



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
```

```

9  Create type preferred_cuisines as varray(5) of varchar2(20);
10 /
11
12 Create table customer(
13     ID int primary key,
14     Name varchar2(250),
15     phone varchar2(12),
16     cuisines preferred_cuisines
17 );
18
19 Create table registers(
20     customerId int ,
21     franchiseId int,
22     foreign key (customerId) references customer(customerId),
23     foreign key (franchiseId) references franchise(franchiseId)
24 );
25
26 Create table branch(
27     Id int primary key,
28     branchName varchar2(250),
29     franchiseId varchar2(250),
30     foreign key(franchiseId) references franchise(franchiseId)
31 );
32
33 Create table chef(
34     Id int primary key,
35     name varchar2(250),
36     branchId int,
37     foreign key(branchId) references branch(branchId)
38 );
39
40 Create table develops(
41     chefId int,
42     cuisineName varchar2(250),
43     menuId int,
44
45     foreign key (chefId) references chef(Id),
46     foreign key (menuId) references menu(Id)
47 );
48
49 Create table menu(
50     Id int primary key,
51     name varchar2(250),
52     ingredients varchar2(250),
53     price int,
54     calorie_count int
55 );
56
57 Create table order(
58     Id int primary key,
59     customerId int,
60     menuId int,
61     rating int,
62

```

```
63     foreign key(customerId) references customer(customerId),
64     foreign key(menuId) references menu(menuId)
65 );
```

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

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

b) Average rating for each menu item among all franchises

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

c) The 5 top most popular items.

```
1 select count(order.Id) as orders, menu.name
2 from order, menu where menu.Id = order.menuId
3 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

```
1 select customer.name
2 from customer join registers on customer.Id = registers.customerId
3 join franchise on registers.franchiseId = franchise.Id
4 join develops on franchise.Id = develops.franchiseId
5 join menu on develops.menuId = menu.Id
6 join order on order.menuId = menu.Id
7 group by customer.name
8 having count(distinct franchise.Id) >= 2;
```

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

```
1 select name from customer
2 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.