

PROJECT PROPOSAL

RESTAURANT

**students name :**

▪ Mayadh Altunsi 2110871

▪ Hadel Alghamdi 2210609 ▪ Reema Alghamdi 2009891

Dr. Amatulrahman Muhammad al-Harbi

First Semester 1445 A.H – 2023 A.D.

# PROJECT CONTENTS

1 ...................................................................................................................................................Title

3.......................................................................................................................................... Introduction

3 ....................................................................................................................................About the project

4......................................................................................................................................Database Entities

5 ............................................................................................................................................... Phase2

5............................................................................................................Entity Relationship Diagram "ERD"

6.................................................................................................................................................. Phase3

6.................................................................................................... Relational diagram & Normalization

7…………......................................................................................................... Normalization in Steps

10................................................................................................................................................ Phase4

10................................................................................................... Physical Database Implementation

11...........................................................................................................................Few Table Outputs

15...............................................................................................................................Database Query

# Introduction

In recent years, the database has become an important and indispensable part in all stores. Examples of these stores include restaurants. Instead of the traditional method of storing information, a database must be created and developed to improve restaurant management, enhance customer service, provide a customized solution for effective daily operations, and improve guest experiences.

# About the project

Our project aims to develop or create a database of a specific restaurant and simplifying and organizing its unique operations. It includes menu items, pricing, customers, orders, reservations, Review, customer information and employee information for that specific establishment.

The project includes the following steps:

▪ Organizing all data and facilitating access to it

▪ Providing a better and easier customer experience

▪ Improving the quality of the restaurant and customer feedback

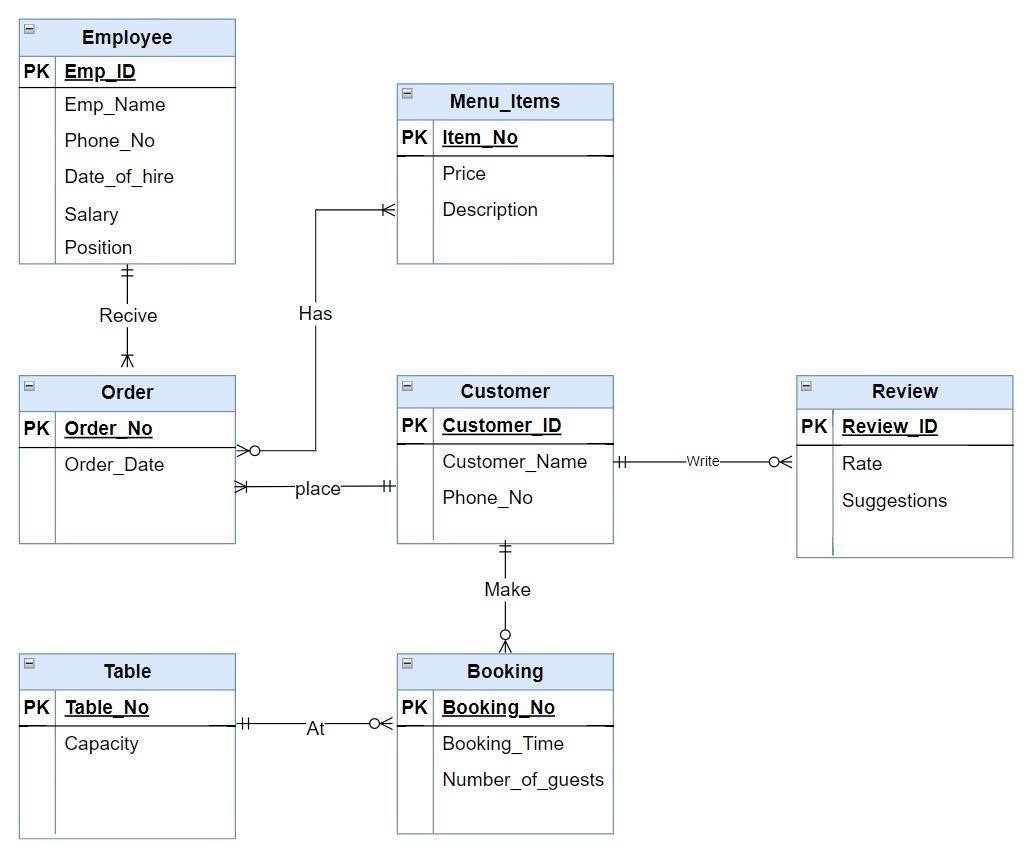
▪ Maintain a detailed record of employee information and all customer invoices

Database Entities :

