University of Crete – Department of Computer Science

HY360 – Files and Databases

Teacher: Dimitrios Plexousakis

Winter Semester 2023-2024

EVOL

GO GREEN GO ELECTRIC

Members

Alexandros Papafragakis, csd5084

Ioannis Trikalopoulos, csd5018

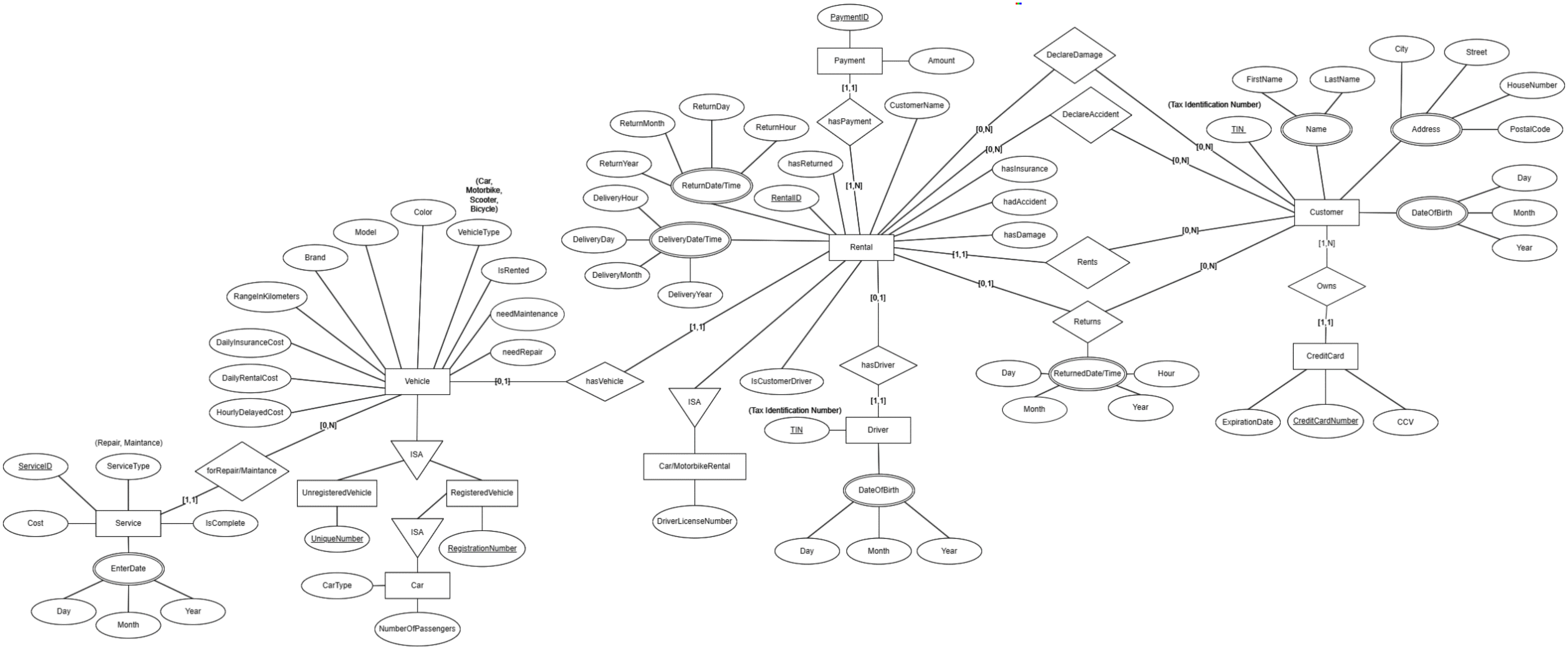
Giorgos Syngelakis, csd4747

Think and describe what you plan to do and why it will be useful.

Contents

1. Entity-relationship diagram

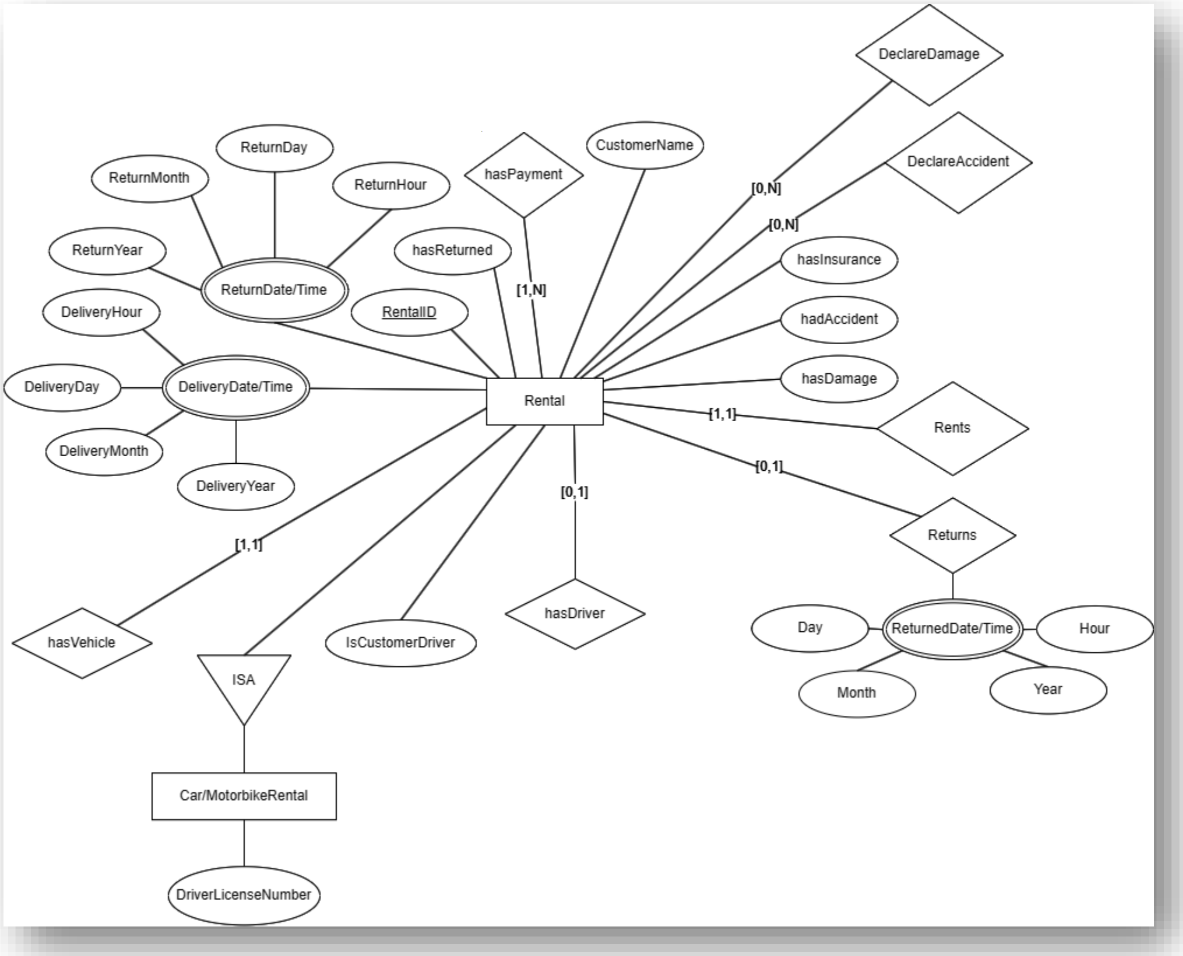
The entity-relationship diagram is below:



i) Entities

For the construction of the entity-relationship diagram we used 3**basic** entities:

1. Rental: This entity contains all the basic information a rental needs, these are:



i. Rental ID: It is the primary key of the Rental entity and its type is an integer.

ii. HasInsurance: This identifier has been added to indicate whether a rental includes insurance of the rented vehicle or not. This way when a customer reports an accident of a rental vehicle, it will be seen if insurance has been paid for the rental. The type of the identifier is true or false (Boolean).

iii. Rental Date/Time (RentalDate/Time): A composite identifier of the format DD/MM/YYYY HH/24h. As a composite, it is divided into three obvious simple acquaintances:
(1) Day (RentalDay)
(2) RentalMonth
(3) Years (RentalYear)
(4) Time (DeliveryHour)

iv. Return Date/Time (DeliveryDate/Time): This composite identification of the format DD/MM/YYYY HH/24h declares the date/time when the customer must return his rented vehicle. This date is determined when renting the vehicle. As a composite, it is divided into three obvious simple acquaintances:
(1) Day (DeliveryDay)
(2) Delivery Month
(3) Years (DeliveryYear)
(4) Time (DeliveryHour)

v. HasReturned: This identifier shows whether a rental has been returned, i.e. it has been completed and the customer has paid for it.
The identifier type is Boolean (true false). Initialized to FALSE.

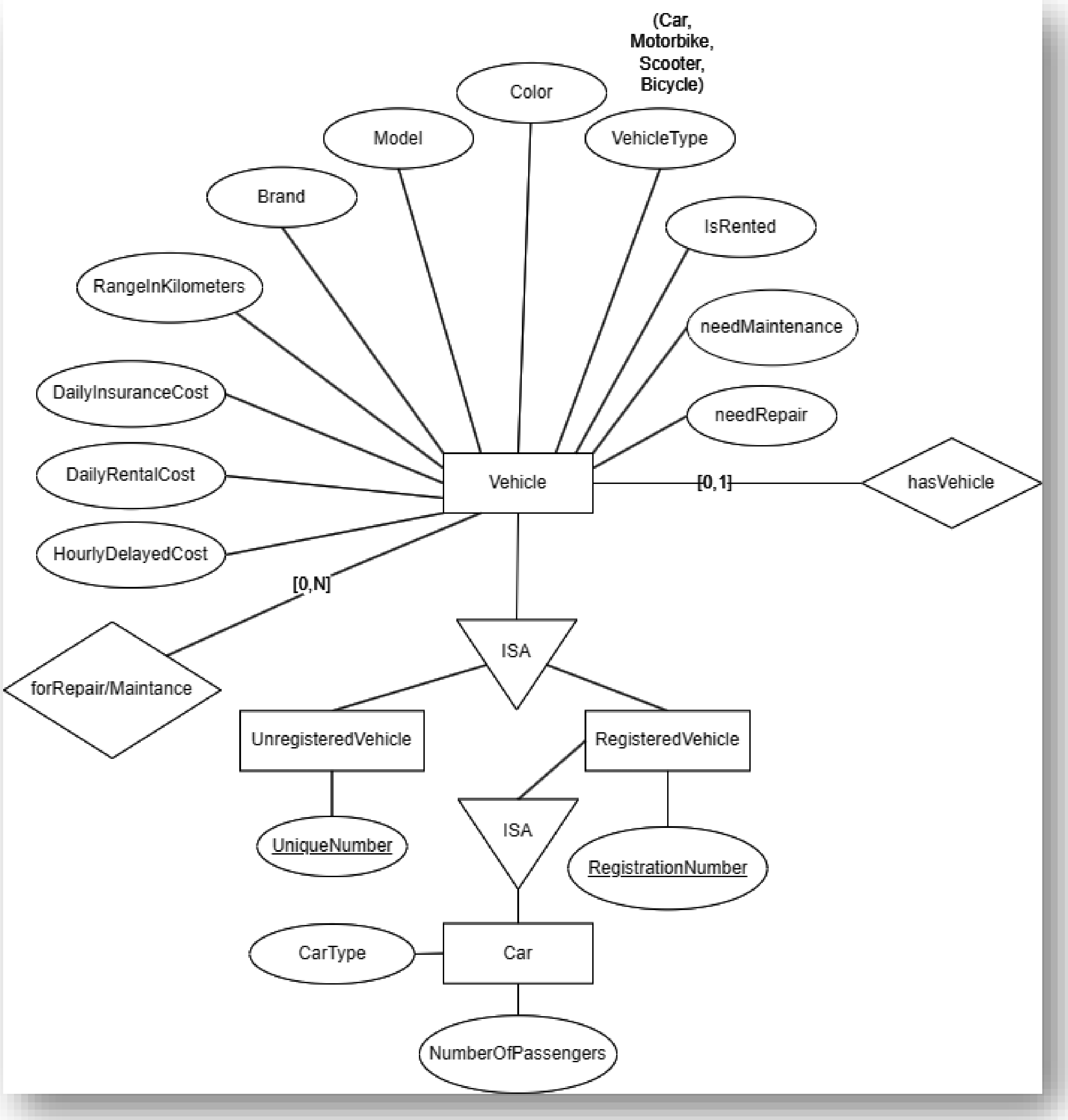
vi. Had accident (HadAccident): This information shows if there has been an accident with the vehicle involved in any rental. The identifier type is Boolean (true false). Initialized to FALSE.

vii. HasDamage: This information shows if there has been damage to the vehicle involved in any rental. The identifier type is Boolean (true false). Initialized to FALSE.

viii. The customer is a driver (IsCustomerDriver): This identifier shows if the customer is the driver in any rental. For the value FALSE, the client must enter a driver in the base (See Driver entity). The identifier type is Boolean.

ix. Insurance is included (hasInsurance): This identifier is of Boolean type, and indicates whether a rental includes insurance for the vehicle in question or not. In the event of an accident and non-insurance of the rental vehicle, the customer pays three times the rental cost.

