
# AMERICAN INTERNATIONAL UNIVERSITY–BANGLADESH (AIUB)

## DEPARTMENT OF COMPUTER SCIENCE(CS)

## FACULTY OF SCIENCE & TECHNOLOGY

## INTRODUCTION TO DATABASE

## Fall 2022-2023

**Section: A || Group No: 02**

# Food Delivery Management System

**SUBMITTED TO**

JUENA AHMED NOSHIN

**Submitted By**

| Name | ID | Contribution |
|---|---|---|
| 1. Maeed Ahammed | **22-46280-1** | ER-Diagram, Normalisation, Table,Creation,Data Insertion |
| 2. Nazifa Tabushum Turaba | **22-46271-1** | ER-Diagram, Normalisation,Table Creation, Data Insertion |
| 3. Shahriyar Rahman Simoon | **22-46247-1** | Scenario, Sub-Query, Join,View, Single-Row Function, Multi-row Function |
| 4. Umme Kaynat Afra | **22-47167-1** | Scenario, Sub-Query, Join,View, Single-Row Function, Multi-row Function |

Date of Submission: **26 December, 2022**

## Content page:

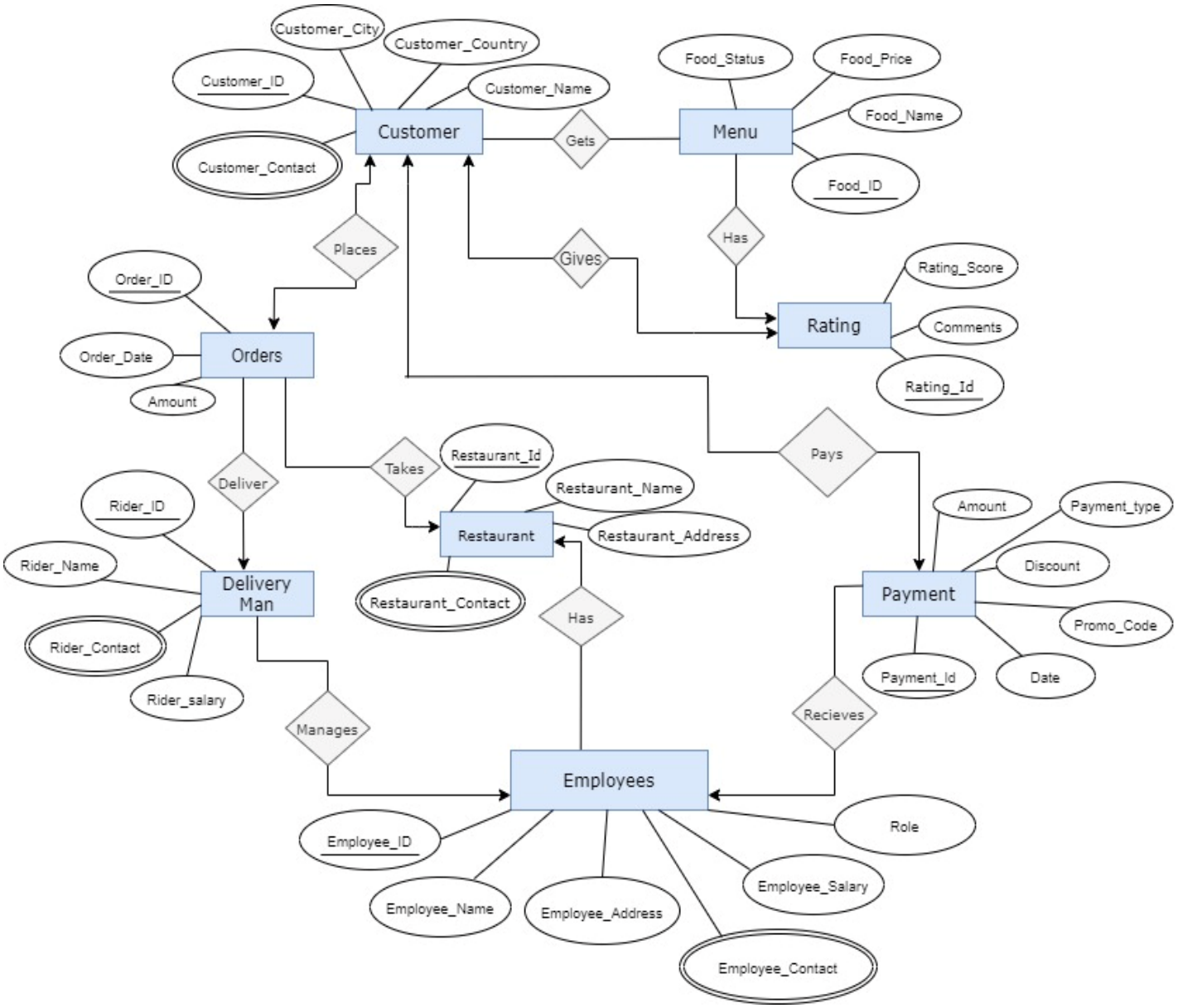
## Introduction:

This DBMS contains data about a food delivery management system. By logging into this database, a user can access many variations, and manage restaurants, orders, and employee as well. The customer can organize and manage their food orders and must pay an adequate price. Customer can perform their actions to manage this food delivery system, to fulfill their craving desires.

## Scenario

The Food Delivery Management System offers an efficient working system for Restaurant-Catering. This system has a Menu in their system where food_name, food_id, food_price, and the status of food are stored. A customer can request menu details. One customer can submit a single order. A customer is identified as customer_id, customer_name, customer_address, and contact_number. However, customers can give ratings on the menu. The rating has a unique rating_id, rating_score, and comment. Order details are associated with order_id, order_date, and amount. When a customer completes an order, then they process their payments like payment_id, amount, paid_by, discount, and date. A food delivery management system has many delivery men to serve food to customers' residences. The attributes for the delivery man are rider_id, rider_name, rider_address, rider_contact, and Rider_salary. Each order should produce one serve, and each service is conducted by one rider. Also, there are several employees who run the system and manage the Delivery man and payments of the customers. Their attributes are employee_id, employee_name, employee_address, employee_contact, salary, and their role.

**ER MODEL**


Customer_City
Customer_Country
Customer_ID
Customer_Name
Customer
Customer_Contact
Gets
Food_Status
Food_Price
Food_Name
Menu
Food_ID
Has
Places
Gives
Order_ID
Order_Date
Amount
Orders
Rating_Score
Comments
Rating_Id
Rating
Pays
Takes
Restaurant_Id
Restaurant_Name
Restaurant_Address
Restaurant
Restaurant_Contact
Has
Deliver
Rider_ID
Rider_Name
Rider_Contact
Rider_salary
Delivery Man
Manages
Amount
Payment_type
Discount
Promo_Code
Payment_Id
Date
Payment
Recieves
Employees
Employee_ID
Employee_Name
Employee_Address
Employee_Salary
Employee_Contact
Role

## Normalization

1. UNF: Gets (Customer_id, Customer_name, Customer_contact, Customer_city, Customer_country, Food_id, Food_name, Food_status, Food_Price)

1NF: Customer_contact Multivalued Attribute.

