CSE 3330 Project 2 Part 1

Car rental Database

Shubham Arya, Yugal Shrestha, Bishwamitra Sapkota

The University of Texas at Arlington

# Introduction

In this project, we design and implement a database for storing data and keeping track of information about a car rental company. In this project, we will first design the ER schema diagram based on the requirements presented to us in the project description. Then we will convert the ER schema diagram to a relational database schema by using a drawing tool. This will be the base of our project for the next parts.

# Description

The Car Rental Database keeps track of all the customers. A customer has a unique identification number, their name and a phone number. This database also keeps track of cars available for rental and are categorized into 6 types based on their size. Each type of car has its own rental rate and weekly rate and there is only one rental location. A car has three attributes namely vehicle identification number, model, and the year in which it was manufactured.

This database will only keep track of cars currently rented or scheduled to be rented. There are types of rental: daily and weekly. The daily rental will include information about car, customer, number of days, start date and return date. The weekly rental will include information about the car, customer, number of weeks, start date and return date. The rental database will also store amount due information which is calculated from the rental and weekly rates for a specific type of car. Lastly, the database will also keep track of which cars are available for rental during which period.

In this Car Rental Database, there are 4 entities that we have designed based on the requirements. The entities are CUSTOMER, CAR, RATE and RENTAL.

**CUSTOMER**

This keeps track of the customers coming to rent the cars. A customer has three attributes: IdNo, Name and Phone.

* IdNo- This is an identification number which is generated by the system. It is of type Int and it is a unique number as stated by the requirements.
* Name- This is a string consisting of single initial and last name of the customer.
* Phone- This is a string of 12 characters that stores the phone number of the customer. 10 of the 12 characters are for the numbers. The other 2 characters are for the ‘-’ in between the numbers.

**CAR**

This stores the data of all the cars that the car rental company possess. A car has four attributes: VehicleID, Model, Year and Type.

* VehicleID- This is a unique number for each car to identify the car. It is of type Int and it is a unique number as stated by the requirements.
* Model- This is the model name of the car of type string.
* Year- This is the year in which the car was manufactured and has integer type.
* Type- A car can be categorized into 6 types. These types are compact, medium, large, SUV, truck and van. This has a data type of string. This is a foreign key in the CAR entity.

**RATE**

This has the rate of each type of car. It has three attributes: Type, DailyRate and weeklyRate.

* Type- This is a primary key in this entity and has a data type of string. It holds the value of 6 different types of cars discussed above.
* DailyRate- This is the daily rate to rent a car depending on the type. It has a data type of float since rate can have decimal points to it.
* weeklyRate- This is the weekly rate to a rent a car depending on the type of car. It has a data type of float since rate can have decimal digits at the end.

**RENTAL**

The database will keep track of the current rentals and the scheduled rentals for each car. A person can rent a car either daily or weekly. The attributes we have constructed for this database are IdNo from CUSTOMERS, VehicleID from CARs, DW, Time, StartDate and AmountDue.

* IdNo- This is the IdNo from CUSTOMERs. This is a foreign key in the RENTAL entity.
* VehicleID- This is the VehicleID from CARs. This is a foreign key in the RENTAL entity.
* DW- DW (Daily/Weekly) is of type string which contains either ‘daily’ or ‘weekly’. This is to check if a rental is a daily type or weekly type. We use this attribute to calculate values of other attributes.
* Time- This hold an integer type with a number. If the DW type is ‘daily’, then the time would be in number of days. If the DW type is ‘weekly’ then the time would be in number of weeks.
* StartDate- This is of date type. It is to tell when the customer is renting a car for. Using the Time attribute and the DW attribute.
* ReturnDate- We can calculate the return date of the car by using the StartDate, DW and Time. This is of type date and is a derived attribute.
* AmountDue- This the amount due in (US Dollars) by the customer for the car they rented. It is of type float since it can have decimal points. This is calculated by multiplying the Rate on the DW type, so this is a derived attribute.

# Schema Diagram of Database

**CUSTOMER**

CUSTOMER (IdNo, Name, Phone)

**CAR**

CAR (VehicleID, Model, Year, Type)

**RATE**

RATE (Type, DailyRate, weeklyRate)

**RENTAL**RENTAL (IdNo, VehicleID, DW, Time, StartDate, ReturnDate, AmountDue)

# ER Schema Diagram

**Diagram

Description automatically generated**

**CUSTOMER** RENT\_FROM **RENTAL.** This is a many to one relation. A customer can only rent once from a rental. The rental company may have at least one or more customers to rent to.

**RENTAL** LEND **CAR.** The rental company may have 1 or more cars to rent. These cars can only belong to one rental company (assuming that the company owns the cars). This is a many to one relationship.

**CAR** HAS\_PRICE **RATE**. A car can only have one rate. But a type of rate can have at least 1 car associated with it. This a many to one relationship.

In the RENTAL entity, we have an attribute called DW which stands for Daily or Weekly, which tells us if the tuple is a rental weekly or a daily weekly. This is done to avoid 2 separate entity that would store the same data and make some values redundant. Since we are making use of DW, we have an attribute called, which depending on DW could either mean number of days or number of weeks.

# Relational Database Schema

**CUSTOMER**

|  |  |  |
| --- | --- | --- |
| IdNo | Name | Phone |

**CAR**

|  |  |  |  |
| --- | --- | --- | --- |
| VehicleID | Model | Year | Type |

**RENTAL**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| IdNo | VehicleID | DW | Time | AmountDue | StartDate | ReturnDate |

**RATE**

|  |  |  |
| --- | --- | --- |
| Type | DailyRate | weeklyRate |

In RENTAL, IdNo and VehicleID are foreign keys who get their values from the IdNo and VehicleID attribute from CUSTOMER entity and CAR entity respectively, both of which are primary keys. In the RENTAL table, IdNo and VehicleID together form the primary key. This is because the rest of the attributes in the RENTAL entity are dependent on these 2 values together.

In CAR, Type is also a foreign key. It gets its value from the Type attribute in Rate which is also used by RENTAL by accessing it through VehicleID.

Assumption: Here we are assuming that the database doesn’t store the past information of the cars rented by the customers in RENTAL. The requirement only states that the database will only keep track of the current and the scheduled RENTAL of each CAR. A customer can also not rent two cars at a time or schedule a car before their rental period for the first car is not over. Hence, RENTAL may have two cars with the same VehicleID, but it cannot have the same customer with same IdNo.

**HONOR CODE**

**I pledge, on my honor, to uphold UT Arlington's tradition of academic integrity, a tradition that values hard work and honest effort in the pursuit of academic excellence.**

**I promise that I will submit only work that I personally create or that I contribute to group collaborations, and I will appropriately reference any work from other sources. I will follow the highest standards of integrity and uphold the spirit of the Honor Code.**

**Signature:** Shubham Arya, Yugal Shrestha, Bishwamitra Sapkota