



**JUST IT, London**

**DATA TECHNICIAN BOOT CAMP**

# **PROJECT 2**

## **MySQL Intro and Assignment**

- Installation
- Workbench
- Schemas
- Tables
- Basic manipulations
- Basic Queries
- Creating an EER Diagram

**STUDENT: Alexei Kosyhin**

**Date: 20 Feb 2022**

<b>Content</b>	<b>Pages</b>
- General information about MySQL .....	3
- Installation .....	3
- Starting MySQL .....	6
- Layout: Schemas .....	7
- Basic queries.....	8
- EEL Model .....	12

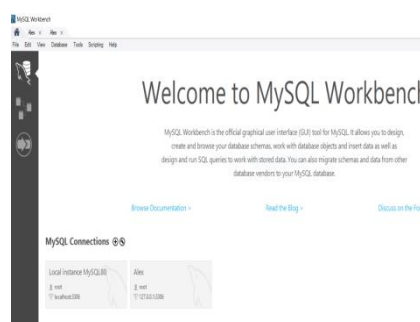
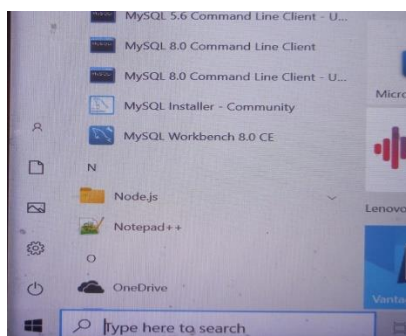
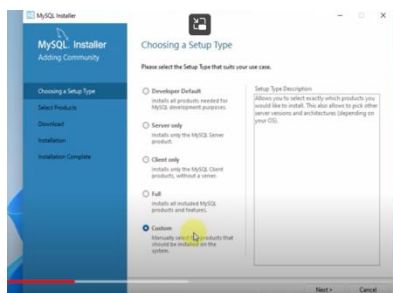
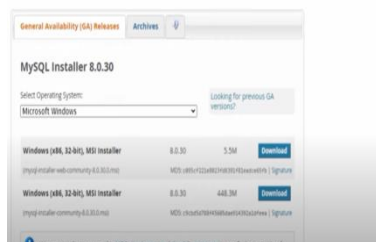
**From Wikipedia.** MySQL (/ˌmaɪˌɛsˌkjuːˈɛl/)[5] is an open-source relational database management system (RDBMS).[5][6] Its name is a combination of "My", the name of co-founder Michael Widenius's daughter My,[7] and "SQL", the acronym for Structured Query Language. A relational database organizes data into one or more data tables in which data may be related to each other; these relations help structure the data. SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.

## Installation

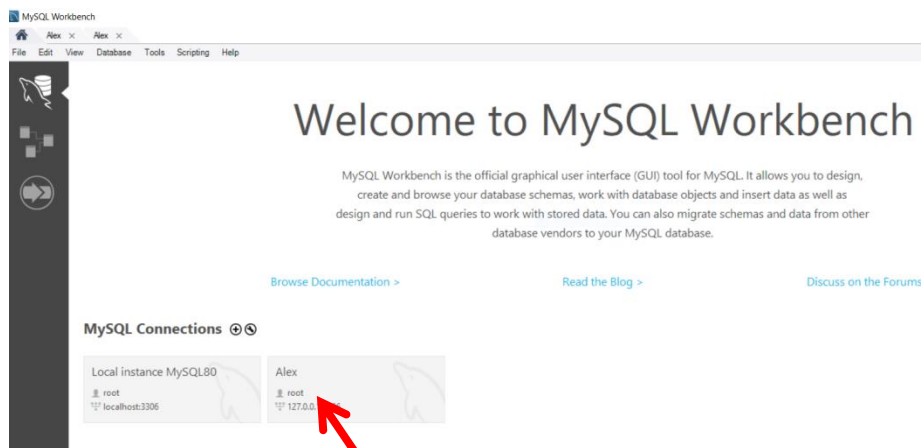
I followed the link [MySQL :: Download MySQL Installer](#)

I watched youtube tutorial [How to install MySQL 8.0.30 Server and Workbench latest version on Windows 10 - YouTube](#)