- Customer_id, Customer_name, Customer_contact, Customer_city, Customer_country, Food_id, Food_name, Food_status, Food_Price

2NF:

- Customer_id, Customer_name, Customer_contact, Customer_city, Customer_country
- Food_id, Food_name, Food_status, Food_Price

3NF:

- Customer_id, Customer_name, Customer_contact
- Food_id, Food_name, Food_status, Food_Price
- Customer_city, Customer_country

TABLE:

- Customer_id, Customer_name, Customer_contact, **a_id**
- Food_id, Food_name, Food_status, Food_Price
- a_id, Customer_city, Customer_country
- **Customer_id, Food_id**

2. UNF: Gives (Customer_id, Customer_name, Customer_contact, Customer_city, Customer_country, Rating_id, Rating_Score, Comments)

1NF: Customer_contact Multivalued Attribute.

- Customer_id, Customer_name, Customer_contact, Customer_city, Customer_country, Rating_id, Rating_Score, Comments

2NF:

- Customer_id, Customer_name, Customer_contact, Customer_city, Customer_country
- Rating_id, Rating_Score, Comments

3NF:

- Customer_id, Customer_name, Customer_contact
- Rating_id, Rating_Score, Comments

- Customer_city, Customer_country

TABLE:

- Customer_id, Customer_name, Customer_contact, **a_id, Rating_id**
- Rating_id, Rating_Score, Comments
- a_id, Customer_city, Customer_country

3. UNF: Has (Food_id, Food_name, Food_status, Food_Price, Rating_id, Rating_Score, Comments)

1NF: There is no Multivalued Attribute.

- Food_id, Food_name, Food_status, Food_Price, Rating_id, Rating_Score, Comments

2NF:

- Food_id, Food_name, Food_status, Food_Price
- Rating_id, Rating_Score, Comments

3NF: There is no Transitive Dependency so the relation is already in 3NF.

- Food_id, Food_name, Food_status, Food_Price
- Rating_id, Rating_Score, Comments

TABLE:

- Food_id, Food_name, Food_status, Food_Price, **Rating_id**
- Rating_id, Rating_Score, Comments

4. UNF: Pays (Customer_id, Customer_name, Customer_contact, Customer_city, Customer_country, Payment_id, Amount, Payment_type, Discount, Promo_code, Date)

1NF: Customer_contact Multivalued Attribute.

- Customer_id, Customer_name, Customer_contact, Customer_city, Customer_country, Payment_id, Amount, Payment_type, Discount, Promo_code, Date

2NF:

- Customer_id, Customer_name, Customer_contact, Customer_city, Customer_country
- Payment_id, Amount, Payment_type, Discount, Promo_code, Date

3NF:

- Customer_id, Customer_name, Customer_contact
- Payment_id, Amount, Payment_type, Discount, Promo_code, Date

- Customer_city, Customer_country

TABLE:

- Customer_id, Customer_name, Customer_contact, **a_id,**
- Payment_id, Amount, Payment_type, Discount, Promo_code, Date, **Customer_id**
- a_id, Customer_city, Customer_country

5. UNF: Places (Customer_id, Customer_name, Customer_contact, Customer_city, Customer_country, Order_id, Order_date, Amount)

1NF: Customer_contact Multivalued Attribute.

- Customer_id, Customer_name, Customer_contact, Customer_city, Customer_country, Order_id, Order_date, Amount

2NF:

- Customer_id, Customer_name, Customer_contact, Customer_city, Customer_country
- Order_id, Order_date, Amount

3NF:

- Customer_id, Customer_name, Customer_contact
- Order_id, Order_date, Amount
- Customer_city, Customer_country

TABLE:

- Customer_id, Customer_name, Customer_contact, **a_id**
- Order_id, Order_date, Amount**, Customer_id**
- a_id, Customer_city, Customer_country

6. UNF: Recieves (Payment_id, Amount, Payment_type, Discount, Promo_code, Dates, Employee_id , Employee_name, Employee_address, Employee_contact, Employee_salary, Role)

1NF: Employee_contact Multivalued Attribute.

- Payment_id, Amount, Payment_type, Discount, Promo_code, Dates, Employee_id , Employee_name, Employee_address, Employee_contact, Employee_salary, Role

2NF:

- Payment_id, Amount, Payment_type, Discount, Promo_code, Dates
- Employee_id , Employee_name, Employee_address, Employee_contact, Employee_salary, Role

3NF:

There is no Transitive Dependency so the relation is already in 3NF.

- Payment_id, Amount, Payment_type, Discount, Promo_code, Dates
- Employee_id , Employee_name, Employee_address, Employee_contact, Employee_salary, Role

TABLE:

- Payment_id, Amount, Payment_type, Discount, Promo_code, Dates, **Employee_id**
- Employee_id , Employee_name, Employee_address, Employee_contact, Employee_salary, Role

7. UNF: Manages (Employee_id , Employee_name, Employee_address, Employee_contact, Employee_salary, Role, Rider_id, Rider_name, Rider_contact, Rider_salary)

1NF: Employee_contact and Rider_contact both are Multivalued Attributes.

- Employee_id, Employee_name, Employee_address, Employee_contact, Employee_salary, Role, Rider_id, Rider_name, Rider_contact, Rider_salary

2NF:

- Employee_id , Employee_name, Employee_address, Employee_contact, Employee_salary, Role
- Rider_id, Rider_name, Rider_contact, Rider_salary

3NF:

There is no Transitive Dependency so the relation is already in 3NF.

- Employee_id , Employee_name, Employee_address, Employee_contact, Employee_salary, Role
- Rider_id, Rider_name, Rider_contact, Rider_salary

TABLE:

- Employee_id , Employee_name, Employee_address, Employee_contact, Employee_salary, Role
- Rider_id, Rider_name, Rider_contact, Rider_salary, **Employee_id**

8. UNF: Has (Employee_id , Employee_name, Employee_address, Employee_contact, Employee_salary, Role, Restaurant_id, Restaurant_name, Restaurant_address)

1NF: Employee_contact both are Multivalued Attributes.

- Employee_id, Employee_name, Employee_address, Employee_contact, Employee_salary, Role, Restaurant_id, Restaurant_name, Restaurant_address

2NF:

- Employee_id , Employee_name, Employee_address, Employee_contact, Employee_salary, Role

- Restaurant_id, Restaurant_name, Restaurant_address

3NF:

There is no Transitive Dependency, so the relation is already in 3NF.

- Employee_id , Employee_name, Employee_address, Employee_contact, Employee_salary, Role
- Restaurant_id, Restaurant_name, Restaurant_address

TABLE:

- Employee_id , Employee_name, Employee_address, Employee_contact, Employee_salary, Role, **Restaurant_id**
- Restaurant_id, Restaurant_name, Restaurant_address

9. UNF: Takes (Order_id, Order_date, Amount, Restaurant_id, Restaurant_name, Restaurant_address)

1NF: There are no Multivalued Attributes.

- Order_id, Order_date, Amount, Restaurant_id, Restaurant_name, Restaurant_address

2NF:

- Restaurant_id, Restaurant_name, Restaurant_address
- Order_id, Order_date, Amount

3NF:

There is no Transitive Dependency, so the relation is already in 3NF.

- Restaurant_id, Restaurant_name, Restaurant_address
- Order_id, Order_date, Amount

TABLE:

- Restaurant_id, Restaurant_name, Restaurant_address
- Order_id, Order_date, Amount, **Restaurant_id**

10. UNF: Delivers (Order_id, Order_date, Amount, Rider_id, Rider_name, Rider_contact, Rider_salary)

1NF: Rider_contact is a Multivalued Attributes.

