CSCI 5408

Data Management, Warehousing, And Analytics

Lab 2: Entity-Relationship Modelling

Prepared By

Bhavisha Oza (B00935827)

Summary

The ERD design goes through phases like conceptual, logical, and physical modeling, aiding in requirement gathering, designing, and debugging [2]. Chen and Crow's foot models are commonly used ERD notations. Task is to prepare the ERD for Restaurant. Here first 2 phases (Conceptual and Logical) of ER diagram are prepared with Chen's model and physical phase is created with MySQL Workbench which has forward engineering to generate database code from an ERD.

Steps followed:

- Identified the entities and attributes for the Restaurant.
- Designed a basic conceptual, logical, physical model on paper.
- Created a basic conceptual, logical model using draw.io tool [2, 3].
- Create an ERD in MySQL workbench using forward engineering [1, 4].
- Exported the ERD and the queries [2].

Lab exercise:

1) **Entities and Attributes of Restaurant System:**
   a) Customer Entity:
      i) Customer ID (Integer): Unique identifier for each customer.
      ii) Customer Name (Text): The full name of the customer.
      iii) Customer Email (Text): Customer's email address for communication.
      iv) Customer Contact number (Text): Phone number of the customer.
      v) Customer Address (Text): Residential address of the customer.

   b) Employee Entity:
      i) Employee ID (Integer): Unique identifier for each employee.
      ii) Employee Name (Text): The full name of the employee.
      iii) Employee Gender (Text): The gender of the employee.
      iv) Employee Email (Text): Employee's email address for communication.
      v) Employee Contact number (Text): Phone number of the employee.
      vi) Employee Address (Text): Residential address of the employee.
      vii) Employee Date of Birth (Date/Time): Birthdate of the employee
      viii) Employee Age (Integer): The age of the employee
      ix) Employee Date of Joining (Date/Time): Joining date of the employee
      x) Employee Designation (Text): Employee's position in the restaurant

   c) Employee Dependent Entity:
      i) Dependent Name (Text): The full name of the dependent.
      ii) Dependent Gender (Text): Dependent's gender.
      iii) Dependent Date of Birth (Date/Time): Birthdate of the dependent.
      iv) Dependent Relationship (Text): The relation of dependent with the employee.

   d) Restaurant Entity:
      i) Restaurant ID (Integer): Unique identifier for each restaurant.
      ii) Restaurant Name (Text): The name of the restaurant.

iii) Restaurant Email (Text): Restaurant's email address for communication.
iv) Restaurant Contact number (Text): Phone number of the restaurant.
v) Restaurant Address (Text): Address of the restaurant.
vi) Restaurant Working hours (Text): Operating time of the restaurant.

e) Branches Entity:
   i) Branch ID (Integer): Unique identifier for each restaurant's branch.
   ii) Branch City Name (Text): The city name of the restaurant's branch.
   iii) Branch Location (Text): Restaurant's branch address.
   iv) Branch Contact number (Text): Phone number of the restaurant's branch.

f) Menu Entity:
   i) Menu ID (Integer): Unique identifier for each menu.
   ii) Menu Name (Text): The name of the menu.
   iii) Menu Price (Decimal): Costing of the menu items
   iv) Menu Ingredients (Text): Menu's list of ingredients

g) Category Entity:
   i) Category ID (Integer): identifier for each cuisine.
   ii) Category Name (Text): The name of the cuisine from the menu.

h) Orders Entity:
   i) Order ID (Integer): Unique identifier for each order.
   ii) Order Date and Time (Text): Date and time of the order placed.
   iii) Order Quantity (Text): Total quantity of the placed order.
   iv) Order Total Bill (Decimal): Summation of the amount of money for the placed order.
   v) Delivery Time (Date/Time): Estimated/Actual time for the delivery of the order.
   vi) Delay (Date/Time): Delay in order based on order date and deliver time.
   vii) Order Status (Text): The status of the order (e.g., "Processing," "Delivered").

i) Payment Entity:
   i) Payment ID (Integer): Unique identifier for each order's payment.
   ii) Email (Text): Email id used for the payment.
   iii) Payment Type (Text): Mode of payment like Card or Cash

2) Conceptual Model for Restaurant System:

