

---

# CAR RENTAL DATABASE MANAGEMENT SYSTEM

## PROJECT DESIGN REPORT

### Team Members:

Hasan Cenk Yavru – 270201021

Yusuf Emre Beskan - 270201005

Cüneyd Çelik - 270201007

Burak Keçeci - 290201103

Mehmet Kutay Akpınar – 300201121

# Contents

1. Introduction
2. Requirements
3. Entities
4. Relations
5. Business Rules
6. Assumptions
7. Users
8. ER Diagram
9. Relational Schema
10. References

# ***1. Introduction***

The number of people using cars has been increasing day by day. While these cars may belong to the person, they can also be rented by individuals within a certain time period. In this rental process, people face some difficulties. We designed a car rental database system to minimize these concerns that people will encounter when renting a car.

We have grouped the difficulties that customers encounter while renting a car in the following areas.

- Documentation
- Takes a lot of time
- Communication problems
- Car service tracking

Documentation:

There are many documents that the customer has to follow when renting a car with traditional methods.

Takes a lot of time:

It has been noticed that when renting a car with traditional methods, it takes a lot of time to complete in accordance with the procedures.

Communication issues:

While renting a car with traditional methods, communication issues between the customer and the car rental company have been observed.

Car service tracking:

While renting a car with traditional methods, in case of a breakdown, the concerns in the process of sending the vehicles to the service were noted.

Thus, the purpose of our system is to allow our customers to rent a car in line with their wishes. It can pick up customers' cars from various pickup points and deliver them to specific drop-off locations as well. If the rental car is not returned within a specified date and time, a certain delay fee must be paid.

## 2. Requirements

- Car rental company must have a collection of cars.
- Each car must belong to a particular car model and each car must belong to a particular location.
- A person must rent a car to become a customer.
- Customers rent a car based on their city, rental interval, and car model preferences.
- List of cars available to be rented must be shown along with available date and time interval.
- Billing is generated when a car is booked.
- Customers can return the car before or after its due date. In addition, if the customer returns the car after the delivery date, an additional delay fee is calculated and added to the invoice.
- Car daily price will be calculated based on the selected car model.
- Car rental company must provide car insurance for each car they have.
- Car rental company must provide a customer service for dealing with customer's concerns.
- Car rental company must have a car service which provides maintenance of cars.

## 3. Entities

### 1) Customer:

A customer is a person who can reserve, rent and drive a car and according to their membership type they get a discount rate. Staff can also be customers, they get a discount rate accordingly. Customer has identity number, first name, middle name, last name, city, e-mail, phone number, licence class, licence expiration date, type, created datetime, updated datetime, password, and availability.

- a. identity\_number
- b. customer\_first\_name
- c. customer\_middle\_name
- d. customer\_last\_name
- e. customer\_city
- f. customer\_email
- g. customer\_phone\_number
- h. customer\_licence\_class
- i. customer\_licence\_expiration\_date
- j. customer\_type
- k. customer\_created\_datetime
- l. customer\_updated\_datetime
- m. customer\_password
- n. customer\_available

## 2) Staff:

Staff is a customer who works in a rental company and can communicate with customers about problems. Staff has staff id, first name, middle name, last name, e-mail, phone number, created datetime, updated datetime, and availability.

- a. staff\_id
- b. staff\_first\_name
- c. staff\_middle\_name
- d. staff\_last\_name
- e. staff\_phone\_number
- f. staff\_email
- g. staff\_password
- h. staff\_created\_datetime
- i. staff\_updated\_datetime
- j. staff\_available

## 3) Car:

A car is a transportation vehicle that customers rent and use. There are various models of cars. The car has a chassis number, availability, mileage, rating, created datetime, and updated datetime. The car's model, maker and similar features are available in the car model entity.

- a. chassis\_number
- b. car\_availability
- c. car\_mileage
- d. car\_rating
- e. car\_created\_datetime
- f. car\_updated\_datetime

## 4) Car Model:

Car model is the specification of the car that customers choose to rent. Car model has model id, maker, name, class, year, cost per day, and late fee per hour.

- a. model\_id
- b. maker
- c. model\_name
- d. model\_class
- e. model\_year
- f. cost\_per\_day
- g. late\_fee\_per\_hour

## 5) Car Insurance:

It is the table that holds the insurance of cars. Car insurance has the policy number, issuer name, issuer city, issuer phone number, expiration date.

