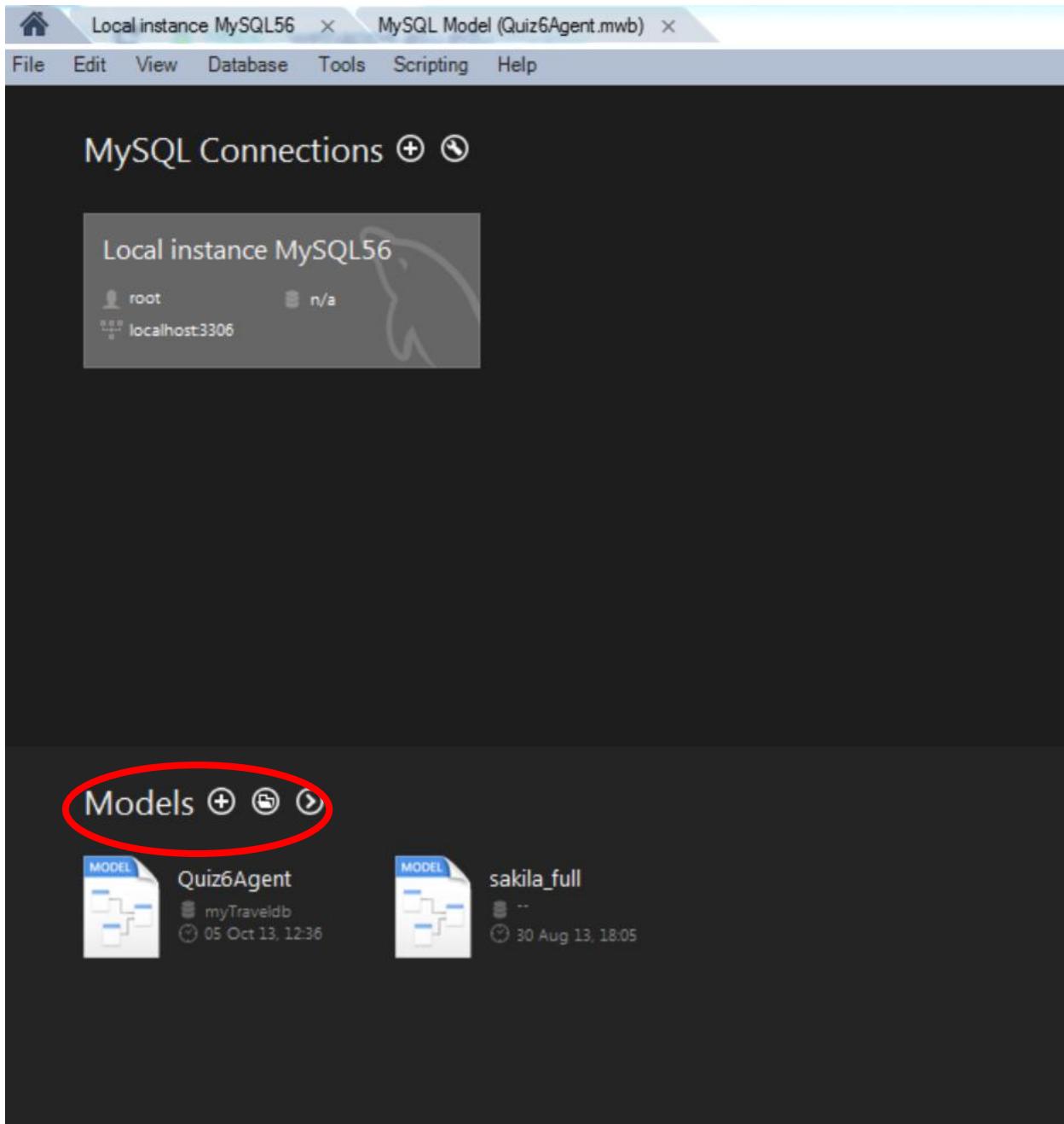
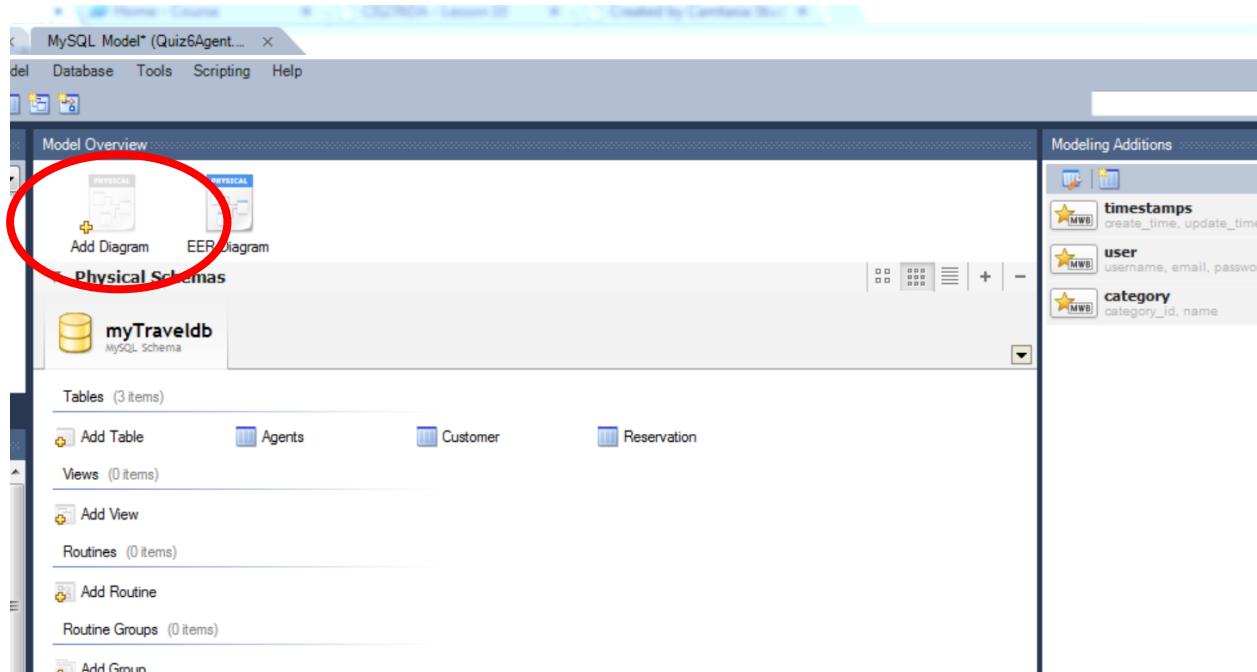


Create ER Diagram in MySQL for Sale-Fast Real Estate Agency

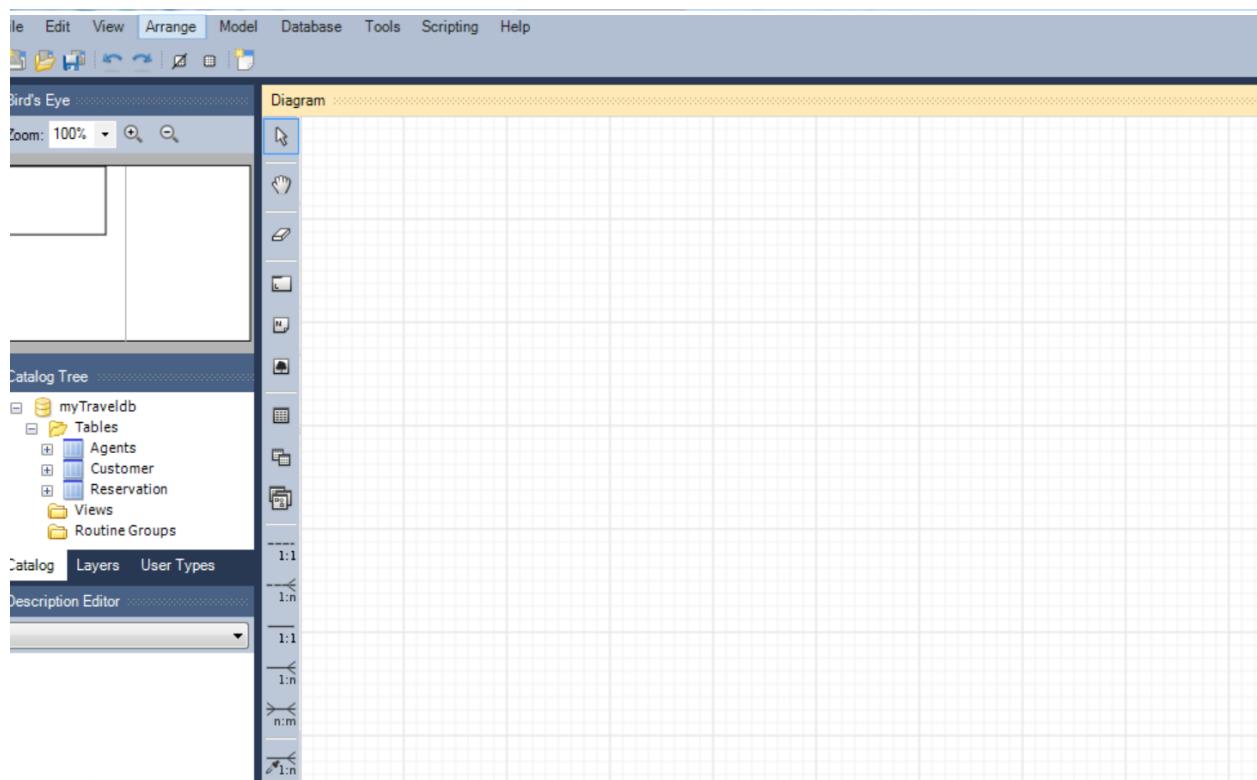
When you are in MySQL Workbench, click on the Models (left-bottom) of the workbench.



