****

**UNIVERSITY OF TEXAS AT DALLAS**

**JINDAL SCHOOL OF MANAGEMENT**

**“RESTAURANT DATABASE”**

**TECHNICAL REPORT**

**BUAN 6320.0W2**

**AKSHAR KUMAR H**

**PROF. GASAN ELKHODARI**

**Table of Contents**

[Introduction 3](#_Toc165432784)

[Assumptions and Special Considerations 4](#_Toc165432785)

[Assumptions 4](#_Toc165432786)

[Constraints 4](#_Toc165432787)

[Requirements Definition Document 5](#_Toc165432788)

[Entity and Attribute Description 5](#_Toc165432789)

[Relationship and Cardinality Description 6](#_Toc165432790)

[Outputs 8](#_Toc165432791)

[Queries 12](#_Toc165432792)

# Introduction

The purpose of this technical report is to document the design and implementation of the database system for a Restaurant. This database serves as a critical component to manage and organize customer orders, employees, Restaurant branches and customers.

**Objective**

The goal is to ensure efficient data storage, retrieval, and management to support the functionalities of the Restaurant.

# Assumptions and Special Considerations

## Assumptions

In our system, we assume that each entity has a unique serial number (Primary Key), thus making it a unique product. the ERD is for a single database for one or more restaurants under the same administration.

## Constraints

In terms of constraints/deviations from the lab project requirements, we have ensured that all requirements are met, and we have avoided all M: M relationships to avoid hurdles going forward.

# Requirements Definition Document

## Entity and Attribute Description

***Entities***

Entity Name: RESTAURANT

Entity Description: The restaurant entity is a business that provides food and drinks.

Main Attributes of RESTAURANT

Restaurant\_ID: (Primary Key) A unique identifier of the restaurant to establish a unique reference for the Restaurant, ensuring data integrity, and allowing efficient data retrieval, and referencing.

Restaurant\_Name: The name of the Restaurant

Restaurant\_Location: The location of the restaurant

Restaurant\_Cuisine: The cuisine style offered by the restaurant.

Restaurant\_Hours: The working hours of the restaurant.

Entity Name: EMPLOYEE

Entity Description: The people who work at the restaurant are represented by the employee entity.

Main Attributes of EMPLOYEE

Employee\_ID: (Primary Key) This attribute establishes a unique identifier for each employee.

Restaurant\_ID: (Foreign Key) This attribute is used to establish a unique reference to the restaurant, ensuring data integrity, and allowing efficient data retrieval, and referencing.

Employee\_Role: The role of an employee in the restaurant

Employee\_Name: The name of the employee

Employee\_Salary: The salary details of each employee

Entity Name: ORDER

Entity Description: An order entity is a representation of a customer's meal order transaction.

Main Attributes of ORDER

Order\_ID: (Primary Key) This attribute establishes a unique identifier for the order.

Employee\_ID: (Foreign Key) This attribute is used to establish a unique reference for each employee, ensuring data integrity, and allowing efficient data retrieval, and referencing.

Order\_Total: Identifies the total of the order.

Order\_Table\_Number: Identifies the table number of the order.

Order\_Payment: Identifies the payment method of the order.

Order\_Status: States the order's status (ordered, preparing, served, paid) is shown by the order status

Entity Name: CUSTOMER

Entity Description: A customer entity is a person who makes restaurant purchases of meals.

Main Attributes of CUSTOMER

Customer\_ID: (Primary Key) This attribute establishes a unique identifier for the order.

Order\_ID: (Foreign Key) This attribute is used to establish a unique reference for each order, ensuring data integrity, and allowing efficient data retrieval, and referencing.

Customer\_Contact: identifies the contact number of the customer.

Customer\_Email: The email details of the customer.

Customer\_Name: Identifies the name of the customer.

## Relationship and Cardinality Description

Relationship: Restaurant to employee (“WORKS”)

Business rule: every employer is employed by a single restaurant. Restaurants may have several staff members.

Cardinality: 1:M

Relationship: customer to order (“PLACES”)

Business rule: a customer can place several orders. Every order is placed by a single client.