- a. policy\_number
- b. expiration\_date

## 6) Insurance Company:

It is a table that keeps information about the companies that the car rental company insures its own cars with. The insurance company table contains id, name, city, phone number, created datetime, updated datetime, and availability.

- a. company\_id
- b. company\_name
- c. company\_city
- d. company\_phone\_number
- e. company\_created\_datetime
- f. company\_updated\_datetime
- g. company\_available

## 7) Car Service:

The car service is responsible for repairing damage to cars in case of accidents and these records are kept in this table. Car service has an id, name, city, phone number, created datetime, updated datetime, and availability.

- a. car\_service\_id
- b. car\_service\_name
- c. car\_service\_city
- d. car\_service\_phone\_number
- e. car\_service\_created\_datetime
- f. car\_service\_updated\_datetime
- g. car\_service\_available

## 8) Booking:

Booking is when customers book a vehicle for themselves when they want to rent a car. Booking has id, from datetime, return datetime, actual return datetime, created datetime, and updated datetime.

- a. booking\_id
- b. booking\_from\_datetime
- c. booking\_return\_datetime
- d. booking\_actual\_return\_datetime
- e. booking\_created\_datetime
- f. booking\_updated\_datetime

## 9) Payment:

It is the table that keeps the price of the difference between when customers rent and drop off cars. Payment has an id, date, bank name, and payment type.

- a. payment\_id
- b. payment\_type

## 10) Billing:

The billing table is the table that holds the invoices for paid rentals. Billing has an id, datetime, total fee, and total late fee.

- a. billing\_id
- b. billing\_datetime
- c. total\_fee
- d. total\_late\_fee

## 11) Location:

Location is the table that holds the locations of the cars. Location has an id, name, and city.

- a. location\_id
- b. location\_name
- c. location\_city

## 12) Customer Type

Customer type is a table where discount options take place. Customer type has an id, name, and discount rate.

- a. customer\_type\_id
- b. customer\_type\_name
- c. customer\_type\_discount\_rate

## 4. Relations

- a) **Customer to Car:** A car rented by just one customer for each booking.. Car rental is allowed according to the expiration date of the customer's driving licence. The relation name is 'Rents' and this relation has the rents\_created\_datetime field.
- b) **Car Insurance to Car:** A car has many car insurance because for each car, but for each car insurance there must be at most one car. The relation 'has' has a has\_created\_datetime field. This field holds the time that car insurance created.
- c) **Customer to Booking:** For each booking there must be at most one customer and at most one car that is rented by this customer. The relation 'Rents' has the field named by rents\_created\_datetime.
- d) **Car to Booking:** For a specific booking there must be at most one car that is rented by one customer. Also the relation 'Rents' holds rents\_created\_datetime.
- e) **Car to Car Model:** The customer chooses the model of the car and rents it. Each model has a separate price. A car must have at most one car model also a car model must have at least one car. Cars have their own model. The relation's name is identified\_by.
- f) **Car to Car Service:** When the car breaks down, it is fixed by the car service. Every car must have a maintenance service. The car service can fix many cars. The relation is 'fixes' and this relation has fixed\_status\_datetime, fixed\_start\_datetime, fixed\_finished\_datetime, fixed\_failed\_outcome fields.
- g) **Insurance Company to Car Service:** The insurance company can contact many car services in case of accidents, maintenance of cars. Also one customer service can be contacted by many insurance company. The relation is 'contacts' and this relation has contact\_created\_time, contact\_status, contact\_reason, contact\_failed\_outcome.
- h) **Car Insurance to Insurance Company:** Each car insurance created by just one insurance company. A insurance company can create many car insurance. The relation 'create' has a field called as create\_created\_datetime that holds the time insurance created.
- i) **Customer to Staff:** The customer can contact just one staff member when there is a problem with the car, but a staff member can serve to many customers for different times. The relation is 'serves' and this relation has serves\_created\_datetime, serves\_status, serves\_reason and serves\_failed\_outcome fields.
- j) **Car to Location:** For every car there must be one location but for each location can have many car. The relation name is 'belongs to' and this relation has belongs\_to\_created\_datetime.
- k) **Booking to Location:** When a customer books any car, the car must have a location. For each booking there is at most one drop off and pick up car location. Each car location have many bookings. The relation 'drop\_off\_location' has drop\_off\_created\_datetime attribute and the relation 'pick\_up\_location' has a attribute called as pick\_up\_created\_datetime.
- l) **Booking to Billing:** When customer has booking, the booking gives at most one billing but one billing has at least one booking. The relation name is 'gives' and has gives\_created\_datetime, gives\_status, gives\_reason, gives\_failed\_outcome attributes.



