GINA CODY
SCHOOL OF ENGINEERING
AND COMPUTER SCIENCE

Database
COMP- 353 Project:
Winter 2022

Team: COMP-ISRR

INTRODUCTION

Mission database is designed to help steam line the process of managing customer, employees, reservation, invoices, and payments of RENTRACK INC.

Requirements:

RENTRACK Inc.

RENTRACK Company, a young Montreal firm, specialized in the rental of trucks with driver. It has a fleet of a hundred trucks of all types, and currently has a thousand regular customers, mostly businesses. Its business is growing strongly. As the CEO he decided to automate gradually the management of the company.

Description of the different activities related to the rental of tracks.

A. MISSIONS

The rentals have a variable duration. The drivers return necessarily to the garage with their truck at the weekend. Each mission cannot exceed 5 days (Monday morning to Friday evening). So a rental is treated as N missions for a maximum duration of five days each.
A mission begins when the truck leaves the garage (even if the appointment is set at another location) and ends when the truck returns to the garage.
The number of missions generally varies from 100 to 800 per week.
The driver goes with the rented truck to the appointment fixed by the customer, and is available to him for the whole duration of the mission (until Friday evening at the latest).
A driver can drive any vehicle if he has the corresponding driving license: tourism, heavyweight, super heavyweight.

B. RESERVATIONS

Reservations are made by phone with the sales department. A customer can book a truck several weeks in advance for a period of maximum one year. He may of course modify or cancel all or part of a reservation, if he does it no later than one week before starting a mission.
The form used by RENTRACK for reservations is designed to easily decompose a reservation to missions. Thus, the information provided by the client can establish for each mission a record including:

- Number, name or business name and address of the client;
- Type of vehicle desired;
- The place of rendezvous;
- The date and time of appointment;
- The expected duration of making disposal of vehicle and driver.

The portfolio of missions may include up to 10,000 missions to realize.

C. PREPARATION OF MISSIONS

The staff of the sales department then performs the following tasks:

- It determines the date and time of start and end of missions, from information provided by the client when booking;
- It performs the assignment of drivers and vehicles for each mission. This assignment is currently done manually, using schedules wall. The procedure, though heavy, will be kept for some time: the computerization will be the subject of further study.

This allows it to harmonize booking requests and opportunities for the company's service, for each mission.

They plan to computerize the subsequent procedure.

Thus the "mission sheets" definitely established will be daily entered on the terminals in order to create for each mission a record in the database "**Missions**". They seek to minimize the volume of the seizure.

In weekend, they plan to publish "mission sheets" for drivers. These documents define the missions they will be performed in next week, due to a mission per sheet. They will also be used for billing of rentals.

We will not deal, in this study, the case of bookings made in midweek for the current week.

D. BILLING

A weekly invoice is sent to customers.

A customer can rent several vehicles simultaneously or successively. In this case, the invoice includes a line by mission.

Invoices are partly compiled from the "mission sheet" completed by drivers. They enroll in effect:

- Actual dates and hours of start and end of the mission, if they differ from what was expected;
- The value of the vehicle odometer at the start of the garage and return.

The data necessary for billing on the “mission sheet” are daily entered from terminals.

We look again to minimize the volume of the seizure.

