**ITEC 4200 Advanced Database Semester Project**

**Student Name:** Fahao Wen & Biruk Binchamo

**Semester:** Fall 2022

**Name of Project:** Restaurant Database

**Email:** [fwen@ggc.edu](mailto:fwen@ggc.edu) bbinchamo@ggc.edu

**Phone number:** (404)-940-6861

Table of Contents

[**I.** **Executive Summary and ERD Design** 4](#_Toc119225798)

[**Objective** 4](#_Toc119225799)

[**Goals** 4](#_Toc119225800)

[**Solution** 4](#_Toc119225801)

[**Benefits to Users** 4](#_Toc119225802)

[**Project Outline** 4](#_Toc119225803)

[**Schema Design** 5](#_Toc119225804)

[**ERD Design** 6](#_Toc119225805)

[**II.** **DDL: Schema Implementation** 7](#_Toc119225806)

[**Code:** 7](#_Toc119225807)

[**III.** **Schema Implementation** 11](#_Toc119225808)

[**Code:** 11](#_Toc119225809)

[**Figures** 27](#_Toc119225810)

[**IV.** **Queries** 32](#_Toc119225811)

[**Query 1** 32](#_Toc119225812)

[Purpose 32](#_Toc119225813)

[Code 32](#_Toc119225814)

[Figure 32](#_Toc119225815)

[**Query 2** 33](#_Toc119225816)

[Purpose 33](#_Toc119225817)

[Code 33](#_Toc119225818)

[Figure 33](#_Toc119225819)

[**Query 3** 34](#_Toc119225820)

[Purpose 34](#_Toc119225821)

[Code 34](#_Toc119225822)

[Figure 34](#_Toc119225823)

[**Query 4** 35](#_Toc119225824)

[Purpose 35](#_Toc119225825)

[Code 35](#_Toc119225826)

[Figure 35](#_Toc119225827)

[**Query 5** 36](#_Toc119225828)

[Purpose 36](#_Toc119225829)

[Code 36](#_Toc119225830)

[Figure 36](#_Toc119225831)

[**Query 6** 37](#_Toc119225832)

[Purpose 37](#_Toc119225833)

[Code 37](#_Toc119225834)

[Figure 37](#_Toc119225835)

[**Query 7** 38](#_Toc119225836)

[Purpose 38](#_Toc119225837)

[Code 38](#_Toc119225838)

[Figure 38](#_Toc119225839)

[**Query 8** 39](#_Toc119225840)

[Purpose 39](#_Toc119225841)

[Code 39](#_Toc119225842)

[Figure 39](#_Toc119225843)

[**Query 9** 40](#_Toc119225844)

[Purpose 40](#_Toc119225845)

[Code 40](#_Toc119225846)

[Figure 40](#_Toc119225847)

[**Query 10** 41](#_Toc119225848)

[Purpose 41](#_Toc119225849)

[Code 41](#_Toc119225850)

[Figure 41](#_Toc119225851)

[**Query 11** 42](#_Toc119225852)

[Purpose 42](#_Toc119225853)

[Code 42](#_Toc119225854)

[Figure 42](#_Toc119225855)

[**Query 12** 43](#_Toc119225856)

[Purpose 43](#_Toc119225857)

[Code 43](#_Toc119225858)

[Figure 43](#_Toc119225859)

[**Query 13** 44](#_Toc119225860)

[Purpose 44](#_Toc119225861)

[Code 44](#_Toc119225862)

[Figure 44](#_Toc119225863)

[**Query 14** 45](#_Toc119225864)

[Purpose 45](#_Toc119225865)

[Code 45](#_Toc119225866)

[Figure 46](#_Toc119225867)

[**Query 15** 47](#_Toc119225868)

[Purpose 47](#_Toc119225869)

[Code 47](#_Toc119225870)

[Figure 47](#_Toc119225871)

[**V.** **Schema Objects** 48](#_Toc119225872)

[**View 1** 48](#_Toc119225873)

[Figure 48](#_Toc119225874)

[**View 2** 49](#_Toc119225875)

[Figure 49](#_Toc119225876)

[**Index** 50](#_Toc119225877)

[**Flashback** 50](#_Toc119225878)

# **Executive Summary and ERD Design**

## **Objective**

The purpose of this database is to help restaurants keep track of their business records including information of restaurants, customers, tables, and orders.

## **Goals**

The goal of this database project is to create a user-friendly database system that stores and organizes information and data for restaurants. Additionally, we will use queries to build business report about the important entities such as customer information, order information, table information, and so on to help improve business.

## **Solution**

The solution of this project is to create table and build relationship between them. The table includes restaurant, order, drink menu, food menu, customer, table, and employee.

There are the six min entities of the database, with these table, we could use queries to build reports and analysis the essential information of the business. The project will be implemented using Oracle 18c Express.

## **Benefits to Users**

The restaurant database will help users reduce the amount of time on managing data; it can help users analyze data in variety of ways; it will improve the quality and consistency of information.