1. Employee
2. Customer
3. Order
4. Menu Item
5. Booking
6. Review
7. Table

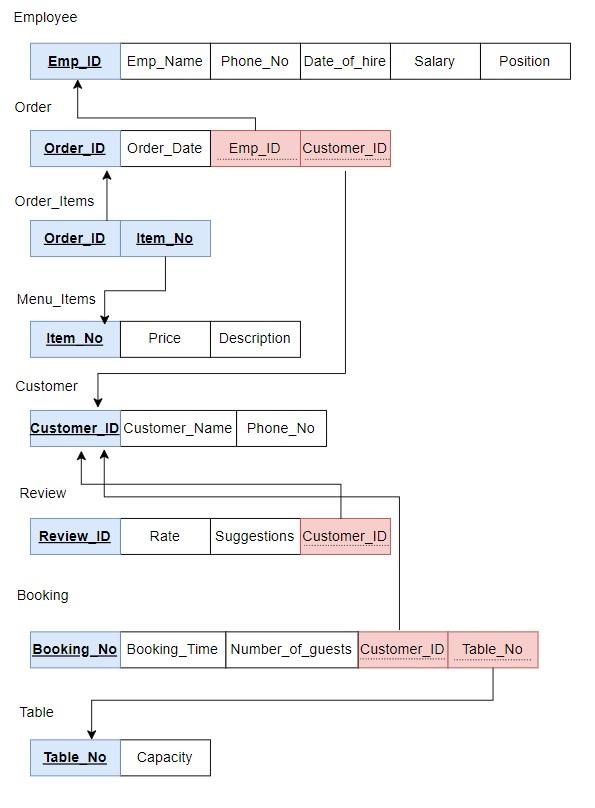
# Phase2

Entity Relationship Diagram "ERD"



# Phase3

Relational diagram



Normalization in Steps

|  |
| --- |
| **Functional Dependencies** |
| **FD** **Emp\_ID** → **Emp\_Name , Phone\_No , Date\_of\_hire , Salary ,Position** |
| **FD Order\_ID** → **Order\_Date ,Emp\_ID , Customer\_ID** |
| **FD Item\_No** → **Price , Description** |
| **FD Customer\_ID** → **Customer\_Name ,Phone\_No** |
| **FD Review\_ID** → **Rate, Suggestions ,Customer\_ID** |
| **FD Booking\_ID**→ **Booking\_Time , Number\_of\_guests , Customer\_ID, Table\_No** |
| **FD**  **Table\_No** → **Capacity** |

**The condition for 1NF:**

1-All relation must have unique primary key.

2-Remove repeating group or remove multi value attribute.

3-remove composite attribute.

Schema already satisfied to the 1Nf rules, all attribute atomic.

**The condition for 2NF:**

1-Remove partial functional dependencies.

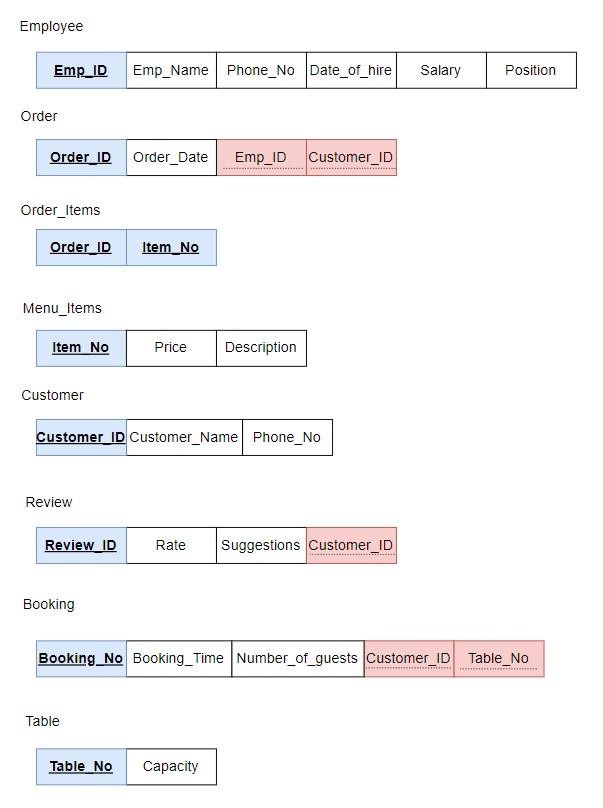
2- All regular attribute depended on a primary key or composite primary key

Schema already satisfied to the 2Nf rules.

**The condition for 3NF:**

1-Remove transitive dependencies.

The schema already in 3NF; No transitive dependency.