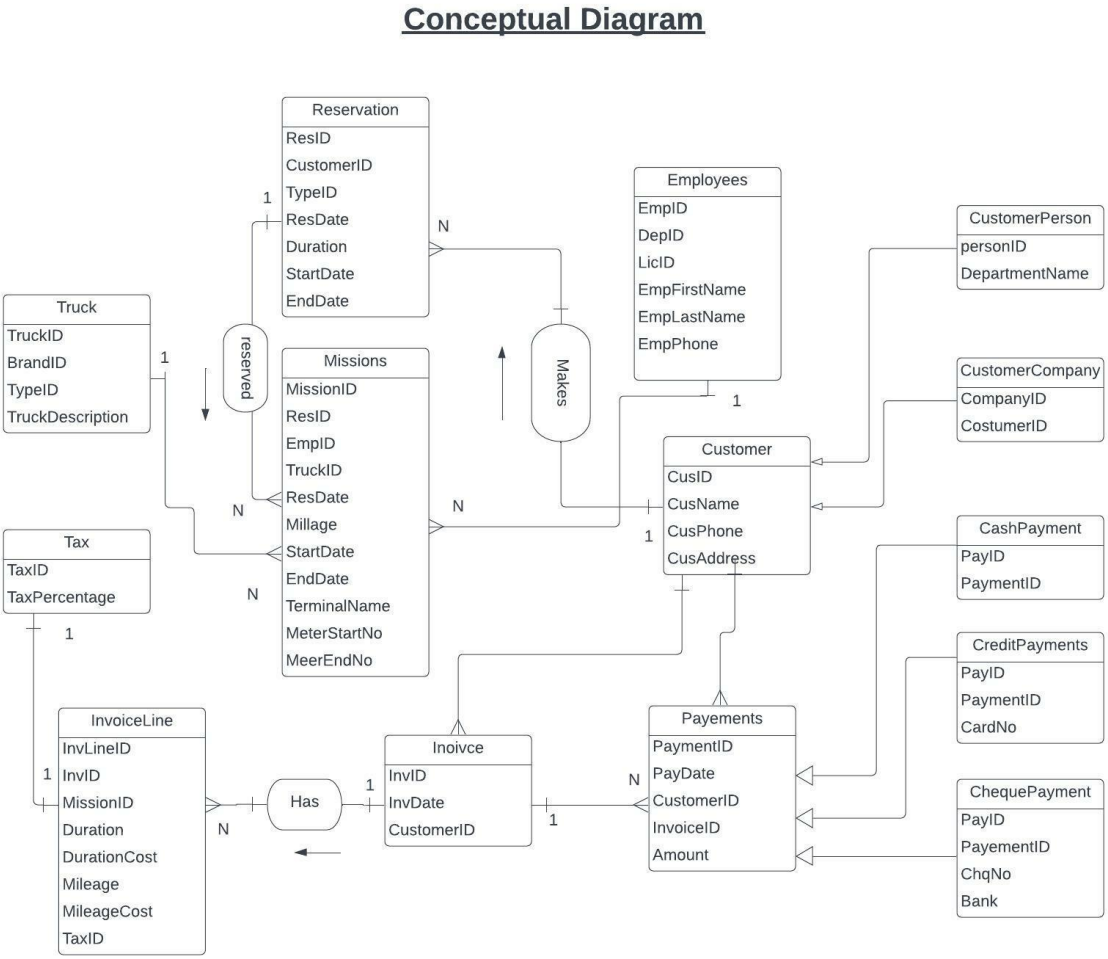
The rental cost charged includes:

- A proportional fraction to the duration of the mission;
- And a proportional part to the browsed kilometers.