## **Project Outline**

The project will contain the following major components:

* Schema Design
* Entity-Relationship Diagram
* Table implementation
* Queries
* Reports

## **Schema Design**

Restaurant (Restaurant\_ID, Restaurant\_Name, Street, City, Zip, Phone, Cusine\_Type)

Drink\_Menu (Drink\_Menus\_ID, Drink\_Name, Drink\_Price, Restaurant\_ID)

FK Restaurant\_ID 🡪 Restaurant

Food\_Menu (Food\_Menu\_ID, Food\_Name, Food\_Price, Calories, Restaruant\_ID)

FK Restaurant\_ID 🡪 Restaurant

Employee (Employee\_ID, First\_Name, Last\_Name, Title, SSN, Phone, Street, City, State, Zip\_Code, BOA, Restaurant\_ID)

FK Restaurant\_ID 🡪 Restaurant

Table (Table\_ID, Table\_Size, Employee\_ID, Order\_ID, Customer\_ID, Reservation\_Time, Party\_Size)

FK Order\_ID 🡪 Order

FK Customer\_ID 🡪 Customer

FK Employee\_ID 🡪 Employee

Customer (Customer\_ID, First\_Name, Last\_Name, Phone)

Order (Order\_ID, Price, Time)

## **ERD Design**

Diagram

Description automatically generated

**Entity-Relationship Diagram**

# **DDL: Schema Implementation**

## **Code:**

-- drop table for multiple runs

Drop TABLE Restaurant CASCADE CONSTRAINTS;

DROP TABLE Drink\_Menu CASCADE CONSTRAINTS;

DROP TABLE Food\_Menu CASCADE CONSTRAINTS;

DROP TABLE Employee CASCADE CONSTRAINTS;

DROP TABLE Customer CASCADE CONSTRAINTS;

DROP TABLE Orders CASCADE CONSTRAINTS;

DROP TABLE Tabl CASCADE CONSTRAINTS;

CREATE TABLE Restaurant(

Restaurant\_ID Int NOT NULL ,

Restaurant\_Name Varchar(45),

Street Varchar(45),

City Varchar(10),

State Char(2),

Zip Int,

Phone Int,

Cuisine\_Type Varchar(45),

CONSTRAINT Restaurant\_Restaurant\_ID\_Pk PRIMARY KEY (Restaurant\_ID));

-- create Drink\_menu table

CREATE TABLE Drink\_Menu(

Drink\_Menu\_ID Int NOT NULL,

Drink\_Name Varchar(25),

Drink\_Price int,

Restaurant\_ID Int,

CONSTRAINT Drink\_Menu\_ID\_Pk PRIMARY KEY (Drink\_Menu\_ID),

CONSTRAINT Drink\_Menu\_Restaurant\_ID\_Fk FOREIGN KEY (Restaurant\_ID)

REFERENCES Restaurant (Restaurant\_ID));

-- create Food\_Menu table

CREATE TABLE Food\_Menu(

Food\_Menu\_ID Int NOT NULL,

Food\_Name Varchar(45),

Food\_Price int,

Calories int,

Restaurant\_ID Int,

CONSTRAINT Food\_Menu\_ID\_Pk PRIMARY KEY (Food\_Menu\_ID),

CONSTRAINT Food\_Menu\_Restaurant\_ID\_Fk FOREIGN KEY (Restaurant\_ID)

REFERENCES Restaurant (Restaurant\_ID));

-- create Employee table

CREATE TABLE Employee(

Employee\_ID int NOT NULL,

First\_Name Varchar(10),

Last\_Name Varchar(10),

Title Varchar(20),

SSN Int,

Phone Int,

Street Varchar(25),

City Varchar(25),

State Char(2),

Zip Int,

BOA Date,

Restaurant\_ID int,

CONSTRAINT Employee\_Employee\_ID\_Pk PRIMARY KEY(Employee\_ID),

CONSTRAINT Employee\_Restaurant\_ID\_Fk FOREIGN KEY(Restaurant\_ID)

REFERENCES Restaurant (Restaurant\_ID));

-- create Customer table

CREATE TABLE Customer (

Customer\_ID Int NOT NULL,

First\_Name Varchar(10),

Last\_Name Varchar(10),

Phone int,

CONSTRAINT Customer\_Customer\_ID\_Pk PRIMARY KEY(Customer\_ID));

-- create Orders table

CREATE TABLE Orders(

Orders\_ID Int NOT NULL,

Price int,

Times Date,

CONSTRAINT Orders\_Orders\_ID\_Pk PRIMARY KEY (Orders\_ID));

-- create Tables table

CREATE TABLE Tabl(

Tabl\_ID int NOT NULL,

Tabl\_Size int,

Reservation\_Time Date,

Customer\_ID Int,

Orders\_ID int,

Employee\_ID int,

CONSTRAINT Tabl\_Tables\_ID\_Pk PRIMARY KEY(Tabl\_ID),

CONSTRAINT Tabl\_Customer\_ID\_Fk FOREIGN KEY(Customer\_ID)

REFERENCES Customer(Customer\_ID),

CONSTRAINT Tables\_Orders\_ID\_Fk FOREIGN KEY(Orders\_ID)

REFERENCES Orders(Orders\_ID),

CONSTRAINT Tables\_Employee\_ID\_Fk FOREIGN KEY(Employee\_ID)

REFERENCES Employee (Employee\_ID));

COMMIT;

# **Schema Implementation**

## **Code:**

------------------------------------------------------------------------------

-- Create Sequence

CREATE SEQUENCE rest\_seq

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;

------------------------------------------------------------------------------

-- Insert Value for Restaurant Table

INSERT INTO Restaurant(Restaurant\_ID,Restaurant\_Name,Street,City,State,Zip,Phone,Cuisine\_Type)

VALUES(rest\_seq.nextval,'CollegeBar','1823 Hope Road','Lawrence','GA',30088,6787899087,'Fast FooD');

INSERT INTO Restaurant(Restaurant\_ID,Restaurant\_Name,Street,City,State,Zip,Phone,Cuisine\_Type)

VALUES(rest\_seq.nextval,'Lomi Restaurant Bar','2555 Delk Rd SE Suite A7','Marietta','GA',30067,7703808280,'Fast Food');

INSERT INTO Restaurant(Restaurant\_ID,Restaurant\_Name,Street,City,State,Zip,Phone,Cuisine\_Type)