Cardinality: 1:M

Relationship: employee to order (“HANDLES”)

Business rule: an employee can handle multiple orders; one employee must handle each order

Cardinality: 1:M

# DATA DEFINITION LANGUAGE (DDL)

-- Drop Tables if they exist

DROP TABLE IF EXISTS Restaurant CASCADE;

DROP TABLE IF EXISTS Employee CASCADE;

DROP TABLE IF EXISTS Orders CASCADE;

DROP TABLE IF EXISTS Customer CASCADE;

-- Drop Sequences if they exist

DROP SEQUENCE IF EXISTS Employee\_Seq CASCADE;

DROP SEQUENCE IF EXISTS Order\_Seq CASCADE;

DROP SEQUENCE IF EXISTS Customer\_Seq CASCADE;

-- Drop Schema if it exists---

DROP SCHEMA IF EXISTS Restaurant CASCADE;

CREATE SCHEMA Restaurant;

SET SEARCH\_PATH TO Restaurant;

-------CREATE SEQUENCE---

CREATE SEQUENCE Employee\_Seq

START WITH 101

INCREMENT BY 1;

CREATE SEQUENCE Order\_Seq

START WITH 1001

INCREMENT BY 1;

CREATE SEQUENCE Customer\_Seq

START WITH 7001

INCREMENT BY 1;

-------CREATE TRIGGERS-----

CREATE OR REPLACE FUNCTION update\_employee\_salary()

RETURNS TRIGGER AS $$

DECLARE

commission DECIMAL(10, 2);

BEGIN

-- Calculate commission based on order total

commission := NEW.order\_total \* 0.05; -- Assuming a 5% commission

-- Update employee's salary

UPDATE employee

SET salary = salary + commission

WHERE employee\_id = NEW.employee\_id;

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER after\_order\_insert

AFTER INSERT ON orders

FOR EACH ROW

EXECUTE FUNCTION update\_employee\_salary();

------CREATE TABLE-----

CREATE TABLE Restaurant (

restaurant\_id SERIAL PRIMARY KEY,

name VARCHAR(255),

location VARCHAR(255),

cuisine VARCHAR(255),

Hours VARCHAR(255)

);

CREATE TABLE Employee (

Employee\_ID INT PRIMARY KEY DEFAULT NEXTVAL('Employee\_Seq'),

Name VARCHAR(100),

Role VARCHAR(50),

Salary DECIMAL(10, 2),

restaurant\_id INT,

FOREIGN KEY (restaurant\_id) REFERENCES Restaurant(restaurant\_id)

);

CREATE TABLE Orders (

Order\_ID INT PRIMARY KEY DEFAULT NEXTVAL('Order\_Seq'),

Employee\_ID INT,

Order\_Total DECIMAL(10, 2),

Order\_Table\_Number INT,

Order\_Payment\_Method VARCHAR(50),

Order\_Status VARCHAR(50),

FOREIGN KEY (Employee\_ID) REFERENCES Employee(Employee\_ID)

);

CREATE TABLE Customer (

Customer\_ID INT PRIMARY KEY DEFAULT NEXTVAL('Customer\_Seq'),

Order\_ID INT,

Customer\_Contact BIGINT,

Customer\_email VARCHAR(50),

Customer\_name VARCHAR(50),

FOREIGN KEY (Order\_ID) REFERENCES Orders(Order\_ID)

);

# DATA MANIPULATION LANGUAGE (DML)

Select \* from Restaurant

INSERT INTO Restaurant (name, location, cuisine, Hours) VALUES ('Olive Garden California','1350 North First Street,San Jose', 'Italian','11:00 AM to 2:00 AM');

INSERT INTO Restaurant (name, location, cuisine, Hours) VALUES ('Olive Garden Texas','12870 Gulf Freeway,Houston','Mexican','11:00 AM to 2:00 AM');

INSERT INTO Restaurant (name, location, cuisine, Hours) VALUES ('Olive Garden Florida','1555 Sand Lake Road,Orlando','Thai','11:00 AM to 2:00 AM');

INSERT INTO Restaurant (name, location, cuisine, Hours) VALUES ('Olive Garden New york','2001 South Road,Poughkeepsie', 'Italian','11:00 AM to 2:00 AM');

INSERT INTO Restaurant (name, location, cuisine, Hours) VALUES ('Olive Garden Illinois','30 East Anthony Drive,Champaign','Mexican','11:00 AM to 2:00 AM');

