1. **SQL and installation of client/server**

Install SQL server and SSMS system:

1. Windows OS - <https://www.youtube.com/watch?v=kGdTg-vGs-E>

2. Mac OS

- Docker - <https://www.youtube.com/watch?v=glxE7w4D8v8>

- MySQL workbench- <https://youtu.be/-wpzS5NcYT8?si=UgsYv9QB2HRBxO6L>

**SSMS –**

**The Server need SQL client to query data, store data.**

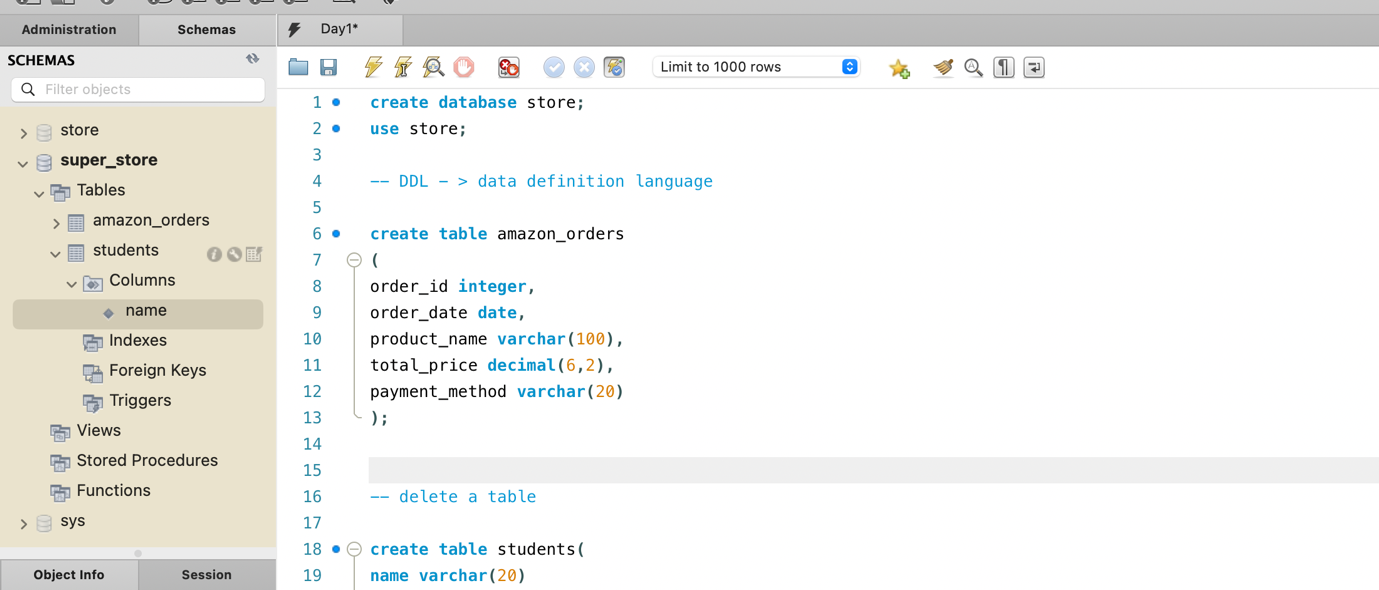
* SQL Server Management Studio (SSMS) is a software that helps
  + people manage and work with databases.
  + It can be used to set up,
  + keep an eye on, and
  + control different parts of a database, like SQL Server and Azure SQL Database.
* SSMS provides a variety of tools to make
  + these tasks easier,
  + such as the Object Explorer,
  + which allows users to browse and
  + act upon objects within the server.
* It's a useful tool for
  + database administrators,
  + developers,
  + testers, and
  + others who work with databases.