Create the normalized tables using liveSQL:

A white object with a black circle

Description automatically generatedA white screen with a black screen

Description automatically generated

A white rectangular object with a black line

Description automatically generated

Populate tables with 5 rows at least:

1. Employee Table

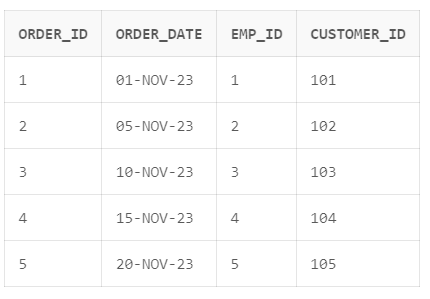


1. Customer Table

A screenshot of a phone number

Description automatically generated

1. Orders Table



1. Menu Items Table

A screenshot of a menu

Description automatically generated

1. Order Items Table

A screenshot of a cell phone

Description automatically generated

1. Booking Table

A white sheet with black text

Description automatically generated

1. Review Table

A white text on a white background

Description automatically generated

1. Table Info Table

A screenshot of a white sheet

Description automatically generated

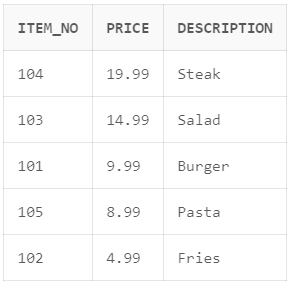
Design and implement 4 queries.

1. Get the menu items sorted by their prices in descending order.

**SELECT Item\_No, Price, Description**

**FROM Menu\_Items**

**ORDER BY Price DESC;**



1. Retrieve number of bookings made by each customer.

**SELECT Customer\_ID, COUNT(Booking\_No) AS TotalBookings**

**FROM Booking**

**GROUP BY Customer\_ID;**



1. Retrieve orders with customer names placed between January and November:

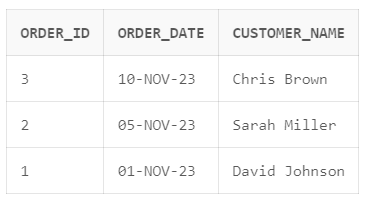
**SELECT Orders.Order\_ID, Orders.Order\_Date, Customer.Customer\_Name**

**FROM Orders**

**JOIN Customer ON Orders.Customer\_ID = Customer.Customer\_ID**

**WHERE Orders.Order\_Date BETWEEN DATE'2023-01-01' AND DATE'2023-11-10'**

**ORDER BY Orders.Order\_Date DESC;**



1. Retrieve customers who have placed orders for Pasta:

**SELECT Customer\_ID, Customer\_Name**

**FROM Customer**

**WHERE Customer\_ID IN (**

**SELECT Customer\_ID**

**FROM Orders**

**WHERE Order\_ID IN (**

**SELECT Order\_ID**

**FROM Orders\_Items**

**WHERE Item\_No IN (**

**SELECT Item\_No**

**FROM Menu\_Items**

**WHERE Description = 'Pasta'**

**)**

**)**

**);**

A close-up of a white box

Description automatically generated

Design two stored procedures:

1. Create a PARAMETER based SELECT QUERY stored procedure which return records based on parameters?

- It retrieves all orders associated with a specific customer based on the provided customer ID.

**CREATE OR REPLACE PROCEDURE GetOrdersByCustomer(CustomerID IN NUMBER)**

**AS**

**CURSOR executive IS**

**SELECT \***

**FROM Orders O**

**WHERE O.Customer\_ID = CustomerID;**

**BEGIN**

**FOR v\_cursrec IN executive LOOP**

**dbms\_output.put\_line('The customer with this ID: '|| v\_cursrec. Customer\_ID||' , Has ordered order number: '|| v\_cursrec. Order\_ID);**

**END LOOP;**

**END GetOrdersByCustomer;**

**A screenshot of a computer

Description automatically generated**

1. Create an UPDATE query based stored procedure.

* this procedure update the price of certain item.

**CREATE PROCEDURE UpdateItemPrice( ITEMNO IN NUMBER, NewPrice IN NUMBER)**

**AS**

**BEGIN**

**UPDATE Menu\_Items**

**SET Price = NewPrice**

**WHERE ITEM\_NO = ITEMNO;**

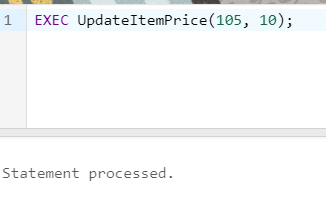
**END UpdateItemPrice;**

* **ITEM BEFORE :**

A screenshot of a menu

Description automatically generated

* **ITEM AFTER:**

****

**A white square with black lines

Description automatically generated with medium confidence**