- Order_id, Order_date, Amount, Rider_id, Rider_name, Rider_contact, Rider_salary

2NF:

- Order_id, Order_date, Amount
- Rider_id, Rider_name, Rider_contact, Rider_salary

3NF:

There is no Transitive Dependency, so the relation is already in 3NF.

- Order_id, Order_date, Amount
- Rider_id, Rider_name, Rider_contact, Rider_salary

TABLE:

- Order_id, Order_date, Amount, **Rider_id**
- Rider_id, Rider_name, Rider_contact, Rider_salary

## Temporary TABLE:

- ~~Customer_id, Customer_name, Customer_contact, **a_id**~~
- ~~Food_id, Food_name, Food_status, Food_Price~~
- a_id, Customer_city, Customer_country
- **Customer_id, Food_id**
- Customer_id, Customer_name, Customer_contact, **a_id, Rating_id**
- Rating_id, Rating_Score, Comments
- ~~a_id, Customer_city, Customer_country~~
- Food_id, Food_name, Food_status, Food_Price, **Rating_id**
- ~~Rating_id, Rating_Score, Comments~~
- ~~Customer_id, Customer_name, Customer_contact, **a_id,**~~
- Payment_id, Amount, Payment_type, Discount, Promo_code, Date, **Customer_id**
- ~~a_id, Customer_city, Customer_country~~
- ~~Customer_id, Customer_name, Customer_contact, **a_id**~~
- Order_id, Order_date, Amount, **Customer_id**

- ~~a_id, Customer_city, Customer_country~~
- Payment_id, Amount, Payment_type, Discount, Promo_code, Dates, **Employee_id**
- ~~Employee_id , Employee_name, Employee_address, Employee_contact, Employee_salary, Role~~
- ~~Employee_id , Employee_name, Employee_address, Employee_contact, Employee_salary, Role~~
- Rider_id, Rider_name, Rider_contact, Rider_salary, **Employee_id**
- Employee_id , Employee_name, Employee_address, Employee_contact, Employee_salary, Role, **Restaurant_id**
- Restaurant_id, Restaurant_name, Restaurant_address
- ~~Restaurant_id, Restaurant_name, Restaurant_address~~
- Order_id, Order_date, Amount, **Restaurant_id**
- Order_id, Order_date, Amount, **Rider_id**
- ~~Rider_id, Rider_name, Rider_contact, Rider_salary~~

## FINAL TABLE:

- a_id, Customer_city, Customer_country
- **Customer_id, Food_id**
- Customer_id, Customer_name, Customer_contact, **a_id, Rating_id**
- Rating_id, Rating_Score, Comments
- Food_id, Food_name, Food_status, Food_Price, **Rating_id**
- Payment_id, Amount, Payment_type, Discount, Promo_code, Date, **Customer_id**

- Order_id, Order_date, Amount**, Customer_id**

- Payment_id, Amount, Payment_type, Discount, Promo_code, Dates, **Employee_id**

- Rider_id, Rider_name, Rider_contact, Rider_salary**, Employee_id**

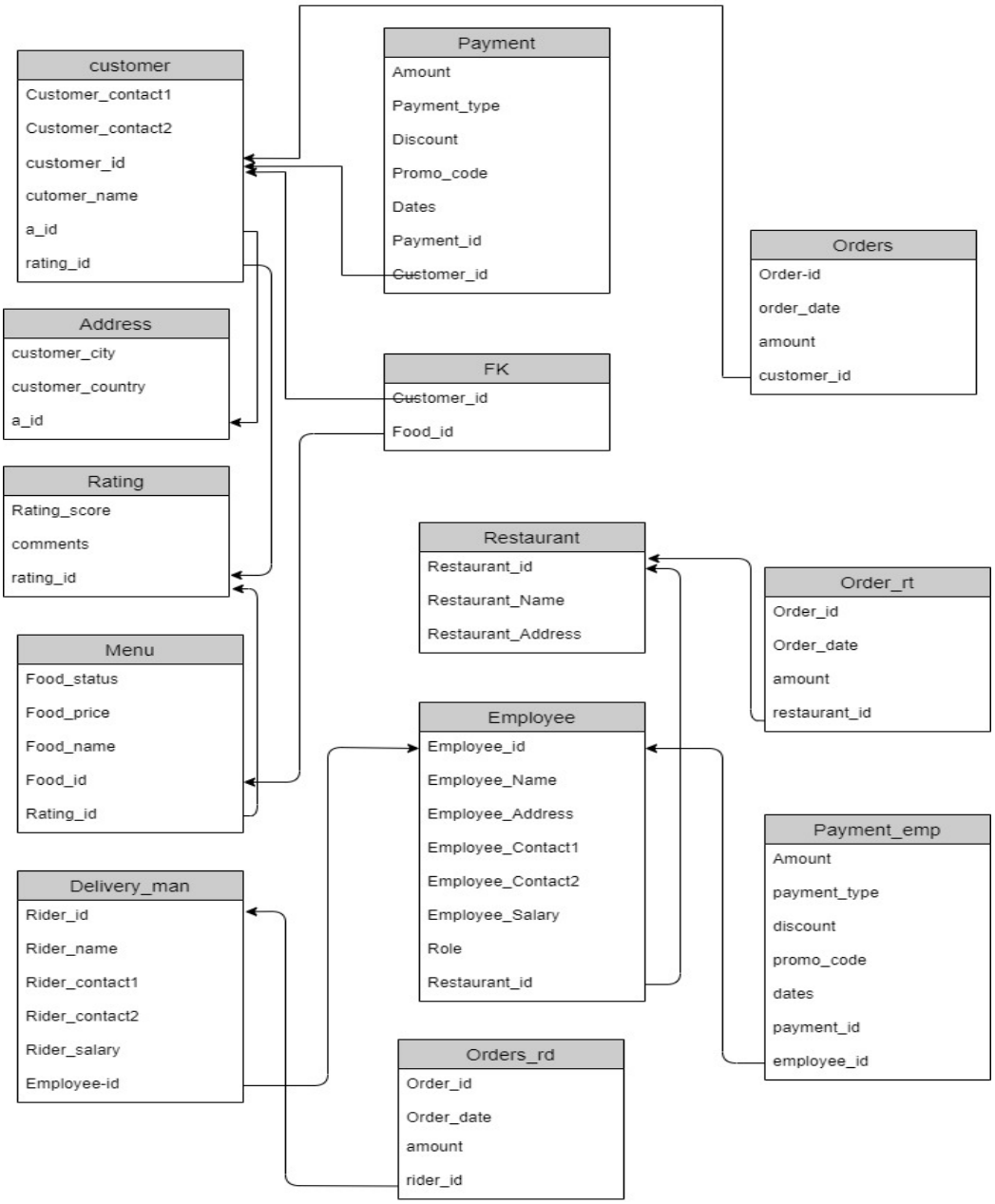
- Employee_id , Employee_name, Employee_address, Employee_contact, Employee_salary, Role, **Restaurant_id**