**DATABASE CREATION IMPORTANCE (MICROSOFT SQL DB SERVER):**

* Imagine databases as a system analogous to
  + having various drives on a computer. In this analogy,
  + each drive represents a database, and
  + within each database (drive),
  + you can establish multiple folders.
* In the context of a server, consider
  + it as a central unit. For instance, you
  + might have different database instances,
  + similar to having distinct drives (D/E).
  + Within each database instance, you can further envision individual databases as distinct folders (C/D).
* Let's take the example of a server. Within this server,
  + there are default system databases, such as the model database. Additionally, you can
  + create new databases as needed.
  + Creating a new database is a straightforward process; you can do this by clicking on "New Database" and providing a name for it.
  + Alternatively, you can use commands like "CREATE DATABASE" followed by the database name.
* When working within a database,
  + you can create multiple tables.
  + To initiate a new query, click on "New Query," and the system will present a list of available databases for you to choose from.
  + Once you select a database,
  + you can perform operations within it.
  + This flexibility allows you to organize and
  + manage data efficiently.

**DB SCHEMA in workbench:**

**Navigator:**



## **SQL Database**

## **The SQL CREATE DATABASE Statement**

The CREATE DATABASE statement is used to create a new SQL database.

### Syntax

CREATE DATABASE databasename;

## **CREATE DATABASE Example**

The following SQL statement creates a database called "testDB":

### Example

CREATE DATABASE store;

**Tip:** Make sure you have admin privilege before creating any database. Once a database is created, you can check it in the list of databases with the following SQL command: SHOW DATABASES;

Its syntax allows users to perform various actions such as accessing,

retrieving, sorting, and updating information in the database.

The step of creating a database is crucial in a server because it allows for the storage and management of data,

providing data access for authorized users

**USE** testDB;

The **USE** statement in SQL is used to select

* a specific database from a list of available databases.
* It allows you to perform operations on the selected database,
* such as creating tables,
* inserting data,
* updating data, and
* deleting data

[1](https://www.geeksforgeeks.org/sql-use-database-statement/" \t "_blank)

The syntax for the **USE** statement is **USE DatabaseName;**, where **DatabaseName** is the name of the database you want to select

After selecting the database,

* you can perform various operations on it,
* such as creating tables and inserting data

**AWS -**

AWS offers a range of cloud services with underlying data centers distributed globally.

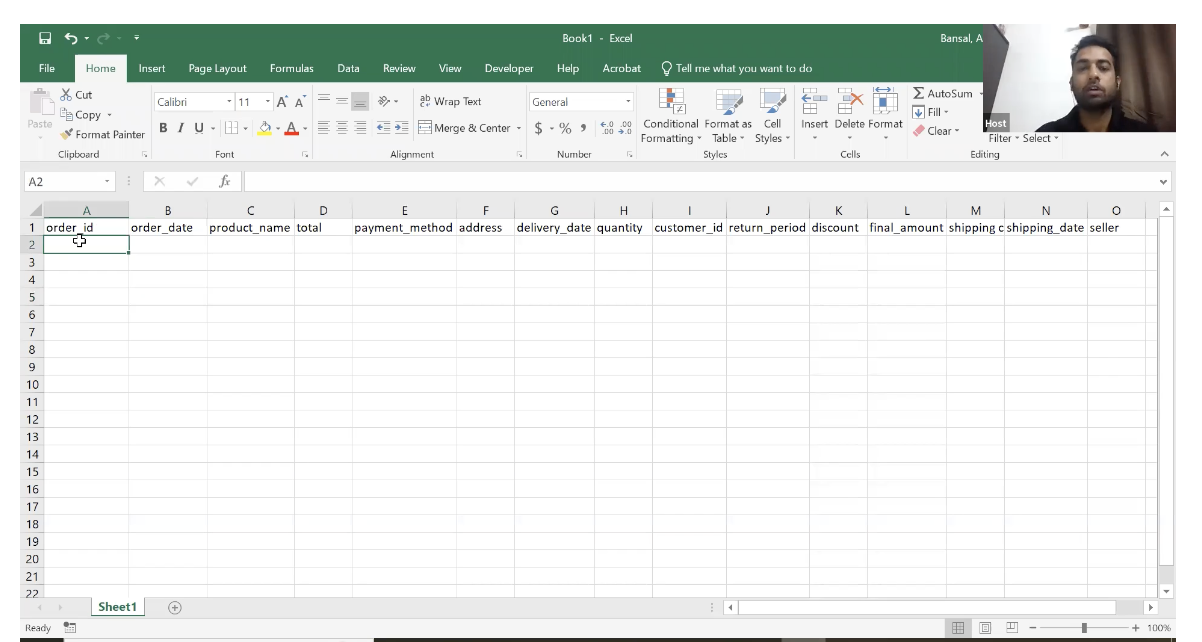
These data centers house hardware infrastructure, which is geographically dispersed. we take on the responsibility of managing these data centers, assessing the resource needs for each, such as the required race, CPUs, and other components.

