Course Title: Database

Car2Go

Instructor: Jaina Sheth

Team Members: Mubeen Khan, Hussain Amin, Sayem Shah, Maxym Galenko

Due date: Tuesday, May 10th, 2022

Table of Contents

[Introduction 2](#_Toc103116672)

[Scenario description 2](#_Toc103116673)

[Description: 2](#_Toc103116674)

[Business Rules and Assumptions: 2](#_Toc103116675)

[Conceptual Design of the Database 3](#_Toc103116676)

[Logical Database Schema 4](#_Toc103116677)

[Functional Dependencies: 4](#_Toc103116678)

[Database Normalization: 5](#_Toc103116679)

[Database Tables: Attributes and Constraints 6](#_Toc103116680)

[SNAPSHOTS: 7](#_Toc103116681)

[Challenges and Suggestions for improvements 9](#_Toc103116682)

[Challenges: 9](#_Toc103116683)

[Suggestions for improvements: 9](#_Toc103116684)

[Conclusions and Future Work 9](#_Toc103116685)

[Conclusions: 9](#_Toc103116686)

[Future Work: 9](#_Toc103116687)

# Introduction

The scenario we have selected consist of a car rental company called Car2Go. It offers different car models, colors and “year made” to customers in different company locations. The objective of this project is to create a Database that will follow the instructions given to our scenario. For the team roles, we haven’t really given a role or a big task to anyone in particular. We all worked equally at each step of the project and helped each other out.

# Scenario description

## Description:

The company Car2Go is a car rental company that has several locations and offers a multitude of classes such as subcompact, compact, sedan and luxury. Each car has a different make, model, year made and color. Also, they each have a unique identification number and a unique license plate. When a car is rented, the company keeps track of the mileage, tank, and the date (before and after a car is rented). The rented car can be returned to a different location. A customer can only rent one car at a time and request a specific class. For every customer, the company keeps record of their full name, mailing address, zero or more phone numbers, zero or more email addresses and the driver’s license number (unique for each customer). If the company does not have a vehicle of the class requested, the customer can get a higher class for the same price (free upgrade). All cars of the same class are priced the same. The drop-off charge is calculated by the car’s class, the duration in days (minimum 1) and if there is any weekly promotion. In certain weeks of the year, the company has a promotion (usually 50%, not always) that only affects a single class of car at a time.

## Business Rules and Assumptions:

1. Each car has a unique license plate
2. Customer can only rent one car at a time
3. If the car requested by the customer is not available, they can get a car of higher class at the same price as the car requested
4. Company keeps track of the rented cars
5. Company keeps record of the customer
6. All cars in the same class have the same price
7. Drop-off charge can be found with the price of the car, duration in days, and promotion, if there are any.
8. For every car rented, the company keeps the odometer reading before it is rented and after it is returned
9. The cars rented in a particular location may be returned to a different location
10. When a car is being returned the company records the tank Volume and indicate whether the tank is empty, quarter full, half full, three quarters full, or full.
11. The company also keeps track of the day a car was rented and returned

# Conceptual Design of the Database

Diagram, schematic

Description automatically generated

# Logical Database Schema

* **Email** **(** driver\_license\_no, email\_address **)**
* **Phone\_Number (** driver\_license\_no, country\_code, area\_code, local\_number **)**
* **Address (** address\_ID, city, province, postal\_code, building\_number, street\_name **)**
* **Customer** **(** driver\_license\_no, address\_ID, first\_name, last\_name, apartment\_number, country **)**
* **Company\_Location (** location\_ID, address\_ID **)**
* **Car** **(** license\_plate, class\_description, location\_ID, make, model, year\_made, color **)**
* **Car\_Class (** class\_description, car\_class\_price **)**
* **Renting\_Detail (** return\_ID, day\_rented, day\_returned, license\_plate, driver\_license\_no, odometer\_before, odometer\_after, tank\_volume, num\_days\_rented, pickup\_company\_location, dropoff\_company\_location **)**
* **Promotion (** class\_description , promotion\_ID, discount, start\_date, end\_date **)**
* **Payment** **(** billing\_ID, class\_description, return\_ID, promotion\_ID, total\_amount **)**

|  |  |  |
| --- | --- | --- |
| **FOREIGN KEYS** | **REFERENCING RELATION** | **REFERENCED RELATION** |
| driver\_license\_no | EMAIL | CUSTOMER |
| driver\_license\_no | PHONE\_NUMBER | CUSTOMER |
| Address\_ID | CUSTOMER | ADDRESS |
| Address\_ID | COMPANY\_LOCATION | ADDRESS |
| Location\_ID | CAR | COMPANY\_LOCATION |
| class\_description | CAR | CAR\_CLASS |
| license\_plate | RENTING\_DETAIL | CAR |
| driver\_license\_no | RENTING\_DETAIL | CUSTOMER |
| pickup\_company\_location | RENTING\_DETAIL | COMPANY\_LOCATION |
| dropoff\_company\_location | RENTING\_DETAIL | COMPANY\_LOCATION |
| class\_description | PROMOTION | CAR\_CLASS |
| class\_description | PAYMENT | CAR\_CLASS |
| return\_id | PAYMENT | RENTING\_DETAIL |

