CIS 451/551 Final Project

Fall 2019

name(s): James Kang

project title: Uber database

Connection information

port number: 3141
guest account login/password: guest/ guest
database name: mydb

project URL: https://ix.cs.uoregon.edu/~jkang2/CIS451Fproj/finalproj.html

highlights:

CIS 451 Database Processing

Final Project Report

Submitted to:
Prof. Chris Wilson

Author:
<James Kang>

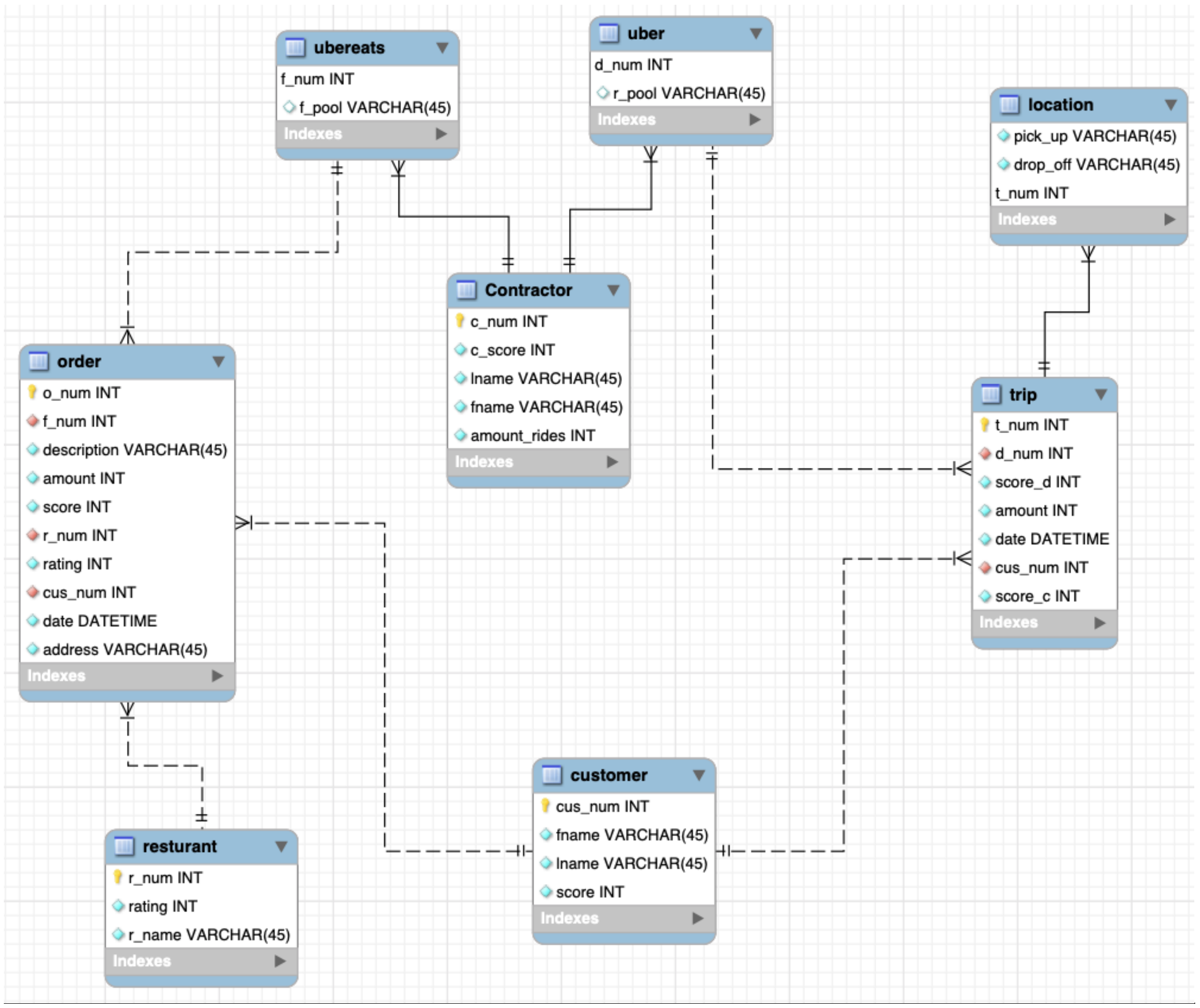
Table of Contents:

Link to the URL: https://ix.cs.uoregon.edu/~jkang2/CIS451Fproj/finalproj.html

Summary

I am creating a database for the company of Uber. In this database the main idea is that a "contractor" is assigned to work either at Uber or Uber eats. Uber is simply a ride giving company and Uber eats is a spinoff of Uber where they deliver food instead of giving rides. In this scenario I will keep a log of every trip made by the contractor that is also connected by each customers. There's also a score rating (out of 5) that will hopefully be implemented in the code as well. There's multiple points of this database, one is to show the customer who the good drivers are. Second is to show the drivers who the good customers are. With so many people being bad driver (i.e. sexual assault happening) and so many riders being bad (i.e. puking and such) this would allow both parties to see the other and potentially decline the service from one another. We also can have things like restaurant scores that the customer could look at as well as each induvial orders and rides that transpired. There could be a lot more implementations with the database however I did not want to go over the limit of 10 (I kept it to 9). And then some of the things implemented could be used from Uber the company to see how much people spend more than a certain amount to maybe target those customers into keep using their services. Lyft (Ubers competitor) gives riders with good ratings a discount code and those who don't use their services often as well. Uber I don't think has any of that in place which is why I think this might help them a lot with the business.