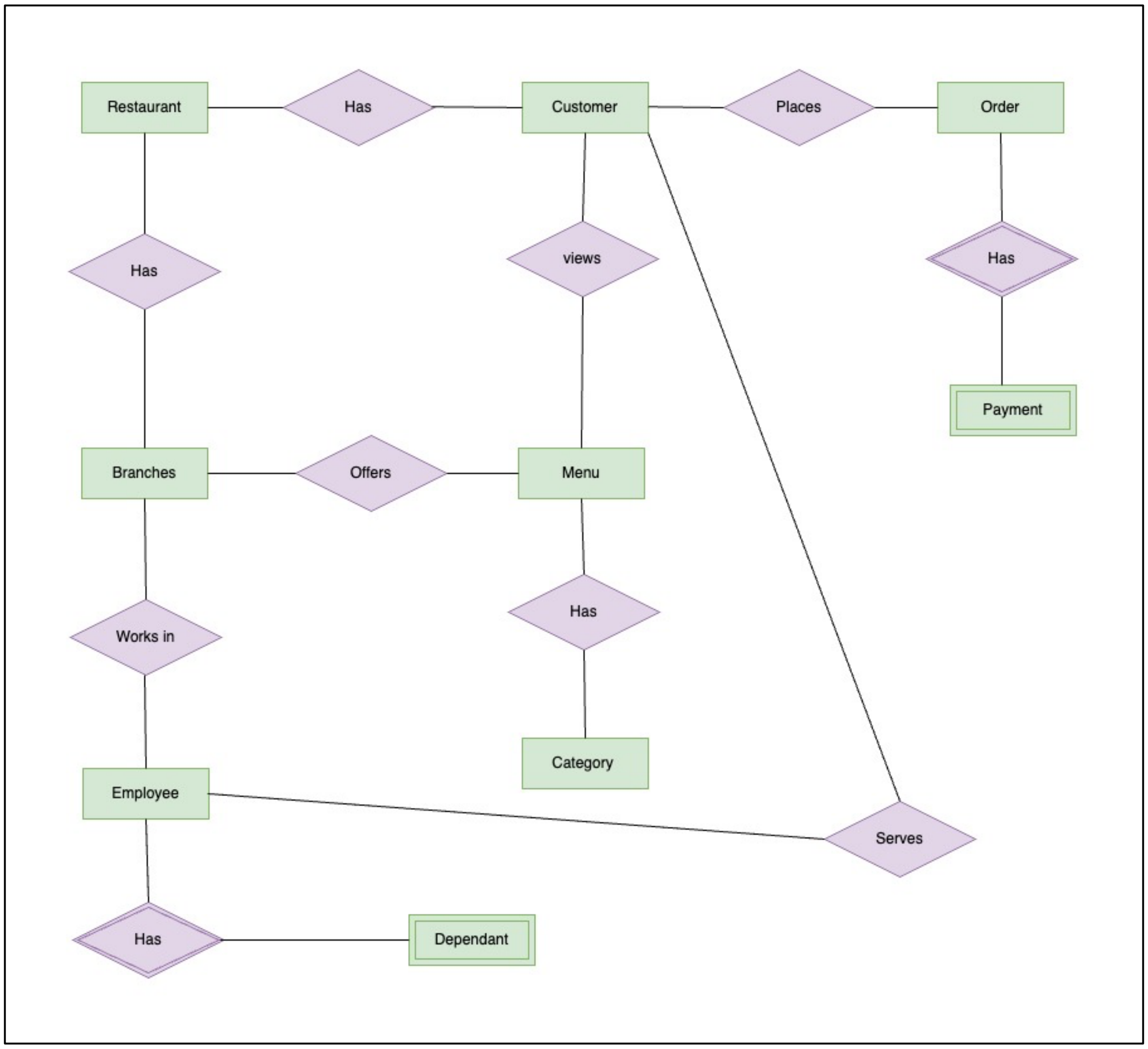

Figure 1: Identified Entities and Relationships in conceptual model for Restaurant.

Explanation: There are **9 entities** defined for the Restaurant. The defined relationships among them are as follows:

- Restaurant has Customer.
- Restaurant has Branches.
- Customer places Order
- Customer views Menu
- Branches offers Menu.
- Employee works in Branches.
- Employee serves Customer.
- Employee has Dependent.
- Menu has Category.
- Order has Payment.