## Functional Dependencies:

* Address\_ID -> (city, province, postal\_code, building\_number, street\_name)
* Driver\_liscence\_no -> (address\_ID, first\_name, last\_name, apartment\_number, country)
* Location\_ID -> (address\_ID)
* License\_plate -> (class\_description, location\_ID, make, model, year\_made, color)
* Class\_description -> (car\_class\_price)
* Return\_ID -> (day\_rented, day\_returned, license\_plate, driver\_license\_no, odometer\_before, odometer\_after, tank\_volume, num\_days\_rented, pickup\_company\_location, dropoff\_company\_location)
* class\_description , promotion\_ID -> (discount, start\_date, end\_date)
* billing\_ID -> (class\_description, return\_ID, promotion\_ID, total\_amount)

## Database Normalization:

For the Normalization, we removed the columns “Email” and “Phone\_Number” from Customer table because we would have more than one value for some attributes. Instead, we created two separate tables called “Email” and “Phone\_Number”. This change allowed us to successfully implement the First Normal Form (1NF) of our ER Diagram. For the Second Normal Form, the table Address was added to the ER Diagram in order to identify the addresses for each record in the tables Customer and Company\_Location.

# Database Tables: Attributes and Constraints

ADDRESS TABLE:

|  |
| --- |
| **address\_ID,** |
| PRIMARY KEY |
| NOT NULL |

CUSTOMER TABLE:

|  |  |
| --- | --- |
| **DRIVER\_LICENSE\_NO** | **ADDRESS\_ID** |
| PRIMARY KEY | NOT NULL |
| NOT NULL | FOREIGN KEY |

EMAIL TABLE:

|  |  |
| --- | --- |
| **DRIVER\_LICENSE\_NO** | **EMAIL\_ADDRESS** |
| PRIMARY KEY | PRIMARY KEY |
| NOT NULL |  |
| FOREIGN KEY |  |

PHONE\_NUMBER TABLE:

|  |  |  |  |
| --- | --- | --- | --- |
| **DRIVER\_LICENSE\_NO** | **COUNTRY\_CODE** | **AREA\_CODE** | **LOCAL\_NUMBER** |
| PRIMARY KEY | PRIMARY KEY | PRIMARY KEY | PRIMARY KEY |
| FOREIGN KEY |  |  |  |

COMPANY\_LOCATION TABLE:

|  |  |
| --- | --- |
| **LOCATION\_ID** | **ADDRESS\_ID** |
| PRIMARY KEY | NOT NULL |
| NOT NULL | FOREIGN KEY |

CAR\_CLASS TABLE:

|  |
| --- |
| **CLASS\_DESCRIPTION** |
| PRIMARY KEY |
| NOT NULL |

CAR TABLE:

|  |  |  |
| --- | --- | --- |
| **LICENSE\_PLATE** | **CLASS\_DESCRIPTION** | **LOCATION\_ID** |
| PRIMARY KEY | NOT NULL | NOT NULL |
| FOREIGN KEY | FOREIGN KEY | FOREIGN KEY |

PROMOTION TABLE:

|  |  |
| --- | --- |
| **PROMOTION\_ID** | **CLASS\_DESCRIPTION** |
| PRIMARY KEY | PRIMARY KEY |
|  | NOT NULL |

RENTING\_DETAIL TABLE:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **RETURN\_ID** | **DAY\_RENTED** | **DAY\_RETURNED** | **LICENSE\_PLATE** | **DRIVER\_LICENSE** | **PICKOFF** | **DROPOFF** |
| PRIMARY K | PRIMARY K | PRIMARY K | FOREIGN K | FOREIGN K | FOREIGN K | FOREIGN K |
| NOT NULL | NOT NULL | NOT NULL | NOT NULL | NOT NULL | NOT NULL | NOT NULL |

PAYMENT TABLE:

|  |  |  |  |
| --- | --- | --- | --- |
| **BILLING\_ID** | **CLASS\_DESCRIPTION** | **PROMOTION\_ID** | **RETURN\_ID** |
| UNIQUE | FOREIGN KEY | CHECK | CHECK |
|  | NOT NULL |  | NOT NULL |