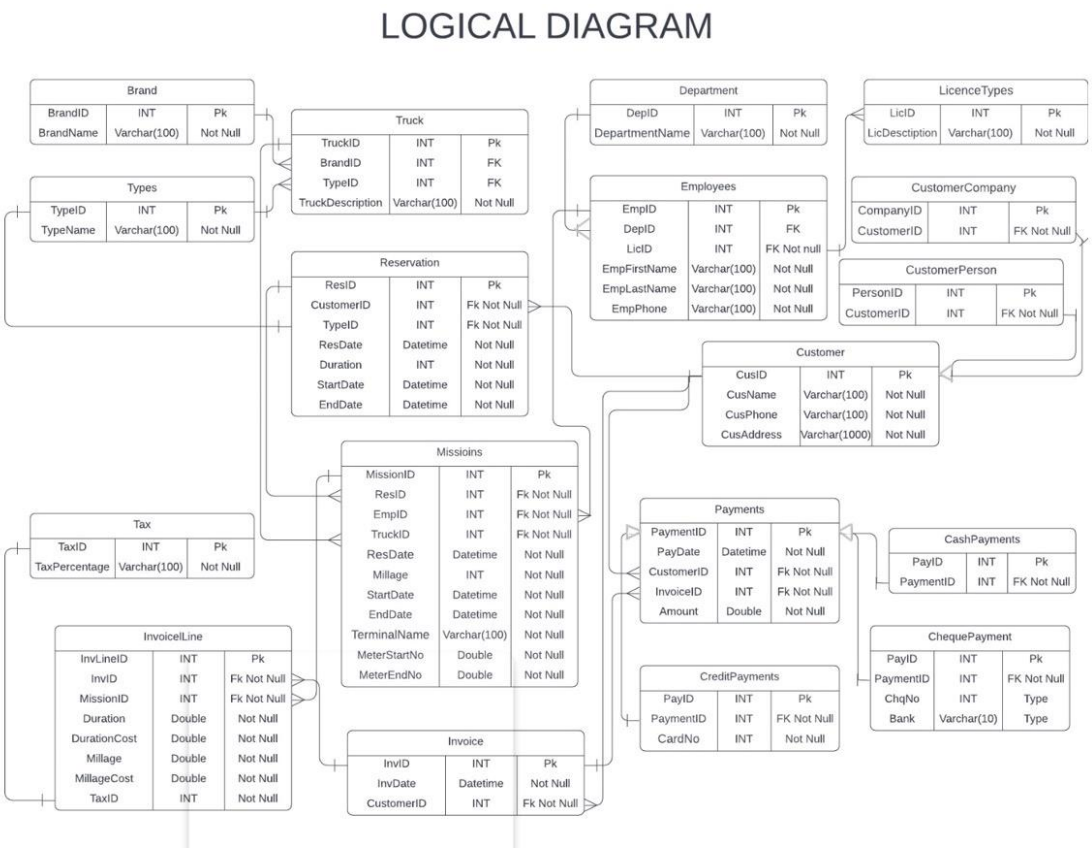
These two costs obviously depend on the rented vehicle.

The invoice payment is being made by credit card, by cash or check.

1- Conceptual Diagram



2- Logical Diagram



3- Normalization

Our logical diagram is already in 3NF as all the transitive functional dependency are removed from the tables. The tables Brand, Tax, department, LicenseType, PaymentType and invoice are already in 3nf. For Employees table to record the type of employee like driver, sales, or other departments we have created a department table and it has a foreign key DepID that is stored in the employees table same for recording the license type LicenceType table is created, and its foreign key LicID is stored in employees table to record which type of license the employee has.

4- Constraints

Brand table is consisting of two columns first is primary key and should be unique and cannot be null and the same for brand name which also cannot be null be brand name is required to identity.

Types table is consisting of two columns first is primary key and should be unique and cannot be null and the same for type of truck which also cannot be null because type name is required to identity.

Tables tax, department, licenceType, Types consist of two columns first is primary key and should be unique and cannot be null as same for description which also cannot be null because it is required to identity.

Trucks table contains four columns first is id column which should be unique and not null, 2nd for brand id which is foreign key to brand table it should also be not null, 3rd for truck type id which is foreign key to brand table it should also be null and 4th for storing truck name which also required to identify truck name having not null constraint.

Customers table have five columns which all are required to store the information related to customer for recording reservation details and send invoices to their address, so all the field have not null constraint.

Reservation and mission tables also have not null fields all the information should be provided in order to create missions, assigning drivers to them, knowing which trucks and drivers are available for a specific date.

Payments table have six columns to record the information related to customer and invoice against which payment is received so all the fields in the table are required and have not null constraint.

Invoice line table fields are not null as some of them are foreign keys linked to other tables like invoiceID and missionID to record multiple line items to invoice.

Customer Type table created with two fields one as primary key and other for type description both are required so they cannot be null.

Payments table consists of four columns first is for the primary key, 2nd is for date which is also cannot be null,3rd and 4th are foreign keys for customer table and invoice table respectively, so both are not null.

