Restaurant reviews in Yelp

Project Proposal

Yudong Lu, Amal Byju, Chunghao Lee, Rachel Pothen

Brand:

Top Information Per Starver (TIPS)

Business processes/transactions:

Users can post a question or answer a question in the community section

Users can review and rate a restaurant

Users can view the menu of the restaurant

Users can save a restaurant in a collection

Entity types:

* + Users
  + Restaurant
  + Review
  + Menu
  + Collection (set of selected restaurants by a user)
  + Post (Question & Answer)

Relationship types:

Create: binary relationship

1 user to 0 or more collections

1 collection to 1 user

Contains (savedDate): binary relationship

1 collection to 0 or more restaurants

1 restaurant to 0 or more collections

Have: binary relationship

1 restaurant to 0 or more menu items

1 menu item to 1 restaurant

Send: binary relationship

1 user to 0 or more posts

1 post to 1 user

To: binary relationship

1 post to 1 restaurant

1 restaurant to 0 or more posts

Answer: unary relationship

1 post to 0 or more posts

1 post to 0 or 1 post

ER schema:

User(**userID**, userName, -userFName, -userLName, userDOJ, userLoc, userElite)

Review(**reviewID**, reviewRating, reviewDescription, reviewUseful, reviewFunny, reviewCool, reviewDate)

MenuItem(**itemName**, itemDesc, itemPrice)

Restaurant (**restaurantID**, restaurantName, restaurantAddr, restaurantWeb, restaurantCategory, restaurantPriceRange, =restaurantRating, =reviewCount)

Collection (**collectionID**, collectionName, collectionDescription, collectionPublicity)

Post (**postID**, postName, postContent, postType, postDate)

ER Diagram:

Diagram

Description automatically generated

Mission statement:

To analyse user reviews, ratings, posts and tags for different restaurants on Yelp website. To build insights on user preferences, popular restaurants by category, rating distributions and so on.

Mission objectives:

1. Find the top 5 highest rated restaurants in each category
2. Find the top 5 highest rated restaurants in each price range
3. Find the number of 1-star, 2-star, 3-star, 4-star and 5-star reviews for a chosen restaurant.
4. Find the number of 1-star, 2-star, 3-star, 4-star and 5-star reviews for a chosen user.
5. Find the top 5 most popular collections.
6. Find the top 5 most useful reviews for a chosen restaurant.
7. Find the top 5 categories with the least number of restaurants.
8. Find the top 5 categories with the most number of restaurants.
9. Find the top 10 users who have posted the maximum number of reviews or posts and find the number of elite users among them.
10. Extract the top 3 most recent posts for a chosen restaurant.

Relational schema:

User(**userID**, userFName, userLName, userDOJ, userLoc, userElite)

Restaurant (**restaurantID**, restaurantName, restaurantAddr, restaurantWeb, restaurantCategory, restaurantPriceRange)

Review(**reviewID**, reviewRating, reviewDescription, reviewUseful, reviewFunny, reviewCool, reviewDate, *userID*, *restaurantID*)

MenuItem(***restaurantID***, **itemName**, itemDesc, itemPrice)

Collection (**collectionID**, collectionName, collectionDescription, collectionPublicity, *userID*)

Post (**postID**, postName, postContent, postType, postDate, *userID*, *restaurantID*, *questionID*)

Contains(***collectionID***, ***restaurantID***, savedDate)

Functional dependencies:

userID -> userFName, userLName, userDOJ, userLoc, userElite

reviewID -> reviewRating, reviewDescription, reviewUseful, reviewFunny, reviewCool, reviewDate, userID, restaurantID

restaurantID, itemName -> itemDesc, itemPrice

restaurantID -> restaurantName, restaurantAddr, restaurantWeb, restaurantCategory, restaurantPriceRange

collectionID -> collectionName, collectionDescription, collectionPublicity, userID

postID -> postName, postContent, postType, postDate, userID, restaurantID, questionID

collectionID, restaurantID -> savedDate

Business rules and referential integrity actions:

[R1] When user information is deleted from or changed in the database, then their reviews, collections and posts should be deleted or changed accordingly.

