**Brand Name:**

**Terpilicious Eats**

**#FeartheFood**

****

(credit: <https://twitter.com/FreeFoodUMD> )

**Business Processes & Transactions:**

* Each restaurant is described by a unique ID, name, address, and phone number.
* The location represents the general area of the restaurant, for instance Stamp. Each location is described by a unique ID, name, and its longitude and latitude.
* The menu that each restaurant sells is stored by a unique ID and a URL address to the restaurant’s menu.
* Each customer can write a review for the restaurant. For each review, we give it a unique ID, then store the rating (on a scale of 0 to 5 stars) and the text of the review.
* We store the information of customers who eat at the restaurants. Give each a unique customer ID then store their last name, first name, email address, zip code, and date of birth.
* Each employee who works at the restaurants is described by a unique ID, their last name, first name, position, salary, and their manager ID.
* Restaurants must have at least one employee. An Employee should only be able to work at one restaurant.
* A location may have no restaurants or many restaurants. A restaurant can be part of only one location.
* Each restaurant should have their menu information where each menu pertains to at least 1 restaurant.
* A restaurant can have no reviews or one review per customer. A customer can write one review per restaurant. Each review must be written by just one customer.
* Many customers can eat at a restaurant and, similarly, a single customer can eat at many restaurants. For a customer to be considered, they need to have eaten at least one restaurant. However, for a restaurant to be considered, it need not have any customers yet.

**ER Schema:**

Entities, Attributes and Primary Keys

Restaurant (**restaurantId**, restaurantName, restaurantAddress, -restaurantStreet, -restaurantCity, - restaurantState, -restaurantZipCode, restaurantPhoneNo)

Location (**locationId**, locationName, locationCoordinate, -locationLongitude, -locationLatitude)

Menu (**menuId**, menuUrl)

Review (**reviewId**, reviewRating, reviewText)

Customer (**customerId**, customerName, -customerLastName, -customerFirstName, customerEmail, customerZipCode, customerDOB)

Employee (**employeeId**, employeeName, -employeeLastName, -employeeFirstName, employeePosition, employeeSalary)

Relationships, Attributes, Degrees, Participating Entities and Constraints

Work: Binary Relationship

1 employee to 1 restaurant.

1 restaurant to 1 or more employees.

Located: Binary Relationship

1 location to 0 or more restaurants.

1 restaurant to 1 location.

Serve: Binary Relationship

1 restaurant to 1 menu.

1 menu to 1 or more restaurants.

Write: Ternary Relationship

1 restaurant and 1 customer to 0 or 1 reviews.

1 restaurant and 1 review to 1 customer.