3) Logical Model for Restaurant System:

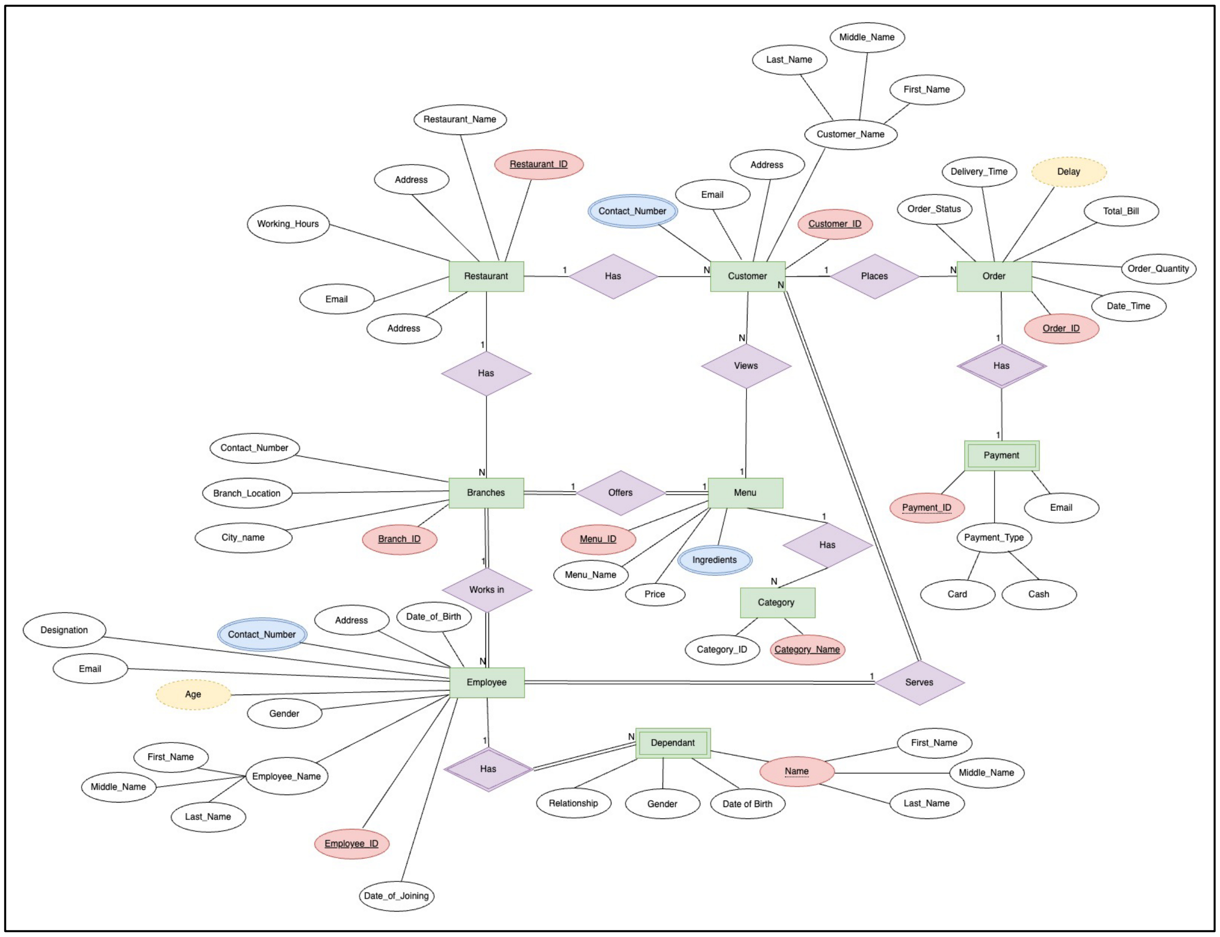

Figure 2: Identified Cardinalities, Prime keys, and attributes in logical model for Restaurant.

Explanation: The cardinalities between the entities are as follows:

- 1 Restaurant has many customers.
- 1 Restaurant has many branches.
- Many employees work in 1 branch.
- 1 Employee has many dependents.
- 1 Employee serves many customers.
- 1 Customer places many orders.
- Many customers view 1 Menu.
- 1 Branch offers 1 Menu.
- 1 Menu has many categories.
- 1 Order has 1 payment.