## SNAPSHOTS:

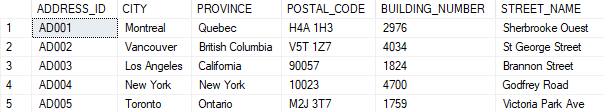


Figure : Address Table

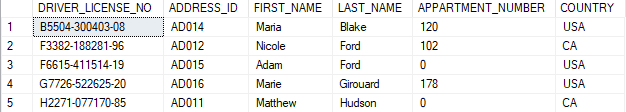


Figure : Customer Table

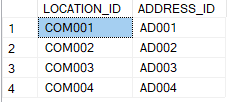


Figure : Company\_Location Table

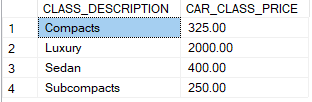


Figure : Car\_Class Table



Figure : Car Table

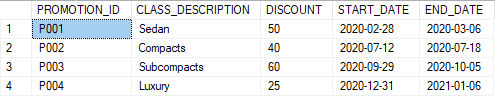


Figure : Promotion Table

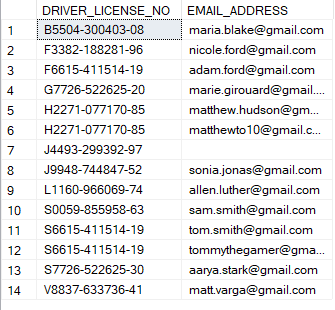


Figure : Email Table

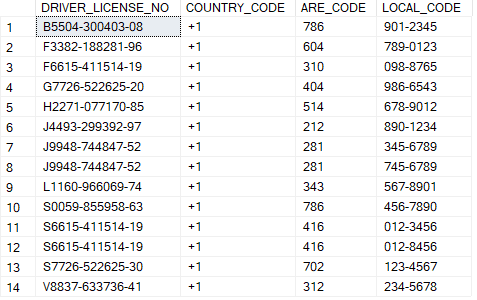


Figure : Phone\_Number Table

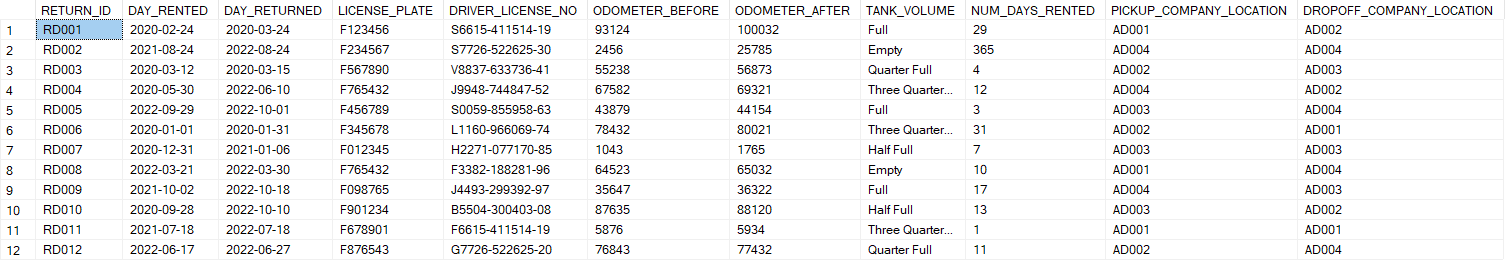


Figure : Renting\_Detail Table

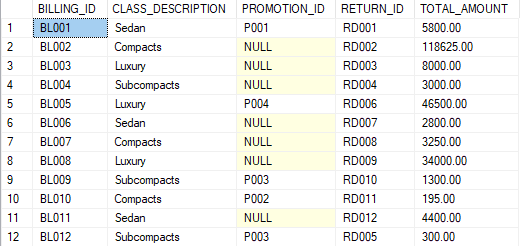


Figure : Payment Table

# Challenges and Suggestions for improvements

## Challenges:

* ER Diagram
* Connecting the tables
* Finding the primary keys
* Creating functions
* Creating attributes (constraint errors)
* Having a good number of tables
* Normalization (2NF)
* Variable datatype
* Query Number 10:
* Calculating the price with and without promotion
* Solution: Cartesian Product

## Suggestions for improvements:

We would like to organize the source code better to make it cleaner and clearer (easier to understand). Also, we would like to write the code in a way that you just need to hit execute without selecting different parts of the code to run it. Finally, we want to make sure the ER diagram is well done and complete before coding to make it easier.

# Conclusions and Future Work

## Conclusions:

In conclusion, this project helped us learn a lot of important things such as creating a database, using SQL, and managing our time and work as a team.

## Future Work:

For data analyst roles, SQL is again the most in-demand skill, listed in 57.4% of all data analyst jobs. As a result, having experience with SQL and creating database will help us a lot getting a good job in the future.