LAB REPORT

CSE 4410

DATABASE MANAGEMENT SYSTEMS II LAB

LAB_01: Relational Model

NAME: CHOWDHURY ASHFAQ

STUDENT ID: 200042123

PROGRAM: SWE

GROUP: 1A

DATE: 13/01/23

Tasks:

1. Convert the scenario into DDL using standard SQL denoting the appropriate constraints.
2. Write SQL statements for the following queries:
   (a) Find the total number of customers for each franchise.
   (b) Find the avg rating for each menu item among all franchises.
   (c) Find the 5 top most popular items. It should be based on the number of times they were ordered.
   (d) Find the names of all customers who have preferred food that is offered from at least 2 different franchises.
   (e) Find the names of all customers who have not placed any orders.

Analysis of the problem:

A scenario was given from where we needed to identify the entities to build an ERd diagram. The main reason to build the ERd diagram was to make the work of writing DDL statements for the given scenario easier. With the ERd diagram we can identify the numbers of table required and write the code easily. Next we had to perform some queries too to extract information.

Solution:

```
CREATE TABLE FRANCHISE(
    franchise_name varchar2(50),
    CONSTRAINT pk_franchise PRIMARY KEY(franchise_name)
);



CREATE TABLE CUSTOMER(
    customer_id number,
    customer_name varchar2(50),
    CONSTRAINT pk_customer PRIMARY KEY(customer_id)
);

CREATE TABLE BRANCH(
    branch_id number,
    branch_name varchar2(50),
    franchise_name varchar2(50),
    CONSTRAINT pk_branch PRIMARY KEY(branch_id),
    CONSTRAINT fk_branch_franchise FOREIGN KEY(franchise_name) REFERENCES FRANCHISE(franchise_name)
);


CREATE TABLE CHEF(
    chef_id number,
    branch_id number,
    chef_name varchar2(50),
    menu_id number,
    CONSTRAINT pk_chef PRIMARY KEY(chef_id),
    CONSTRAINT fk_chef_branch FOREIGN KEY(branch_id) REFERENCES BRANCH(branch_id),
    CONSTRAINT fk_chef_menu FOREIGN KEY(menu_id) REFERENCES MENU(menu_id),
);
```

```
CREATE TABLE MENU(
    menu_id number,
    menu_name varchar2(50),
    cuisine_id number,
    main_ingredients varchar2(100),
    price number,
    calorie_count,
    CONSTRAINT pk_menu PRIMARY KEY(menu_id),
    CONSTRAINT fk_menu_cuisine FOREIGN KEY(cuisine_id) REFERENCES CUISINE(cuisine_id)
);


CREATE TABLE CUISINE(
    cuisine_id number,
    cuisine_name varchar2(50),
    CONSTRAINT pk_cuisine PRIMARY KEY(cuisine_id)
);


CREATE TABLE PREF_CUISINE(
    customer_id number,
    cuisine_id number,,
    CONSTRAINT pk_pref_cuisine PRIMARY KEY(customer_id,cuisine_id)
);


CREATE TABLE RATING(
    rating_id number,
    customer_id number,
    cuisine_id number,
    rating number,
    CONSTRAINT pk_rating PRIMARY KEY(rating_id),
    CONSTRAINT fk_rating_customer FOREIGN KEY(customer_id) REFERENCES CUSTOMER(customer_id),
    CONSTRAINT fk_rating_cuisine FOREIGN KEY(cuisine_id) REFERENCES CUISINE(cuisine_id)
);
```

```
CREATE TABLE FRANCHISE_CUSTOMER(
    customer_id number,
    franchise_id number,
    CONSTRAINT pk_franchise_customer PRIMARY KEY(customer_id,menu_id),
     CONSTRAINT fk_franchise_customer_customer FOREIGN KEY(customer_id) REFERENCES CUSTOMER(customer_id),
      CONSTRAINT fk_franchise_customer_franchise FOREIGN KEY(franchise_id) REFERENCES FRANCHISE(franchise_id)
);


CREATE TABLE CHEF_MENU(
    chef_id number,
    menu_id number,
    CONSTRAINT pk_chef_menu PRIMARY KEY(chef_id,menu_id),
     CONSTRAINT fk_chef_menu_chef FOREIGN KEY(chef_id) REFERENCES CHEF(chef_id),
      CONSTRAINT fk_chef_menu_menu FOREIGN KEY(menu_id) REFERENCES MENU(menu_id)
);


CREATE TABLE FRANCHISE_MENU(
    franchise_id number,
    menu_id number,
    CONSTRAINT pk_franchise_menu PRIMARY KEY(franchise_id,menu_id),
     CONSTRAINT fk_franchise_menu_franchise FOREIGN KEY(franchise_id) REFERENCES FRANCHISE(franchise_id),
      CONSTRAINT fk_franchise_menu_menu FOREIGN KEY(menu_id) REFERENCES MENU(menu_id)
);


CREATE TABLE ORDER(
    order_id number,
    customer_id number,
    cuisine_id number,
    CONSTRAINT pk_order PRIMARY KEY(order_id),
    CONSTRAINT fk_customer_order FOREIGN KEY(customer_id) REFERENCES CUSTOMER(customer_id),
    CONSTRAINT fk_order_cuisine FOREIGN KEY(cuisine_id) REFERENCES CUISINE(cuisine_id)
);
```

```
select count(*), franchise_name from FRANCHISE
GROUP BY franchise_name;


select avg(rating), cuisine_name
from RATING, CUISINE
WHERE RATING.cuisine_id=CUISINE.cuisine_id
GROUP BY cuisine_name;
```

Explanation:

At first after reading the whole scenario I identified the entities at first and simultaneously the attributes of the entities where identified. Then I identified the relationships between the entities. As we know for many to many relationship we need a junction table so we identified the junction tables as well.

After the above process was completed we got twelve tables in total. The attributes, primary keys, and foreign keys of each of the entities where identified beforehand. So we just created the tables with the DDL statements with the help of the entity relationship diagram.

Once the tables where created I wrote some queries according to the necessity provided in the task.