Cashpayment, creditPayments and ChequePayments also have all not null field as they all are important.

5- SQL Script of the creation of tables.

```
CREATE DATABASE proj;
drop database proj;
USE proj;
-- SQL Script of the creation of tables.
show tables;
CREATE TABLE Brand
(
BrandID INT PRIMARY KEY NOT NULL,
BrandName VARCHAR(100) NOT NULL
);

CREATE TABLE Types
(
TypeID INT PRIMARY KEY NOT NULL,
TypeName VARCHAR(100) NOT NULL
);

CREATE TABLE Tax
(
TaxID INT PRIMARY KEY NOT NULL,
TaxPercentage VARCHAR(100) NOT NULL
);

CREATE TABLE Department
(
DepartmentID INT PRIMARY KEY NOT NULL,
DepartmentName VARCHAR(100) NOT NULL
);
```

```
CREATE TABLE LicenceTypes
(
LicID INT PRIMARY KEY NOT NULL,
LicDescription VARCHAR(100) NOT NULL
);
CREATE TABLE Truck
(
TruckID INT PRIMARY KEY NOT NULL,
BrandID INT NOT NULL,
TypeID INT NOT NULL,
TruckDescripion VARCHAR(100) NOT NULL,
FOREIGN KEY (BrandID) REFERENCES Brand(BrandID)
);

CREATE TABLE Employees
(
EmpID INT PRIMARY KEY NOT NULL,
DepID INT NOT NULL,
LicID INT NOT NULL,
EmpFirstName VARCHAR(100) NOT NULL,
EmpLastName VARCHAR(100) NOT NULL,
EmpPhone VARCHAR(100) NOT NULL,
FOREIGN KEY (DepID) REFERENCES department(DepartmentID),
FOREIGN KEY (LicID) REFERENCES licencetypes(LicID)
);

CREATE TABLE Customers
(
CusID INT PRIMARY KEY NOT NULL,
CusName VARCHAR(100) NOT NULL,
CusPhone VARCHAR(100) NOT NULL,
```

```
CusAddress VARCHAR(1000) NOT NULL
);
CREATE TABLE CustomerCompany
(
CompanyID INT PRIMARY KEY NOT NULL,
CustomerID INT  NOT NULL,
FOREIGN KEY (CustomerID) REFERENCES Customers (CusID)
);
CREATE TABLE CustomerPerson
(
PersonID INT PRIMARY KEY NOT NULL,
CustomerID INT  NOT NULL,
FOREIGN KEY (CustomerID) REFERENCES Customers (CusID)
);

CREATE TABLE Reservation
(
RevID INT PRIMARY KEY NOT NULL,
CustomerID INT NOT NULL,
TypeID INT NOT NULL,
ResDate DATETIME NOT NULL,
Duration INT NOT NULL,
StartDate DATETIME NOT NULL,
EndDate DATETIME NOT NULL,
FOREIGN KEY (CustomerID) REFERENCES Customers(CusID),
FOREIGN KEY (TypeID) REFERENCES types(TypeID)
);

CREATE TABLE Mission
(
MissionID INT PRIMARY KEY NOT NULL,
```

```
ResID INT NOT NULL,
EmpID INT NOT NULL,
TruckID INT NOT NULL,
ResDate DATETIME NOT NULL,
Millage INT NOT NULL,
StartDate DATETIME NOT NULL,
EndDate DATETIME NOT NULL,
TerminalName VARCHAR(100),
MeterStartNo DOUBLE NOT NULL,
MeterEndNo DOUBLE NOT NULL,
FOREIGN KEY (EmpID) REFERENCES employees(EmpID),
FOREIGN KEY (TruckID) REFERENCES truck(truckID)
);

CREATE TABLE Invoice
(
InvID INT PRIMARY KEY NOT NULL,
InvDate DATETIME NOT NULL,
CustomerID INT NOT NULL,
FOREIGN KEY (CustomerID) REFERENCES customers(CusID)
);

CREATE TABLE InvoiceLine
(
InvLineID INT PRIMARY KEY NOT NULL,
InvID INT NOT NULL,
MessionID INT NOT NULL,
Duration INT NOT NULL,
DurationCost DOUBLE NOT NULL,
Millage DOUBLE NOT NULL,
MillageCost DOUBLE NOT NULL,
```

```
TaxID INT NOT NULL,
FOREIGN KEY (InvID) REFERENCES invoice(InvID) ,
FOREIGN KEY (MessionID) REFERENCES mission(missionID),
FOREIGN KEY (TaxID) REFERENCES tax(taxID)
);

CREATE TABLE Payments
(
PaymentID INT PRIMARY KEY NOT NULL,
PayDate DATETIME NOT NULL,
CustomerID INT NOT NULL,
InvoiceID INT NOT NULL,
Amount DOUBLE NOT NULL,
FOREIGN KEY (InvoiceID) REFERENCES invoice(InvID) ,
FOREIGN KEY (CustomerID) REFERENCES mission(missionID)
);
CREATE TABLE CashPayment
(
PayID INT PRIMARY KEY NOT NULL,
PaymentID INT NOT NULL
);

CREATE TABLE ChequePayment
(
PayID INT PRIMARY KEY NOT NULL,
PaymentID INT NOT NULL,
ChqNo INT NOT NULL,
Bank VARCHAR(100) NOT NULL,
FOREIGN KEY (PaymentID) REFERENCES Payments (PaymentID)

);
```

```
CREATE TABLE CreditPayment
(
PayID INT PRIMARY KEY NOT NULL,
PaymentID INT NOT NULL,
CardNo INT NOT NULL,
FOREIGN KEY (PaymentID) REFERENCES Payments (PaymentID)
);
```

