

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.

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

You can expect a response as shown below.

► Click here for a sample response

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.

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

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

Generate the Schema Design

To generate the schema design, you are going to navigate to [DBdiagram.io](https://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".

The screenshot shows the DBdiagram.io homepage. At the top, there is a navigation bar with links for Pricing, Docs, Roadmap, Community, dbx, and a Go To App button. Below the navigation bar, it says "built by holistics". The main heading is "Draw Entity-Relationship Diagrams, Painlessly 😊". Below the heading, it says "A free, simple tool to draw ER diagrams by just writing code." and "Designed for developers and data analysts.". A button labeled "Create your diagram" is highlighted with a red box. In the bottom left, there is a dark panel showing some SQL code related to tables like countries, orders, order_items, users, and merchants. In the center, there is an Entity-Relationship diagram with tables: order_items, orders, and users. The orders table has columns id, user_id, status, and created_at. The order_items table has columns order_id, product_id, and quantity. The users table has columns id, full_name, email, and gender. Relationships are shown between orders and order_items, and between users and orders.

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.

Untitled Diagram

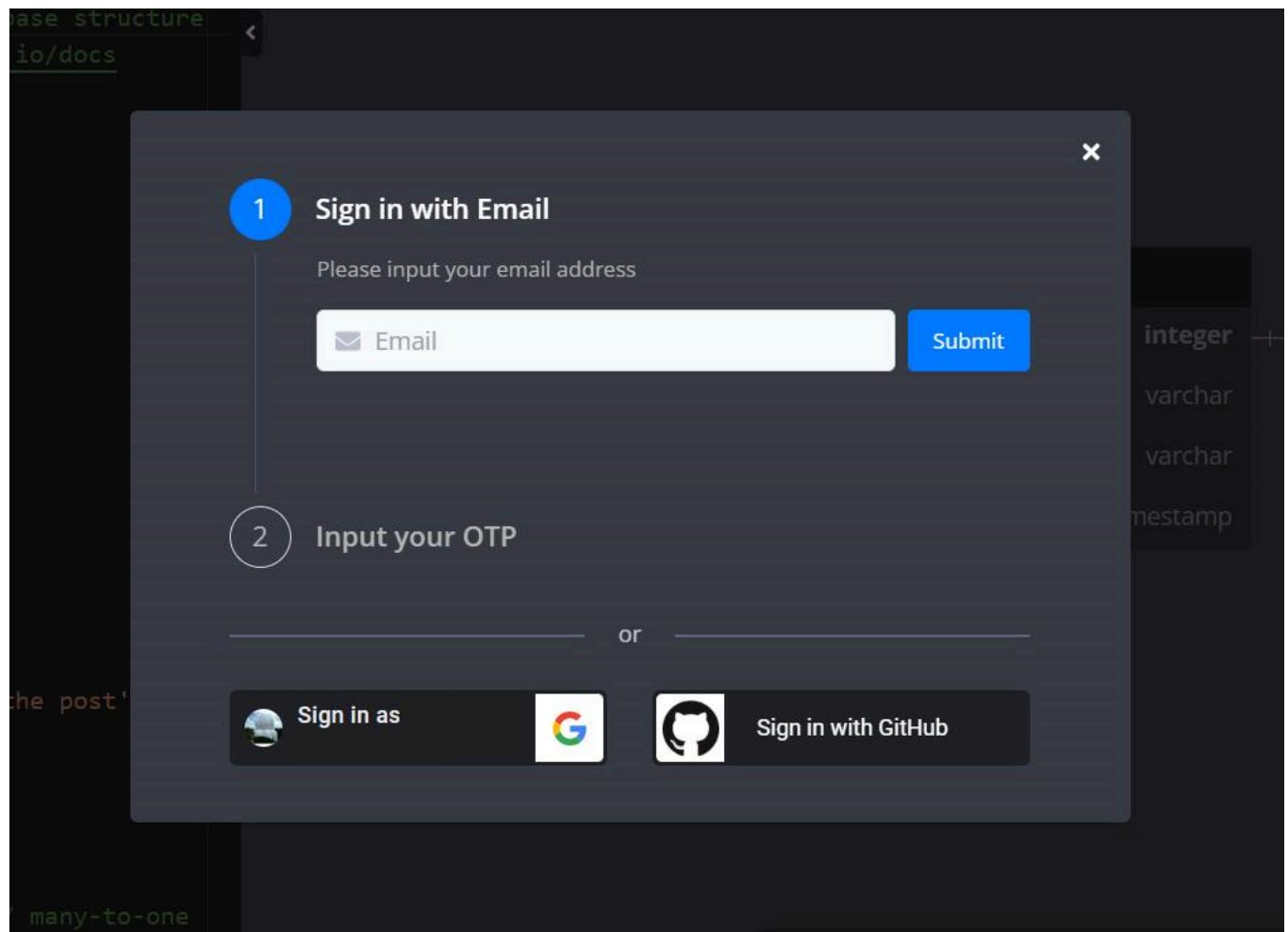
```

1 // Use DBML to define your database structure
2 // Docs: https://dbml.dbdiagram.io/docs
3
4 Table follows {
5     following_user_id integer
6     followed_user_id integer
7     created_at timestamp
8 }
9
10 Table users {
11     id integer [primary key]
12     username varchar
13     role varchar
14     created_at timestamp
15 }
16
17 Table posts {
18     id integer [primary key]
19     title varchar
20     body text [note: 'Content of the post']
21     user_id integer
22     status varchar
23     created_at timestamp
24 }
25
26 Ref: posts.user_id > users.id // many-to-one
27
28 Ref: users.id < follows.following_user_id
29

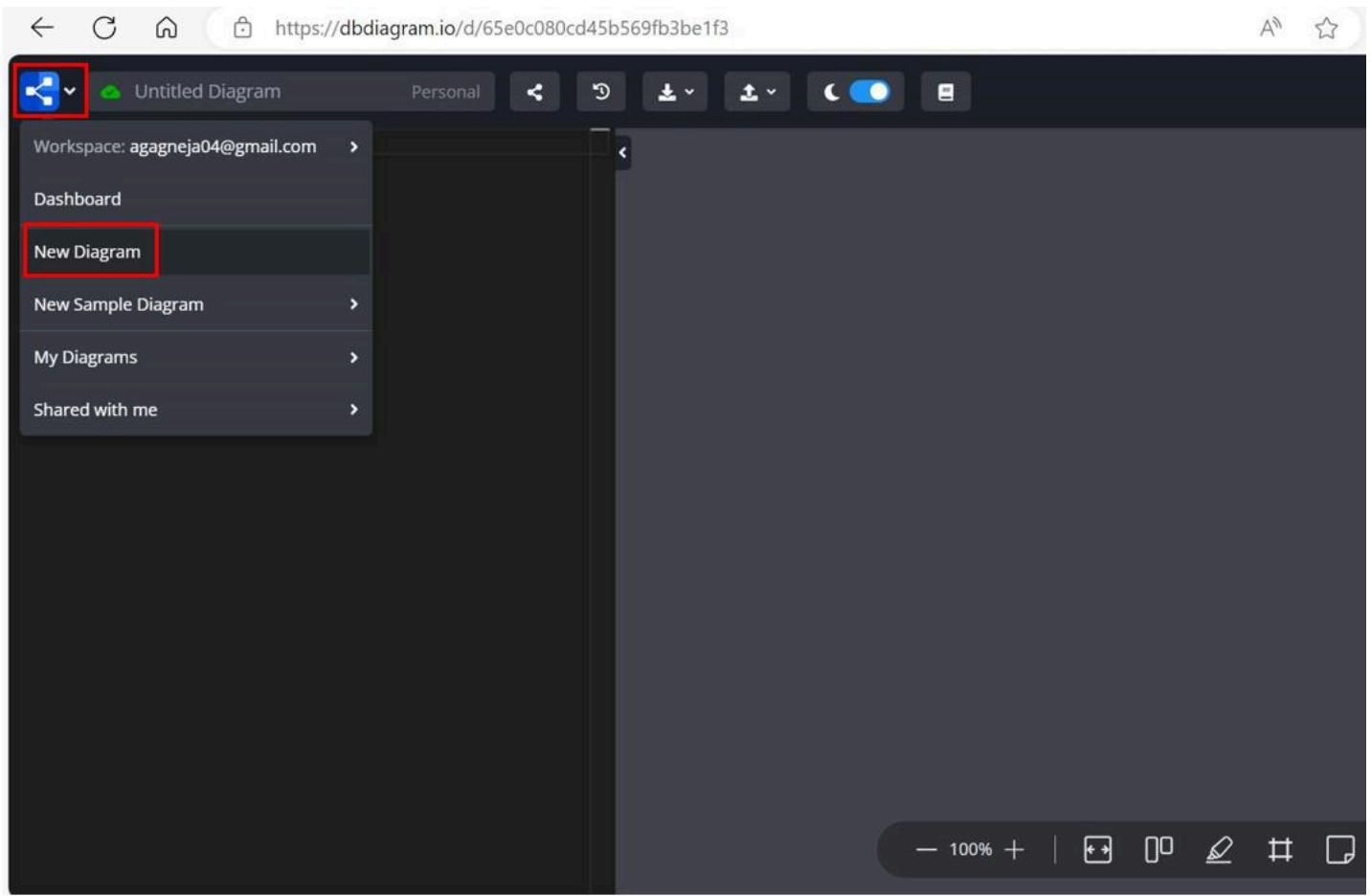
```

