Husky Restaurant Review Website

MISM6213: Group 13

Phase 1: Project Proposal

Problem Statement:

The Husky Restaurant Review website aims to provide restaurants feedback on how they can improve their service, and help users discover restaurants they might be interested in. On their website, companies will be able to list relevant information about their restaurants, and users will be able to leave ratings and reviews about their experience. Users will also be able to search for restaurants, recommend restaurants to others, and make reservations through the website.

In this business case, data is deeply interconnected and cannot be maintained using traditional file systems. Building a database will be essential to store and manage this data in order to ensure data consistency across the website and provide quick execution of search queries for users. It will also eliminate data redundancy and ensure that information can be quickly updated in real-time.

A well-designed database will make it possible for users to easily discover restaurants based on their preferences, read reviews, make reservations and recommend them to others. Restaurant owners will also be able to read through reviews and identify areas for improvement with ease.

Functionality:

The Restaurant Review Database will maintain relevant information about Restaurants including address, cuisine, timings, menu, etc. It will also record which company owns the restaurant and what kind of dining service the restaurant provides (i.e. sit down, takeout, food truck). The database will also store information about users, allowing them to make reservations, write reviews and make recommendations to other users of the website. Records about reservations, reviews and recommendations will be stored in separate tables. The database will also maintain credit card information that a user can optionally add.

Phase 2: Conceptual Design

Entities:

1. User : Identified by unique ‘Username’
2. Credit Card : Primary key ‘Card ID’
3. Review : Primary key ‘Review ID’
4. Recommendations : Primary key ‘Recommend ID’
5. Reservations : Primary key ‘Reservation ID’
6. Company : Primary key ‘Company ID’
7. Restaurants : Primary key ‘Restaurant ID’.

   Subtypes include:

   i. Sit Down
   ii. Takeout
   iii. Food Truck

Relationships between entities:

Below are the relationships between entities that can be inferred from the business rules:

User ⇄ Credit Card
User ⇄ Reviews
User ⇄ Recommendations
User ⇄ Reservations
Restaurants ⇄ Reviews
Restaurants ⇄ Recommendations
Sit Down Restaurants ⇄ Reservations
Restaurants ⇄ Company

Cardinalities of relationships among entities:

User (mandatory one) ⇄ Credit Card (optional many)
User (mandatory one) ⇄ Reviews (optional many)
User (mandatory one) ⇄ Recommendations (optional many)
User (mandatory one) ⇄ Reservations (optional many)
Restaurants (mandatory one) ⇄ Reviews (optional many)
Restaurants (mandatory one) ⇄ Recommendations (optional many)
Sit Down (mandatory one) ⇄ Reservations (optional many)
Restaurants (mandatory many) ⇄ Company (mandatory one)

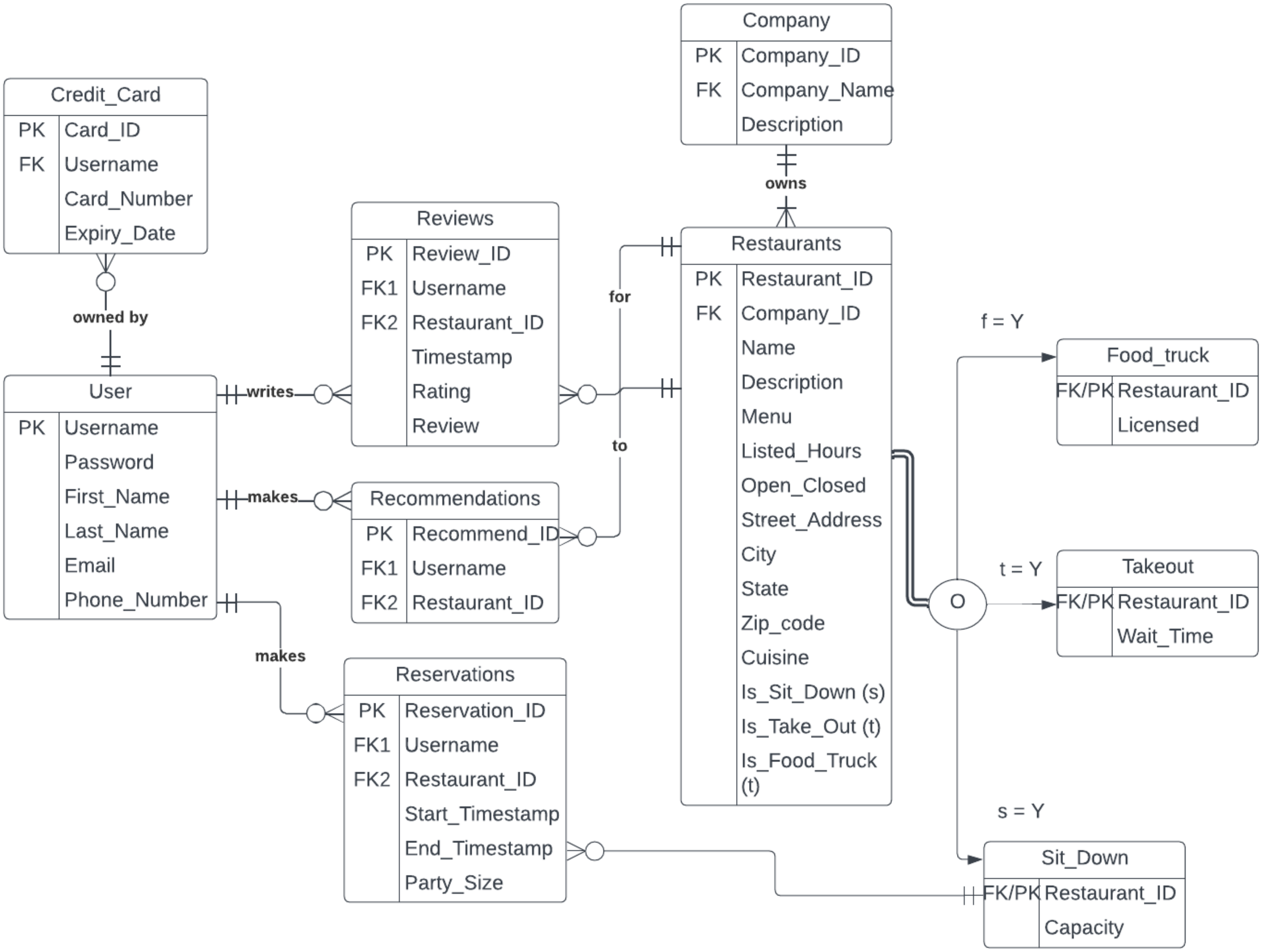
Attributes of each entity:

Entity	Attributes
User	Username, Password, First_Name, Last_Name, Email, Phone_Number
Credit_Card	Card_ID, Card_Number, Expiry Date
Reviews	Review_ID, Timestamp, Rating, Review
Recommendations	Recommend_ID
Reservations	Reservation_ID, Start_Timestamp, End_Timestamp, Party_Size
Restaurants*	Restaurant_ID, Name, Description, Menu, Listed_Hours, Open_Close, Street_Address, City, State, Zipcode, Cuisine, Is_Sit_Down,, Is_Take_Out, Is_Food_Truck
Company	Company_ID, Company_Name, Description

*The Restaurants is fully specialized into three overlapping subtypes as following:

Entity	Attribute
Sit_Down	Capacity
Takeout	Wait_Time
Food_Truck	Licensed

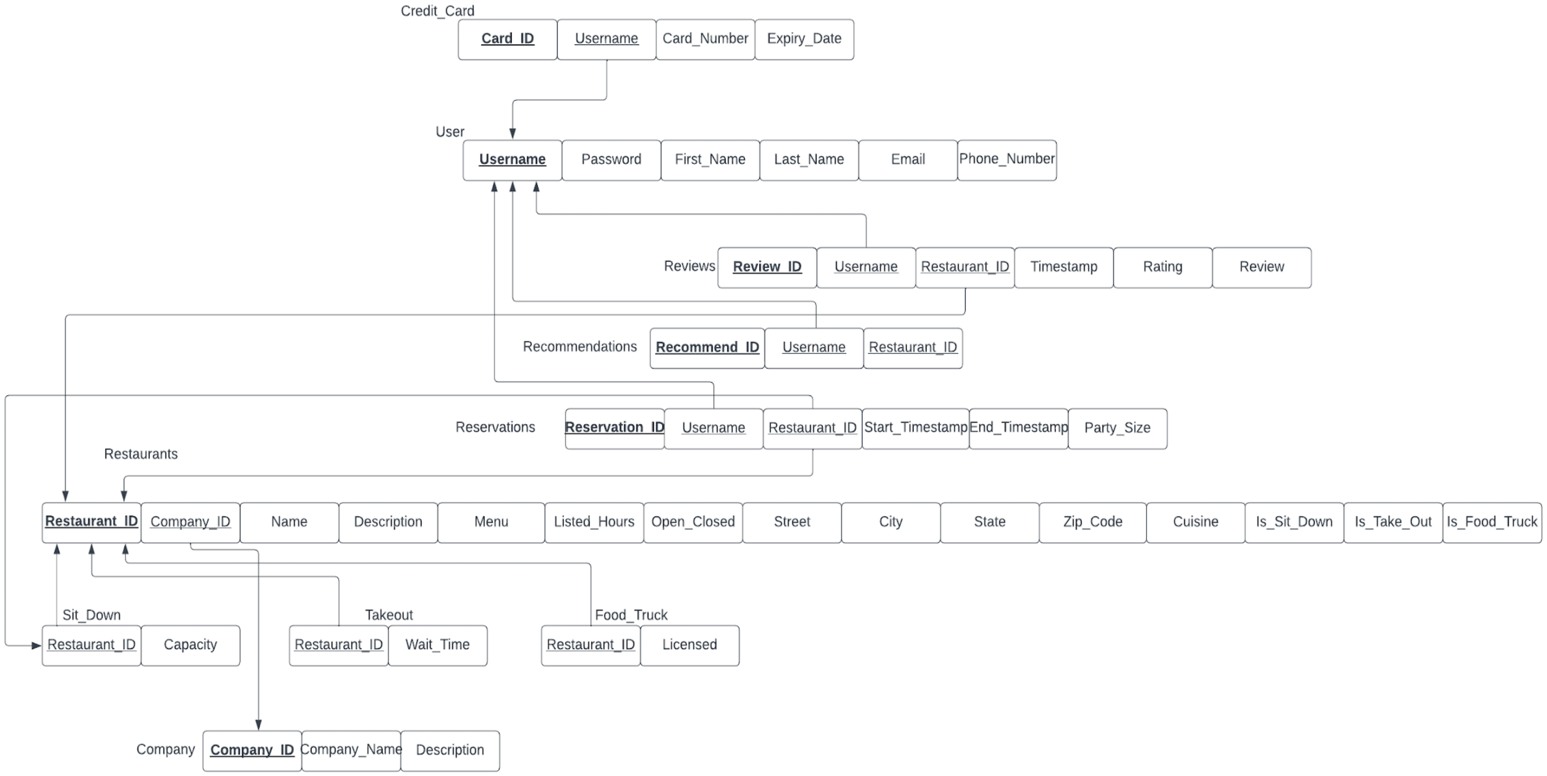
E-R Diagram:


Company
PK Company_ID
FK Company_Name
Description
owns
Credit_Card
PK Card_ID
FK Username
Card_Number
Expiry_Date
owned by
Reviews
PK Review_ID
FK1 Username
FK2 Restaurant_ID
Timestamp
Rating
Review
for
Restaurants
PK Restaurant_ID
FK Company_ID
Name
Description
Menu
Listed_Hours
Open_Closed
Street_Address
City
State
Zip_code
Cuisine
Is_Sit_Down (s)
Is_Take_Out (t)
Is_Food_Truck (t)
f = Y
Food_truck
FK/PK Restaurant_ID
Licensed
User
PK Username
Password
First_Name
Last_Name
Email
Phone_Number
writes
makes
to
Recommendations
PK Recommend_ID
FK1 Username
FK2 Restaurant_ID
O
t = Y
Takeout
FK/PK Restaurant_ID
Wait_Time
makes
Reservations
PK Reservation_ID
FK1 Username
FK2 Restaurant_ID
Start_Timestamp
End_Timestamp
Party_Size
s = Y
Sit_Down
FK/PK Restaurant_ID
Capacity

Phase 3: Logical Design

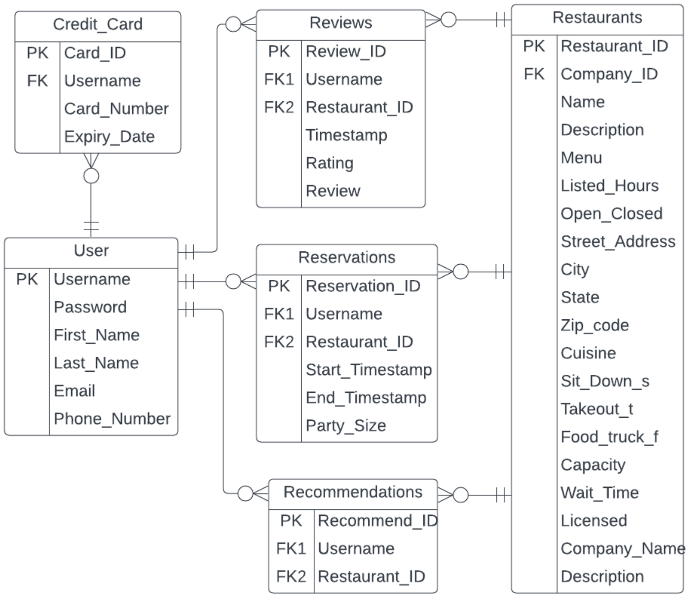
ER Diagram to Relational Schema:

At this step into the process, we transform the conceptual schema for the website database into a relational data model.