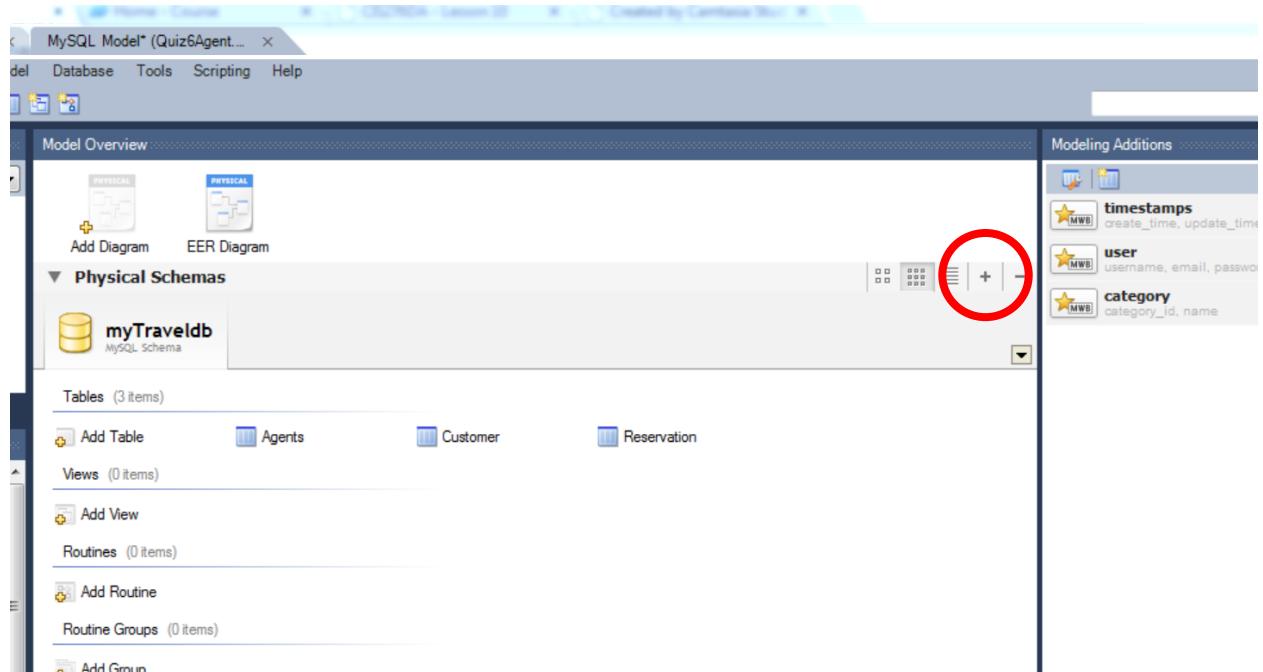
In the next screen, enter your password to login to MySQL Models screen. In the screen, click on Add Diagram.



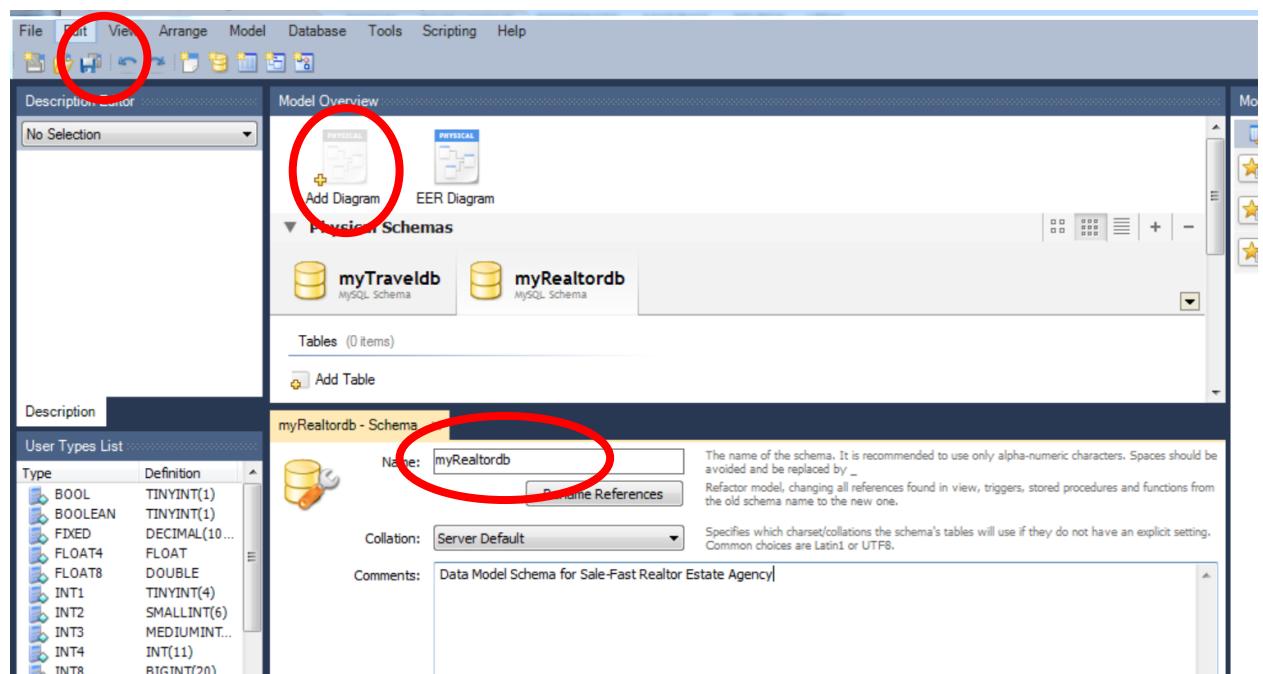
If this is your first ER diagram in MySQL, then start drawing the Diagram in the window as shown in below.



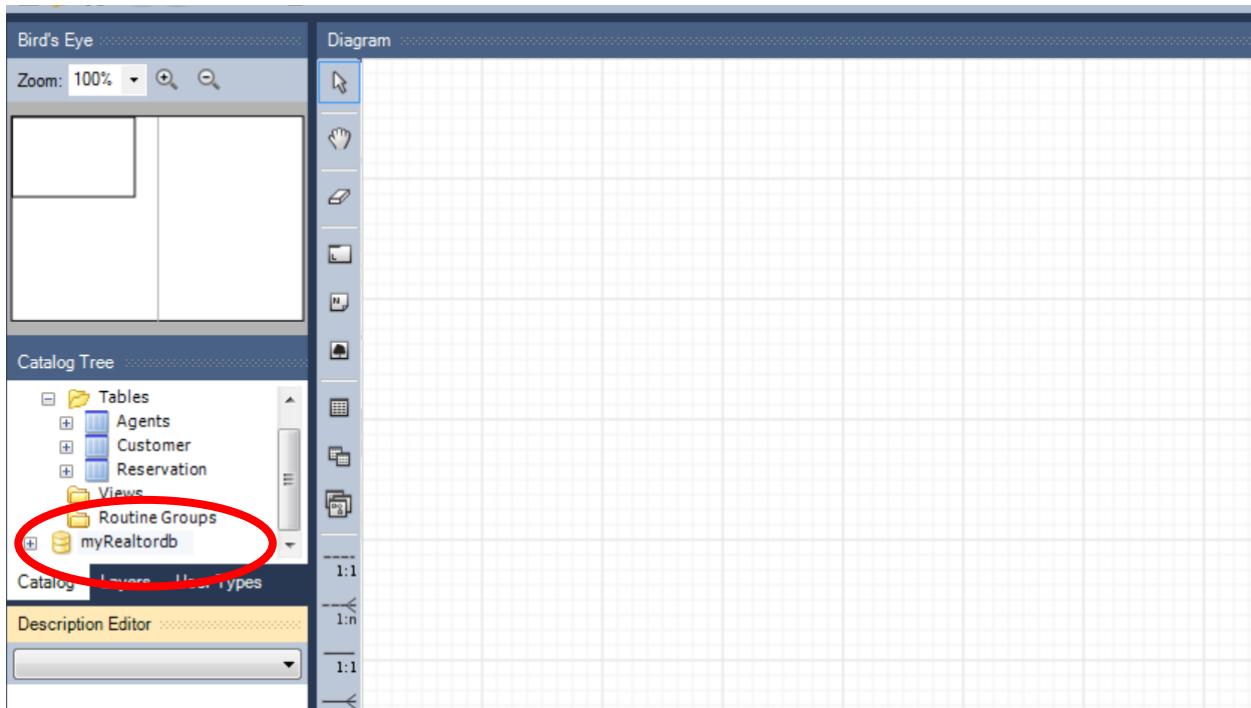
To create another ER diagram, click on the + sign as shown below to add another MySQL Schema.



Since I have already created myTraveldb data schema, I will click the + sign to create myRealtordb data schema. In the new_schema – Schema section, enter the name of the schema and Save it. Then, click Add Diagram to create the new ER diagram.