4) Physical Model for Restaurant System:

Step-1: Opened MySQL Workbench and clicked on File > New Model.
Step-2: Created a new schema by clicking on '+'.
Step-3: Click on 'Add Diagram' to create a new ERD.
Step-4: Clicked on 'Place a New Table' option and clicked anywhere to create a table.
Step-5: Changed the table name and filled the column name of the table.
Step-6: Created all required tables.
Step-7: Given all references for the foreign key for each table.
Step-8: Clicked on Database > Forward Engineer > Followed the instruction prompted > Saved the SQL file with generated DDL statements.
Step-9: To cross check, if the tables are created or not, opened the SQL Editor > Checked the tables with the command "SHOW tables;"

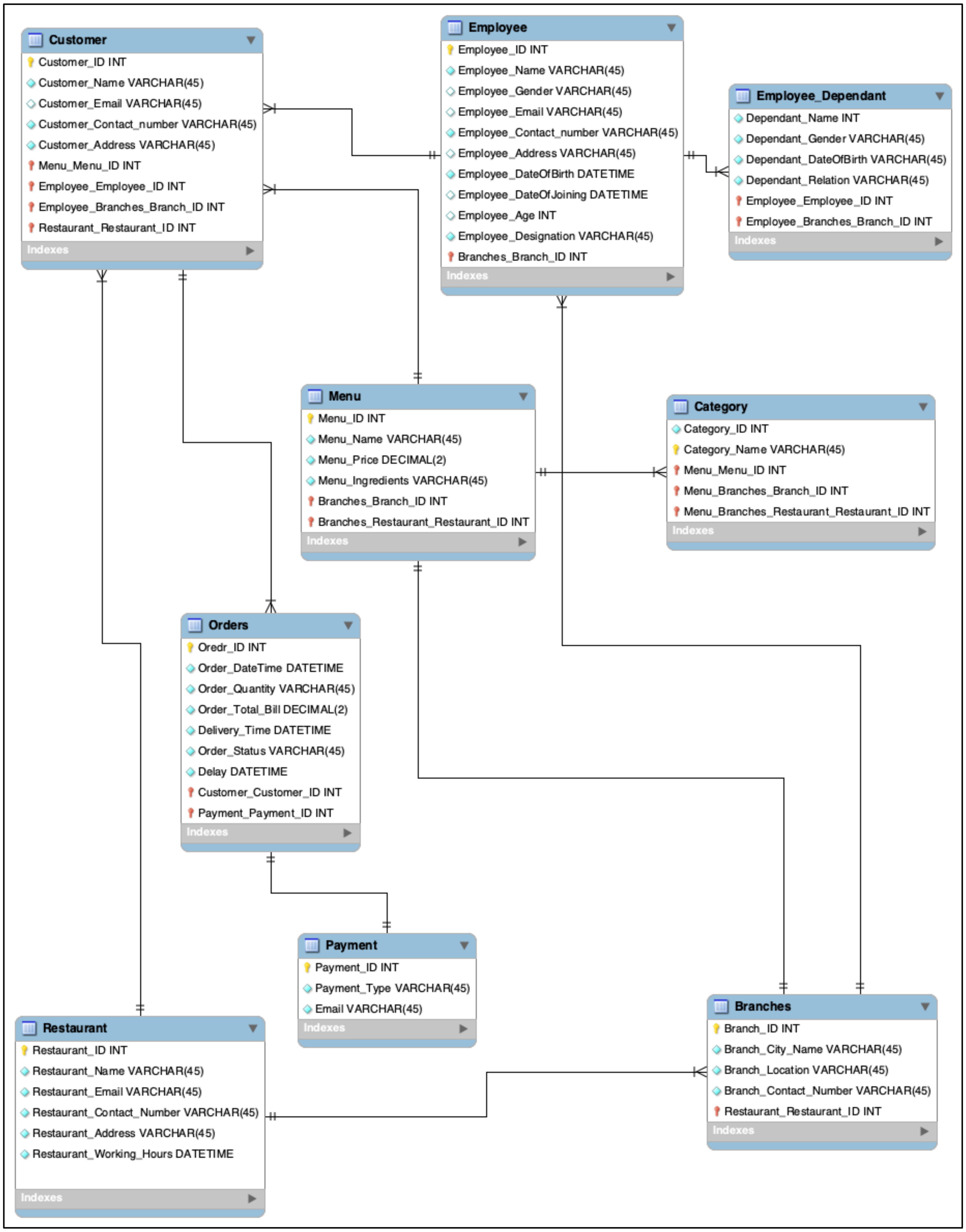
Customer
Customer_ID INT
Customer_Name VARCHAR(45)
Customer_Email VARCHAR(45)
Customer_Contact_number VARCHAR(45)
Customer_Address VARCHAR(45)
Menu_Menu_ID INT
Employee_Employee_ID INT
Employee_Branches_Branch_ID INT
Restaurant_Restaurant_ID INT
Indexes
Employee
Employee_ID INT
Employee_Name VARCHAR(45)
Employee_Gender VARCHAR(45)
Employee_Email VARCHAR(45)
Employee_Contact_number VARCHAR(45)
Employee_Address VARCHAR(45)
Employee_DateOfBirth DATETIME