INSERT INTO Restaurant (name, location, cuisine, Hours) VALUES ('Olive Garden Georgia','905 Holcomb Bridge Road,Roswell','Thai','11:00 AM to 2:00 AM');

INSERT INTO Restaurant (name, location, cuisine, Hours) VALUES ('Olive Garden Virginia','7505 Broken Branch Ln,Manassas', 'Italian','11:00 AM to 2:00 AM');

INSERT INTO Restaurant (name, location, cuisine, Hours) VALUES ('Olive Garden Washington','11325 NE 124th Street,Kirkland','Mexican','11:00 AM to 2:00 AM');

INSERT INTO Restaurant (name, location, cuisine, Hours) VALUES ('Olive Garden Arizona','1261 West Irvington Road,Tucson','Thai','11:00 AM to 2:00 AM');

INSERT INTO Restaurant (name, location, cuisine, Hours) VALUES ('Olive Garden Colorado','3255 Cinema Point,Colorado Springs', 'Italian','11:00 AM to 2:00 AM');

Select \* from Employee

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('John Doe', 'Manager', 50000.00, 1);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Jane Smith', 'Chef', 40000.00, 1);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Alice Johnson', 'Waitress', 30000.00, 1);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Mark John', 'Waiter', 30000.00, 1);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Bob Brown', 'Cleaner', 25000.00, 1);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('John Neol', 'Chef', 40000.00, 1);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Mary Deb', 'Waitress', 30000.00, 1);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Nik Johnson', 'Waiter', 30000.00, 1);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Maria Hussen', 'Cleaner', 25000.00, 1);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Maria Hussen', 'Cleaner', 25000.00, 1);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Dana White', 'Manager', 55000.00, 2);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Charlie Davis', 'Chef', 40000.00, 2);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Fiona Green', 'Waitress', 28000.00, 2);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Edward King', 'Waiter', 30000.00, 2);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Hannah Scott', 'Cleaner', 25000.00, 2);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('George Long', 'Chef', 45000.00, 2);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Mia Wong', 'Waitress', 30000.00, 2);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Eli Nash', 'Waiter', 30000.00, 2);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Zeo Ford', 'Cleaner', 25000.00, 2);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Eva Roy', 'Cleaner', 25000.00, 2);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Leo Yan', 'Manager', 55000.00, 3);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Ian Kay', 'Chef', 40000.00, 3);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Lia Cox', 'Waitress', 28000.00, 3);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Noah Lee', 'Waiter', 30000.00, 3);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Hamy Tao', 'Cleaner', 25000.00, 3);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Max Joy', 'Chef', 45000.00, 3);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Tara Reid', 'Waitress', 30000.00, 3);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Sam Fox', 'Waiter', 30000.00, 3);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Ian Lowe', 'Cleaner', 25000.00, 3);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Mia Hill', 'Cleaner', 25000.00, 3);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Leo Hunt', 'Manager', 55000.00, 4);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Zoe Bird', 'Chef', 40000.00, 4);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Max Cook', 'Waitress', 28000.00, 4);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Lily Shaw', 'Waiter', 30000.00, 4);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Jake Pool', 'Cleaner', 25000.00, 4);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Tia Lee', 'Chef', 45000.00, 4);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Jay Kim', 'Waitress', 30000.00, 4);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Zoe Cox', 'Waiter', 30000.00, 4);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Kai Tao', 'Cleaner', 25000.00, 4);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Eva Ray', 'Cleaner', 25000.00, 4);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Ian Joy', 'Manager', 55000.00, 5);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Ada Fox', 'Chef', 40000.00, 5);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Leo Wu', 'Waitress', 28000.00, 5);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Lia Zhu', 'Waiter', 30000.00, 5);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Ben Roy', 'Cleaner', 25000.00, 5);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Ava Lee', 'Chef', 45000.00, 5);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Tom Day', 'Waitress', 30000.00, 5);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Zoe May', 'Waiter', 30000.00, 5);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Eva Joy', 'Cleaner', 25000.00, 5);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Kai Low', 'Cleaner', 25000.00, 5);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Ava Lee', 'Manager', 55000.00, 6);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Tom Day', 'Chef', 40000.00, 6);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Zoe May', 'Waitress', 28000.00, 6);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Ian Poe', 'Waiter', 30000.00, 6);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Eva Joy', 'Cleaner', 25000.00, 6);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Max Roe', 'Chef', 45000.00, 6);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Nia Fox', 'Waitress', 30000.00, 6);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Leo Bay', 'Waiter', 30000.00, 6);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Lia Fay', 'Cleaner', 25000.00, 6);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Kai High', 'Cleaner', 25000.00, 6);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Shwe Krish', 'Manager', 55000.00, 7);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Sue Kim', 'Chef', 40000.00, 7);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Ben Zhu', 'Waitress', 28000.00, 7);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Mia Tao', 'Waiter', 30000.00, 7);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Ray Cox', 'Cleaner', 25000.00, 7);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Eli Dee', 'Chef', 45000.00, 7);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Amy Ray', 'Waitress', 30000.00, 7);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Tim Kay', 'Waiter', 30000.00, 7);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Ned Joy', 'Cleaner', 25000.00, 7);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Liz Rue', 'Cleaner', 25000.00, 7);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Nia Moss', 'Manager', 55000.00, 8);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Rob Finn', 'Chef', 40000.00, 8);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Zoe Hart', 'Waitress', 28000.00, 8);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Leo Gage', 'Waiter', 30000.00, 8);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Eva Wood', 'Cleaner', 25000.00, 8);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Sam Cook', 'Chef', 45000.00, 8);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Lia West', 'Waitress', 30000.00, 8);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Jay Wolf', 'Waiter', 30000.00, 8);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Mia Shaw', 'Cleaner', 25000.00, 8);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Jon Snow', 'Cleaner', 25000.00, 8);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Eve Park', 'Manager', 55000.00, 9);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Tim Reed', 'Chef', 40000.00, 9);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Ana Hill', 'Waitress', 28000.00, 9);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Max Lake', 'Waiter', 30000.00, 9);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Kim Dale', 'Cleaner', 25000.00, 9);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Pat Dean', 'Chef', 45000.00, 9);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Rex Tate', 'Waitress', 30000.00, 9);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Zac Best', 'Waiter', 30000.00, 9);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Dot King', 'Cleaner', 25000.00, 9);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Lou Bird', 'Cleaner', 25000.00, 9);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Kai Cole', 'Manager', 55000.00, 10);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Ned Bell', 'Chef', 40000.00, 10);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Tia Rose', 'Waitress', 28000.00, 10);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Ian Dunn', 'Waiter', 30000.00, 10);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Fay Hope', 'Cleaner', 25000.00, 10);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Sid Wade', 'Chef', 45000.00, 10);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Meg Lane', 'Waitress', 30000.00, 10);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Roy Earl', 'Waiter', 30000.00, 10);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Gus Ford', 'Cleaner', 25000.00, 10);