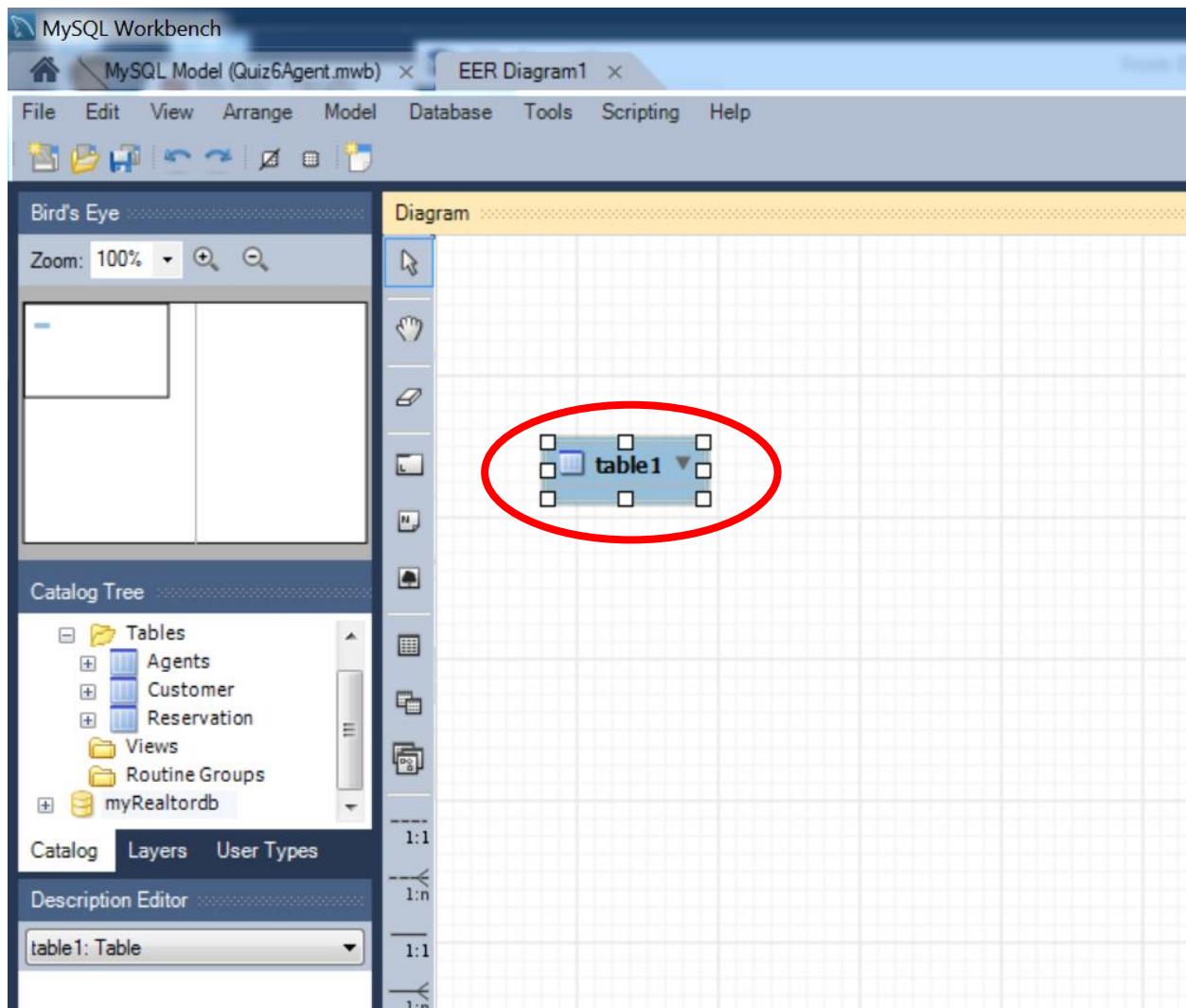


After click the Add Diagram option, you need to select the right Catalog in the Catalog Tree, as shown in the following diagram.



Now, let's start adding new tables to the ER diagram

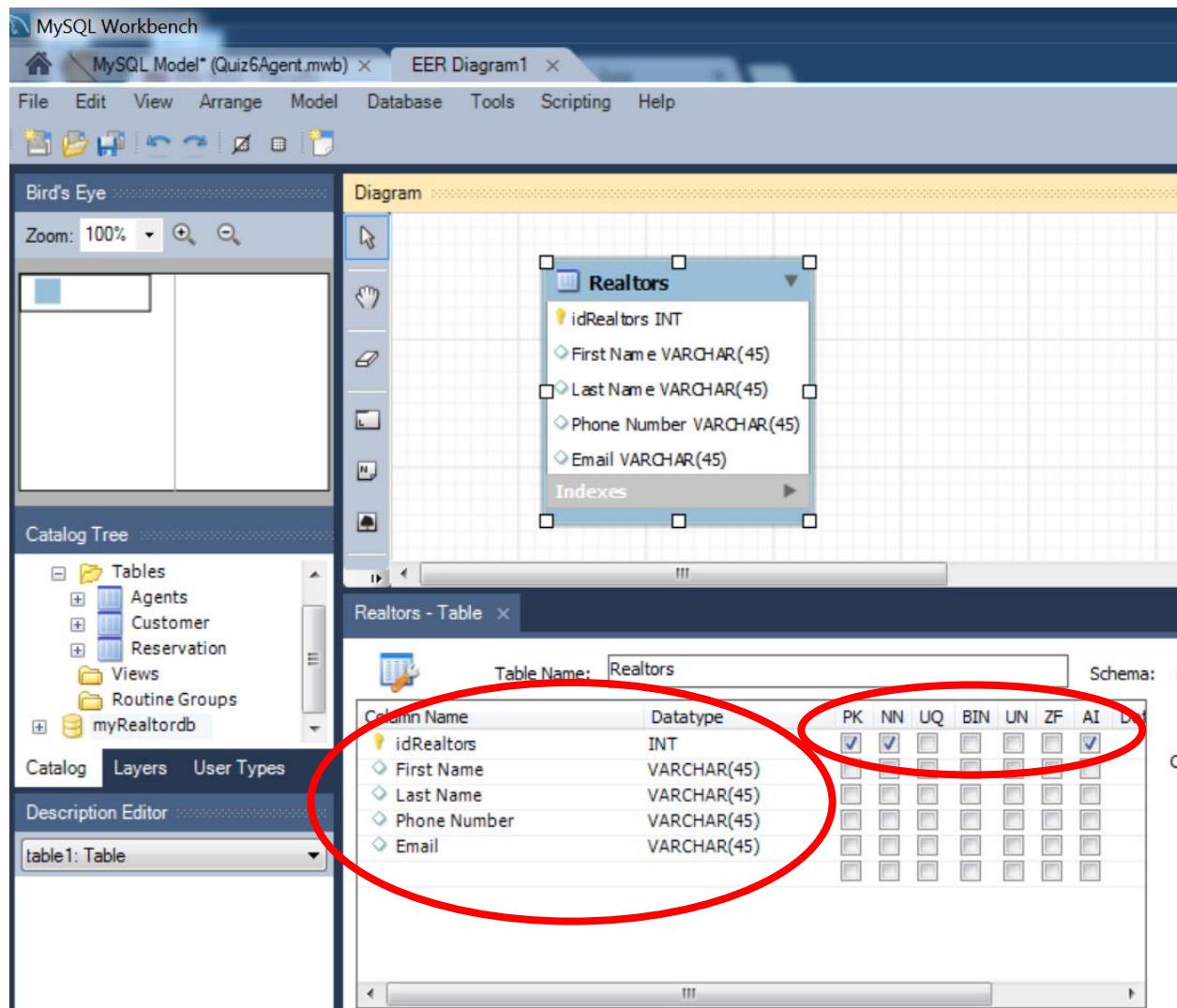
Double click the 7th icon from the top Pointer icon to create a new table and drag it to the gridded area. Then, a table symbol will be added to the gridded area. Then, double click on the table symbol.