The diagram shows two tables: 'follows' and 'users'. The 'follows' table has columns 'following_user_id' (integer), 'followed_user_id' (integer), and 'created_at' (timestamp). The 'users' table has columns 'id' (primary key, integer), 'username' (varchar), 'role' (varchar), and 'created_at' (timestamp). There is a many-to-one relationship between 'posts' and 'users' via 'user_id', and a many-to-one relationship between 'follows' and 'users' via 'following_user_id'.

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.

A screenshot of a modal dialog box titled 'Import from MySQL'. At the top left is a file icon with the text 'Import from MySQL'. To the right is a 'Instructions' toggle switch which is turned on. The main content area is titled 'MySQL' and contains two numbered steps: '1. Install mysqldump.' and '2. In your terminal, run the following command (On Linux, you might need to add sudo before it):'. Below these steps is a code block with a black background and white text:

```
$ mysqldump -h <host> -u <username> -P <port> -p --no-data <database_name> > <output_path>
```

At the bottom left of the dialog is the word 'Example:' followed by a copy icon. At the bottom right are two buttons: 'Cancel' and 'Submit'. Above the 'Submit' button is a checked checkbox labeled 'Append converted DBML to the end'.

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

[Import from MySQL](#)Instructions [Upload .sql](#)

MySQL

1. Install mysqldump.

2. In your terminal, run the following command (On Linux, you might need to add `sudo` before it):

```
$ mysqldump -h <host> -u <username> -P <port> -p --no-data <database_name> > <output_path>
```