INSERT INTO Employee (Name, Role, Salary, restaurant\_id) VALUES ('Amy Peak', 'Cleaner', 25000.00, 10);

Select \* from Employee

SELECT \* from Orders

-- DML for Orders table

DO $$

DECLARE

emp\_id INTEGER;

BEGIN

FOR emp\_id IN 101..200 LOOP

-- Insert 10 orders for each employee

FOR i IN 1..10 LOOP

INSERT INTO Orders (Employee\_ID, Order\_Total, Order\_Table\_Number, Order\_Payment\_Method, Order\_Status)

VALUES (

emp\_id,

ROUND((RANDOM() \* 275 + 25)::numeric, 2), -- Random order total between 25 and 300

CAST((RANDOM() \* 20 + 1) AS INTEGER), -- Random table number between 1 and 20

CASE CAST((RANDOM() \* 1) AS INTEGER) -- Random payment method

WHEN 0 THEN 'Cash'

ELSE 'Credit Card'

END,

CASE CAST((RANDOM() \* 1) AS INTEGER) -- Random order status

WHEN 0 THEN 'Completed'

ELSE 'In Progress'

END

);

END LOOP;

END LOOP;

END $$;

CREATE EXTENSION IF NOT EXISTS pgcrypto;

INSERT INTO Customer (Order\_ID, Customer\_Contact, Customer\_email, Customer\_name)

SELECT

(1000 + random() \* 1000)::BIGINT, -- Random Order\_ID between 1001 and 2000

floor(random() \* 10000000000)::BIGINT, -- Random Customer\_Contact (10 digits)

substr(md5(random()::text), 0, 20) || '@example.com', -- Random Customer\_email

substr(md5(random()::text), 0, 10) -- Random Customer\_name

FROM generate\_series(1, 1000); -- Insert 1000 random customer entries

SELECT \* FROM Customer

------DROP Table customer---

---Query----

-- Select all columns and all rows from one table

SELECT \* FROM Restaurant;