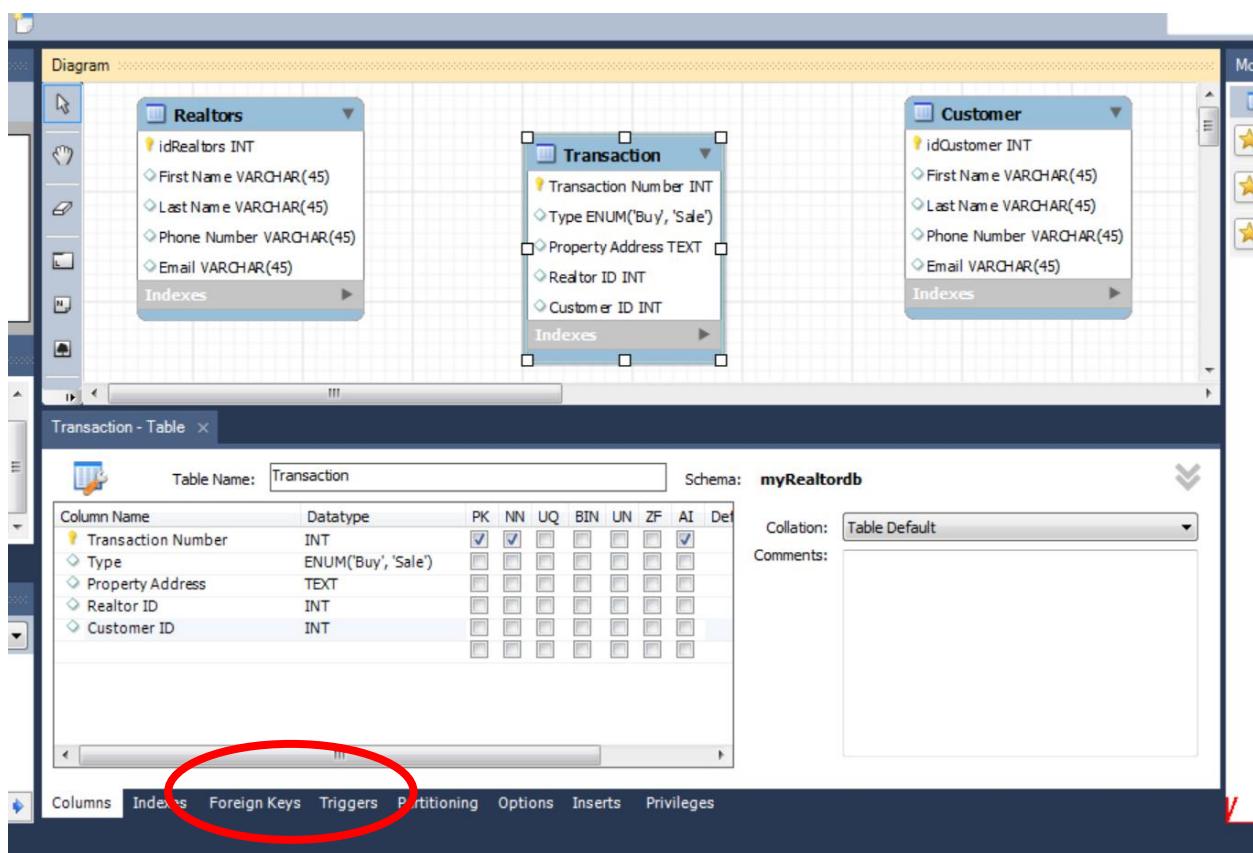
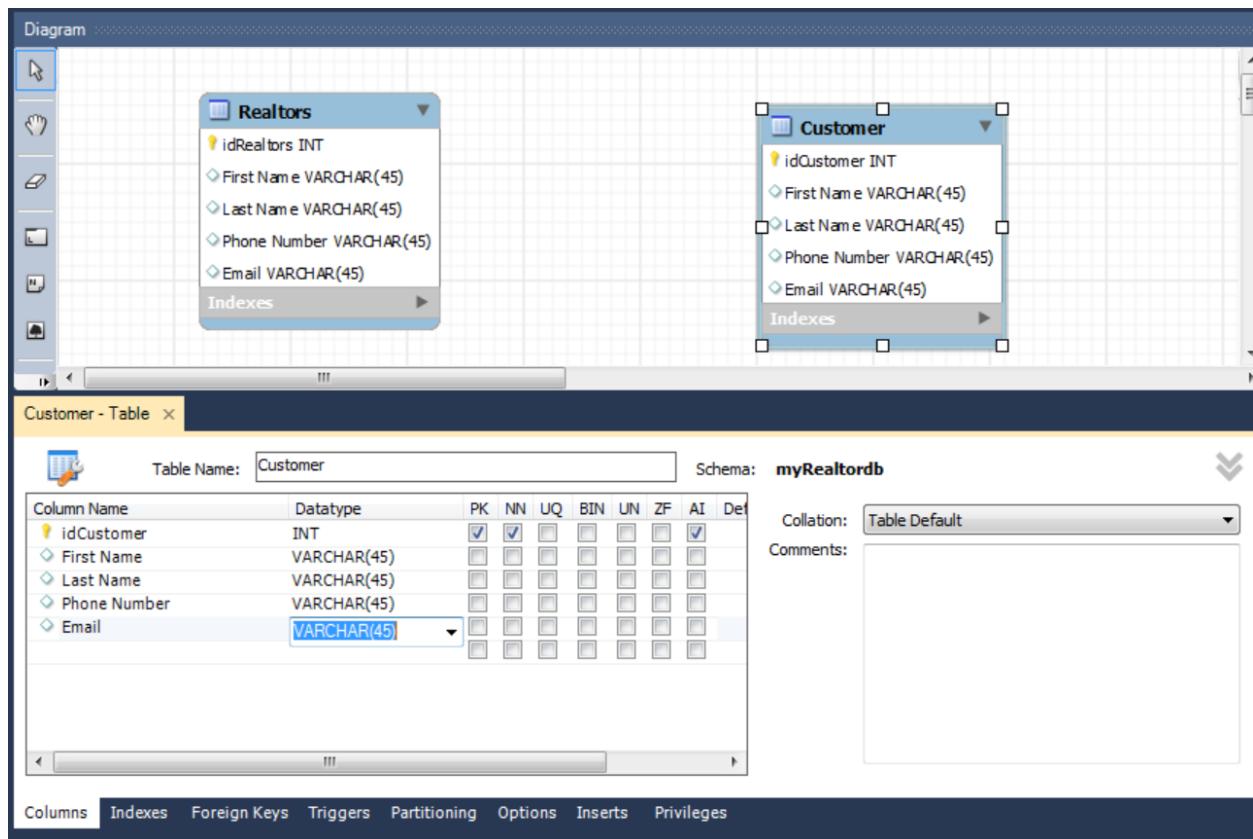
Enter Table name and attributes information in the bottom section of the screen.

Please NOTE: the Table name and attributes will be added into the Table Diagram by MySQL.

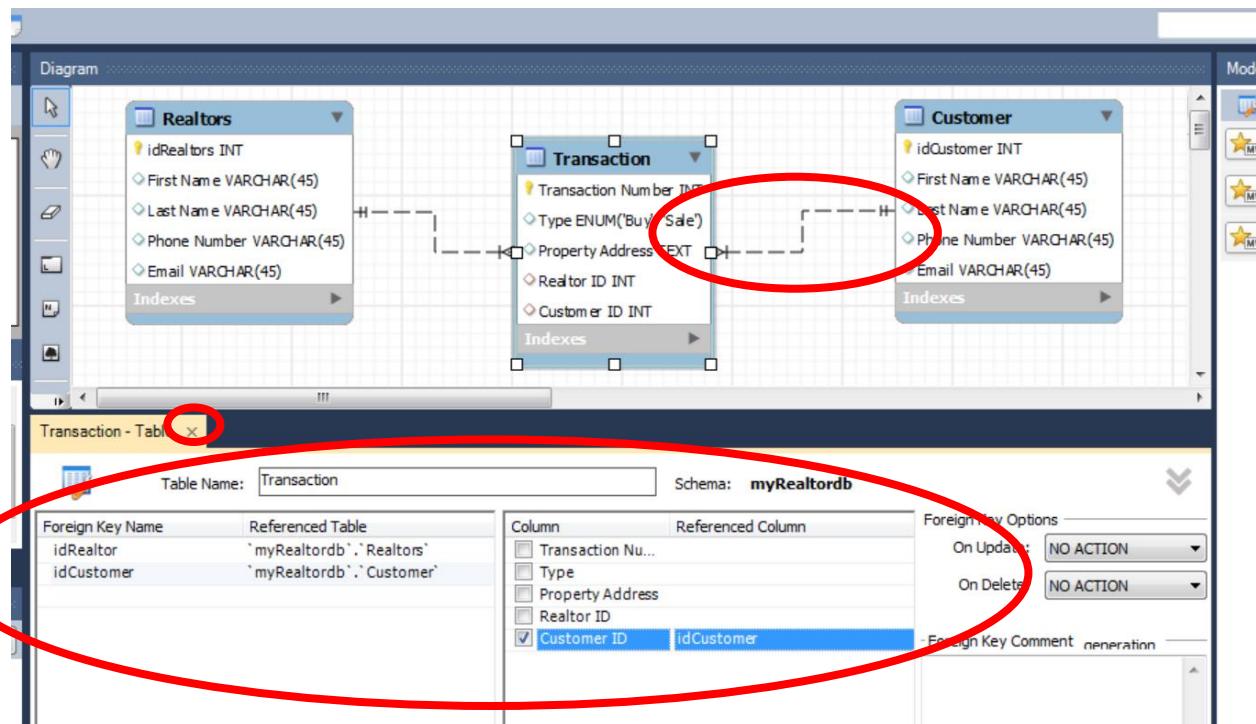
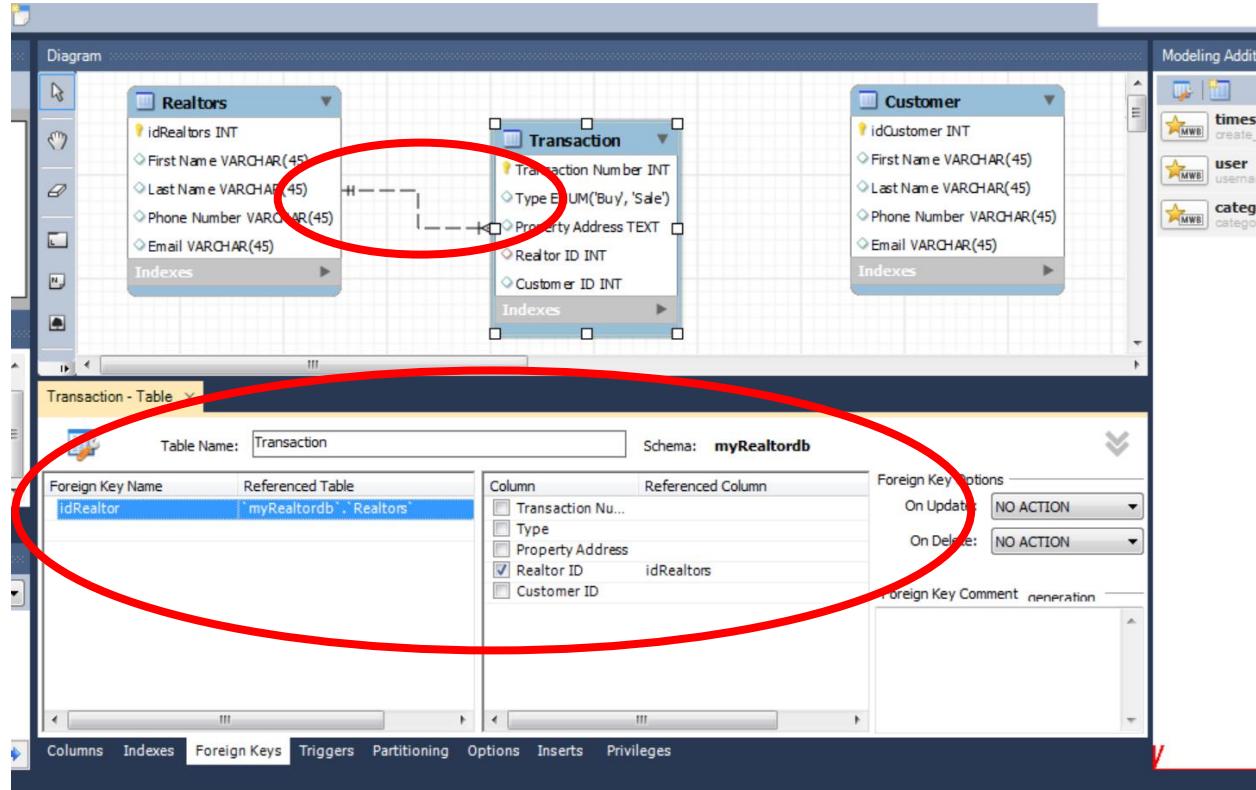
In the idRealtor row, check PK, NN (Not Null), and AI (Auto Increment) fields because it is the PK for the Realtor table.