Our operational tasks involve forecasting these requirements and collaborating with manufacturers and other vendors to ensure the timely availability of all necessary elements.

Amazon Web Services (AWS) is a cloud computing platform provided by Amazon that offers a wide range of cloud services,

* including infrastructure-as-a-service (IaaS),
* platform-as-a-service (PaaS), and
* software-as-a-service (SaaS) offerings.
* AWS provides tools such as
* compute power,
* database storage, and
* content delivery services, and it
* has become a highly reliable,
* scalable, and
* low-cost infrastructure platform used by hundreds of thousands of businesses worldwide.
* It allows users to access data from a remote server and
* store and access data from anywhere across the world.
* AWS has over 200 services and
* data centers located around the world,
* making it a popular choice for organizations looking to leverage cloud computing for various business needs.
* AWS is used by many companies like ESPN, Adobe, Twitter, Netflix, Facebook, and BBC, among others

**Example – Amazon Order**



Let's delve into the technical details of the order placement process on an e-commerce platform, using Amazon as an example.

When a user decides to make a purchase,

* they initiate the transaction by selecting an item, such as shoes, and
* proceeding to the checkout.
* Upon confirming the order,
* the system captures various details, and
* this information is then stored in the backend.

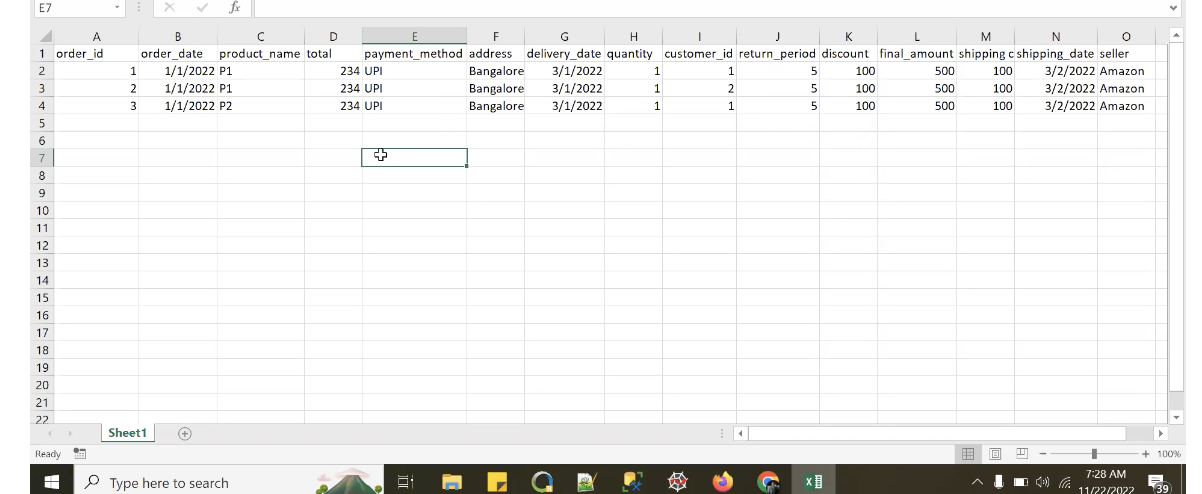
In a hypothetical scenario where the data is stored in a spreadsheet (akin to an early stage of Amazon without a sophisticated database), the key attributes and their corresponding values include:

1. Order Number/ID: This serves as a unique identifier for the order.
2. Order Date: Indicates when the order was placed.
3. Product Name: Specifies the name of the purchased item.
4. Total Price: Reflects the overall cost of the order, encompassing the product cost, shipping charges, and any applicable promotions or discounts.
5. Quantity: Represents the quantity of the product bought in the order.
6. Customer ID/User ID: Identifies the customer who placed the order.
7. Delivery Address: Specifies the address where the order will be delivered.
8. Delivery Date: Indicates the scheduled date for order delivery.
9. Product ID: A unique identifier for the purchased product.
10. Return Information: Captures whether the customer opted for a return.
11. Discount Amount: Displays any discounts applied to the order.
12. Final Amount: Reflects the total amount to be paid after considering discounts and additional charges.
13. Shipping Charges: Specifies any costs associated with shipping.
14. Shipping Date: Denotes the date when the order is shipped.
15. Seller Information: Identifies the seller offering the product.

It's important to note that this data storage mechanism is illustrative, assuming a basic system represented by an Excel spreadsheet.

In reality, modern e-commerce platforms employ sophisticated databases to manage and retrieve such information efficiently. The mentioned attributes collectively form a comprehensive record of each transaction, allowing users to review their order history and facilitating efficient order management within the system.

**EXCEL WITH DATA**



The information discussed remains consistent for all customers.

For example,

* if another customer places a new order,
* the order ID will increment (e.g., Order ID 2),
* and their customer ID will be distinct (e.g., Customer ID 2).
* Similarly, if there is a third order (Order ID 3) from the same customer, the customer ID remains constant.
* When you access the order history page, the system identifies your customer ID.
* Subsequently, it queries the backend to retrieve all orders associated with that specific customer ID.
* This information is then presented on your user interface (UI).
* Customer ID serves as a crucial identifier, allowing the system to personalize and display relevant order history upon login.

