

University of Ottawa
School of Electrical Engineering and Computer Science
CSI2132- Winter 2018
Database Project Specification: Restaurant Rating Database

Instructions

1. Complete this project in a group of two (2) to three (3) students.
2. Demonstrate the project on **Monday April 9th, 2018** in the allocated timeslot for each group.
3. Send a list of names and student no. of your group members (only one member per group send these information) to your Lab TA, before **Monday March 12th, 2018**.
4. Use PostgreSQL to complete this project, together with Java and JSP, or PHP, to create your Web-based front-end.

Deliverables:

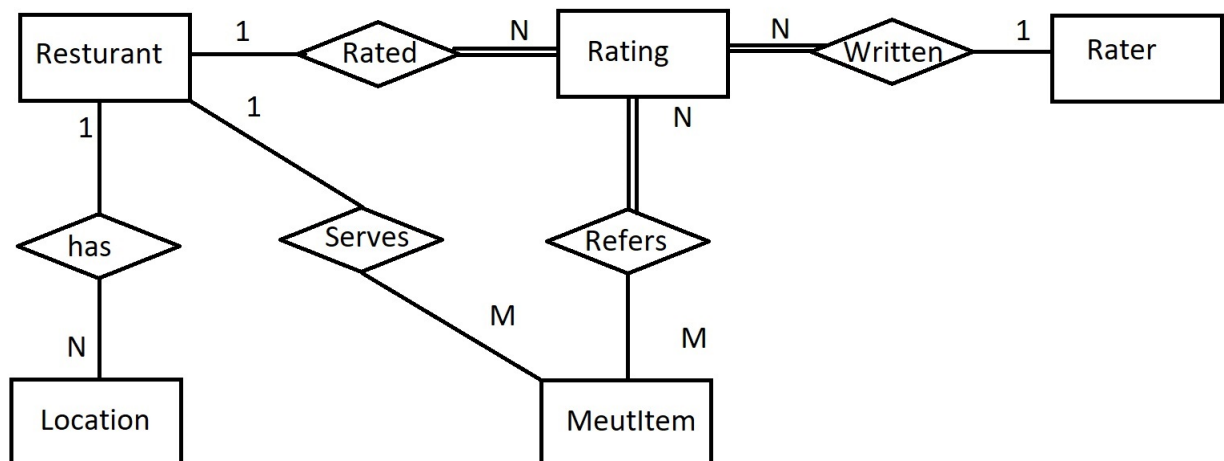
Submit all your source code via Virtual Campus, **before Monday April 9th, 2018 at 14h00**. (That is, before the demonstrations will start.) Note that all group members should submit the source code, not just one per group. All group members should attend the project demonstrations.

Your task:

Consider a database that contains ratings for restaurants in the Ottawa region. In this database, raters need to register and can then complete one or more ratings of multiple restaurants. This database is accessible via the Internet and thus provides a forum for the public to use when deciding where to dine. All raters need to register to post a rating on the website.

(You may want to take a look at <http://www.urbanspoon.com/c/250/Ottawa-restaurants.html> to see an example of such a website. This website contains Menus, Photos, Ratings and Reviews for over 2684 Restaurants in Ottawa.)

Here is a high level summary of the EER diagram, which only shows the entities and relationships.



Here is a list of some assumptions that were made.

1. A person (rater) may rate one to many restaurants, but a rating corresponds to only one restaurant at a time.
2. There may be zero to many ratings for a specific restaurant, but a rating corresponds to exactly one restaurant.
3. A restaurant has one or more locations, and a location corresponds to exactly one restaurant.
4. A rating may refer to zero or more menu items and a specific menu item may be referenced in zero or more ratings. Here, food and beverages are both considered as menu items.
5. A restaurant offers one or more menu items, and a specific menu item is associated with exactly one restaurant. That is, each restaurant has a unique identifier for items.

Here is a partial definition of the relations you will need to create. You may want to add additional information/attributes to personalise the project.

Rater(UserID, email, name, join-date, type, reputation, ...)

The join-date is used to show when this rater first joined the website. The name field corresponds to an alias such as SuperSizeMe. Type refers to the type of rater (blog, online, food critic) and reputation takes a value between 1 and 5. The value of this field is based on the number of people who found this rater's opinion helpful, and the default value is 1 (lowest).

Rating(UserID, Date, Price, Food, Mood, Staff, Comments, ..., RestaurantID)

The Price, Food, Mood and Staff attributes may take a value between 1 (low) to 5 (high). The comments field is reserved for free text and will be used, in future, for sentiment analysis. Note that UserID and RestaurantID are foreign keys.

Restaurant(RestaurantID, Name, Type, URL, ...)

This relation contains general information about a restaurant and is useful in the case where a restaurant chain has many locations. The type attribute contains details about the cuisine, such as Italian, Indian, Middle Eastern, and so on.