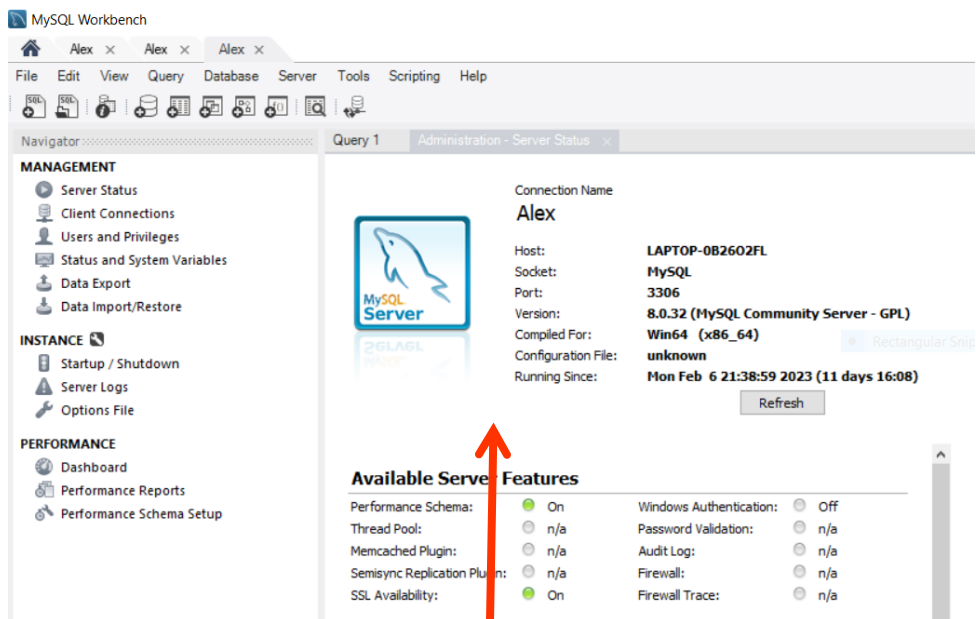
MySQL Community Downloads  
MySQL Installer



