Data Management, Warehousing, And Analytics

Lab2: Entity-Relationship Modelling (Summer 2023)

Submitted by: Arihant Dugar (B00917961)

GitLab repo: https://git.cs.dal.ca/dugar/csci5408_s23_b00917961_arihant_dugar/-/tree/main/Lab2

Summary:

In this Lab assignment, a restaurant database was developed with a focus on efficient data management. The initial step involved identifying the entities and attributes relevant to the restaurant, such as Restaurant, Menu, Customer, Order, and Employee. A conceptual, logical, and physical model was then designed to establish the database's structure, incorporating primary keys, foreign keys, and relationships. Using forward engineering, an Entity-Relationship Diagram (ERD) was generated to visualize the entity relationships. Finally, a schema was created, and tables were generated based on the identified entities, ensuring a well-organized and comprehensive restaurant database

Steps Performed:

1. Identifying the entities and attributes.
2. Create a conceptual model.
3. Create a logical model.
4. Create a physical model.
5. Create a new schema and ERD using 'Add Diagram' in mysql workbench.
6. Export the SQL script and the ERD.
7. Create the tables using forward engineering.

Lab Exercise:

1. Identify the entities and attributes for a restaurant :
   i. Restaurant
      - Id (PRIMARY KEY)
      - Name
      - Phone
      - Email
      - Address
      - OwnerId (FOREIGN KEY referencing OWNER entity)

   ii. Owner
      - Id (PRIMARY KEY)
      - Name
      - Phone

- Email

iii. Employee
- Id (PRIMARY KEY)
- Name
- Phone
- Email
- Designation
- Salary
- Address
- RestaurantId (FOREIGN KEY referencing RESTAURANT entity)

iv. Menu
- Id (PRIMARY KEY)
- Name
- Description
- RestaurantId (FOREIGN KEY referencing RESTAURANT entity)

v. Dish
- Id (PRIMARY KEY)
- Name
- Description
- Price
- Ingredients
- SpicyLevel
- MenuId (FOREIGN KEY referencing MENU entity)

vi. Table
- Id (PRIMARY KEY)
- Number
- Capacity
- RestaurantId (FOREIGN KEY referencing RESTAURANT entity)

vii. Reservation
- Id (PRIMARY KEY)
- CustomerId (FOREIGN KEY referencing CUSTOMER entity)
- TableId (FOREIGN KEY referencing TABLE entity)
- Date
- Time

viii. Customer
- Id (PRIMARY KEY)
- Name
- Phone
- Email

ix. Order
- Id (PRIMARY KEY)
- TableId (FOREIGN KEY referencing TABLE entity)
- EmployeeId (FOREIGN KEY referencing EMPLOYEE entity)
- Date
- Amount

2. Design a basic conceptual, logical and physical model
    i. Conceptual Model –
    A conceptual model has entities and relationships associated with them. Below is the conceptual diagram for restaurant [1].

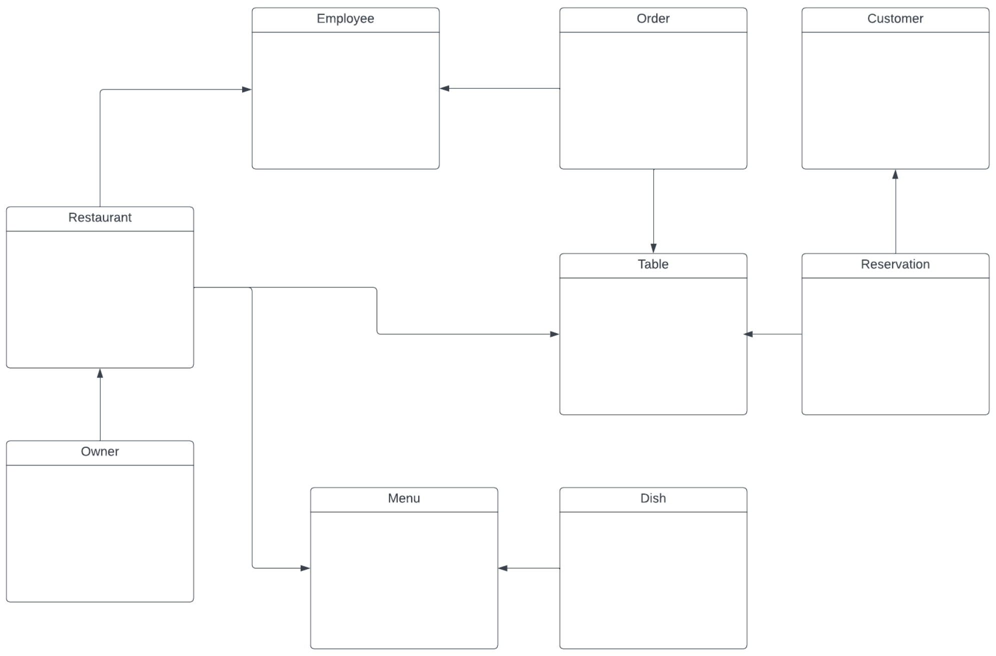

Figure 2.1: The conceptual model diagram for restaurant

    ii. Logical Model –
    A logical model contains the attributes, primary and foreign keys.