- m) **Billing to Payment:** Invoices are paid in cash or by bank card. An invoice is created with the payment. Each billing can be paid by at most one payment type. A payment type can have many billings. The relation name is 'payment\_made' and this relation has pays\_created\_datetime field.
- n) **Customer to Customer Type:** Each customer type can be defined by many customers but a specific customer defines at most one customer type.
- o) **Staff to Insurance Company:** When there is a problem with the car, the customer contacts the staff, who then contacts the insurance company to resolve the problem. Also an insurance company can be informed by more than one staff member. The relation name is 'inform' and this relation has inform\_created\_datetime, inform\_status, inform\_reason, inform\_failed\_outcome fields.

## 5. *Business Rules*

- Customer that is NORMAL membership has %0 discount rate.
- Customer that is BRONZE membership has %10 discount rate.
- Customer that is company's staff has %15 discount rate.
- Customer that is GOLD membership has %20 discount rate.
- Customer that is PREMIUM membership has %30 discount rate.
- The car rental company must renew the insurance of all cars it owns every year.
- Customer should contact the car rental company's staff for his concerns.
- Billing is calculated by product of customer's membership, cost per day and interval of the time they rent.
- The total fee is applied only by multiplying the difference between the return date and the rental from date by the discount rate. In case of late arrival, the late fee is charged without discount.
- Customer can pay the rental fee with various options.
- The customer can pick up the car at a specific location and drop it off at a specific location between the time they rent it.
- Staff must be assigned to the customer.
- Staff may inform insurance company for customer's concerns related with car she used.
- Insurance company may contact car service for customer's concerns related with car she used.
- Customer can rent a car at most 11 months.

## 6. Assumptions

- Each customer can have at most one membership type. They can be updated over time.
- Each membership gives the customer a different discount rate.
- Each customer can have at most one customer service staff. The customer can contact the customer service staff for various matters of interest.
- For a specific car and customer, there is at most one booking.
- For a specific customer and booking, there is at most one car.
- For a specific car and booking, there is at most one customer.
- The staff can inform insurance company when any problem occurs about car.
- The insurance company contact car services when any problem occurs about car.
- Each car can be fixed by at least one car service.
- Each car identified by at most one car model.
- Each car belongs to at most one location.
- Each booking has at most one drops off location and at most one pickup location.
- There must be at least one billing for each booking.
- Each billing at most one payment type.
- If someone has the role of 'Customer' in the system, they are over 21 years old and have a licence for 1 year.
- Users must be logged in into the system to get benefit discounts. Guest users cannot benefit from any discounts.

## ***7. Users***

### **1. Customer:**

The customer determines the rental time interval, pick-up, and drop-off locations on the system. The customer can select the car model and perform the rental process. The customer can add additional features to the rented car. The customer can contact the customer representative in case of any problem.

### **2. Administrator:**

Administration has access to all tables. He/she is the person who manages the car rental system. It can add and delete records from all tables.

### **3. Insurance Company:**

The car insurance company communicates with the car service with the information coming from the staff. It also makes the annual insurance of the cars and keeps their dates.

