CSE-3330

Database & File Structures

Project 2-1

Fall 2020

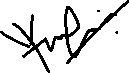
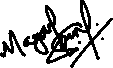
Prof. Guizani

**Team Members:**

* **Yunika Upadhayaya ♕ (Team Leader), Student ID: 1001631183, Section: 003**
* Created EER schema diagram for the Car Rental database.
* **Pratik Mahato, Student ID: 1001661375, Section: 004**
* Created docx file that includes necessary explanations for the EER diagram and Relational Database schema.
* **Suman Thapa Magar, Student ID: 1001643016, Section: 003**
* Created Relational Database Schema for the Car Rental Database.

**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.



*Pratik Mahato Yunika Upadhayaya Suman Thapa Magar*



Contents

[Introduction and Overview 3](#_Toc55302024)

[Mini-World 3](#_Toc55302025)

[Entity Relation Diagram 4](#_Toc55302026)

[Schema Diagram 4](#_Toc55302027)

[Explanation 5](#_Toc55302028)

[Cardinality Relationships 5](#_Toc55302029)

[Relational Database Schema 6](#_Toc55302030)

[Diagram 6](#_Toc55302031)

[Explanation 7](#_Toc55302032)

[Probable Missing Entities 7](#_Toc55302033)

[Assumptions & Constraints 8](#_Toc55302034)

[References 8](#_Toc55302035)

# Introduction and Overview

A relational database application system is to be designed and implemented for keeping track of information about a car rental company. For this Phase 1, relationships between necessary entities is tracked with the help of mini-world constraints and assumptions. These relationships are shown through Entity-Relation Diagram and Relational Database Schema which was made with the help of an online app for conceptual database design tool - **Creately.** After the effective implementation of these relationships in the Schema, several queries can be made to obtain reports or information as needed by the company.

# Mini-World

Through the requirement and analysis phase, the mini world for the Car Rental Company includes the data of real-world entities like:

CUSTOMER, CAR, VEHICLE\_TYPE, DAILY\_RENTAL, WEEKLY\_RENTAL

* The rental company has only one rental location to allow CUSTOMER(s) to rent one or many cars. Depending on whether CUSTOMER rents for days or weeks, it is of 2 types : DAILY\_RENTAL & WEEKLY\_RENTAL.
* The rental company has many CAR(s) for the CUSTOMER(s) to rent. Each car consists of the model and year that it was manufactured along with unique Vehicle ID. They also have status attribute of whether they are available for rent or not.
* CUSTOMER have option to choose CAR of certain VEHICLE\_TYPE where the same type of CAR(s) has fixed weekly rate and daily rate charges when they are chosen for rent.
* Each CUSTOMER has their name, phone no., unique Identification Number (IdNo) and unique VehicleID. The rental database will especially keep track of CUSTOMER’s IdNo and Vehicle ID so that it could be updated immediately when the customer rents or returns a car(s).

# Entity Relation Diagram

### Schema Diagram

Diagram

Description automatically generated

### Explanation

As shown in ER Schema Diagram, following are the entity types with their respective attributes:

* CUSTOMER with attributes Name, Phone and IdNo. The attribute ***IdNo*** is the primary key which is unique to each customer. Customer will be using car rental system to rent a car.
* CAR entity will have list of cars in rental system. Each car will have with attributes like Model, Year, Status and VehicleID. The attribute Status represents if the chosen car is available to be rent or not. Each car has unique ***VehicleID***, hence it is its primary key.
* VEHICLE\_TYPE entity with attributes DailyRate, WeeklyRate and TypeName. ***TypeName*** is the primary key.
* DAILY\_RENTAL entity allows customer to rent a car in basis of days. It has attributes IdNo, VehicleID, NoOfDays, StartDate, ReturnDate, AmountDue (derived). Both primary keys ***IdNo and VehicleID*** together forms composite key.
* WEEKLY\_RENTAL allows customer to rent a car in basis of weeks. It has attributes IdNo, VehicleID, NoOfDays, StartDate, ReturnDate, AmountDue (derived). Both primary keys ***IdNo and VehicleID*** together forms composite key.

### Cardinality Relationships

There are possible 6 relationships as shown below:

* Between the entity types CUSTOMER and CAR, there is a relationship of **1:N**. That is, one customer can *rent* many cars while one car can only be rented by one customer at a time.
* CUSTOMER has relationship of **N:1** with both DAILY\_RENTAL and WEEKLY\_RENTAL. As per the requirement, there is only one rental location. It depends on customer’s *choice* to rent a car weekly or daily.
* Between the CAR and VEHICLE\_TYPE, there is a relationship of N:1. Many cars can *have* same vehicle types like SUV, Truck… and same types have same daily rates and weekly rates for rent. But a car can only have one type.
* CAR has relationship of 1:N with both DAILY\_RENTAL and WEEKLY\_RENTAL. These rentals *have* many cars to rent but these cars only have one rental location.

# Relational Database Schema

### Diagram

*Graphical user interface, timeline

Description automatically generated*

### Explanation

With the help of relational database schema, necessary attributes were added so that queries could be made for the database system successfully.

* Entity type CUSTOMER has foreign key VehicleID which references primary key Vehicle ID of entity type CAR. This is added to know which customer has rent the specific car. Initially it is set to NULL if the customer has not rent a car yet.
* Entity type CAR has a foreign key Type\_name which references primary key of entity type VEHICLE\_TYPE.
* Entity type WEEKLY\_RENTAL has two foreign keys VehicleID and IdNo. which references primary keys of entity type CUSTOMER and CAR, respectively. This is to keep track of customer and car details when a car is rent.
* Entity type DAILY\_RENTAL has two foreign keys VehicleID and IdNo. which references primary keys of entity type CUSTOMER and CAR, respectively. This is to keep track of customer and car details when a car is rent.

Attributes of VEHICLE\_TYPE have fixed 6 rows. Since, there are only 6 car types (as attribute TypeName) : SUV, Truck, Medium, Large, Compact, Van. And each type has fixed DailyRate and WeeklyRate.

# Probable Missing Entities

* Billing -> Attributes: Bill ID, Bill Date, Bill Status, Tax, Late Fee, Amount Due, Total Amount.

Billing entity represents overall final transaction, beginning from car being rent to being returned.

* Car Rental Insurance -> Attributes: Insurance Code, Name, Cost per Day.

Car Rental Insurance represents the insurance of the car. The customer is also liable to pay for insurance cost per day.

* Late Penalty -> Late Fee, IdNo, VehicleID

This entity represents the extra cost payable to rental company if customer returns car past due of the estimated returned date.

# Assumptions & Constraints

Following assumptions and constraints were made for the min-world entities:

* Each booking is associated with only one car reservation at a time.
* Car available in the data base system is present only in one location.
* Not all cars rented will have billing associated because of the cancelled bookings.
* Car rental insurance may or may not be there for the car because customer might have their own insurance.

# References

Elmasri, R., & Navathe, S. (2007). Fundamentals of database systems. Boston: Pearson/Addison Wesley.