Student Name: **A Z Hasnain Kabir**
Student ID: **200042102**

CSE 4410
Database Management Systems II
Lab 1

Introduction

In this lab, we explored various SQL queries on a database containing tables for franchise locations, customers, orders, and menu items. The following are the tasks and queries that were completed

Task 1 Creating Tables

The first task was to create the following tables in the database:

- franchise: stores information about franchise locations, including the ID, name, address, and city.
- customer: stores information about customers, including the ID, name, phone number, and preferred cuisines.
- registers: stores information about which customers are registered at which franchise locations.
- branch: stores information about branches of franchises, including the ID, branch name, and franchise ID.
- chef: stores information about chefs, including the ID, name, and branch ID.
- develops: stores information about which chefs develop which menu items.
- menu: stores information about menu items, including the ID, name, ingredients, price, and calorie count.
- order: stores information about orders, including the ID, customer ID, menu ID, and rating.

Code

```
1
2  Create table franchise(
3      ID int primary key,
4      Name varchar2(250),
5      Address varchar2(250),
6      city varchar2(250)
7  );
8
```

```
Create type preferred_cuisines as varray(5) of varchar2(20);
/

Create table customer(
    ID int primary key,
    Name varchar2(250),
    phone varchar2(12),
    cuisines preferred_cuisines
);

Create table registers(
    customerId int ,
    franchiseId int,
    foreign key (customerId) references customer(customerId),
    foreign key (franchiseId) references franchise(franchiseId)
);

Create table branch(
    Id int primary key,
    branchName varchar2(250),
    franchiseId varchar2(250),
    foreign key(franchiseId) references franchise(franchiseId)
);

Create table chef(
    Id int primary key,
    name varchar2(250),
    branchId int,
    foreign key(branchId) references branch(branchId)
);

Create table develops(
    chefId int,
    cuisineName varchar2(250),
    menuId int,

    foreign key (chefId) references chef(Id),
    foreign key (menuId) references menu(Id)
);

Create table menu(
    Id int primary key,
    name varchar2(250),
    ingredients varchar2(250),
    price int,
    calorie_count int
);

Create table order(
    Id int primary key,
    customerId int,
    menuId int,
    rating int,

```

```
    foreign key(customerId) references customer(customerId),
    foreign key(menuId) references menu(menuId)
);
```

Difficulties

There are different patterns of thinking while constructing a set of tables from a given problem statement. The number of tables may vary depending on the thought patterns. Also, after properly understanding the queries, the schema structure may change as well.

Task 2 Queries

We are to generate results based on the given queries:

a) Number of customers per franchise

```
    select count(customer.Id), a.franchise.name from customer join
(select * from registers join franchise on franchise.Id = registers.franchiseId) as a
on customer.Id = a.customerId group by franchise.name;
```

b) Average rating for each menu item among all franchises

```
select avg(rating), franchise.name from order, menu, develops, chef, branch, franchise
where order.menuId = menu.Id and menu.Id = develops.menuId
and develops.chefId = chef.Id and chef.branchId = branch.Id
and branch.franchiseId = franchise.Id group by franchise.name;
```

c) The 5 top most popular items.

```
select count(order.Id) as orders, menu.name
from order, menu where menu.Id = order.menuId
group by menu.name order by orders desc having rownum<=5;
```

d) The names of all customers who have preferred food that is offered from at least 2 different franchises

```
select customer.name
from customer join registers on customer.Id = registers.customerId
join franchise on registers.franchiseId = franchise.Id
join develops on franchise.Id = develops.franchiseId
join menu on develops.menuId = menu.Id
where develops.cuisineName in customer.preferred_cuisines
group by customer.name
having count(distinct franchise.Id) >= 2;
```

e) The names of all customers who have not placed any orders.

```
select name from customer
where customer.Id not in (select customerId from `order`);
```

Difficulties

It was difficult to brainstorm and generate all these queries in a short time during the lab as most of my time was spent trying to come up with a good database schema.