VALUES(rest\_seq.nextval,'MikChans','479 Flat Shoals Ave SE','Atlanta','GA',30316,4044360645,'Mexico');

INSERT INTO Restaurant(Restaurant\_ID,Restaurant\_Name,Street,City,State,Zip,Phone,Cuisine\_Type)

VALUES(rest\_seq.nextval,'Atlanta Teriyaki','477 Flat Shoals Ave SE','Atlanta','GA',30316,4045582052,'Chinese');

INSERT INTO Restaurant(Restaurant\_ID,Restaurant\_Name,Street,City,State,Zip,Phone,Cuisine\_Type)

VALUES(rest\_seq.nextval,'Gaja Korean Bar','491 Flat Shoals Ave SE ','Atlanta','GA',30316,4048352126,'Korean');

INSERT INTO Restaurant(Restaurant\_ID,Restaurant\_Name,Street,City,State,Zip,Phone,Cuisine\_Type)

VALUES(rest\_seq.nextval,'Bare Bones Steakhouse','101 E Main St NE','Buford','GA',30518,4048352126,'Steak House');

INSERT INTO Restaurant(Restaurant\_ID,Restaurant\_Name,Street,City,State,Zip,Phone,Cuisine\_Type)

VALUES(rest\_seq.nextval,'Aqua Terra Bistro','55 E Main St NE','Lawrence','GA',30088,7702713000,'Italian ');

INSERT INTO Restaurant(Restaurant\_ID,Restaurant\_Name,Street,City,State,Zip,Phone,Cuisine\_Type)

VALUES(rest\_seq.nextval,'Teds Montana Grill','1680 Mall of Georgia Blvd Suite','Buford','GA',30519,6785463631,'BBQ');

INSERT INTO Restaurant(Restaurant\_ID,Restaurant\_Name,Street,City,State,Zip,Phone,Cuisine\_Type)

VALUES(rest\_seq.nextval,'5Church Buckhead','3379 Peachtree Rd NE STE','Atlanta','GA',30326,4708194841,'Inidian');

-- Tested Restaurant table

SELECT \* from Restaurant;

------------------------------------------------------------------------------

-- Insert Value to Drink\_Menu Table

INSERT INTO Drink\_Menu(Drink\_Menu\_ID,Drink\_Name,Drink\_Price,Restaurant\_ID)

VALUES(1,'Mojito',6,7);

INSERT INTO Drink\_Menu(Drink\_Menu\_ID,Drink\_Name,Drink\_Price,Restaurant\_ID)

VALUES(2,'Teqyila Sunrise',5,5);

INSERT INTO Drink\_Menu(Drink\_Menu\_ID,Drink\_Name,Drink\_Price,Restaurant\_ID)

VALUES(3,'Pepsi',1.87,1);

INSERT INTO Drink\_Menu(Drink\_Menu\_ID,Drink\_Name,Drink\_Price,Restaurant\_ID)

VALUES(4,'Coke',1.99,2);

INSERT INTO Drink\_Menu(Drink\_Menu\_ID,Drink\_Name,Drink\_Price,Restaurant\_ID)

VALUES(5,'Red Wine',9,6);

INSERT INTO Drink\_Menu(Drink\_Menu\_ID,Drink\_Name,Drink\_Price,Restaurant\_ID)

VALUES(6,'Bloody Mary',8.9,7);

INSERT INTO Drink\_Menu(Drink\_Menu\_ID,Drink\_Name,Drink\_Price,Restaurant\_ID)

VALUES(7,'Old Fashion',9,3);

INSERT INTO Drink\_Menu(Drink\_Menu\_ID,Drink\_Name,Drink\_Price,Restaurant\_ID)

VALUES(8,'Peach Margarita ',7.56,3);

INSERT INTO Drink\_Menu(Drink\_Menu\_ID,Drink\_Name,Drink\_Price,Restaurant\_ID)

VALUES(9,'Espresso Martini',9.99,9);

INSERT INTO Drink\_Menu(Drink\_Menu\_ID,Drink\_Name,Drink\_Price,Restaurant\_ID)

VALUES(10,'White Wine',9,6);

INSERT INTO Drink\_Menu(Drink\_Menu\_ID,Drink\_Name,Drink\_Price,Restaurant\_ID)

VALUES(11,'Ice Tea',2.5,4);

INSERT INTO Drink\_Menu(Drink\_Menu\_ID,Drink\_Name,Drink\_Price,Restaurant\_ID)

VALUES(12,'Long Island',6.6,3);

INSERT INTO Drink\_Menu(Drink\_Menu\_ID,Drink\_Name,Drink\_Price,Restaurant\_ID)

VALUES(13,'Coffee',2.5,8);

INSERT INTO Drink\_Menu(Drink\_Menu\_ID,Drink\_Name,Drink\_Price,Restaurant\_ID)

VALUES(14,'Lemonade',2.5,8);

INSERT INTO Drink\_Menu(Drink\_Menu\_ID,Drink\_Name,Drink\_Price,Restaurant\_ID)

VALUES(15,'Beer',5.15,8);

-- Tested Food Menus;

SELECT \* from Drink\_Menu;

------------------------------------------------------------------------------

-- Insert Value into Food\_Menu Table

INSERT INTO Food\_Menu(Food\_Menu\_ID,Food\_Name,Food\_Price,Calories,Restaurant\_ID)

VALUES(1,'Wiskey Steak',30.25,320,6);

INSERT INTO Food\_Menu(Food\_Menu\_ID,Food\_Name,Food\_Price,Calories,Restaurant\_ID)

VALUES(2,'Pork Chop Biskey',12.75,450,6);

INSERT INTO Food\_Menu(Food\_Menu\_ID,Food\_Name,Food\_Price,Calories,Restaurant\_ID)