Location(LocationID, first-open-date, manager-name, phone-number, street-address, hour-open, hour-close, ..., RestaurantID)

This relation contains the location-specific data, such as the manager's details, the phone number, the address, and so on. Note that RestaurantID is the foreign key. This design assumes that the restaurant opens and closes at the same time every day; you may modify this design if you wish.

MenuItem(ItemID, name, type, category, description, price, ..., RestaurantID)

Here we include the item name, as on the menu, the category (starter, main, desert) as well as the type (food or beverage). RestaurantID is the foreign key.

RatingItem(UserID, Date, ItemID, rating, comment, ...)

A rater may explicitly select the menu item, and add a specific rating between 1 (low) to 5 (high) and a free text comment. All menu items should be selected from a list.

Requirements and Mark Allocation

You are required to complete the following tasks: **(Total 100 marks)**

1. **(10 marks)** Transform the description into a relational model and create all the tables in PostgreSQL. Add all other relevant attributes and remember to enforce entity and referential integrity.
2. **(10 marks)** Populate the tables with your own data, using the restaurants of your own choice. It follows that your data, and the attribute values you choose, should be sufficient in order to implement and test the queries specified below. Your database should include at least 40 different menu items. There should be at least 12 restaurants that contain around 8 ratings each. The number of raters should be at least 15, with the highest number of rating by a specific rater around 10.
3. **(10 marks)** Provide the user with the ability to add data to, and delete data from, the following tables in your database: Restaurant, Rater and MenuItems.
4. **(40 marks)** Create a number of SQL queries to explore this data. The following is a list of “typical” queries that should be implemented.

Restaurants and menus

- a. Display all the information about a user-specified restaurant. That is, the user should select the name of the restaurant from a list, and the information as contained in the restaurant and location tables should then be displayed on the screen.
- b. Display the full menu of a specific restaurant. That is, the user should select the name of the restaurant from a list, and all menu items, together with their prices, should be displayed on the screen. The menu should be displayed based on menu item categories.
- c. For each user-specified category of restaurant, list the manager names together with the date that the locations have opened. The user should be able to select the category (e.g. Italian or Thai) from a list.
- d. Given a user-specified restaurant, find the name of the most expensive menu item. List this information together with the name of manager, the opening hours, and the URL of the restaurant. The user should be able to select the restaurant name (e.g. El Camino) from a list.
- e. For each type of restaurant (e.g. Indian or Irish) and the category of menu item (appetiser, main or desert), list the average prices of menu items for each category.

Ratings of restaurants

- f. Find the total number of rating for each restaurant, for each rater. That is, the data should be grouped by the restaurant, the specific raters and the numeric ratings they have received.
- g. Display the details of the restaurants that have not been rated in January 2015. That is, you should display the name of the restaurant together with the phone number and the type of food.
- h. Find the names and opening dates of the restaurants that obtained Staff rating that is lower than any rating given by rater X. Order your results by the dates of the ratings. (Here, X refers to any rater of your choice.)
- i. List the details of the Type Y restaurants that obtained the highest Food rating. Display the restaurant name together with the name(s) of the rater(s) who gave these ratings. (Here, Type Y refers to any restaurant type of your choice, e.g. Indian or Burger.)

- j. Provide a query to determine whether Type Y restaurants are “more popular” than other restaurants. (Here, Type Y refers to any restaurant type of your choice, e.g. Indian or Burger.)
Yes, this query is open to your own interpretation!

Raters and their ratings

- k. Find the names, join-date and reputations of the raters that give the highest overall rating, in terms of the Food and the Mood of restaurants. Display this information together with the names of the restaurant and the dates the ratings were done.
- l. Find the names and reputations of the raters that give the highest overall rating, in terms of the Food or the Mood of restaurants. Display this information together with the names of the restaurant and the dates the ratings were done.
- m. Find the names and reputations of the raters that rated a specific restaurant (say Restaurant Z) the most frequently. Display this information together with their comments and the names and prices of the menu items they discuss. (Here Restaurant Z refers to a restaurant of your own choice, e.g. Ma Cuisine).
- n. Find the names and emails of all raters who gave ratings that are lower than that of a rater with a name called John, in terms of the combined rating of Price, Food, Mood and Staff. (Note that there may be more than one rater with this name).
- o. Find the names, types and emails of the raters that provide the most diverse ratings. Display this information together with the restaurants names and the ratings. For example, Jane Doe may have rated the Food at the Imperial Palace restaurant as a 1 on 1 January 2015, as a 5 on 15 January 2015, and a 3 on 4 February 2015. Clearly, she changes her mind quite often.

Please note that, in the above, the queries may also return only one name. (That is, there may be only one person satisfying these queries. Be sure to have enough test cases to ensure your queries are robust.)

- 5. **(30 marks)** Create a web-based front-end, for the user to directly query the database.
- 6. Additional effort, such as creating a superb front-end, conducting sentiment analysis (i.e., text mining), or including a multimedia component, may earn you up to **20 bonus** marks.