Introduction to Databases

Stage 4 Report:

Demo and Final Report

For Del Delivery Drivers Web Application

By: Morgan Houston

May 03rd, 2019

Introduction:

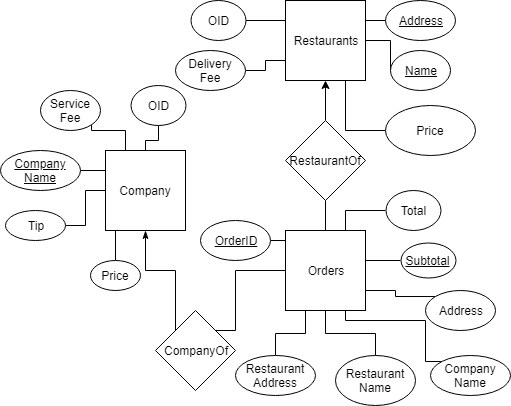
The application I have created compares the total price between different delivery companies. I chose this application to help people see the price difference between each company. I have experience with the delivery business, as I used to work for Door Dash. I always wondered if other companies were cheaper. I use delivery food services sometimes, so my motivation for this project is to see if I can save a few dollars.

Del Delivery Drivers Database is a very useful application. There is no need to sign-in or create an account. Compare prices from different delivery services instantly. My project has a google map element to give a visual view of restaurants in the area. I have added an advanced function that when you select a restaurant, it will automatically insert the restaurant name and address into the submission form. Then all you must enter is your address and the orders subtotal for which you want to compare. This will help determine the delivery fee (if any) as well as the service fee and a %20 tip for Door Dash and Grub Hub orders. Uber Eats only lets you tip with cash. On my website you will also be able to delete and update queries.

Database Details:

I thought of the three main components we needed to design the database. That is an order itself, the Restaurant whom is making the order, and the company who will be delivering the order. From there we have all the information needed to calculate fees to find the total amount for each company.

ER Diagram: The following is an entity relation model for our database.



Relational Schema:

The following relational schema defines our database relations.

|  |  |
| --- | --- |
| **Orders** | |
| PK1 | OrderID |
| FK1 | RN |
| FK2 | RA |
| FK3 | CN |
|  | Address |
| PK2 | Subtotal |
|  | Total |

|  |  |
| --- | --- |
| **Restaurants** | |
| PK1 | Name |
| PK2 | Address |
| FK1 | OrderID |
| FK2 | Price |
|  | Delivery Fee |

|  |  |
| --- | --- |
| **Company** | |
| PK | CompanyName |
| FK1 | OrderID |
| FK2 | Price |
|  | Tip |
|  | Service Fee |

Functional Dependency: If there is an order entity there must be a Restaurant and Company entity as well.

Functionality Details:

Implementation Details:

Experiences:

References:

I have used the following references to help construct my webpage and database:

Awa Melvine: <https://www.youtube.com/watch?v=mjVuBlwXASo>