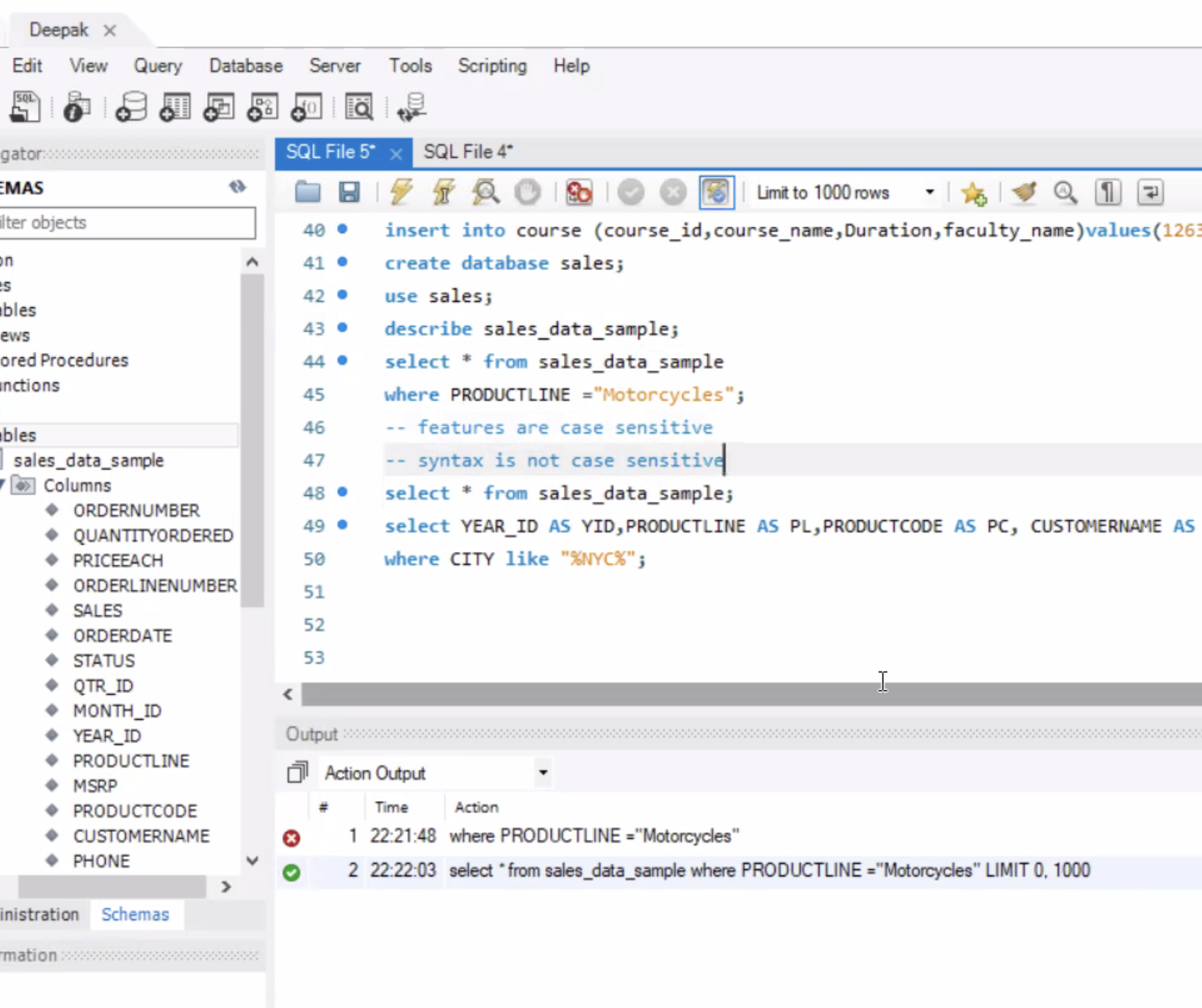
However, as Amazon experiences substantial growth and accumulates a significant volume of data,

* relying on an Excel-based system becomes impractical.
* Excel, with its limitations, such as a maximum of 1,000,000 records and
* performance issues with extensive data,
* poses challenges for efficient data handling and analysis.
* Managing large datasets in Excel becomes unfeasible,
* leading to the necessity of transitioning to a more robust solution: a database.

**DATABASE**

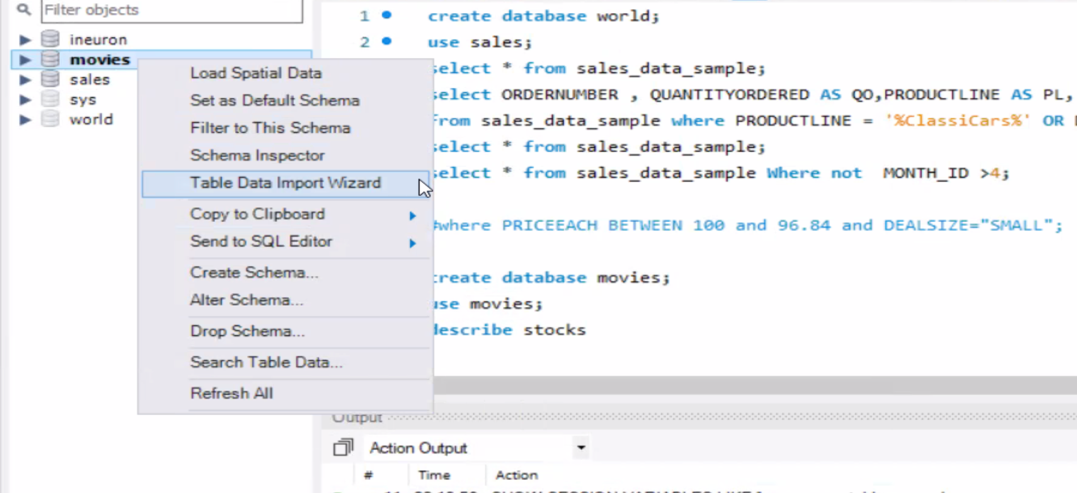
* To overcome these limitations,
  + one must transition to using a database management system (DBMS).
* A database is a
  + ***structured collection of data*** that is managed by a DBMS.
  + There are various DBMS options available,
  + including Oracle,
  + Microsoft SQL,
  + Teradata,
  + MySQL, and others.
* Unlike Excel, databases offer
  + scalability and
  + flexibility,
  + allowing users to increase capacity,
  + memory, and
  + hardware based on their requirements.
  + There are no inherent constraints on the amount of data that can be stored in a database.
* In a database, users can create objects
  + to organize and
  + store data efficiently. Additionally,
  + databases facilitate easy sharing of data among multiple users,
  + each with their respective access permissions.
* Unlike Excel, which may struggle with performance when handling large datasets,
  + databases can handle vast amounts of data
  + without significant performance issues.
* SQL Server is one such database management system, and users are encouraged to install it to leverage its capabilities.