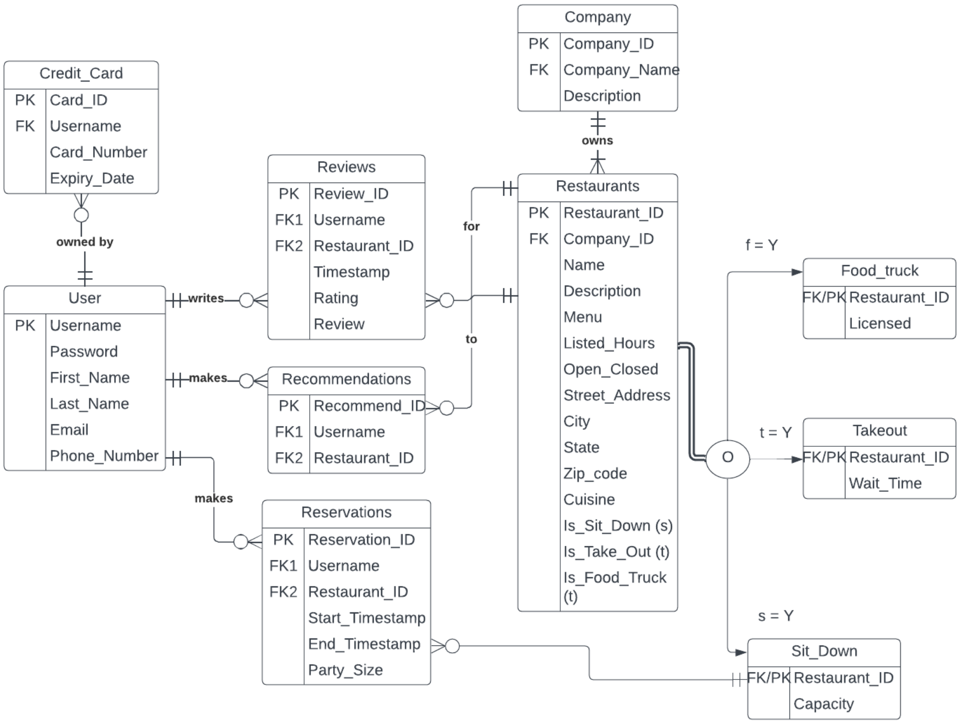


Data Normalization:

1NF & 2NF:
The schema at this stage has no repeating attributes, and also doesn't have any partial dependencies, so the data in 1NF and 2NF are identical.



3NF:
The schema at this stage has no transitive dependencies.



Summary table for each entity:

Entity : Users
Primary Key: Username
Details: All fields are required, email must be unique.

Attribute	**Datatype**
Username	VARCHAR(255)
Password	VARCHAR(255)
First_Name	VARCHAR(255)
Last_Name	VARCHAR(255)
Email	VARCHAR(255)
Phone_Number	VARCHAR(12)

Entity : Company
Primary Key : Company_ID
Details : All fields required

Attribute	**Datatype**
Company_ID	CHAR(8)
Company_Name	VARCHAR(255)
Description	VARCHAR(255)

Entity : Credit_Card
Primary Key : Card_ID
Foreign Key : Username.
Details : All fields are required. Cascading referential integrity constraint applied for foreign key ‘Username’.

Attribute	**Datatype**
Card_ID	CHAR(8)
Username	VARCHAR(255)
Card_Number	CHAR(16)
Expiry_Date	DATE

Entity : Restaurants
Primary Key : Restaurant_ID
Foreign Key : Company_ID
Details : All fields are required.
Cascading referential integrity constraint applied for foreign key ‘Company_ID’.
Check constraint applied to ‘Is_Sit_Down’, ‘Is_Take_Out’ and ‘Is_Food_Truck’ for boolean values of 1 and 0.
Check constraints also applied to ‘Open_Closed’ and ‘Cuisine’ to standardize responses.

Attribute	**Datatype**
Restaurant_ID	CHAR(8)
Company_ID	CHAR(8)
Name	VARCHAR(255)
Description	VARCHAR(255)
Menu	VARCHAR(255)
Listed_Hours	VARCHAR(255)
Open_Closed	VARCHAR(6)
Street_Address	VARCHAR(255)
City	VARCHAR(255)
State	VARCHAR(255)
Zip_Code	CHAR(5)
Cuisine	VARCHAR(255)
Is_Sit_Down	CHAR(1)
Is_Take_Out	CHAR(1)
Is_Food_Truck	CHAR(1)

Entity : Sit_Down

Primary Key & Foreign Key : Restaurant_ID

Details : All fields are required.

Attribute	**Datatype**
Restaurant_ID	CHAR(8)
Capacity	INT

Entity : Food_Truck

Primary Key & Foreign Key : Restaurant_ID

Details : All fields are required.

Check constraint applied to 'Licensed' attribute to limit responses to 'YES' and 'NO'.

Attribute	**Datatype**
Restaurant_ID	CHAR(8)
Licensed	VARCHAR(3)

Entity : Reviews

Primary Key : Review_ID

Foreign Key1: Username

Foreign Key2: Restaurant_ID

Details : All fields are required.

Check constraint on 'Rating' attribute to restrict values from 1 to 5.

Attribute	**Datatype**
Review_ID	CHAR(8)
Username	VARCHAR(255)
Restaurant_ID	CHAR(8)
Timestamp	TIMESTAMP
Review	VARCHAR(1000)
Rating	INT

Entity : Takeout

Primary Key & Foreign Key : Restaurant_ID

Details : All fields are required.

Attributes	**Datatype**
Restaurant_ID	CHAR(8)
Wait_Time	VARCHAR(10)

Entity : Recommendations

Primary Key : Recommend_ID

Foreign Key1: Username

Foreign Key2: Restaurant_ID

Details : All fields are required.

Attribute	**Datatype**
Recommend_ID	CHAR(8)
Username	CHAR(255)
Restaurant_ID	CHAR(8)

Entity : Reservations

Primary Key : Reservation_ID

Foreign Key1: Username

Foreign Key2: Restaurant_ID (from specialized entity 'Sit_Down')

Details : All fields are required.

