

Database Project

Restaurant Recommendation System

NYU Tandon - CS-GY_6083 - Fall 2021

Team Members:

Shardha Koul (sk9225)

Viren Parmar (vmp2018)

Description:

We have created a restaurant database system. End users are the primary endpoint that will interact with the system. Users can perform various functional tasks such as: Registering themselves, registering their addresses, reserving a table, writing a review, search for restaurants based on zip code, city or ratings, view given ratings for a particular restaurant, view his/her profile and can find discount coupons for a particular restaurant.

We have the following entity sets:

User: A user registers himself with the application. They provide the following details: Name, email, phone number, date of birth and address. User is uniquely identified by an email. One user can store multiple addresses.

Restaurant: Each restaurant has restaurant id, name, complete address (with city, state and zip code). Restaurants also showcase the approximate cost of dining for two people. They also have their operating hours for each day of the week. They also indicate if they have the following amenities: Wifi, parking, wheelchair access and what kind of alcohol they serve.

Reviews: A user can provide reviews to restaurants that they have visited. The reviews include an rating on a scale 5 for the following items: Ambience, Food Quality, Service and Overall Experience. Users can also provide an optional description.

Coupons: A coupon has a Coupon id, expiry date and the discount amount. User can view the coupons they are applicable for

Reservation: A user reserves a table in a restaurant. They can have multiple reservations at the same restaurant on different days. We store past reservations as well. A reservation has a date, time and table number.

Entity Sets:

- users
- reviews
- address
- reservations
- amenities
- restaurant
- coupons

Relationship sets:

- users_address
- user_coupons
- operating_hours
- restaurant_coupons
- Restaurant_address

Constraints:

- One user can have multiple addresses: User has a participation constraint on addresses.
- One restaurant will have exactly one address: Restaurant has key and participation constraint on addresses.
- Operating hours of restaurants can only exist if a restaurant is present in the database i.e. it is a weak entity set
- Amenities of restaurants can only exist if a restaurant is present in the database i.e. is a weak entity set.

Dataset:

We have taken the data from kaggle.com. Link is as listed as below:

<https://www.kaggle.com/yelp-dataset/yelp-dataset>

Our dataset contains:

Business: This collection consists of several details related to every business such as location, business id, review count, operating hours and what type of amenities it has etc. It contains almost every detail related to the business.

Review: The review collection of this dataset contains columns such as the user id, their rating and the message that they post as review.

User: We have created a dummy data table for users. This contains several details of the yelp users such as the email id, name, phone number, dob and address.

Coupons: We have created a dummy data table for coupons with coupon id, expiry date and discount amount which a restaurant provides or a user has.

User Interaction:

- There can be one type of users: a customer(user)
- A user can perform the below tasks:
 - 1) Basic details(email, name, dob, mobile number) and view the same in their profile
 - 2) Add addresses, search for restaurants in a particular zip code or city.
 - 3) Users can give a review to a restaurant(ambience, food quality, service etc) and also provide a text description.
 - 4) Users can also search a restaurant using review accordingly as per the amenities(ambience, food quality, service)

For example: If a user enters their preference as food quality they will receive a list of restaurants according to the reviews.

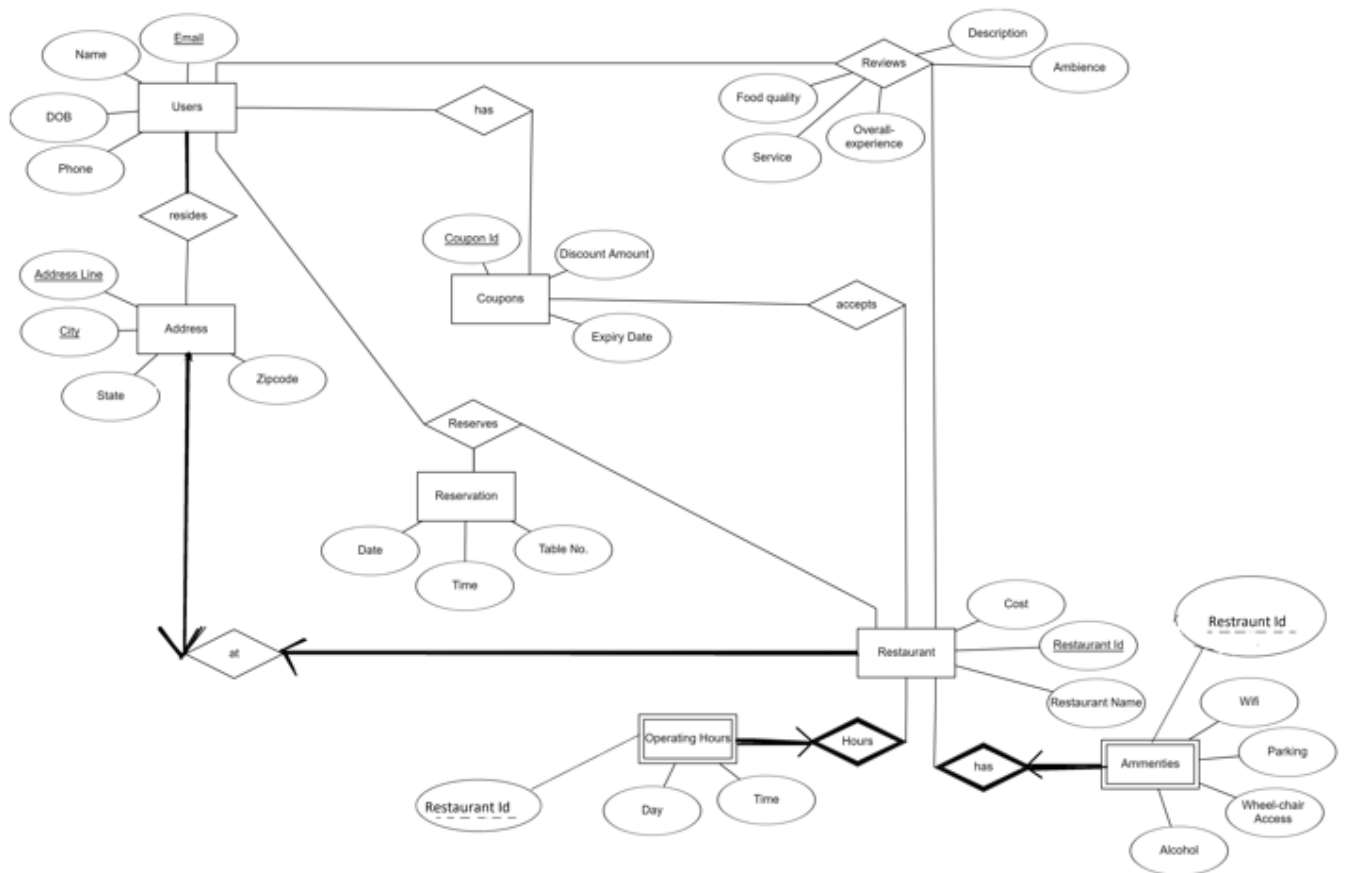
- 5) Book a table at a restaurant by selecting a table number for a particular date and time.