2. Vehicle:



i. Vehicle ID (VehicleId): It is the primary key of the Vehicle relationship and is included in any type of Vehicle. Its type is positive integer

ii. Manufacturing Company (Brand): This identifier is alphanumeric and states the manufacturing company of each vehicle.

iii. Model: This identifier is alphanumeric and indicates the model of the vehicle.

iv. Vehicle Type (VehicleType): This identification is of alphanumeric type and declares one of the following four types of vehicle:

(1) Car
(2) Motorcycle
(3) Scooter
(4) Bicycle

v. Color: This identifier is alphanumeric and indicates the color of the vehicle.

vi. Daily Rental Cost (DailyRentalCost): This identification is of the type of natural number and declares the cost of the daily rental of the vehicle.

vii. HourlyDelayedCost: The quote is of the natural number type and states the cost incurred by the customer if he does not return his rented vehicle by the hour.

viii. Daily Insurance Cost (DailyInsuranceCost): The quote is a natural number type and states the cost that the customer pays per day if the rented vehicle is insured.

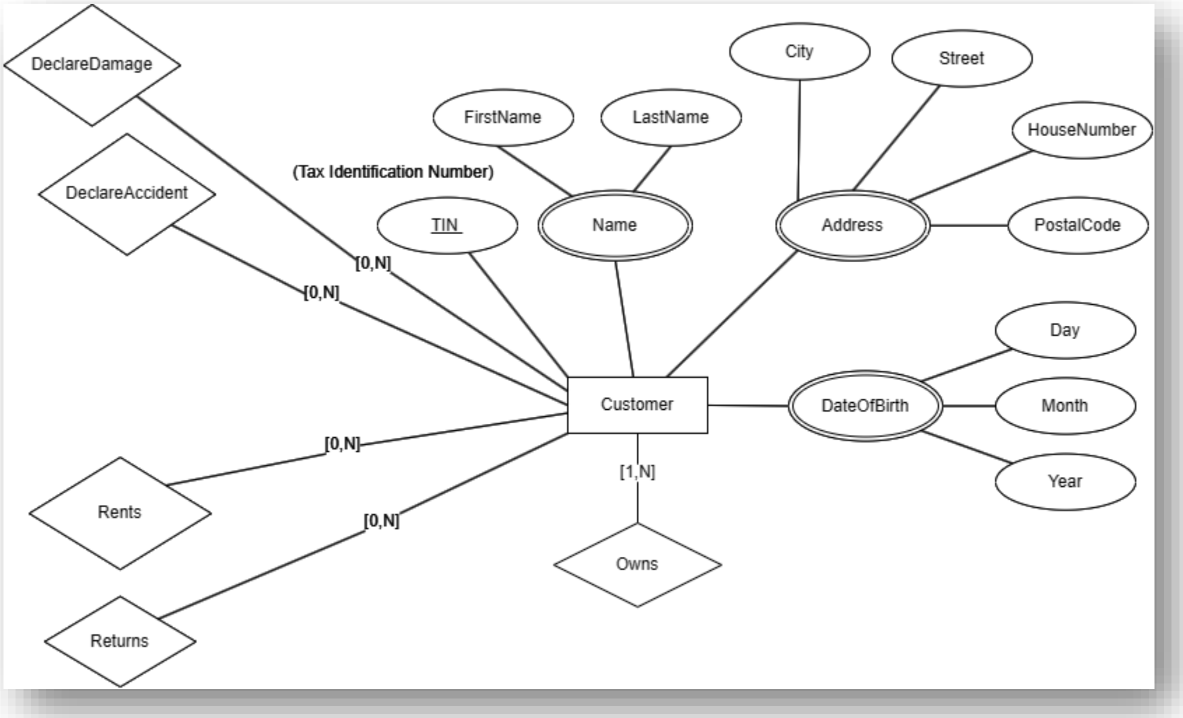
ix. Autonomy in kilometers (RangeInKilometers): The identification is of the type of natural number and declares the autonomy that each vehicle has in kilometers.

x. IsRented: The identifier is of Boolean type and indicates whether a vehicle has been rented or not. When procuring vehicles, the acquaintance takes the value FALSE.

xi. Need Maintenance (needMaintenance): The identifier is of Boolean type and states how much the vehicle needs maintenance. If maintenance is needed, it is withdrawn for a day. Initialized to FALSE.

xii. Need Repair (needRepair): The identification is of Boolean type and states how much the vehicle needs repair. If it needs to be repaired, it is withdrawn for three days. Initialized to FALSE.

3. Customer:



i. VAT number (TIN): The primary key of the customer relationship, has a ten-digit alphanumeric format and declares the VAT number of each customer.

ii. Full Name: A composite argument that declares the customer's full name. It is divided into the following two simple acquaintances

(1) FirstName, alphanumeric
(2) LastName (LastName), alphanumeric

iii. Address (Address): Composite identifier that indicates the customer's address. It consists of four simple acquaintances:

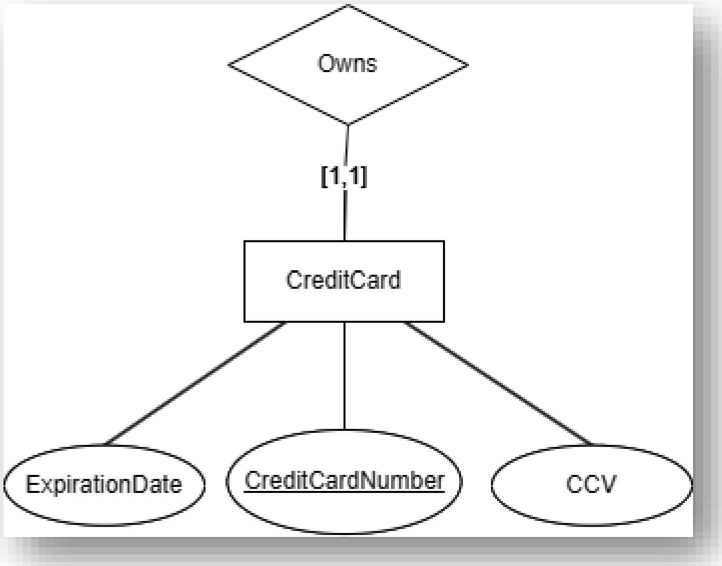
(1) City, alphanumeric
(2) Street, alphanumeric
(3) House Number, 3 character alphanumeric.
(4) Postal Code, 5 character alphanumeric type

iv. Date of Birth (DateOfBirth): Composite identifier indicating the customer's date of birth. It consists of six simple acquaintances:

(1) Day (BirthDay)
(2) BirthMonth
(3) Time (BirthYear)

Additional entities used are the following:

- Credit Card (Credit Card):



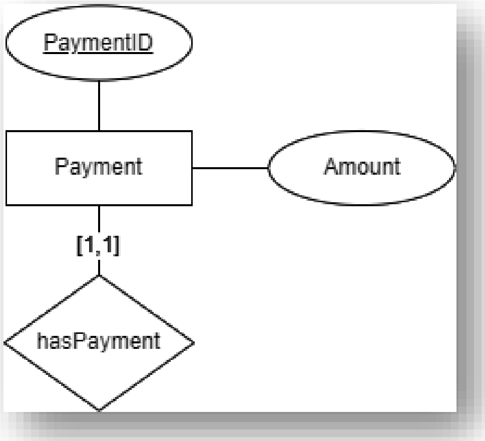
i. Card Number (CreditCardNumber): It is the key of the entity and declares the customer's card number. His type identifier is a 16-digit alphanumeric code.

ii. Expiration Date (ExpirationDate): Composite identifier that appears to be divided into six simple ones:

1. Expiration Month
2. Expiration Year (ExpirationYear)

iii. CCV: The identification is alphanumeric with three characters and states the CCV of the customer's credit card.

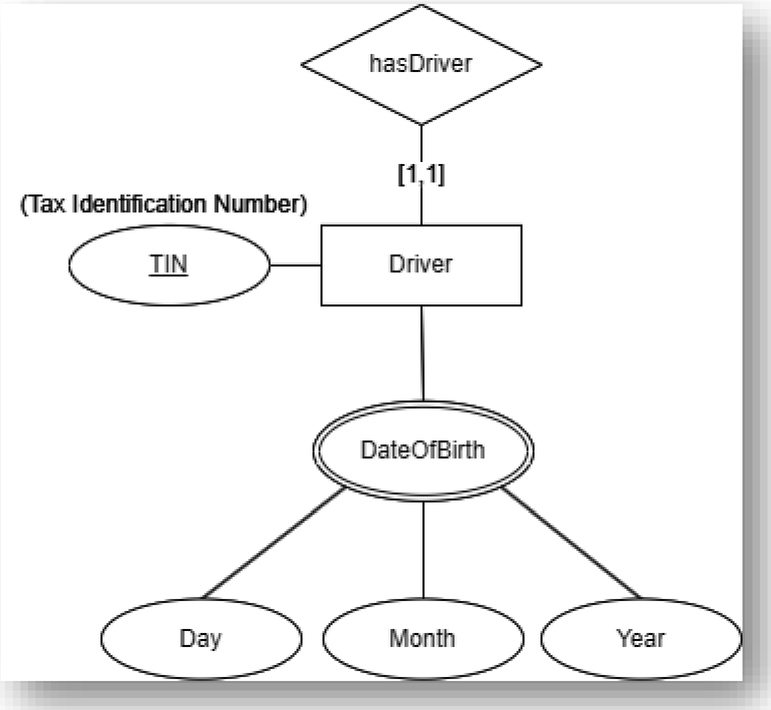
Payment:



i. Payment ID: The primary key of the Payment relationship and its type is integer

ii. Amount: Knowledge of a natural number type, indicating the amount of the payment.

- Guide (Drivers):

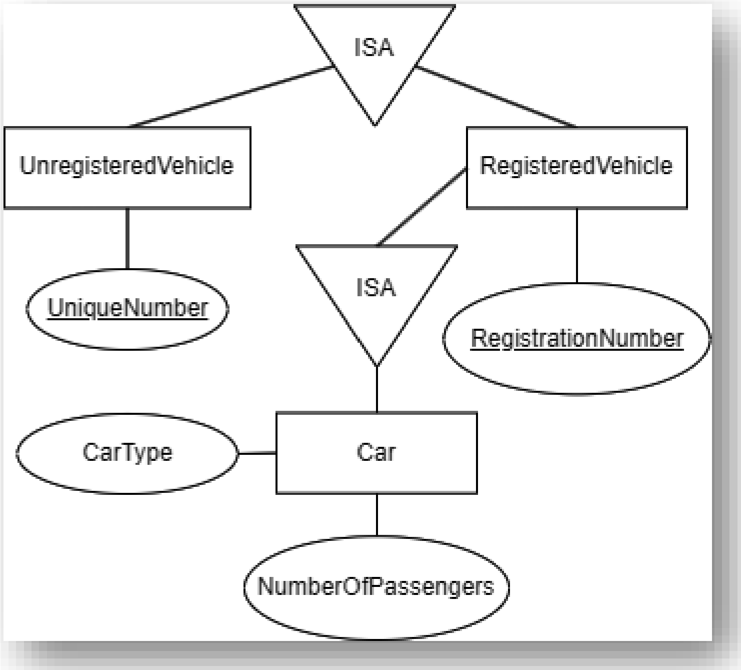


i. Tax Identification Number (TIN): Primary identifier for the driver's entity, it is a ten-digit alphanumeric number.

ii. Date of Birth (DateOfBirth): Composite identifier indicating the driver's date of birth. This is how it looks if the driver is over 16 (to be able to drive a skateboard or bicycle) or over 18 (to be able to drive a car or motorbike). It consists of the following acquaintances:

1. Birthday
2. BirthMonth
3. Time (BirthYear)

- Car/motorcycle rental (Car/MotorbikeRental):
  i. It is a subset of the rental entity so it inherits all of its identities, as well as its primary key (RentalID).
  ii. Driver's license number (DriverLicenseNumber): A 5-character alphanumeric code containing letters (first 2) and three integers (last three characters). It states the driver's license number.