- Restaurant_id, Restaurant_name, Restaurant_address

- Order_id, Order_date, Amount, **Restaurant_id**

- Order_id, Order_date, Amount, **Rider_id**

# Schema Diagram:



# TABLE CREATION:

## CUSTOMER

create table customer(customer_id number(15), customer_name varchar2(20), customer_contact1 number(15), customer_contact2 number(15), a_id number(15),rating_id number(15), primary key(customer_id), foreign key(a_id) references address(a_id), foreign key(rating_id) references rating(rating_id));

Object Type **TABLE** Object **CUSTOMER**

| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
|---|---|---|---|---|---|---|---|---|---|
| CUSTOMER | CUSTOMER_ID | Number | - | 15 | 0 | 1 | - | - | - |
| | CUSTOMER_NAME | Varchar2 | 20 | - | - | - | ✓ | - | - |
| | CUSTOMER_CONTACT1 | Number | - | 15 | 0 | - | ✓ | - | - |
| | CUSTOMER_CONTACT2 | Number | - | 15 | 0 | - | ✓ | - | - |
| | A_ID | Number | - | 15 | 0 | - | ✓ | - | - |
| | RATING_ID | Number | - | 15 | 0 | - | ✓ | - | - |

1 - 6

## ADDRESS

create table address(a_id number(15) primary key,customer_city varchar2(20), customer_country varchar2(20));

Object Type **TABLE** Object **ADDRESS**

| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
|---|---|---|---|---|---|---|---|---|---|
| ADDRESS | A_ID | Number | - | 15 | 0 | 1 | - | - | - |
| | CUSTOMER_CITY | Varchar2 | 20 | - | - | - | ✓ | - | - |
| | CUSTOMER_COUNTRY | Varchar2 | 20 | - | - | - | ✓ | - | - |

1 - 3

## RATING

create table rating(rating_id number(15) primary key,rating_score number(15), comments varchar(20));

Object Type **TABLE** Object **RATING**

| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
|---|---|---|---|---|---|---|---|---|---|
| RATING | RATING_ID | Number | - | 15 | 0 | 1 | - | - | - |
| | RATING_SCORE | Number | - | 15 | 0 | - | ✓ | - | - |
| | COMMENTS | Varchar2 | 20 | - | - | - | ✓ | - | - |
| | | | | | | | | | 1 - 3 |

## MENU

create table menu(food_id number(15),food_name varchar2(20), food_status varchar2(20), food_price number(15), rating_id number(15), primary key(food_id), foreign key(rating_id) references rating(rating_id));

Object Type **TABLE** Object **MENU**

| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
|---|---|---|---|---|---|---|---|---|---|
| MENU | FOOD_ID | Number | - | 15 | 0 | 1 | - | - | - |
| | FOOD_NAME | Varchar2 | 20 | - | - | - | ✓ | - | - |
| | FOOD_STATUS | Varchar2 | 20 | - | - | - | ✓ | - | - |
| | FOOD_PRICE | Number | - | 15 | 0 | - | ✓ | - | - |
| | RATING_ID | Number | - | 15 | 0 | - | ✓ | - | - |
| | | | | | | | | | 1 - 5 |

## PAYMENT

create table payment(payment_id number(15), customer_id number(15), amount number(15),payment_type varchar2(20), discount number(15), promo_code number(15), dates varchar2(20), primary key(payment_id), foreign key(customer_id) references customer(customer_id));

Object Type **TABLE** Object **PAYMENT**

| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
|---|---|---|---|---|---|---|---|---|---|
| PAYMENT | PAYMENT_ID | Number | - | 15 | 0 | 1 | - | - | - |
| | CUSTOMER_ID | Number | - | 15 | 0 | - | ✓ | - | - |
| | AMOUNT | Number | - | 15 | 0 | - | ✓ | - | - |
| | PAYMENT_TYPE | Varchar2 | 20 | - | - | - | ✓ | - | - |
| | DISCOUNT | Number | - | 15 | 0 | - | ✓ | - | - |
| | PROMO_CODE | Number | - | 15 | 0 | - | ✓ | - | - |
| | DATES | Varchar2 | 20 | - | - | - | ✓ | - | - |
| | | | | | | | | | 1 - 7 |

## FK

create table fk(customer_id number(15), food_id number(15), primary key(customer_id,food_id),foreign key(customer_id) references customer(customer_id), foreign key(food_id) references menu(food_id));

Object Type **TABLE** Object **FK**

| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
|---|---|---|---|---|---|---|---|---|---|
| FK | CUSTOMER_ID | Number | - | 15 | 0 | 1 | - | - | - |
| | FOOD_ID | Number | - | 15 | 0 | 2 | - | - | - |
| | | | | | | | | | 1 - 2 |

## RESTAURANT

create table restaurant(restaurant_id number(15) primary key, restaurant_name varchar2(20), restaurant_address varchar2(20));

Object Type **TABLE** Object **RESTAURANT**

| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
|---|---|---|---|---|---|---|---|---|---|
| RESTAURANT | RESTAURANT_ID | Number | - | 15 | 0 | 1 | - | - | - |
| | RESTAURANT_NAME | Varchar2 | 20 | - | - | - | ✓ | - | - |
| | RESTAURANT_ADDRESS | Varchar2 | 20 | - | - | - | ✓ | - | - |
| | | | | | | | | | 1 - 3 |

## EMPLOYEE

create table employee(employee_id number(15),employee_name varchar2(20), employee_address varchar2(20), employee_contact1 number(15), employee_contact2 number(15), employee_salary number(15), role varchar2(20), restaurant_id number(15),primary key(employee_id),foreign key(restaurant_id) references restaurant(restaurant_id));

Object Type **TABLE** Object **EMPLOYEE**

| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
|---|---|---|---|---|---|---|---|---|---|
| EMPLOYEE | EMPLOYEE_ID | Number | - | 15 | 0 | 1 | - | - | - |
| | EMPLOYEE_NAME | Varchar2 | 20 | - | - | - | ✓ | - | - |
| | EMPLOYEE_ADDRESS | Varchar2 | 20 | - | - | - | ✓ | - | - |
| | EMPLOYEE_CONTACT1 | Number | - | 15 | 0 | - | ✓ | - | - |
| | EMPLOYEE_CONTACT2 | Number | - | 15 | 0 | - | ✓ | - | - |
| | EMPLOYEE_SALARY | Number | - | 15 | 0 | - | ✓ | - | - |
| | ROLE | Varchar2 | 20 | - | - | - | ✓ | - | - |
| | RESTAURANT_ID | Number | - | 15 | 0 | - | ✓ | - | - |

1 - 8

## PAYMENT_EMP

create table payment_emp(payment_id number(15), employee_id number(15), amount number(15),payment_type varchar2(20), discount number(15), promo_code number(15), dates varchar2(20), primary key(payment_id), foreign key(employee_id) references employee(employee_id));

Object Type **TABLE** Object **PAYMENT_EMP**

| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
|---|---|---|---|---|---|---|---|---|---|
| PAYMENT_EMP | PAYMENT_ID | Number | - | 15 | 0 | 1 | - | - | - |
| | EMPLOYEE_ID | Number | - | 15 | 0 | - | ✓ | - | - |
| | AMOUNT | Number | - | 15 | 0 | - | ✓ | - | - |
| | PAYMENT_TYPE | Varchar2 | 20 | - | - | - | ✓ | - | - |
| | DISCOUNT | Number | - | 15 | 0 | - | ✓ | - | - |
| | PROMO_CODE | Number | - | 15 | 0 | - | ✓ | - | - |
| | DATES | Varchar2 | 20 | - | - | - | ✓ | - | - |