-- 6- Insertion of data

```
-- Brand

INSERT INTO brand (BrandID ,BrandName) VALUES (1,'GMC');
INSERT INTO brand (BrandID ,BrandName) VALUES (2,'BMW');
INSERT INTO brand (BrandID ,BrandName) VALUES (3,'TOYOTA');
INSERT INTO brand (BrandID ,BrandName) VALUES (4,'HONDA');
INSERT INTO brand (BrandID ,BrandName) VALUES (5,'MERCEDES');

-- Types

INSERT INTO Types (TypeID ,TypeName ) VALUES (1,'Regular');
INSERT INTO Types (TypeID ,TypeName ) VALUES (2,'Pickup');
INSERT INTO Types (TypeID ,TypeName ) VALUES (3,'Cement');
INSERT INTO Types (TypeID ,TypeName ) VALUES (4,'Chiller');
INSERT INTO Types (TypeID ,TypeName ) VALUES (5,'Crane');

-- Tax

INSERT INTO tax (TaxID  ,TaxPercentage ) VALUES (1,10);
INSERT INTO tax (TaxID  ,TaxPercentage ) VALUES (2,5);
```

```
INSERT INTO tax (TaxID  ,TaxPercentage ) VALUES (3,3);
INSERT INTO tax (TaxID  ,TaxPercentage ) VALUES (4,15);
INSERT INTO tax (TaxID  ,TaxPercentage ) VALUES (5,12);

-- Department:

INSERT INTO department (DepartmentID   ,DepartmentName  ) VALUES (1,'Sales');
INSERT INTO department (DepartmentID   ,DepartmentName  ) VALUES (2,'Driver');
INSERT INTO department (DepartmentID   ,DepartmentName ) VALUES (3,'Accounts');
INSERT INTO department (DepartmentID   ,DepartmentName  ) VALUES (4,'HR');
INSERT INTO department (DepartmentID   ,DepartmentName  ) VALUES (5,'Finance');

-- Licencetypes
INSERT INTO licencetypes (LicID    ,LicDescription   ) VALUES (1,'tourism');
INSERT INTO licencetypes (LicID    ,LicDescription  ) VALUES (2,'heavyweight');
INSERT INTO licencetypes (LicID    ,LicDescription ) VALUES (3,'Super heavyweight');
INSERT INTO licencetypes (LicID    ,LicDescription ) VALUES (4, 'Motorbike');
INSERT INTO licencetypes (LicID    ,LicDescription ) VALUES (5,' LTV');

-- Truck

INSERT INTO truck (TruckID , BrandID ,TypeID ,TruckDescripion  ) VALUES (1,1,1,'Truck 1');
INSERT INTO truck (TruckID , BrandID ,TypeID ,TruckDescripion  ) VALUES (2,1,1,'Truck 2');
INSERT INTO truck (TruckID , BrandID ,TypeID ,TruckDescripion  ) VALUES (3,2,2,'Truck 3');
INSERT INTO truck (TruckID , BrandID ,TypeID ,TruckDescripion  ) VALUES (4,3,3,'Truck 4');
INSERT INTO truck (TruckID , BrandID ,TypeID ,TruckDescripion  ) VALUES (5,4,4,'Truck 5');
INSERT INTO truck (TruckID , BrandID ,TypeID ,TruckDescripion  ) VALUES (6,5,5,'Truck 6');
INSERT INTO truck (TruckID , BrandID ,TypeID ,TruckDescripion  ) VALUES (7,1,3,'Truck 7');
INSERT INTO truck (TruckID , BrandID ,TypeID ,TruckDescripion  ) VALUES (8,2,2,'Truck 8');
INSERT INTO truck (TruckID , BrandID ,TypeID ,TruckDescripion  ) VALUES (9,2,1,'Truck 9');
INSERT INTO truck (TruckID , BrandID ,TypeID ,TruckDescripion  ) VALUES (10,5,3,'Truck 10');
```