Check constraint to ensure Start_Timestamp is smaller than End_Timestamp.

Attribute	**Datatype**
Reservation_ID	CHAR(8)
Username	CHAR(255)
Restaurant_ID	CHAR(8)
Start_Timestamp	TIMESTAMP
End_Timestamp	TIMESTAMP
Party_Size	INT

Phase 4: Physical Design

Creation of Tables:

```
CREATE TABLE Users (
    Username VARCHAR(255) NOT NULL,
    Password VARCHAR(255) NOT NULL,
    First_Name VARCHAR(255) NOT NULL,
    Last_Name VARCHAR(255) NOT NULL,
    Email VARCHAR(255) UNIQUE NOT NULL,
    Phone_Number VARCHAR(12) NOT NULL,
    CONSTRAINT Users_PK PRIMARY
KEY(username)
);
```

```
CREATE TABLE Credit_Card (
    Card_ID CHAR(8) NOT NULL,
    Username VARCHAR(255) NOT NULL,
    Card_Number CHAR(16) NOT NULL,
    Expiry_Date DATE NOT NULL,
    CONSTRAINT CreditCard_PK PRIMARY
KEY(Card_ID),
    CONSTRAINT CreditCard_FK FOREIGN
KEY(Username)
    REFERENCES Users(Username) ON DELETE
CASCADE
);
```

```
CREATE TABLE Company (
    Company_ID CHAR(8) NOT NULL,
    Company_Name VARCHAR(255) NOT NULL,
    Description VARCHAR(255) NOT NULL,
    CONSTRAINT Company_PK PRIMARY
KEY(Company_ID)
);
```

```
CREATE TABLE Restaurants (
    Restaurant_ID CHAR(8) NOT NULL,
    Company_ID CHAR(8) NOT NULL,
    Name VARCHAR(255) NOT NULL,
    Description VARCHAR(255) NOT NULL,
    Menu VARCHAR(255) NOT NULL,
    Listed_Hours VARCHAR(255),
    Open_Closed VARCHAR(6) NOT NULL,
    Street_Address VARCHAR(255) NOT NULL,
    City VARCHAR(255) NOT NULL,
    State VARCHAR(255) NOT NULL,
    Zip_Code CHAR(5) NOT NULL,
    Cuisine VARCHAR(255) NOT NULL,
    Is_Sit_Down CHAR(1) CHECK (Is_Sit_Down
IN('1','0')) NOT NULL,
    Is_Take_Out CHAR(1) CHECK (Is_Take_Out
IN('1', '0')) NOT NULL,
    Is_Food_Truck CHAR(1)CHECK
(Is_Food_Truck IN('1', '0')) NOT NULL,
    CONSTRAINT Restaurants_PK PRIMARY
KEY(Restaurant_ID),
    CONSTRAINT Restaurants_FK FOREIGN
KEY(Company_ID)
    REFERENCES Company(Company_ID) ON DELETE
CASCADE,
    CONSTRAINT Open_Closed_check CHECK
(Open_Closed IN ('Open', 'Closed')),
    CONSTRAINT Cuisine_check CHECK (Cuisine
IN ('African', 'American', 'Asian',
'European', 'Hispanic'))
);
```

```
CREATE TABLE Food_Truck (
    Restaurant_ID CHAR(8) NOT NULL,
    Licensed VARCHAR(3) CHECK (Licensed
IN ('YES', 'NO')) NOT NULL,
    CONSTRAINT Truck_PK PRIMARY KEY
(Restaurant_ID),
    CONSTRAINT Truck_FK FOREIGN KEY
(Restaurant_ID)
    REFERENCES Restaurants(Restaurant_ID)
);
```

```
CREATE TABLE Sit_Down (
    Restaurant_ID CHAR(8) NOT NULL,
    Capacity INT NOT NULL,
    CONSTRAINT Sit_Down_PK PRIMARY KEY
(Restaurant_ID),
    CONSTRAINT Sit_Down_FK FOREIGN KEY
(Restaurant_ID) REFERENCES
Restaurants(Restaurant_ID)
);
```

```
CREATE TABLE Takeout (
    Restaurant_ID CHAR(8) NOT NULL,
    Wait_Time VARCHAR(10) NOT NULL,
    CONSTRAINT Takeout_PK PRIMARY KEY
(Restaurant_ID),
    CONSTRAINT Takeout_FK FOREIGN KEY
(Restaurant_ID)
    REFERENCES Restaurants(Restaurant_ID)
);
```

```
CREATE TABLE Reviews (
    Review_ID CHAR(8) NOT NULL,
    Username VARCHAR(255) NOT NULL,
    Restaurant_ID CHAR(8) NOT NULL,
    Timestamp TIMESTAMP NOT NULL,
    Review VARCHAR(1000) NOT NULL,
    Rating INT NOT NULL,
    CONSTRAINT Rating_Check CHECK (Rating >=
1 AND Rating <= 5),
    CONSTRAINT Review_PK PRIMARY KEY
(Review_ID),
    CONSTRAINT Review_FK1 FOREIGN KEY
(Username) REFERENCES Users(Username),
    CONSTRAINT Review_FK2 FOREIGN KEY
(Restaurant_ID) REFERENCES
Restaurants(Restaurant_ID)
);
```

```
CREATE TABLE Recommendations (
    Recommend_ID CHAR(8) NOT NULL,
    Username VARCHAR(255) NOT NULL,
    Restaurant_ID CHAR(8) NOT NULL,
    CONSTRAINT Recommend_PK PRIMARY KEY
(Recommend_ID),
    CONSTRAINT Recommend_FK1 FOREIGN KEY
(Username) REFERENCES Users(Username),
    CONSTRAINT Recommend_FK2 FOREIGN KEY
(Restaurant_ID) REFERENCES
Restaurants(Restaurant_ID)
);
```

```
CREATE TABLE Reservations (
    Reservation_ID CHAR(8) NOT NULL,
    Username VARCHAR(255) NOT NULL,
    Restaurant_ID CHAR(8) NOT NULL,
    Start_Timestamp TIMESTAMP NOT NULL,
    End_Timestamp TIMESTAMP NOT NULL,
    Party_Size INT NOT NULL,
    CONSTRAINT Reservations_PK PRIMARY KEY
(Reservation_ID),
    CONSTRAINT Reservation_FK1 FOREIGN KEY
(Username) REFERENCES Users(Username),
    CONSTRAINT Reservation_FK2 FOREIGN KEY
(Restaurant_ID) REFERENCES
Sit_Down(Restaurant_ID),
    CONSTRAINT Reservation_check CHECK
(Start_Timestamp < End_Timestamp)
);
```