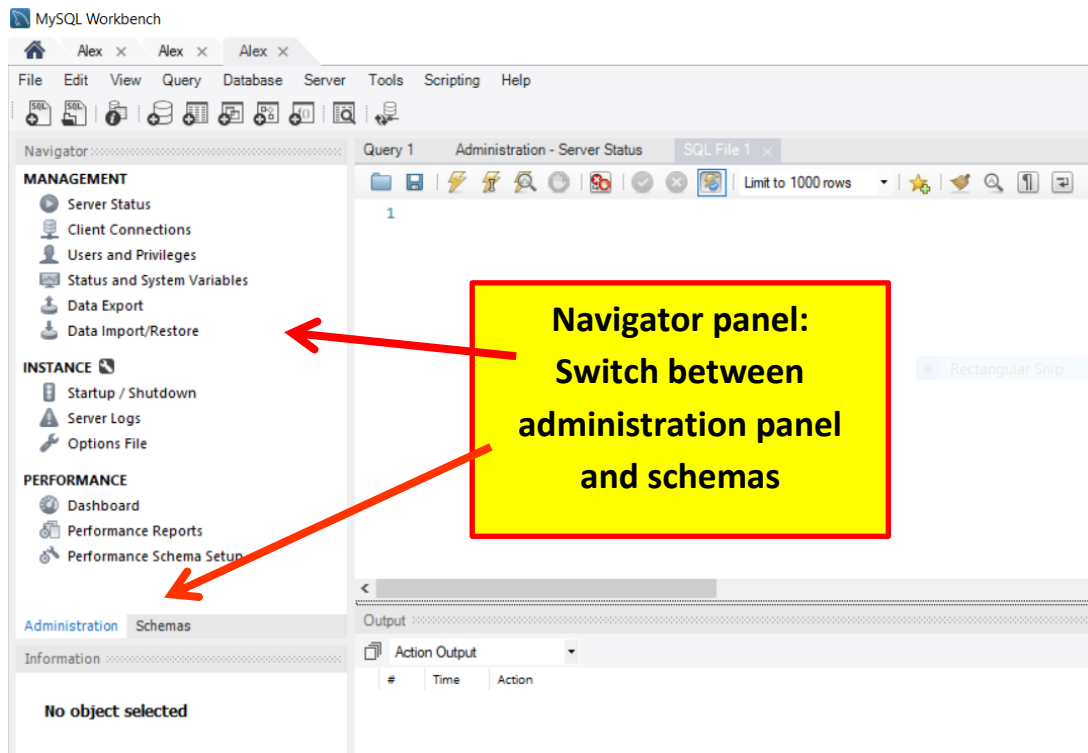
When we open MySQL we see this window



**MySQL  
connection.**

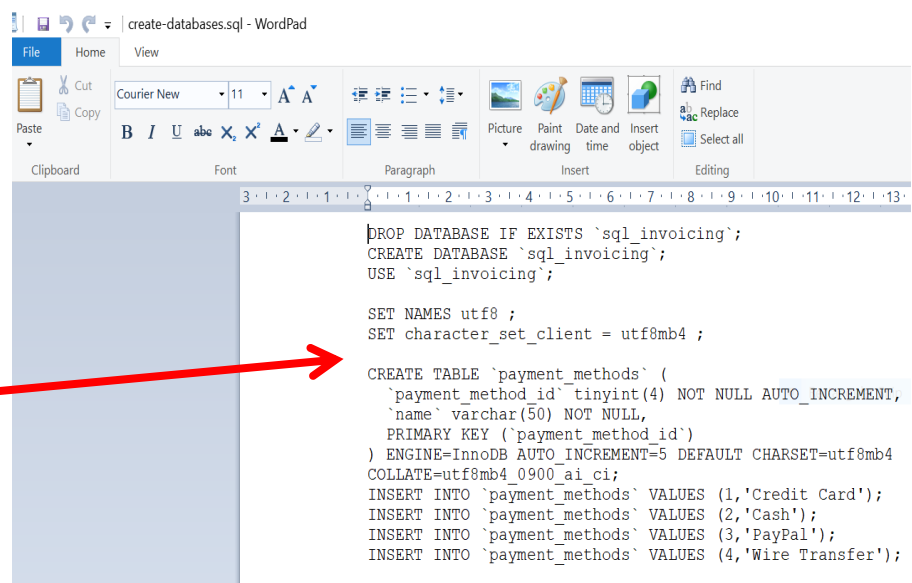


**MySQL  
Workbench 8.0**

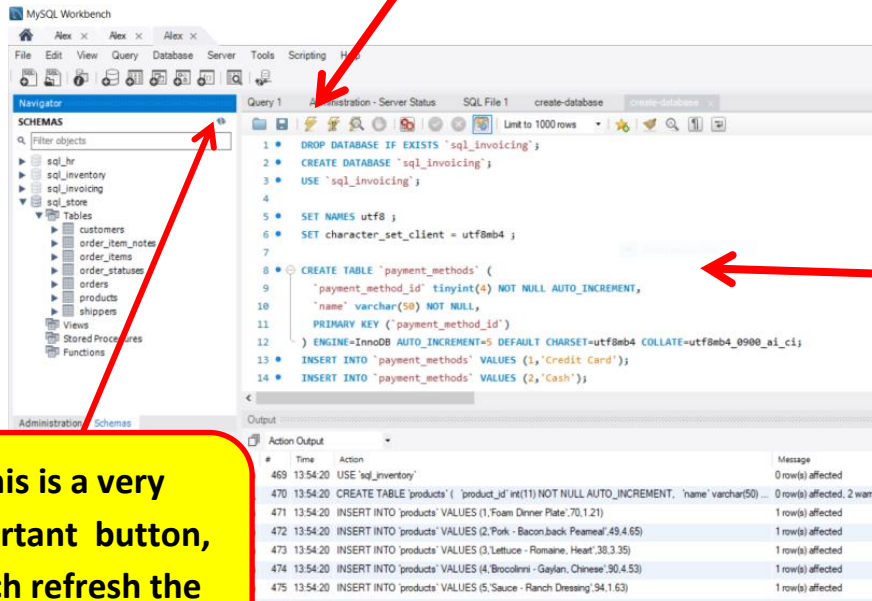


We open this text file create-databases.sql, select everything, copy and paste into the SQL workbench.