[R2] When restaurant information is deleted from or changed in the database, then their reviews, menu items and posts should be deleted or changed accordingly.

[R3] When a collection is deleted or updated, then the restaurants contained in the collection should be deleted or updated as well.

[R4] When a restaurant that is present in a collection is deleted or updated, then the collection should be deleted or updated as well.

[R5] When a post is deleted or modified, then the posts answering that post should be deleted or modified as well.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Relation | Foreign Key | Base Relation | Primary Key | Business Rule | Constraint: ON DELETE | Business Rule | Constraint: ON UPDATE |
| Review | userID | User | userID | R1 | CASCADE | R1 | CASCADE |
| Review | restaurantID | Restaurant | restaurantID | R2 | CASCADE | R2 | CASCADE |
| MenuItem | restaurantID | Restaurant | restaurantID | R2 | CASCADE | R2 | CASCADE |
| Collection | userID | User | userID | R1 | CASCADE | R1 | CASCADE |
| Post | userID | User | userID | R1 | CASCADE | R1 | CASCADE |
| Post | restaurantID | Restaurant | restaurantID | R2 | CASCADE | R2 | CASCADE |
| Post | questionID | Post | postID | R5 | NO ACTION | R5 | NO ACTION |
| Contains | collectionID | Collection | collectionID | R3 | CASCADE | R3 | CASCADE |
| Contains | restaurantID | Restaurant | restaurantID | R4 | CASCADE | R4 | CASCADE |

Sample data:

User

(**userID**, userFName, userLName, userDOJ, userLoc, userElite)

('U00000001','Kevin','W.','1-Sep-09','Saint Louis',1),

('U00000002','Tamara','W.','15-Jan-13','Laurel',0)

Restaurant

(**restaurantID**, restaurantName, restaurantAddr, restaurantWeb, restaurantCategory, restaurantPriceRange)

('R00000001','The Foundry Bakery','Maryland Heights','thefoundrybakery.com','Bakery','$'),

('R00000002','Mission BBQ','Greenbelt','mission-bbq.com','Barbeque','$$')

Review

(**reviewID**, reviewRating, reviewDescription, reviewUseful, reviewFunny, reviewCool, reviewDate, *userID*, *restaurantID*)

('W00000001','5','This is a must try for anyone. My recommendations for a first visit are the taro globe and Portuguese egg tart.',1,0,2,'22-Oct-21','U00000001','R00000001'),

('W00000002','2','The ribs are always too salty even if you douse with a lot of sweet BBQ sauce. Not enough meat on the bones.',0,0,0,'1-Jul-21','U00000002','R00000002')

MenuItem

(***restaurantID***, **itemName**, itemDesc, itemPrice)

('R00000001','Umami Burst','Japanese miso with shiitake mushroom',10),

('R00000001','Chocolate Cherry','French cocoa powder with Belgium chocolate chips and Michigan tart cherries',10)

Collection

(**collectionID**, collectionName, collectionDescription, collectionPublicity, *userID*)

('C00000001','Fast Food Restaurants','My preferred fast food restaurants',51,'U00000001'),

('C00000002','College Park Top Picks','My preferred restaurants in College Park, MD',74,'U00000002')

Post

(**postID**, postName, postContent, postType, postDate, *userID*, *restaurantID*, *questionID*)

('P00000005','Covid','Open during Covid? Dine in?','Q','20-Aug-17','U00000005','R00000025',NULL),

('P00000006','Covid','Yes','A','25-Aug-17','U00000006','R00000025','P00000005')

Contains

(***collectionID***, ***restaurantID***, savedDate)

('C00000001','R00000015','4-Apr-17'),

('C00000001','R00000029','15-Apr-17')