-- Select five columns and all rows from one table

SELECT Employee\_ID, Name, Role, Salary, Restaurant\_ID FROM Employee;

-- Select all columns from all rows from one view

CREATE VIEW Review AS

SELECT Employee.Name, Employee.Role, Restaurant.Name AS Restaurant\_Name, Orders.Order\_ID, Orders.Order\_Total

FROM Employee

JOIN Restaurant ON Employee.Restaurant\_ID = Restaurant.Restaurant\_ID

JOIN Orders ON Employee.Employee\_ID = Orders.Employee\_ID

-- Query from the view

SELECT \* FROM Review;

SELECT Employee.Name, Employee.Role, Restaurant.Name AS Restaurant\_Name, Order.Order\_ID, Order.Total\_Amount

FROM Employee

JOIN Restaurant ON Employee.Restaurant\_ID = Restaurant.Restaurant\_ID

JOIN Order ON Employee.Employee\_ID = Order.Employee\_ID

-- Using a join on 2 tables, select all columns and all rows from the tables without the use of a Cartesian product

SELECT \* FROM Employee

JOIN Restaurant ON Employee.Restaurant\_ID = Restaurant.Restaurant\_ID;

-- Select and order data retrieved from one table

SELECT \* FROM Employee

ORDER BY Salary DESC;

-- Using a join on 3 tables, select 5 columns from the 3 tables. Use syntax that would limit the output to 3 rows

SELECT Employee.Name, Employee.Role, Restaurant.Name AS Restaurant\_Name, Orders.Order\_ID, Orders.order\_Total

FROM Employee

JOIN Restaurant ON Employee.Restaurant\_ID = Restaurant.Restaurant\_ID

JOIN Orders ON Employee.Employee\_ID = Orders.Employee\_ID

LIMIT 3;

-- Select distinct rows using joins on 3 tables

SELECT DISTINCT Employee.Name, Restaurant.Name AS Restaurant\_Name, Orders.Order\_ID

FROM Employee

JOIN Restaurant ON Employee.Restaurant\_ID = Restaurant.Restaurant\_ID

JOIN Orders ON Employee.Employee\_ID = Orders.Employee\_ID;

-- Use GROUP BY and HAVING in a select statement using one or more tables

SELECT Employee\_ID, COUNT(\*) AS Num\_Orders, SUM(Order\_total) AS Total\_Sales

FROM Orders

GROUP BY Employee\_ID

HAVING COUNT(\*) > 5;

-- Use IN clause to select data from one or more tables

SELECT \* FROM Employee

WHERE salary IN (28000);

-- Select length of one column from one table (use LENGTH function)

SELECT Name, LENGTH(Name) AS Name\_Length FROM Employee;

-- Delete one record from one table. Use select statements to demonstrate the table contents before and after the DELETE statement. Make sure you use ROLLBACK afterwards so that the data will not be physically removed

-- Display before deletion

SELECT \* FROM Customer;

-- Deleting an employee record

DELETE FROM Customer WHERE Customer\_ID = 7001;

-- Display after deletion

SELECT \* FROM Customer;

-- Rollback the deletion to preserve data

ROLLBACK;

-- Update One Record from One Table

-- Display before update

SELECT \* FROM Employee WHERE Employee\_ID = 122;

-- Update the record

UPDATE Employee SET Salary = 55000.00 WHERE Employee\_ID = 122;

-- Display after update

SELECT \* FROM Employee WHERE Employee\_ID = 122;

-- Rollback the update to preserve original data

ROLLBACK;

-- Advanced Query 1: Aggregated Sales by Restaurant

SELECT Restaurant.Name, COUNT(Orders.Order\_ID) AS Total\_Orders, SUM(Order\_total) AS Total\_Sales

FROM Orders

JOIN Employee ON Orders.Employee\_ID = Employee.Employee\_ID

JOIN Restaurant ON Employee.Restaurant\_ID = Restaurant.Restaurant\_ID

GROUP BY Restaurant.Name

ORDER BY Total\_Sales DESC;