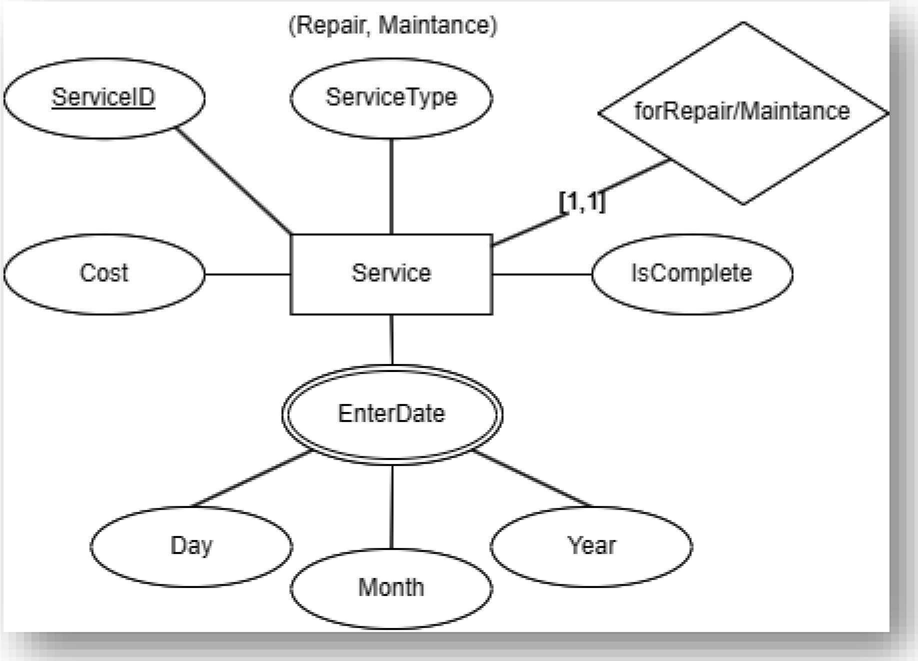


- Vehicle without registration number (UnregisteredVehicle):
  - i. It inherits all the identities of the Vehicle relationship through the IsA relationship, since UnregisteredVehicle isA Vehicle, and the primary key is VehicleID
  - ii. Unique Number: Knowing an alphanumeric type of five characters, it is also the key of the relationship. It shows the unique number of skates/bikes and not the unique number among all vehicles.

- Vehicle with registration number (RegisteredVehicle):
  - i. It inherits all the identities of the Vehicle relationship through the IsA relationship, since RegisteredVehicle isA Vehicle, and the primary key is VehicleID
  - ii. Registration Number: Unique number among the vehicles with a registration number, so it is a key to the relationship. Its type is 10-digit alphanumeric with 10 integer characters.

- Car (Cars):
  - i. It is a subset of the entity Vehicle with registration number since it inherits all the identifiers of this relationship, so keys are VehicleID and RegistrationNumber.
  - ii. Car type (CarType): Knowing an alphanumeric type that identifies the type of car.
  - iii. Number of Passengers (NumberOfPassengers): A natural number type identifier that determines the maximum number that can fit in the car.

- Baggage/Maintenance Services (Service):



i. Service ID (ServiceID): The primary key of the relationship, alphanumeric type with digits.

ii. Date of entry (EnterDate): Composite identification, which declares the entry date of a vehicle for service. This knowledge is used to decide when a repair is complete (the repair takes three days while the maintenance takes one). It consists of six simple acquaintances:

1. Enter Day (EnterDay), alphanumeric
2. Enter Month (EnterMonth), alphanumeric
3. Enter Year (EnterYear), alphanumeric

iii. Cost (Cost): Decimal-type data, which declares the cost of the service.

iv. It has been completed (IsComplete): Boolean information, it states how complete the service is, i.e. whether the repair or maintenance has been done. Initialized to FALSE.

v. ServiceType: An alphanumeric type knowledge that takes values in the set {'Maintenance','Repair'}. It states the type of service that each vehicle has.

Unless otherwise stated, all identifiers are initialized when the corresponding constellation is entered into the database.

ii) Relationships between entities

- **Has a vehicle (hasVehicle) [Vehicle – Rental]**
  This relationship combines the vehicle involved in a rental.
  i. Each vehicle can be included in at most one rental [0,1], while a rental refers to exactly one vehicle [1,1].

- **Rents (Rents) [Rental – Customer]**
  This relationship combines the customer and the rental he wants to do for a vehicle.
  i. Each lease has exactly one client [1,1], while a registered client can perform as many leases as it wants [0,N].

- **Claims Damage (DeclareDamage) [Rental – Customer]**
  The relationship combines the customer who wants to declare damage to the vehicle that concerns his rental.
  i. Each rental of a vehicle can be damaged at most once [0,1], while a customer can declare damage as many times as necessary (for different rentals) ie [0,N].

- **Report Accident (DeclareAccident) [Rental – Customer]**
  The relationship brings together the customer who wants to report an accident to the vehicle that concerns his rental.
  i. Each rental of a vehicle can suffer at most one accident [0,1], while a customer can declare an accident as many times as needed (for different rentals) i.e. [0,N].

- **Returns (Returns) [Rental – Customer]**
  The relationship brings together the customer who wants to return their rental vehicle, thus completing their rental.
  i. Each rental can be completed, i.e. the vehicle referred to in it, returned at most once [0,1]. Each customer can return a rented vehicle as many times as he wants (different rented vehicles from the same customer) or not to return any if he has not made any rental, so [0,N].

  ii. Returned Date/Time (ReturnedDate/Time):
  Composite identifier of the format DD/MM/YYYY HH/24h that states the date the customer returned the rented vehicle, or completed the rental. This introduction has no value until the customer completes his rental. It consists of six simple acquaintances:

  (1) ReturnedDay (ReturnedDay), natural number
  (2) Returned Month, natural number
  (3) Returned Year (ReturnedYear), natural number
  (4) ReturnedHour, natural number

- **Own (Owns) [Customer – CreditCard]**
  The relationship connects the customer with their credit card.
  i. A customer must register at least one credit card [1,N], while one card must belong to exactly one customer.

- **Has a driver (hasDriver) [Rental – Driver]**
  The relationship combines the entity of the rental with the driver of the rental vehicle that concerns it.
  i. A rental can have at most one driver. In case the driver is the same customer, no separate driver is created and the Boolean value of Rental isCustomerDriver becomes True, i.e. [0,1]. A driver can drive exactly one rental vehicle at a time, ie [1,1].

- **For repair/maintenance (forRepair/Maintenance) [Vehicle – Service]** The relationship combines the entity of the vehicle with the maintenance service that concerns it.
  i. A vehicle can suffer damage or need maintenance as many times as needed (same vehicle for different rentals, one damage/maintenance per rental), so [0,N]. A towing/maintenance service is about exactly one vehicle since the cardinality is [1,1].

2. Relational model

The translation of the previous entity relationship diagram into a relational model is the following tables. Underlined letters show the foreign keys, while bold letters are the keys of each relation.