Insertion of Data in Tables:

USERS TABLE:

```
INSERT INTO Users (username, password, first_name, last_name, email, phone_number)
VALUES('bill_nye', 'password1', 'Bill', 'Nye', 'bill_nye@mail.com', '111-111-1111');
INSERT INTO Users (username, password, first_name, last_name, email, phone_number)
VALUES('jill_biden', 'passwyrd4', 'Jill', 'Biden', 'jill_bi@email.com',
'222-333-1234');
INSERT INTO Users (username, password, first_name, last_name, email, phone_number)
VALUES('charlie', '1234qwer', 'Charlie', 'Joe', 'cjoe22@fmail.com', '143-234-5456')
```

CREDIT_CARD TABLE:

```
INSERT INTO Credit_Card (card_id, username, card_number, expiry_date)
VALUES('11112222', 'bill_nye', '1234123478907890', TO_DATE('12/23', 'MM/YY'));
INSERT INTO Credit_Card (card_id, username, card_number, expiry_date)
VALUES('21213434', 'jill_biden', '3241564787698561', TO_DATE('04/27', 'MM/YY'));
INSERT INTO Credit_Card (card_id, username, card_number, expiry_date)
VALUES('17686544', 'charlie', '7688545590082333', TO_DATE('07/29', 'MM/YY'));
```

COMPANY TABLE:

```
INSERT INTO Company (company_id, company_name, description)
VALUES('00000001', 'Good Eats Co.', 'Offers Healthy Meals On A Budget');
INSERT INTO Company (company_id, company_name, description)
VALUES('00000002', 'Gourmet Sushi Group', 'Premium Sushi At Locations Across the
Country');
INSERT INTO Company (company_id, company_name, description)
VALUES('00000003', 'Tex-Mex Trucks', 'Incredible Burritos Served on College Campuses
Every Friday');
```

RESTAURANTS TABLE:

```
INSERT INTO Restaurants (restaurant_id, company_id, name, description, menu,
listed_hours, open_closed, street_address, city, state, zip_code, cuisine,
is_sit_down, is_take_out, is_food_truck)
VALUES('00000010', '00000001', 'Sauls Deli', 'Delicious Sandwiches On The Go',
'www.saulsdeli.com/menu', 'Monday-Friday: 9:00AM - 5:00PM', 'Open', '123 Huntington
St.', 'Boston', 'MA', '02453', 'American', '1', '1','0');
INSERT INTO Restaurants (restaurant_id, company_id, name, description, menu,
listed_hours, open_closed, street_address, city, state, zip_code, cuisine,
is_sit_down, is_take_out, is_food_truck)
VALUES('00000011', '00000002', 'Sakana Sushi', 'Elevated Restaurant Service Classic
Japanese Dishes', 'www.sakanasushi.com/menu', 'Monday-Sunday: 1:00PM - 9:30PM',
'Open', '430 Massachusetts Ave.', 'Cambridge', 'MA', '02138', 'Asian', '1', '1',
'0');
```

```
INSERT INTO Restaurants (restaurant_id, company_id, name, description, menu,
listed_hours, open_closed, street_address, city, state, zip_code, cuisine,
is_sit_down, is_take_out, is_food_truck)
VALUES('00000012', '00000003', 'La Mexicana Food Truck', 'Variety Of Mexican Meals In
A Food Truck Next To Boston Common', 'www.lamexicanatrucks.com/menu',
'Tuesday-Sunday: 11:00AM - 7:00PM', 'Closed', '115 Boylston St', 'Boston', 'MA',
'02116', 'Hispanic', '0', '1', '1');
```

SIT_DOWN TABLE:

```
INSERT INTO Sit_Down (restaurant_id, capacity)
VALUES('00000010', 20);
INSERT INTO Sit_Down (restaurant_id, capacity)
VALUES('00000011', 40);
```

TAKEOUT TABLE:

```
INSERT INTO Takeout (restaurant_id, wait_time)
VALUES('00000010', '35 mins');
INSERT INTO Takeout (restaurant_id, wait_time)
VALUES('00000011', '20 mins');
INSERT INTO Takeout (restaurant_id, wait_time)
VALUES('00000012', '5 mins');
```

FOOD_TRUCK TABLE:

```
INSERT INTO Food_Truck (restaurant_id, licensed)
VALUES('00000012', 'YES');
```

REVIEWS TABLE:

```
INSERT INTO Reviews (review_id, username, restaurant_id, timestamp, rating, review)
VALUES('00000100', 'bill_nye', '00000010', TO_TIMESTAMP('14:30 02-12-23', 'HH24:MI
MM-DD-YY'), 4, 'Convenient deli with decent sandwiches, bit expensive but the quality
of ingredients justifies the price.');
INSERT INTO Reviews (review_id, username, restaurant_id, timestamp, rating, review)
VALUES('00000101', 'jill_biden', '00000011', TO_TIMESTAMP('20:47 02-20-23', 'HH24:MI
MM-DD-YY'), 5, 'Incredible hidden gem of a sushi restaurant, owners are very
friendly. Ordered the sushi platter and it was delicious!.');
INSERT INTO Reviews (review_id, username, restaurant_id, timestamp, rating, review)
VALUES('00000102', 'charlie', '00000012', TO_TIMESTAMP('11:03 02-24-23', 'HH24:MI
MM-DD-YY'), 3, 'Grabbed a chicken burrito last night at this busy food truck near
Boston Common. Tasted good but was quite small, was still hungry after eating it.');
```

RECOMMENDATIONS TABLE:

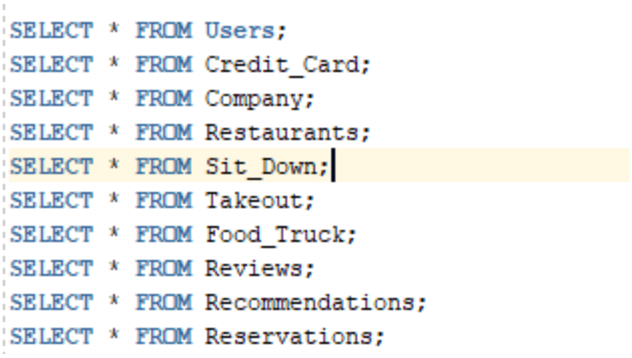
```
INSERT INTO Recommendations (recommend_id, username, restaurant_id)
VALUES('00000001', 'bill_nye', '00000010');
INSERT INTO Recommendations (recommend_id, username, restaurant_id)
```

```
VALUES('00000002', 'jill_biden', '00000011');
INSERT INTO Recommendations (recommend_id, username, restaurant_id)
VALUES('00000003', 'charlie', '00000012');
```

