Hands-on Lab: Generative AI for Database, Data Warehouse Schema Design

Estimated Effort: 30 mins

Introduction

Data warehouse schema design is crucial for organizing and optimizing data storage, ensuring efficient query performance, maintaining data integrity, and facilitating scalability and flexibility. In this lab, you will learn how we can leverage Generative AI to propose a data warehouse schema, create the said data warehouse in an SQL engine, and generate the schema diagram.

Objectives

For a given scenario, use Generative AI to perform the following tasks:

- Propose a data warehouse schema
- Generate SQL code to create the said data warehouse
- Generate the image of the schema design for the data warehouse

Scenario

A fashion retail organization has hired you as a data engineer to create a data warehouse for them. They want to include data from the following sources.

1. Employee data
2. Inventory data
3. Sales data
4. Customer Profile data
5. Seller data

The employer has asked you to come up with a fully planned schema diagram for the said data warehouse. You have the flexibility to propose any and all relevant attributes and tables as per requirement.

Propose a data warehouse schema

Warehouse designing may become a daunting process in terms of the planning required to connect the data coming from various sources. In this regard, Generative AI is a useful tool to generate a proposal of how the data should be recorded comprehensively and how the data from different tables can be linked together.

Consider the following prompt.

```
1. 1
2. 2
3. 3

1. Create a detailed data warehouse schema for a fashion retail store that should contain employee data, sales data, inventory data, customer profiles,
2. The sales data should connect to employee data, customer profile, and inventory tables.
3. The inventory table should further be connected to the employee table and the seller table.
```

Copied!

You can expect a response as shown below.

▼ Click here for a sample response
To create a detailed data warehouse schema for a fashion retail store, you can design the following tables with their respective headers, primary keys, and foreign keys:

1. Employee Data:
    - employee_id (Primary Key)
    - employee_name
    - position
    - department
2. Sales Data:
    - sale_id (Primary Key)
    - employee_id (Foreign Key referencing Employee Data)
    - customer_id (Foreign Key referencing Customer Profiles)
    - product_id (Foreign Key referencing Inventory Data)
    - sale_date
    - sale_amount
3. Inventory Data:
    - product_id (Primary Key)
    - product_name

- price
- quantity
- employee_id (Foreign Key referencing Employee Data)
- seller_id (Foreign Key referencing Seller Information)

4. Customer Profiles:
    - customer_id (Primary Key)
    - customer_name
    - email
    - address
    - phone_number
5. Seller Information:
    - seller_id (Primary Key)
    - seller_name
    - seller_type

You can further establish relationships between the tables based on the foreign keys mentioned above. This schema design will allow you to effectively store and manage data for a fashion retail store in a data warehouse.

Creating the Warehouse

Once the tables and their content have been defined, you can use Python and SQL to create the said warehouse. Generative AI can again help you in creating the said warehouse by proposing a code that connects to an SQL server and creates the required data warehouse.

You will leverage the chatting ability of the Generative AI model and ask it to provide an SQL code to create the data warehouse in SQL.

```
1
```

```
Provide an SQL code to create the data warehouse discussed above.
```

Copied!

This prompt will generate an SQL code that can be used to create the said data warehouse in any SQL interface.

▼ Click here for a sample response
Certainly! Here is an SQL code to create the data warehouse schema for a fashion retail store:

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
```

```
CREATE TABLE Employee (
    employee_id INT PRIMARY KEY,
    employee_name VARCHAR(50),
    employee_role VARCHAR(50),
    employee_department VARCHAR(50)
);

CREATE TABLE SalesData (
```

```
    sale_id INT PRIMARY KEY,
    sale_date DATE,
    sale_amount DECIMAL(10, 2),
    employee_id INT,
    customer_id INT,
    inventory_id INT,
    FOREIGN KEY (employee_id) REFERENCES Employee(employee_id),
    FOREIGN KEY (customer_id) REFERENCES CustomerProfiles(customer_id),
    FOREIGN KEY (inventory_id) REFERENCES Inventory(inventory_id)
);

CREATE TABLE Inventory (
    inventory_id INT PRIMARY KEY,
    product_name VARCHAR(50),
    product_category VARCHAR(50),
    quantity_available INT,
    seller_id INT,
    FOREIGN KEY (seller_id) REFERENCES SellerInformation(seller_id)
);

CREATE TABLE CustomerProfiles (
    customer_id INT PRIMARY KEY,
    customer_name VARCHAR(50),
    customer_email VARCHAR(50),
    customer_phone VARCHAR(20)
);