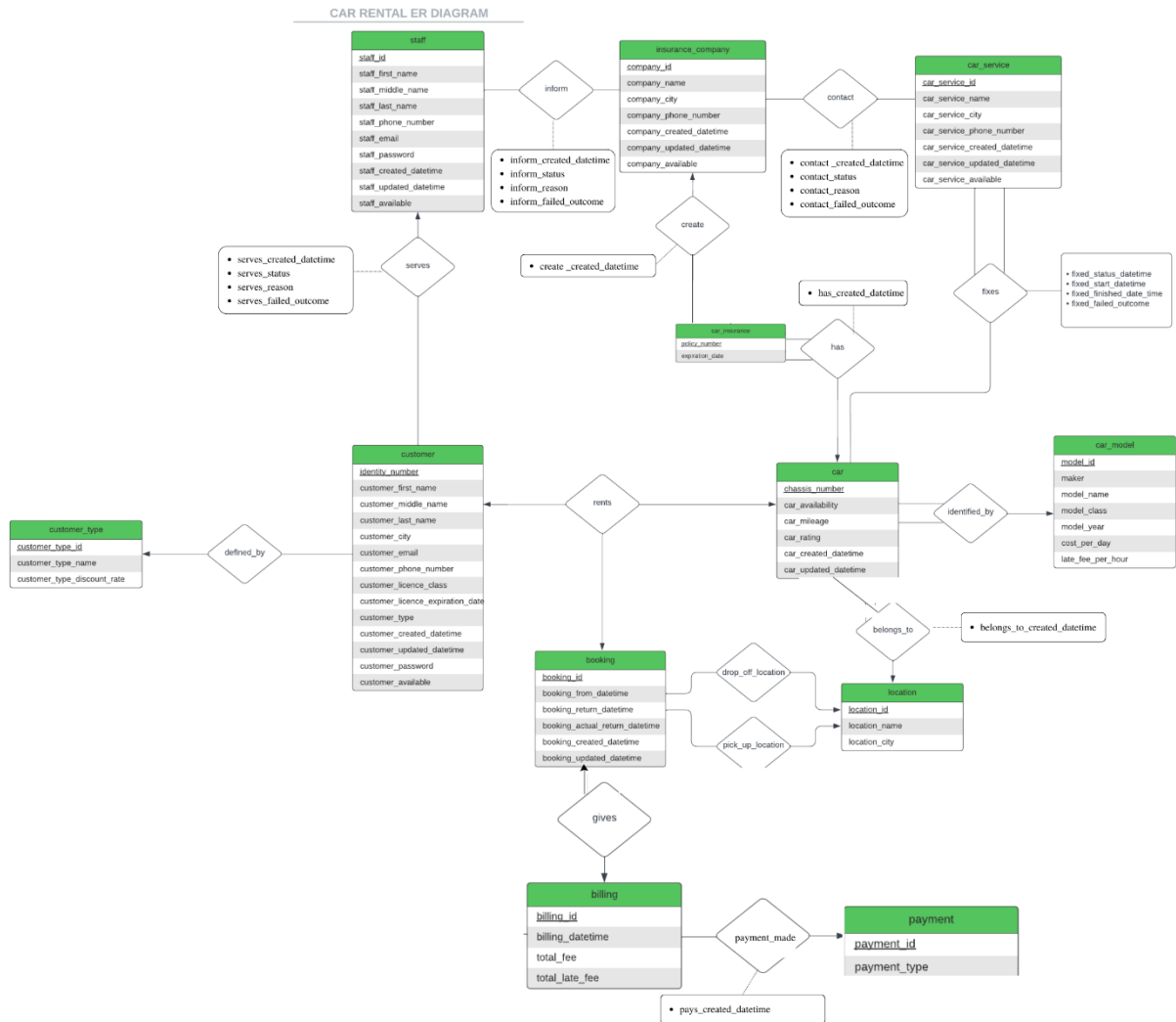
### **4. Car Service Company:**

Car service is the unit that receives information from car insurance about the situation and repairs the car. It determines the repair date, repair status, completion date of the repair and failed outcome information.

### **5. Staff:**

Staff is the unit that deals with customer feedback. It determines the creation days, status, reasons, failed outcomes of the developments related to the feedback. It informs the insurance company about the processes it deems necessary.

