Ivy Truong

4/26/20

CSCI 3761

Homework 2

Part 2

Relational Schema description

* Additional modifications and assumptions from ER diagram
  + For the relationship between car and rental entities, they have a 1 to many relationship instead of a many to many relationship. When renting out a car, a customer can only choose to have a daily rental or a weekly rental.
  + Assume each car type has the same daily and weekly rental rates.
    - In the ER diagram, the attributes for daily and weekly rates were added to each subclass in the car entity rather than having the attributes in the car entity.
  + Changed rental entity’s relationship with the customer entity. Rental no longer depends on customer. I changed this because the database should keep track of scheduled and active rentals, meaning that not all cars will be rented out by a customer. This will allow the ID number in rentals to be NULL when a rental is active.
    - If the ID number attribute is not NULL and the Available\_now attribute is false, then that means the car has been rented.
* Customer relation
  + 4 attributes: ID number, First initial, Last name, Phone number
  + ID number is the primary key
* Car relation
  + 5 attributes: Vehicle ID, Car type, Model, Year, ID number
  + Vehicle ID is the primary key
  + ID number is a foreign key that points to the primary key ID number in the customer relation
    - ID number can be NULL if the car has not been rented by a customer
  + Car type (type) is a foreign key that points to the primary key type in the car type relation
* Car type relation
  + The main reason I created this relation is because the daily rates and the weekly rates for each type of car is constant. For example, all compact cars have a daily rate of $11.80 and a weekly rate of $59.00. Instead of combining the car type and the daily and weekly rate attributes with the car relation, it was easier to create a separate relation for car types and have records of each type.
  + 3 attributes: Car type (type), daily rates, weekly rates
  + Car type is the primary key
* Rentals relation
  + 10 attributes: ID number, Vehicle ID, Available Now, Periods Available, Amount Due, Rtype, Number of days, Number of weeks, start date, end date
  + Vehicle ID is the primary key
  + Rtype will either be ‘daily’ or ‘weekly’, it will be NULL if the car has not been rented out yet (if Available now is true)
  + Available Now will be a Boolean variable that will be true if no customer has rented the car and false if a customer is renting the car
    - Depends on ID number, which means that ID number can be NULL
  + Periods Available will be a variable that will store the number of minutes after the last rental before the car can be rented again
    - Changed variable to waitTime
    - Problem is, how will the database simulate the waitTime before making the available now attribute true? (do in PHP program)
    - **NOTE: Did not include this requirement in program, did not know how to interpret this requirement**
  + ID number can be NULL because it is not a primary or unique key for this relation
    - It is NULL if a car has not been rented by a customer, therefore a rental payment can’t be established
  + If the rtype is either daily or weekly, then number of days or number of weeks would be filled, so number of days can be NULL or number of weeks can be NULL
* Assumptions considered in database
  + In the rentals relation, a new record for the rentals relation will only have the Vehicle\_ID initialized because it has just added a car to the system
  + In the rentals relation, a record will always update their attributes when a new scheduled rental for the car is made (if it is available). Once the car is returned, all the attributes except the Vehicle ID attribute are reset to the default values to show that the rental is free. Therefore, the rentals relation in the database will not keep track of any of the previous scheduled rentals from customers.
    - Transaction 3 will calculate the Amount\_due while transaction 4 will retrieve the Amount\_due.
  + All customers in the database must have rented at least one car, if they are no longer renting a car, they will be removed from the database
    - Customers can rent multiple cars

Website to implement mysql using php: <https://www.w3schools.com/PHP/php_ref_mysqli.asp>