

Data Warehousing with IBM Cloud Db2 Warehouse

Cloud Computing

Nalaya Thiran Project

by

Aashif Ali N	211521104001
K.S.Nithish Kumar	211521104099
K.Prabakar	211521104108
N.Maheshwaran	211521104084
P.Praveen Kumar	211521104114

from

Panimalar Institute of Technology

B.E

Computer Science Engineering

ABSTRACT

Data warehousing is the act of gathering, organising, and managing data from various sources in order to deliver useful business insights and projections to users. It is a relational database management system (RDBMS) design that is used to satisfy the needs of transaction processing systems. It is a database that stores information geared towards meeting decision-making needs. Data warehousing entails data cleaning, integration, and consolidation. The data in the data warehouse is not the same as the data in the operating environment. A data warehouse is built by combining data from disparate sources to support analytical reporting, structured and/or ad hoc queries, and decision making. This project will involve designing, creating, and putting into use a cutting-edge data warehousing system that is specifically customised to the needs of our organisation. It includes a number of crucial elements, including data storage, governance, security, and user access as well as data extraction, transformation, and loading (ETL). The successful execution of the Data Warehousing Project will equip our organization with a powerful platform for data-driven decision-making, enabling us to unlock valuable insights, optimize operations, and gain a competitive edge in today's dynamic business environment. This project signifies our commitment to leveraging data as a strategic asset to drive innovation and growth.

INTRODUCTION :

In today's world, organisations are always looking for new methods to leverage the power of data to make wise decisions and gain a competitive edge in today's data-driven world. By offering a centralised location for the storage, processing, and analysis of enormous amounts of data, data warehousing plays a significant part in this process. To ensure that an organization's unique demands and objectives are met, however, the creation of an efficient data warehouse necessitates thorough problem description and design thinking. The process for problem definition and design thinking for data warehousing is described in this document.

Problem Definition :

Data warehousing is a complex and challenging process, and there are many potential problems that can arise. Some of the most common problems include:

- Identifying Business Goals: State the organization's business goals and objectives clearly. Determine how data warehousing can help you achieve these objectives, such as better decision-making, better customer insights, or better operations.
- Problems with data warehouses: Although Gartner predicts that up to 80% of an organization's data is unstructured, traditional data warehouses can only store clean, highly structured data. They do not support communications frameworks like HL7, JSON, and XML, as well as unstructured data types like text, photos, and IoT data.
- Performance: Data warehouses must be able to efficiently handle high amounts of queries and transactions despite their potential size and complexity.
- Cost: Data warehousing can be expensive, both in terms of hardware and software costs, as well as the costs of implementing and maintaining a data warehouse.

Design Thinking :

When it comes to data warehousing, design thinking can be used to create a data warehouse that meets the needs of the users. Here are some steps that can be followed:

- Empathise:
 - To understand the needs and pain points of end users, such as analysts, managers, and executives, conduct interviews and workshops with them.
 - To represent various user groups and their demands within the data warehouse, create user personas.
- Define the problem:
 - The next step is to outline the problem that needs to be solved when the users' needs have been understood.
 - Clarifying the data warehouse's objectives, outlining the specific issues that it will be used to solve, and establishing the project's parameters are possible steps in this process.
- Ideate:
 - Identify potential architectures, features, and functionalities for data warehouses that would address the user stories that have been defined.
 - To develop creative ideas, promote collaboration across functional boundaries.
- Prototype:
 - After generating a number of concepts, the following step is to prototype the most promising solutions.
 - This entails developing a functioning model of the solution in order for it to be tested and assessed.
- Test:
 - Test the prototype with users and stakeholders to get feedback.
 - This feedback can be used to refine the solution.

- Data sources:
 - Determine and document all data sources that will be fed into the data warehouse. Databases, external APIs, flat files, CRM systems, and other sources can all be used.
 - Understand these sources' data formats, schemas, and data quality.
- Data integration:
 - Data warehouses frequently integrate data from several sources, which can be challenging and complex.
 - It's possible for data from several sources to have various formats, structures, and degrees of quality.
- ETL Processes:
 - ETL stands for Extract, Transform, and Load. It is a data integration process that combines data from multiple sources into a single, consistent data store that is loaded into a data warehouse or other target system.
 - Data Extraction (E):
 - Create procedures for data extraction from source systems. Setting up ETL (Extract, Transform, Load) pipelines or data integration technologies is frequently required for this.
 - Ensure fast, trustworthy, and minimally disruptive data extraction from source systems.
 - Data Transformation (T):
 - As data goes from the source to the data warehouse, it must be transformed and cleaned.
 - This could include data cleansing, enrichment, aggregation, and format changes.
 - During this stage, apply business rules, calculations, and data quality checks.
 - Data Loading (L):
 - Load the transformed data into the data warehouse. Loading can be done in different ways, such as batch processing or real-time streaming, depending on the organization's needs.
 - Ensure that the loading process is optimized for performance and scalability.

Algorithm:

Step1 : Create an IBM account by providing the necessary details to open an IBM account.

Step2 : Create an IBM Cloud Db2 Warehouse instance.

Step3 : Connect Db2 Warehouse to the data sources. Databases, flat files, cloud storage, APIs, and other similar sources may be used.

Step4 : To move data from source systems to the Db2 Warehouse, use ETL (Extract, Transform, Load) processes.

Step5 : Create and design the data warehouse schema by creating tables, relationships, and data types.

Step6 : Load the data into Db2 Warehouse.

Step7 : To clean, enrich, and prepare data for analysis, use data transformations.

Step8 : Use SQL transformations tools to do the data transformations.

Step9 : Eventhough the IBM Cloud Db2 Warehouse do provisioning or managing infrastructure, perform some configuration and management tasks such as creating users, roles, and schemas.

Step10 : Begin designing and deploying applications for data querying and analysis.

Step11 : Optimize SQL queries and performance with tools such as Db2's query optimization features or by designing efficient data models.

Step12 : Utilize monitoring tools and utilities of IBM Cloud Db2 Warehouse to implement monitoring and alerting to keep an eye on system performance and data quality.

Step13 : Execute maintenance chores like data cleaning, indexing, and backup/restore operations on a regular basis.

Step14 : Create backup and disaster recovery plans to guarantee data integrity and availability in the event of system failures.

Step15 :Adjust the resources allotted to the Db2 Warehouse instance.

Step16 : Ensure that the data warehouse is used in a responsible and ethical manner.

Step17 :Create documentation for all processes, data dictionaries, and data warehouse architecture.

Step18 :Users and administrators should receive training on how to utilize and maintain the data warehouse.

Step19 :Ensure that the data model and queries are performance optimized.

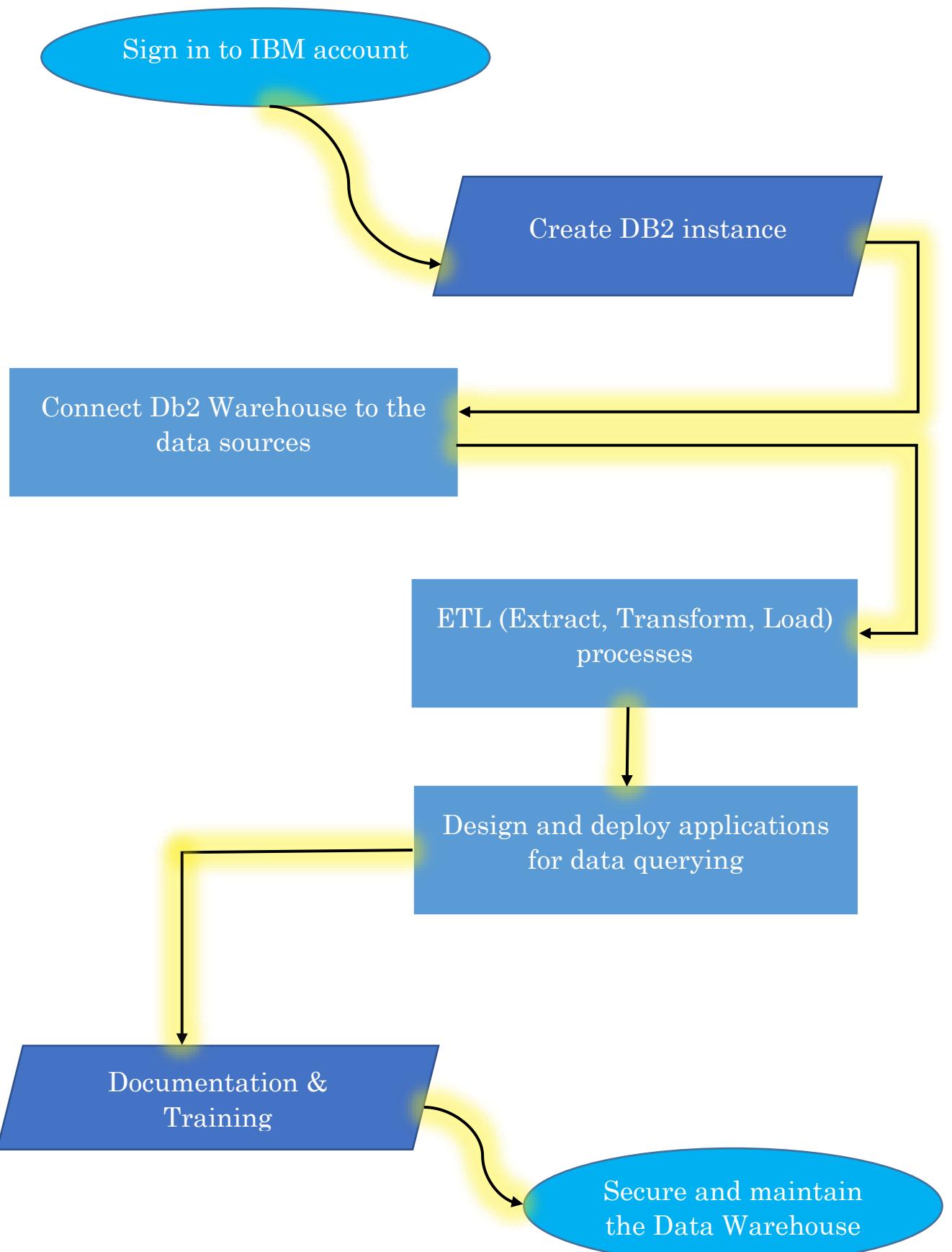
Step20 :As the workload changes, continuously check and optimize the data warehouse's performance.

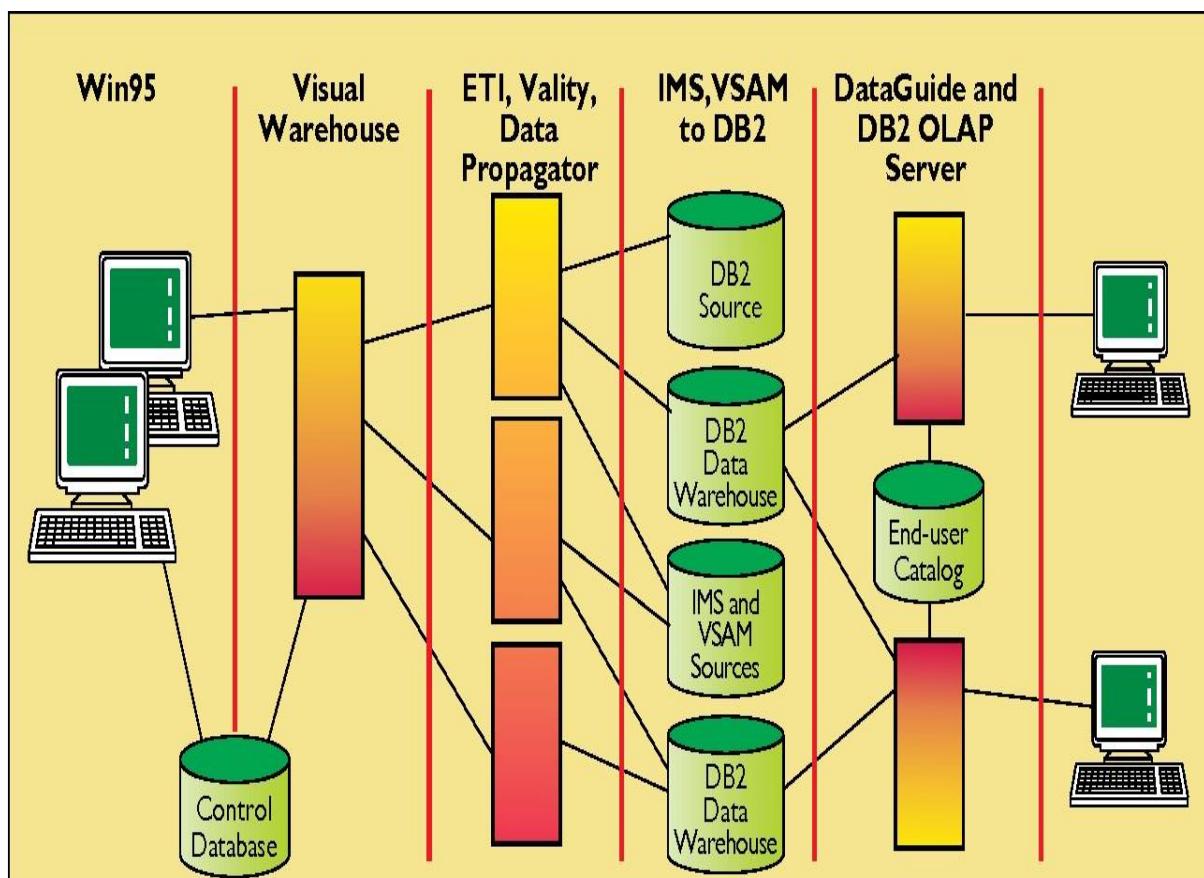
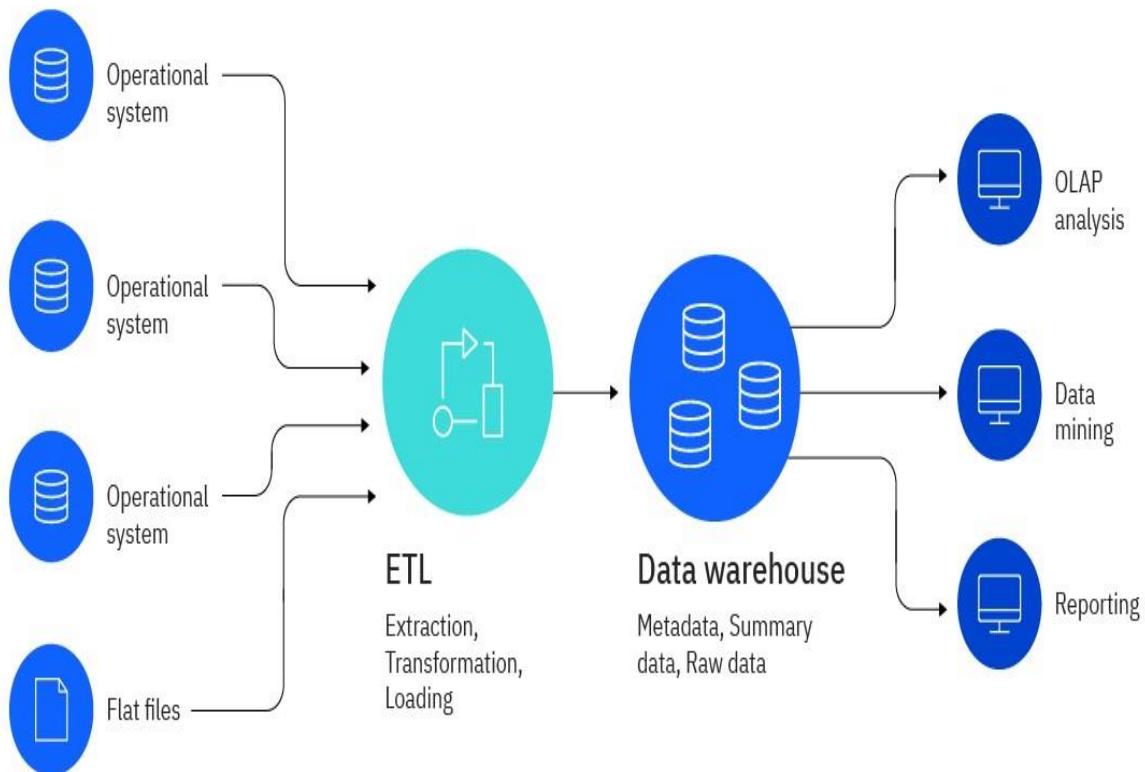
Step21 : Implement adequate security measures to prevent unauthorized access to the data warehouse.

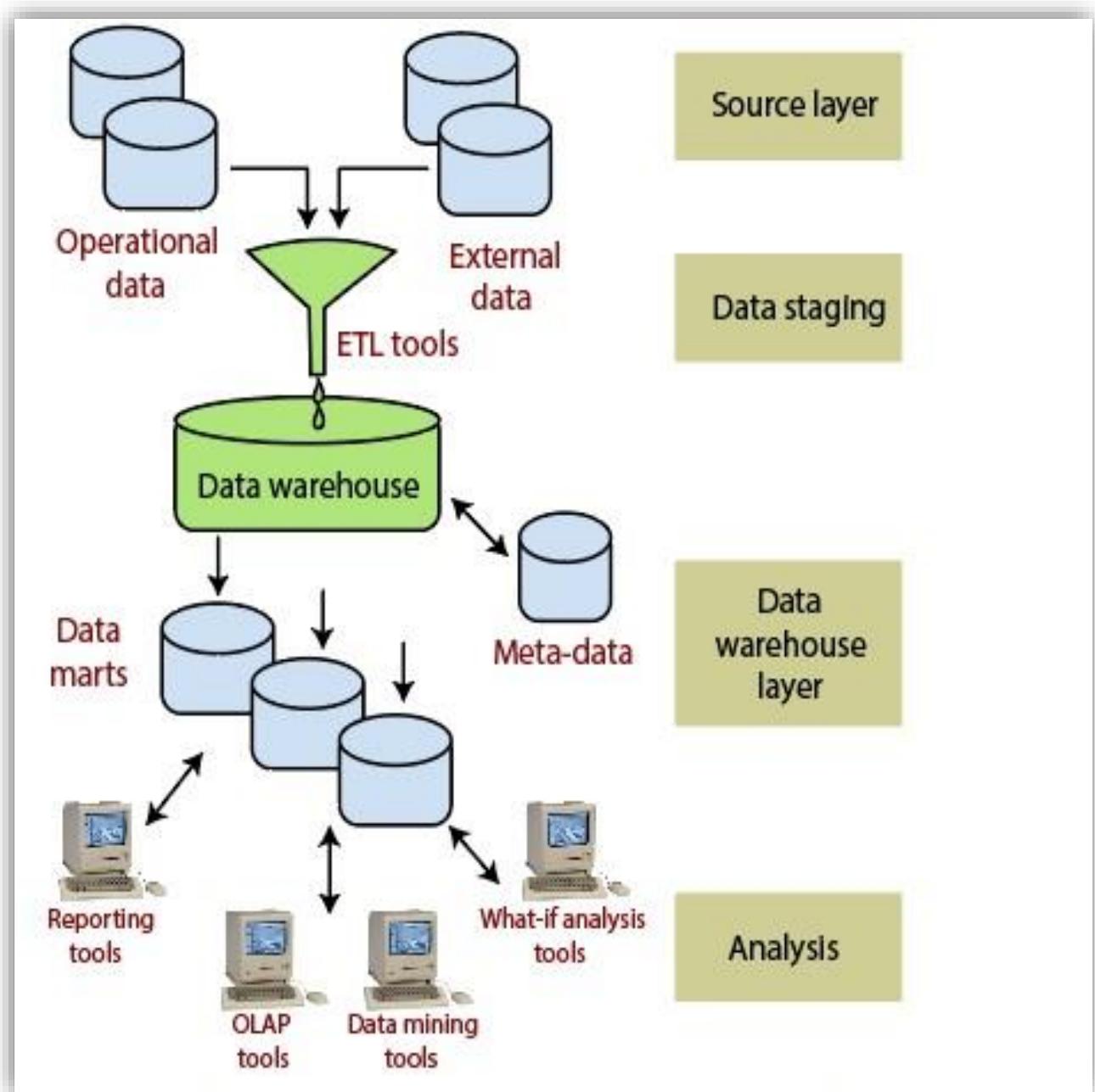
Step22 : Put data archiving and disposal rules into place to manage data preservation and compliance

Step23 : Prepare to scale up or down the Db2 Warehouse setup as needed.

Flow Chart :