## 8. ER Diagram



## 9. Relational Schema

### 9.1 Reduction to Relational Schema

#### 9.1.1 Entities become tables

- Customer (identity\_number, customer\_first\_name, customer\_middle\_name, customer\_last\_name, customer\_city, customer\_email, customer\_phone\_number, customer\_licence\_class, customer\_licence\_expiration\_date, customer\_type, customer\_created\_datetime, customer\_updated\_datetime, customer\_password, customer\_available)
- Customer\_Type (customer\_type\_id, customer\_type\_name, customer\_type\_discount\_rate)
- Car (chassis\_number, car\_availability, car\_mileage, car\_rating, car\_created\_datetime, car\_updated\_datetime)
- Booking (booking\_id, booking\_from\_datetime, booking\_return\_datetime, booking\_actual\_return\_datetime, booking\_created\_datetime, booking\_updated\_datetime)
- Billing (billing\_id, billing\_datetime, total\_fee, total\_late\_fee)
- Car\_Insurance (policy\_number, expiration\_date)
- Location (location\_id, location\_name, location\_city)
- Car Model (model\_id, maker, model\_name, model\_class, model\_year, cost\_per\_day, late\_fee\_per\_hour)
- Staff (staff\_id, staff\_first\_name, staff\_middle\_name, staff\_last\_name, staff\_email, staff\_phone\_number, staff\_created\_datetime, staff\_updated\_datetime, staff\_password, staff\_available)
- Insurance\_Company (company\_id, company\_name, company\_city, company\_phone\_number, company\_created\_datetime, company\_updated\_datetime, company\_available)
- Car\_Service (car\_service\_id, car\_service\_name, car\_service\_city, car\_service\_phone\_number, car\_service\_created\_datetime, car\_service\_updated\_datetime, car\_service\_available)
- Payment (payment\_id, payment\_type)

### 9.1.2 Relationships become tables

- Has (has\_id, chassis\_number, policy\_number, has\_created\_datetime)
- Rents (rents\_id, identity\_number, chassis\_number, booking\_id, rents\_created\_datetime)
- Belongs\_To (belongs\_to\_id, chassis\_number, location\_id, belongs\_to\_created\_datetime)
- Pick Up Location (pick\_up\_id, booking\_id, location\_id, pick\_up\_created\_datetime)
- Drop Off Location (drop\_off\_id, booking\_id, location\_id, drop\_off\_created\_datetime)
- Identified\_By (chassis\_number, model\_id)
- Payment\_Made (payment\_id, billing\_id, payment\_id, pays\_created\_datetime)
- Defined\_By (defined\_by\_id, identity\_number, customer\_type\_id)
- Inform (inform\_id, staff\_id, company\_id, inform\_created\_datetime, inform\_status, inform\_reason, inform\_failed\_outcome)
- Fixes (fixes\_id, chassis\_number, car\_service\_id, fixes\_created\_datetime, fixes\_status, fixes\_reason, fixes\_failed\_outcome)
- Contact (contact\_id, company\_id, car\_service\_id, contact\_created\_datetime, contact\_status, contact\_reason, contact\_failed\_outcome)
- Create (create\_id, company\_id, policy\_number, create\_created\_datetime)
- Serves (serves\_id, staff\_id, identity\_number, serves\_created\_datetime, serves\_status, serves\_reason, serves\_failed\_outcome)
- Gives (gives\_id, booking\_id, billing\_id)

### 9.1.3 Some tables receive foreign keys

•Customer (identity\_number, customer\_first\_name, customer\_middle\_name, customer\_last\_name, customer\_city, customer\_email, customer\_phone\_number, customer\_licence\_class, customer\_licence\_expiration\_date, customer\_type, customer\_created\_datetime, customer\_updated\_datetime, customer\_password, customer\_available, **customer\_type\_id**)

•Car (chassis\_number, car\_availability, car\_mileage, car\_rating, car\_created\_datetime, car\_updated\_datetime, **car\_model\_id**, **car\_location\_id**)

•Booking (booking\_id, booking\_from\_datetime, booking\_return\_datetime, booking\_actual\_return\_datetime, booking\_created\_datetime, booking\_updated\_datetime, **customer\_id**, **car\_id**, **pick\_up\_location\_id**, **drop\_off\_location\_id**)

•Billing (billing\_id, billing\_datetime, total\_fee, total\_late\_fee, **payment\_id**, **booking\_id**)

•Car\_Insurance (policy\_number, expiration\_date, **insurance\_company\_id**)

### 9.1.4 Some tables are removed

•Has (~~has\_id~~, chassis\_number, policy\_number, has\_created\_datetime)

•Rents (~~rents\_id~~, ~~identity\_number~~, ~~chassis\_number~~, ~~booking\_id~~, ~~rents\_created\_datetime~~)