CREATE TABLE SellerInformation (
    seller_id INT PRIMARY KEY,
    seller_name VARCHAR(50),
    seller_contact VARCHAR(50),
    seller_location VARCHAR(50)
);
```

Copied!

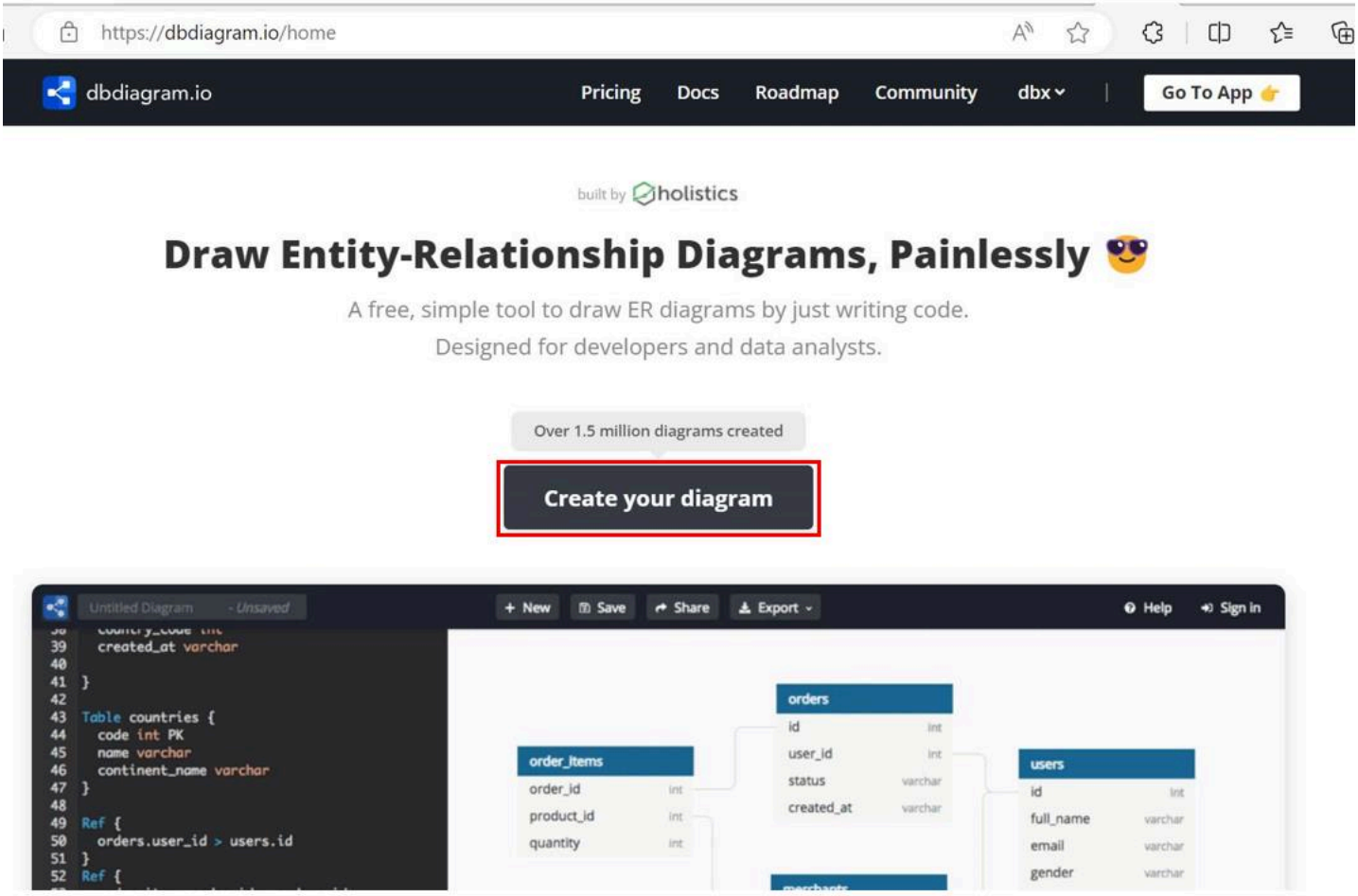
This SQL code will create the necessary tables with the specified columns and foreign key relationships as discussed in the data warehouse schema for a fashion retail store.

Generate the Schema Design

