CS4111 Project 1 Part 2

Xinyue Wu, Xilin Wang
uni: xw2766, xw2767

PostgreSQL account name: xw2767
Password: 7210

Queries:

1. Return the restaurant name and average rating for Japanese restaurants that have an average rating greater than 4.4 and offer vegan dishes.

```
SELECT R.name, AVG(UR.rating)
FROM Restaurant R, u_writes_for UW, User_Reviews UR, Satisfies S
WHERE R.category = 'Japanese'
        AND R.rid = UW.rid
        AND R.rid = S.rid
        AND UW.urid = UR.urid
        AND S.name = 'Vegan'
GROUP BY R.name
HAVING AVG(UR.rating)>=4.4;
```

2. Find name and location of Chinese restaurants that offer dine-in options and Gluten free dishes.

```
SELECT R.name, L.number, L.street
FROM Restaurant R NATURAL JOIN Is_at_Locations L
WHERE R.rid in (
            SELECT O.rid
            FROM Offers O NATURAL JOIN Satisfies S
            WHERE O.type LIKE '%dine-in%' AND S.name = 'Gluten Free')
            AND R.category ='Chinese';
```

3. Find details about user 'Ari Jiang' including which restaurant they reviewed recently, the restaurant category, review date, rating and review details. This information could be used by us to recommend restaurants to the user in the future.

```
SELECT U.name, R.name, R.category, UW.review_date, UR.rating, UR.detail
FROM users U, u_writes_for UW, User_Reviews UR, Restaurant R
WHERE U.name = 'Ari Jiang'
        AND U.uid = UW.uid
        AND UW.urid = UR.urid
        AND UW.rid = R.rid;
```

SQL Schema:

```
CREATE TABLE Restaurant (
    rid VARCHAR(60),
    name VARCHAR(30),
    website VARCHAR(512), -- Estimating url length here
    phone VARCHAR(14) -- formatted phone number
    category VARCHAR(20),
    PRIMARY KEY (rid));

CREATE TABLE Order_Options (
    type VARCHAR(20), -- can be dine-in, take-out, delivery, etc
    commission VARCHAR(30),
    PRIMARY KEY (type));

CREATE TABLE offers (
    rid VARCHAR(60),
    type VARCHAR(20),
    PRIMARY KEY (rid, type),
    FOREIGN KEY (rid) REFERENCES Restaurant
    ON DELETE CASCADE
    ON UPDATE CASCADE,
    FOREIGN KEY (type) REFERENCES Order_Options
    -- participation of Restaurant in offers is total
    );

CREATE TABLE Users (--OK
    uid VARCHAR(22),
    name VARCHAR(30),
--  email VARCHAR(50), -- estimation of email length deleted
    PRIMARY KEY (uid));
    --We deleted email here because we couldn't get email addresses of users from the data we crawled, instead, we got the user's id from google api which is guaranteed to be unique

CREATE TABLE User_Reviews (
    urid VARCHAR(23),
    rating SMALLINT CHECK (rating >=0 AND rating <=5),
    detail VARCHAR(5000), -- yelp has max 5000 characters constraints
    PRIMARY KEY (urid));-- User_Reviews' participation in u_writes_for is total
```

```
CREATE TABLE u_writes_for (
    uid VARCHAR(22),
    urid VARCHAR(23),
    rid VARCHAR(60) NOT NULL,
    review_date TIMESTAMP,
    PRIMARY KEY (uid,urid),
    FOREIGN KEY (uid) REFERENCES Users
    ON DELETE CASCADE,
    FOREIGN KEY (urid) REFERENCES User_Reviews
    ON DELETE CASCADE,
    FOREIGN KEY (rid) REFERENCES Restaurant
    ON DELETE CASCADE
    ON UPDATE CASCADE);


CREATE TABLE Dietary_Need (
    name VARCHAR(20),
    PRIMARY KEY (name));

CREATE TABLE Has (
    uid VARCHAR(22),
    name VARCHAR(20),
    PRIMARY KEY (uid, name),
    FOREIGN KEY (uid) REFERENCES Users,
    FOREIGN KEY (name) REFERENCES Dietary_Need);

CREATE TABLE Satisfies (
    name VARCHAR(20),
    rid VARCHAR(60),
    mid VARCHAR(20),
    PRIMARY KEY (name, rid, mid),
    FOREIGN KEY (name) REFERENCES Dietary_Need,
    FOREIGN KEY (rid,mid) REFERENCES Provides_Menu
    ON DELETE CASCADE);

CREATE TABLE Provides_Menu (--OK
    mid VARCHAR(20),
    rid VARCHAR(60),
    PRIMARY KEY (rid, mid),
    FOREIGN KEY (rid) REFERENCES Restaurant
    ON DELETE CASCADE);
```

```
CREATE TABLE Is_at_Locations ( -- combining Is_at and Locations to one table
    rid VARCHAR(20),
    number VARCHAR(5),
    street VARCHAR(50),
    city VARCHAR(10),
    state VARCHAR(10),
    zip CHAR(5), VARCHAR(10)
    PRIMARY KEY (rid, number, street, zip),
    FOREIGN KEY (rid) REFERENCES Restaurant
    ON DELETE CASCADE
    -- Restaurant's participation in Is_at_Locations is total
);
```