**In SQL:**

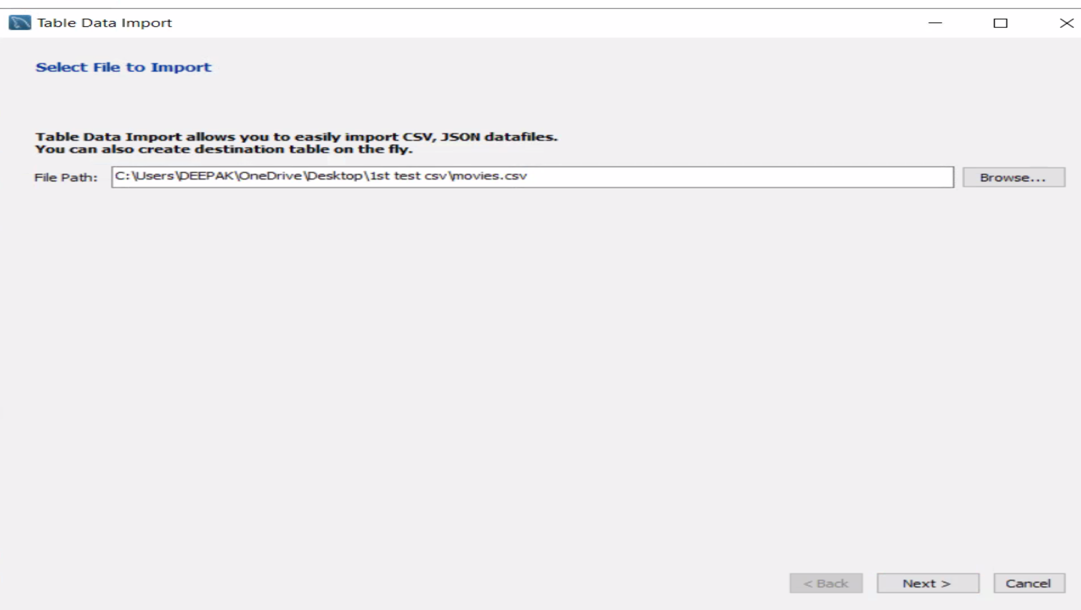


**LOAD DATA IN MYSQL WORKBENCH :**

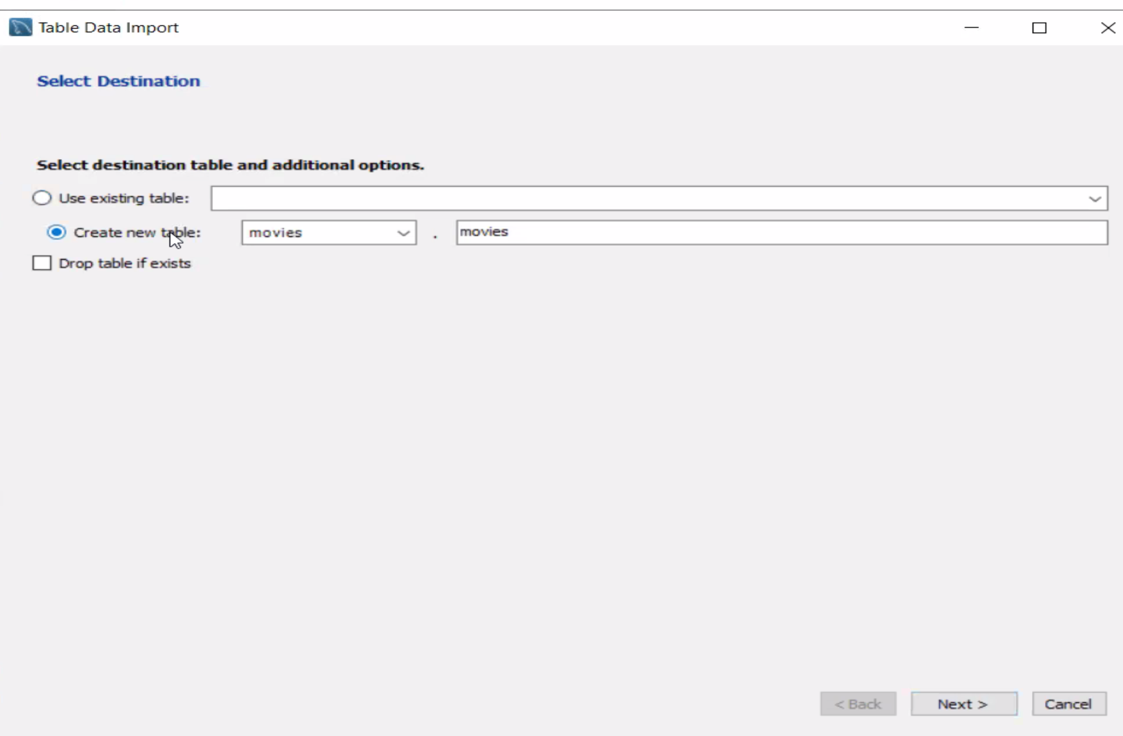