-- Employees

INSERT INTO employees (EmpID , DepID ,LicID ,EmpFirstName,EmpLastName ,EmpPhone) VALUES (1,1,1,'James','Mark','03556652');

INSERT INTO employees (EmpID , DepID ,LicID ,EmpFirstName,EmpLastName ,EmpPhone) VALUES (2,1,1,'Robert','','03556642');

INSERT INTO employees (EmpID , DepID ,LicID ,EmpFirstName,EmpLastName ,EmpPhone) VALUES (3,2,1,'John','Kevin','03556122');

INSERT INTO employees (EmpID , DepID ,LicID ,EmpFirstName,EmpLastName ,EmpPhone) VALUES (4,2,2,'William','Donald','03556633');

INSERT INTO employees (EmpID , DepID ,LicID ,EmpFirstName,EmpLastName ,EmpPhone) VALUES (5,2,3,'David','Ryan','03556655');

INSERT INTO employees (EmpID , DepID ,LicID ,EmpFirstName,EmpLastName ,EmpPhone) VALUES (6,2,2,'Richard','','03556677');

INSERT INTO employees (EmpID , DepID ,LicID ,EmpFirstName,EmpLastName ,EmpPhone) VALUES (7,2,3,'Joseph','','03556699');

INSERT INTO employees (EmpID , DepID ,LicID ,EmpFirstName,EmpLastName ,EmpPhone) VALUES (8,2,2,'Thomas','Donald','0355822');

INSERT INTO employees (EmpID , DepID ,LicID ,EmpFirstName,EmpLastName ,EmpPhone) VALUES (9,3,1,'John','Ryan','03556122');

INSERT INTO employees (EmpID , DepID ,LicID ,EmpFirstName,EmpLastName ,EmpPhone) VALUES (10,5,1,'William','Kevin','03556123');

-- Customers

INSERT INTO customers (CusID ,CusName ,CusPhone ,CusAddress) VALUES(1,'Gary','366661','Street 1');

INSERT INTO customers (CusID ,CusName ,CusPhone ,CusAddress) VALUES(2 ,'Nicholas','314241','Street 2');

INSERT INTO customers (CusID ,CusName ,CusPhone ,CusAddress) VALUES(3 ,'Eric','312311','Street 1');