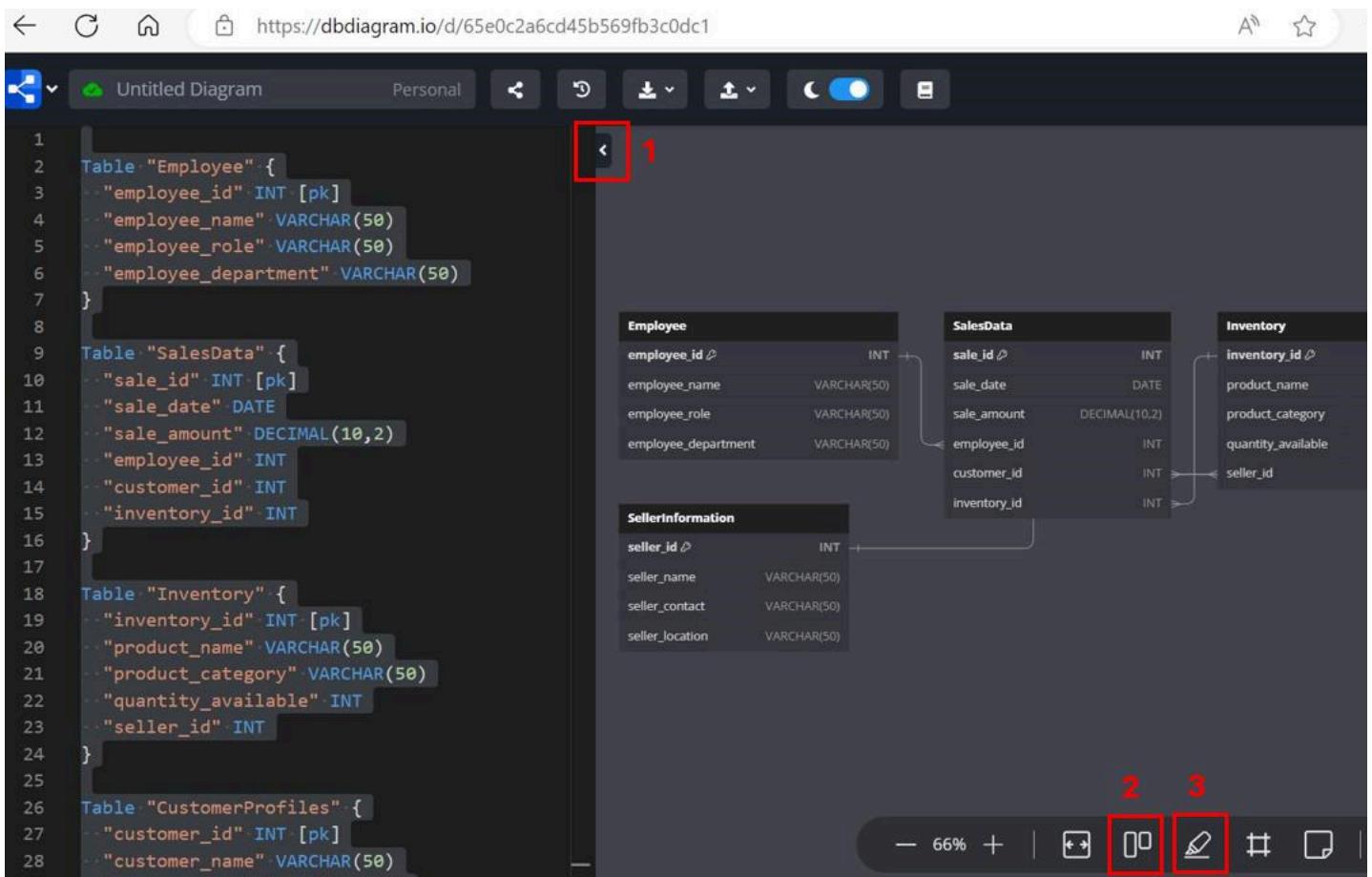
Example:

```
1 CREATE TABLE Employee (
2     employee_id INT PRIMARY KEY,
3     employee_name VARCHAR(50),
4     employee_role VARCHAR(50),
5     employee_department VARCHAR(50)
6 );
7
8 CREATE TABLE SalesData (
9     sale_id INT PRIMARY KEY,
10    sale_date DATE,
11    sale_amount DECIMAL(10, 2),
12    employee_id INT,
13    customer_id INT,
14    inventory_id INT,
15    FOREIGN KEY (employee_id) REFERENCES Employee(employee_id),
16    FOREIGN KEY (customer_id) REFERENCES CustomerProfiles(customer_id),
17    FOREIGN KEY (inventory_id) REFERENCES Inventory(inventory_id)
```