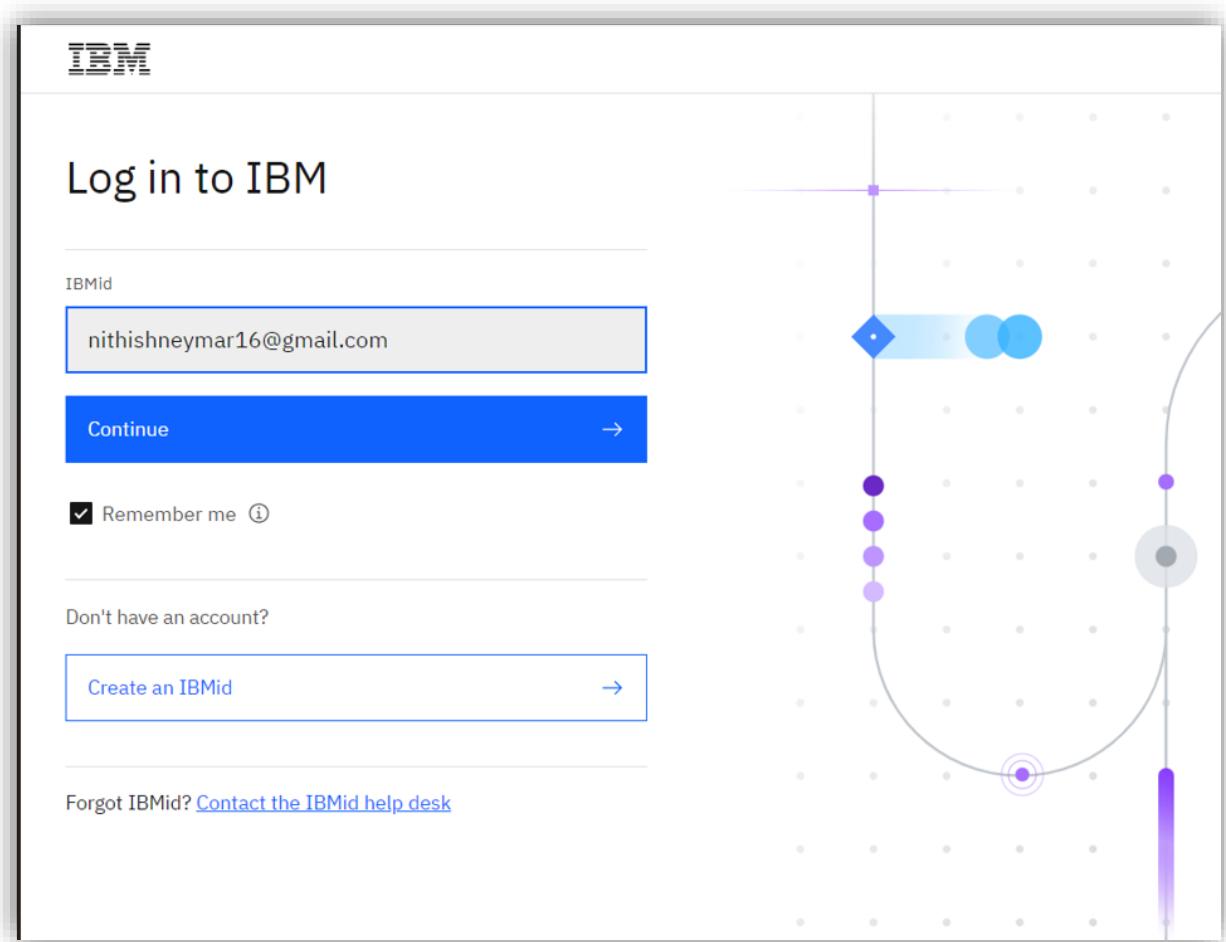


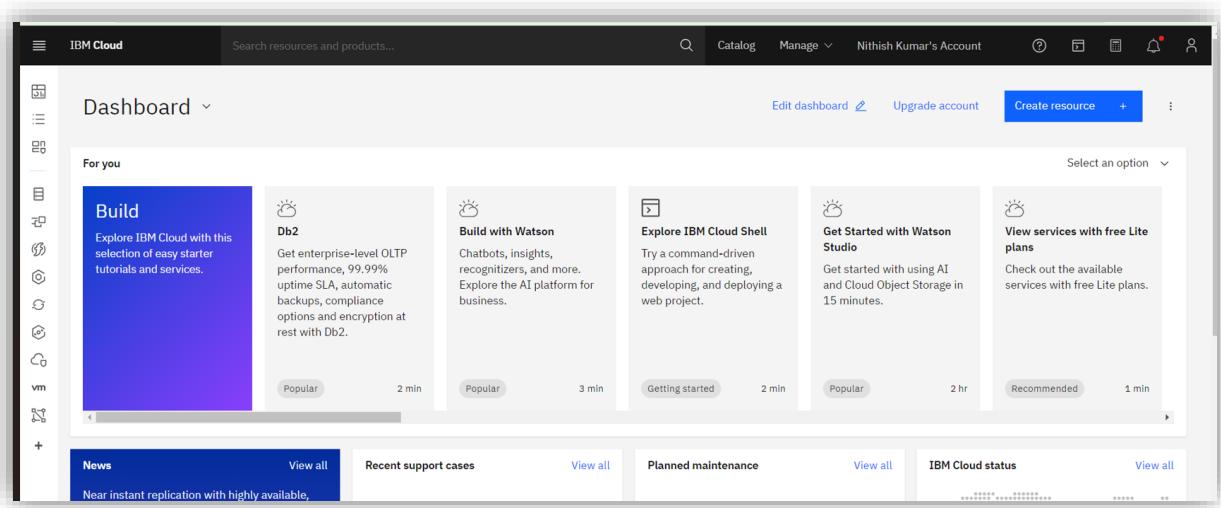
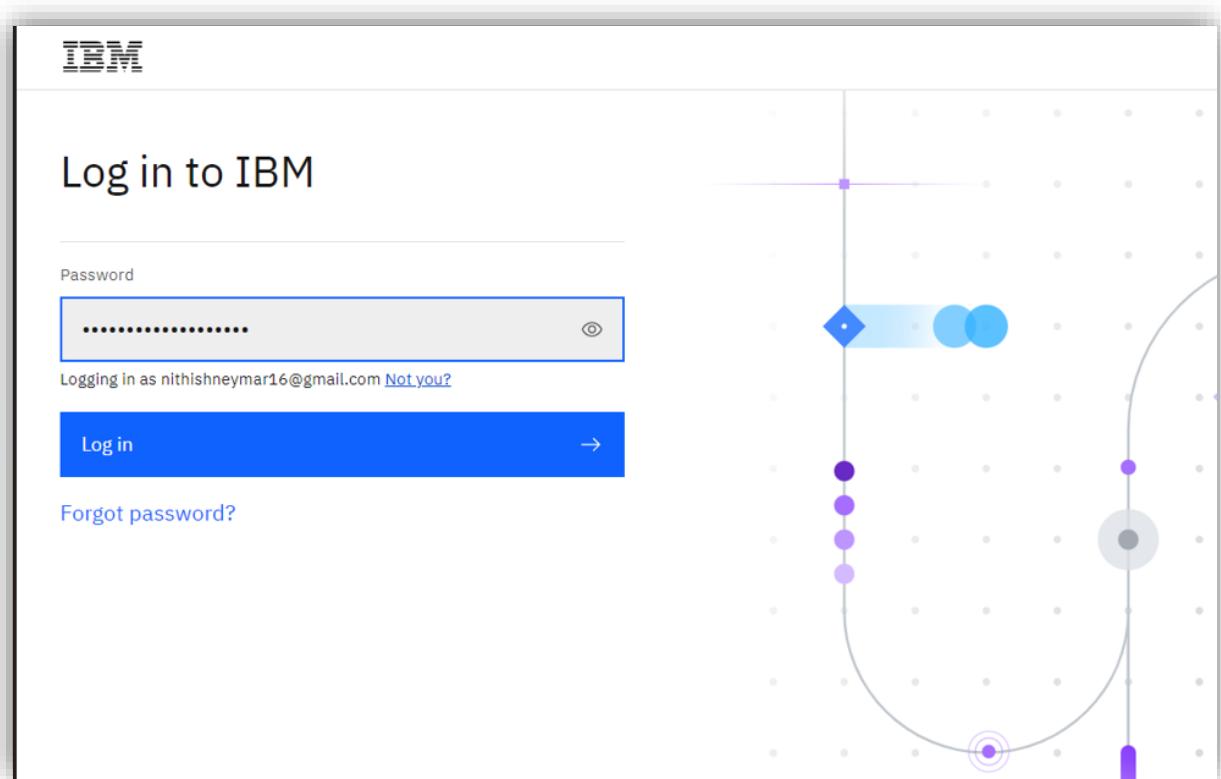


Phase 3

The dataset is loaded into the IBM cloud for further processing. Identify data sources and develop an integration plan for them in the data warehouse.

- The first step is to log in to the IBM account.

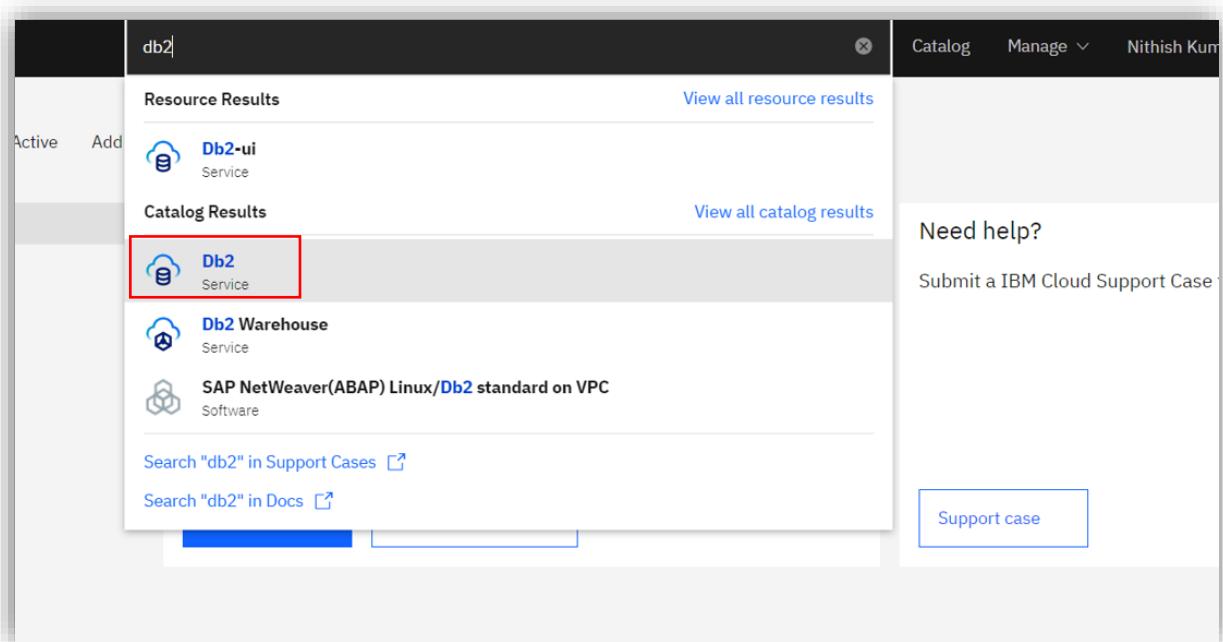




- Then the dashboard will be displayed.

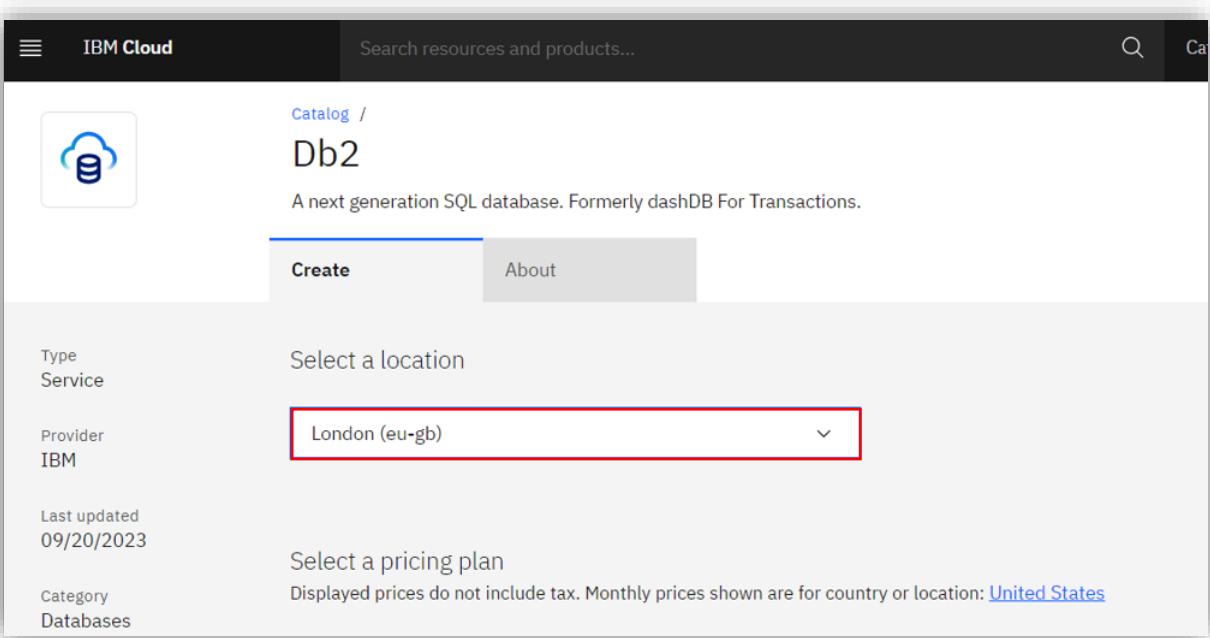
The second step is to create a DB2 service.

- Then search **Db2** in the search bar.



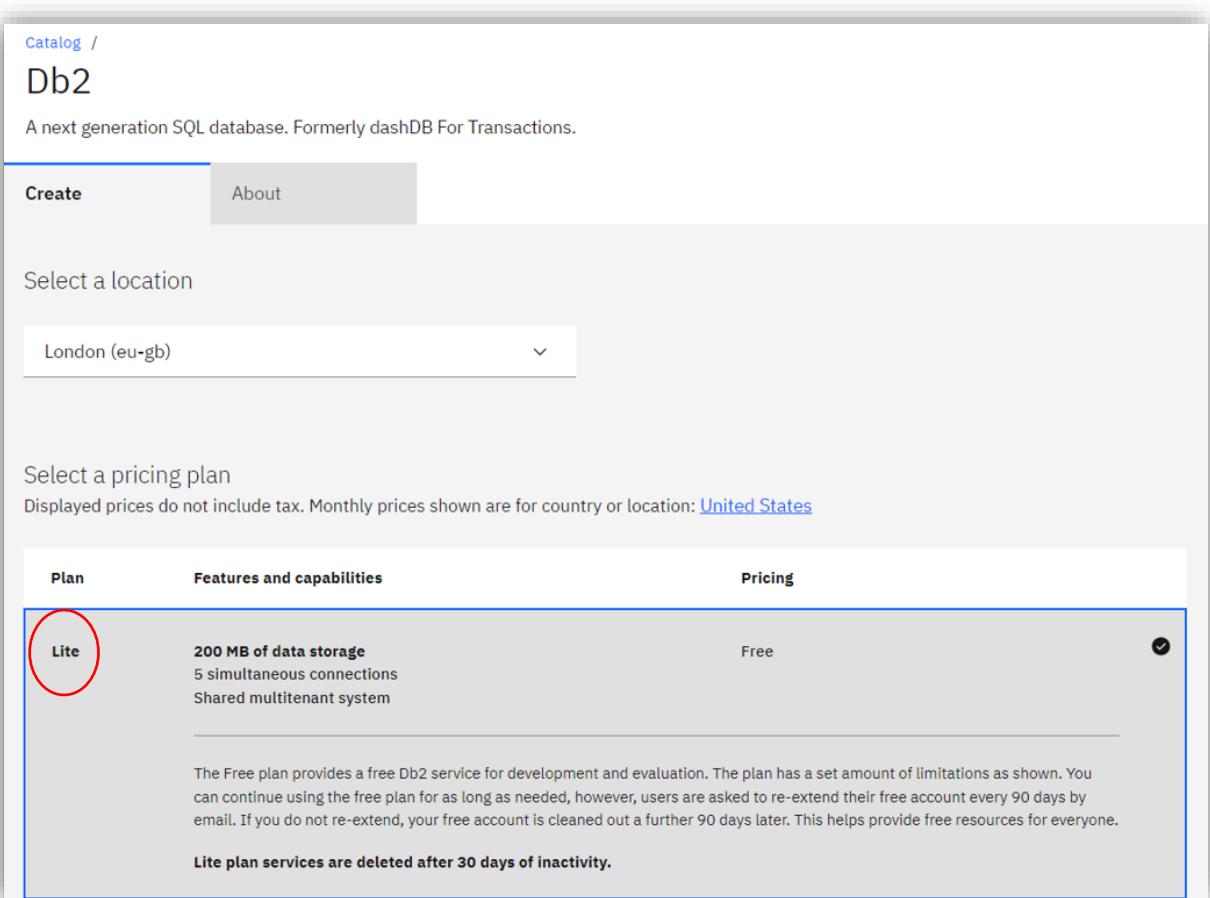
- In that choose **Db2 service**.

- Select the location as [London](#).



The screenshot shows the IBM Cloud Catalog interface. At the top, there's a navigation bar with 'IBM Cloud' and a search bar. Below it, the 'Catalog' section is displayed for 'Db2'. A sub-header states: 'A next generation SQL database. Formerly dashDB For Transactions.' There are two tabs: 'Create' (which is active) and 'About'. On the left, there are service details: 'Type Service', 'Provider IBM', 'Last updated 09/20/2023', and 'Category Databases'. In the center, there's a 'Select a location' dropdown menu with 'London (eu-gb)' selected. Below the dropdown, there's a 'Select a pricing plan' section with a note: 'Displayed prices do not include tax. Monthly prices shown are for country or location: [United States](#)'.

- Choose the **Lite** version which is provided by the IBM.



Catalog /

Db2

A next generation SQL database. Formerly dashDB For Transactions.

Create About

Select a location

London (eu-gb) ▾

Select a pricing plan

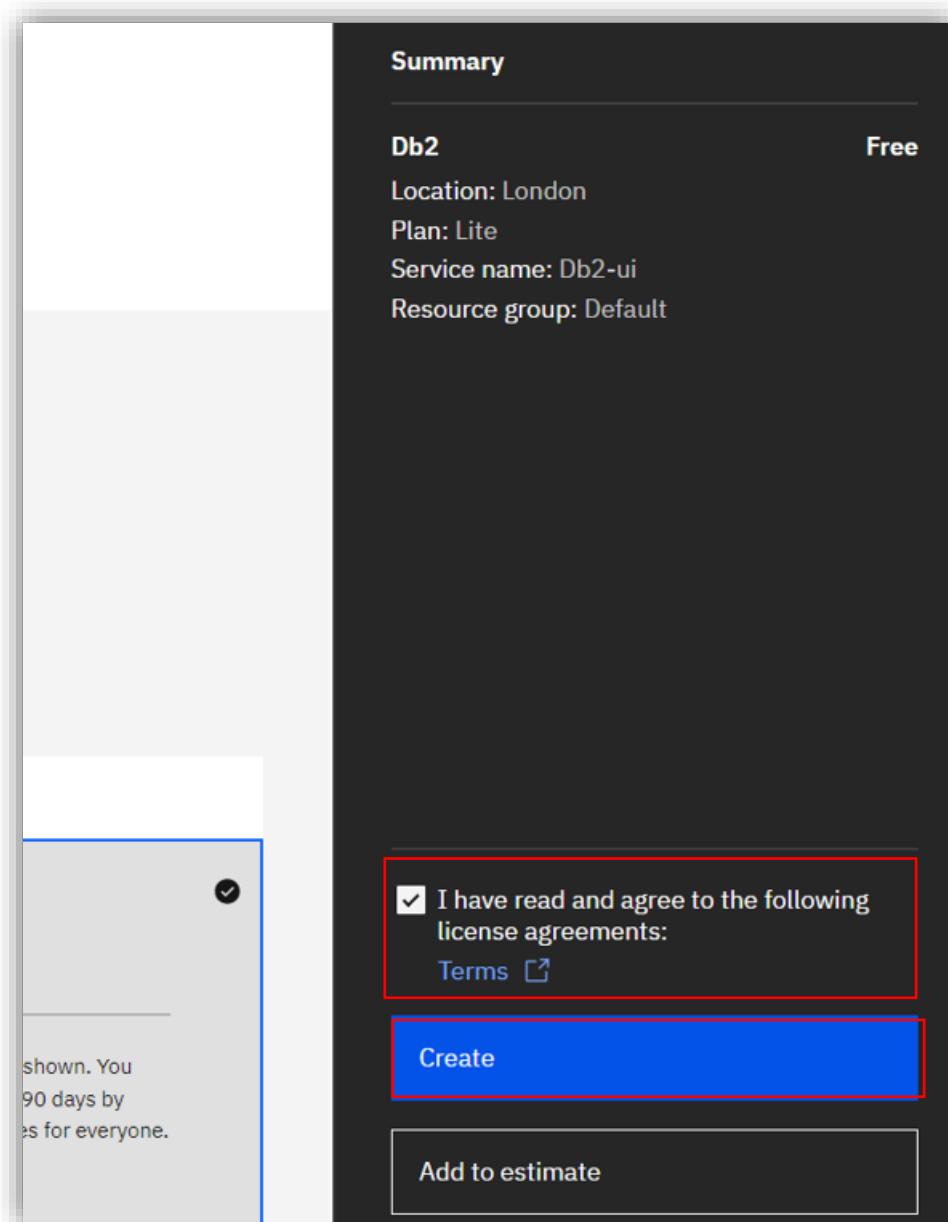
Displayed prices do not include tax. Monthly prices shown are for country or location: [United States](#)

Plan	Features and capabilities	Pricing
Lite	200 MB of data storage 5 simultaneous connections Shared multitenant system	Free <input checked="" type="checkbox"/>

The Free plan provides a free Db2 service for development and evaluation. The plan has a set amount of limitations as shown. You can continue using the free plan for as long as needed, however, users are asked to re-extend their free account every 90 days by email. If you do not re-extend, your free account is cleaned out a further 90 days later. This helps provide free resources for everyone.

Lite plan services are deleted after 30 days of inactivity.

- Tick the box to agree the terms then click **create** to create the service.



Db2

A next generation SQL database. Formerly dashDB For Transactions.

Create About

Select a location

London (eu-gb)

Service created

This service might take additional time to be created. You're being redirected to the resource list where you can see all of the resources in your account.

Select a pricing plan

Displayed prices do not include taxes or fees.

Plan	Features and capabilities	Pricing
Lite	200 MB of data storage 5 simultaneous connections Shared multitenant system	Free

The Free plan provides a free Db2 service for development and evaluation. The plan has a set amount of limitations as shown. You can upgrade at any time.

Summary

Db2

Location: London

Plan: Lite

Service name: Db2

Resource group: Default

I have read the license agreement and accept it.

Creating...

- Then the service created message will be shown.
- Go to resource list to check if the service is created or not.

Resource list

Create resource +

Name	Group	Location	Product	Status	Tags
IBM Cognos Dashboard Embedded-q2	Default	London	IBM Cognos Dashboard Embedde...	Active	-
Db2-ui	Default	London	Db2	Provision in progress...	-

The next step is to create the service credentials.

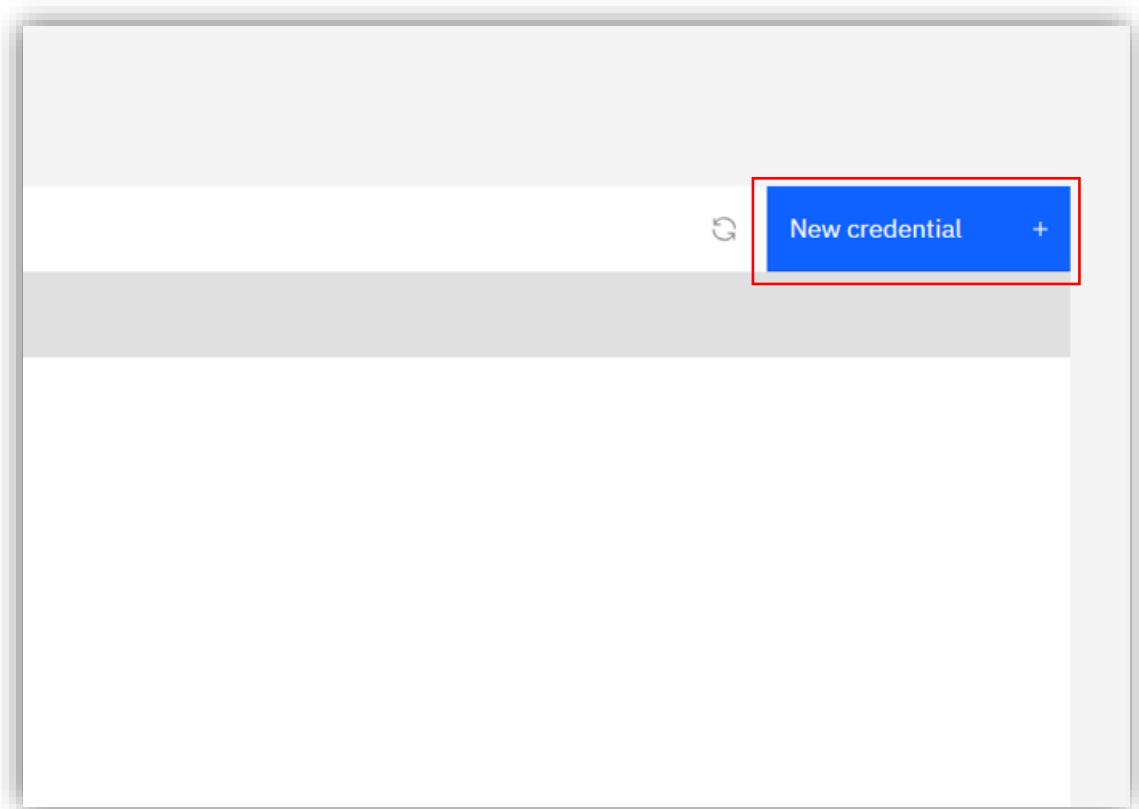
- On that click the **Db2** service.

This screenshot shows the 'Service details' page for a Db2 service named 'Db2-ui'. The top navigation bar includes 'Resource list /', the service name 'Db2-ui' with an 'Active' status and a 'Add tags' button, and a 'Manage' button. Below the navigation is a sidebar with 'Getting started', 'Service credentials' (which is highlighted with a red box), and 'Connections'. The main content area is titled 'Getting started' and contains instructions on how to find credentials by clicking 'Service Credentials' in the sidebar. It also mentions that users can click 'Manage in IBM Cloud' to open the dashboard if they don't see the menu. At the bottom are 'Go to UI' and 'Getting started docs' buttons, and a 'Need help?' sidebar with a 'Support case' button.

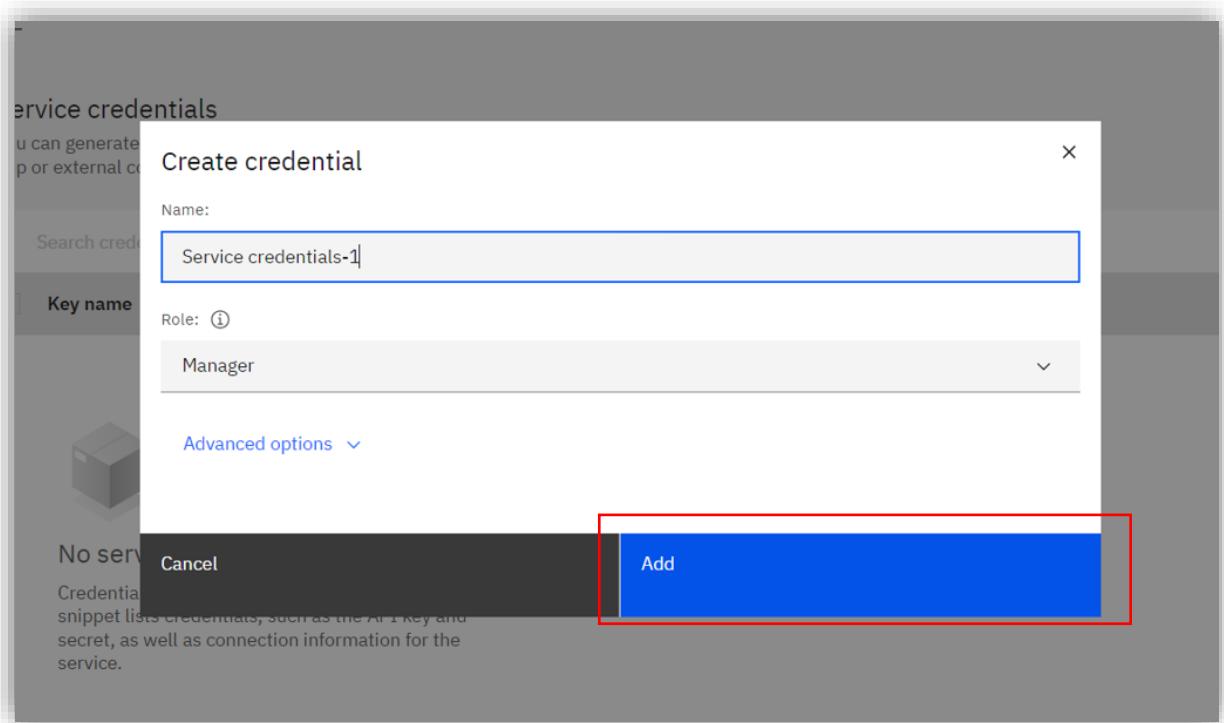
- Then click the **Service credentials**.