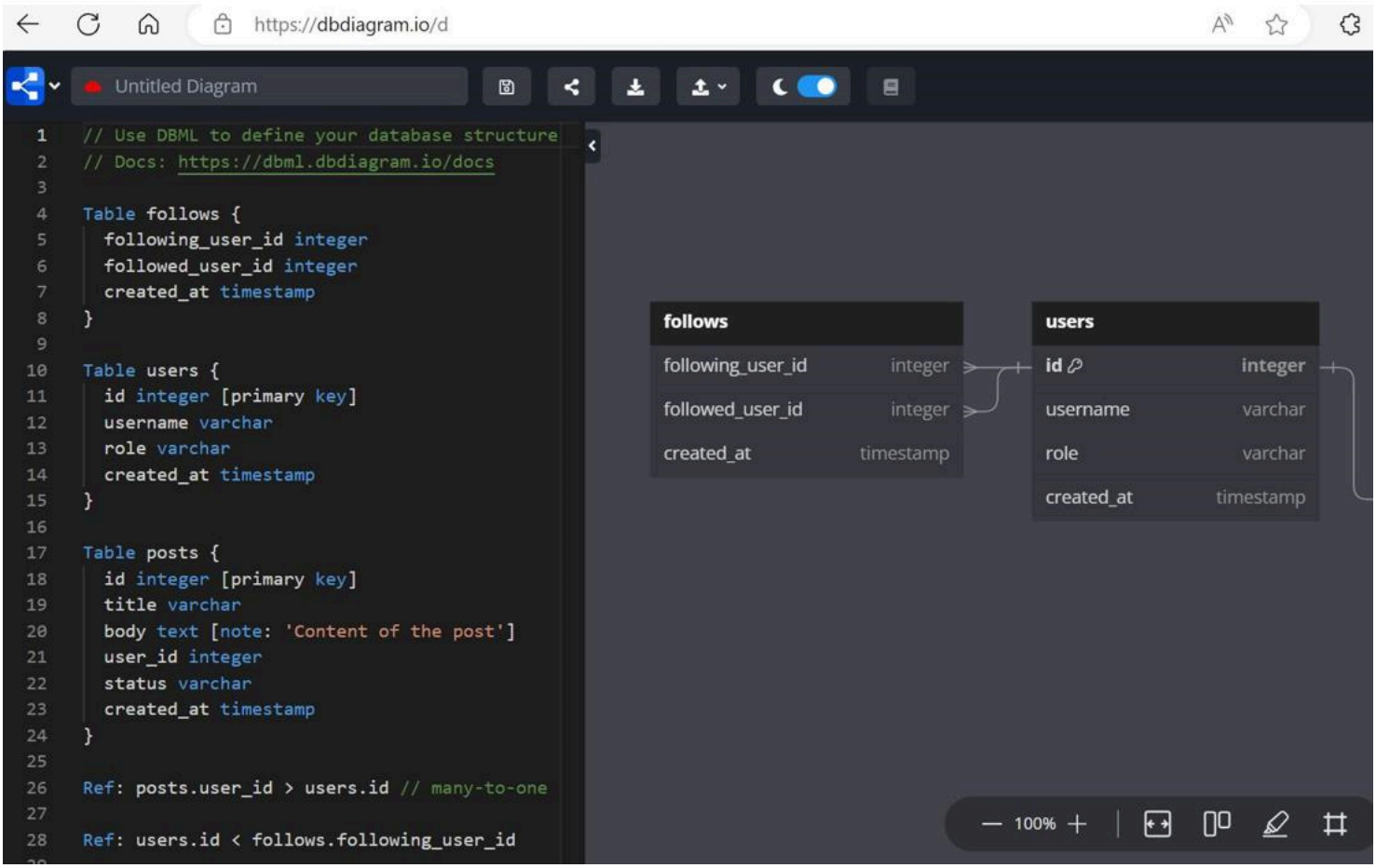
To generate the schema design, you are going to navigate to DBdiagram.io. This website provides an online tool that can be used to create the schema diagram for any database. You are going to use the SQL code generated above to visualize the schema design for the proposed data warehouse.

Steps:

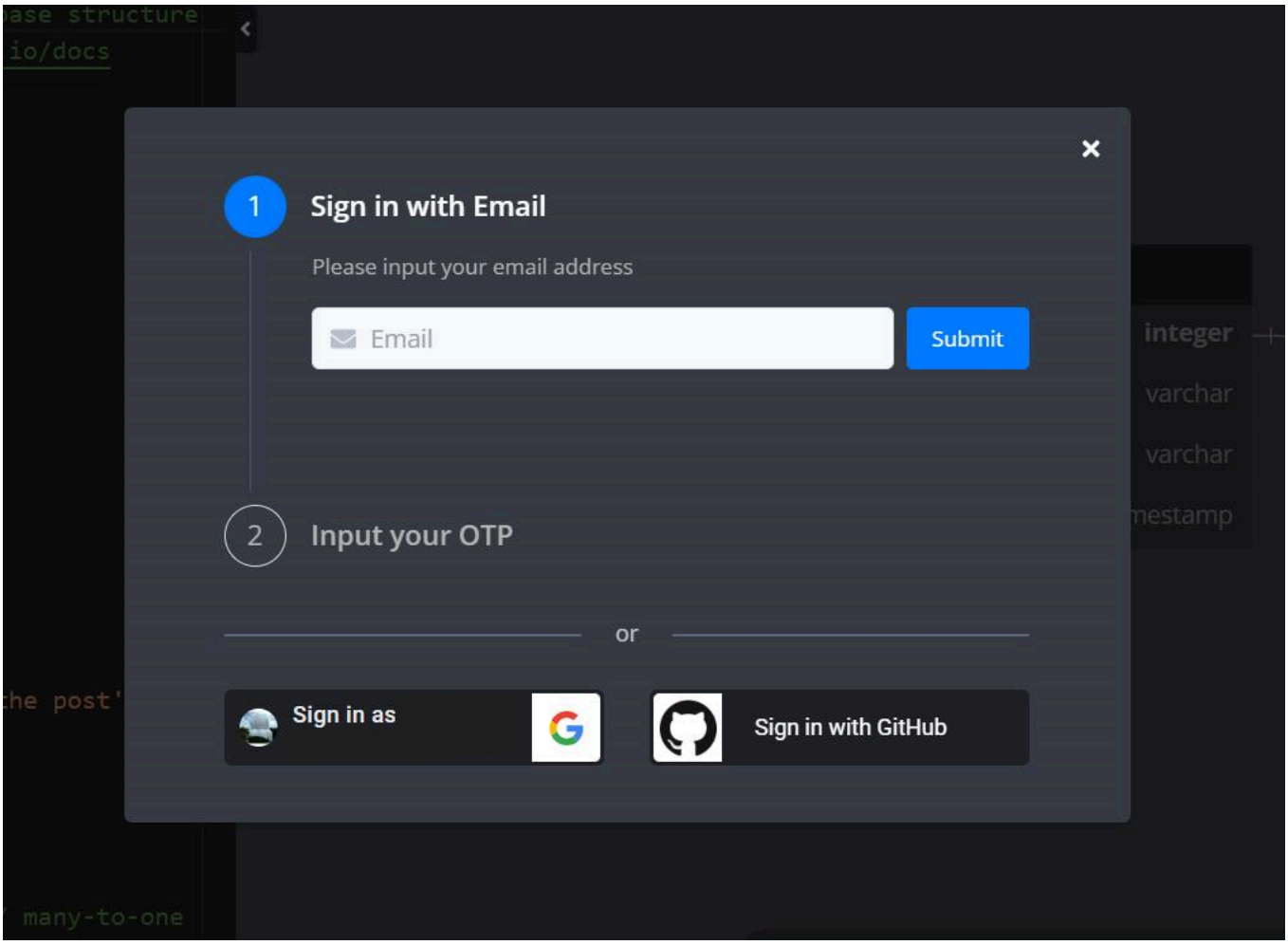
1. Click here to navigate to the DBdiagram.io.