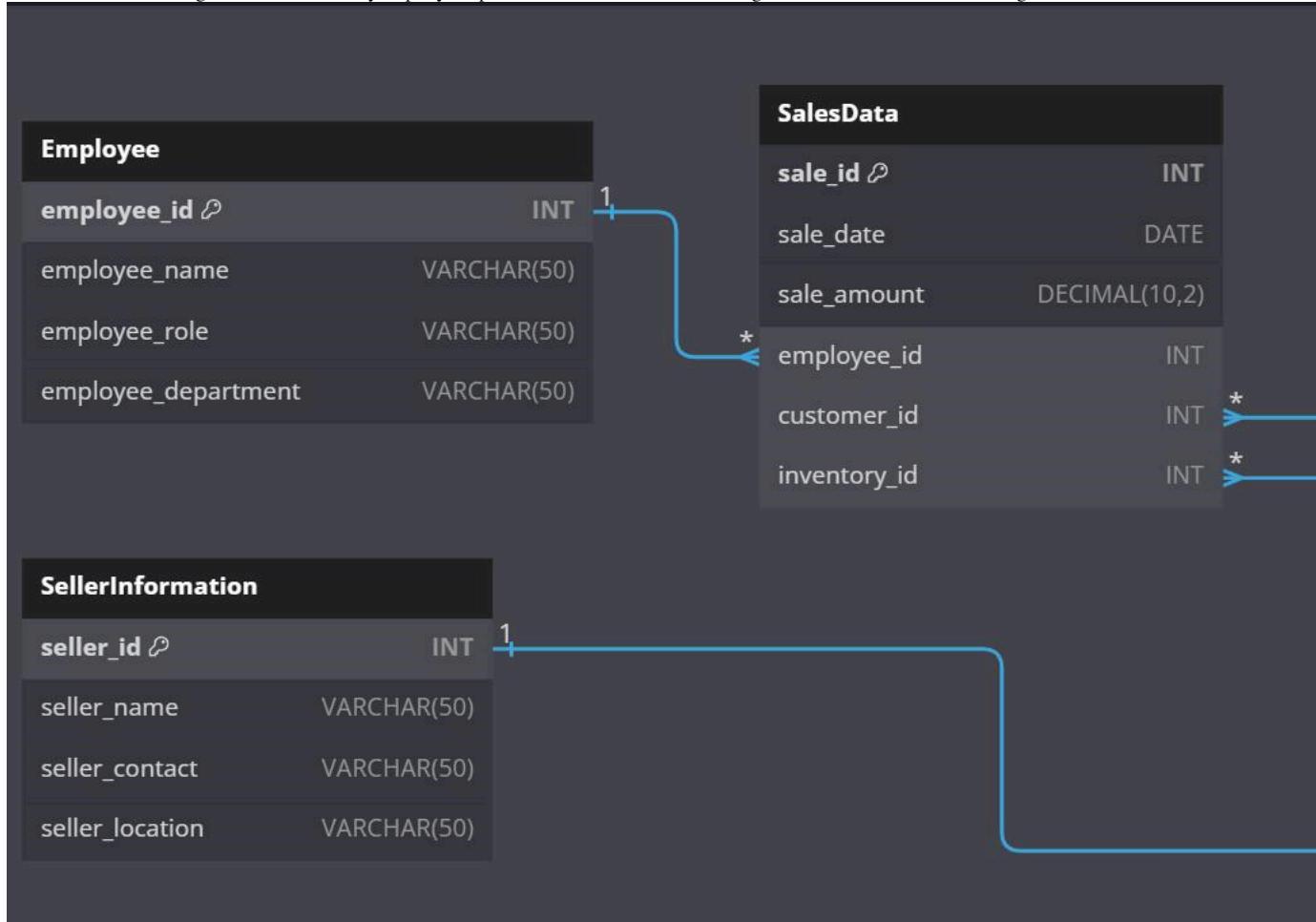
Append converted DBML to the end

[Cancel](#)[Submit](#)