VALUES(3,'Fried Rice',9.58,250,4);

INSERT INTO Food\_Menu(Food\_Menu\_ID,Food\_Name,Food\_Price,Calories,Restaurant\_ID)

VALUES(4,'Pork Belly Bao',6.55,150,5);

INSERT INTO Food\_Menu(Food\_Menu\_ID,Food\_Name,Food\_Price,Calories,Restaurant\_ID)

VALUES(5,'Orange Chicken',7.45,255,4);

INSERT INTO Food\_Menu(Food\_Menu\_ID,Food\_Name,Food\_Price,Calories,Restaurant\_ID)

VALUES(6,'Burger',5.56,400,1);

INSERT INTO Food\_Menu(Food\_Menu\_ID,Food\_Name,Food\_Price,Calories,Restaurant\_ID)

VALUES(7,'French Fried',2.25,350,2);

INSERT INTO Food\_Menu(Food\_Menu\_ID,Food\_Name,Food\_Price,Calories,Restaurant\_ID)

VALUES(8,'Angle Hair Pasta ',11.58,240,7);

INSERT INTO Food\_Menu(Food\_Menu\_ID,Food\_Name,Food\_Price,Calories,Restaurant\_ID)

VALUES(9,'Meat Ball',12.58,120,7);

INSERT INTO Food\_Menu(Food\_Menu\_ID,Food\_Name,Food\_Price,Calories,Restaurant\_ID)

VALUES(10,'Chicken Tikka Masala',14.58,341,9);

INSERT INTO Food\_Menu(Food\_Menu\_ID,Food\_Name,Food\_Price,Calories,Restaurant\_ID)

VALUES(11,'BBQ Ribs',15.99,440,8);

INSERT INTO Food\_Menu(Food\_Menu\_ID,Food\_Name,Food\_Price,Calories,Restaurant\_ID)

VALUES(12,'Burrito',13.65,240,3);

INSERT INTO Food\_Menu(Food\_Menu\_ID,Food\_Name,Food\_Price,Calories,Restaurant\_ID)

VALUES(13,'Salad',9.11,100,2);

INSERT INTO Food\_Menu(Food\_Menu\_ID,Food\_Name,Food\_Price,Calories,Restaurant\_ID)

VALUES(14,'Pizza',13.34,350,7);

INSERT INTO Food\_Menu(Food\_Menu\_ID,Food\_Name,Food\_Price,Calories,Restaurant\_ID)

VALUES(15,'Curry',15.56,250,9);

--Tested Food\_Menu Table

SELECT \* FROM Food\_Menu;

------------------------------------------------------------------------------

--Insert Value into Employee table

INSERT INTO Employee(Employee\_ID,First\_Name,Last\_Name,BOA,SSN,Title,Phone,Street,City,State,Zip,Restaurant\_ID)

VALUES(1,'Keli','Smith',DATE'1999-11-23',188908129,'Cashier',4509099080,'1990 court road','Snellville','GA',30098,5);

INSERT INTO Employee(Employee\_ID,First\_Name,Last\_Name,BOA,SSN,Title,Phone,Street,City,State,Zip,Restaurant\_ID)

VALUES(2,'Savior','Johnson',DATE'2000-11-21',234890989,'Manager',6788909876,'234 lawrenville street','Stone Mountain','GA',30088,4);

INSERT INTO Employee(Employee\_ID,First\_Name,Last\_Name,BOA,SSN,Title,Phone,Street,City,State,Zip,Restaurant\_ID)

VALUES(3,'Shamo','Williams',DATE'1998-05-09',134092154,'Waiter',5670919078,'1234 forest Park','Forest Park','GA',30098,3);

INSERT INTO Employee(Employee\_ID,First\_Name,Last\_Name,BOA,SSN,Title,Phone,Street,City,State,Zip,Restaurant\_ID)

VALUES(4,'Fool','Brown',DATE'1993-04-28',430129087,'Chef',5677162342,'321 sreet road','Rome','GA',30079,8);

INSERT INTO Employee(Employee\_ID,First\_Name,Last\_Name,BOA,SSN,Title,Phone,Street,City,State,Zip,Restaurant\_ID)

VALUES(5,'Camerron','Jones',DATE'1992-12-09',231890878,'Supervisor',6780790989,'1827 north road','Snellville','GA',30098,9);

INSERT INTO Employee(Employee\_ID,First\_Name,Last\_Name,BOA,SSN,Title,Phone,Street,City,State,Zip,Restaurant\_ID)

VALUES(6,'James','Garcia',DATE'1996-07-31',544785454,'Waiter',2125559113,'1205 Metropolitan','Atlanta','GA',30316,1);

INSERT INTO Employee(Employee\_ID,First\_Name,Last\_Name,BOA,SSN,Title,Phone,Street,City,State,Zip,Restaurant\_ID)

VALUES(7,'Robert','Davis',DATE'1989-06-04',714544896,'Cashier',6465551052,'1105 Metropolitan','Atlanta','GA',30316,5);

INSERT INTO Employee(Employee\_ID,First\_Name,Last\_Name,BOA,SSN,Title,Phone,Street,City,State,Zip,Restaurant\_ID)

VALUES(8,'Michael','Millers',DATE'1974-05-30',1154567486,'Waiter',6465551150,'253 Bedford St','Columbus','GA',31902,2);

INSERT INTO Employee(Employee\_ID,First\_Name,Last\_Name,BOA,SSN,Title,Phone,Street,City,State,Zip,Restaurant\_ID)

VALUES(9,'William','Rodriguez',DATE'1979-06-06',4894864789,'Manager',2125559488,'456 4th Street','Atlanta','GA',30328,2);