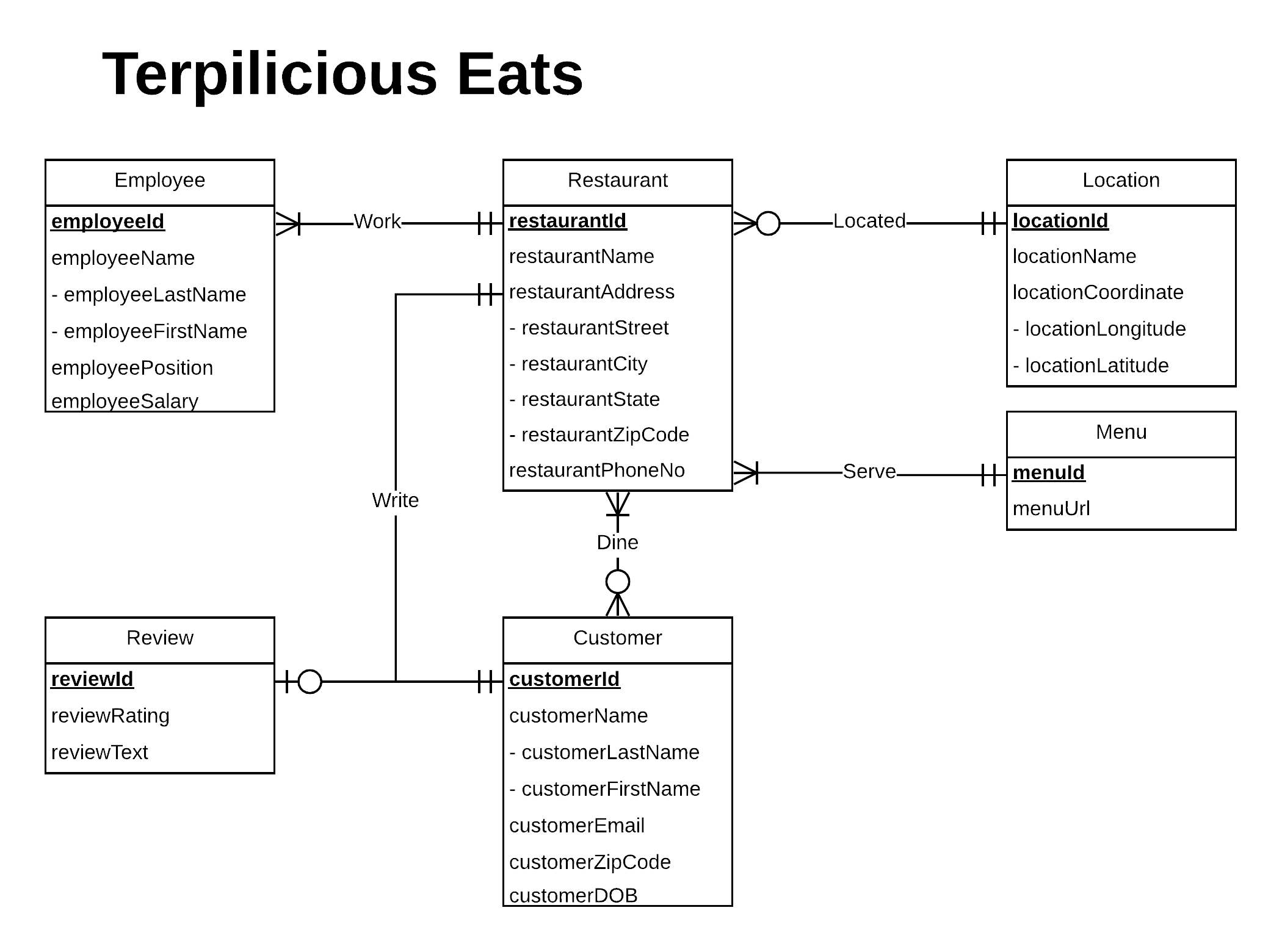
1 customer and 1 review to 1 restaurant.

Dine: Binary Relationship

1 customer to 1 or more restaurants.

1 restaurant to 0 or more customers.

**E-R Diagram:**



**Mission Statement:**

Our mission is to offer an informative online platform that provides useful data and information for both customers and restaurants to help make optimal food decisions around the University of Maryland, College Park.

**Mission Objectives:**

* We strive to provide, maintain and improve useful data and information about restaurants and customer preferences in College Park, MD
* The platform shall help users make restaurant decisions based on ratings and reviews from customers about their restaurant experiences, which can also help restaurants to improve their services and offers
* Increase public awareness and knowledge with the goal of helping people make informed food decisions
* Connect students, faculty, and other residents with local restaurants, and vice versa

**Relations:**

Restaurant(**restaurantId**, restaurantName, restaurantStreet, restaurantCity,  restaurantState, restaurantZipCode, restaurantPhoneNo, *locationId)*

Location(**locationId**, locationName, locationLongitude, locationLatitude)

Menu(**menuId**, menuUrl)

Review(**reviewId**, reviewRating, reviewText)

Customer(**customerId**, customerLastName, customerFirstName, customerEmail, customerZipCode, customerDOB)

Employee(**employeeId**, employeeLastName, employeeFirstName, employeePosition, employeeSalary, *restaurantId*)

Write( ***reviewId***,*customerId, restaurantId*)

Dine(***restaurantId***, ***customerId***)

