

MySQL & Python Ride Share App

CPSC408

Suppose that a new start-up is trying to create a rideshare app and hires you to design their database. Think about the main information that the app must keep track of and create some deliverables to show your client as you go.

You must create:

1. An ER diagram
2. A fully filled out schema
3. A database on MySQL locally
4. Sample data within your database to test
5. A simple interactive python program.

For the interactive part:

1. You will ask the user if they are a new user or a returning user.
2. If the user is a new user, let them create either a rider or driver account (see below for requirements on each account)
3. If the user is a returning user, they will provide their user ID and you will determine whether they are a driver or a rider (a user cannot be both, for simplicity)
4. If the user is a driver, you will give them the following options:
 - a. View Rating: This will show the driver their current rating
 - b. View Rides: This will show the driver the list of all rides they have driven for
 - c. Activate/Deactivate Driver Mode: This updates a flag on their record that they are able to drive and are now accepting ride requests
5. If the user is a rider, you will give them the following options:
 - a. View Rides: This will show the rider the list of all rides they have taken
 - b. Find a driver: Match the rider with a driver that has their driver mode activated
 - i. The rider will then provide the following info:
 1. Pick up location
 2. Drop off location
 - ii. Then, you will create a ride and provide the rider with a ride ID
 - iii. You will then send the rider back to their options menu
 - c. Rate my driver:
 - i. You will look up the rider's last ride and get the driver's ID
 - ii. You will then print this information to the user and ask if it is the correct ride
 - iii. If it is not the correct ride, you will have them enter the rideID of the ride they want to rate and confirm that information.
 - iv. Then, calculate the driver's new rating by taking their current rating + their new rating and dividing by 2.

Whatever database schema you think can accomplish the above use case is allowed.

ENSURE YOU PLAN USING THE CORRECT STEPS.

1. Create you ER Diagram
2. Assign relationships and then determine their cardinality
3. Use our slides on referential integrity and foreign keys to ensure you translate those relationships correctly into a schema
4. Make sure you check that your database schema is NORMALIZED TO THE 3RD NORMAL FORM
5. Then, go on to create your database and schema, and the app around it similar to how we built the playlist app.

You may come to me at any time asking to confirm your plan looks good before you implement it. If you implement a non-normalized database I WILL make you redo it to get full points.