INSERT INTO Employee(Employee\_ID,First\_Name,Last\_Name,BOA,SSN,Title,Phone,Street,City,State,Zip,Restaurant\_ID)

VALUES(10,'Thomas','Wilson',DATE'2000-08-02',1564864564,'Chef',6345556874,'493 West Pearl Dr.','Snellville','GA',31082,3);

INSERT INTO Employee(Employee\_ID,First\_Name,Last\_Name,BOA,SSN,Title,Phone,Street,City,State,Zip,Restaurant\_ID)

VALUES(11,'Christopher','Taylor',DATE'2005-08-09',9974564564,'Chef',9175554533,'25 Pacific Rd. ','Bowdon','GA',30108,5);

INSERT INTO Employee(Employee\_ID,First\_Name,Last\_Name,BOA,SSN,Title,Phone,Street,City,State,Zip,Restaurant\_ID)

VALUES(12,'Anthony','Lopez',DATE'1960-10-15',7894567864,'Waiter',2125558988,'7924 Dogwood St.','Kite','GA',31049,6);

INSERT INTO Employee(Employee\_ID,First\_Name,Last\_Name,BOA,SSN,Title,Phone,Street,City,State,Zip,Restaurant\_ID)

VALUES(13,'Mark','Lee',DATE'1999-12-12',7647481234,'Janitor',2125548988,'7987 West 4th Drive','Savannah','GA',31402,6);

INSERT INTO Employee(Employee\_ID,First\_Name,Last\_Name,BOA,SSN,Title,Phone,Street,City,State,Zip,Restaurant\_ID)

VALUES(14,'Steven','Yang',DATE'1985-05-05',7774564564,'Manager',6456844687,'5566 Read Road','Suwanne','GA',30336,1);

INSERT INTO Employee(Employee\_ID,First\_Name,Last\_Name,BOA,SSN,Title,Phone,Street,City,State,Zip,Restaurant\_ID)

VALUES(15,'William','Moore',DATE'1979-11-06',2311234641,'Cashier',1856445444,'1212 Penhurst Way','Peachtree Corner','GA',30157,4);

-- Test Employee Table

Select \* From Employee;

------------------------------------------------------------------------------

--Insert Value into Customer Table

INSERT INTO Customer(Customer\_ID,First\_Name,Last\_Name,Phone)

VALUES(1,'Metsi','Zerga',4705450989);

INSERT INTO Customer(Customer\_ID,First\_Name,Last\_Name,Phone)

VALUES(2,'Hand','Kina',2107899087);

INSERT INTO Customer(Customer\_ID,First\_Name,Last\_Name,Phone)

VALUES(3,'Gelata','Hanksi',3508900980);

INSERT INTO Customer(Customer\_ID,First\_Name,Last\_Name,Phone)

VALUES(4,'Sherly','Brown',4508798909);

INSERT INTO Customer(Customer\_ID,First\_Name,Last\_Name,Phone)

VALUES(5,'Fikir','Zema',5609009089);

INSERT INTO Customer(Customer\_ID,First\_Name,Last\_Name,Phone)

VALUES(6,'Nancy','Reid',4199809037);

INSERT INTO Customer(Customer\_ID,First\_Name,Last\_Name,Phone)

VALUES(7,'Thomas','Chill',9874270266);

INSERT INTO Customer(Customer\_ID,First\_Name,Last\_Name,Phone)

VALUES(8,'Marcy','Vitale',6086491890);

INSERT INTO Customer(Customer\_ID,First\_Name,Last\_Name,Phone)

VALUES(9,'Shula','Savinis',4383692804);

INSERT INTO Customer(Customer\_ID,First\_Name,Last\_Name,Phone)

VALUES(10,'Lisa','Paulino',5866223415);

INSERT INTO Customer(Customer\_ID,First\_Name,Last\_Name,Phone)

VALUES(11,'Toni','Sumulong',5057016510);

INSERT INTO Customer(Customer\_ID,First\_Name,Last\_Name,Phone)

VALUES(12,'Elizabeth','Smith',7013997819);

INSERT INTO Customer(Customer\_ID,First\_Name,Last\_Name,Phone)

VALUES(13,'Walter','Smith',4722552004);

INSERT INTO Customer(Customer\_ID,First\_Name,Last\_Name,Phone)

VALUES(14,'Vincent','John',6215375418);

INSERT INTO Customer(Customer\_ID,First\_Name,Last\_Name,Phone)

VALUES(15,'John','Greer',2449051786);

-- Test Customer Table

SELECT \* FROM CUSTOMER;

------------------------------------------------------------------------------

-- Insert Values into Orders table

INSERT INTO Orders(Orders\_ID,Price,Times)

VALUES(1,57.85,DATE'2022-09-01');

INSERT INTO Orders(Orders\_ID,Price,Times)

VALUES(2,66.455,DATE'2022-09-03');

INSERT INTO Orders(Orders\_ID,Price,Times)

VALUES(3,79.45,DATE'2022-09-03');

INSERT INTO Orders(Orders\_ID,Price,Times)

VALUES(4,100.65,DATE'2022-09-04');

INSERT INTO Orders(Orders\_ID,Price,Times)

VALUES(5,215.01,DATE'2022-09-05');

INSERT INTO Orders(Orders\_ID,Price,Times)

VALUES(6,30.12,DATE'2022-09-06');

INSERT INTO Orders(Orders\_ID,Price,Times)

VALUES(7,15.3,DATE'2022-09-07');

INSERT INTO Orders(Orders\_ID,Price,Times)

VALUES(8,22.11,DATE'2022-09-08');

INSERT INTO Orders(Orders\_ID,Price,Times)

VALUES(9,97,DATE'2022-09-09');