Select  
CNTL + A



**RUN BUTTON**

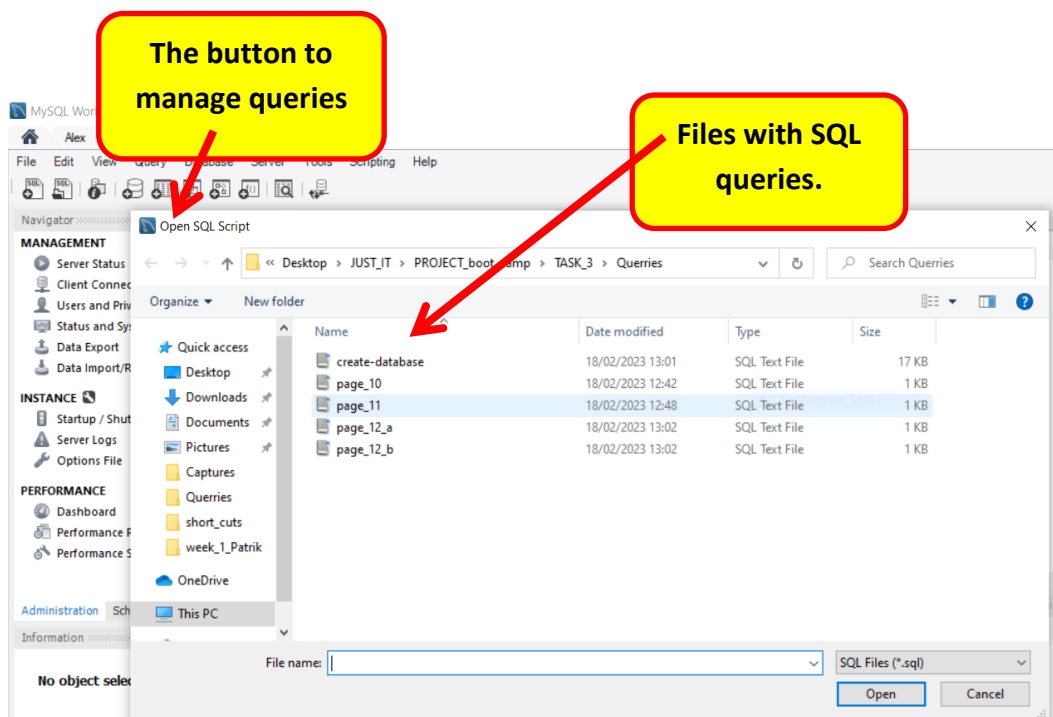


**The place where we write a code, a query**

**This is a very important button, which refresh the SQL tables.**

Once we pasted the code and run the query, the result of the query will appear bellow the code. Particularly this code generated a few tables in the schema. These tables can be viewed in the schema section.

Once we run the query we can save it for next time. On the picture below I showed all queries saved in one folder.



# First SQL queries

1)

**SELECT \*** -- displaying everything

**FROM Customers** -- from table Customers

The screenshot shows the SQL Developer interface. On the left, the 'SCHEMAS' pane shows a tree view with 'sql\_store' expanded, showing 'Tables' and 'Columns'. The 'Columns' pane lists columns for 'customer\_id', 'first\_name', 'last\_name', 'birth\_date', 'phone', 'address', 'city', 'state', and 'points'. The main query editor shows the following SQL code:

```
1
2 • USE sql_store;
3
4 • SELECT *
5 FROM CUSTOMERS
```

The 'Result Grid' shows the following data:

customer_id	first_name	last_name	birth_date	phone	address	city	state	points
1	Babara	MacCaffrey	1986-03-28	781-932-9754	0 Sage Terrace	Waltham	MA	2273
2	Ines	Brushfield	1986-04-13	804-427-9456	14187 Commercial Trail	Hampton	VA	947
3	Freddi	Boagey	1985-02-07	719-724-7869	251 Springs Junction	Colorado Springs	CO	2967
4	Ambur	Roseburgh	1974-04-14	407-231-8017	30 Arapahoe Terrace	Orlando	FL	457
5	Clemmie	Betchley	1973-11-07	312-480-8498	5 Spohn Circle	Arlington	TX	3675
6	Elka	Twiddell	1991-09-04	7 Manley Drive	Chicago	IL	3073	