This screenshot shows the 'Service credentials' page for the 'Db2-ui' service. The top navigation bar includes 'Resource list /', the service name 'Db2-ui' with an 'Active' status and a 'Add tags' button, and 'Details' and 'Actions...' buttons. The sidebar on the left has 'Manage', 'Getting started', 'Service credentials' (which is highlighted with a blue box), and 'Connections'. The main content area is titled 'Service credentials' and explains that users can generate new credentials for manual connection. It features a search bar 'Search credentials...', a table header with 'Key name' and 'Date created' columns, and a note stating 'No service credentials'. A small note at the bottom says 'Credentials are provided in JSON format. The JSON'.

- Click **New credential** to create a new service.

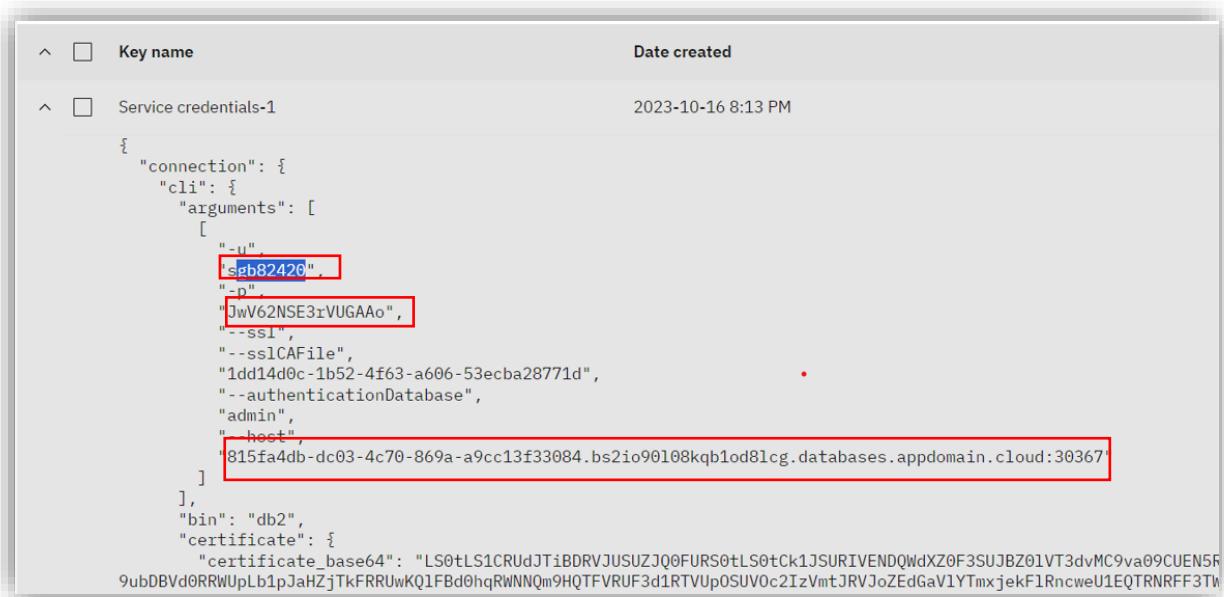


- Name that service and select the role.

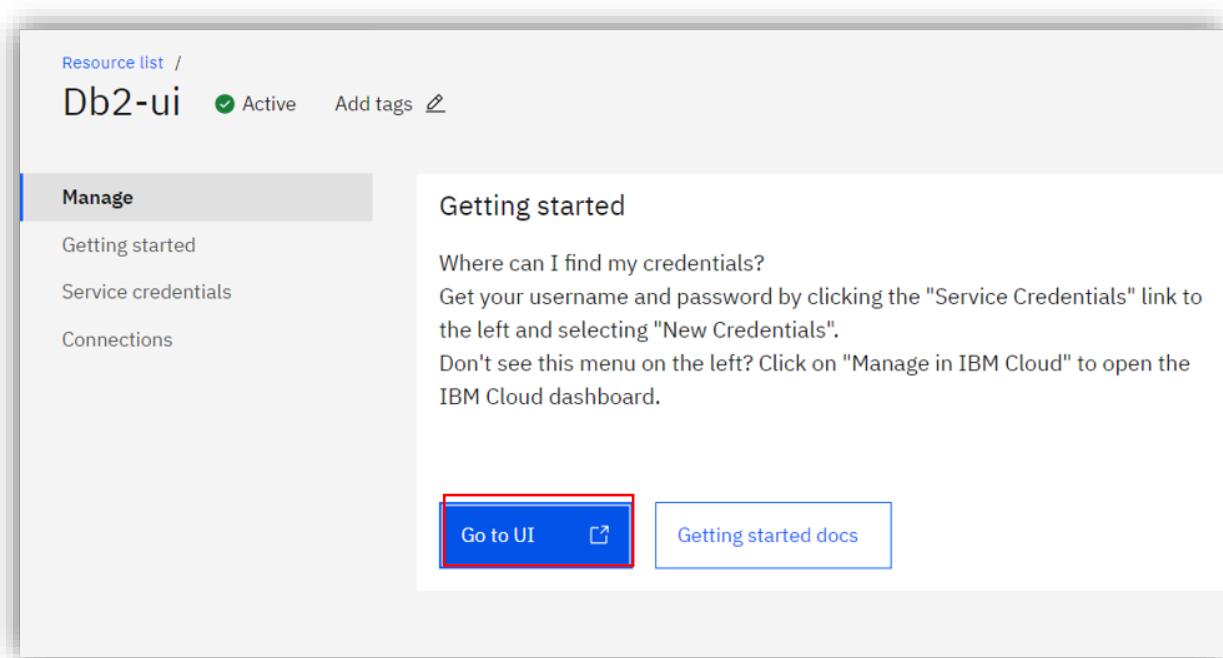


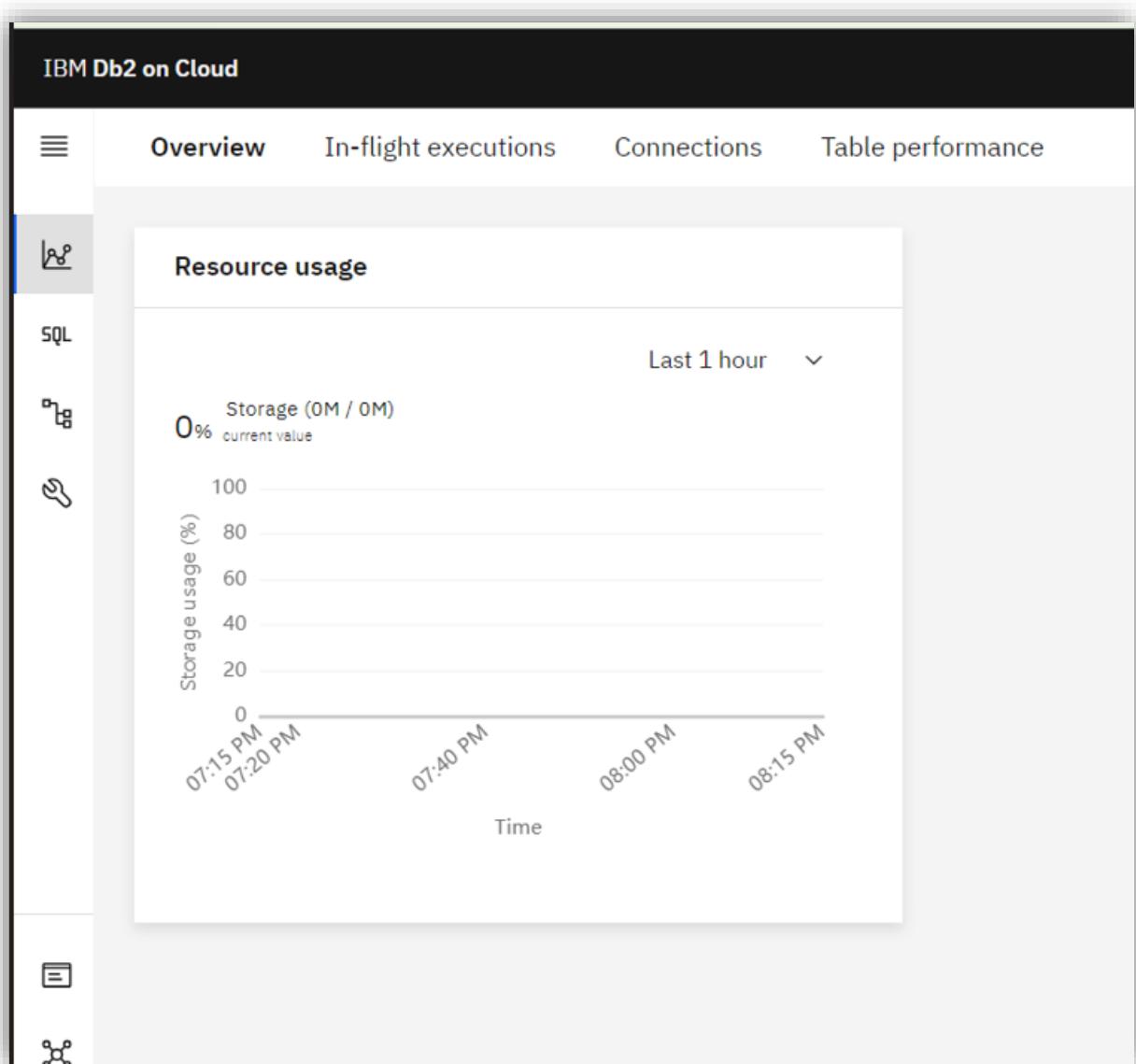
- Finally click **Add**.

- Save the **username, password** and **host** for future use.



- Then go to manage and click [Go to UI](#) button.

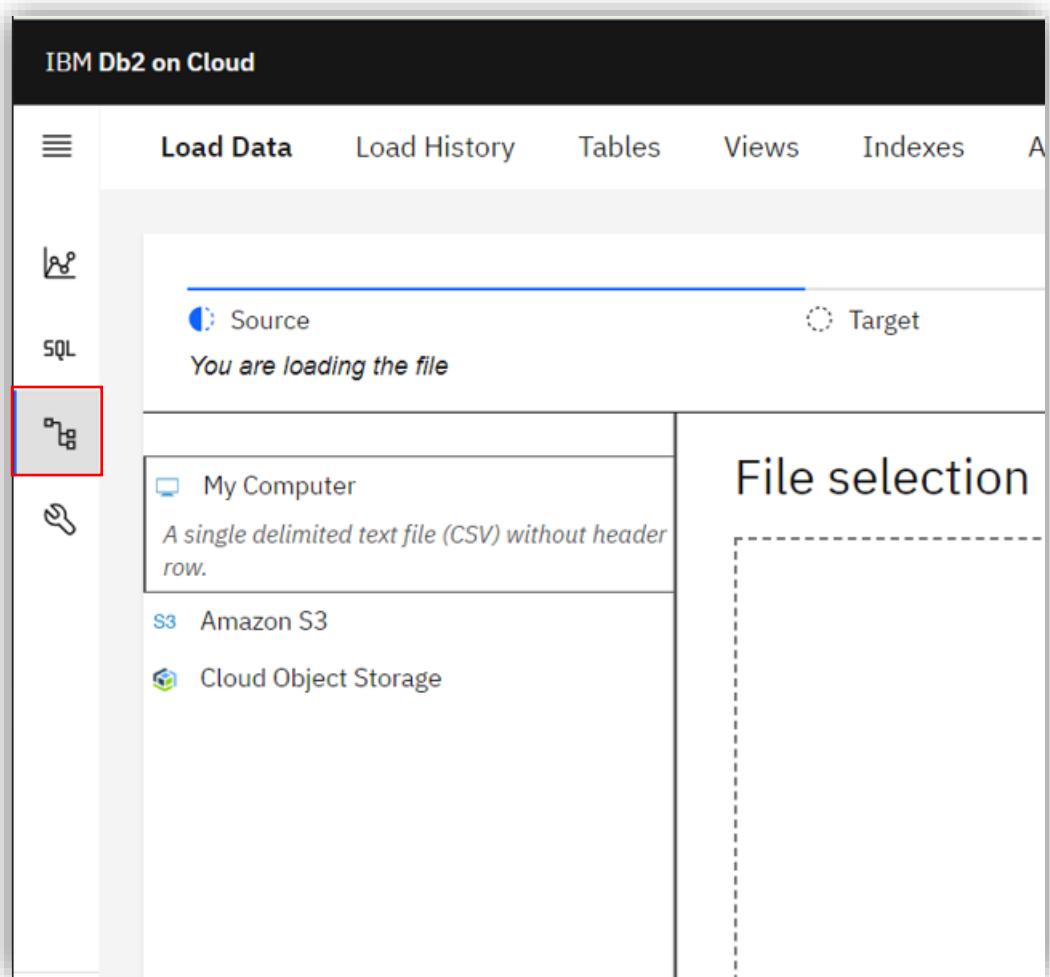




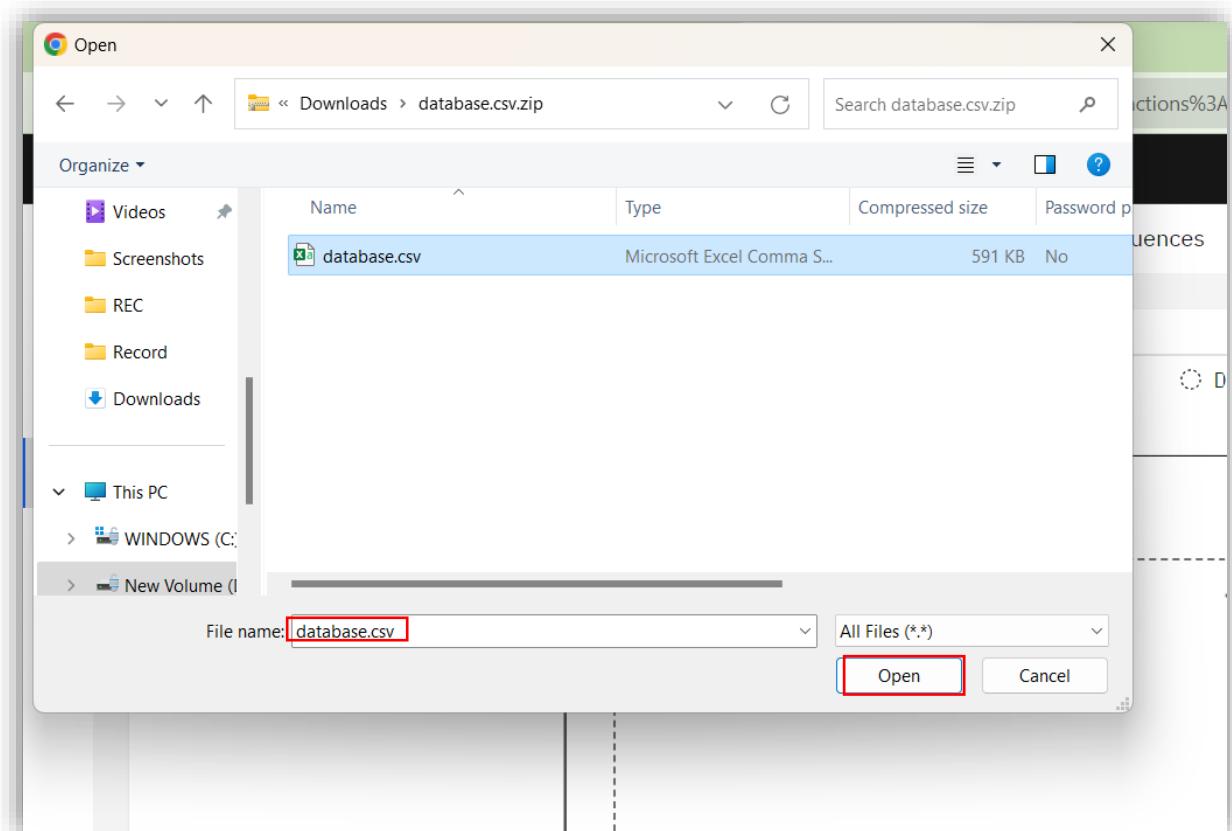
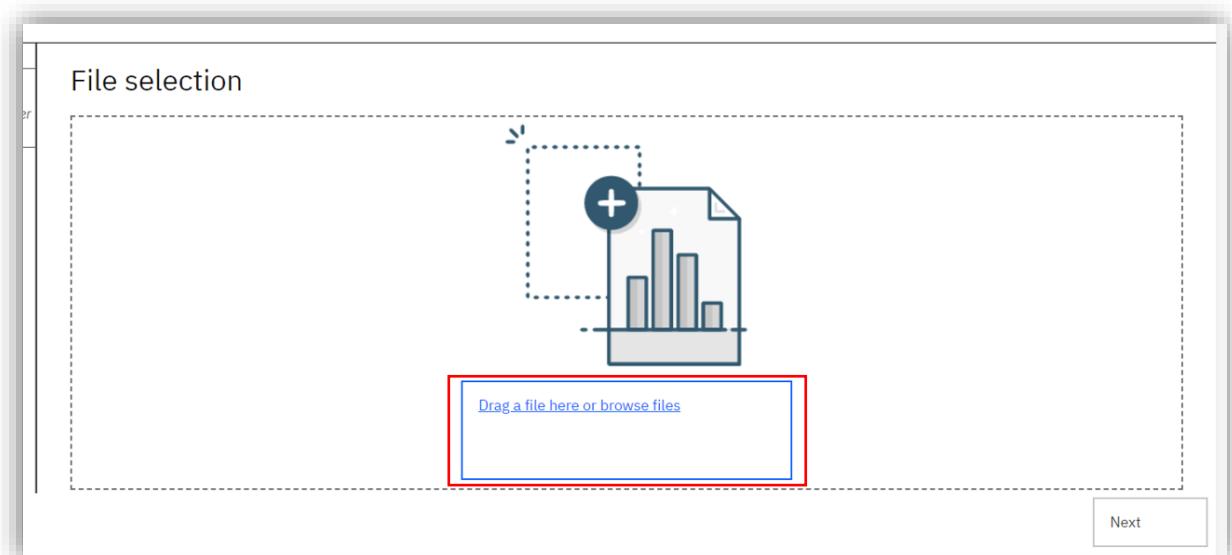
- Then the UI will be displayed.

The final step is to load the dataset into the IBM cloud.

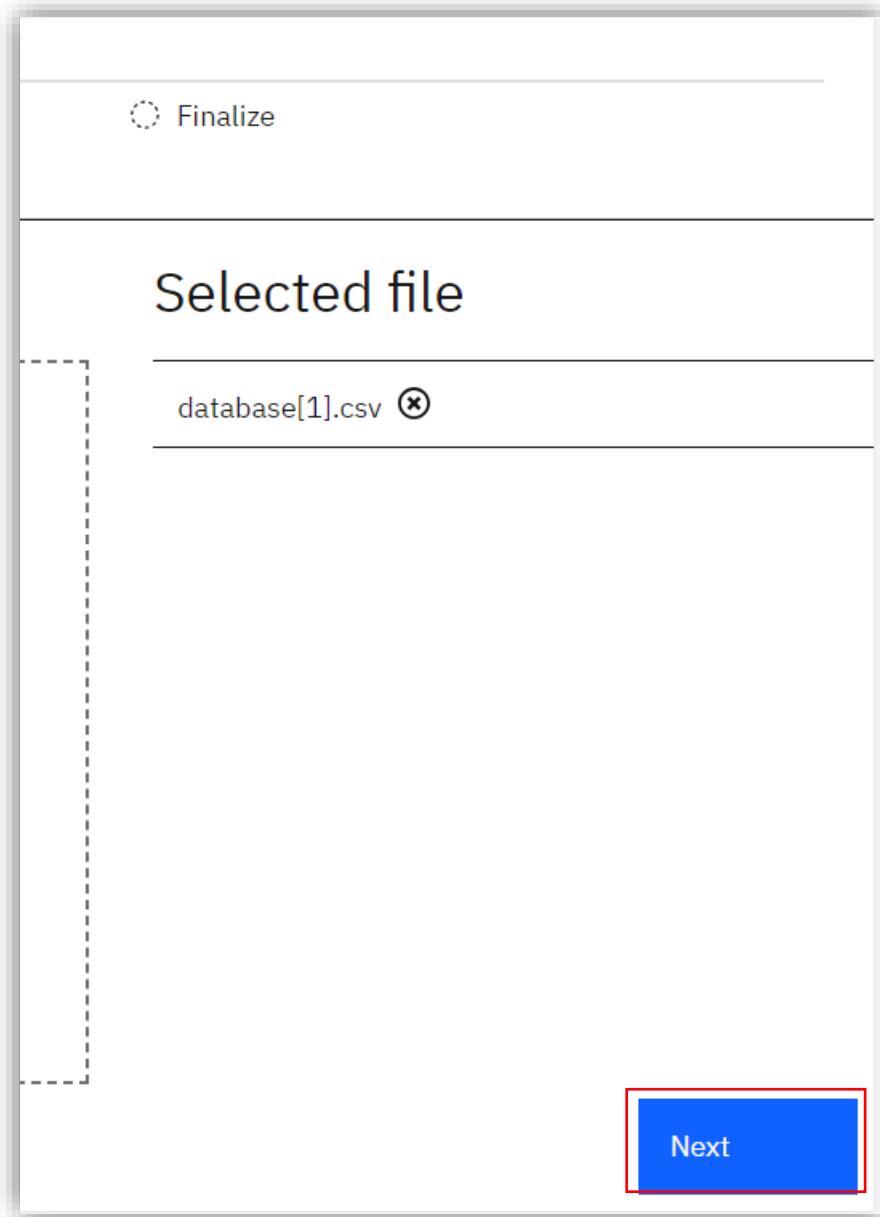
- Move to the third tab as shown in the figure below.



- Select the necessary file for further processing.



➤ Click **Next**.



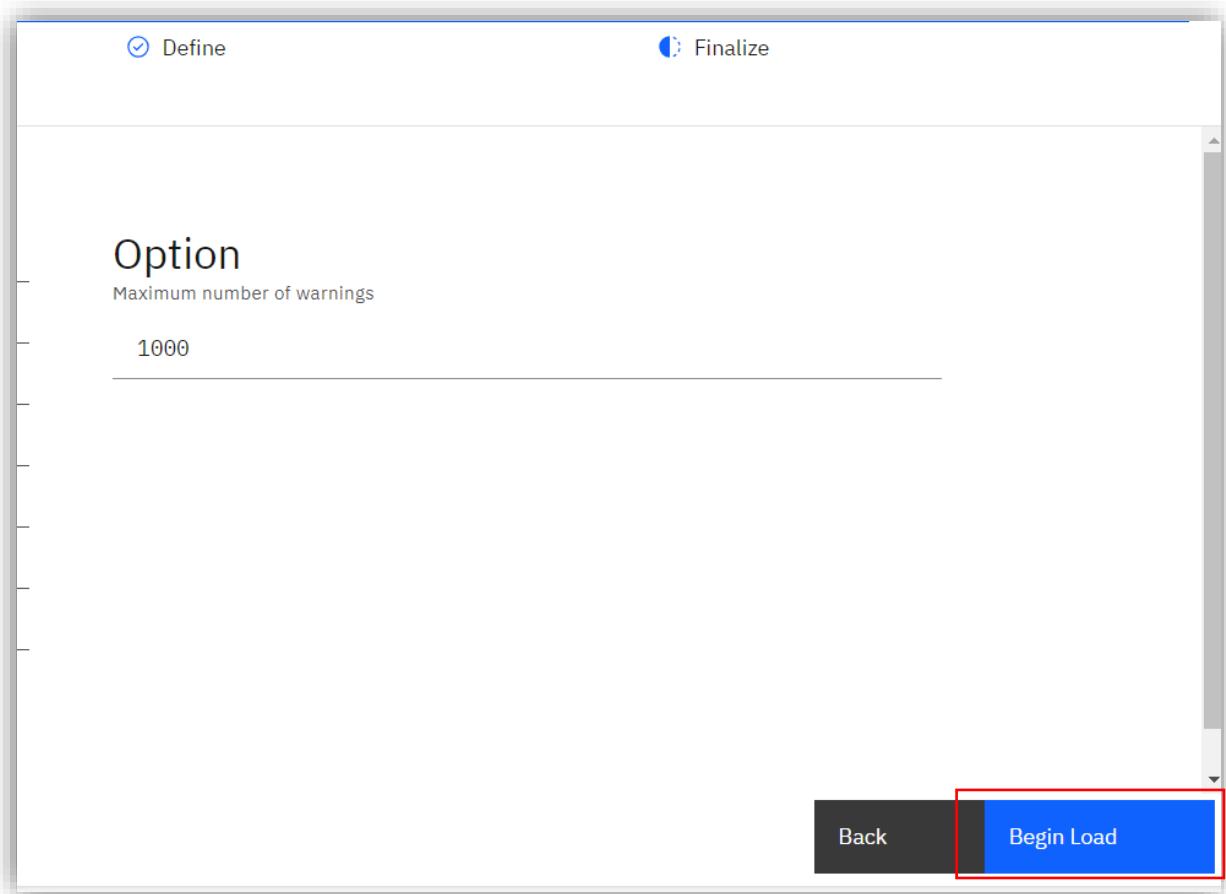
- Select the created schema.