INSERT INTO Orders(Orders\_ID,Price,Times)

VALUES(10,86,DATE'2022-09-10');

INSERT INTO Orders(Orders\_ID,Price,Times)

VALUES(11,9.34,DATE'2022-09-11');

INSERT INTO Orders(Orders\_ID,Price,Times)

VALUES(12,77.45,DATE'2022-09-12');

INSERT INTO Orders(Orders\_ID,Price,Times)

VALUES(13,98.41,DATE'2022-09-13');

INSERT INTO Orders(Orders\_ID,Price,Times)

VALUES(14,15.85,DATE'2022-09-14');

INSERT INTO Orders(Orders\_ID,Price,Times)

VALUES(15,45.56,DATE'2022-09-15');

--Tested Orders Table

SELECT \* FROM ORDERS;

------------------------------------------------------------------------------

--Insert Values into Tabl Table

INSERT INTO Tabl(Tabl\_ID,Tabl\_Size,Reservation\_Time,Customer\_ID,Orders\_ID,Employee\_ID)

VALUES(1,6,DATE'2022-09-01',15,1,5);

INSERT INTO Tabl(Tabl\_ID,Tabl\_Size,Reservation\_Time,Customer\_ID,Orders\_ID,Employee\_ID)

VALUES(2,2,DATE'2022-09-03',14,2,9);

INSERT INTO Tabl(Tabl\_ID,Tabl\_Size,Reservation\_Time,Customer\_ID,Orders\_ID,Employee\_ID)

VALUES(3,2,DATE'2022-09-03',13,3,8);

INSERT INTO Tabl(Tabl\_ID,Tabl\_Size,Reservation\_Time,Customer\_ID,Orders\_ID,Employee\_ID)

VALUES(4,4,DATE'2022-09-04',12,4,7);

INSERT INTO Tabl(Tabl\_ID,Tabl\_Size,Reservation\_Time,Customer\_ID,Orders\_ID,Employee\_ID)

VALUES(5,6,DATE'2022-09-05',11,5,4);

INSERT INTO Tabl(Tabl\_ID,Tabl\_Size,Reservation\_Time,Customer\_ID,Orders\_ID,Employee\_ID)

VALUES(6,2,DATE'2022-09-06',10,6,6);

INSERT INTO Tabl(Tabl\_ID,Tabl\_Size,Reservation\_Time,Customer\_ID,Orders\_ID,Employee\_ID)

VALUES(7,4,DATE'2022-09-07',9,7,2);

INSERT INTO Tabl(Tabl\_ID,Tabl\_Size,Reservation\_Time,Customer\_ID,Orders\_ID,Employee\_ID)

VALUES(8,6,DATE'2022-09-08',8,8,1);

INSERT INTO Tabl(Tabl\_ID,Tabl\_Size,Reservation\_Time,Customer\_ID,Orders\_ID,Employee\_ID)

VALUES(9,4,DATE'2022-09-09',7,9,10);

INSERT INTO Tabl(Tabl\_ID,Tabl\_Size,Reservation\_Time,Customer\_ID,Orders\_ID,Employee\_ID)

VALUES(10,8,DATE'2022-09-10',6,10,12);

INSERT INTO Tabl(Tabl\_ID,Tabl\_Size,Reservation\_Time,Customer\_ID,Orders\_ID,Employee\_ID)

VALUES(11,2,DATE'2022-09-11',5,11,13);

INSERT INTO Tabl(Tabl\_ID,Tabl\_Size,Reservation\_Time,Customer\_ID,Orders\_ID,Employee\_ID)

VALUES(12,6,DATE'2022-09-12',4,12,11);

INSERT INTO Tabl(Tabl\_ID,Tabl\_Size,Reservation\_Time,Customer\_ID,Orders\_ID,Employee\_ID)

VALUES(13,4,DATE'2022-09-13',3,13,15);

INSERT INTO Tabl(Tabl\_ID,Tabl\_Size,Reservation\_Time,Customer\_ID,Orders\_ID,Employee\_ID)

VALUES(14,4,DATE'2022-09-14',2,14,14);

INSERT INTO Tabl(Tabl\_ID,Tabl\_Size,Reservation\_Time,Customer\_ID,Orders\_ID,Employee\_ID)

VALUES(15,6,DATE'2022-09-15',1,15,3);

-- Test Tal Table

SELECT \* FROM TABL;

COMMIT;

## **Figures**

Graphical user interface, text, application

Description automatically generated

**Restaurant Table**

Table

Description automatically generated

**Drink Menu Table**

Table

Description automatically generated

**Food menu table**

Graphical user interface, application

Description automatically generated

**Employee table**

Table

Description automatically generated

**Customer Table**

A picture containing text, light

Description automatically generated

**Order Table**

Table

Description automatically generated

**Tabl Table**

# **Queries**

## **Query 1**

### Purpose

In this query we select the last name and first name combine as a full name and select employee information such as title, phone number, and birthday from employee who from Atlanta or has a first name as William, and sort the list by the birthday

### Code

SELECT last\_name||','||first\_name AS "Full name", title AS "Job Title", phone AS "Phone Number",boa AS "Birthday"

FROM employee

WHERE city IN ('Atlanta') OR first\_name IN ('William')

ORDER BY boa;

### Figure

Table

Description automatically generated

## **Query 2**

### Purpose

We have an input for the second letter from the food name, and we displace all food name, calories, and price that are match the second letter from the food name letter o or r would work. And we sort the list by the food price.