2)

**SELECT \*** -- displaying everything

**FROM Customers** -- from table Customers

**--WHERE CUSTOMER\_ID = 1**

**ORDER BY first\_name ;** -- sorted by first name

The screenshot shows the SQL Developer interface. On the left, the 'SCHEMAS' pane shows a tree view with 'sql\_store' expanded, showing 'Tables' and 'Columns'. The 'Columns' pane lists columns for 'customer\_id', 'first\_name', 'last\_name', 'birth\_date', 'phone', 'address', 'city', 'state', and 'points'. The main query editor shows the following SQL code:

```
3
4 • SELECT *
5 FROM CUSTOMERS
6 -- WHERE CUSTOMER_ID = 1
7 ORDER BY first_name;
```

The 'Result Grid' shows the following data:

customer_id	first_name	last_name	birth_date	phone	address	city	state	points
4	Ambur	Roseburgh	1974-04-14	407-231-8017	30 Arapahoe Terrace	Orlando	FL	457
1	Babara	MacCaffrey	1986-03-28	781-932-9754	0 Sage Terrace	Waltham	MA	2273
5	Clemmie	Betchley	1973-11-07	312-480-8498	5 Spohn Circle	Arlington	TX	3675
6	Elka	Twiddell	1991-09-04	7 Manley Drive	Chicago	IL	3073	
3	Freddi	Boagey	1985-02-07	719-724-7869	251 Springs Junction	Colorado Springs	CO	2967
7	Ilene	Dowson	1964-08-30	615-641-4759	50 Lillian Crossing	Nashville	TN	1672

3) Task: - Select last\_name, first\_name, points, points +10  
- from CUSTOMERS

The screenshot shows a SQL query editor window titled "SQL File 5\*" with a toolbar and a query window. The query is as follows:

```
1  
2 • USE sql_store;  
3  
4 • SELECT last_name, first_name, points, points + 10  
5 FROM CUSTOMERS  
6
```

Below the query editor, the "Result Grid" is displayed, showing the results of the query. The grid has five columns: last\_name, first\_name, points, and points + 10. The results are as follows:

	last_name	first_name	points	points + 10
▶	MacCaffrey	Babara	2273	2283
	Brushfield	Ines	947	957
	Boagey	Freddi	2967	2977
	Roseburgh	Ambur	457	467
	Betchley	Clemmie	3675	3685

Result 4 x

4) Task: Using the Query 2 you created change the points to reads times by 10 and plus 100. Record your results in your word document

The screenshot shows a SQL query editor window titled "SQL File 5\*" with a toolbar and a query window. The query is as follows:

```
1  
2 • USE sql_store;  
3  
4 • SELECT last_name, first_name, points, points * 10 + 100  
5 FROM CUSTOMERS  
6
```

Below the query editor, the "Result Grid" is displayed, showing the results of the query. The grid has five columns: last\_name, first\_name, points, and points \* 10 + 100. The results are as follows:

	last_name	first_name	points	points * 10 + 100
▶	MacCaffrey	Babara	2273	22830
	Brushfield	Ines	947	9570
	Boagey	Freddi	2967	29770
	Roseburgh	Ambur	457	4670
	Betchley	Clemmie	3675	36850

Result 7 x



5)

**Task: Change the Query 2 code to create a discount factor so the table now shows a discount header and changing the (point + 10) \*100**

The screenshot shows a SQL query window with the following code:

```

1
2 • USE sql_store;
3
4 • SELECT last_name, first_name, points, (points + 10) * 100
5   FROM CUSTOMERS
6

```

The results are displayed in a grid with the following columns: last\_name, first\_name, points, and (points + 10) \* 100. The data is as follows:

last_name	first_name	points	(points + 10) * 100
MacCaffrey	Babara	2273	228300
Brushfield	Ines	947	95700
Boagey	Freddi	2967	297700
Roseburgh	Ambur	457	46700
Betchley	Clemmie	3675	368500

6)

