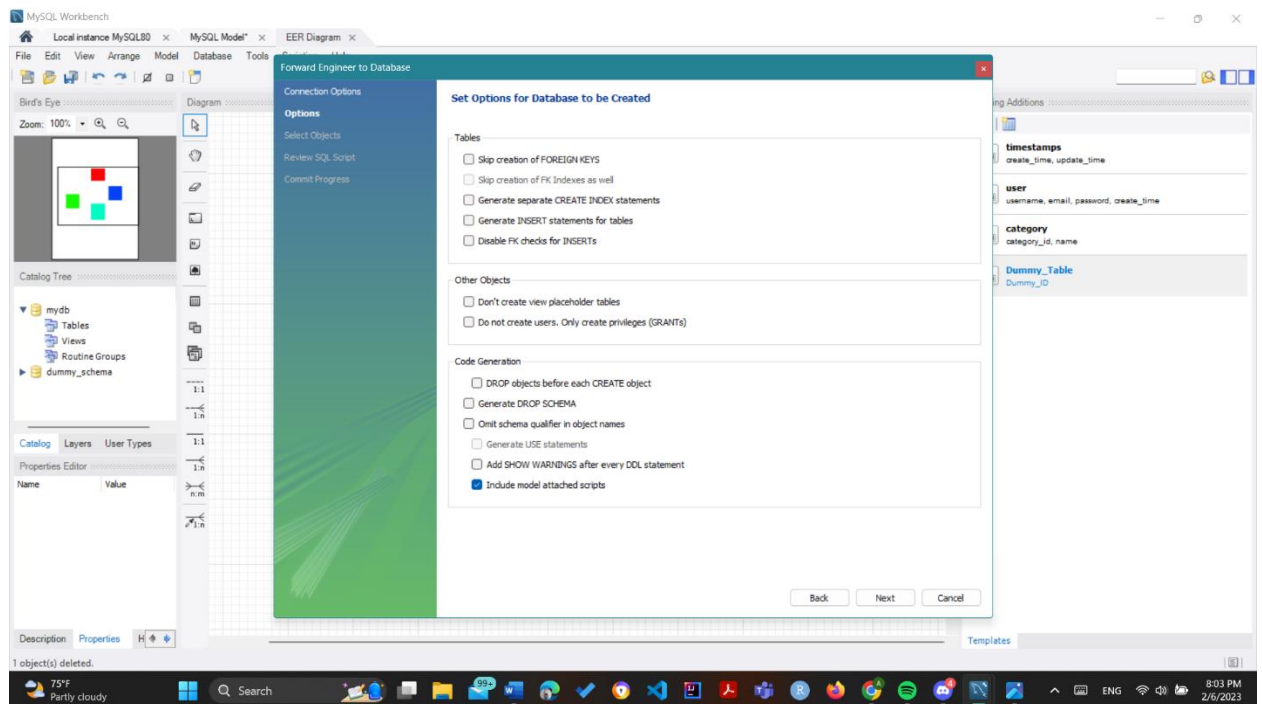
I. Click Database in top menu then Forward Engineering



II. Fill Required details

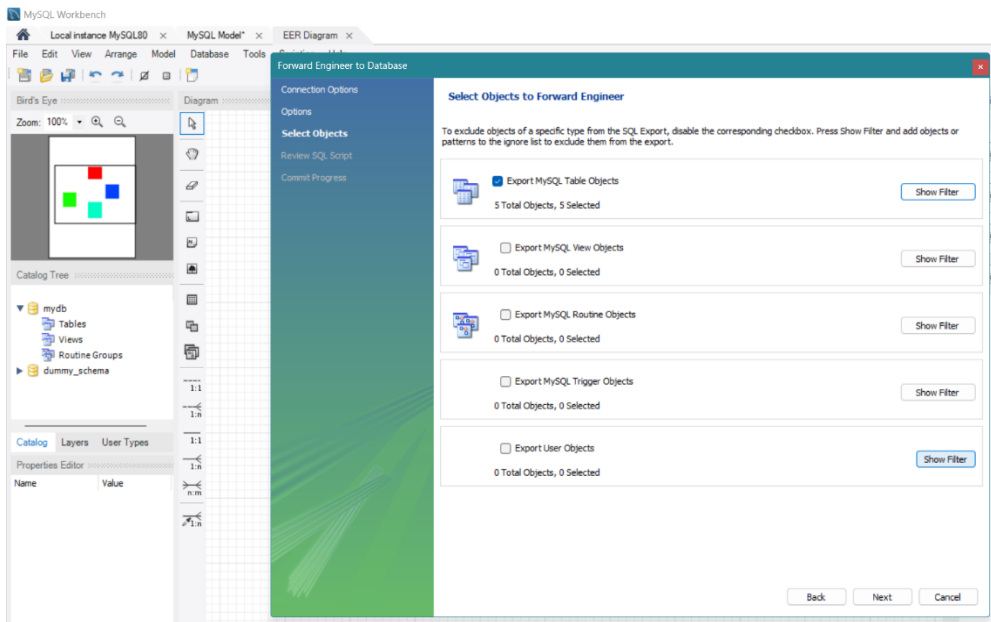


III. Select any options if needed

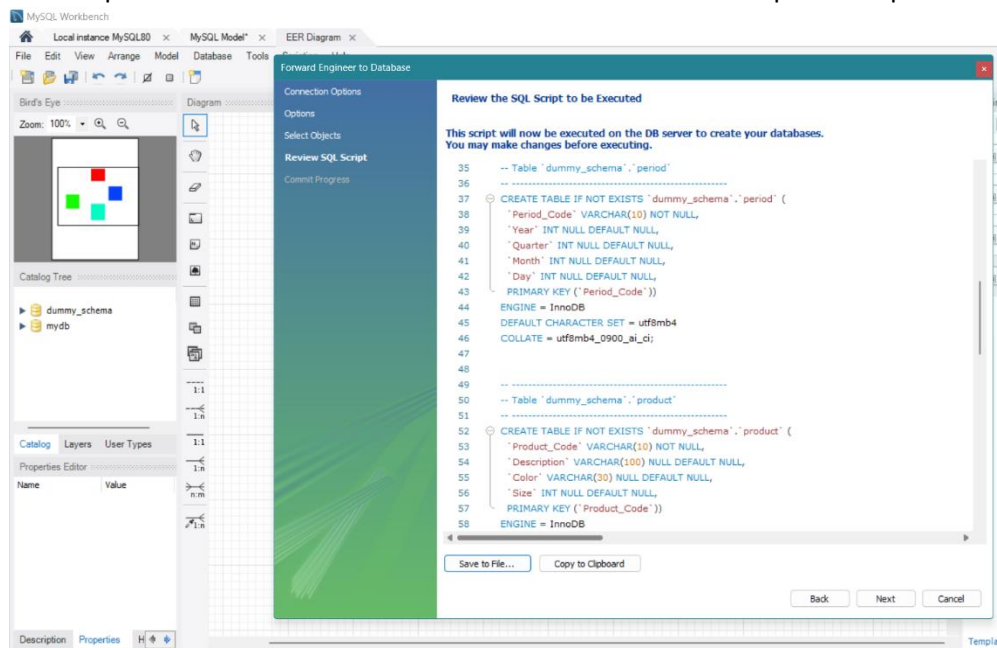




#### IV. Select any options if needed



#### 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