The screenshot shows a user interface titled "Select a load target". In the "Schema" section, there is a search bar labeled "Find schemas" with a magnifying glass icon. Below the search bar is a list of schemas. The schema "SGB82420" is highlighted with a red rectangular border. The background of the interface has a light gray gradient.

- Create table and click **Next**.

The screenshot shows a user interface titled "Table". At the top right, there is a "Refresh" button with a circular arrow icon and a "New table +" button. Below the title, there is a search bar labeled "Find tables in SGB82420" with a magnifying glass icon. A list of tables is displayed, with "DATABASE1" selected and highlighted by a red rectangular border. To the right of "DATABASE1" is a blue checkmark icon inside a red-bordered box. Below the table list, there are other entries: "DATABASE" and "SAMPLE-SPREADSHEET-10000-ROWS". At the bottom of the interface, there are two buttons: "Back" and "Next". The "Next" button is highlighted with a red rectangular border.

- Then click **Begin Load** button.



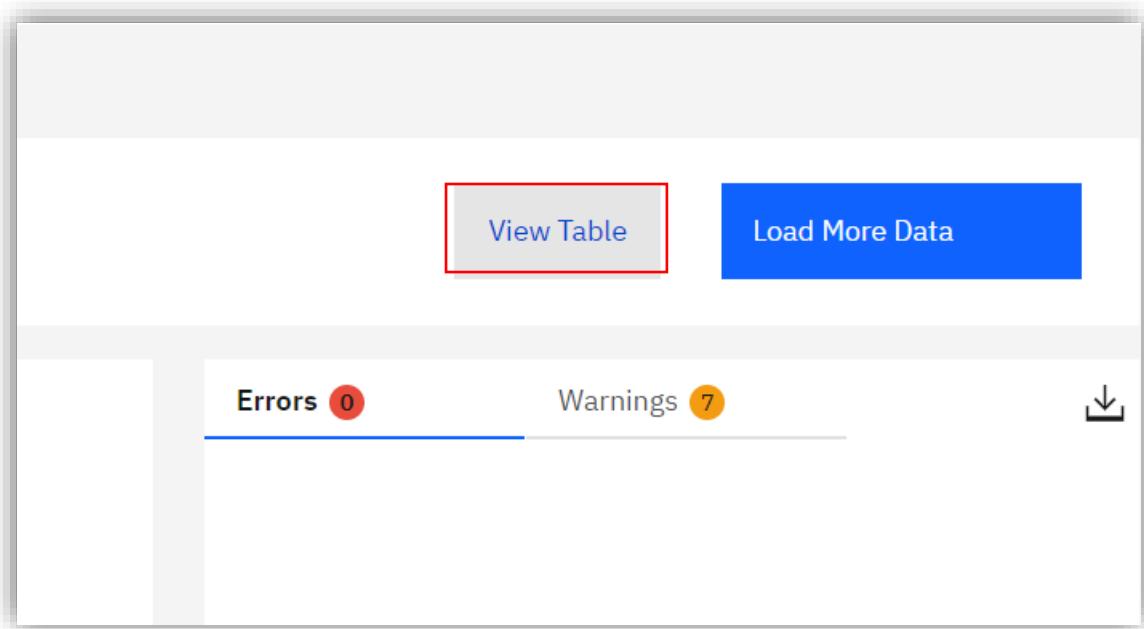
- Then the dataset will be finally loaded.

The screenshot shows the 'Load Data' dashboard for IBM Db2 on Cloud. The top navigation bar includes 'Load Data', 'Load History', 'Tables', 'Views', 'Indexes', 'Aliases', 'MQTs', 'Sequences', 'Application objects', 'Notifications' (with 1 notification), and 'Clear all'. The 'Notifications' section displays a message: 'Load completed with minor error' (Load database[1].csv from My Computer to SGB82420.DATABASE1) with a timestamp of '2023/10/16, 11:15 PM' and a 'View details' link. The main area is titled 'Load details' and shows a summary table:

SQL	My computer	Target database[1].csv	SGB82420.DATABASE1
-----	-------------	------------------------	--------------------

Below the table, there are two tabs: 'Status' (selected) and 'Settings'. The 'Status' tab displays a large blue circular progress indicator. Below the indicator, the statistics are shown: '23,412 Rows read', '23,412 Rows loaded', and '0 Rows rejected'. At the bottom of the status section, it says 'Start time: 10/16/2023 11:15:39 PM' and 'End time: 10/16/2023 11:15:44 PM'. To the right of the status section, a message states 'The data load job succeeded.' and 'You can now work with your data.' In the bottom right corner of the dashboard, it says 'No errors' and 'But, there are 7 warnings.'

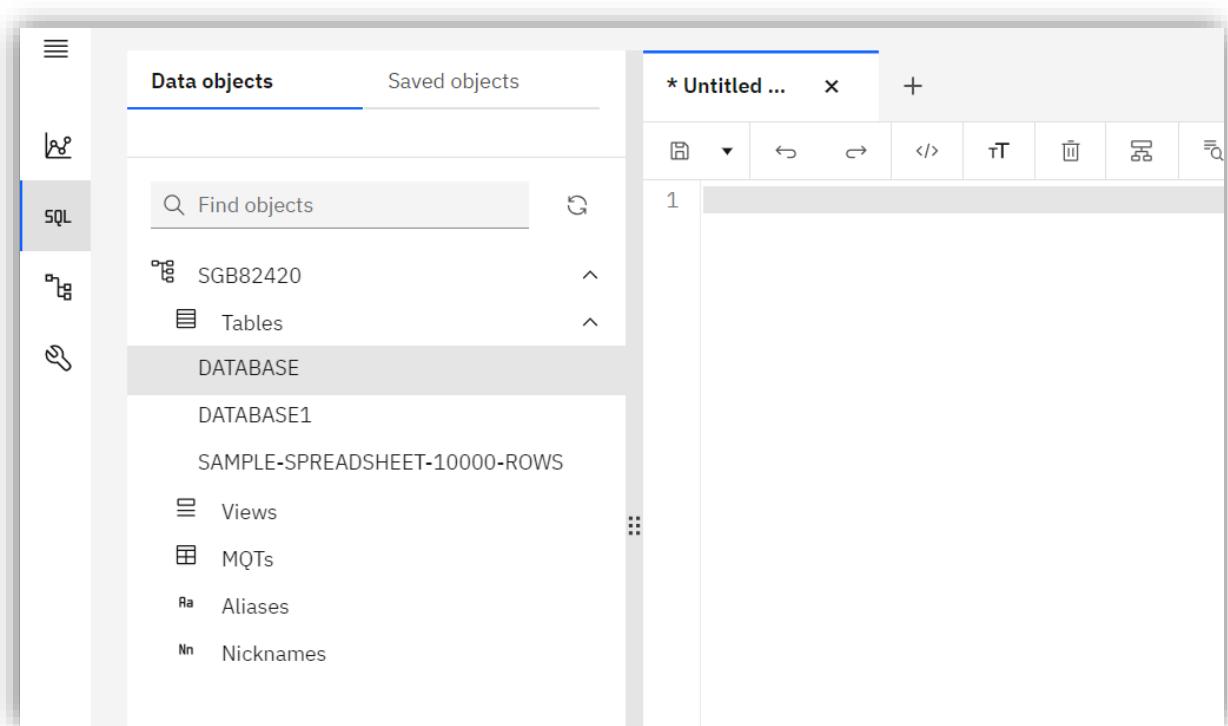
- We can view the loaded table by clicking the [View Table](#) button.



- Atlast the dataset is loaded.

SGB82420.DATABASE1								
	DATE VARCHAR(24)	TIME VARCHAR(24)	LATITUDE DECIMAL(16, 7)	LONGITUDE DECIMAL(17, 7)	TYPE VARCHAR(17)	DEPTH DECIMAL(7, 3)	DEPTH_ERROR DECIMAL(7, 3)	DEPTH_SEISMI. SMALLINT
1	01/01/1967	07:05:51	-15.2370000	-173.6080000	Earthquake	30.000		
2	01/01/1967	21:59:00	-11.2000000	165.4160000	Earthquake	30.000		
3	01/01/1969	09:07:06	51.0960000	-179.3920000	Earthquake	45.000		
4	01/01/1970	17:11:00	-29.4000000	-177.1690000	Earthquake	35.000		
5	01/01/1971	07:58:03	-4.1900000	141.1830000	Earthquake	35.000		
6	01/01/1972	22:05:55	-17.0210000	174.9030000	Earthquake	10.000		
7	01/01/1973	11:42:37	-35.5130000	-16.2110000	Earthquake	33.000		

- Using the SQL tab we can query the loaded dataset.

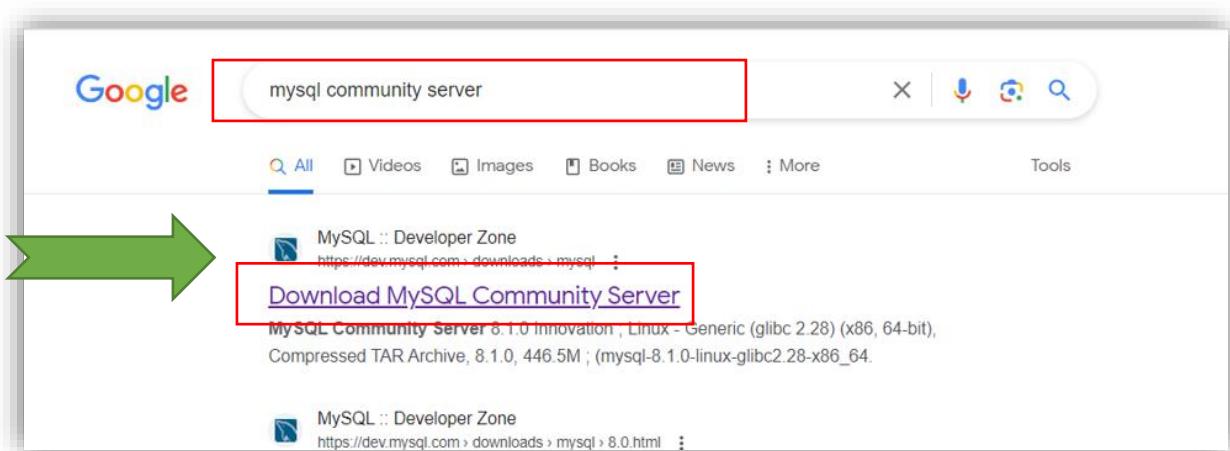


Phase 4

The IBM DB2 lite version, as of phase 2, only allows us to import data, perform ETL procedures, and query datasets. With these constraints, we are unable to perform additional operations such as adding users and granting database rights so that they can explore and analyze our data for business insights.

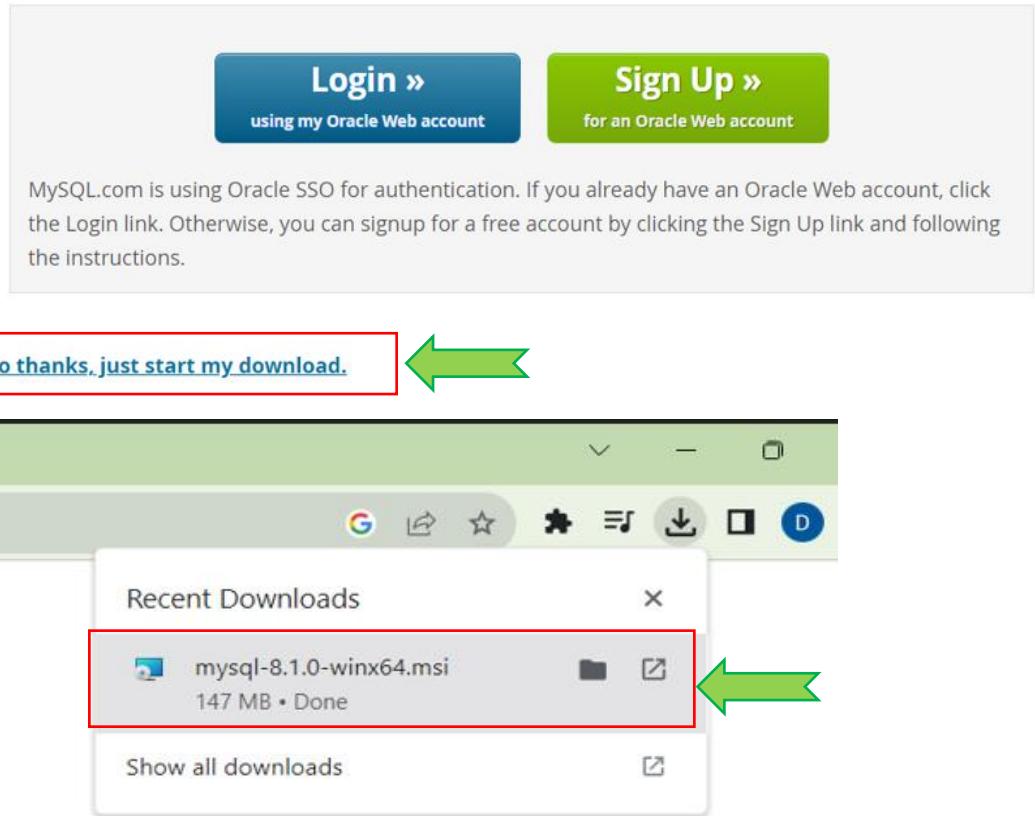
So we downloaded and installed MySQL Community Server to create a server and generate a database, as well as DBeaver to run SQL queries on the datasets and create new users for the data architects.

- The first step is search mysql community server in google and choose the correct website.

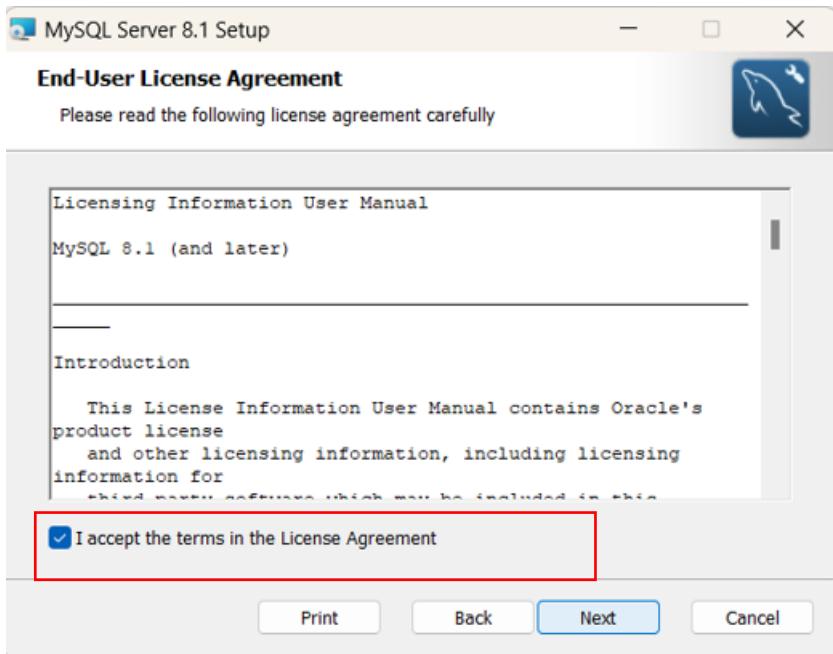


- After entering into the website click the download button of windows MSI Installer.

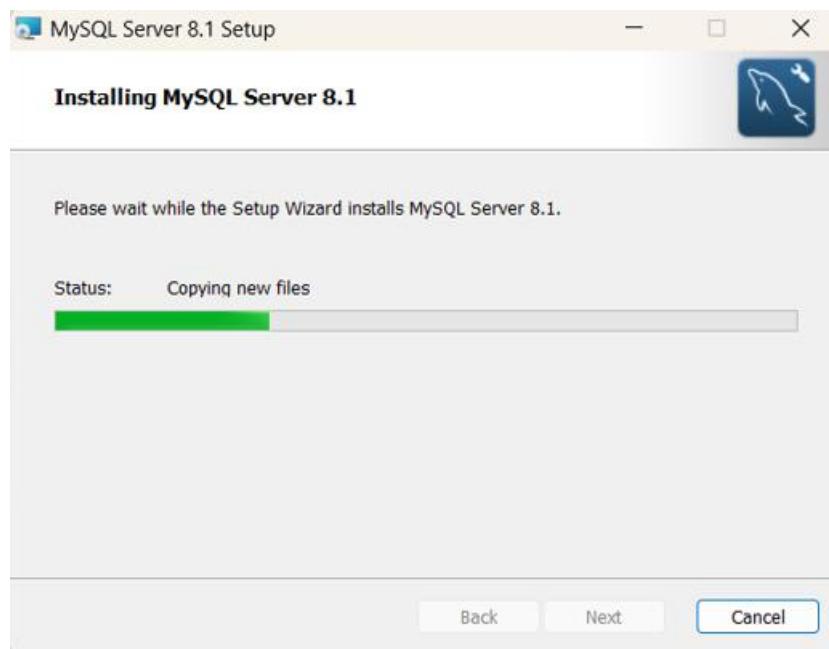
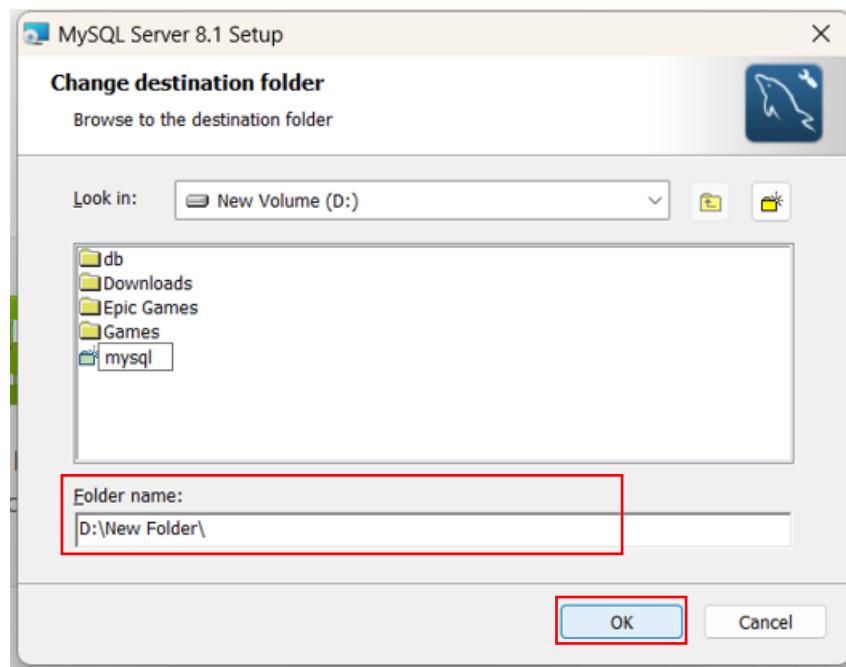
- Avoid logging in and click the no thanks button and start the download.



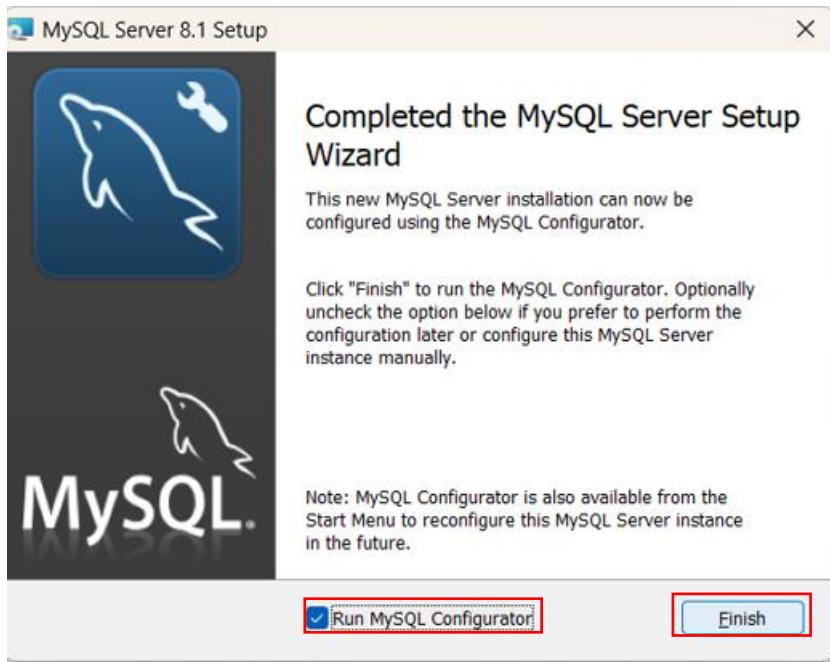
- Accept the agreement and click next.



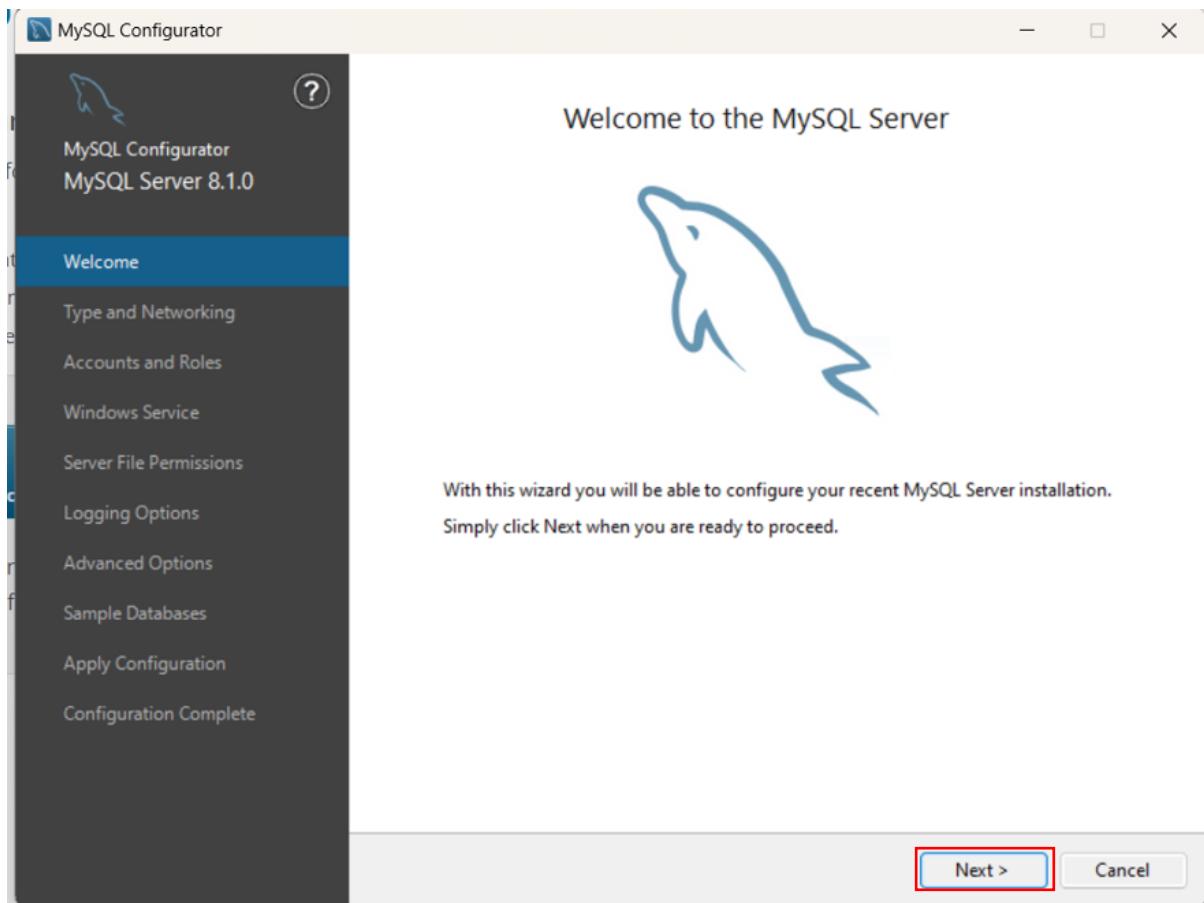
- Edit the path.