For example: If a user wants to reserve a table they will enter the time and date and also the table number.

- 6) A user can also view all reviews for a restaurant.

- A restaurant can perform the below tasks:
 - 1) A restaurant to add/edit their address
 - 2) A restaurant can check reviews received on their restaurants. They can also check the reservations made at their restaurant by a user.

ER Diagram:



Schema (Relational Model:

--Table: Users

Drop table users_address;

Drop table user_coupons;

Drop table operating_hours;

Drop table restaurant_coupons;

Drop table restaurant_address;

Drop table reviews;

Drop table address;

Drop table reservations;

Drop table amenities;

Drop table restaurant;

Drop table coupons;

Drop table users;

```
create table users(  
email char(128) primary key,  
phone char(10),  
name varchar(128),  
Dob date  
);
```

--Table: Address

```
create table address(  
  address_line varchar(128),  
  city char(64),  
  state char(64),  
  zipcode integer,  
  Primary key(address_line, city)  
);
```

--Table: Restaurants

```
Create table restaurant(  
  Restaurant_id char(32) primary key,  
  restuarant_name varchar(128),  
  cost real  
);
```

--Table: Addresses of users

```
Create table users_address(  
  U_email char(128),  
  Address_line varchar(128),  
  City char(64),  
  Primary key(u_email, address_line, city),  
  Foreign key(u_email) references users(email)  
);
```

--Table: Address of restaurant

```
Create table restaurant_address(  
  Reataurant_id char(32) primary key,  
  Address_line varchar(128),  
  City char(64),  
  Foreign key(Reataurant_id) references Restaurant(restaurant_id)  
);
```

--Table: reviews

```
create table reviews(  
  ambience integer check (ambience <= 5 and ambience >= 0),  
  food_quality integer check (food_quality <= 5 and food_quality >= 0),
```


service integer check (service <= 5 and service >= 0),
overall_experience integer check (overall_experience <= 5 and overall_experience >= 0),
Description varchar(256),
User_email char(128),
Restaurant_id char(32),
Primary key (User_email, Restaurant_id),
Foreign key(user_email) references users(email),
Foreign key(Restaurant_id) references restaurant(restaurant_id)
)

--Table:Reservations

Create table reservations(
date_reserved date,
time_reserved time,
Table_no integer,
User_email char(128),
Restaurant_id char(32),
Primary key(user_email, restaurant_id),
Foreign key(User_email) references users(email),
Foreign key(Restaurant_id) references restaurant(restaurant_id)
);

--Table: Operating hours

```
Create table operating_hours(  
  Restaurant_id char(32),  
  Day char(10),  
  Time char(12),  
  Primary key(restaurant_id, day, time),  
  Foreign key(restaurant_id) references restaurant(restaurant_id)  
    on delete cascade  
);
```

--Table: Amenities

```
Create table amenities(  
  Restaurant_id char(32) primary key,  
  Wifi char(1),  
  Parking char(1),  
  Wheelchair char(1),  
  Alcohol char(64),  
  Foreign key(restaurant_id) references restaurant(restaurant_id)  
    on delete cascade  
);
```

--Table: Coupons

```
Create table coupons(  
  Coupon_id char(32) primary key,  
  Expiry_date date,  
  Discount_amt real  
);
```

--Table: User coupons

```
Create table user_coupons(  
  user_email char(128),  
  Coupon_id char(32),  
  Primary key(user_email, coupon_id),  
  Foreign key(User_email) references users(email),  
  Foreign key(coupon_id) references coupons(Coupon_id)  
);
```

--Table: Restaurant coupons

```
Create table restaurant_coupons(  
  Restaurant_id char(32),  
  Coupon_id char(32),  
  Primary key(restaurant_id, coupon_id),  
  Foreign key(restaurant_id) references restaurant(restaurant_id),  
  Foreign key(coupon_id) references coupons(Coupon_id)  
);
```