RESERVATIONS TABLE:

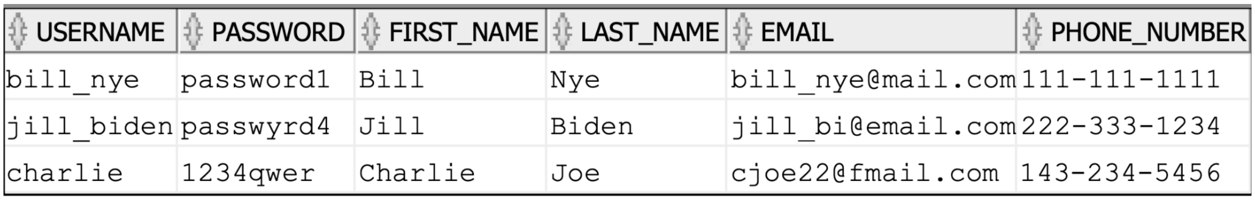
```
INSERT INTO Reservations (reservation_id, username, restaurant_id, start_timestamp,
end_timestamp, party_size)
VALUES('00000001', 'bill_nye', '00000011', TO_TIMESTAMP('19:00 03-01-23', 'HH24:MI
MM-DD-YY'), TO_TIMESTAMP('20:00 03-01-23', 'HH24:MI MM-DD-YY'), 4);
INSERT INTO Reservations (reservation_id, username, restaurant_id, start_timestamp,
end_timestamp, party_size)
VALUES('00000002', 'jill_biden', '00000011', TO_TIMESTAMP('18:30 03-02-23', 'HH24:MI
MM-DD-YY'), TO_TIMESTAMP('19:30 03-02-23', 'HH24:MI MM-DD-YY'), 6);
INSERT INTO Reservations (reservation_id, username, restaurant_id, start_timestamp,
end_timestamp, party_size)
VALUES('00000003', 'charlie', '00000010', TO_TIMESTAMP('13:00 03-03-23', 'HH24:MI
MM-DD-YY'), TO_TIMESTAMP('13:30 03-03-23', 'HH24:MI MM-DD-YY'), 2);
```

Data Loaded in Database:



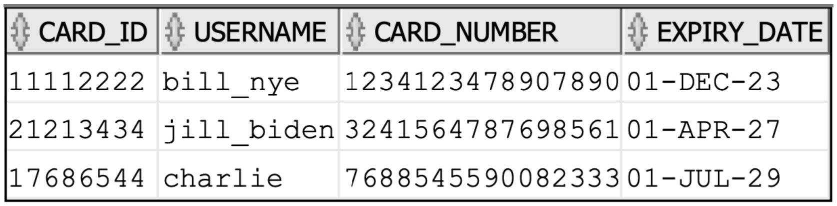
```
SELECT * FROM Users;
SELECT * FROM Credit_Card;
SELECT * FROM Company;
SELECT * FROM Restaurants;
SELECT * FROM Sit_Down;
SELECT * FROM Takeout;
SELECT * FROM Food_Truck;
SELECT * FROM Reviews;
SELECT * FROM Recommendations;
SELECT * FROM Reservations;
```

Users Table:



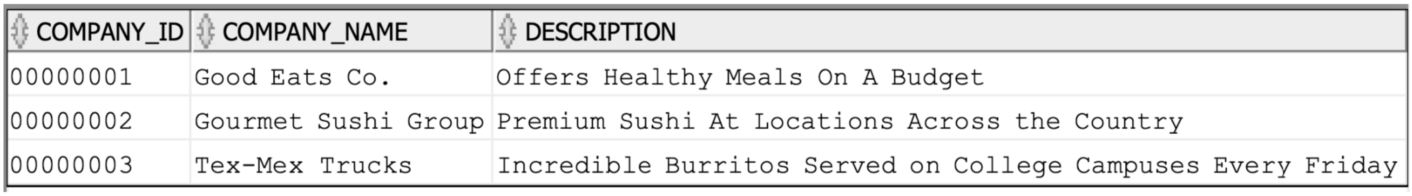
USERNAME	PASSWORD	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER
bill_nye	password1	Bill	Nye	bill_nye@mail.com	111-111-1111
jill_biden	passwyrd4	Jill	Biden	jill_bi@email.com	222-333-1234
charlie	1234qwer	Charlie	Joe	cjoe22@fmail.com	143-234-5456

Credit_Card Table:



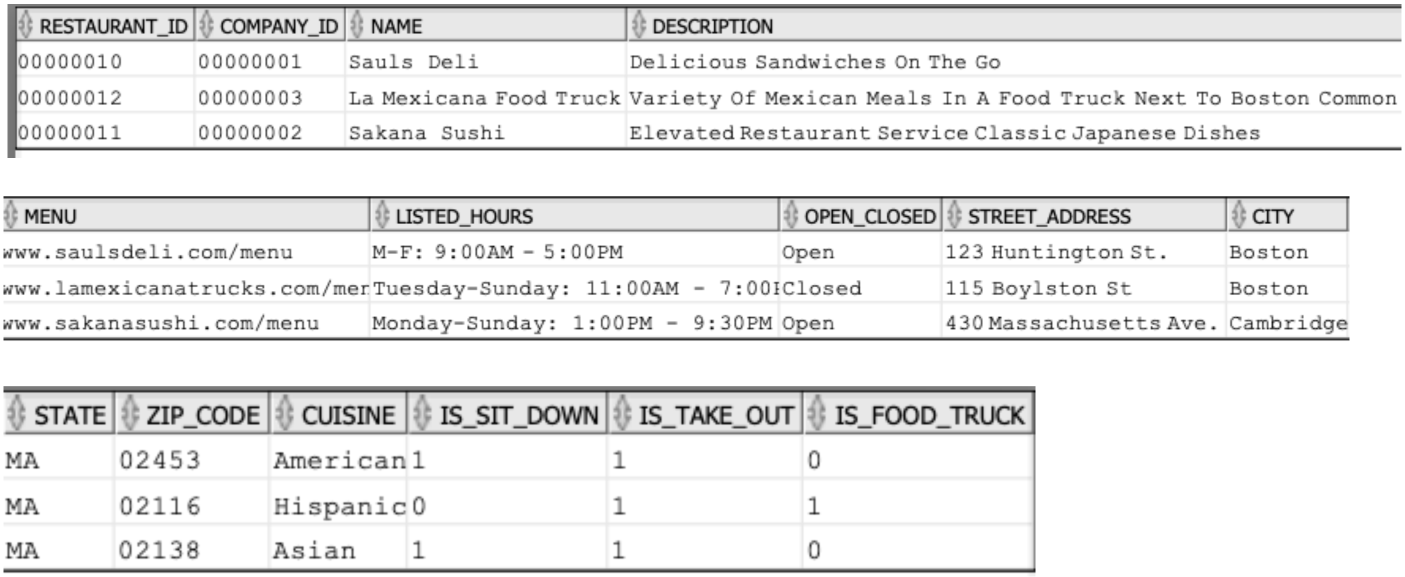
CARD_ID	USERNAME	CARD_NUMBER	EXPIRY_DATE
11112222	bill_nye	1234123478907890	01-DEC-23
21213434	jill_biden	3241564787698561	01-APR-27
17686544	charlie	7688545590082333	01-JUL-29