INSERT INTO customers (CusID ,CusName ,CusPhone ,CusAddress) VALUES(4 ,'Stephen','155211','Street 3');

```
INSERT INTO customers (CusID ,CusName ,CusPhone ,CusAddress ) VALUES(5 ,'Larry','142161','Street 5');

INSERT INTO customers (CusID ,CusName ,CusPhone ,CusAddress ) VALUES(6 ,'Ethan','1421161','Street 6');

INSERT INTO customers (CusID ,CusName ,CusPhone ,CusAddress ) VALUES(7 ,'William','14142161','Street 7');

INSERT INTO customers (CusID ,CusName ,CusPhone ,CusAddress ) VALUES(8 ,'David','14252161','Street 8');

INSERT INTO customers (CusID ,CusName ,CusPhone ,CusAddress ) VALUES(9 ,'Richard','14244161','Street 9');

INSERT INTO customers (CusID ,CusName ,CusPhone ,CusAddress ) VALUES(10 ,'Robert','14472161','Street 10');

INSERT INTO customers (CusID ,CusName ,CusPhone ,CusAddress ) VALUES(11 ,'Daniel','1427161','Street 11');


-- CustomerCompany


INSERT INTO CustomerCompany (CompanyID  ,CustomerID ) VALUES(1,1);

INSERT INTO CustomerCompany (CompanyID  ,CustomerID ) VALUES(2 ,2);

INSERT INTO customercompany (CompanyID  ,CustomerID  ) VALUES(3 ,3);

INSERT INTO customercompany (CompanyID  ,CustomerID  ) VALUES(4 ,6);

INSERT INTO customercompany (CompanyID  ,CustomerID  ) VALUES(5 ,7);


-- CustomerPerson


INSERT INTO customerperson (PersonID   ,CustomerID ) VALUES(1,4);

INSERT INTO CustomerPerson (PersonID   ,CustomerID ) VALUES(2 ,5);

INSERT INTO CustomerPerson (PersonID   ,CustomerID ) VALUES(3 ,8);

INSERT INTO CustomerPerson (PersonID   ,CustomerID ) VALUES(4 ,9);

INSERT INTO CustomerPerson (PersonID   ,CustomerID ) VALUES(5 ,10);


-- Reservation
```

```
INSERT INTO reservation (RevID  ,CustomerID ,TypeID ,ResDate ,Duration ,StartDate ,EndDate )
VALUES (1,1,1,'2022-02-01',2,'2022-02-01','2022-02-02');
INSERT INTO reservation (RevID  ,CustomerID ,TypeID ,ResDate ,Duration ,StartDate ,EndDate )
VALUES (2,1,2,'2022-02-12',1,'2022-02-17','2022-02-17');
INSERT INTO reservation (RevID  ,CustomerID ,TypeID ,ResDate ,Duration ,StartDate ,EndDate )
VALUES (3,3,3,'2022-03-01',5,'2022-03-14','2022-03-18');
INSERT INTO reservation (RevID  ,CustomerID ,TypeID ,ResDate ,Duration ,StartDate ,EndDate )
VALUES (4,4,2,'2022-03-01',1,'2022-03-31','2022-03-31');
INSERT INTO reservation (RevID  ,CustomerID ,TypeID ,ResDate ,Duration ,StartDate ,EndDate )
VALUES (5,5,2,'2022-03-15',1,'2022-03-18','2022-03-18');
INSERT INTO reservation (RevID  ,CustomerID ,TypeID ,ResDate ,Duration ,StartDate ,EndDate )
VALUES (6,4,2,'2022-03-17',3,'2022-02-17','2022-03-19');
INSERT INTO reservation (RevID  ,CustomerID ,TypeID ,ResDate ,Duration ,StartDate ,EndDate )
VALUES (7,3,2,'2022-03-01',1,'2022-03-31','2022-03-31');
INSERT INTO reservation (RevID  ,CustomerID ,TypeID ,ResDate ,Duration ,StartDate ,EndDate )
VALUES (8,3,1,'2022-03-02',1,'2022-03-05','2022-03-05');
INSERT INTO reservation (RevID  ,CustomerID ,TypeID ,ResDate ,Duration ,StartDate ,EndDate )
VALUES (9,3,3,'2022-03-01',1,'2022-03-11','2022-03-11');
INSERT INTO reservation (RevID  ,CustomerID ,TypeID ,ResDate ,Duration ,StartDate ,EndDate )
VALUES (10,2,3,'2022-03-21',1,'2022-03-21','2022-03-21');

-- Mission
INSERT INTO Mission (MissionID ,ResID ,EmpID ,TruckID ,ResDate ,Millage ,StartDate ,EndDate
,TerminalName ,MeterStartNo ,MeterEndNo ) VALUES(1,1,3,1,'2022-02-01',5000,'2022-02-01','2022-
02-02','Terminal 1',15363,20363);
INSERT INTO Mission (MissionID ,ResID ,EmpID ,TruckID ,ResDate ,Millage ,StartDate ,EndDate
,TerminalName ,MeterStartNo ,MeterEndNo ) VALUES(3,3,5,3,'2022-03-01',7100,'2022-03-14','2022-
03-18','Terminal 2',7380,14480);

INSERT INTO Mission (MissionID ,ResID ,EmpID ,TruckID ,ResDate ,Millage ,StartDate ,EndDate
,TerminalName ,MeterStartNo ,MeterEndNo ) VALUES(4,4,6,2,'2022-03-01',7500,'2022-03-14','2022-
03-18','Terminal 2',150015,157515);
```