2. On the homepage, click "Create your diagram".



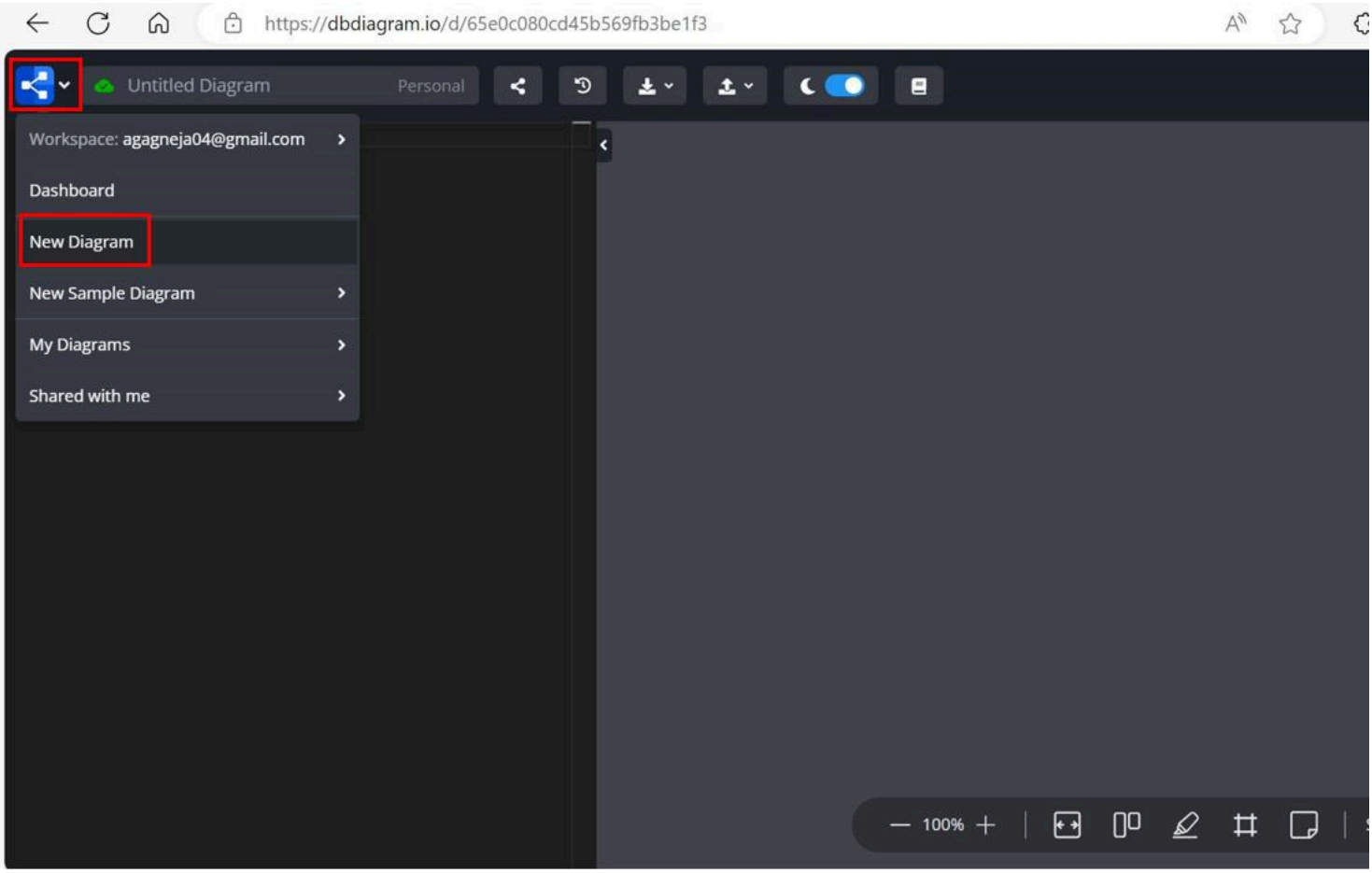
3. You will see an interface with a default example. Click the top right symbol to login to the interface, as shown in the image below.



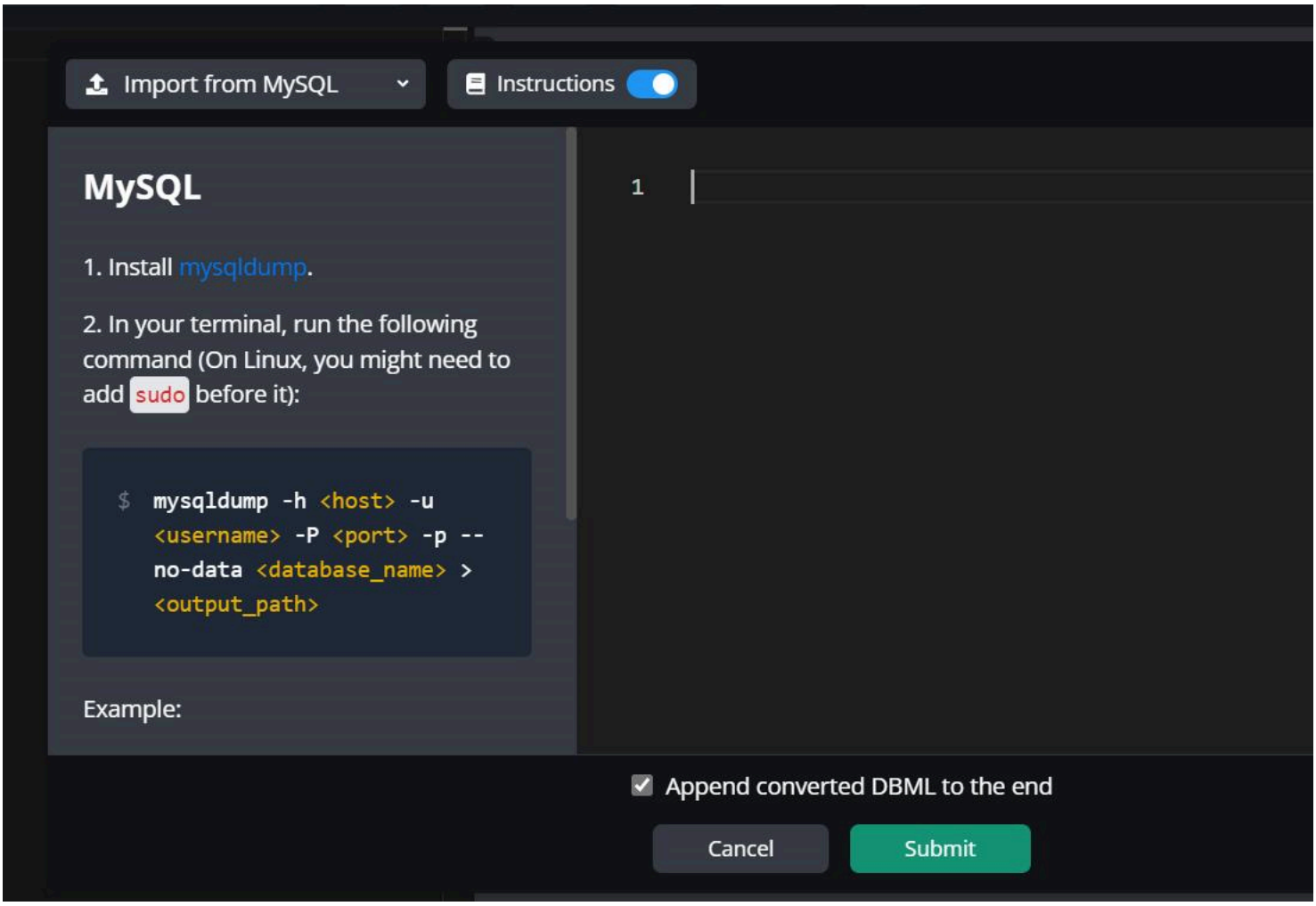
4. Here, you will get an option to login using either your Google, GitHub or any other email ID for free. Login to the system using any of the means available.