Logical Design



Physical Design

We have 8 total tables; they are contractor, Uber, Uber eats, trip, location, order, restaurant, customer. Here are the descriptions of each table and the elements within the tables.

Contractor: The contractor otherwise viewed as the driver who work with Uber. They aren't called employees because under Ubers term and condition they are individual contractors and so they are not viewed as employees.

- C_num is the contractor's identification number. This is the primary key for the table contractors
- Score is the amount of score that the driver averages per their trip (out of 5.0)
- Lname and fname are last names and first names of the drivers
- Amount_rides is the number of rides that the driver has given in the past (to determine their experience level)

Uber eats: This is the table which determines which diver gets to go next. I don't think I have enough skills and experience to determine how to get this to work. But the main idea here is that each driver or "contractor" is in a queue based on FIFO where they wait base on their ordering. The pool given is the queue in front of the person aka how long the driver has to wait until they can be assigned a task.

- C_num from the contractor this is a foreign key. The Uber eats table does not have a primary key
- Pool this is the queue of people that will be assigned a task before our driver identified as c_num

Uber: This table is really similar to the Uber eats, the only difference here is that it will lead the driver to do a trip instead of a food delivery.

- All the attributes for this class is also identical to the Uber eats table

Orderr: This table records the data that Uber eats brings in. The point here is that we are able to look back at our past spending if you're a customer. Alternatively, the restaurant is also able to access this to see if someone made a mistake and who the mistake is to be blamed on.

- O_num is the primary key which determines what each order will be referred to as.
- C_num is the foreign key brought from the Uber Eats table
- Description is the description of the items that the customer has ordered
- Amount is the amount of money that the customer has paid
- Score is the score given to the driver from the customer (out of 5)
- R_num is the identification number given from the restaurant table. This is a foreign key
- Rating is the rating that the food itself got. This will be given to the restaurant
- Cus_num is the identification number of the customer
- Date is the date of when the order was completed
- Address is the address where the food is being delivered to

Restaurant: This is the table that the restaurants information is stored in.

- R_num this is the primary key in the table restaurants which determines what restaurant we are taking about
- Rating is the rating of the restaurant given by the customers based on their food (out of 5)
- R_name is the restaurant's name

Trip: This is the table where each induvial trips given to the customers via the Uber app is stored in.

- T_num is the identification number given to each trip to determine which trip we are taking about. This is the primary key for the table
- C_num is the number of the contractor that is working on the trip
- Score_d is the score that this trip has gotten from the customer to the driver (out of 5)
- Score_c is the score that this trip has gotten from the driver to the customer (out of 5)

- Amount if the amount of money that was spent for this ride (including tips)
- Date is the date that the ride was given
- Cus_num is the identification number of the customer that rode in this trip. This is a foreign key

Address: This table is an extension of the trips page. This is where the information about the addresses are located.

- Pick_up is the address of where the customer is being picked up
- Drop_off is the address where the customer is being dropped off
- T_num is our foreign key which determines what trip we are referring to

Customer: This table stores the information of the customer.

- Cus_num is the primary key for this table. It determines what customer we are talking about
- Fname is the first name of our customer
- Lname is the last name of our customer
- Score is the score that our customer averages per ride (out of 5)

List of applications

- Given a rating of 1-5 give us all the drivers nearby that are higher or equal to that score (affects the Contractor table)
- Given a rating of 1-5 give us all the riders nearby that are higher or equal to that score (affects the customer table)
- Given a rating of 1-5 give us all the rider that were higher or equal to that score (affects the trip table)
- Given a rating of 1-5 give us all the food delivery orders nearby that are higher or equal to that score (affects the orderr table)
- Given a rating of 1-5 give us all the restaurants nearby that are higher or equal to that score (affects the restaurants table)
- Given an amount of money give us the name of the customer who was in the ride and how much they spent (affects customer and trip table)
- Given an amount of money give us the name of the customer who ordered the food and how much they spent (affects customer and order table)
- Given the Uber ID (1-10) give us a queue of the Uber ID's of drivers who are ahead in the queue (empty means that we are next)
- Given the Uber ID (1-10) give us a queue of the Uber ID's of food deliverers who are ahead in the queue (empty means that we are next)

User's guide

Some of my implementations will be used by driver and some of them could be used for restaurants and Uber themselves. But most of the applications I made will be used by the consumer.

Click on the following links to be directed to each of the applications described by the wording above it.

Contents of Table

I feel that this is what I did in the physical design.

Implementation code

All of my code is available at the site. If you go down to the bottom there should be a Content of the page along with adequate files.

Conclusion

I think the hardest part about this project was connecting my database to ix and the PHP stuff. Coming from the CIS major I was never thought how to do any front end things and I think that's what threw me off the most. I think if I had more time I would have done more advanced joining techniques that would have given more information about each things. Like for example instead of trip with rating giving the customer's name only I would have done. Customers name, drivers name, the address of pick up and drop off. I just didn't really know how to do that with php which is why I did not have crazy joins.