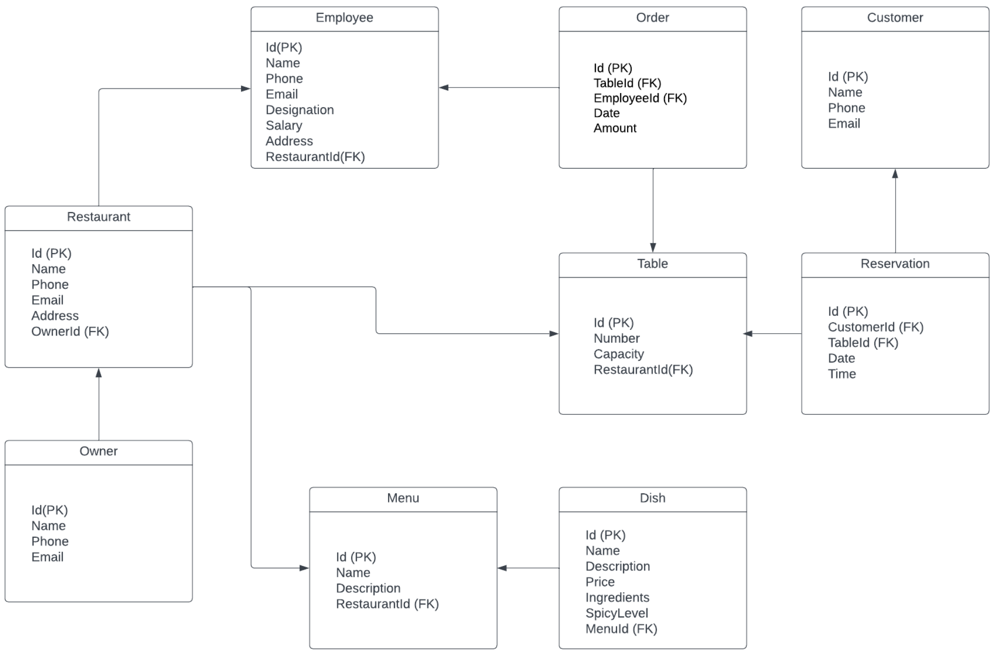

Figure 2.2: The logical model diagram for restaurant

iii. Physical Model –
A physical model contains the column names and the column data types as well mentioned.