Write a SQL query to return all the products in our database in the result set. I want to make three new columns, name, unit price, and new column called new price which is based on this expression, (unit price \* 1.1 ). So what you are doing is increasing the product price of each by 10%. So with the query we want all the products the original price and the new price:

The screenshot shows a SQL query window with the following code:

```

1
2 • SELECT name, unit_price, unit_price * 1.1
3   FROM products;
4
5

```

The results are displayed in a grid with the following columns: name, unit\_price, and unit\_price \* 1.1. The data is as follows:

name	unit_price	unit_price * 1.1
Foam Dinner Plate	1.21	1.3310
Pork - Bacon,back Peameal	4.65	5.1150
Lettuce - Romaine, Heart	3.35	3.6850
Brocolinni - Gaylan, Chinese	4.53	4.9830
Sauce - Ranch Dressing	1.63	1.7930
Petit Baguette	2.39	2.6290
Sweet Pea Sprouts	3.29	3.6190
Island Oasis - Raspberry	0.74	0.8140
Longan	2.26	2.4860
Broom - Push	1.09	1.1990

7)

In this task create a new query to find all the customers with a birth date of > '1990-01-01'  
Enter the following:

```
SELECT *  
FROM Customers  
WHERE birth_date > '1990-01-01'
```

The screenshot shows a database management tool interface. On the left is a 'SCHEMAS' tree view with a search filter. The main area displays a SQL query in a text editor, and below it, a 'Result Grid' showing the query results.

**SCHEMAS**

Filter objects

- sql\_hr
- sql\_inventory
- sql\_invoicing
  - Tables
    - clients
      - Columns
        - client\_id
        - name
        - address
        - city
        - state
        - phone
      - Indexes
      - Foreign Keys
      - Triggers
    - invoices
    - payment\_methods
    - payments
    - Views

**Query 1** Administration - Server Status SQL File 5\* SQL File 6\* x

Limit to 1000 rows

```
1  
2 • SELECT *  
3 FROM Customers  
4 WHERE birth_date > '1990-01-01'  
5  
6
```

**Result Grid** Filter Rows: Edit: Export/Import: Wrap Cell Content: I A

	customer_id	first_name	last_name	birth_date	phone	address	city	state	points
▶	6	Elka	Twiddell	1991-09-04	312-480-8498	7 Manley Drive	Chicago	IL	3073
	8	Thacher	Naseby	1993-07-17	941-527-3977	538 Mosinee Center	Sarasota	FL	205
	9	Romola	Rumgay	1992-05-23	559-181-3744	3520 Ohio Trail	Visalia	CA	1486
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