Vehicle

VehicleID	Color	Model	Brand	RangeInKilometers	DailyRentalCost	DailyInsuranceCost
needMaintenance	needRepair	IsRented	VehicleType	HourlyDelayedCost		

UnregisteredVehicle

VehicleID	**UniqueNumber**

RegisteredVehicle

VehicleID	**RegistrationNumber**

Car

CarType	NumberOfPassengers	**VehicleID**

Rental

ReturnHour	ReturnDay	ReturnMonth	ReturnYear	ReturnedHour	ReturnedDay	ReturnedMonth	ReturnedYear
DeliveryHour	DeliveryDay	DeliveryMonth	DeliveryYear	hasReturned	hasDamage	hadAccident	hasInsurance
RentalID	TIN (Customer)	VehicleID (Vehicle)					

Car/MotorbikeRental

DriverLicenseNumber	**RentalID**

Payment

PaymentID	Amount	RentalID

Driver

TIN	BirthDay	BirthMonth	BirthYear	RentalID

Customer

TIN	FirstName	LastName	City	Street	HouseNumber	PostalCode	BirthDay	BirthMonth	BirthYear

Credit Card

CreditCardNumber	ExpirationMonth	ExpirationYear	CCV	TIN

Service

ServiceID	Cost	EnterDay	EnterMonth	EnterYear	IsComplete	ServiceType	VehicleID (Vehicle)

i) Functional Dependencies

The functional dependencies resulting from the above relations are as follows:

- ➢ Vehicle
  - **VehicleID**→ VehilcleType, Color, Model, Brand, RangeInKilometers, DailyInsuranceCost, DailyRentalCost, HourlyDelayedCost, needMaintenance, needRepair, IsRented
  - VehicleType, Brand, Model → RangeInKilometers, DailyInsuranceCost, DailyRentalCost, HourlyDelayedCost

- ➢ UnregisteredVehicle
  - **UniqueNumber**→ VehicleID
  - **VehicleID**→ UniqueNumber

- ➢ RegisteredVehicle
  - **VehicleID**→ RegistrationNumber
  - **RegistrationNumber**→ VehicleID

- ➢ Car
  - **VehicleID**→ CarType, NumberOfPassengers

- ➢ Rental
  - **RentalID**→ ReturnHour, ReturnDay, ReturnMonth, ReturnYear, ReturnedHour, ReturnedDay, ReturnedMonth, ReturnedYear, DeliveryHour, DeliveryDay, DeliveryMonth, DeliveryYear, hasReturned, CustomerName, hasDamage, hadAccident, hasInsurance, TIN, VehicleID
  - TIN, VehicleID →**RentalID**
  - **RentalID**→**DriverLicenseNumber**
  - **DriverLicenseNumbe**r →**RentalID**

- ➢ Payment
  - **PaymentID**→ Amount, RentalID

- ➢ Driver
  - **TIN**→ BirthDay, BirthMonth, BirthYear, RentalID

- ➢ Customer
  - **TIN**→ FirstName, LastName, City, Street, HouseNumber, PostalCode, BirthDay, BirthMonth, BirthYear

- ➢ Credit Card
  - **CreditCardNumber**→ExpirationDay,ExpirationMonth, CCV, TIN

- ➢ Service
  - **ServiceID**→ Cost, EnterDay, EnterMonth, EnterYear, IsComplete, ServiceType, UniqeNumber, RegistrationNumber

ii) Third normative form

From the SEs, we notice that the only dependency that does not include a relation key in its left member is:

- VehicleType, Brand, Model ⟶ RangeInKilometers, DailyInsuranceCost, DailyRentalCost, HourlyDelayedCost

This functional dependency breaks the definition of 3NF since the VehicleType, Brand, Model identifiers are not superkeys of the relation and no identifier from the right member is primary.
Therefore to transform the base into 3$_{the}$Normal form (3NF), there must be a decomposition between the first four SEs where there will be no loss of information, i.e. it will be lossless.

'We have:

Vehicle (VehilcleType, Color, Model, Brand, RangeInKilometers, DailyInsuranceCost, DailyRentalCost, HourlyDelayedCost, needMaintenance, needRepair, IsRented,**VehicleID**)

I decompose Vehicle into:
Vehicle (Color, needMaintenance, needRepair, IsRented,**VehicleID** **, VehicleType, Brand, Model**)

VehicleTypeBrandModel (**VehicleType** **,Brand** **,Model** , RangeInKilometers, DailyInsuranceCost, DailyRentalCost, HourlyDelayedCost)

HEAD(Vehicle)∩HEAD(VehicleTypeBrandModel) = VehicleType, Brand, Model which is a key of VehicleTypeBrandModel aka lossless. So the Vehicle relation decomposes into:

Vehicle

VehicleID	Color	needMaintenance	needRepair	IsRented	VehicleType	Brand	Model

And the VehicleTypeBrandModel relation is added.

VehicleTypeBrandModel

VehicleType	**Model**	**Brand**	DailyInsuranceCost
DailyRentalCost	HourlyDelayedCost	RangeInKilometers	

* * The triple (VehicleType,Model,Brand) is the key of the VehicleTypeBrandModel relationship and is in Vehicle as a foreign key.

Finally, the "Vehicle" relationship dependency becomes:

VehicleID⟶ VehicleType, Color, Model, Brand, needMaintenance, needRepair, IsRented

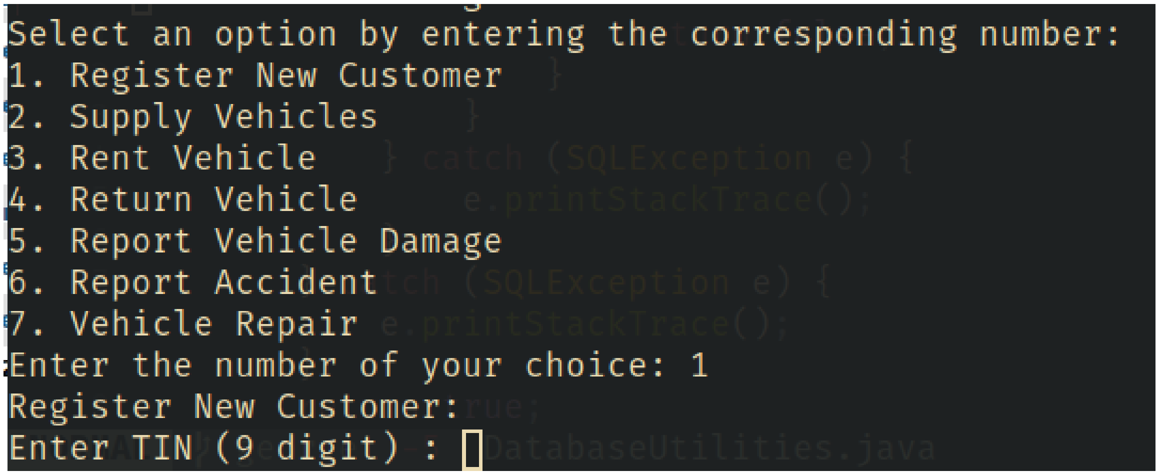
And the relationship “VehicleTypeBrandModel**"**:

VehicleType, Brand, Model⟶ RangeInKilometers, DailyInsuranceCost, DailyRentalCost, HourlyDelayedCost

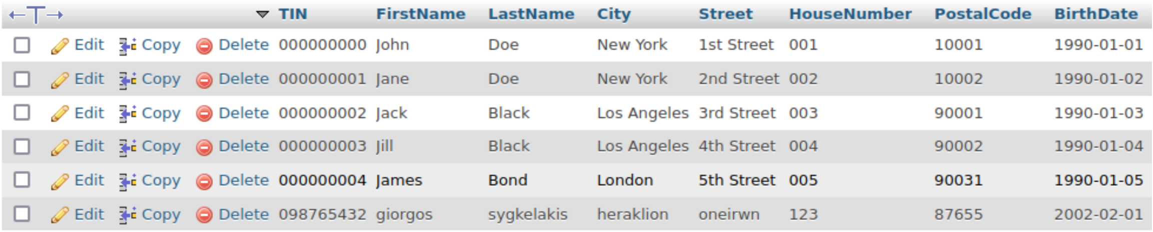
We notice that in the resulting final figure, all SEs have either the key of the relationship on the left side, or a primary acquaintance on the right side (TIN, VehicleID ⟶ **RentalID).**Therefore, by definition, the shape of the base is at3NF (Third Normal Form).

3. Performance examples

- Example of running a customer import

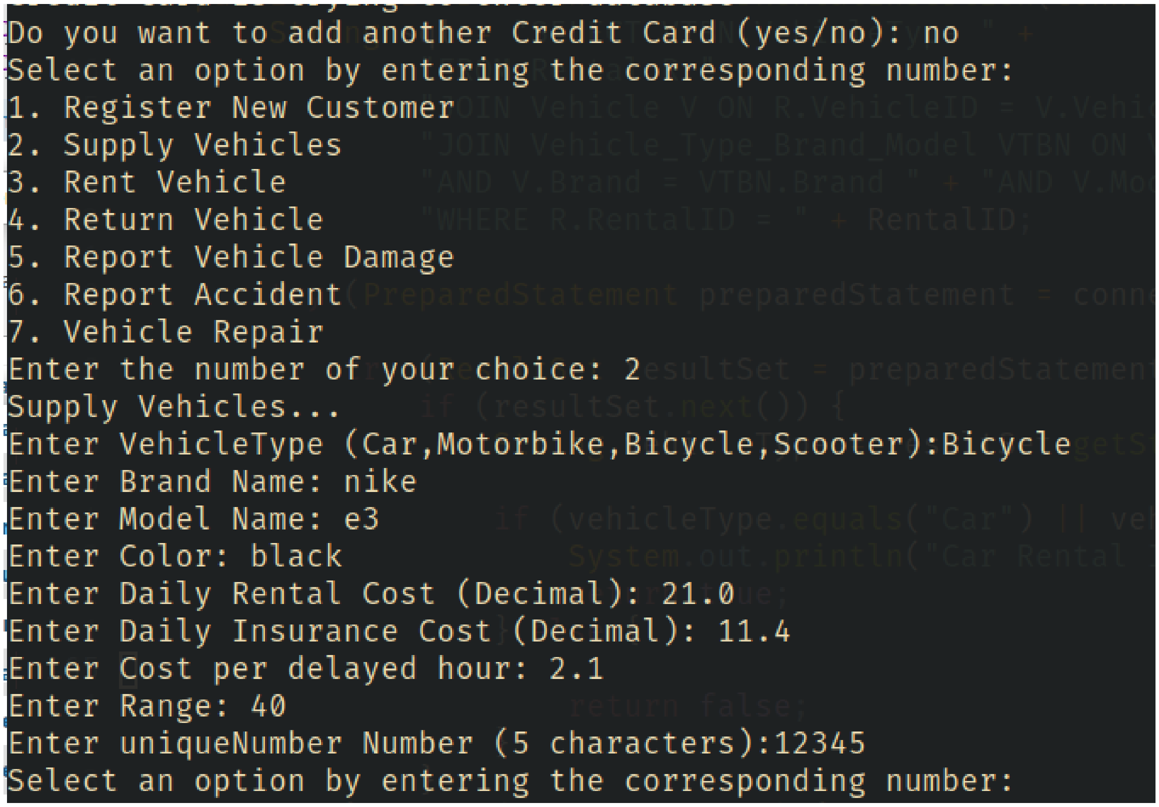


```
Select an option by entering the corresponding number:
1. Register New Customer
2. Supply Vehicles
3. Rent Vehicle
4. Return Vehicle
5. Report Vehicle Damage
6. Report Accident
7. Vehicle Repair
Enter the number of your choice: 1
Register New Customer:
Enter TIN (9 digit) : 
```

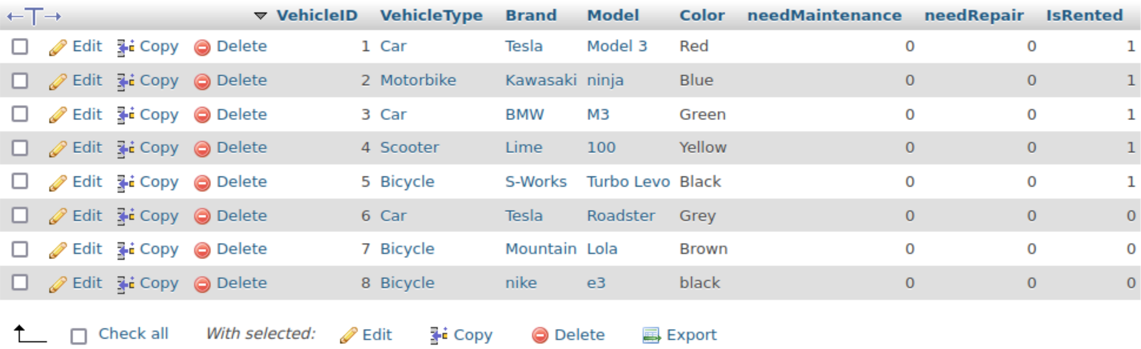


←T→				TIN	FirstName	LastName	City	Street	HouseNumber	PostalCode	BirthDate
□	Edit	Copy	Delete	000000000	John	Doe	New York	1st Street	001	10001	1990-01-01
□	Edit	Copy	Delete	000000001	Jane	Doe	New York	2nd Street	002	10002	1990-01-02
□	Edit	Copy	Delete	000000002	Jack	Black	Los Angeles	3rd Street	003	90001	1990-01-03
□	Edit	Copy	Delete	000000003	Jill	Black	Los Angeles	4th Street	004	90002	1990-01-04
□	Edit	Copy	Delete	000000004	James	Bond	London	5th Street	005	90031	1990-01-05
□	Edit	Copy	Delete	098765432	giorgos	sygkelakis	heraklion	oneirwn	123	87655	2002-02-01