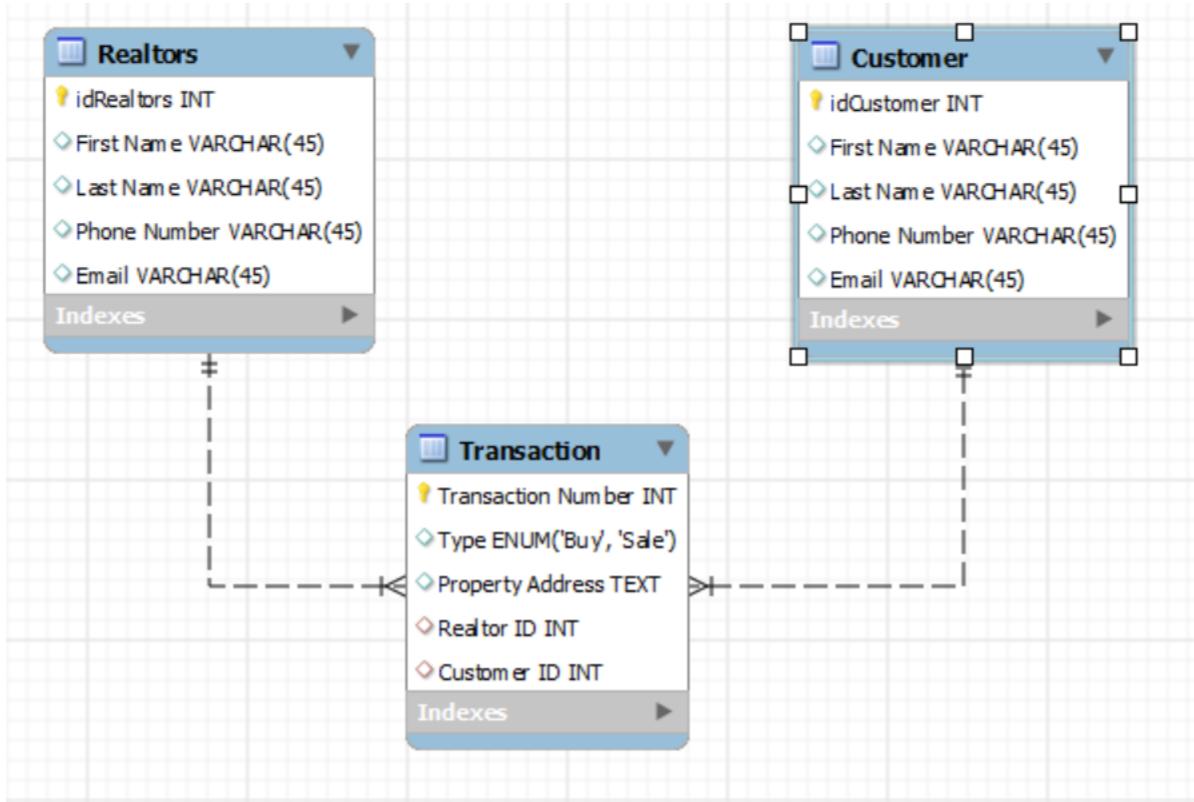
Click on the new table icon again to add other two Tables, Customer and Transaction.



Click on the Foreign Keys Option on the bottom to specify Realtor ID and Customer ID as the FKs for Transaction Table. The Relationship line will be generated by MySQL after the Foreign Key is identified.



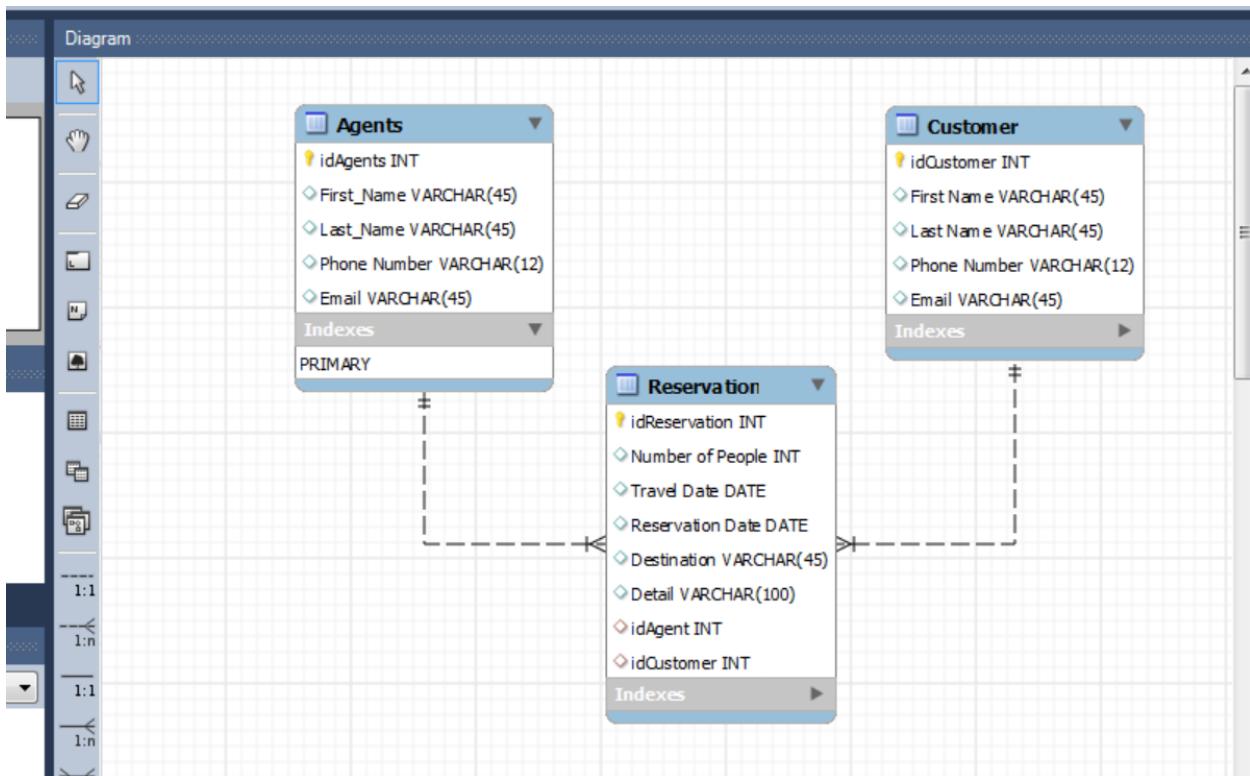
Close the Transaction – Table section by clicking on the x. Now you have created the ER diagram for the Sale-Fast Real Estate Agency.



Export ER diagram to SQL Script:

NOTE: The Quiz 6 case, Exploring Travel Agency, will be used as an example in this section.

ER diagram for Exploring Travel Agency:



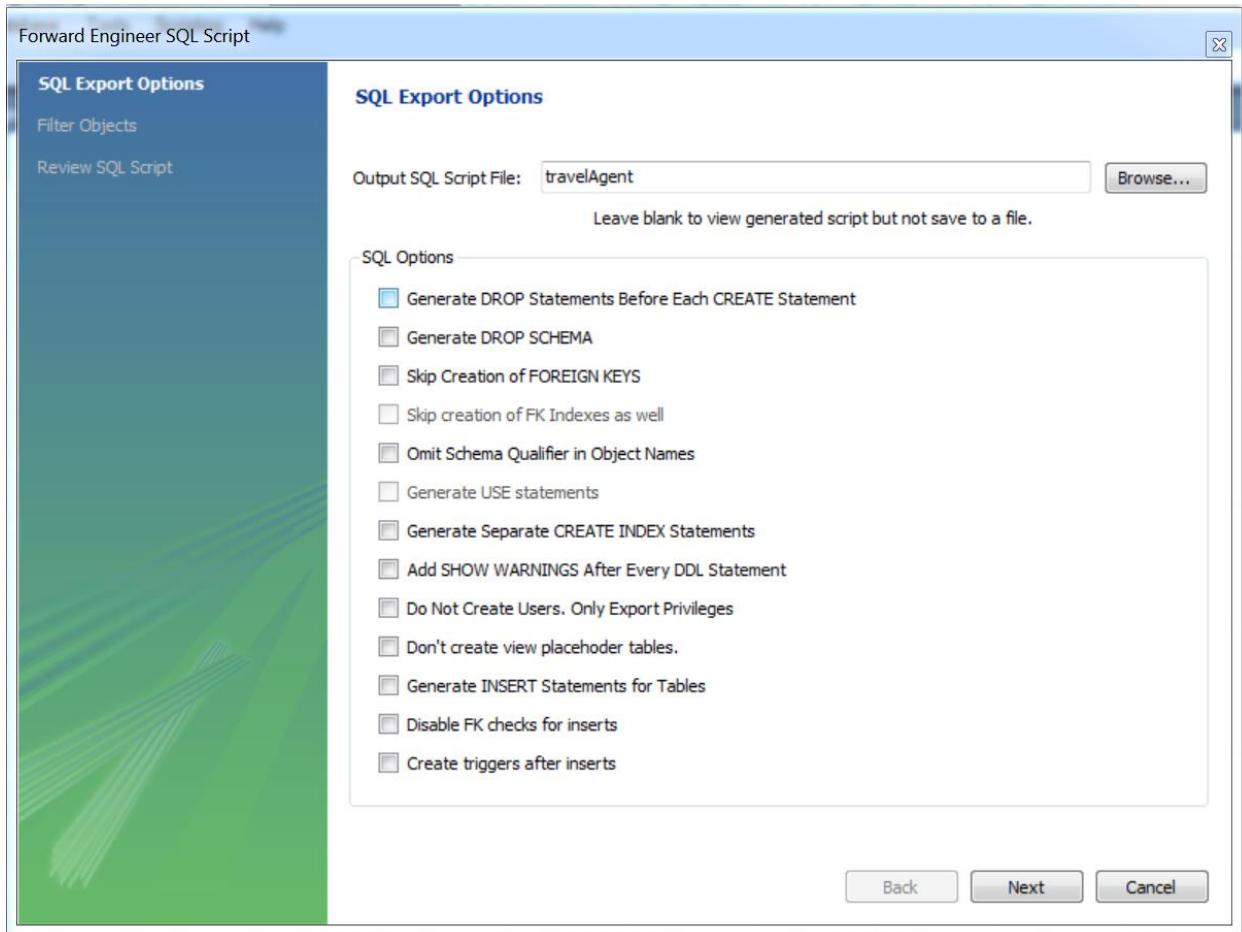
Export ER diagram to SQL Script:

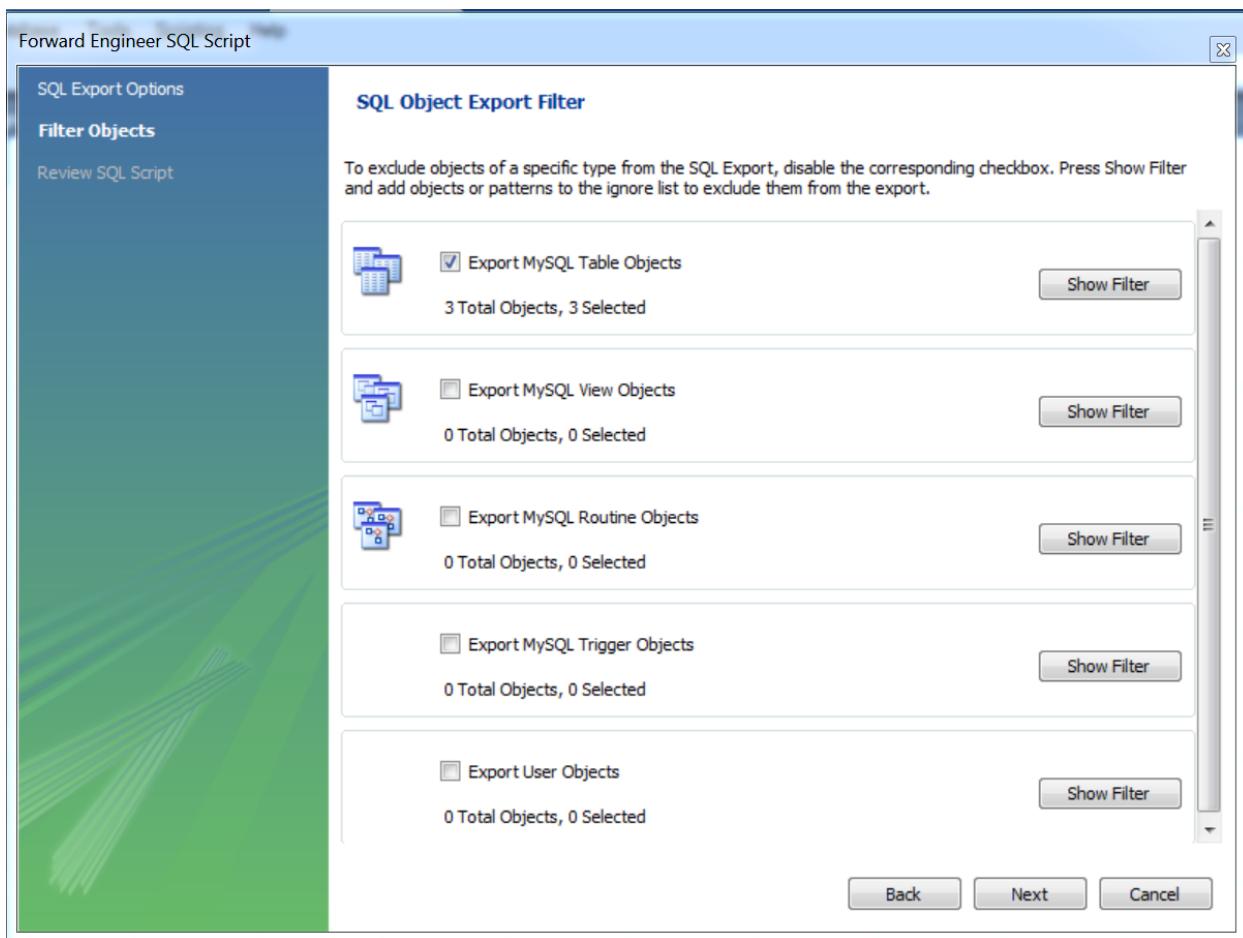
Go to the Menu Bar, select File, Export, Forward Engineer SQL Create Script, then the diagram below will be displayed on screen.

NOTE: If you select File, Export, Export as PNG or Export as a single page PDF, you can save the ER diagram to a PNG or a PDF file.

Give a script name in the Forward Engineer SQL Script Window. In this case, I entered “travelAgent” as the SQL Script file. Click on Browse button to save the script file to the file folder you like.

Forward Engineer SQL Script





It shows how many tables will be created in the exported SQL script.

Click Next and the SQL statements are displayed as shown in below:

Forward Engineer SQL Script

SQL Export Options
Filter Objects
Review SQL Script

Review Generated Script

Review and edit the generated script and press Finish to save.

```

1 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
2 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
3 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';
4
5 CREATE SCHEMA IF NOT EXISTS `myTraveldb` DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci ;
6 USE `myTraveldb` ;
7
8
9 -- Table `myTraveldb`.`Agents`
10
11 □ CREATE TABLE IF NOT EXISTS `myTraveldb`.`Agents` (
12     `idAgents` INT NOT NULL AUTO_INCREMENT,
13     `First_Name` VARCHAR(45) NULL,
14     `Last_Name` VARCHAR(45) NULL,
15     `Phone Number` VARCHAR(12) NULL,
16     `Email` VARCHAR(45) NULL,
17     PRIMARY KEY (`idAgents`)
18 ) ENGINE = InnoDB;
19
20
21 -- Table `myTraveldb`.`Customer`
22
23 □ CREATE TABLE IF NOT EXISTS `myTraveldb`.`Customer` (
24     `idCustomer` INT NOT NULL AUTO_INCREMENT,
25     `First Name` VARCHAR(45) NULL,
26     `Last Name` VARCHAR(45) NULL,
27     `Phone Number` VARCHAR(12) NULL,
28

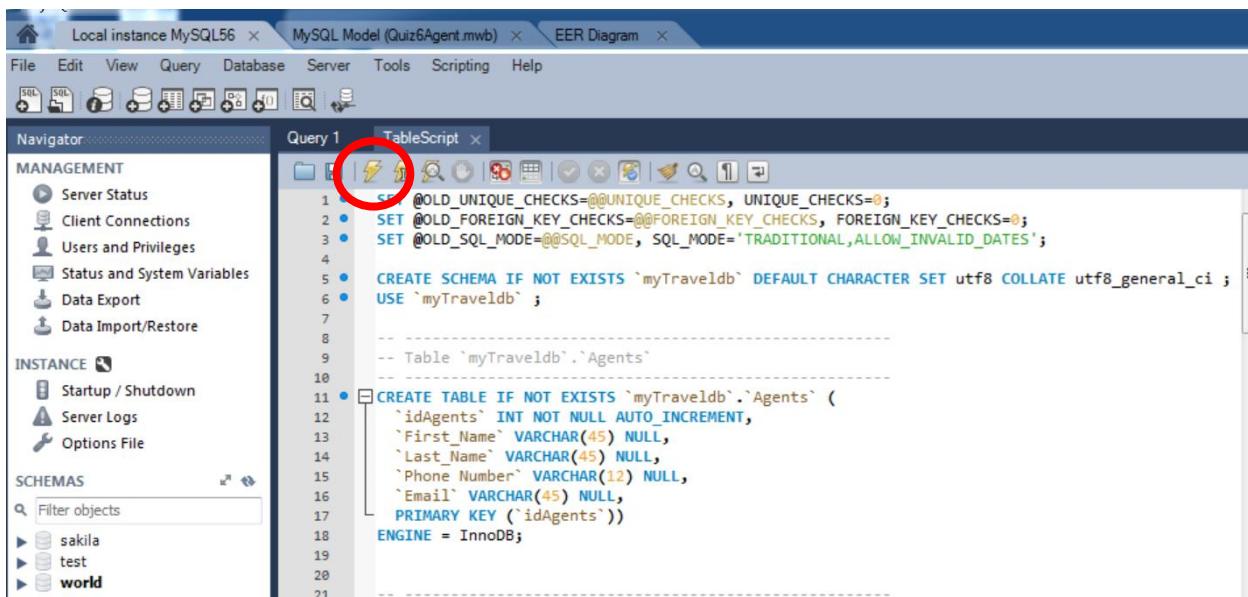
```

Save to Other File... Copy to Clipboard

Back Finish Cancel

Save the script by clicking “Finish” button.