### Code

ACCEPT letter PROMPT 'Enter the letter is contained in the food name'

SELECT food\_name AS "Name", '$'||food\_price AS "Price", calories AS"Calories"

FROM food\_menu

WHERE food\_name LIKE '\_&letter%'

ORDER BY food\_price;

### Figure

**Graphical user interface, text

Description automatically generated**

**Graphical user interface, text, application, email

Description automatically generated**

## **Query 3**

### Purpose

In this query we select the drink name and price form drink menu. we only present the data when drink price is between 3 and 10. We only show the top 5 rows and sort the list by the drink price from low to high.

### Code

SELECT drink\_name AS "Name", '$'||drink\_price AS "Price"

FROM drink\_menu

WHERE drink\_price BETWEEN 3 AND 10

ORDER BY drink\_price ASC

FETCH NEXT 5 ROWS ONLY;

### Figure

Table

Description automatically generated with low confidence

## **Query 4**

### Purpose

The purpose of this query is to display the restaurant address information in a form that restaurant name and its address will be display in one line, and to the right of the restaurant name can only contain up to 30 spaces, and the empty space will be filled with '-'We only select the restaurant with street end with 'NE' or 'SE' and we sort the list we restaurant name; The list is sort by the name.

### Code

SELECT RPAD(restaurant\_name ,30,'-') ||street ||', '||city||', '||state||', '||zip AS "Restaurant address information"

FROM Restaurant

WHERE SUBSTR(street,-2,2) IN('SE') OR SUBSTR(street,-2,2) IN('NE')

ORDER BY restaurant\_name;

### Figure

A picture containing text

Description automatically generated

## **Query 5**

### Purpose

In this query, we calculate the age of the employee’s base on their birthday, and we find the first Monday after their birthday. We display the employee’s name, position, age, and first Monday after born, and we only display employees who are chef or manager. We sort the list by age in a descending order.

### Code

SELECT first\_name||' '||last\_name AS "Employee name",title AS "Employee position",

TRUNC((SYSDATE-boa)/365) AS "Employee Age", NEXT\_DAY(boa,'Monday') AS "First monday after born"

FROM employee

WHERE title IN('Manager') OR title IN('Chef')

ORDER by TRUNC((SYSDATE-boa)/365) DESC;

### Figure

Table

Description automatically generated

## **Query 6**

### Purpose

purpose of the query is to display the employee’s full name, birth month and year in one single line. Moreover, we only display employees who are born after January 1st, 2022, and the first latter of the first name is not S. The list will be sorted by the first name of the employee

### Code

SELECT first\_name||' ' ||last\_name ||TO\_CHAR(BOA,'fm" is born on" Mon,YYYY') AS "Employee Birth Month and Year"

FROM employee

WHERE BOA < TO\_DATE('January 1,2000', 'Month DD, YYYY') AND SUBSTR(first\_name,1,1) NOT IN 'S'

ORDER BY first\_name;

### Figure

Graphical user interface, text, application

Description automatically generated

## **Query 7**

### Purpose

In this query, we categorize the healthy level of the food to very healthy, healthy, not too healthy, and unhealthy, the calories amount. If calories lower than 200 is very healthy, if between 200 and 350 is healthy, if between 350 to 400 not too healthy, any number higher will be unhealthy. We display the food name, price, calories, and healthy level, and sorted the list with calories and then food price.

### Code

SELECT food\_name AS "Name", '$' || food\_price AS "Price", calories||' Cal' AS"Calories",

(CASE WHEN calories<=200 THEN 'Very Healthy'

WHEN calories>200 AND calories<350 THEN 'Healthy'

WHEN calories>=350 AND calories<400 THEN 'NOT Too Healthy'

ELSE 'Unhealthy' END) AS "Healthy Level"

FROM FOOD\_MENU

ORDER BY calories,food\_price;

### Figure

Table

Description automatically generated

## **Query 8**

### Purpose

In this query, we try to find the restaurant most expensive and the cheapest food price. And we only display the result for those restaurants that have more than one price listed. We sorted the list from the lower price to the higher price.

### Code

SELECT restaurant.restaurant\_name AS "Name",

TRIM(TO\_CHAR(MAX(food\_menu.food\_price),'L99G99D99'))AS "Most expensive Price",

TRIM(TO\_CHAR(MIN(food\_menu.food\_price),'L99G99D99')) AS"Cheapest Price"

FROM food\_menu

JOIN restaurant ON food\_menu.restaurant\_id = restaurant.restaurant\_id

GROUP BY restaurant.restaurant\_name

HAVING MAX(food\_menu.food\_price)!= MIN(food\_menu.food\_price)

ORDER BY MAX(food\_menu.food\_price);

SELECT \* FROM FOOD\_MENU;

### Figure

Table

Description automatically generated

## **Query 9**

### Purpose

The purpose of the query is to find the oldest, youngest, and average in each job title, and we only want to know about the title that the average age is between 25 and 40. We sorted the list with the job title.

### Code

SELECT title AS "Job Title",

MAX(TRUNC((SYSDATE - boa) /365))||' years old'AS "Oldest",

MIN(TRUNC((SYSDATE - boa) /365)||' years old')AS "Youngest",

ROUND(AVG(TRUNC((SYSDATE - boa) /365)))||' years old'AS"Average"

FROM employee

GROUP BY title

HAVING ROUND(AVG(TRUNC((SYSDATE - boa) /365))) BETWEEN 25 AND 40

ORDER BY title;

### Figure

Table

Description automatically generated

## **Query 10**

### Purpose

The purpose of this query is to get the information of employee names, the restaurant name that they are working at, and the job title they have. To do so, we need to join the employee table and restaurant table.