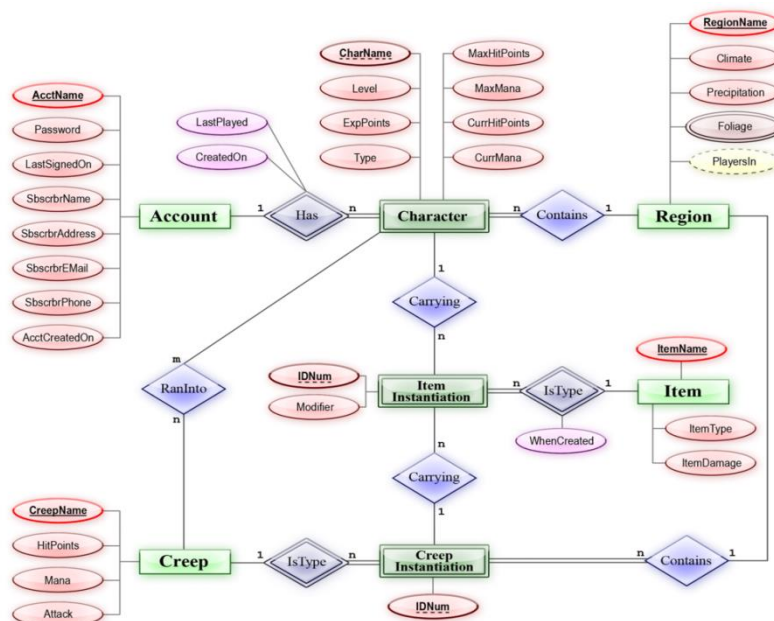
# Creating an EER Diagram



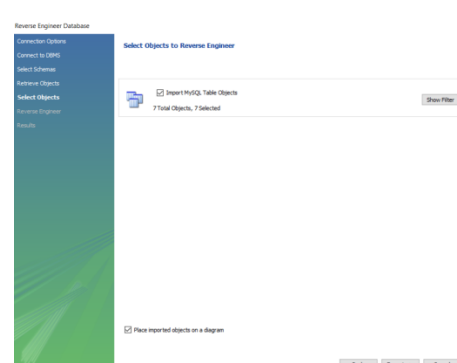
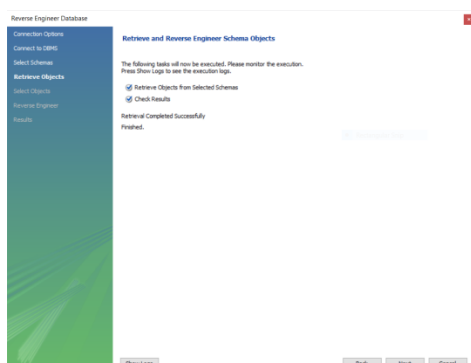
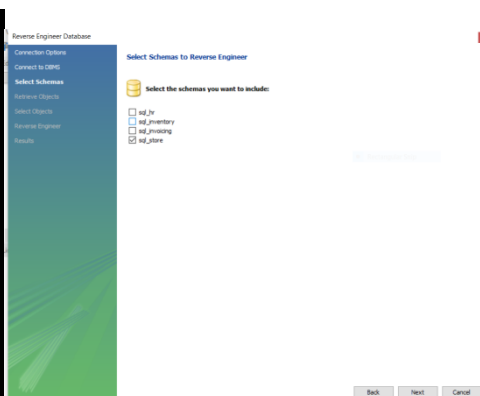
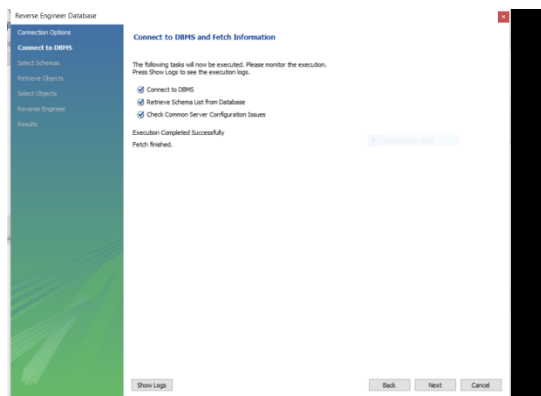
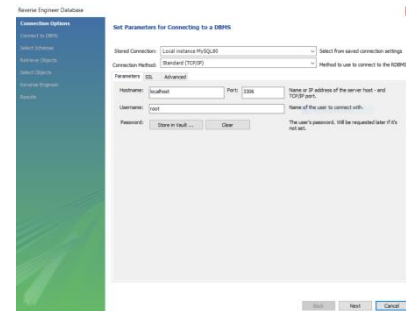
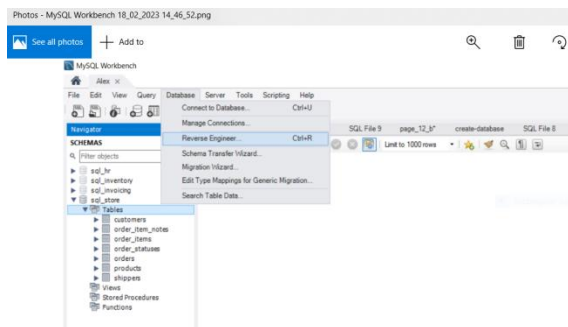
## From Wikipedia

An **entity–relationship model** (or **ER model**) describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between **entities** (instances of those entity types).

In **software engineering**, an ER model is commonly formed to represent things a business needs to remember in order to perform **business processes**. Consequently, the ER model becomes an abstract **data model**, that defines a data or information structure which can be implemented in a **database**, typically a **relational database**.



# Following a step by step instruction



# The EER DIAGRAM

It shows relationships between tables