Then in MySQL “Local instance MySQL56 screen, click File, then Open SQL Script, find the SQL statement you have just saved.



```

1 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
2 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
3 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';
4
5 CREATE SCHEMA IF NOT EXISTS `myTraveldb` DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci ;
6 USE `myTraveldb` ;
7
8
9 -- Table `myTraveldb`.`Agents`
10
11 □ CREATE TABLE IF NOT EXISTS `myTraveldb`.`Agents` (
12     `idAgents` INT NOT NULL AUTO_INCREMENT,
13     `First_Name` VARCHAR(45) NULL,
14     `Last_Name` VARCHAR(45) NULL,
15     `Phone Number` VARCHAR(12) NULL,
16     `Email` VARCHAR(45) NULL,
17     PRIMARY KEY (`idAgents`)
18 ) ENGINE = InnoDB;
19
20
21 -- Table `myTraveldb`.`Customer`
22
23 □ CREATE TABLE IF NOT EXISTS `myTraveldb`.`Customer` (
24     `idCustomer` INT NOT NULL AUTO_INCREMENT,
25     `First Name` VARCHAR(45) NULL,
26     `Last Name` VARCHAR(45) NULL,
27     `Phone Number` VARCHAR(12) NULL,
28

```

Click the Thunder run icon, as shown in the above figure.

In the Output window, it will show all the SQL statements are successfully executed.

The screenshot shows the MySQL Workbench interface. On the left, the Object Navigator displays the database structure for 'mytraveldb', including tables like 'agents', 'customer', and 'reservation'. Below this, the 'information' schema is selected. The main area is the 'Output' window titled 'Action Output', which lists 11 successful SQL statements from step 4 to step 11. Each statement is preceded by a green checkmark and shows the time, action, message, and duration/fetch information.

Time	Action	Message	Duration / Fetch
4 13:02:36	CREATE SCHEMA IF NOT EXISTS `myTraveldb` DEFAULT CHARACTER SET utf8 ...	1 row(s) affected	
5 13:02:36	USE `myTraveldb`	0 row(s) affected	
6 13:02:36	CREATE TABLE IF NOT EXISTS `myTraveldb`.`Agents` (`idAgents` INT NOT NULL ...	0 row(s) affected	
7 13:02:37	CREATE TABLE IF NOT EXISTS `myTraveldb`.`Customer` (`idCustomer` INT NOT N...	0 row(s) affected	
8 13:02:37	CREATE TABLE IF NOT EXISTS `myTraveldb`.`Reservation` (`idReservation` INT N...	0 row(s) affected	
9 13:02:38	SET SQL_MODE=@OLD_SQL_MODE	0 row(s) affected	
10 13:02:38	SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS	0 row(s) affected	
11 13:02:38	SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS	0 row(s) affected	

On the left-hand side of the Output window, right click on any of the database name, and select "Refresh All", you will see the mytraveldb has been created.

In the Query1 window, enter "Select * from agents". The result will be displayed in the Result Set Filter section. On the right-hand side, you will find SQL Additions where you can select the SQL statement that you would like to have more info, e.g. SELECT, and the syntax of the SELECT statement will be displayed.

See the Figure below:

The screenshot shows the MySQL Workbench interface with three main windows highlighted by red circles. The top window is 'Query 1' containing the SQL query 'Select * from agents'. The middle window is the 'Result Set Filter' showing the results of the query with columns: idAgents, First_Name, Last_Name, Phone Number, and Email. The bottom window is the 'Output' window showing the execution log. To the right of the Query 1 window is the 'SQL Additions' panel, which is expanded to show the 'Topic: SELECT' syntax. The syntax includes clauses like SELECT, FROM, WHERE, GROUP BY, and ORDER BY.

```
Topic: SELECT
Syntax:
SELECT
[ALL | DISTINCT | DISTINCTROW ]
[HIGH_PRIORITY]
[STRAIGHT_JOIN]
[SQL_SMALL_RESULT] [SQL_BIG_RESULT] [SQL_BUFFER_RESULT]
[SQL_CACHE | SQL_NO_CACHE] [SQL_CALC_FOUND_ROWS]
select_expr [, select_expr ...]
[FROM table_references
[PARTITION partition_list]
[WHERE where_condition]
[GROUP BY {col_name | expr | position}
[ASC | DESC], ... [WITH ROLLUP]]
[HAVING where_condition]
Context Help | Snippets
```

Enter the data in the Result Set Filter Section:

Query 1

```
1 • Select * from agents
```

Result Set Filter: Edit: Export/Import Wrap Cell Content:

	idAgents	First_Name	Last_Name	Phone Number	Email
1	Elva	Lin		4809652363	Elva.Lin@asu.edu
2	Yzel	Johnson		4809876543	Yzel.Johnson@asu.edu
*	NULL	NULL	NULL	NULL	NULL

agents 1* x Apply Cancel

Output

Action Output

Time	Action	Message
5 13:02:36	USE 'myTraveldb'	0 row(s) affected
6 13:02:36	CREATE TABLE IF NOT EXISTS 'myTraveldb'.'Agents' (`idAgents` INT NOT NULL)	0 row(s) affected
7 13:02:37	CREATE TABLE IF NOT EXISTS 'myTraveldb'.'Customer' (`idCustomer` INT NOT NULL)	0 row(s) affected
8 13:02:37	CREATE TABLE IF NOT EXISTS 'myTraveldb'.'Reservation' (`idReservation` INT NOT NULL)	0 row(s) affected

Then click on the “Insert New Row” icon, as shown in the above figure, to insert the two new rows.

Click on the Export option, on the right hand side of the “Insert New Row” icon, to export the new rows data into a .csv file, which can be opened with Excel.

first 2 rows.csv - Excel

Suh-Yun Lin

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	idAgents	First_Name	Last_Name	Phone Number	Email												
2	1	Elva	Lin	4809652363	Elva.Lin@asu.edu												
3	2	Yzel	Johnson	4809876543	Yzel.Johnson@asu.edu												
4																	
5																	
6																	
7																	
8																	
9																	
10																	
11																	
12																	
13																	
14																	
15																	
16																	
17																	
18																	
19																	
20																	
21																	
22																	
23																	
24																	

READY first 2 rows + 100%

Add more information to the file and then save it, in .csv format.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	idAgents	First_Name	Last_Name	Phone Number	Email											
2	1	Elva	Lin	4809652363	Elva.Lin@asu.edu											
3	2	Yzel	Johnson	4809876543	Yzel.Johnson@asu.edu											
4	3	Eric	Aleyasin Naraghi	480-965-1103	Eric.Aleyasin.Naraghi@asu.edu											
5	4	Rebeca	Brown	480-965-1104	Rebeca.Brown@asu.edu											
6	5	Yinghao	Castorena	480-965-1105	Yinghao.Castorena@asu.edu											
7	6	Matthew	Daulton	480-965-1106	Matthew.Daulton@asu.edu											
8	7	Marco	Foster	480-965-1107	Marco.Foster@asu.edu											
9	8	Juan	Gewargis	480-965-1108	Juan.Gewargis@asu.edu											
10	9	Eric	Guan	480-965-1109	Eric.Guan@asu.edu											
11	10	Ivette	Hakes	480-965-1110	Ivette.Hakes@asu.edu											
12																
13																
14																
15																
16																
17																
18																
19																
20																
21																
22																
23																
24																

Now you can import this file from MySQL to add the new rows to Agent table.

Click on the Import option as shown in the following diagram.

Query 1

```
1 • Select * from agents
```

Result Set Filter: Export/Import Wrap Cell Content:

	idAgents	First_Name	Last_Name	Phone Number	Email
1	1	Elva	Lin	4809652363	Elva.Lin@asu.edu
2	2	Yzel	Johnson	4809876543	Yzel.Johnson@asu.edu
*	NULL	NULL	NULL	NULL	NULL

agents 1*

Output

Action Output

Time	Action	Message
5 13:02:36	USE `myTraveldb`	0 row(s) affected
6 13:02:36	CREATE TABLE IF NOT EXISTS `myTraveldb`.`Agents` (`idAgents` INT NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=utf8	0 row(s) affected
7 13:02:37	CREATE TABLE IF NOT EXISTS `myTraveldb`.`Customer` (`idCustomer` INT NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=utf8	0 row(s) affected
8 13:02:37	CREATE TABLE IF NOT EXISTS `myTraveldb`.`Reservation` (`idReservation` INT NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=utf8	0 row(s) affected

Select the Excel file that you just created.

The screenshot shows the Oracle SQL Developer interface. A query window titled "Query 1" contains the SQL command "Select * from agents". The results pane displays the "agents" table with 10 rows of data. The first two rows, which are the header row and the first data row inserted, are highlighted with a red oval. In the toolbar above the results pane, the "Delete Selected Rows" icon (a trash can) is circled in red. At the bottom right of the results pane, the "Apply" button is also circled in red.

Before you click the Apply button, select the heading rows and the first 2 rows you first inserted (if they are still in the Result Set Filter window, as shown in the diagram below.

This screenshot is similar to the one above, showing the Oracle SQL Developer interface with the "agents" table results. The first two rows are again highlighted with a red oval. The "Delete Selected Rows" icon in the toolbar is circled in red. The "Apply" button at the bottom right is also circled in red.