•Belongs\_To (~~belongs\_to\_id~~, ~~chassis\_number~~, ~~location\_id~~, ~~belongs\_to\_created\_datetime~~)

•Pick Up Location (~~pick\_up\_id~~, ~~booking\_id~~, ~~location\_id~~, ~~pick\_up\_created\_datetime~~)

•Drop Off Location (~~drop\_off\_id~~, ~~booking\_id~~, ~~location\_id~~, ~~drop\_off\_created\_datetime~~)

•Identified\_By (chassis\_number, model\_id)

•Payment\_Made (~~payment\_id~~, ~~billing\_id~~, ~~payment\_id~~, ~~pays\_created\_datetime~~)

•Defined\_By (defined\_by\_id, identity\_number, customer\_type\_id)

•Create (create\_id, company\_id, policy\_number, create\_created\_datetime)

•Gives (gives\_id, booking\_id, billing\_id)

## 9.2 Normalized Relation Schema

•Customer (identity\_number, customer\_first\_name, customer\_middle\_name, customer\_last\_name, customer\_city, customer\_email, customer\_phone\_number, customer\_licence\_class, customer\_licence\_expiration\_date, customer\_type, customer\_created\_datetime, customer\_updated\_datetime, customer\_password, customer\_available, customer\_type\_id)

•Car (chassis\_number, car\_availability, car\_mileage, car\_rating, car\_created\_datetime, car\_updated\_datetime, car\_model\_id, car\_location\_id)

•Booking (booking\_id, booking\_from\_datetime, booking\_return\_datetime, booking\_actual\_return\_datetime, booking\_created\_datetime, booking\_updated\_datetime, customer\_id, car\_id, pick\_up\_location\_id, drop\_off\_location\_id)

•Billing (billing\_id, billing\_datetime, total\_fee, total\_late\_fee, payment\_id, booking\_id)

•Car\_Insurance (policy\_number, expiration\_date, insurance\_company\_id)

•Location (location\_id, location\_name, location\_city)

•Car\_Model (model\_id, car\_type\_id, model\_year, cost\_per\_day, late\_fee\_per\_hour, car\_type\_id)

•Car\_Type (car\_type\_id, car\_type\_maker, car\_type\_model\_name, car\_type\_model\_class)

•Customer\_Type (customer\_type\_id, customer\_type\_name, customer\_type\_discount\_rate)

•Staff (staff\_id, staff\_first\_name, staff\_middle\_name, staff\_last\_name, staff\_email, staff\_phone\_number, staff\_created\_datetime, staff\_updated\_datetime, staff\_password, staff\_available)



- Insurance\_Company (company\_id, company\_name, company\_city, company\_phone\_number, company\_created\_datetime, company\_updated\_datetime, company\_available)
- Car\_Service (car\_service\_id, car\_service\_name, car\_service\_city, car\_service\_phone\_number, car\_service\_created\_datetime, car\_service\_updated\_datetime, car\_service\_available)
- Payment (payment\_id, payment\_type)
- Inform (inform\_id, staff\_id, company\_id, inform\_created\_datetime, inform\_status, inform\_reason, inform\_failed\_outcome)
- Fixes (fixes\_id, chassis\_number, car\_service\_id, fixes\_created\_datetime, fixes\_status, fixes\_reason, fixes\_failed\_outcome)
- Contact (contact\_id, company\_id, car\_service\_id, contact\_created\_datetime, contact\_status, contact\_reason, contact\_failed\_outcome)
- Serves (serves\_id, staff\_id, identity\_number, serves\_created\_datetime, serves\_status, serves\_reason, serves\_failed\_outcome)

## 10. References

- [http://www.cs.rpi.edu/~sibel/csci4380/spring2016/course\\_notes/er.html](http://www.cs.rpi.edu/~sibel/csci4380/spring2016/course_notes/er.html)
- [https://docs.oracle.com/cd/A57673\\_01/DOC/server/doc/SCN73/ch15.htm](https://docs.oracle.com/cd/A57673_01/DOC/server/doc/SCN73/ch15.htm)
- <https://stackoverflow.com/questions/37041285/sql-trigger-calculating-difference-for-timestamps>
- <https://lucid.app/>