Employee_DateOfJoining DATETIME
Employee_Age INT
Employee_Designation VARCHAR(45)
Branches_Branch_ID INT
Indexes
Employee_Dependant
Dependant_Name INT
Dependant_Gender VARCHAR(45)
Dependant_DateOfBirth VARCHAR(45)
Dependant_Relation VARCHAR(45)
Employee_Employee_ID INT
Employee_Branches_Branch_ID INT
Indexes
Menu
Menu_ID INT
Menu_Name VARCHAR(45)
Menu_Price DECIMAL(2)
Menu_Ingredients VARCHAR(45)
Branches_Branch_ID INT
Branches_Restaurant_Restaurant_ID INT
Indexes
Category
Category_ID INT
Category_Name VARCHAR(45)
Menu_Menu_ID INT
Menu_Branches_Branch_ID INT
Menu_Branches_Restaurant_Restaurant_ID INT
Indexes
Orders
Oredr_ID INT
Order_DateTime DATETIME
Order_Quantity VARCHAR(45)
Order_Total_Bill DECIMAL(2)
Delivery_Time DATETIME
Order_Status VARCHAR(45)
Delay DATETIME
Customer_Customer_ID INT
Payment_Payment_ID INT
Indexes
Payment
Payment_ID INT
Payment_Type VARCHAR(45)
Email VARCHAR(45)
Indexes
Restaurant
Restaurant_ID INT
Restaurant_Name VARCHAR(45)
Restaurant_Email VARCHAR(45)
Restaurant_Contact_Number VARCHAR(45)
Restaurant_Address VARCHAR(45)
Restaurant_Working_Hours DATETIME
Indexes
Branches
Branch_ID INT
Branch_City_Name VARCHAR(45)
Branch_Location VARCHAR(45)
Branch_Contact_Number VARCHAR(45)
Restaurant_Restaurant_ID INT
Indexes