1 - 7

## DELIVERY MAN

create table delivery_man(rider_id number(15), rider_name varchar2(20), rider_contact1 number(15), rider_contact2 number(15), rider_salary number(15), employee_id number(15), primary key(rider_id), foreign key(employee_id) references employee(employee_id));

Object Type **TABLE** Object **DELIVERY_MAN**

| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
|---|---|---|---|---|---|---|---|---|---|
| DELIVERY_MAN | RIDER_ID | Number | - | 15 | 0 | 1 | - | - | - |
| | RIDER_NAME | Varchar2 | 20 | - | - | - | ✓ | - | - |
| | RIDER_CONTACT1 | Number | - | 15 | 0 | - | ✓ | - | - |
| | RIDER_CONTACT2 | Number | - | 15 | 0 | - | ✓ | - | - |
| | RIDER_SALARY | Number | - | 15 | 0 | - | ✓ | - | - |
| | EMPLOYEE_ID | Number | - | 15 | 0 | - | ✓ | - | - |

1 - 6

## ORDERS

create table orders(order_id number(15), order_date varchar2(20), amount number(15), customer_id number(14), primary key(order_id), foreign key(customer_id) references customer(customer_id));

Object Type **TABLE** Object **ORDERS**

| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
|---|---|---|---|---|---|---|---|---|---|
| ORDERS | ORDER_ID | Number | - | 15 | 0 | 1 | - | - | - |
| | ORDER_DATE | Varchar2 | 20 | - | - | - | ✓ | - | - |
| | AMOUNT | Number | - | 15 | 0 | - | ✓ | - | - |
| | CUSTOMER_ID | Number | - | 14 | 0 | - | ✓ | - | - |

1 - 4

## ORDERS_RD

create table orders_rd(order_id number(15), order_date varchar2(20), amount number(15), rider_id number(14), primary key(order_id), foreign key(rider_id) references delivery_man(rider_id));

Object Type **TABLE** Object **ORDERS_RD**

| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
|---|---|---|---|---|---|---|---|---|---|
| ORDERS_RD | ORDER_ID | Number | - | 15 | 0 | 1 | - | - | - |
| | ORDER_DATE | Varchar2 | 20 | - | - | - | ✓ | - | - |
| | AMOUNT | Number | - | 15 | 0 | - | ✓ | - | - |
| | RIDER_ID | Number | - | 14 | 0 | - | ✓ | - | - |
| | | | | | | | | | 1 - 4 |

## ORDERS_RT

create table orders_rt(order_id number(15), order_date varchar2(20), amount number(15), restaurant_id number(14), primary key(order_id), foreign key(restaurant_id) references restaurant(restaurant_id));

Object Type **TABLE** Object **ORDERS_RT**

| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
|---|---|---|---|---|---|---|---|---|---|
| ORDERS_RT | ORDER_ID | Number | - | 15 | 0 | 1 | - | - | - |
| | ORDER_DATE | Varchar2 | 20 | - | - | - | ✓ | - | - |
| | AMOUNT | Number | - | 15 | 0 | - | ✓ | - | - |
| | RESTAURANT_ID | Number | - | 14 | 0 | - | ✓ | - | - |
| | | | | | | | | | 1 - 4 |

# SEQUENCE

1) create sequence customer_seq increment by 1 start with 1 maxvalue 500 nocache nocycle;

2) create sequence address_a_id_seq increment by 1 start with 1 maxvalue 500 nocache nocycle;

3) create sequence rating_seq increment by 1 start with 1 maxvalue 500 nocache nocycle;

4) create sequence menu_seq increment by 1 start with 1 maxvalue 500 nocache nocycle;

5) create sequence payment_seq increment by 1 start with 1 maxvalue 500 nocache nocycle;

6) create sequence fk_seq increment by 1 start with 1 maxvalue 500 nocache nocycle;

7) create sequence restaurant_seq increment by 1 start with 1 maxvalue 500 nocache nocycle;

8) create sequence employee_id_seq increment by 1 start with 1 maxvalue 500 nocache nocycle;

9) create sequence payment_emp_seq increment by 1 start with 1 maxvalue 500 nocache nocycle;

10) create sequence delivery_seq increment by 1 start with 1 maxvalue 500 nocache nocycle;

11) create sequence orders_seq increment by 1 start with 1 maxvalue 500 nocache nocycle;

12) create sequence orders_rd_seq increment by 1 start with 1 maxvalue 500 nocache nocycle;

13) create sequence orders_rt_seq increment by 1 start with 1 maxvalue 500 nocache nocycle;

# User Creation:

```
CREATE USER FDMS
        IDENTIFIED BY ADMIN
```

```
CREATE USER FDMS
            IDENTIFIED BY ADMIN
```

Results Explain Describe Saved SQL History

```
User created.

0.00 seconds
```

GRANT PERMIT:

Log in to the system admin

Write the command:

GRANT CONNECT, RESOURCE TO FDMS

```
CREATE USER FDMS
            IDENTIFIED BY ADMIN
GRANT PERMIT:
            Log in to the system admin
            Write the command:


                GRANT CONNECT, RESOURCE TO FDMS
```

Results Explain Describe Saved SQL History

```
Statement processed.

0.00 seconds
```

GRANT create view

TO FDMS

```
GRANT  create view
    TO     FDMS
```

Results Explain Describe Saved SQL History

```
Statement processed.

0.00 seconds
```

# DATA INSERTION

## CUSTOMER

insert into customer values(customer_seq.NEXTVAL,'RAKIB','01978789812','01878789812',1,1);

insert into customer values(customer_seq.NEXTVAL,'TURABA','01778789813','01378789813',2,2);

insert into customer values(customer_seq.NEXTVAL,'DRISHTY','01678789814','01578789814',3,3);

insert into customer values(customer_seq.NEXTVAL,'ALIF','01778789815','01678789815',4,4);

insert into customer values(customer_seq.NEXTVAL,'EMON','01878789816','01378789816',5,5);

| CUSTOMER_ID | CUSTOMER_NAME | CUSTOMER_CONTACT1 | CUSTOMER_CONTACT2 | A_ID | RATING_ID |
|---|---|---|---|---|---|
| 1 | RAKIB | 1978789812 | 1878789812 | 1 | 1 |
| 2 | TURABA | 1778789813 | 1378789813 | 2 | 2 |
| 3 | DRISHTY | 1678789814 | 1578789814 | 3 | 3 |
| 4 | ALIF | 1778789815 | 1678789815 | 4 | 4 |
| 5 | EMON | 1878789816 | 1378789816 | 5 | 5 |

## ADDRESS

insert into address values(address_a_id_seq.NEXTVAL,'DHAKA','BANGLADESH');

insert into address values(address_a_id_seq.NEXTVAL,'JESSORE','BANGLADESH');

insert into address values(address_a_id_seq.NEXTVAL,'KHULNA','BANGLADESH');

insert into address values(address_a_id_seq.NEXTVAL,'CHANDPUR','BANGLADESH');

insert into address values(address_a_id_seq.NEXTVAL,'PABNA','BANGLADESH');

| A_ID | CUSTOMER_CITY | CUSTOMER_COUNTRY |
|---|---|---|
| 1 | DHAKA | BANGLADESH |
| 2 | JESSORE | BANGLADESH |
| 3 | KHULNA | BANGLADESH |
| 4 | CHANDPUR | BANGLADESH |
| 5 | PABNA | BANGLADESH |