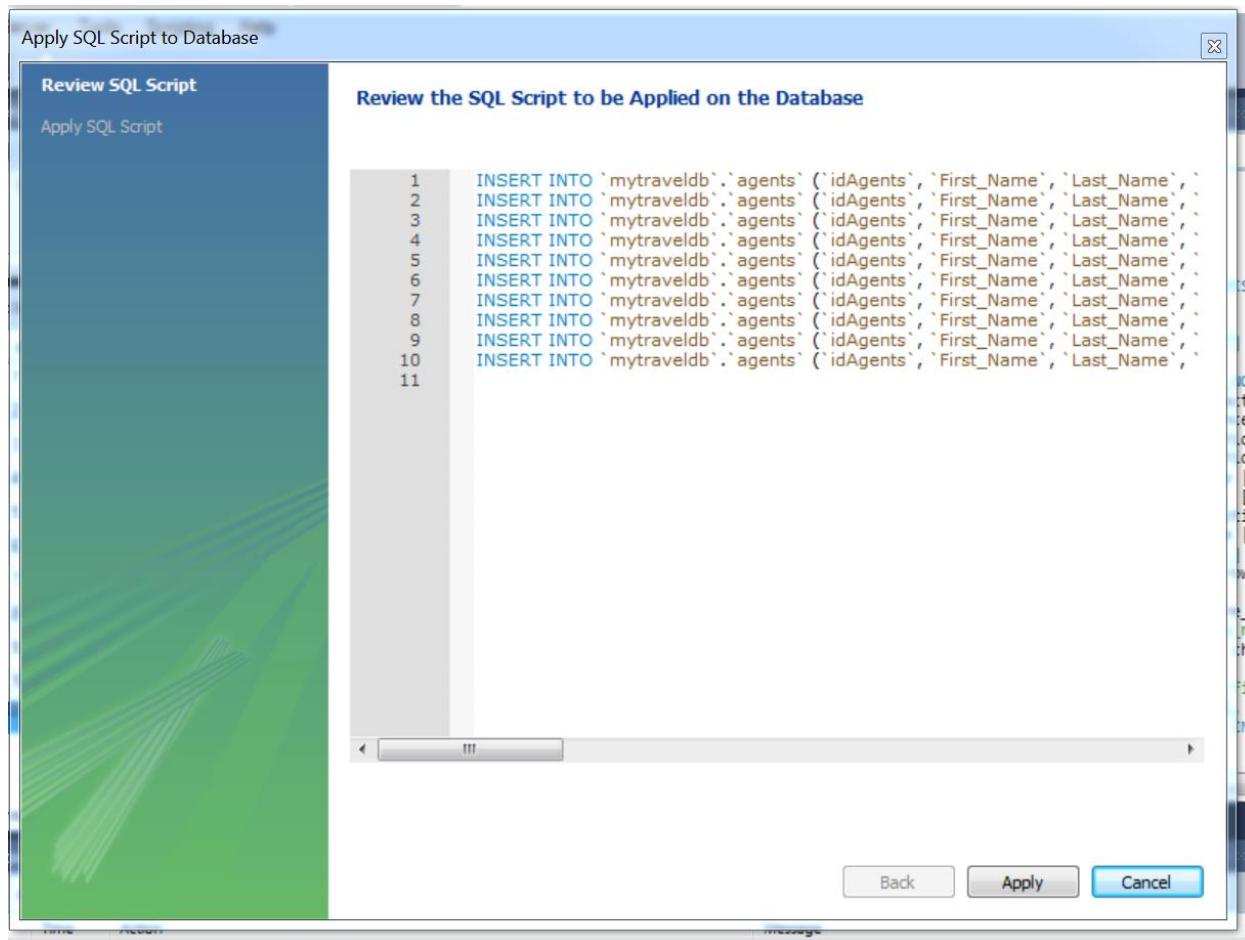
Then click on the Delete selected rows icon to delete those rows, as shown in the diagram above.

Then, click on the “Insert new rows” icon.

The screenshot shows the MySQL Workbench interface. A query window titled 'Query 1' contains the SQL command: 'Select * from agents'. The results pane displays a table named 'agents' with 10 rows of data. The table has columns: idAgents, First_Name, Last_Name, Phone Number, and Email. The last row (idAgents 10) is currently selected. Above the table, there is a toolbar with various icons. One icon, which is a blue square with a white plus sign and a red circle around it, represents the 'Insert new rows' function. This icon is highlighted with a red circle. Below the table, there is a modal dialog box titled 'agents 1*' with two buttons: 'Apply' (highlighted with a red circle) and 'Cancel'. To the right of the main interface, there is a 'Topic: S' sidebar with a 'Syntax' section containing the SELECT statement and a 'RESULT' section showing the table data.

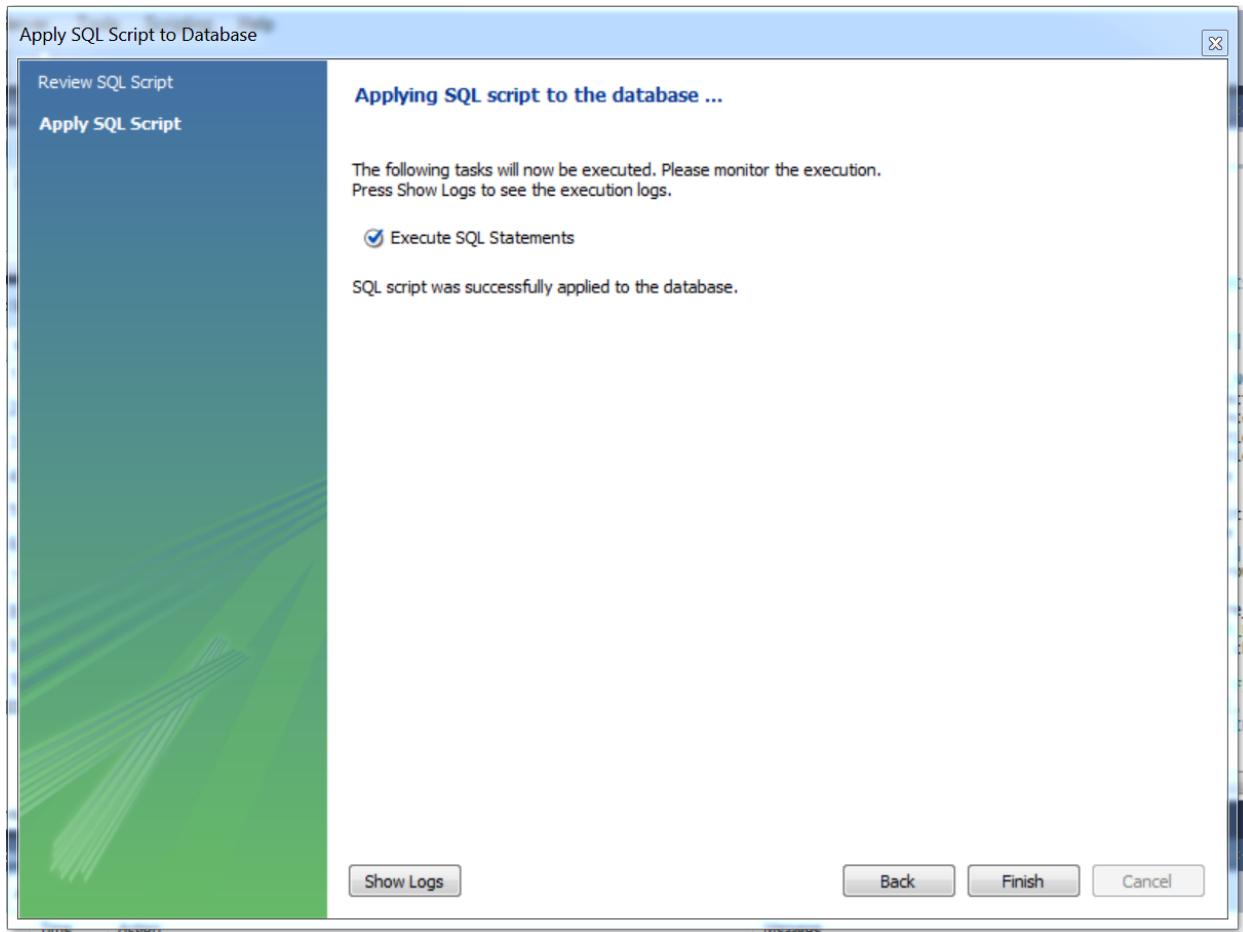
	idAgents	First_Name	Last_Name	Phone Number	Email
1	Elva	Lin		480-965-2363	Elva.Lin@asu.edu
2	Yzel	Johnson		480-987-6543	Yzel.Johnson@asu.edu
3	Eric	Aleyasin Naraghi		480-965-1103	Eric.Aleyasin Naraghi@asu.edu
4	Rebeca	Brown		480-965-1104	Rebeca.Brown@asu.edu
5	Yinghao	Castorena		480-965-1105	Yinghao.Castorena@asu.edu
6	Matthew	Daulton		480-965-1106	Matthew.Daulton@asu.edu
7	Marco	Foster		480-965-1107	Marco.Foster@asu.edu
8	Juan	Gewargis		480-965-1108	Juan.Gewargis@asu.edu
9	Eric	Guan		480-965-1109	Eric.Guan@asu.edu
10	Ivette	Hakes		480-965-1110	Ivette.Hakes@asu.edu
*	NULL	NULL	NULL	NULL	NULL

Then click on the Apply button. MySQL generates all the INSERT statements for these new rows, as shown in the diagram below.



Click Apply button.

The next screen will show those INSERT SQL statements are successfully executed.



Click "Finish" button.

Go to File and open a new query tab. In the new query tab, enter

"Select * from agents" to see the inserted data.

You can see the data is inserted.

Query 1 SQL File 2*

```
1 select * from agents
2
```

Result Set Filter: Edit Export/Import Wrap Cell Content

	idAgents	First_Name	Last_Name	Phone Number	Email
▶	1	Elva	Lin	480-965-2363	Elva.Lin@asu.edu
▶	2	Yzel	Johnson	480-987-6543	Yzel.Johnson@asu.edu
▶	3	Eric	Aleyasin Naraghi	480-965-1103	Eric.Aleyasin Naraghi@asu.edu
▶	4	Rebeca	Brown	480-965-1104	Rebeca.Brown@asu.edu
▶	5	Yinghao	Castorena	480-965-1105	Yinghao.Castorena@asu.edu
▶	6	Matthew	Daulton	480-965-1106	Matthew.Daulton@asu.edu
▶	7	Marco	Foster	480-965-1107	Marco.Foster@asu.edu
▶	8	Juan	Gewargis	480-965-1108	Juan.Gewargis@asu.edu
▶	9	Eric	Guan	480-965-1109	Eric.Guan@asu.edu
▶	10	Ivette	Hakes	480-965-1110	Ivette.Hakes@asu.edu
*	HULL	HULL	HULL	HULL	HULL

agents 2 × Apply Cancel

Output Message

Action Output Time Action