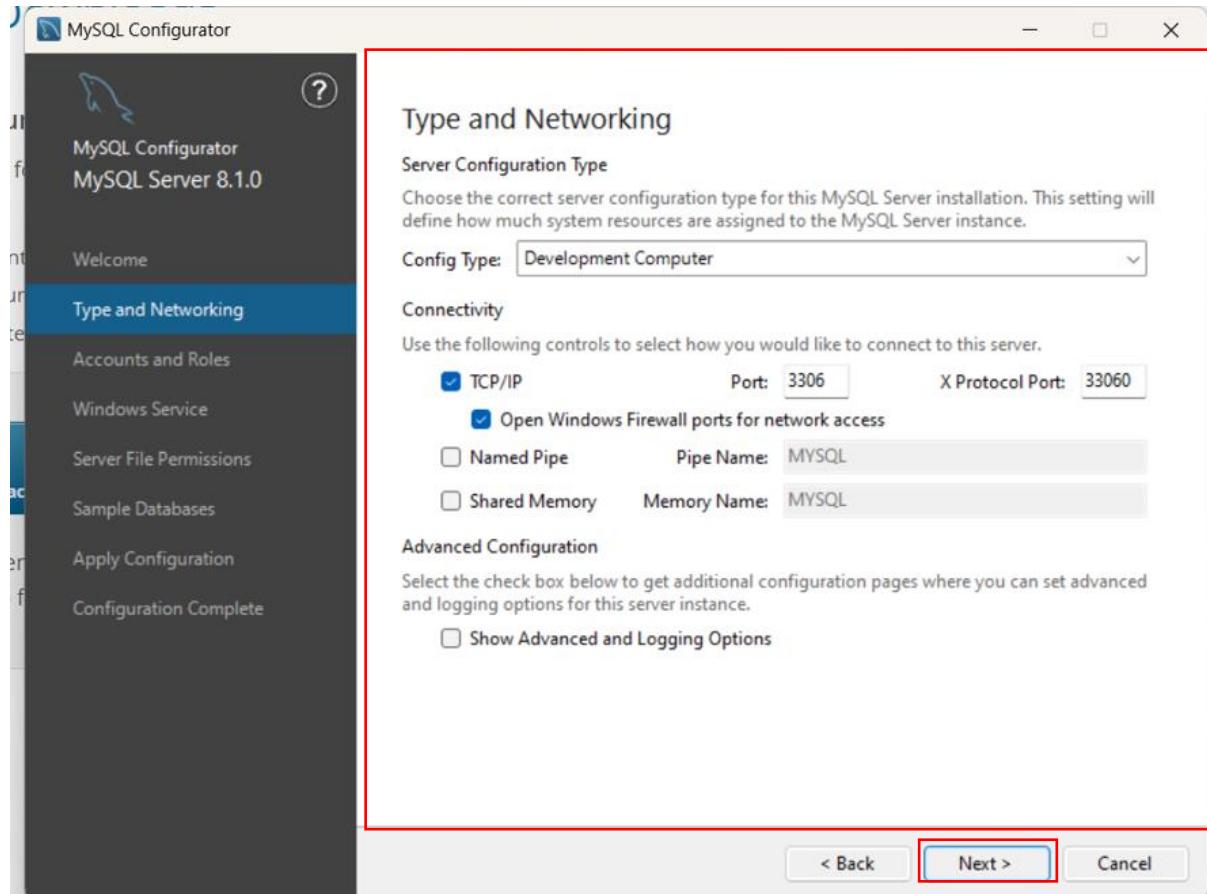
- Finish and run mysql.



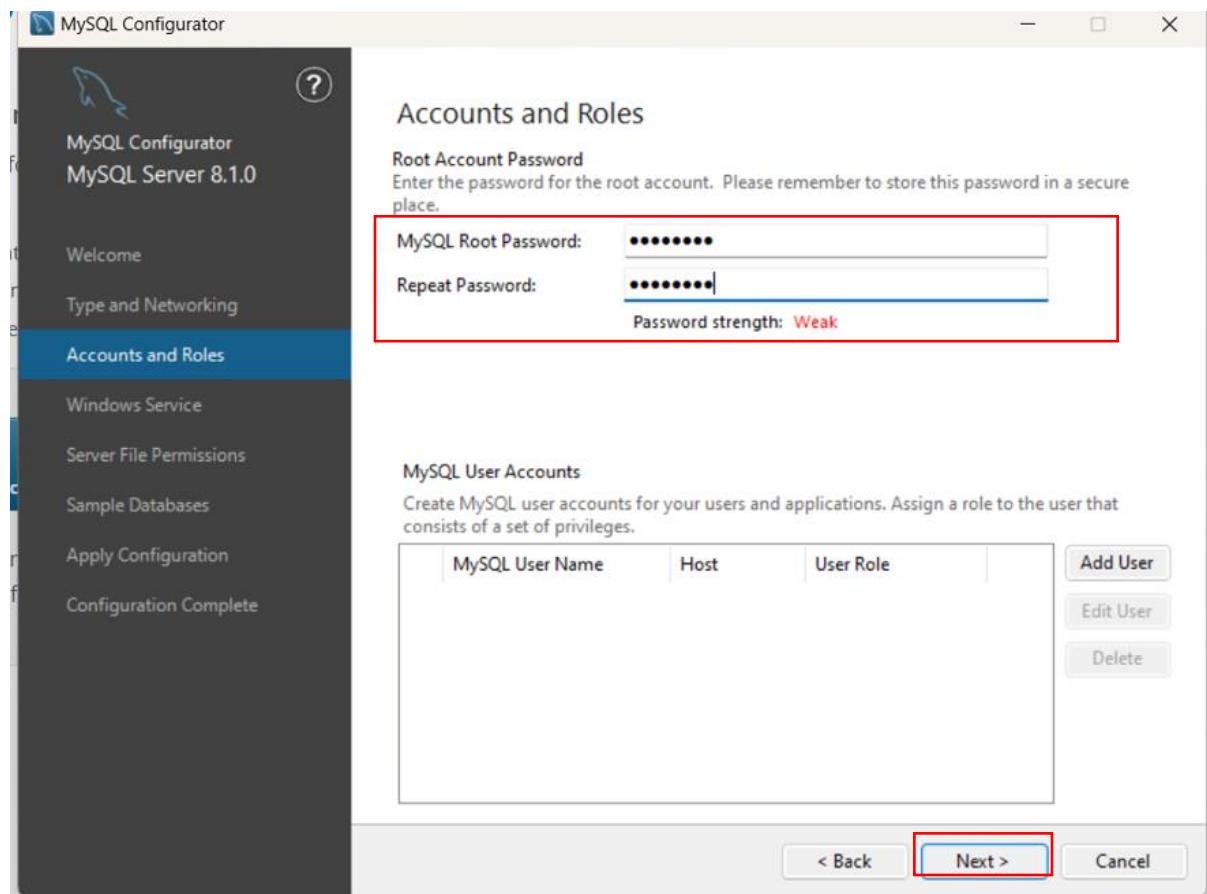
- The second step is to run the MySQL configurator.



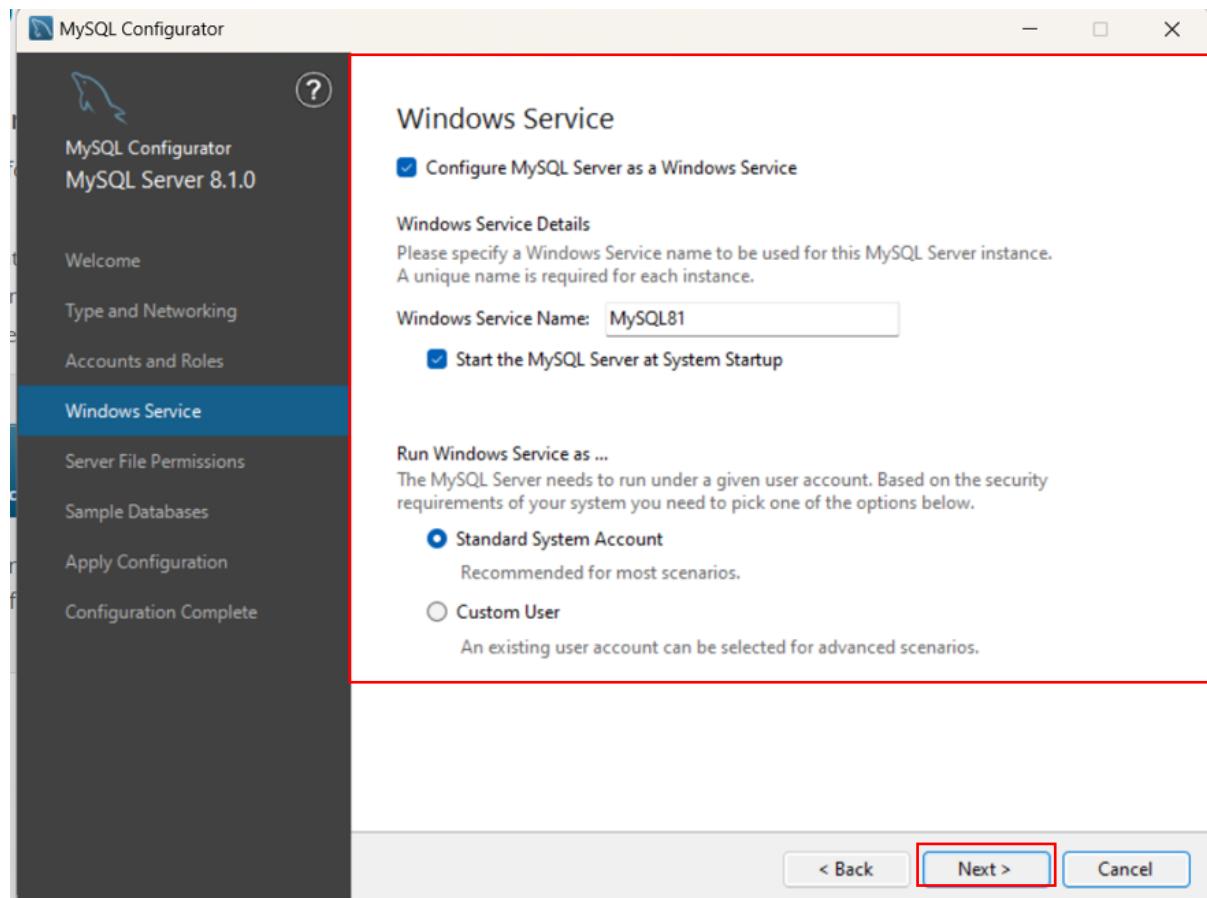
- Fill the correct details and click next.



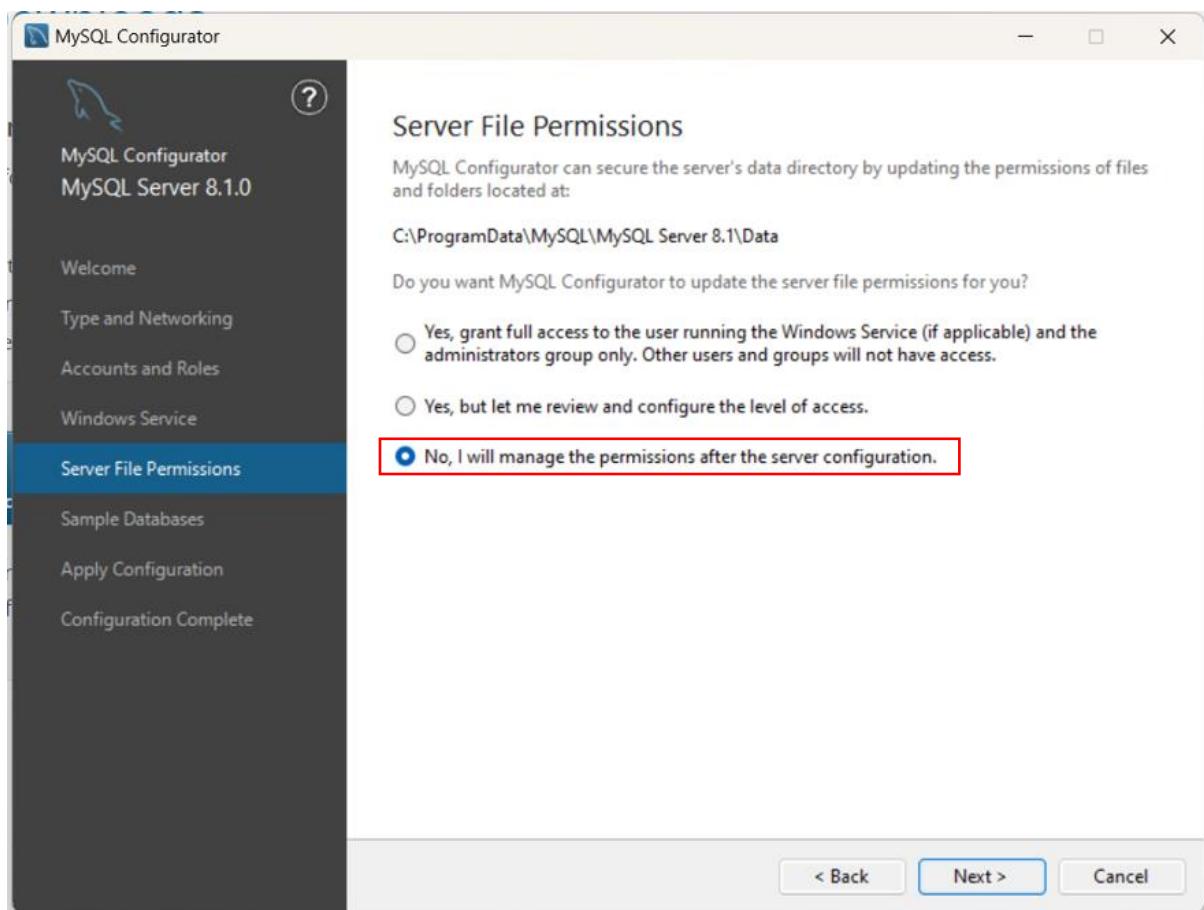
- Create a password for the database and `root` is the default username for the database.



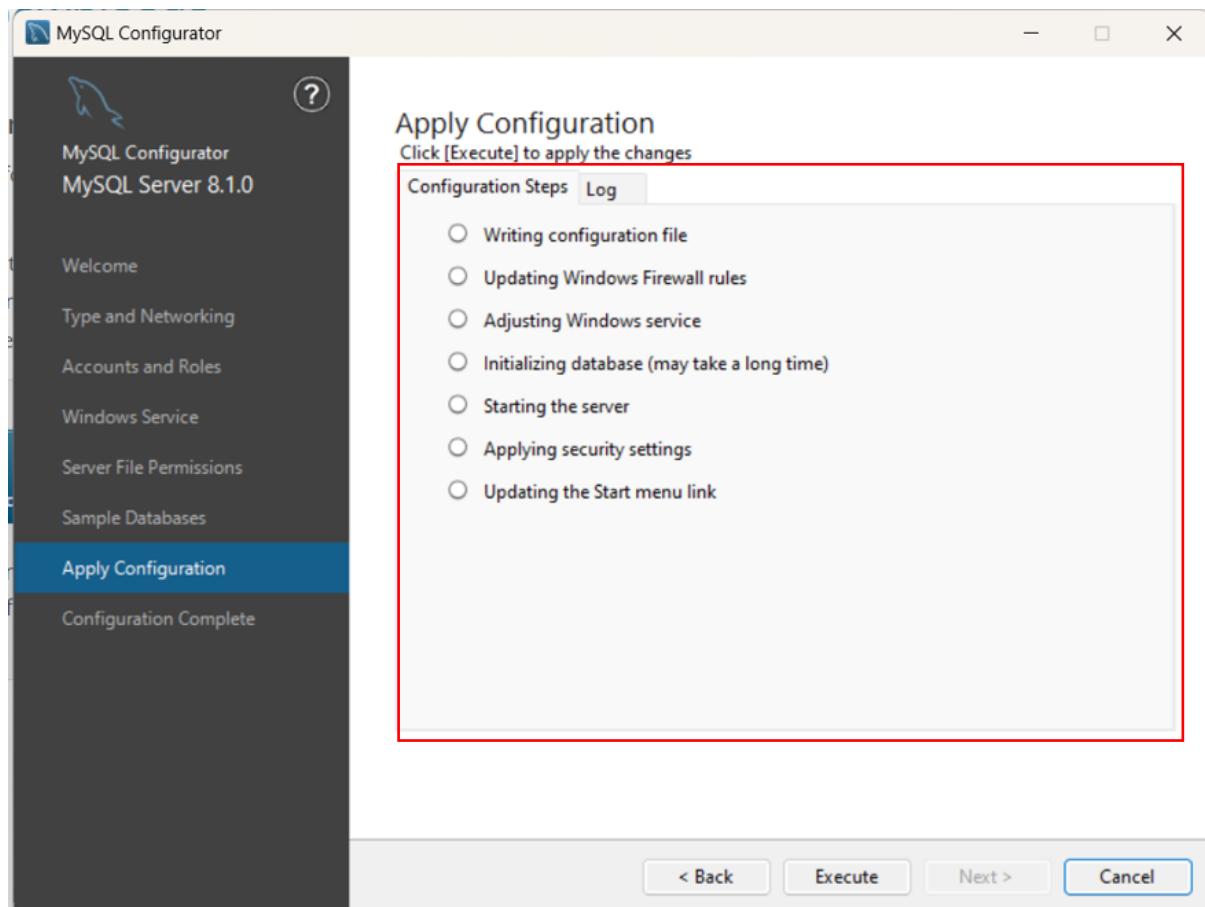
- Choose Standard System Account and click next.



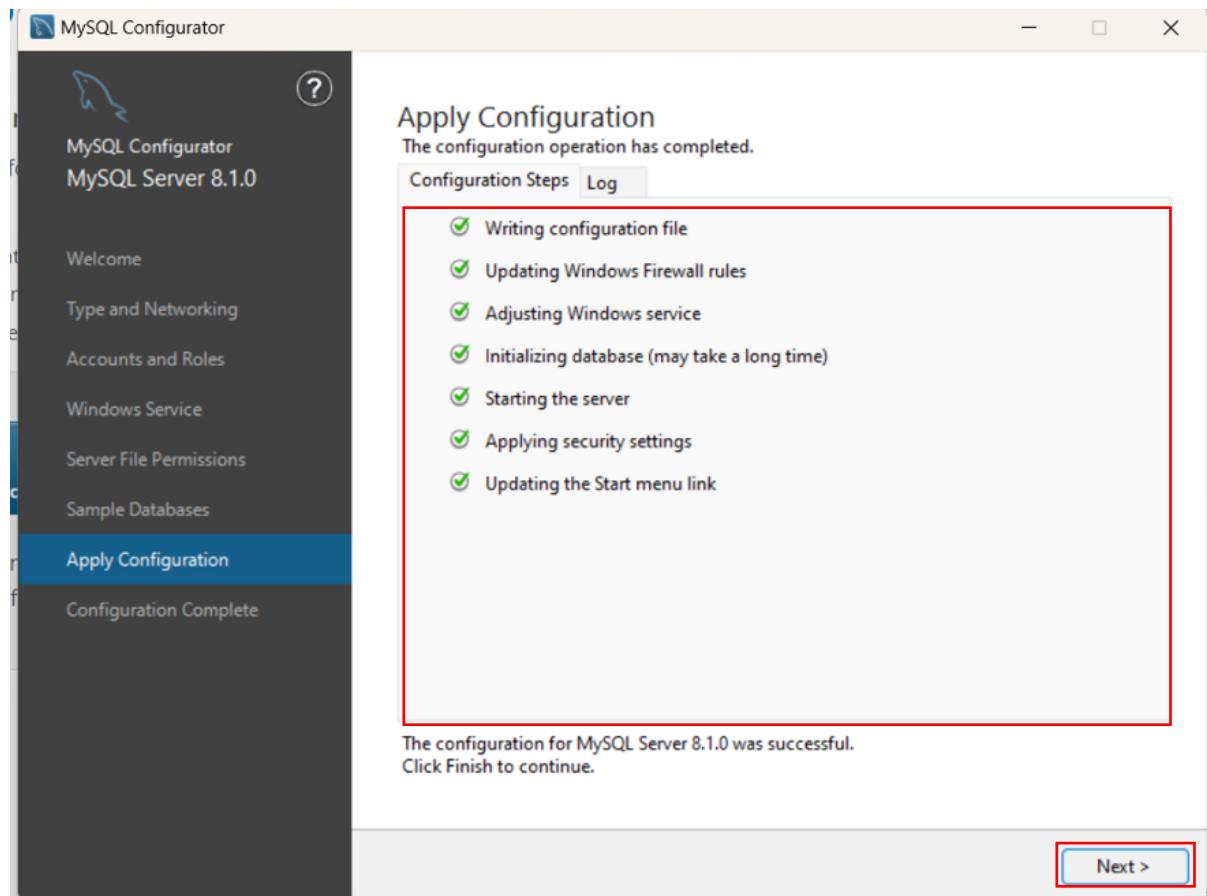
- Choose the last option so that we can manage the permissions of the users and click next.