## RATING

insert into rating values(rating_seq.NEXTVAL,'5','GOOD');

insert into rating values(rating_seq.NEXTVAL,'4','GOOD ENOUGH');

insert into rating values(rating_seq.NEXTVAL,'2','BAD');

insert into rating values(rating_seq.NEXTVAL,'3','AVERAGE');

insert into rating values(rating_seq.NEXTVAL,'5','GOOD');

| RATING_ID | RATING_SCORE | COMMENTS |
|---|---|---|
| 1 | 5 | GOOD |
| 2 | 4 | GOOD ENOUGH |
| 3 | 2 | BAD |
| 4 | 3 | AVERAGE |
| 5 | 5 | GOOD |

## MENU

insert into menu values(menu_seq.NEXTVAL,'KACCHI BIRYANI','AVAILABLE',750,1);

insert into menu values(menu_seq.NEXTVAL,'MUROG PULAO','UN-AVAILABLE',150,2);

insert into menu values(menu_seq.NEXTVAL,'BURGER','AVAILABLE',250,3);

insert into menu values(menu_seq.NEXTVAL,'FALUDA','AVAILABLE',150,4);

insert into menu values(menu_seq.NEXTVAL,'CHOCOLATE WAFFLE','UN-AVAILABLE',200,5);

| FOOD_ID | FOOD_NAME | FOOD_STATUS | FOOD_PRICE | RATING_ID |
|---|---|---|---|---|
| 1 | KACCHI BIRYANI | AVAILABLE | 750 | 1 |
| 2 | MUROG PULAO | UN-AVAILABLE | 150 | 2 |
| 3 | BURGER | AVAILABLE | 250 | 3 |
| 4 | FALUDA | AVAILABLE | 150 | 4 |
| 5 | CHOCOLATE WAFFLE | UN-AVAILABLE | 200 | 5 |

## PAYMENT

insert into payment values(payment_seq.NEXTVAL, 1, 750, 'ONLINE', 5, 1797, '08 NOVEMBER 2022');

insert into payment values(payment_seq.NEXTVAL, 2, 150, 'CASH', 15, 2697, '10 NOVEMBER 2022');

insert into payment values(payment_seq.NEXTVAL, 3, 250, 'CASH', 10, 7697, '10 DECEMBER 2022');

insert into payment values(payment_seq.NEXTVAL, 4, 150, 'ONLINE', 3, 1987, '15 DECEMBER 2022');

insert into payment values(payment_seq.NEXTVAL, 5, 200, 'CASH', 7, 3097, '31 DECEMBER 2022');

| PAYMENT_ID | CUSTOMER_ID | AMOUNT | PAYMENT_TYPE | DISCOUNT | PROMO_CODE | DATES |
|---|---|---|---|---|---|---|
| 1 | 1 | 750 | ONLINE | 5 | 1797 | 08 NOVEMBER 2022 |
| 2 | 2 | 150 | CASH | 15 | 2697 | 10 NOVEMBER 2022 |
| 3 | 3 | 250 | CASH | 10 | 7697 | 10 DECEMBER 2022 |
| 4 | 4 | 150 | ONLINE | 3 | 1987 | 15 DECEMBER 2022 |
| 5 | 5 | 200 | CASH | 7 | 3097 | 31 DECEMBER 2022 |

## FK

insert into fk values(1,1);

insert into fk values(2,2);

insert into fk values(3,3);

insert into fk values(4,4);

insert into fk values(5,5);

| CUSTOMER_ID | FOOD_ID |
|---|---|
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| 4 | 4 |
| 5 | 5 |

# RESTAURANT

```
insert into restaurant values(restaurant_seq.NEXTVAL,'BUKHARA','DHAKA');

insert into restaurant values(restaurant_seq.NEXTVAL,'DINE-IN','JESSORE');

insert into restaurant values(restaurant_seq.NEXTVAL,'BABY-LON','KHULNA');

insert into restaurant values(restaurant_seq.NEXTVAL,'CHILLOX','CHANDPUR');

insert into restaurant values(restaurant_seq.NEXTVAL,'DAWAT-E-DARBAR','PABNA');

insert into restaurant values(restaurant_seq.NEXTVAL,'PIZZERIA','DHAKA');

insert into restaurant values(restaurant_seq.NEXTVAL,'DOMINOES','RAJSHAHI');

insert into restaurant values(restaurant_seq.NEXTVAL,'CHEEZ','DHAKA');

insert into restaurant values(restaurant_seq.NEXTVAL,'PIZZA HUT','DHAKA');
```

| RESTAURANT_ID | RESTAURANT_NAME | RESTAURANT_ADDRESS |
|---|---|---|
| 1 | BUKHARA | DHAKA |
| 2 | DINE-IN | JESSORE |
| 3 | BABY-LON | KHULNA |
| 4 | CHILLOX | CHANDPUR |
| 5 | DAWAT-E-DARBAR | PABNA |
| 6 | PIZZERIA | DHAKA |
| 7 | DOMINOES | RAJSHAHI |
| 8 | CHEEZ | DHAKA |
| 9 | PIZZA HUT | DHAKA |

## EMPLOYEE

insert into employee values(employee_id_seq.NEXTVAL,'MIR MAHIM','MIRPUR',01945679432,01845679432,40000,'MANAGER',1);

insert into employee values(employee_id_seq.NEXTVAL,'ARIFUL ISLAM','AGRABAD',01345679454,01745679442,45000,'CONTENT CREATOR',2);

insert into employee values(employee_id_seq.NEXTVAL,'DIPRA KAR','MOHAMMADPUR',01645679542,01545679489,20000,'FIELD OFFICER',3);

insert into employee values(employee_id_seq.NEXTVAL,'ARIJIT BARUA','DHANMONDI-32',01845679552,01345679467,15000,'OFFICE CLERK',4);

```
insert into employee values(employee_id_seq.NEXTVAL,'AKID
MAHMUD','MUGDA',01645679442,01845679443,50000,'ADMINISTRATION',5);
```

```
insert into employee
values(employee_id_seq.NEXTVAL,'AZMAL','DHANMONDI',01828487943,01845679543,15000,'OFFI
CE CLERK',6);
```

```
insert into employee
values(employee_id_seq.NEXTVAL,'AKMAL','DHANMONDI',01928483332,017453325465,15000,'OF
FICE CLERK',7);
```

```
insert into employee
values(employee_id_seq.NEXTVAL,'JOY','GULISTAN',01928236335,01335556465,12000,'OFFICE
CLERK',8);
```

```
insert into employee values(employee_id_seq.NEXTVAL,'FAISAL
AHMED','UTTARA',01929087943,01789679543,45000,'ADMINISTRATION',9);
```

| EMPLOYEE_ID | EMPLOYEE_NAME | EMPLOYEE_ADDRESS | EMPLOYEE_CONTACT1 | EMPLOYEE_CONTACT2 | EMPLOYEE_SALARY | ROLE | RESTAURANT_ID |
|---|---|---|---|---|---|---|---|
| 1 | MIR MAHIM | MIRPUR | 1945679432 | 1845679432 | 40000 | MANAGER | 1 |
| 2 | ARIFUL ISLAM | AGRABAD | 1345679454 | 1745679442 | 45000 | CONTENT CREATOR | 2 |
| 3 | DIPRA KAR | MOHAMMADPUR | 1645679542 | 1545679489 | 20000 | FIELD OFFICER | 3 |
| 4 | ARIJIT BARUA | DHANMONDI-32 | 1845679552 | 1345679467 | 15000 | OFFICE CLERK | 4 |
| 5 | AKID MAHMUD | MUGDA | 1645679442 | 1845679443 | 50000 | ADMINISTRATION | 5 |
| 6 | AZMAL | DHANMONDI | 1828487943 | 1845679543 | 15000 | OFFICE CLERK | 6 |
| 7 | AKMAL | DHANMONDI | 1928483332 | 17453325465 | 15000 | OFFICE CLERK | 7 |
| 8 | JOY | GULISTAN | 1928236335 | 1335556465 | 12000 | OFFICE CLERK | 8 |
| 9 | FAISAL AHMED | UTTARA | 1929087943 | 1789679543 | 45000 | ADMINISTRATION | 9 |