- Example of car procurement

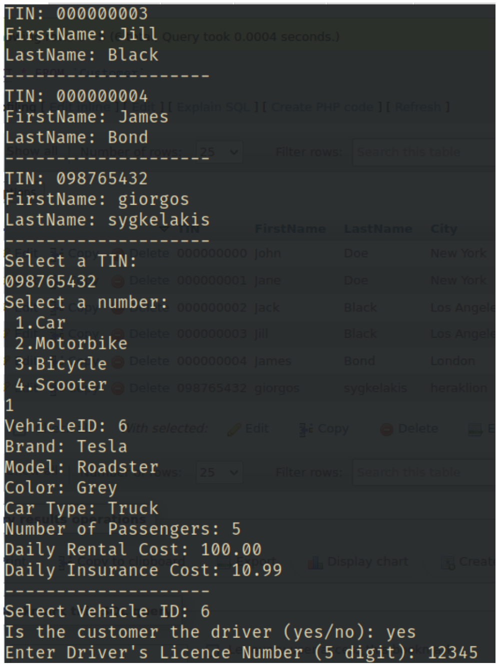


```
Do you want to add another Credit Card (yes/no): no
Select an option by entering the corresponding number:
1. Register New Customer
2. Supply Vehicles
3. Rent Vehicle
4. Return Vehicle
5. Report Vehicle Damage
6. Report Accident
7. Vehicle Repair
Enter the number of your choice: 2
Supply Vehicles...
Enter VehicleType (Car,Motorbike,Bicycle,Scooter):Bicycle
Enter Brand Name: nike
Enter Model Name: e3
Enter Color: black
Enter Daily Rental Cost (Decimal): 21.0
Enter Daily Insurance Cost (Decimal): 11.4
Enter Cost per delayed hour: 2.1
Enter Range: 40
Enter uniqueNumber Number (5 characters):12345
Select an option by entering the corresponding number:
```



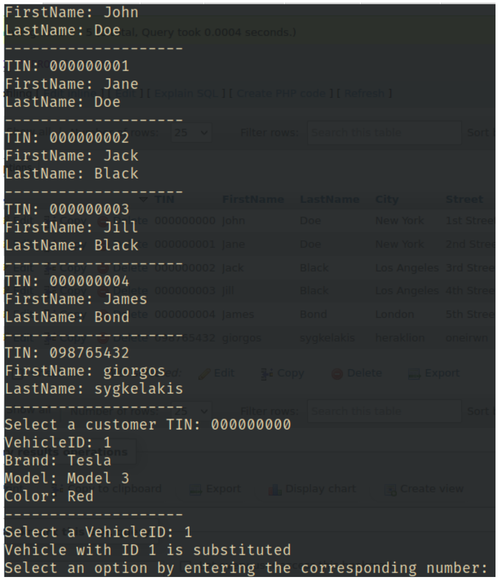
←T→				VehicleID	VehicleType	Brand	Model	Color	needMaintenance	needRepair	IsRented
□	Edit	Copy	Delete	1	Car	Tesla	Model 3	Red	0	0	1
□	Edit	Copy	Delete	2	Motorbike	Kawasaki	ninja	Blue	0	0	1
□	Edit	Copy	Delete	3	Car	BMW	M3	Green	0	0	1
□	Edit	Copy	Delete	4	Scooter	Lime	100	Yellow	0	0	1
□	Edit	Copy	Delete	5	Bicycle	S-Works	Turbo Levo	Black	0	0	1
□	Edit	Copy	Delete	6	Car	Tesla	Roadster	Grey	0	0	0
□	Edit	Copy	Delete	7	Bicycle	Mountain	Lola	Brown	0	0	0
□	Edit	Copy	Delete	8	Bicycle	nike	e3	black	0	0	0

- Example of car rental

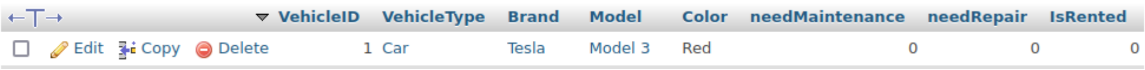


```
TIN: 000000003
FirstName: Jill
LastName: Black
--------------------
TIN: 000000004
FirstName: James
LastName: Bond
--------------------
TIN: 098765432
FirstName: giorgos
LastName: sygkelakis
--------------------
Select a TIN:
098765432
Select a number:
 1.Car
 2.Motorbike
 3.Bicycle
 4.Scooter
1
VehicleID: 6
Brand: Tesla
Model: Roadster
Color: Grey
Car Type: Truck
Number of Passengers: 5
Daily Rental Cost: 100.00
Daily Insurance Cost: 10.99
--------------------
Select Vehicle ID: 6
Is the customer the driver (yes/no): yes
Enter Driver's Licence Number (5 digit): 12345
```

- Example of a bug report



```
FirstName: John
LastName: Doe
--------------------
TIN: 000000001
FirstName: Jane
LastName: Doe
--------------------
TIN: 000000002
FirstName: Jack
LastName: Black
--------------------
TIN: 000000003
FirstName: Jill
LastName: Black
--------------------
TIN: 000000004
FirstName: James
LastName: Bond
--------------------
TIN: 098765432
FirstName: giorgos
LastName: sygkelakis
--------------------
Select a customer TIN: 000000000
VehicleID: 1
Brand: Tesla
Model: Model 3
Color: Red
--------------------
Select a VehicleID: 1
Vehicle with ID 1 is substituted
Select an option by entering the corresponding number:
```



←T→	VehicleID	VehicleType	Brand	Model	Color	needMaintenance	needRepair	IsRented
☐ Edit Copy Delete	1	Car	Tesla	Model 3	Red	0	0	0

4. Opportunities for improvement

- Provide a graphicinterface.

There could be a more commercial image for the client and a cleaner separation of the actions carried out by the company (such as Prometheus vehicle) and those of the client (such as rental).

The creation of a web-app interface was tried, which was not implemented due to limited time, and a rudimentary user-interface was created.