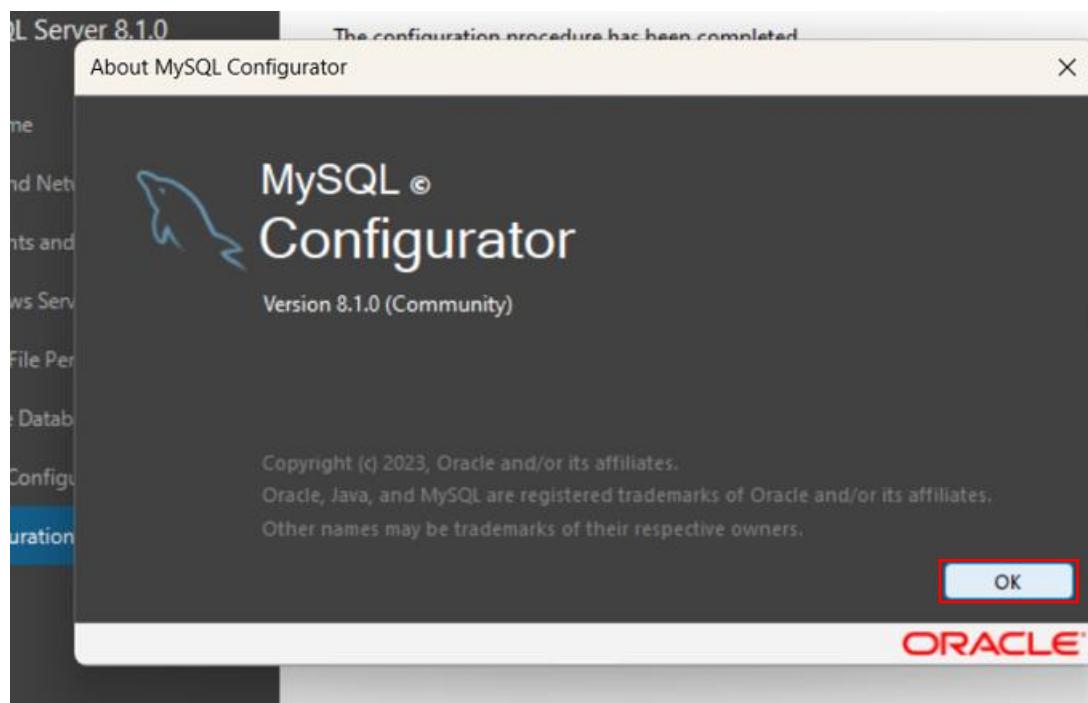
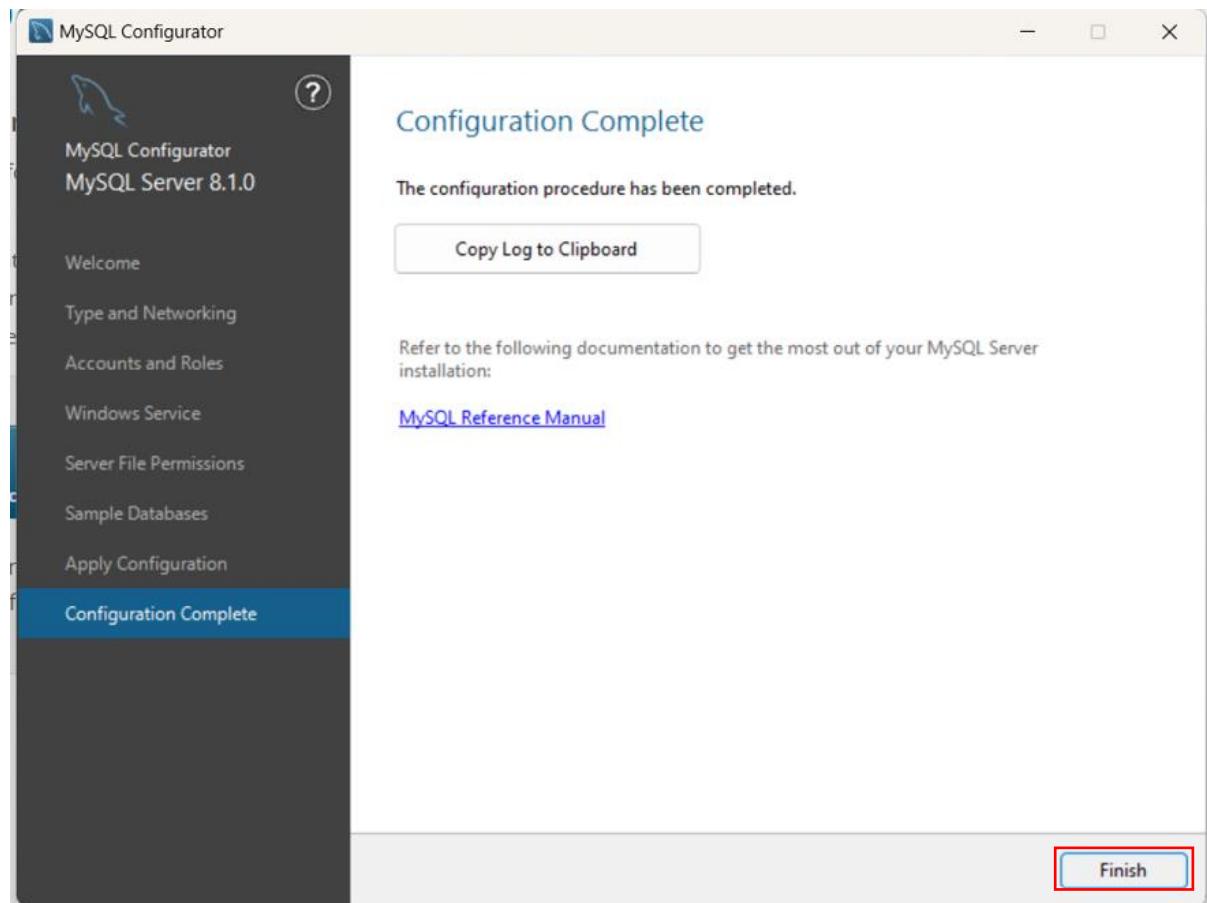
- Now apply configuration.



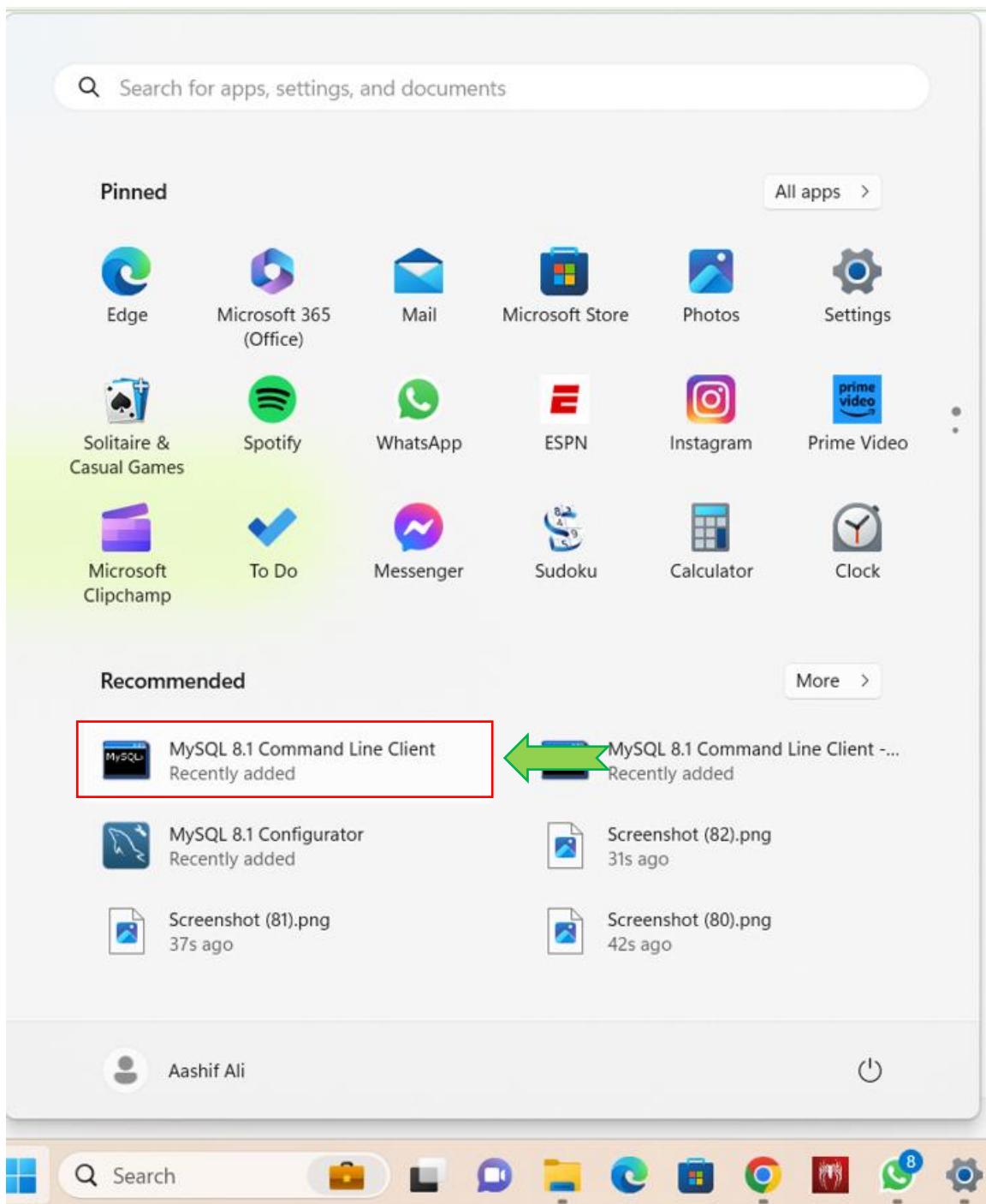
- After the configuration is finished click next.



- Now click the finish and ok buttons.

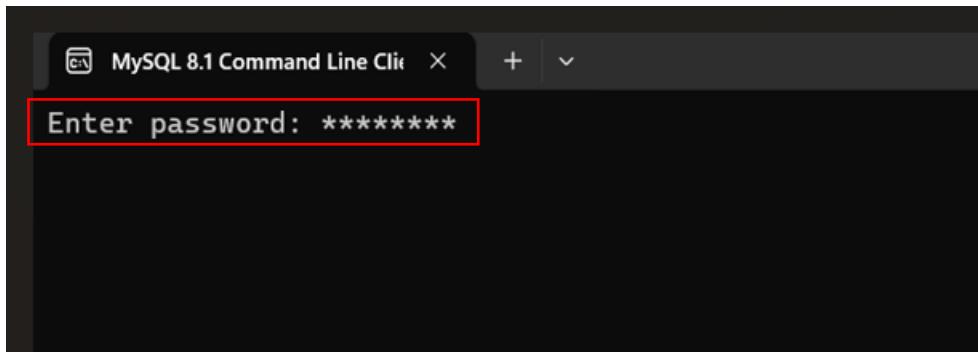


- The third step is to work with the MySQL Command Line Client.



- Open the MySQL 8.1 Command Line Client.

- Enter the password



- Now create a database called dash using queries.

```
MySQL 8.1 Command Line Cli + ▾
Enter password: *****
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.1.0 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database dash;
```

```
MySQL 8.1 Command Line Cli + ▾
Enter password: *****
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.1.0 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

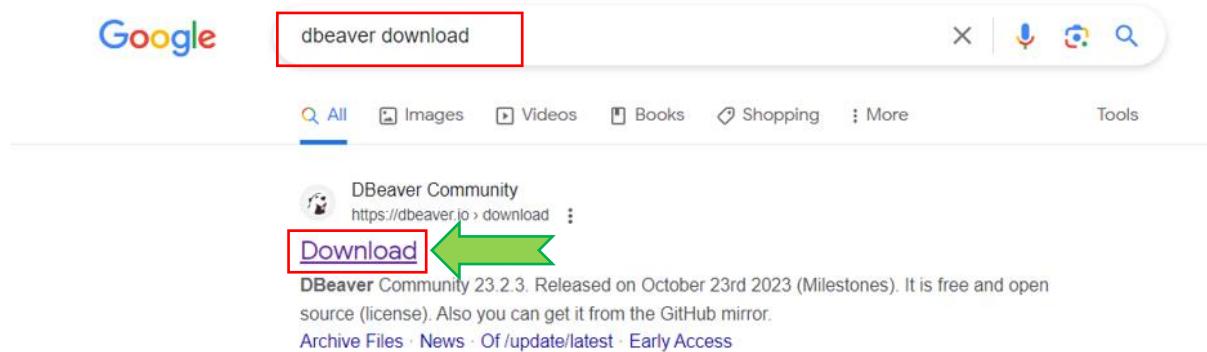
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database dash;
Query OK, 1 row affected (0.01 sec)

mysql> |
```

- The fourth step is to download the dbeaver community for querying and user related processes.



- Click Windows (installer)

Download

DBeaver Community 23.2.3

Released on October 23rd 2023 ([Milestones](#)).

It is free and open source ([license](#)).

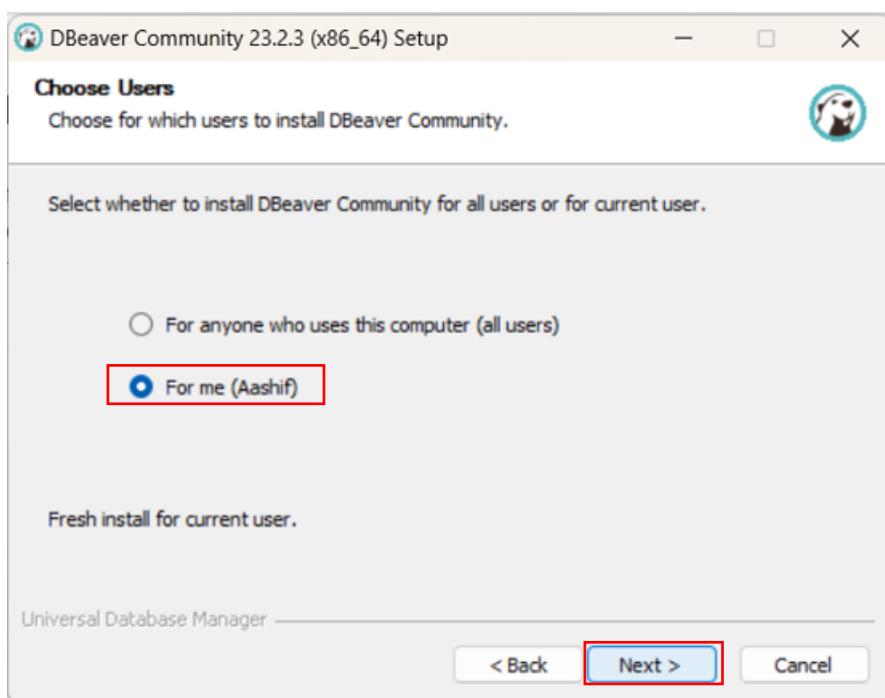
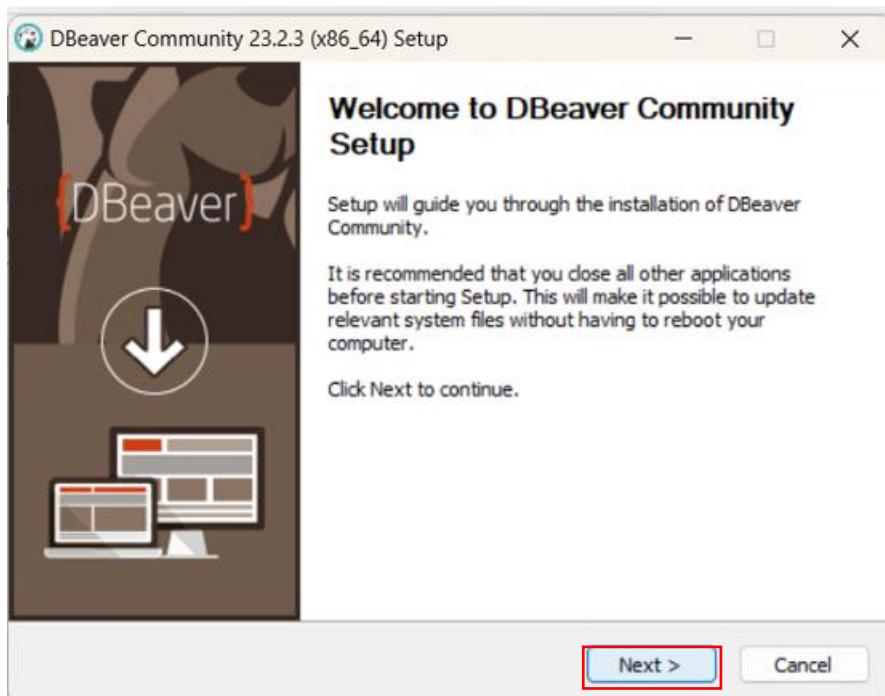
Also you can get it from the [GitHub mirror](#).

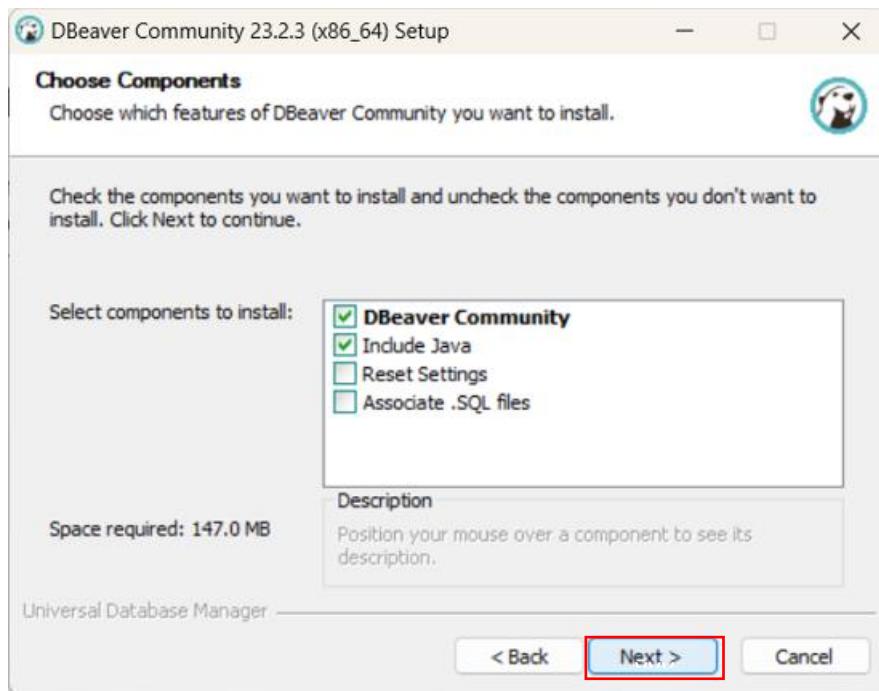
Windows

- [Windows \(installer\)](#)
- [Windows \(zip\)](#)

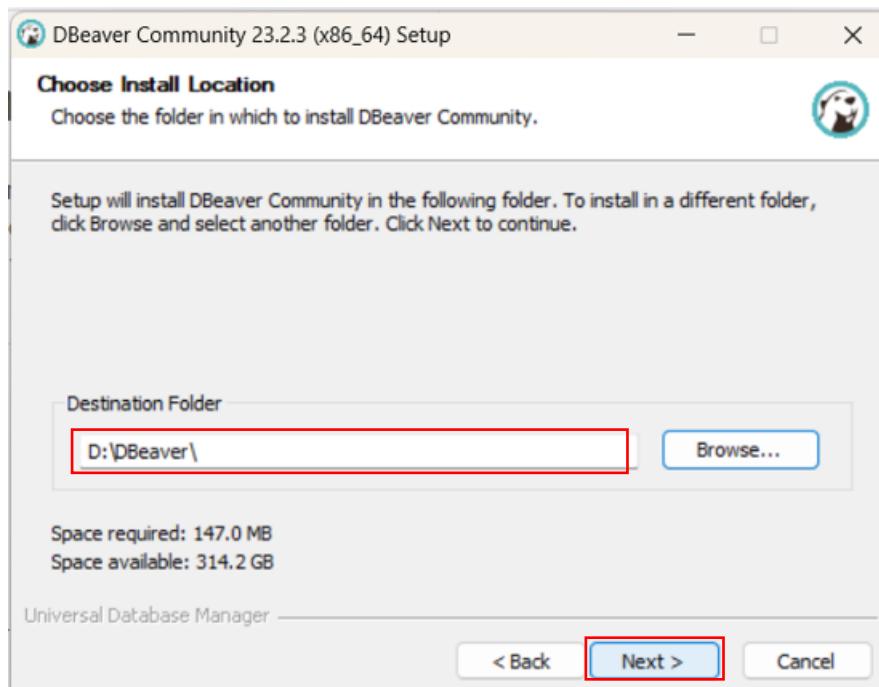
- [Chocolatey \(choco install dbeaver\)](#)
- [Install from Microsoft Store](#)

- Click next.

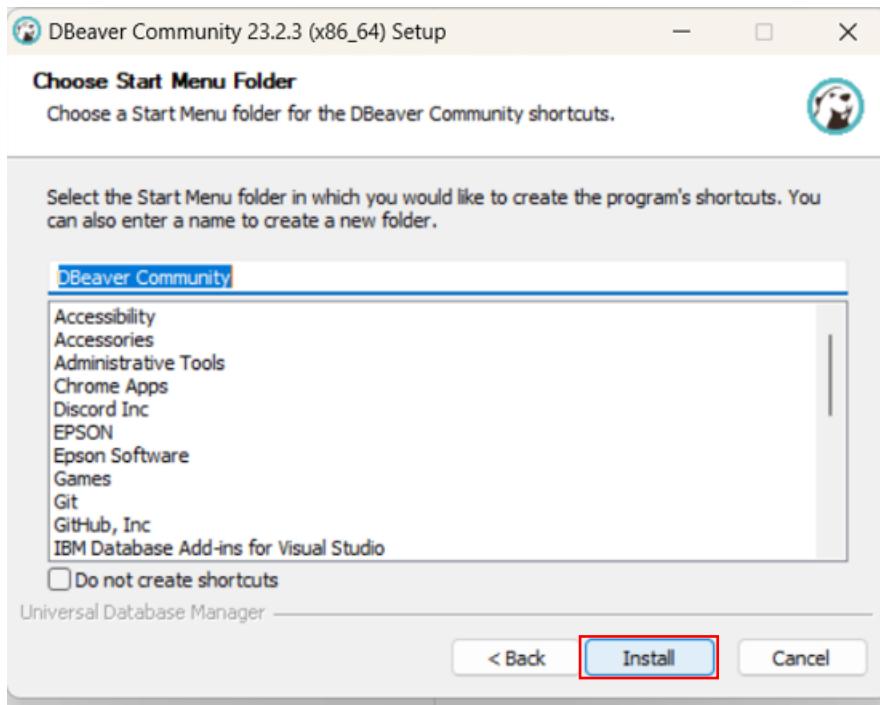




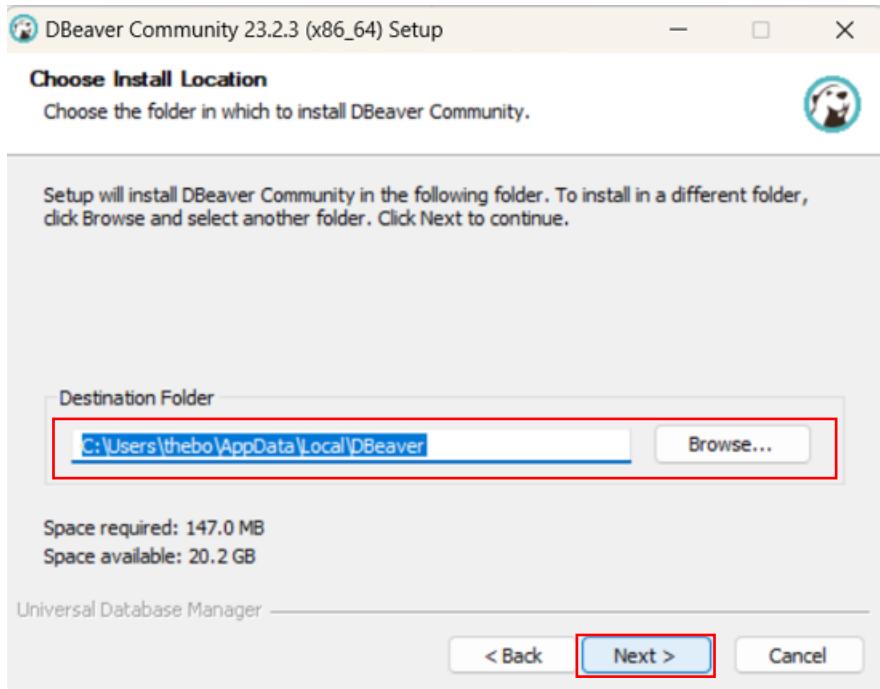
- Select the path and click next.

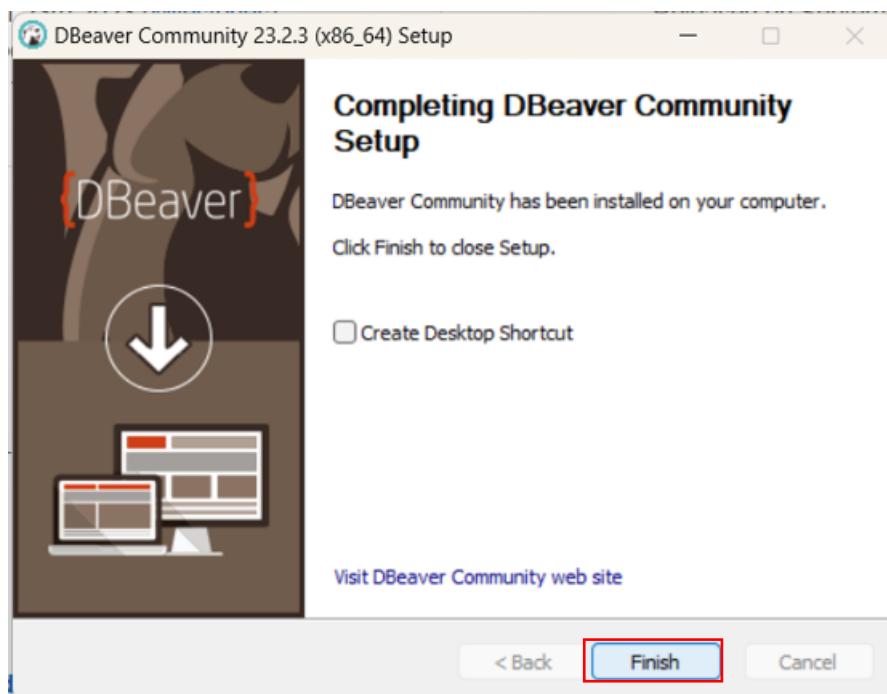
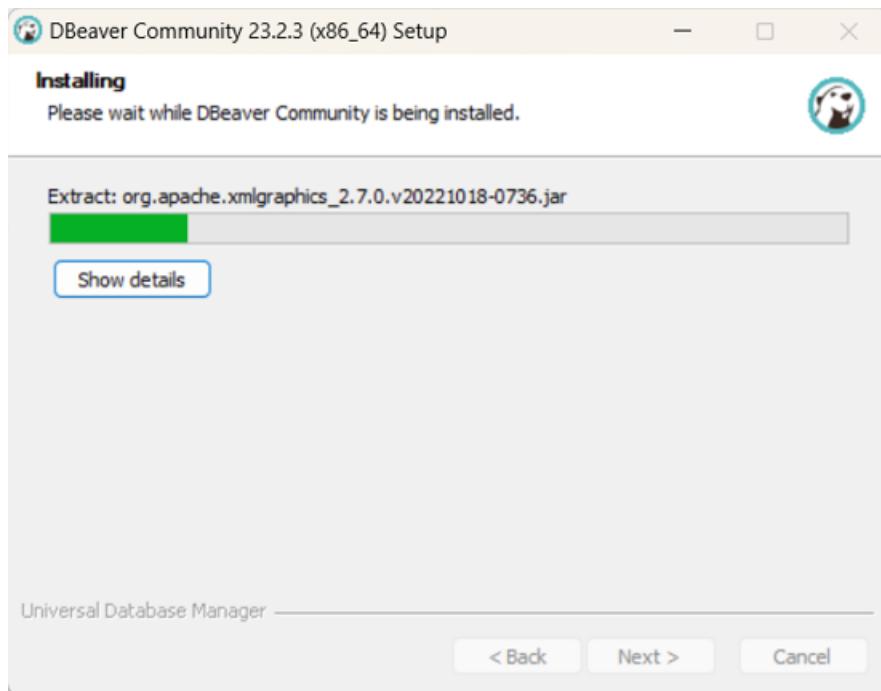


- Now install DBeaver.