## PAYMENT_EMP

```
insert into payment_emp values(payment_emp_seq .NEXTVAL,1,750,'ONLINE',5,1797,'08
NOVEMBER 2022');
```

```
insert into payment_emp values(payment_emp_seq.NEXTVAL,2,150,'CASH',15,2697,'10
NOVEMBER 2022');
```

insert into payment_emp values(payment_emp_seq.NEXTVAL,3,250,'CASH',10,7697,'10 DECEMBER 2022');

insert into payment_emp values(payment_emp_seq.NEXTVAL,4,150,'ONLINE',3,1987,'15 DECEMBER 2022');

insert into payment_emp values(payment_emp_seq.NEXTVAL,5,200,'CASH',7,3097,'31 DECEMBER 2022');

| PAYMENT_ID | EMPLOYEE_ID | AMOUNT | PAYMENT_TYPE | DISCOUNT | PROMO_CODE | DATES |
|---|---|---|---|---|---|---|
| 1 | 1 | 750 | ONLINE | 5 | 1797 | 08 NOVEMBER 2022 |
| 2 | 2 | 150 | CASH | 15 | 2697 | 10 NOVEMBER 2022 |
| 3 | 3 | 250 | CASH | 10 | 7697 | 10 DECEMBER 2022 |
| 4 | 4 | 150 | ONLINE | 3 | 1987 | 15 DECEMBER 2022 |
| 5 | 5 | 200 | CASH | 7 | 3097 | 31 DECEMBER 2022 |

## DELIVERY_MAN

insert into delivery_man values(delivery_seq.NEXTVAL,'ASIF KHAN BG',01960201798,01860201798,10000,1);

insert into delivery_man values(delivery_seq.NEXTVAL,'AKASH BARMAN',01834201798,01840245798,12000,2);

insert into delivery_man values(delivery_seq.NEXTVAL,'MASHFIQUE',01760201798,01860201458,11000,3);

insert into delivery_man values(delivery_seq.NEXTVAL,'PRIOUS CHANDRA',01360201798,01560201767,14000,4);

insert into delivery_man values(delivery_seq.NEXTVAL,'WASI',01460201798,01760201797,9000,5);

| RIDER_ID | RIDER_NAME | RIDER_CONTACT1 | RIDER_CONTACT2 | RIDER_SALARY | EMPLOYEE_ID |
|---|---|---|---|---|---|
| 1 | ASIF KHAN BG | 1960201798 | 1860201798 | 10000 | 1 |
| 2 | AKASH BARMAN | 1834201798 | 1840245798 | 12000 | 2 |
| 3 | MASHFIQUE | 1760201798 | 1860201458 | 11000 | 3 |
| 4 | PRIOUS CHANDRA | 1360201798 | 1560201767 | 14000 | 4 |
| 5 | WASI | 1460201798 | 1760201797 | 9000 | 5 |

## ORDERS

insert into orders values(orders_seq.NEXTVAL,'08 NOVEMBER 2022',750,1);

insert into orders values(orders_seq.NEXTVAL,'10 NOVEMBER 2022',150,2);

insert into orders values(orders_seq.NEXTVAL,'10 DECEMBER 2022',250,3);

insert into orders values(orders_seq.NEXTVAL,'15 DECEMBER 2022',150,4);

insert into orders values(orders_seq.NEXTVAL,'31 DECEMBER 2022',200,5);

| ORDER_ID | ORDER_DATE | AMOUNT | CUSTOMER_ID |
|---|---|---|---|
| 1 | 08 NOVEMBER 2022 | 750 | 1 |
| 2 | 10 NOVEMBER 2022 | 150 | 2 |
| 3 | 10 DECEMBER 2022 | 250 | 3 |
| 4 | 15 DECEMBER 2022 | 150 | 4 |
| 5 | 31 DECEMBER 2022 | 200 | 5 |

## ORDERS_RD

```
insert into orders_rd values(orders_rd_seq.NEXTVAL,'08 NOVEMBER 2022',750,1);

insert into orders_rd values(orders_rd_seq.NEXTVAL,'10 NOVEMBER 2022',150,2);

insert into orders_rd values(orders_rd_seq.NEXTVAL,'10 DECEMBER 2022',250,3);

insert into orders_rd values(orders_rd_seq.NEXTVAL,'15 DECEMBER 2022',150,4);

insert into orders_rd values(orders_rd_seq.NEXTVAL,'31 DECEMBER 2022',200,5);
```

| ORDER_ID | ORDER_DATE | AMOUNT | RIDER_ID |
|---|---|---|---|
| 1 | 08 NOVEMBER 2022 | 750 | 1 |
| 2 | 10 NOVEMBER 2022 | 150 | 2 |
| 3 | 10 DECEMBER 2022 | 250 | 3 |
| 4 | 15 DECEMBER 2022 | 150 | 4 |
| 5 | 31 DECEMBER 2022 | 200 | 5 |

## ORDERS_RD

insert into orders_rt values(orders_rt_seq.NEXTVAL,'08 NOVEMBER 2022',750,1);

insert into orders_rt values(orders_rt_seq.NEXTVAL,'10 NOVEMBER 2022',150,2);

insert into orders_rt values(orders_rt_seq.NEXTVAL,'10 DECEMBER 2022',250,3);

insert into orders_rt values(orders_rt_seq.NEXTVAL,'15 DECEMBER 2022',150,4);

insert into orders_rt values(orders_rt_seq.NEXTVAL,'31 DECEMBER 2022',200,5);

| ORDER_ID | ORDER_DATE | AMOUNT | RESTAURANT_ID |
|---|---|---|---|
| 1 | 08 NOVEMBER 2022 | 750 | 1 |
| 2 | 10 NOVEMBER 2022 | 150 | 2 |
| 3 | 10 DECEMBER 2022 | 250 | 3 |
| 4 | 15 DECEMBER 2022 | 150 | 4 |
| 5 | 31 DECEMBER 2022 | 200 | 5 |

# QUERY WRITING

Single Row Function:

1) **Write a query to display the employee names in lowercase letters.**

select lower(employee_name) from employee;

| LOWER(EMPLOYEE_NAME) |
|---|
| mir mahim |
| ariful islam |
| dipra kar |
| arijit barua |
| akid mahmud |
| azmal |
| akmal |
| joy |
| faisal ahmed |