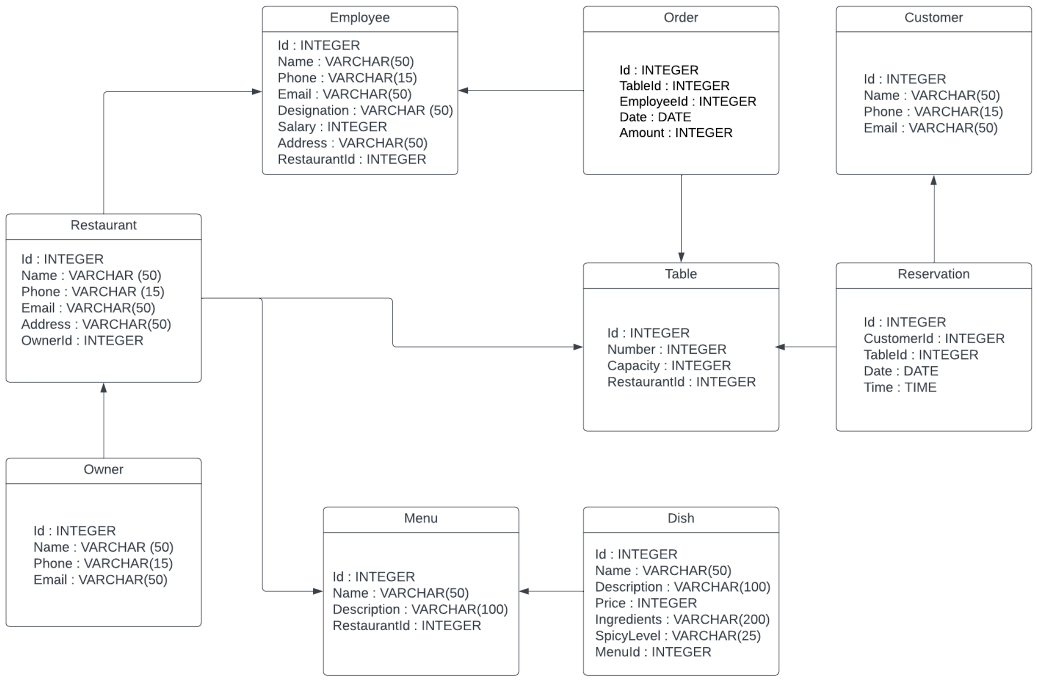

Figure 2.1: The physical model diagram for restaurant

3. ERD Created using Forward Engineering –

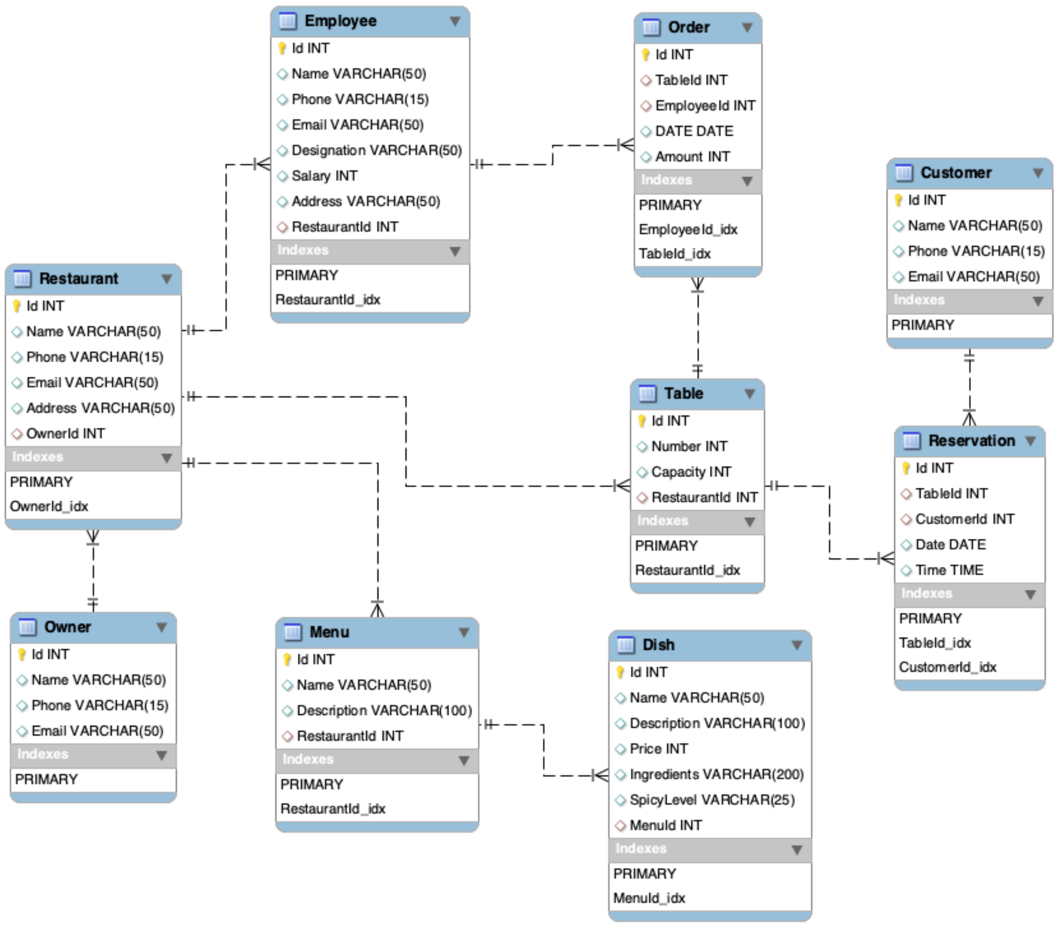

Figure 3: The ERD for restaurant created while performing forward engineering

Steps performed to generate the ERD using forward engineering:

i. Create a new schema and click on ‘Add Diagram’ to create a new ERD.

ii. Place a new table and double click on it to edit the properties, such as table name, add column fields and data types.

iii. Similarly, create all the tables and add the foreign key by going to the foreign key tab for each table. This will create the relationship between tables.

iv. Once the ERD is created, press Command+G to begin forward engineering.

v. All the DDL statements would be created and click Next to generate the structure in the schema. There is also an option to export the query to a SQL file and save it.

Note: The Queries exported are present in the file **Restaurant.sql** uploaded along with this report for reference.

4. What is reverse engineering?

   Reverse engineering of an information system means obtaining the information on software solution architecture via its implementation, that is via code reproduction [2]. Reverse engineering in SQL is the practice of deriving or deducing the arrangement and connections within a database system based on an existing database. This process entails examining the structure of the database, including its schema, tables, columns, constraints, and relationships, in order to gain a comprehensive understanding of the fundamental design principles at play.

   Reverse engineering can be performed in different ways, and some common approaches are using –
   - Data Profiling
   - Data Modeling tools
   - Database Schema scripts

References:

[1] "Data Modeling Levels." 1Keydata.com. Available: https://www.1keydata.com/datawarehousing/data-modeling-levels.html [Accessed: 20-May-2023].

[2] Devart. "Database Reverse Engineering: How to Generate a Data Model from an Existing Database." Devart Blog. Available: https://blog.devart.com/database-reverse-engineering.html . [Accessed: 20-May-2023].