**Functional Dependency:**

restaurantId 🡪 restaurantName, restaurantStreet, restaurantCity,  restaurantState, restaurantZipCode, restaurantPhoneNo, locationId

locationId 🡪 locationName, locationLongitude, locationLatitude

menuId 🡪 menuUrl

reviewId 🡪 reviewRating, reviewText

customerId 🡪 customerLastName, customerFirstName, customerEmail, customerZipCode, customerDOB

customerEmail 🡪 customerLastName, customerFirstName, customerId, customerZipCode, customerDOB

employeeId 🡪 employeeLastName, employeeFirstName, employeePosition, employeeSalary, restaurantId

reviewId 🡪 restaurantId, customerId

restaurantId, customerId 🡪

**Normalizaton:**

Restaurant(**restaurantId**, restaurantName, restaurantStreet, restaurantCity,  restaurantState, restaurantZipCode, restaurantPhoneNo, *locationId)* = 3NF

Location(**locationId**, locationName, locationLongitude, locationLatitude) = 3NF

Menu(**menuId**, menuUrl) = 3NF

Review(**reviewId**, reviewRating, reviewText) = 3NF

Customer(**customerId**, customerLastName, customerFirstName, customerEmail, customerZipCode, customerDOB) = 3NF

Employee(**employeeId**, employeeLastName, employeeFirstName, employeePosition, employeeSalary, *restaurantId*) = 3NF

Write(***restaurantId***, *reviewId*, *customerId*) = 3NF

Dine(***restaurantId***, ***customerId***) = 3NF

**Business rules:**

[R1] A location cannot be deleted unless there are no longer any restaurants in that location.

[R2] When a location’s information is updated, all of the restaurants in that location change accordingly.

[R3] When a restaurant’s information is deleted or changed, all corresponding employee,review writing, and customer dining information is deleted/updated accordingly.

[R4] When a review is deleted or changed, all corresponding entries in the write table are deleted/updated accordingly.

[R5] When a customer has written a review or dined at a restaurant, their information cannot be deleted from the database.

[R6] When a customer’s information has been changed, their writing and dining information is updated accordingly.

[R7] A menu cannot be deleted unless there are no longer any restaurants using that menu.

[R8] If a menu is changed, then the restaurants corresponding to it will be updated to reflect this change.

**Referential integrity:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Relation | Foreign Key | Base Relation | Primary Key | Bus.  Rule | Constraint: ON DELETE | Bus. Rule | Constraint: ON UPDATE |
| Restaurant | locationId | Location | locationId | R1 | NO ACTION | R2 | CASCADE |
| Restaurant | menuId | Menu | menuId | R7 | NO ACTION | R8 | CASCADE |
| Employee | restaurantId | Restaurant | restaurantId | R3 | CASCADE | R3 | CASCADE |
| Write | restaurantId | Restaurant | restaurantId | R3 | CASCADE | R3 | CASCADE |
| Write | reviewId | Review | reviewId | R4 | CASCADE | R4 | CASCADE |
| Write | customerId | Customer | customerId | R5 | NO ACTION | R6 | CASCADE |
| Dine | restaurantId | Restaurant | restaurantId | R3 | CASCADE | R3 | CASCADE |
| Dine | customerId | Customer | customerId | R5 | NO ACTION | R6 | CASCADE |

**Sample Data:**

Restaurant (**restaurantId (00001)**, restaurantName (Dominos Pizza), restaurantStreet (Baltimore Ave), restaurantCity (College Park), restaurantState (MD), restaurantZipCode (20740)

restaurantPhoneNo (+1 8569545849), ***locationId (00012)***)

Location (**locationId (00012)**, locationName (College Park Shopping Center), locationLongitude (-76.937759), locationLatitude (38.989697))

Menu (**menuID (00004)**, menuUrl (<https://aroythairestaurant.com/menu/>))

Review (**reviewId (00001)**, reviewRating (5), reviewText (Awesome pizza as always!))

Customer (**customerId (00001)**, customerLastName (Smith), customerFirstName (Bob), customerEmail ([bob.smith@rhsmith.umd.edu](mailto:bob.smith@rhsmith.umd.edu)), customerZipCode (20740), customerDOB (1928-07-21))

Employee (**employeeId (00001)**, employeeLastName (Deer), employeeFirstName (John), employeePosition (Chef), employeeSalary ($50,000), ***restaurantId (00001)***)

Write (***restaurantId (00001), reviewId (00001), customerId (00001)***)

Dine (***restaurantId (00001), customerId (00001)***)