2) Display the Food name and food_status joining the columns using concatenation function as food details.

   select concat(food_name,food_status) AS Food_Details from menu;

| FOOD_DETAILS |
| --- |
| KACCHI BIRYANIAVAILABLE |
| MUROG PULAOUN-AVAILABLE |
| BURGERAVAILABLE |
| FALUDAAVAILABLE |
| CHOCOLATE WAFFLEUN-AVAILABLE |

Group row Function:

1) Find the average, minimum and maximum salary of the employees. Label the columns AVG, MIN and MAX respectively, according to role.

select role, avg(employee_salary) as "avg",min(employee_salary) as "min",max(employee_salary) as "max" from employee group by role;

| ROLE | Avg | Min | Max |
| --- | --- | --- | --- |
| OFFICE CLERK | 14250 | 12000 | 15000 |
| ADMINISTRATION | 47500 | 45000 | 50000 |
| FIELD OFFICER | 20000 | 20000 | 20000 |
| MANAGER | 40000 | 40000 | 40000 |
| CONTENT CREATOR | 45000 | 45000 | 45000 |

2) Display the number of employees whose role is Office Clerk.

select count(*) from employee where role='OFFICE CLERK';

| COUNT(*) |
| --- |
| 4 |

Sub-Query:

1) Display the names of food whose price is more than Faluda.

select food_name from menu where food_price > (select food_price FROM menu where food_name = 'FALUDA');

| FOOD_NAME |
| --- |
| KACCHI BIRYANI |
| BURGER |
| CHOCOLATE WAFFLE |

2) Display the employee names who are working as manager live in Mirpur Location.

select employee_name FROM employee WHERE role ='Manager' AND employee_address = (SELECT employee_address FROM employee WHERE role = 'MANAGER' AND employee_address = 'MIRPUR');

| EMPLOYEE_NAME |
| --- |
| MIR MAHIM |

Joining:

NON-EQUIJOIN:

Display restaurant_name, resatuarant_id, food_name, food_price from restaurant and menu Table.

select restaurant.restaurant_id, restaurant.restaurant_name, menu.food_name, menu.food_price from restaurant,menu;

| RESTAURANT_ID | RESTAURANT_NAME | FOOD_NAME | FOOD_PRICE |
|---|---|---|---|
| 1 | BUKHARA | KACCHI BIRYANI | 750 |
| 2 | DINE-IN | KACCHI BIRYANI | 750 |
| 3 | BABY-LON | KACCHI BIRYANI | 750 |
| 4 | CHILLOX | KACCHI BIRYANI | 750 |
| 5 | DAWAT-E-DARBAR | KACCHI BIRYANI | 750 |
| 6 | PIZZERIA | KACCHI BIRYANI | 750 |
| 7 | DOMINOES | KACCHI BIRYANI | 750 |
| 8 | CHEEZ | KACCHI BIRYANI | 750 |
| 9 | PIZZA HUT | KACCHI BIRYANI | 750 |
| 1 | BUKHARA | MUROG PULAO | 150 |
| 2 | DINE-IN | MUROG PULAO | 150 |
| 3 | BABY-LON | MUROG PULAO | 150 |
| 4 | CHILLOX | MUROG PULAO | 150 |
| 5 | DAWAT-E-DARBAR | MUROG PULAO | 150 |
| 6 | PIZZERIA | MUROG PULAO | 150 |
| More than 15 rows available. Increase rows selector to view more rows. | | | |

EUIJOIN:

2) Get all the INFORMATION from restaurant and employee where restuarant_id and employee_id is same.

select restaurant.*, employee.* from restaurant, employee where restaurant.restaurant_id = employee.employee_id;

| RESTAURANT_ID | RESTAURANT_NAME | RESTAURANT_ADDRESS | EMPLOYEE_ID | EMPLOYEE_NAME | EMPLOYEE_ADDRESS | EMPLOYEE_CONTACT1 | EMPLOYEE_CONTACT2 | EMPLOYEE_SALARY | ROLE | RESTAURANT_ID |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | BUKHARA | DHAKA | 1 | MIR MAHIM | MIRPUR | 1945679432 | 1845679432 | 40000 | MANAGER | 1 |
| 2 | DINE-IN | JESSORE | 2 | ARIFUL ISLAM | AGRABAD | 1345679454 | 1745679442 | 45000 | CONTENT CREATOR | 2 |
| 3 | BABY-LON | KHULNA | 3 | DIPRA KAR | MOHAMMADPUR | 1645679542 | 1545679489 | 20000 | FIELD OFFICER | 3 |
| 4 | CHILLOX | CHANDPUR | 4 | ARIJIT BARUA | DHANMONDI-32 | 1845679552 | 1345679467 | 15000 | OFFICE CLERK | 4 |
| 5 | DAWAT-E-DARBAR | PABNA | 5 | AKID MAHMUD | MUGDA | 1645679442 | 1845679443 | 50000 | ADMINISTRATION | 5 |
| 6 | PIZZERIA | DHAKA | 6 | AZMAL | DHANMONDI | 1828487943 | 1845679543 | 15000 | OFFICE CLERK | 6 |
| 7 | DOMINOES | RAJSHAHI | 7 | AKMAL | DHANMONDI | 1928483332 | 17453325465 | 15000 | OFFICE CLERK | 7 |
| 8 | CHEEZ | DHAKA | 8 | JOY | GULISTAN | 1928236335 | 1335556465 | 12000 | OFFICE CLERK | 8 |
| 9 | PIZZA HUT | DHAKA | 9 | FAISAL AHMED | UTTARA | 1929087943 | 1789679543 | 45000 | ADMINISTRATION | 9 |

View:

1) Display the contents of the cutomerdetailsview view.

select * from cutomerdetailsview;

| CUSTOMER_ID | CUSTOMER_NAME | CUSTOMER_CONTACT1 |
|---|---|---|
| 1 | RAKIB | 1978789812 |
| 2 | TURABA | 1778789813 |
| 3 | DRISHTY | 1678789814 |
| 4 | ALIF | 1778789815 |
| 5 | EMON | 1878789816 |

2) Write a query to display all customer_name where customer_id is 2 from cutomerdetailsview

select customer_name from cutomerdetailsview where customer_id=2;

| CUSTOMER_NAME |
|---|
| TURABA |

# RELATIONAL ALGEBRA

- Select only those rows of data where employee_address is Dhanmondi from the employee Table.

$\sigma_{employee\_address = \text{“Dhanmondi”}}$ (employee)

- Project the order_id and the amount from the orders_rd table.

$\prod_{order\_id,\ amount}$(order_rd)

Find the name of the customer whose rating_id is 4.

$\prod_{customer\_name}$ ($\sigma_{rating\_id=4}$(customer))

- Find the amount from payment table where amount is greater than 250.

$\prod_{amount}$ ($\sigma_{amount > 250}$ (payment))

- Find the name of the food where food_status is AVAILABLE.

$\prod_{food\_name}$ ($\sigma_{food\_status = \text{“Available”}}$ (menu))

# CONCLUSION

We have shown all the queries to create the tables in 'Oracle 10g'. Also, we had shown the queries to insert the values and took their screen-shots. Here, we made 13 different relations among the entities. The normalization process has made our way easier. In future a lot of features can be added for example: Food Donation, Offer Campaign, Grocery Feature etc.

The End