### Code

SELECT e.first\_name||' '|| e.last\_name||' is working at '|| r.restaurant\_name||' as a ' || e.title AS "Employee name, restaruant, and position"

FROM employee e JOIN restaurant r USING(restaurant\_id)

ORDER BY e.title , e.first\_name;

### Figure

Text

Description automatically generated

## **Query 11**

### Purpose

In this query, we joined the customer, tabl, and orders tables together. The purpose is to obtain the reservation information of who is still waiting, what table is available, and the tab information of table that has an order.

### Code

SELECT (CASE WHEN t.tabl\_id IS NULL AND c.first\_name IS NOT NULL THEN c.first\_name||' is still waiting for table'

WHEN t.tabl\_id IS NOT NULL AND c.first\_name IS NULL THEN 'Table ' || tabl\_id||' is available'

ELSE c.first\_name||' had an order on '|| TO\_CHAR(o.times,'mm/dd/hh:mi')|| ', and the total price of table '|| t.tabl\_id || ' is '|| TRIM(TO\_CHAR(o.price,'L999D99'))

END) AS "Table information and reservation information"

FROM customer c FULL OUTER JOIN tabl t

ON (c.customer\_id = t.customer\_id)

FULL OUTER JOIN orders o

ON (t.orders\_id = o.orders\_id)

ORDER BY t.reservation\_time NULLS FIRST, c.customer\_id NULLS FIRST;

### Figure

Text

Description automatically generated

## **Query 12**

### Purpose

The purpose of the query is to display the food name price, and calories for those food that the price is lower than the average price, sort result with food name.

### Code

SELECT food\_name AS "Name", '$'||Food\_Price AS "Price", calories ||' cal'AS "Calories"

FROM food\_menu main

WHERE food\_price < (SELECT AVG(food\_price)

FROM food\_menu sub

)

ORDER BY food\_name;

### Figure

Graphical user interface, text, application

Description automatically generated

## **Query 13**

### Purpose

The purpose of the query is to find the cities that are in Atlanta, the result will be sorted by the restaurant name.

### Code

SELECT restaurant\_name AS "Restaurant Name", Street||' '||city || ', '||state AS "Address"

FROM restaurant

WHERE city IN (SELECT city

FROM restaurant

WHERE city = 'Atlanta')

ORDER BY restaurant\_name;

### Figure

Text

Description automatically generated

## **Query 14**

### Purpose

The purpose of this query is to put both drink menu and food menu together to obtain the name, price, type of selection, price level, and the restaurant that they belong to. We sort the list with restaurant name, then selection type, then price level, and finally with price.

### Code

SELECT d.drink\_name AS "Name", '$'|| d.drink\_price AS"Price", 'Drink' AS "Type",

(CASE WHEN d.drink\_price < 5 THEN 'Cheap'

WHEN d.drink\_price >= 5 AND d.drink\_price<10 THEN 'Medium'

ELSE 'Expensive'

END) AS "Pirce Levl",r.restaurant\_name AS "Restaurant"

FROM drink\_menu d JOIN Restaurant r USING(restaurant\_ID)

UNION ALL

SELECT f.food\_name, '$'|| f.food\_price, 'Food',

(CASE WHEN f.food\_price <10 THEN 'Cheap'

WHEN f.food\_price >=10 AND f.food\_price < 20 THEN 'Medium'

ELSE 'Expensive'

END),r.restaurant\_name

FROM food\_menu f JOIN Restaurant r USING(restaurant\_ID)

ORDER BY 5,3,4,2;

### Figure

Graphical user interface, table

Description automatically generated

## **Query 15**

### Purpose

The purpose of this query is to find out the last names that both employees and customers have. We sort the list with the last name.

### Code

SELECT 'Both Employee and Customer has this last name: '||last\_name AS "Common Last Name"

FROM Employee

INTERSECT

SELECT 'Both Employee and Customer has this last name: '|| last\_name

FROM Customer

ORDER BY 1;

### Figure

Graphical user interface, text, application, email

Description automatically generated

# **Schema Objects**

## **View 1**

CREATE OR REPLACE VIEW employee\_restaurant\_view (id,firstName,lastName,phone, restaurant)

AS SELECT e.employee\_ID,e.first\_name, e.last\_name, e.phone, r.restaurant\_name

FROM employee e JOIN restaurant r USING(restaurant\_ID);

SELECT \* FROM employee\_restaurant\_view;

SELECT firstName || ' '|| lastName As "Name", restaurant AS "Restaurant"

From employee\_restaurant\_view

WHERE restaurant in('Gaja Korean Bar','Lomi Restaurant Bar');

### Figure

Table

Description automatically generated

## **View 2**

CREATE OR REPLACE VIEW employee\_title\_view (name, title, employees\_count)

AS

SELECT r.restaurant\_name, e.title, COUNT(e.title) AS "Count\_title"

From restaurant r JOIN employee e USING(restaurant\_id)

JOIN tabl t USING(employee\_id)

HAVING COUNT(e.title)!=2

GROUP BY r.restaurant\_name,e.title

ORDER BY 1,2;

SELECT \* FROM employee\_title\_view;

SELECT name AS "Restaurant\_name", title AS "Position",employees\_count As"Number of people"

FROM employee\_title\_view

WHERE title NOT IN 'Cashier'

ORDER BY title;

### Figure

Graphical user interface, table

Description automatically generated

## **Index**

CREATE INDEX restaurant\_name\_index

ON restaurant(restaurant\_name);

## **Flashback**

CREATE TABLE TEMP\_SP (name VARCHAR(45));

INSERT INTO TEMP\_SP VALUES ('Team6');

SELECT \* FROM TEMP\_SP;

DROP TABLE TEMP\_SP;

SELECT \* FROM TEMP\_SP;

SELECT \* FROM user\_RECYCLEBIN;

FLASHBACK TABLE TEMP\_SP TO BEFORE DROP;

SELECT \* FROM TEMP\_SP;