Company Table:



COMPANY_ID	COMPANY_NAME	DESCRIPTION
00000001	Good Eats Co.	Offers Healthy Meals On A Budget
00000002	Gourmet Sushi Group	Premium Sushi At Locations Across the Country
00000003	Tex-Mex Trucks	Incredible Burritos Served on College Campuses Every Friday

Restaurant Table (split into 3 parts for legibility):

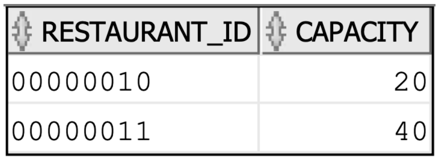


RESTAURANT_ID	COMPANY_ID	NAME	DESCRIPTION
00000010	00000001	Sauls Deli	Delicious Sandwiches On The Go
00000012	00000003	La Mexicana Food Truck	Variety Of Mexican Meals In A Food Truck Next To Boston Common
00000011	00000002	Sakana Sushi	Elevated Restaurant Service Classic Japanese Dishes

MENU	LISTED_HOURS	OPEN_CLOSED	STREET_ADDRESS	CITY
www.saulsdeli.com/menu	M-F: 9:00AM - 5:00PM	Open	123 Huntington St.	Boston
www.lamexicanatrucks.com/mer	Tuesday-Sunday: 11:00AM - 7:00	Closed	115 Boylston St	Boston
www.sakanasushi.com/menu	Monday-Sunday: 1:00PM - 9:30PM	Open	430 Massachusetts Ave.	Cambridge

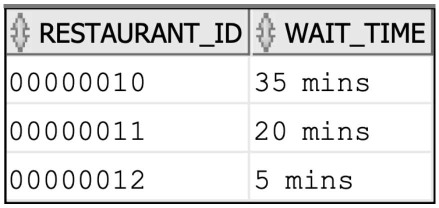
STATE	ZIP_CODE	CUISINE	IS_SIT_DOWN	IS_TAKE_OUT	IS_FOOD_TRUCK
MA	02453	American	1	1	0
MA	02116	Hispanic	0	1	1
MA	02138	Asian	1	1	0

Sit_Down Table:



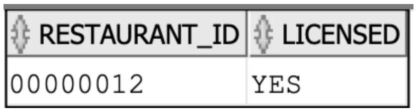
RESTAURANT_ID	CAPACITY
00000010	20
00000011	40

Takeout Table:



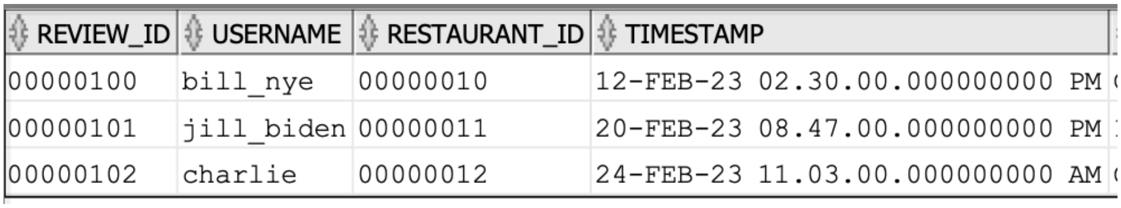
RESTAURANT_ID	WAIT_TIME
00000010	35 mins
00000011	20 mins
00000012	5 mins

Food_Truck Table:

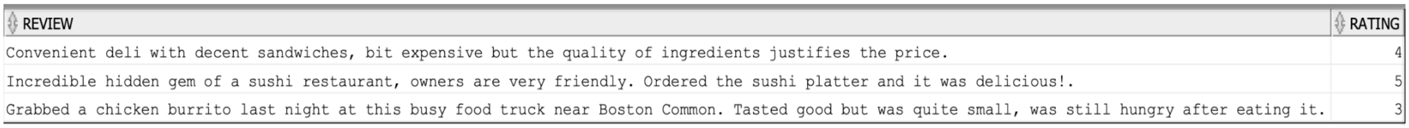


RESTAURANT_ID	LICENSED
00000012	YES

Reviews Table (split into 2 parts for legibility):

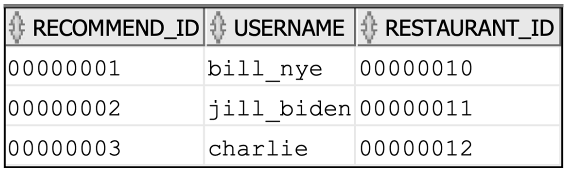


REVIEW_ID	USERNAME	RESTAURANT_ID	TIMESTAMP
00000100	bill_nye	00000010	12-FEB-23 02.30.00.000000000 PM
00000101	jill_biden	00000011	20-FEB-23 08.47.00.000000000 PM
00000102	charlie	00000012	24-FEB-23 11.03.00.000000000 AM



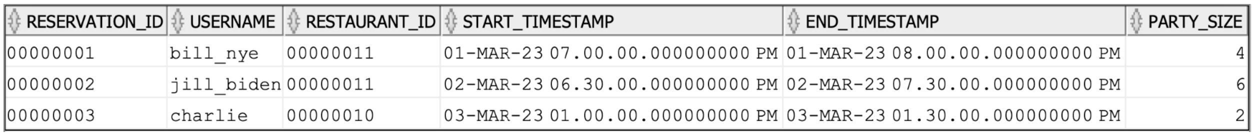
REVIEW	RATING
Convenient deli with decent sandwiches, bit expensive but the quality of ingredients justifies the price.	4
Incredible hidden gem of a sushi restaurant, owners are very friendly. Ordered the sushi platter and it was delicious!.	5
Grabbed a chicken burrito last night at this busy food truck near Boston Common. Tasted good but was quite small, was still hungry after eating it.	3

Recommendations Table:



RECOMMEND_ID	USERNAME	RESTAURANT_ID
00000001	bill_nye	00000010
00000002	jill_biden	00000011
00000003	charlie	00000012