1. Select ‘Table Data Import Wizard’ from database Schema.

****

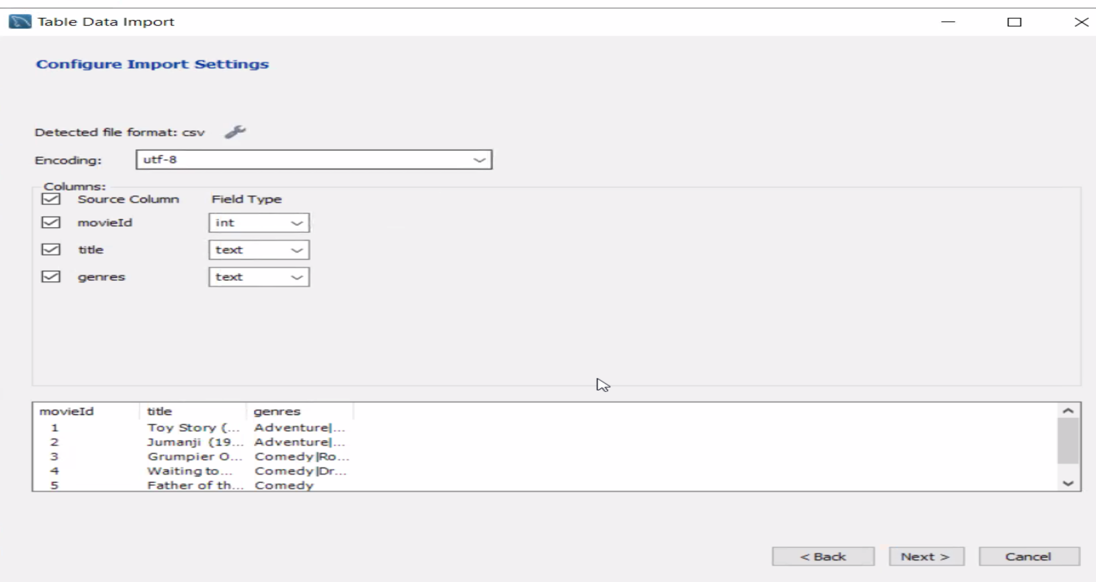
1. Import file from the Location/ Local Drive:

****

1. Select ‘Create New Table’.

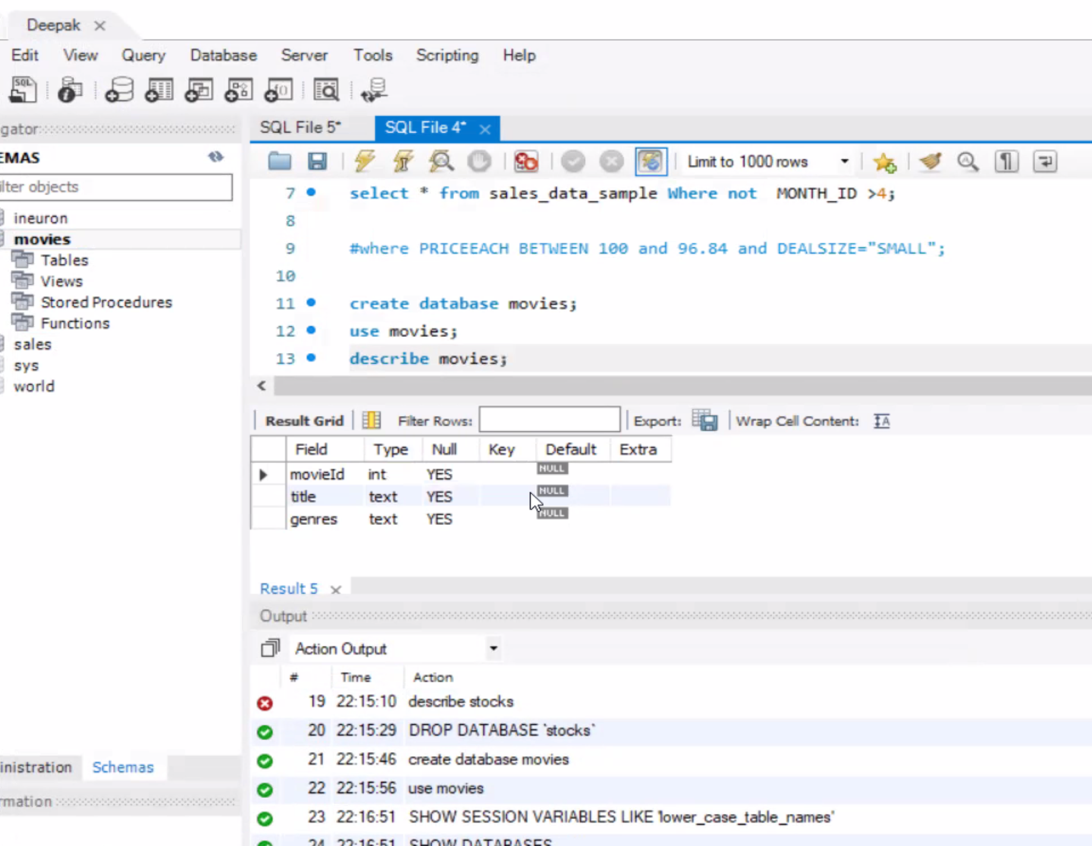
****

1. Check the data type of uploaded features. Change Accordingly if required.

****

1. Check the uploaded dataset.

Command - DESCRIBE movies;

****