- Choose install location.



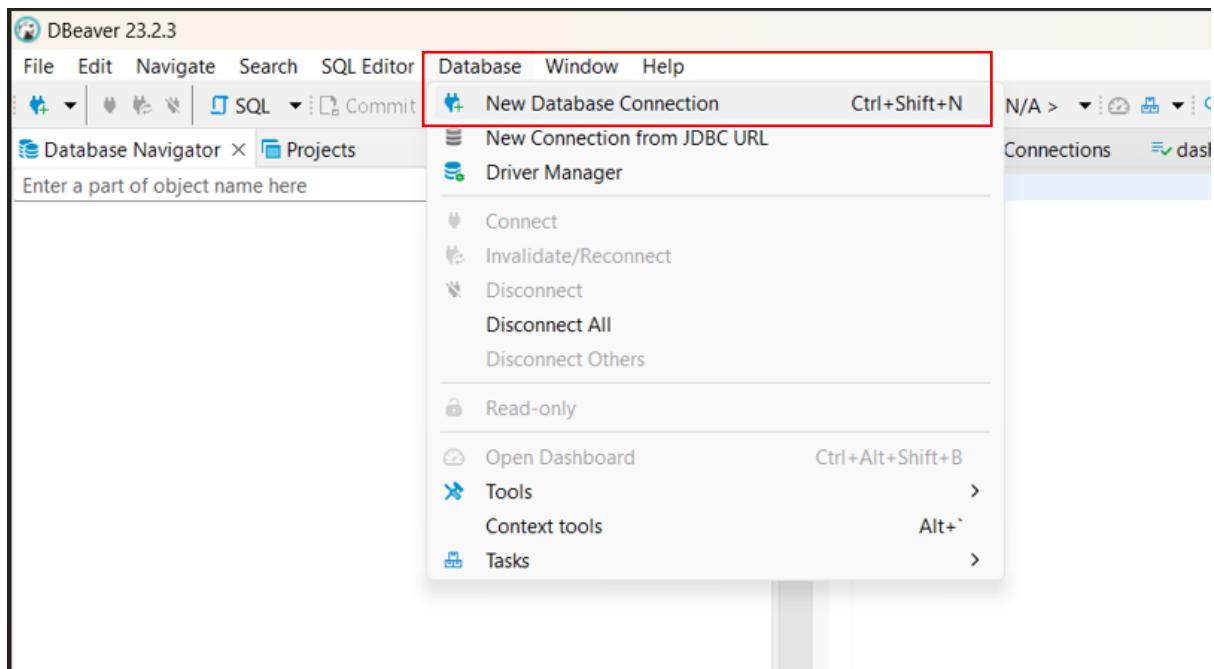


- Now the setup is finished.

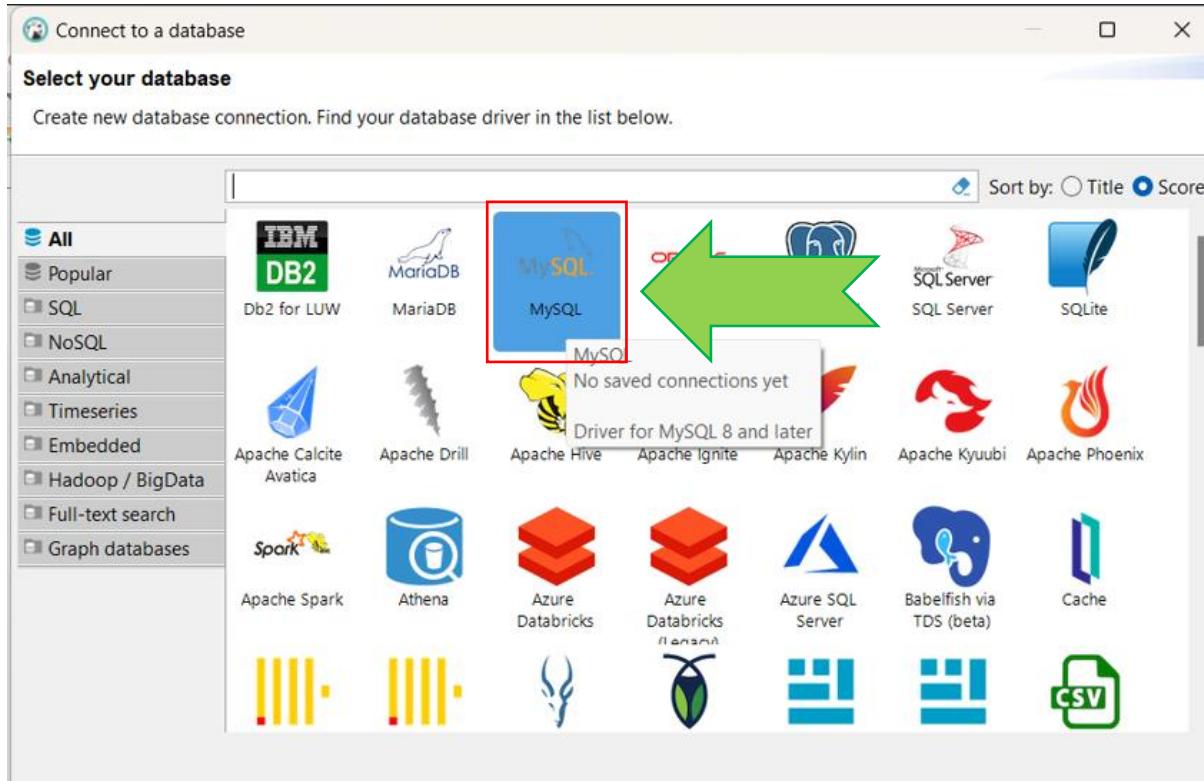
- The fifth step is to connect the database with the DBeaver community.
- Open the DBeaver community.



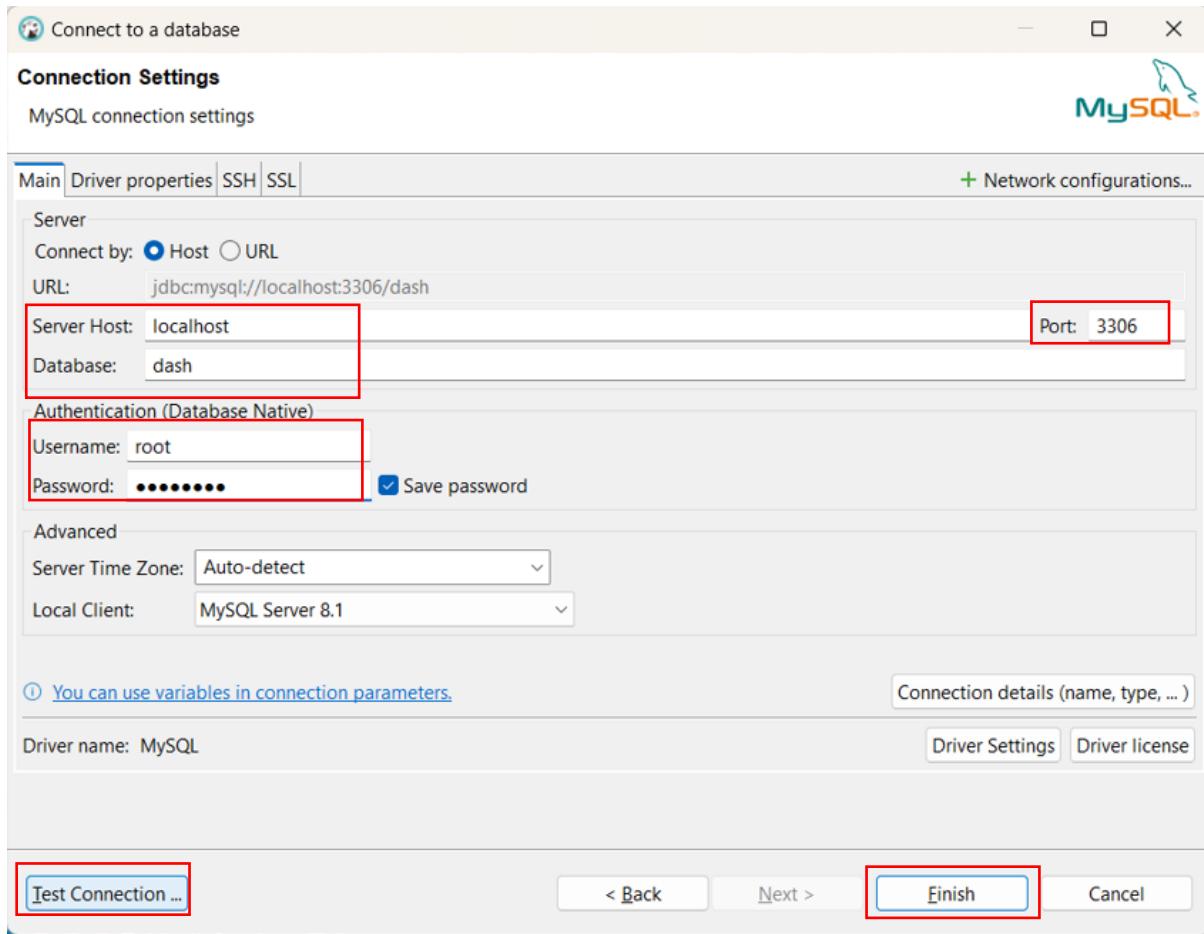
- Click database
- In that click New Database Connection.



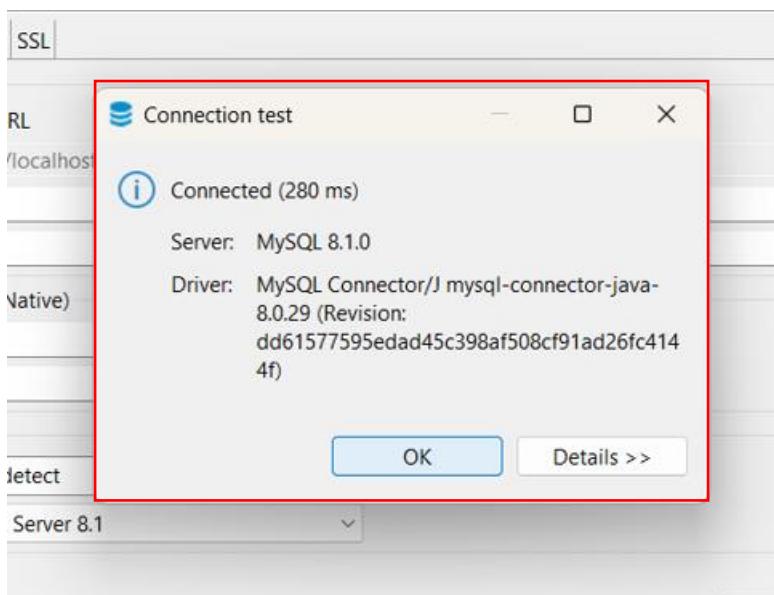
- Choose MySQL



- Fill the correct MySQL server details to connect with the DBeaver.

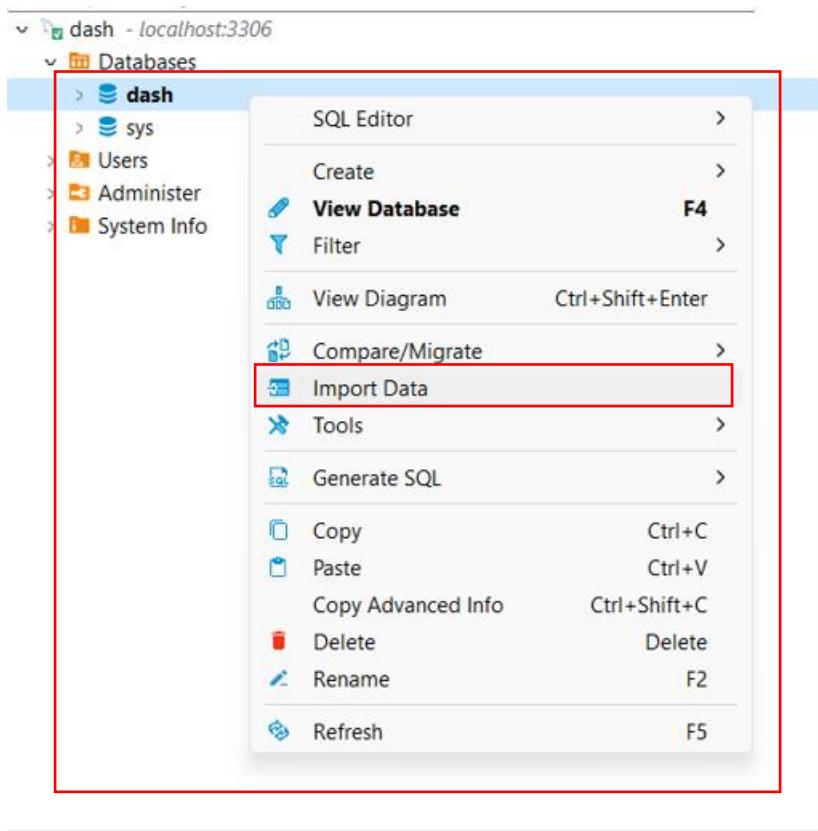


- Click Test connection



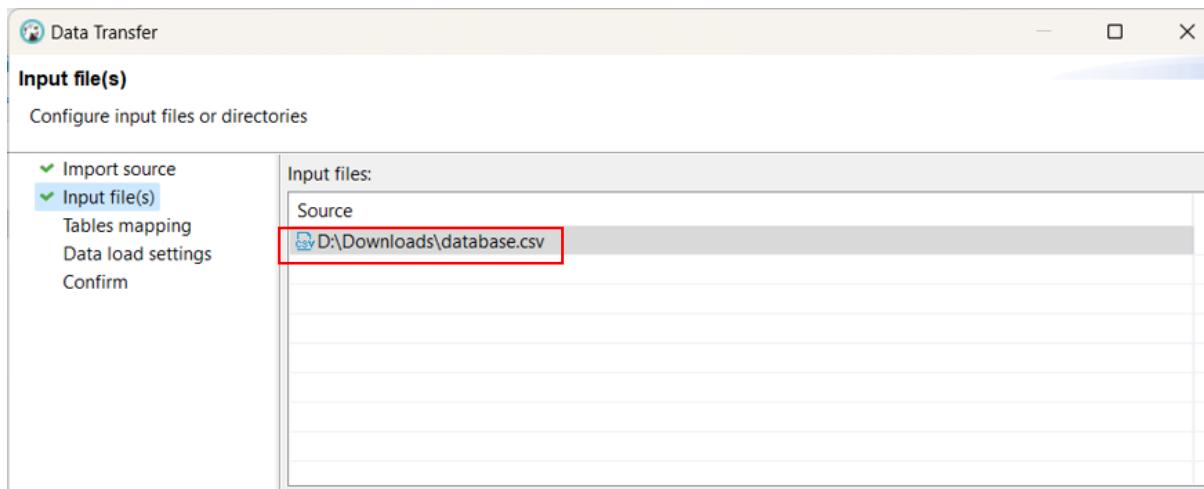
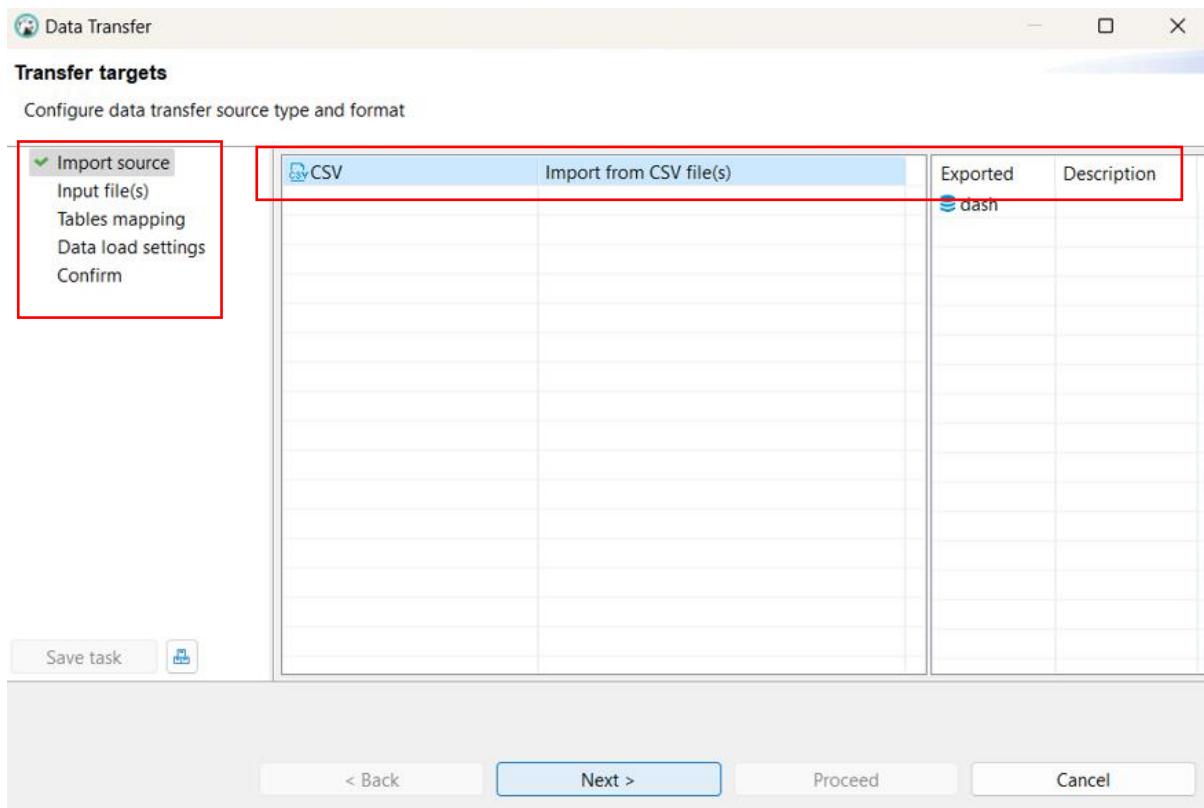
- Now the Database is successfully connected.

- The sixth step is the ETL process.
- We can import the necessary datasets.



- Right click the database and click the import data.

Click next and choose the dataset.



The screenshot shows the 'Data Transfer' application interface. The title bar says 'Data Transfer'. The main window has a header 'Tables mapping' and a subtitle 'Map tables and columns transfer'. On the left, there's a sidebar with three checked items: 'Import source', 'Input file(s)', and 'Tables mapping' (which is highlighted with a blue border). Below these are 'Data load settings' and 'Confirm' buttons. The main area has a table titled 'Target container: dash'. The table has four columns: 'Source', 'Target', 'Mapping', and 'Transform'. One row is visible, showing 'database.csv' as the source and 'database' as the target with 'create' mapping. To the right of the table are several buttons: 'Choose ...', 'Browse ...', 'Configure ...', and 'Preview data'.

Data Transfer

Data load settings

Configuration of table data load

Import source
Input file(s)
Tables mapping
Data load settings

Confirm

Data load

Transfer auto-generated columns
 Truncate target table(s) before load
 Disable referential integrity checks during the transfer

Replace method: <None>

[Replace/Ignore method documentation](#)

General

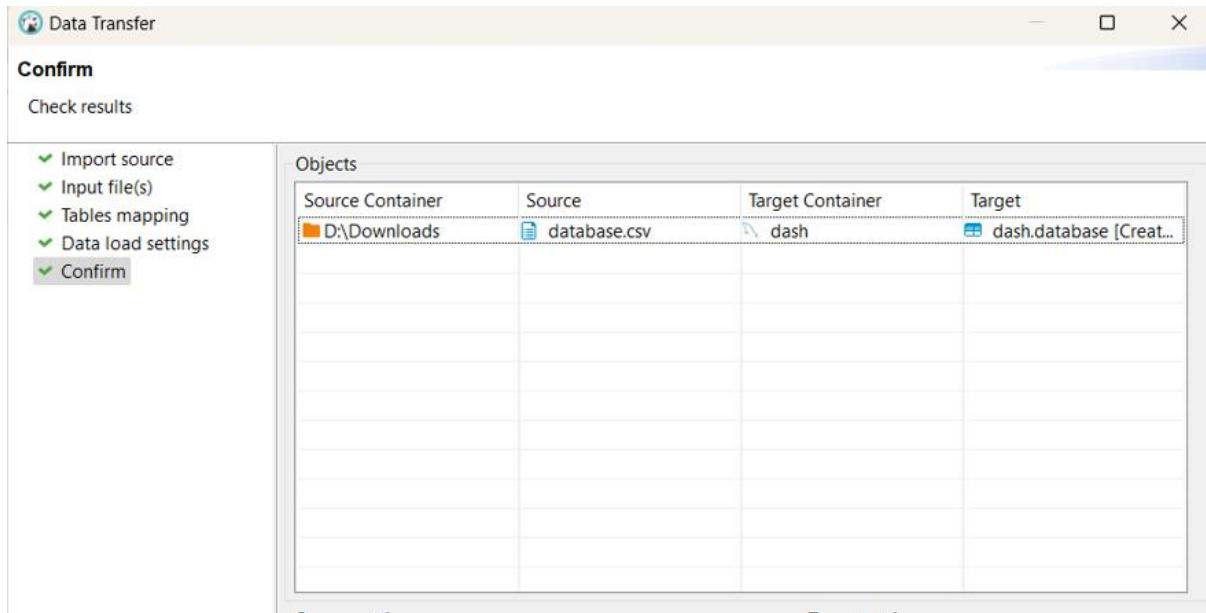
Open table editor on finish
 Show finish message

Performance

Open new connection(s)
 Use transactions

Do Commit after row insert:

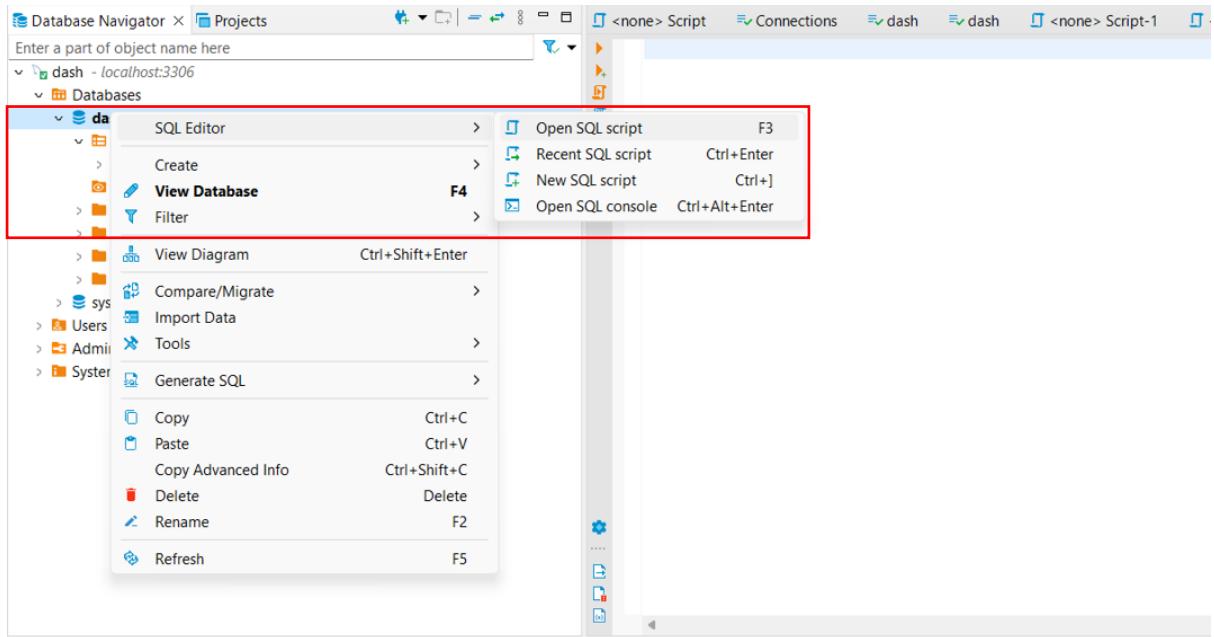
Use multi-row value insert 500
 Skip bind values during insert
 Disable batches
 Ignore duplicate rows errors
 Use bulk load



- Click F4 to view the imported data in a table form.