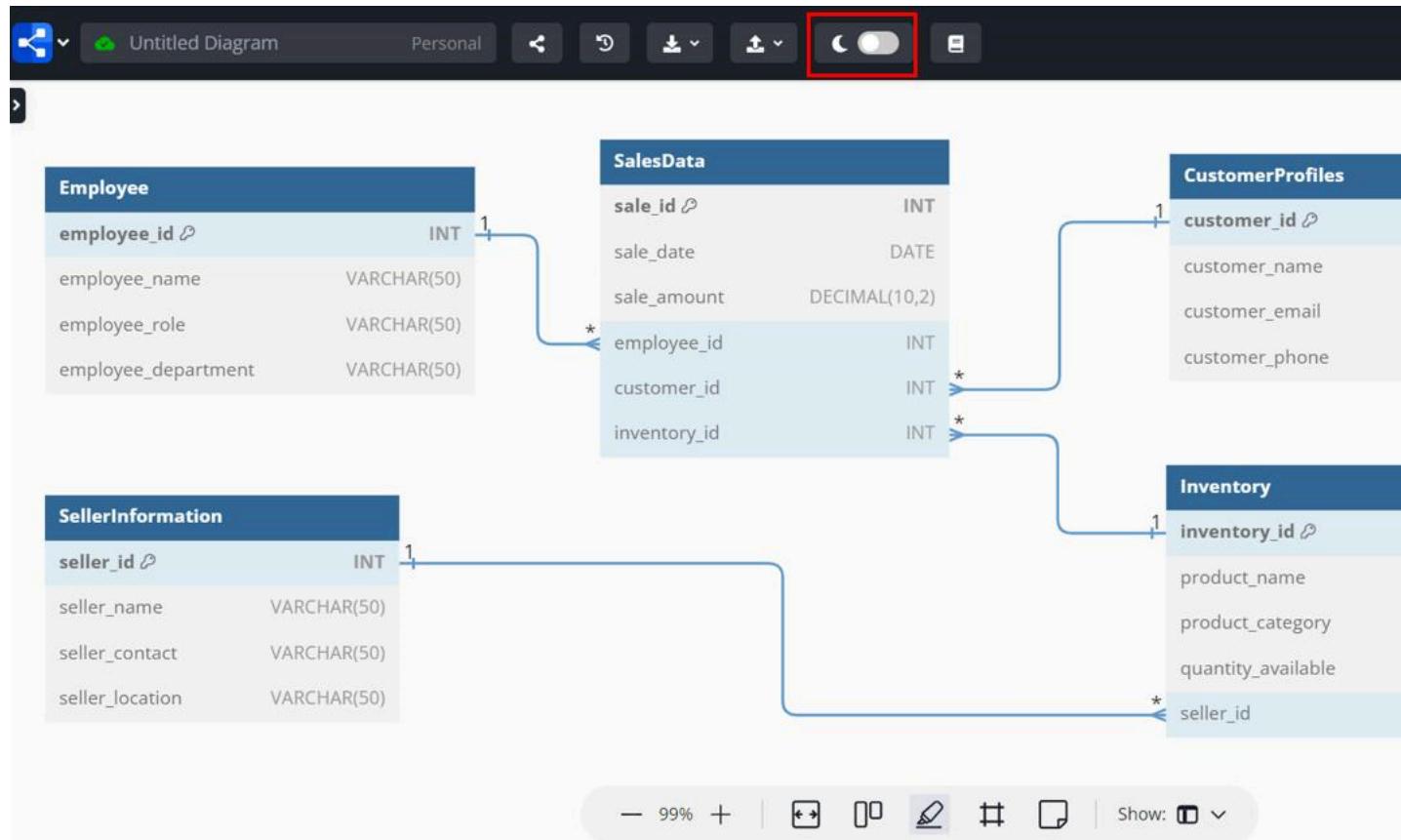
8. The resulting interface will have a modified code, which is being used to create the schema design. In sequence, press the tabs on the interface as shown in the image below. The first tab will close the coding interface, giving the resulting schema design a lot more room for visibility. The second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