Henry Simon, Vraig De Leon

Professor Ratan Dey

CS-UY 3083 Intro to Databases

March 26, 2024

Project Description

There are several laundry facilities (**Laundry_Facility)** each facility has a unique code, a name, and address, number of washing machines.

A delivery truck (**Delivery_Truck**) depends on the facility that it services, has a unique identification number within that facility, has the order capacity of the truck, has a manufacturing company of that truck, has a model number, has manufacturing date, and has an age of the truck. Each of the trucks may need to go through zero or many maintenance procedures and hence won't be available for routes during the maintenance periods. Each maintenance procedure of a particular truck has a start date and time and has an end date and time.

Each facility fulfills delivery routes (**Delivery_Route**), which depends on the facility servicing the route, has a route identification number, has a parent facility, has pickup dates and times, has drop-off date and times, has pickup and dropoff locations, and has the identification number of the delivery truck for the route. Each delivery route is identifiable using route id number and together with the facility it is fulfilled by.

An order (**Order**) can be purchased by a customer, and customer will provide email address, first name, last name, date of birth while purchasing the ticket. Customer needs to pay the calculated order price (based on the number of pounds and price per pound of the order) and need to provide payment information (including card type - credit/debit, card number, name on card, expiration date). System will also record the purchase date and time. Each order will have an order ID number which is unique in this System. A customer can buy multiple orders (i.e. buy orders for different services, different addresses) but customer needs to provide first name, last name, and date of birth for each order separately. Each order will have an option for dropoff/pickup with a doorman or direct-to-door delivery

Anyone (including users not signed in) can see prices per pound based on the delivery address and service. Customers with an order can see the status (processing, out for delivery, delivered) of the flight based on an airline and flight number combination and arrival or departure time. There are two types of users for this system: Customer, and Airline Staff.

Customer: Each Customer has a name (composite attribute consisting of first name and last name), username email address, password, default address (composite attribute consisting of building number, street name, Apartment number, city, state, zip code), and date_of_birth. Each Customer's email address is unique, and they will sign into the system using their email address and password.

Customers can purchase an order if it meets the minimum weight requirement. Order price of a will be determined based on two factors – price per pound as set by the laundry company and the weight (pounds) of the order. If the minimum order is not met, the order price will be updated to reflect a minimum order surcharge. Customer can order using either credit card or debit card. We want to store card information (card number and expiration date and name on the card and security code).

Laundry Staff: Each Laundry Staff has a unique username, a password, a first name, a last name, a date of birth, may have more than one phone number, must have at least one email address. Laundry Staff only works for one facility. Laundry Staff will set order statuses in the system.

They will also be able to see all processing, future, and previous orders for the facility that they work for, as well as a list of customers for the orders.

Part 2

1. Following the techniques we studied, derive a relational schema diagram from the Part 1's ER diagram. Remember to underline primary keys and use arrows from the referencing schema to the referenced schema to indicate foreign key constraints.

2. Write and execute SQL CREATE TABLE statements to create the tables. Choose reasonable types for the attributes.

3. Write and execute INSERT statements to insert data representing one airline's air ticket reservation system. As for example, you can insert data in the appropriate tables as follows or you can insert data for another airline or your own make up airline:

a. One Facility named "Laundromat 1".

b. At least Three laundry attendants named "Henry", “Daniel” and "Vraig".

c. Insert at least three customers with appropriate names and other attributes.

d. Insert at least delivery truck.

f. Insert several routes and appropriate attributes.

g. Insert some orders for corresponding delivery routes and insert some purchase records (customers purchased some orders).

4. Write SQL queries for executing following queries and show the results in your file (SQL query and corresponding answers):

a. Show all the orders out for delivery in the system.

b. Show all of the customers currently awaiting the orders out for delivery in the system.

c. Show the addresses of customers who bought the orders.

d. Show all the orders with at least 15 pounds in them.

Part 3

1. View Order Prices: All users, whether logged in or not, can see the price of an order given they enter a delivery address. They can also see the estimated delivery time and the pickup options.

2. Register: 2 types of user registrations (Customer, and Laundry Staff) option via forms as mentioned in the part 1 of the project.

3. Login: 2 types of user login (Customer, and Laundry Staff). Users enters their username (email address will be used as username for customer) and password. Customers can checkout as a guest if they prefer, and will be prompted to save their information to an account in the process if they prefer. User stays logged in until they manually log out or delete the app.

Customer Use Cases

1. **Search Addresses for Potential Orders:** Provide user with address search and pickup/delivery time estimates for location using MapBox API

2. **View Order in Progress:** Customer views order in progress with MapBox API. Only displays driver tracking updates if customer selects option for meeting driver at door

3. **Customize Order:** Customer customizes detergent, wash type, pickup/dropoff details and adds any special requests via forms

4. **Purchase Order:** Customer confirms/selects payment method and confirms items to be purchased. Likely will be through stripe API or similar.

5. **Cancel Order:** Only applicable before order is picked up. Customer can cancel any time before order is picked up.

6. **Edit Order:** Customer can change detergent type, delivery time, etc. These are dependent on the status of the order (must change detergent type before clothes are processed, etc.)

7. **View Previous Orders:** Customer can view previous orders and reorder them. Shows average order size and price.

8. **Change Information:** Customer can logout or change personal information

9. **Logout:** Guest session replaces user instance and logout message is displayed

Laundry Staff Use Cases

1. **View Delivery Route:** Shows delivery using MapBox API with all stops plotted. Shows timestamps (estimated and actual) for deliveries

2. **Create New Route:** Staff enters all processed orders. Optimized delivery algorithm (manipulated from open source github repository) outputs live mapbox

3. **Update item Status:** Staff updates order status. These updates are reflected in customer-side of frontend.

4. **Add Delivery Truck:** Staff adds new vehicle. Requires authorization

5. **General Metric View:** Staff can see all important metrics. These include the average turnaround time of orders, number of orders in each status designation, total orders, total revenue, average weight/price of orders, most frequent preferences, number of machines in use at parent location
6. **Logout:** Guest session replaces user instance and logout message is displayed