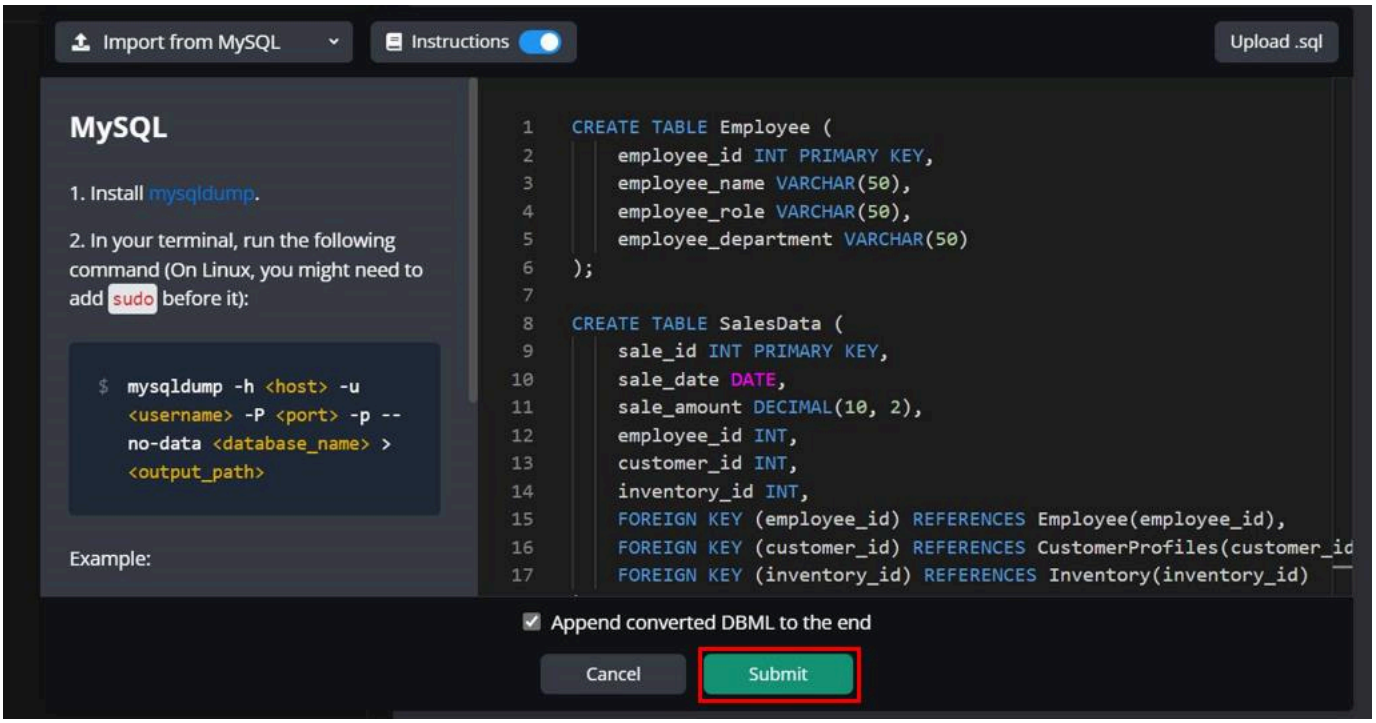
5. On the top left corner of the logged in interface, when you hover your cursor, you can see the option to create a 'New Diagram'.



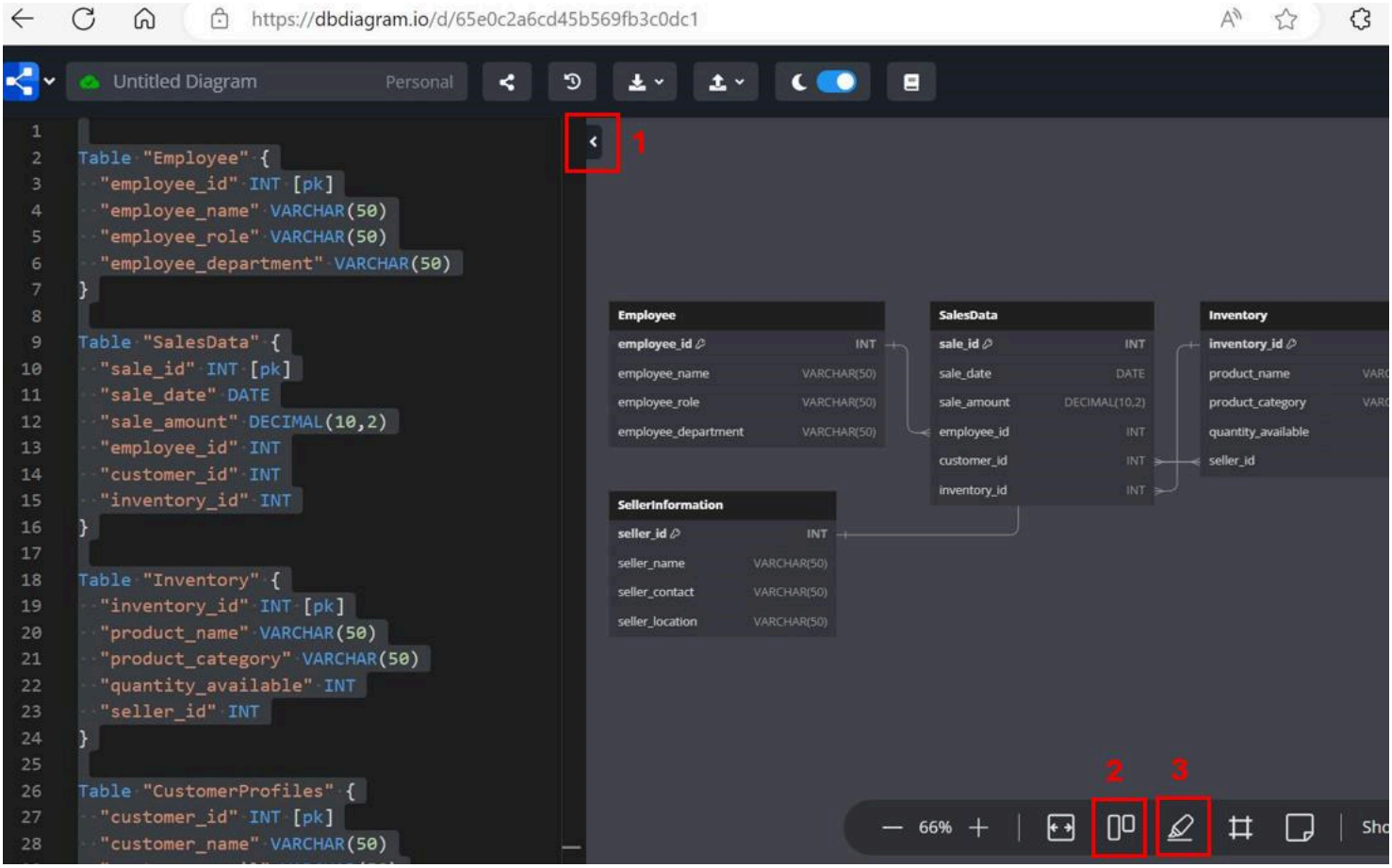
6. For this diagram, you can now hover your cursor over the upload symbol, as shown in the image below, and you will find an option to 'Import from MySQL' which you should click.