Figure 3: ER Diagram for Restaurant using Forward Engineering

The queries after creating ERD with Forward Engineering:

```
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE
,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

-- -----------------------------------------------------
-- Schema lab_2
-- -----------------------------------------------------

-- -----------------------------------------------------
-- Schema lab_2
-- -----------------------------------------------------
CREATE SCHEMA IF NOT EXISTS `lab_2` ;
USE `lab_2` ;

-- -----------------------------------------------------
-- Table `lab_2`.`Restaurant`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `lab_2`.`Restaurant` (
  `Restaurant_ID` INT NOT NULL,
  `Restaurant_Name` VARCHAR(45) NOT NULL,
  `Restaurant_Email` VARCHAR(45) NOT NULL,
  `Restaurant_Contact_Number` VARCHAR(45) NOT NULL,
  `Restaurant_Address` VARCHAR(45) NOT NULL,
  `Restaurant_Working_Hours` DATETIME NOT NULL,
  PRIMARY KEY (`Restaurant_ID`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `lab_2`.`Branches`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `lab_2`.`Branches` (
  `Branch_ID` INT NOT NULL,
  `Branch_City_Name` VARCHAR(45) NOT NULL,
  `Branch_Location` VARCHAR(45) NOT NULL,
  `Branch_Contact_Number` VARCHAR(45) NOT NULL,
  `Restaurant_Restaurant_ID` INT NOT NULL,
  PRIMARY KEY (`Branch_ID`, `Restaurant_Restaurant_ID`),
  INDEX `fk_Branches_Restaurant1_idx` (`Restaurant_Restaurant_ID` ASC) VISIBLE,
  CONSTRAINT `fk_Branches_Restaurant1`
    FOREIGN KEY (`Restaurant_Restaurant_ID`)
    REFERENCES `lab_2`.`Restaurant` (`Restaurant_ID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `lab_2`.`Menu`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `lab_2`.`Menu` (
  `Menu_ID` INT NOT NULL,
  `Menu_Name` VARCHAR(45) NOT NULL,
  `Menu_Price` DECIMAL(2) NOT NULL,
  `Menu_Ingredients` VARCHAR(45) NOT NULL,
  `Branches_Branch_ID` INT NOT NULL,
  `Branches_Restaurant_Restaurant_ID` INT NOT NULL,
  PRIMARY KEY (`Menu_ID`, `Branches_Branch_ID`, `Branches_Restaurant_Restaurant_ID`),
  UNIQUE INDEX `Menu_Ingredients_UNIQUE` (`Menu_Ingredients` ASC) VISIBLE,
  INDEX `fk_Menu_Branches1_idx` (`Branches_Branch_ID` ASC,
`Branches_Restaurant_Restaurant_ID` ASC) VISIBLE,
  CONSTRAINT `fk_Menu_Branches1`
    FOREIGN KEY (`Branches_Branch_ID` , `Branches_Restaurant_Restaurant_ID`)
    REFERENCES `lab_2`.`Branches` (`Branch_ID` , `Restaurant_Restaurant_ID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `lab_2`.`Employee`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `lab_2`.`Employee` (
  `Employee_ID` INT NOT NULL,
  `Employee_Name` VARCHAR(45) NOT NULL,
  `Employee_Gender` VARCHAR(45) NULL,
  `Employee_Email` VARCHAR(45) NULL,
  `Employee_Contact_number` VARCHAR(45) NOT NULL,
  `Employee_Address` VARCHAR(45) NULL,
  `Employee_DateOfBirth` DATETIME NOT NULL,
  `Employee_DateOfJoining` DATETIME NULL,
  `Employee_Age` INT NULL,
  `Employee_Designation` VARCHAR(45) NOT NULL,
  `Branches_Branch_ID` INT NOT NULL,
  PRIMARY KEY (`Employee_ID`, `Branches_Branch_ID`),
  UNIQUE INDEX `Employee_Email_UNIQUE` (`Employee_Email` ASC) VISIBLE,
  INDEX `fk_Employee_Branches1_idx` (`Branches_Branch_ID` ASC) VISIBLE,
  CONSTRAINT `fk_Employee_Branches1`
    FOREIGN KEY (`Branches_Branch_ID`)
    REFERENCES `lab_2`.`Branches` (`Branch_ID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `lab_2`.`Customer`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `lab_2`.`Customer` (
  `Customer_ID` INT NOT NULL,
  `Customer_Name` VARCHAR(45) NOT NULL,
  `Customer_Email` VARCHAR(45) NULL,
  `Customer_Contact_number` VARCHAR(45) NOT NULL,
  `Customer_Address` VARCHAR(45) NOT NULL,
  `Menu_Menu_ID` INT NOT NULL,
  `Employee_Employee_ID` INT NOT NULL,
  `Employee_Branches_Branch_ID` INT NOT NULL,
  `Restaurant_Restaurant_ID` INT NOT NULL,
  PRIMARY KEY (`Customer_ID`, `Menu_Menu_ID`, `Employee_Employee_ID`,
`Employee_Branches_Branch_ID`, `Restaurant_Restaurant_ID`),
  UNIQUE INDEX `Customer_Email_UNIQUE` (`Customer_Email` ASC) VISIBLE,
  INDEX `fk_Customer_Menu1_idx` (`Menu_Menu_ID` ASC) VISIBLE,
  INDEX `fk_Customer_Employee1_idx` (`Employee_Employee_ID` ASC,