	Date	Time	Latitude	Longitude	Type	Depth	Depth Error	Dep
1	01/02/1965	13:44:18	19.246	145.616	Earthquake	131.6		
2	01/04/1965	11:29:49	1.863	127.352	Earthquake	80		
3	01/05/1965	18:05:58	-20.579	-173.972	Earthquake	20		
4	01/08/1965	18:49:43	-59.076	-23.557	Earthquake	15		
5	01/09/1965	13:32:50	11.938	126.427	Earthquake	15		
6	01/10/1965	13:36:32	-13.405	166.629	Earthquake	35		
7	01/12/1965	13:32:25	27.357	87.867	Earthquake	20		
8	01/15/1965	23:17:42	-13.309	166.212	Earthquake	35		
9	01/16/1965	11:32:37	-56.452	-27.043	Earthquake	95		
10	01/17/1965	10:43:17	-24.563	178.487	Earthquake	565		
11	01/17/1965	20:57:41	-6.807	108.988	Earthquake	227.9		
12	01/24/1965	00:11:17	-2.608	125.952	Earthquake	20		
13	01/29/1965	09:35:30	54.636	161.703	Earthquake	55		
14	02/01/1965	05:27:06	-18.697	-177.864	Earthquake	482.9		
15	02/02/1965	15:56:51	37.523	73.251	Earthquake	15		
16	02/04/1965	03:25:00	-51.84	139.741	Earthquake	10		
17	02/04/1965	05:01:22	51.251	178.715	Earthquake	30.3		
18	02/04/1965	06:04:59	51.639	175.055	Earthquake	30		
19	02/04/1965	06:37:06	53.530	172.007	Earthquake	25		

- Now right click on the dataset and click SQL editor to start the querying.



- Start doing the SQL queries.

The screenshot shows the SQL editor with the following query:

```
select * from `database` ;
```

The results are displayed in a grid titled "database 1". The data is as follows:

	Date	Time	Latitude	Longitude	Type	Depth	Depth Error	Dep
1	01/02/1965	13:44:18	19.246	145.616	Earthquake	131.6		
2	01/04/1965	11:29:49	1.863	127.352	Earthquake	80		
3	01/05/1965	18:05:58	-20.579	-173.972	Earthquake	20		
4	01/08/1965	18:49:43	-59.076	-23.557	Earthquake	15		
5	01/09/1965	13:32:50	11.938	126.427	Earthquake	15		
6	01/10/1965	13:36:32	-13.405	166.629	Earthquake	35		

The screenshot shows the MySQL Workbench interface. At the top, there are two tabs: 'Script-2' and 'database'. Below them is a toolbar with various icons. The main area contains a database grid titled 'database 1'. The grid has columns labeled 'Date', 'Time', and 'Depth'. The data is as follows:

	Date	Time	Depth
1	01/02/1965	13:44:18	131.6
2	01/04/1965	11:29:49	80
3	01/05/1965	18:05:58	20
4	01/08/1965	18:49:43	15
5	01/09/1965	13:32:50	15
6	01/10/1965	13:36:32	35
7	01/12/1965	13:32:25	20

At the bottom of the grid, there are buttons for Refresh, Save, Cancel, and navigation.

The screenshot shows the MySQL Workbench interface. At the top, there are two tabs: 'database' and 'Script-3'. Below them is a toolbar with various icons. The main area contains a SQL editor window with the following code:

```
create table customer(customerno int(3),age int(3),gender varchar(6),size int(4),purchaseno int(5))
```

To the right of the editor is an 'Output' pane with the message: 'Enter a part of a'. Below the editor is a 'Statistics 1' pane containing the following information:

Name	Value
Updated Rows	0
Query	create table customer(customerno int(3),age int(3),gender varchar(6),size int(4),purchaseno int(5))
Start time	Thu Oct 26 21:58:07 IST 2023
Finish time	Thu Oct 26 21:58:07 IST 2023

dash dash *<dash> Script-3

```
create table customer(customerno int(3),age int(3),gender varchar(6),size int(4),purchaseno int(5));
insert into customer (customerno ,age ,gender ,size ,purchaseno) values(1,24,'Male','40',43);
```

Statistics 1

Name	Value
Updated Rows	1
Query	insert into customer (customerno ,age ,gender ,size ,purchaseno) values(1,24,'Male','40',43)
Start time	Thu Oct 26 22:02:10 IST 2023
Finish time	Thu Oct 26 22:02:10 IST 2023

dash dash *<dash> Script-3

```
create table customer(customerno int(3),age int(3),gender varchar(6),size int(4),purchaseno int(5));
insert into customer (customerno ,age ,gender ,size ,purchaseno) values(1,24,'Male','40',43);
select * from customer;
ALTER TABLE customer MODIFY COLUMN size varchar(4);
```

Statistics 1

Name	Value
Updated Rows	1
Query	ALTER TABLE customer MODIFY COLUMN size varchar(4)
Start time	Thu Oct 26 22:07:28 IST 2023
Finish time	Thu Oct 26 22:07:28 IST 2023

dash dash *<dash> Script-3

```
create table customer(customerno int(3),age int(3),gender varchar(6),size int(4),purchaseno int(5));
insert into customer (customerno ,age ,gender ,size ,purchaseno) values(1,24,'Male','40',43);
select * from customer;
ALTER TABLE customer MODIFY COLUMN size varchar(4);
update customer
set size= 'L'
where age=24;
```

Statistics 1

Name	Value
Updated Rows	1
Query	update customer set size= 'L' where age=24
Start time	Thu Oct 26 22:10:37 IST 2023
Finish time	Thu Oct 26 22:10:37 IST 2023

dash dash *<dash> Script-3

```

create table customer(customerno int(3),age int(3),gender varchar(6),size int(4),purchaseno int(5));
insert into customer (customerno ,age ,gender ,size ,purchaseno) values(1,24,'Male','40',43);
select * from customer;
ALTER TABLE customer MODIFY COLUMN size varchar(4);
update customer
set size= 'L'
where age=24;
select * from customer;

```

customer 1

select * from customer Enter a SQL expression to filter results (use Ctrl+Space)

Grid	123 customerno	123 age	gender	size	123 purchaseno	Value
Grid	1	24	Male	L	43	1

*<dash> Script-3

```

insert into customer (customerno ,age ,gender ,size ,purchaseno) values(2, 36,'Female','XL',12);
insert into customer (customerno ,age ,gender ,size ,purchaseno) values(3, 48,'Male','M',26);
insert into customer (customerno ,age ,gender ,size ,purchaseno) values(4, 25,'Male','XXL',98);

```

*<dash> Script-2

```

INSERT INTO Customer (customerno ,age ,gender, `size` , purchaseno)
VALUES (3,55,'Male','XXL',87);
select * from customer c ;

```

customer 1

select * from customer c Enter a SQL expression to filter results (use Ctrl+Space)

Grid	123 customerno	123 age	gender	size	123 purchaseno
Grid	1	24	Male	L	43
Text	2	36	Female	XL	12
Text	3	48	Male	M	37
Text	4	20	Female	XL	26
Text	5	55	Male	XXL	87

*<dash> Script-2 X

```
① INSERT INTO Customer (customerno ,age ,gender, `size`, purchaseno)
VALUES (3,55,'Male','XXL',87);
select * from customer c ;
② SELECT * FROM Customer
ORDER BY customerno ;
```

customer 1 X

SELECT * FROM Customer ORDER BY customer | Enter a SQL expression to filter results (use Ctrl+Space)

	customerno	age	gender	size	purchaseno
1	1	24	Male	L	43
2	2	36	Female	XL	12
3	3	55	Male	XXL	87
4	4	48	Male	M	37
5	5	20	Female	XL	26

The screenshot shows the MySQL Workbench interface. The top part is a script editor titled "Script-2" containing the following SQL code:

```
create table items(ItemPurchased varchar(30),Category varchar(30),
purchaseno int(30),Location varchar(30), payment varchar(30));
select * from items;
```

The bottom part is a results grid titled "items 1" with one row of data:

	ItemPurchased	Category	purchaseno	Location	payment
1	laptop	Electronics	123	New York	Credit Card

The screenshot shows a database management interface with two main panes. The top pane is a 'Script-2' editor titled 'customer' containing the following SQL code:

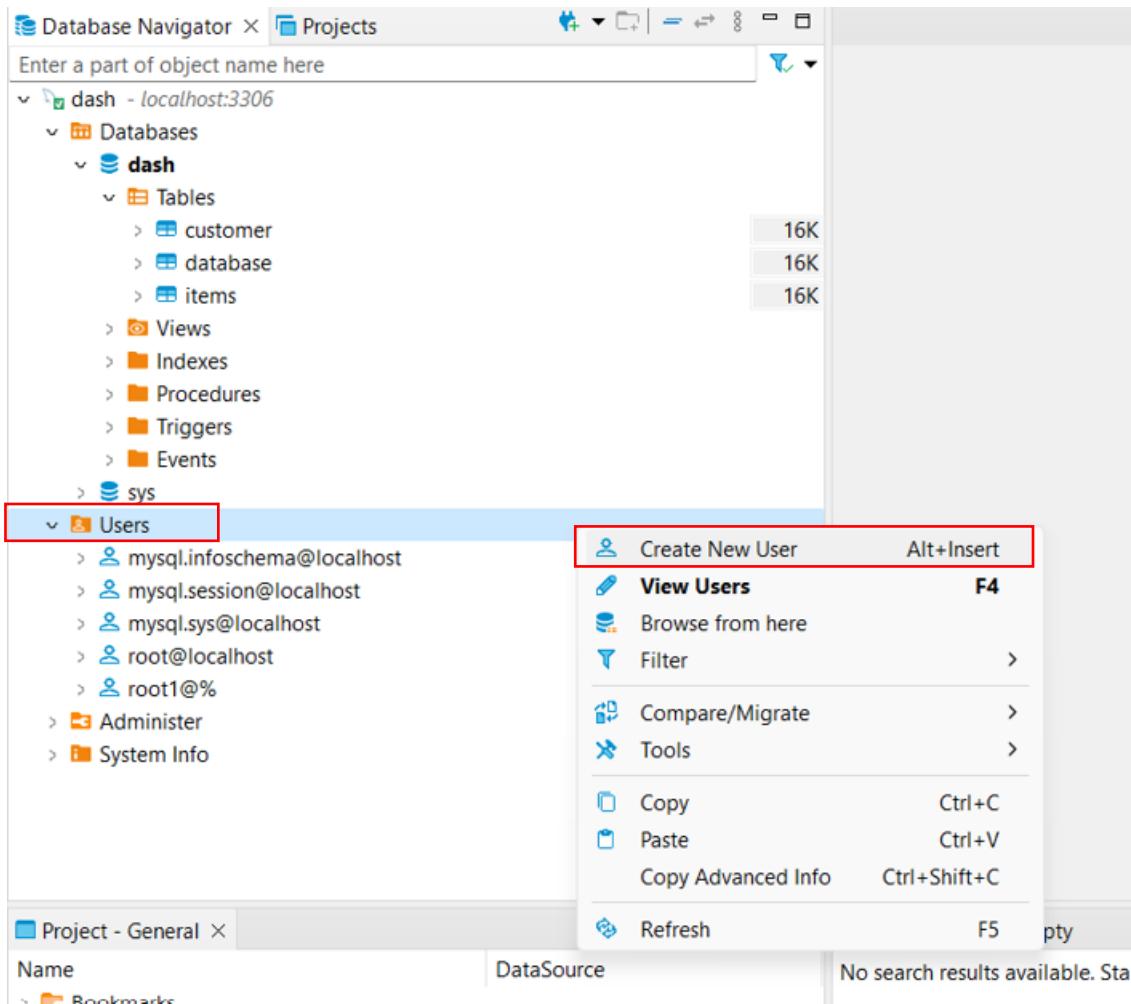
```
create table items(ItemPurchased varchar(30),Category varchar(30),
purchaseno int(30),Location varchar(30), payment varchar(30));
select * from items;
insert into items(ItemPurchased ,Category ,purchaseno ,Location , payment)
values ('Coat','clothing',87,'Mumbai','UPI');
SELECT purchaseno FROM Customer
UNION ALL
SELECT purchaseno FROM items
ORDER BY purchaseno;
```

The bottom pane is a 'Results 1' grid titled 'purchaseno'. It displays the following data:

Grid	purchaseno
5	37
6	37
7	43
8	43
9	87
10	87

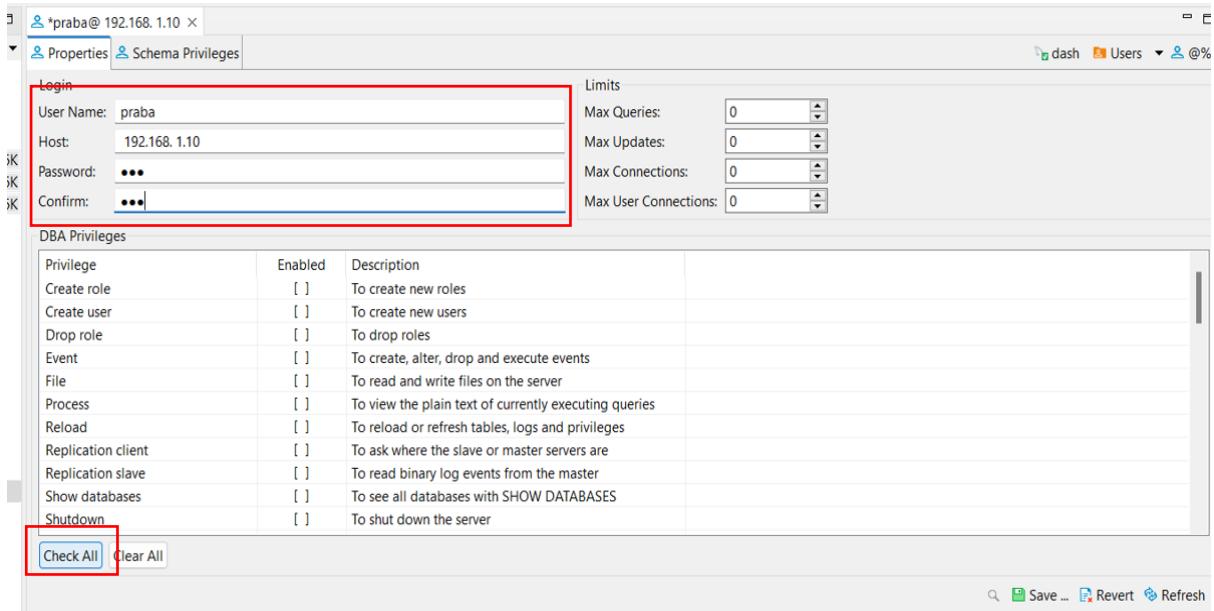
- Now the ETL process is finished.

- The seventh step is to enable data architects so that they can explore and analyze our data for better decision making.

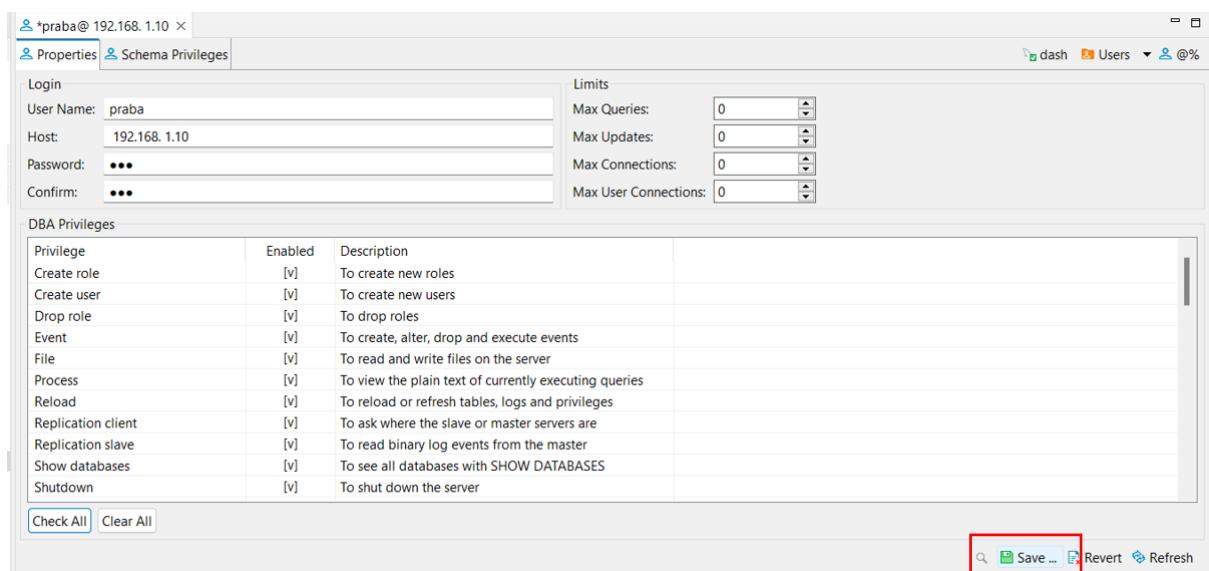


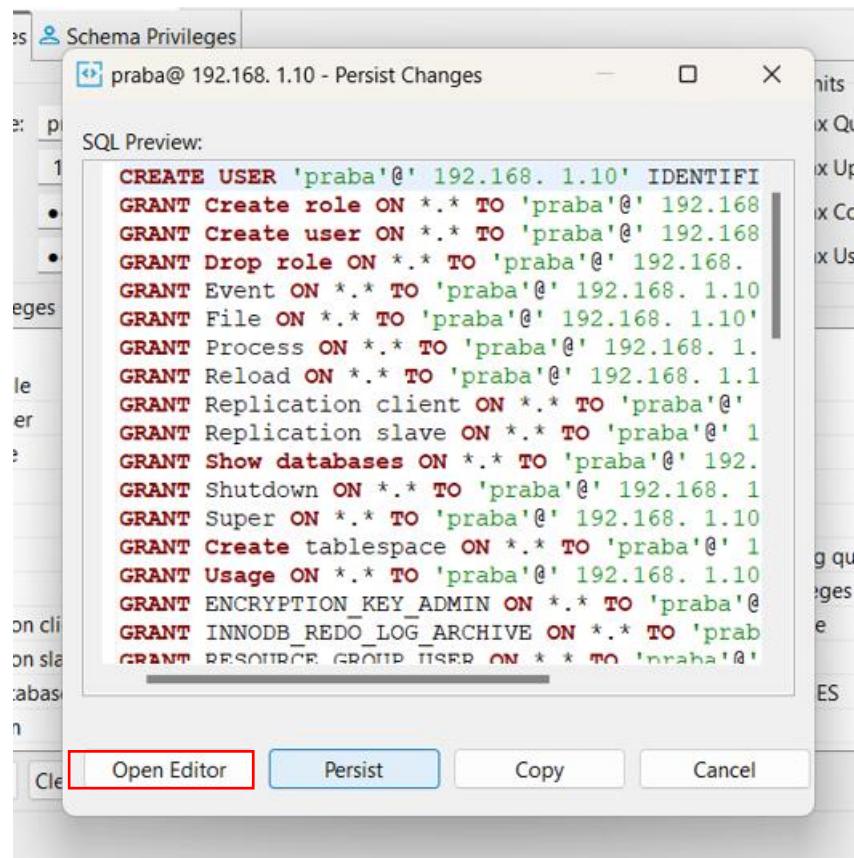
- Right click the users and click New user to create a new user.

- Fill the data architect details like username, password and architect's IP address.



- Click check all to grant all permissions of our database.
- Save the progress.





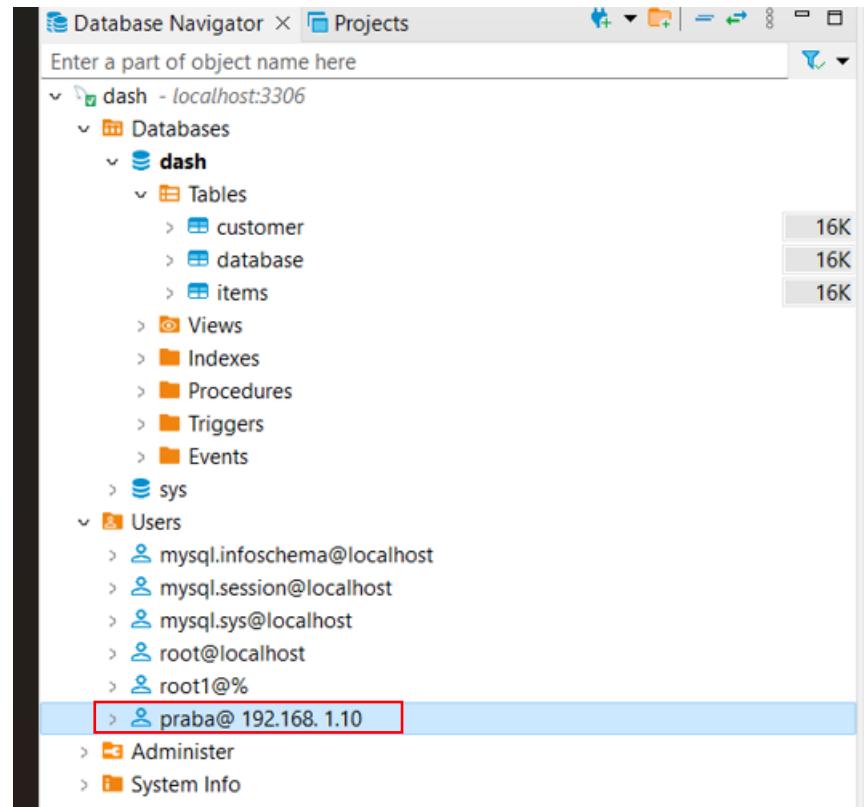
The screenshot shows the MySQL Workbench interface with a dialog box titled "Schema Privileges". The connection is set to "praba@ 192.168.1.10 - Persist Changes". The "SQL Preview" pane contains the following SQL code:

```
CREATE USER 'praba'@'192.168.1.10' IDENTIFI
GRANT Create role ON *.* TO 'praba'@'192.168
GRANT Create user ON *.* TO 'praba'@'192.168
GRANT Drop role ON *.* TO 'praba'@'192.168.
GRANT Event ON *.* TO 'praba'@'192.168.1.10
GRANT File ON *.* TO 'praba'@'192.168.1.10'
GRANT Process ON *.* TO 'praba'@'192.168.1.
GRANT Reload ON *.* TO 'praba'@'192.168.1.1
GRANT Replication client ON *.* TO 'praba'@
GRANT Replication slave ON *.* TO 'praba'@'1
GRANT Show databases ON *.* TO 'praba'@'192.
GRANT Shutdown ON *.* TO 'praba'@'192.168.1
GRANT Super ON *.* TO 'praba'@'192.168.1.10
GRANT Create tablespace ON *.* TO 'praba'@'1
GRANT Usage ON *.* TO 'praba'@'192.168.1.10
GRANT ENCRYPTION_KEY ADMIN ON *.* TO 'praba'@
GRANT INNODB REDO LOG ARCHIVE ON *.* TO 'prab
GRANT RESOURCE GROUP IF NOT EXISTS ON *.* TO 'praba'@'
```

At the bottom of the dialog, there are four buttons: "Open Editor", "Persist", "Copy", and "Cancel". The "Persist" button is highlighted with a red border.

- Click persist button to confirm all the permissions.

- Now we can see that the new user is created for data exploration.



CONCLUSION:

In conclusion, data warehousing has been a noteworthy undertaking that has yielded important insights into data analysis and database management. We set up a MySQL database successfully, establishing tables, identifying relationships, and optimizing the setup for our individual purposes. This groundwork was critical to the project's success.

We effectively imported data into our database using DBeaver, modifying and cleaning it as necessary. Ensuring the quality and usability of the data required taking this crucial step. We were able to conduct complicated queries and undertake in-depth data analysis because of DBeaver's excellent SQL capabilities. DBeaver's SQL and data analysis capabilities allow users to extract valuable insights from the data stored in MySQL.