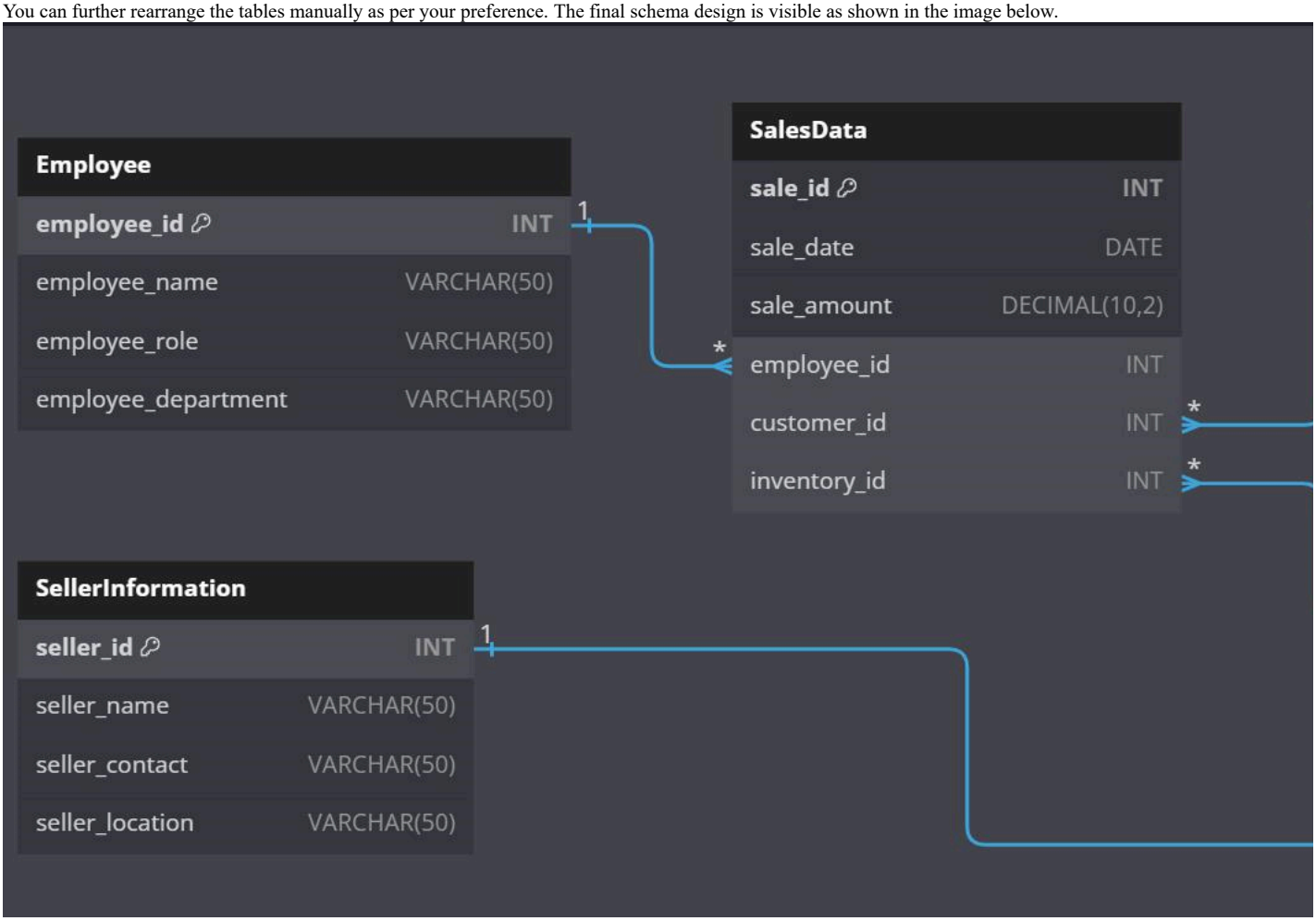
7. Copy the code generated in the previous step of this lab and paste it in this interface. Once done, click `Submit`.



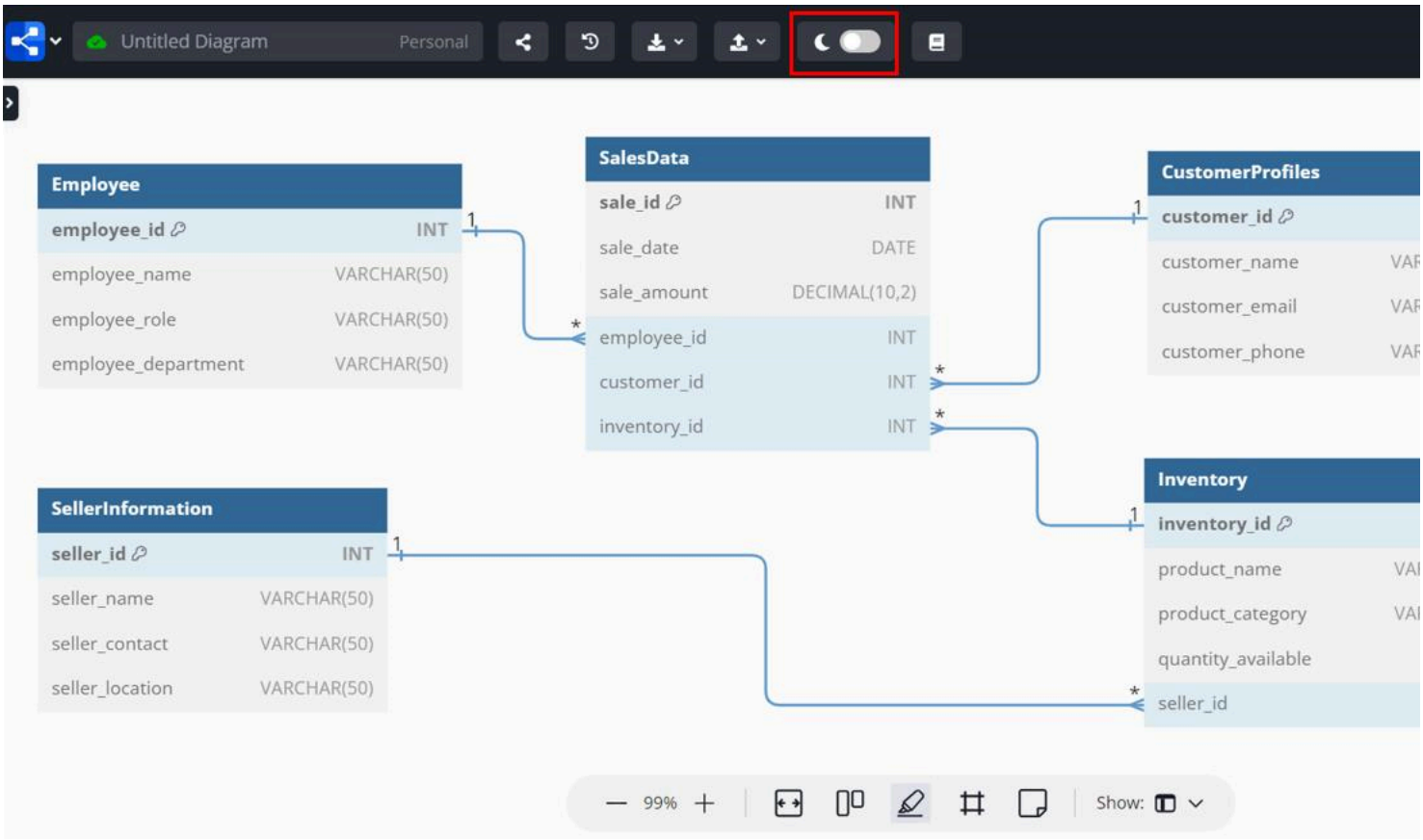
8. The resulting interface will have a modified code, which is being used to create the schema design. In sequence, press the tabs on the interface as shown in the image below. The first tab will close the coding interface, giving the resulting schema design a lot more room for visibility. The second tab resizes the diagram to fit the screen. The third tab highlights the interconnections between the tables.



9. You can further rearrange the tables manually as per your preference. The final schema design is visible as shown in the image below.



By disabling the dark mode, as shown in the image below, the schema can also be seen in the light mode.



Conclusion

Congratulations on completing this lab. You are now able to:

- Use Generative AI to create a data warehouse proposal
- Use Generative AI to generate SQL code that can be used to create the said data warehouse
- Generate the schema diagram of the data warehouse from the code using DBdiagram.io

Author(s)

Abhishek Gagneja

© IBM Corporation. All rights reserved.