`Employee_Branches_Branch_ID` ASC) VISIBLE,
  INDEX `fk_Customer_Restaurant1_idx` (`Restaurant_Restaurant_ID` ASC) VISIBLE,
  CONSTRAINT `fk_Customer_Menu1`
    FOREIGN KEY (`Menu_Menu_ID`)
    REFERENCES `lab_2`.`Menu` (`Menu_ID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Customer_Employee1`
    FOREIGN KEY (`Employee_Employee_ID` , `Employee_Branches_Branch_ID`)
    REFERENCES `lab_2`.`Employee` (`Employee_ID` , `Branches_Branch_ID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Customer_Restaurant1`
    FOREIGN KEY (`Restaurant_Restaurant_ID`)
    REFERENCES `lab_2`.`Restaurant` (`Restaurant_ID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- -----------------------------------------------------
-- Table `lab_2`.`Employee_Dependant`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `lab_2`.`Employee_Dependant` (
  `Dependant_Name` INT NOT NULL,
  `Dependant_Gender` VARCHAR(45) NOT NULL,
  `Dependant_DateOfBirth` VARCHAR(45) NOT NULL,
  `Dependant_Relation` VARCHAR(45) NOT NULL,
  `Employee_Employee_ID` INT NOT NULL,
  `Employee_Branches_Branch_ID` INT NOT NULL,
  PRIMARY KEY (`Employee_Employee_ID`, `Employee_Branches_Branch_ID`),
  CONSTRAINT `fk_Employee_Dependant_Employee1`
    FOREIGN KEY (`Employee_Employee_ID` , `Employee_Branches_Branch_ID`)
    REFERENCES `lab_2`.`Employee` (`Employee_ID` , `Branches_Branch_ID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `lab_2`.`Category`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `lab_2`.`Category` (
  `Category_ID` INT NOT NULL,
  `Category_Name` VARCHAR(45) NOT NULL,
  `Menu_Menu_ID` INT NOT NULL,
  `Menu_Branches_Branch_ID` INT NOT NULL,
  `Menu_Branches_Restaurant_Restaurant_ID` INT NOT NULL,
  PRIMARY KEY (`Category_Name`, `Menu_Menu_ID`, `Menu_Branches_Branch_ID`,
`Menu_Branches_Restaurant_Restaurant_ID`),
  INDEX `fk_Category_Menu1_idx` (`Menu_Menu_ID` ASC, `Menu_Branches_Branch_ID`
ASC, `Menu_Branches_Restaurant_Restaurant_ID` ASC) VISIBLE,
  CONSTRAINT `fk_Category_Menu1`
    FOREIGN KEY (`Menu_Menu_ID` , `Menu_Branches_Branch_ID` ,
`Menu_Branches_Restaurant_Restaurant_ID`)
    REFERENCES `lab_2`.`Menu` (`Menu_ID` , `Branches_Branch_ID` ,
`Branches_Restaurant_Restaurant_ID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `lab_2`.`Payment`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `lab_2`.`Payment` (
  `Payment_ID` INT NOT NULL,
  `Payment_Type` VARCHAR(45) NOT NULL,
  `Email` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`Payment_ID`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `lab_2`.`Orders`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `lab_2`.`Orders` (
  `Oredr_ID` INT NOT NULL,
  `Order_DateTime` DATETIME NOT NULL,
  `Order_Quantity` VARCHAR(45) NOT NULL,
  `Order_Total_Bill` DECIMAL(2) NOT NULL,
  `Delivery_Time` DATETIME NOT NULL,
  `Order_Status` VARCHAR(45) NOT NULL,
  `Delay` DATETIME NOT NULL,
  `Customer_Customer_ID` INT NOT NULL,
  `Payment_Payment_ID` INT NOT NULL,
  PRIMARY KEY (`Oredr_ID`, `Customer_Customer_ID`, `Payment_Payment_ID`),
  INDEX `fk_Orders_Customer1_idx` (`Customer_Customer_ID` ASC) VISIBLE,
  INDEX `fk_Orders_Payment1_idx` (`Payment_Payment_ID` ASC) VISIBLE,
  CONSTRAINT `fk_Orders_Customer1`
    FOREIGN KEY (`Customer_Customer_ID`)
    REFERENCES `lab_2`.`Customer` (`Customer_ID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Orders_Payment1`
    FOREIGN KEY (`Payment_Payment_ID`)
    REFERENCES `lab_2`.`Payment` (`Payment_ID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

References:

[1] "MySQL Community Downloads," *MySQL* [Online]. Available: https://dev.mysql.com/downloads/workbench/ [Accessed: May 10, 2023].

[2] "Lab-2," *Brightspace Dalhousie University* [Online]. Available: https://dal.brightspace.com/d2l/le/content/271677/viewContent/3628976/View [Accessed: May 17, 2023].

[3] "Flowchart Maker & Online Diagram Software," *Draw.io* [Online]. Available: https://app.diagrams.net/ [Accessed: May 17, 2023].

[4] "MySQL Workbench Manual:: 9.4.1.1 Forward Engineering Using an SQL Script," *MySQL*. [Online]. Available: https://dev.mysql.com/doc/workbench/en/wb-forward-engineering-sql-scripts.html. [Accessed: May 16, 2023].