Reservations Table:



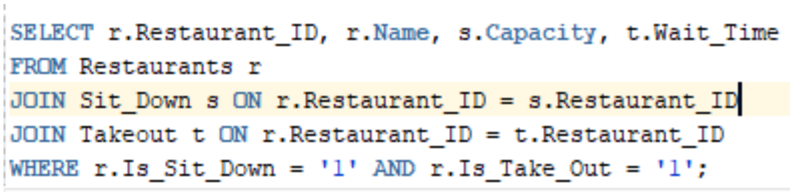
RESERVATION_ID	USERNAME	RESTAURANT_ID	START_TIMESTAMP	END_TIMESTAMP	PARTY_SIZE
00000001	bill_nye	00000011	01-MAR-23 07.00.00.000000000 PM	01-MAR-23 08.00.00.000000000 PM	4
00000002	jill_biden	00000011	02-MAR-23 06.30.00.000000000 PM	02-MAR-23 07.30.00.000000000 PM	6
00000003	charlie	00000010	03-MAR-23 01.00.00.000000000 PM	03-MAR-23 01.30.00.000000000 PM	2

Phase 5: Querying The Database

Queries:

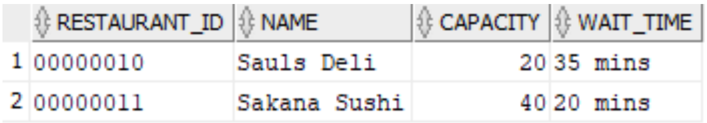
Q1. Which restaurants offer both sit down and takeout services, and what is their capacity and wait time?

SQL Query:



```
SELECT r.Restaurant_ID, r.Name, s.Capacity, t.Wait_Time
FROM Restaurants r
JOIN Sit_Down s ON r.Restaurant_ID = s.Restaurant_ID
JOIN Takeout t ON r.Restaurant_ID = t.Restaurant_ID
WHERE r.Is_Sit_Down = '1' AND r.Is_Take_Out = '1';
```

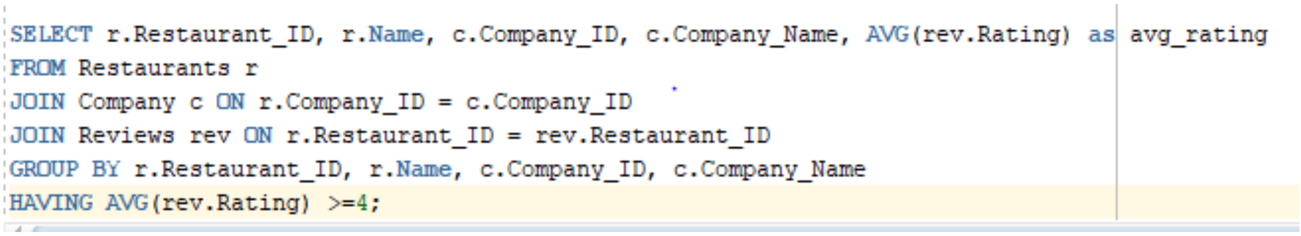
Output:



	RESTAURANT_ID	NAME	CAPACITY	WAIT_TIME
1	00000010	Sauls Deli	20	35 mins
2	00000011	Sakana Sushi	40	20 mins

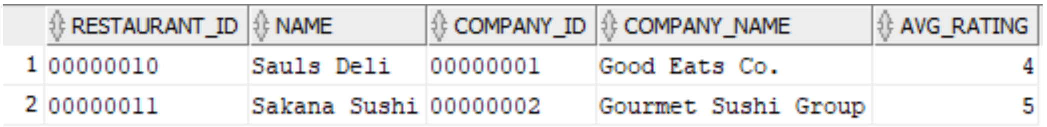
Q2. Which restaurants have an average rating of at least 4, and which companies do they belong to?

SQL Query:



```
SELECT r.Restaurant_ID, r.Name, c.Company_ID, c.Company_Name, AVG(rev.Rating) as avg_rating
FROM Restaurants r
JOIN Company c ON r.Company_ID = c.Company_ID
JOIN Reviews rev ON r.Restaurant_ID = rev.Restaurant_ID
GROUP BY r.Restaurant_ID, r.Name, c.Company_ID, c.Company_Name
HAVING AVG(rev.Rating) >=4;
```

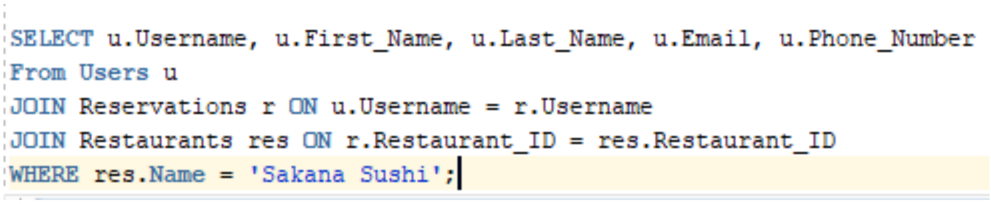
Output:



	RESTAURANT_ID	NAME	COMPANY_ID	COMPANY_NAME	AVG_RATING
1	00000010	Sauls Deli	00000001	Good Eats Co.	4
2	00000011	Sakana Sushi	00000002	Gourmet Sushi Group	5

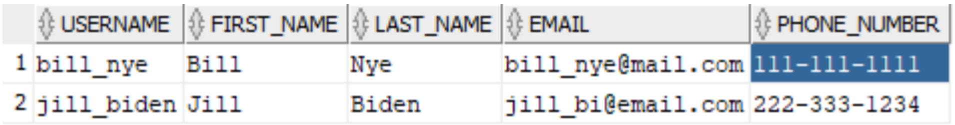
Q3. Which users made reservations at Sakana Sushi, and what are their emails and phone numbers?

SQL Query:



```
SELECT u.Username, u.First_Name, u.Last_Name, u.Email, u.Phone_Number
From Users u
JOIN Reservations r ON u.Username = r.Username
JOIN Restaurants res ON r.Restaurant_ID = res.Restaurant_ID
WHERE res.Name = 'Sakana Sushi';
```

Output:



	USERNAME	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER
1	bill_nye	Bill	Nye	bill_nye@mail.com	111-111-1111
2	jill_biden	Jill	Biden	jill_bi@email.com	222-333-1234

Learnings:

This project helped our group better understand the different phases of database design. Starting with only a set of business rules, the project provided us the opportunity to work through each step in the design process, giving us a better understanding of why these steps are necessary. Working on this project provided important clarity on how the theoretical schemas we have been building through this course actually get translated into a functioning database.