INSERT INTO Mission (MissionID ,ResID ,EmpID ,TruckID ,ResDate ,Millage ,StartDate ,EndDate ,TerminalName ,MeterStartNo ,MeterEndNo) VALUES(5,5,3,2,'2022-03-15',2500,'2022-03-31','2022-03-31','Terminal 2',252015,254515);

INSERT INTO Mission (MissionID ,ResID ,EmpID ,TruckID ,ResDate ,Millage ,StartDate ,EndDate ,TerminalName ,MeterStartNo ,MeterEndNo) VALUES(6,6,7,2,'2022-03-17',1000,'2022-03-18','2022-03-18','Terminal 4',252015,253015);

INSERT INTO Mission (MissionID ,ResID ,EmpID ,TruckID ,ResDate ,Millage ,StartDate ,EndDate ,TerminalName ,MeterStartNo ,MeterEndNo) VALUES(7,7,3,2,'2022-03-01',1000,'2022-02-17','2022-03-19','Terminal 4',252015,253015);

INSERT INTO Mission (MissionID ,ResID ,EmpID ,TruckID ,ResDate ,Millage ,StartDate ,EndDate ,TerminalName ,MeterStartNo ,MeterEndNo) VALUES(8,8,5,1,'2022-03-01',2000,'2022-03-31','2022-03-31','Terminal 3',253015,255015);

INSERT INTO Mission (MissionID ,ResID ,EmpID ,TruckID ,ResDate ,Millage ,StartDate ,EndDate ,TerminalName ,MeterStartNo ,MeterEndNo) VALUES(9,9,7,3,'2022-03-02',2000,'2022-03-11','2022-03-11','Terminal 3',255015,257015);

INSERT INTO Mission (MissionID ,ResID ,EmpID ,TruckID ,ResDate ,Millage ,StartDate ,EndDate ,TerminalName ,MeterStartNo ,MeterEndNo) VALUES(10,10,3,3,'2022-03-01',500,'2022-03-21','2022-03-21','Terminal 1',252365,252865);

-- Invoice

INSERT INTO Invoice (InvID ,InvDate ,CustomerID)VALUES(1,'2022-02-01',1);

INSERT INTO Invoice (InvID ,InvDate ,CustomerID)VALUES(2,'2022-04-01',1);

INSERT INTO Invoice (InvID ,InvDate ,CustomerID)VALUES(3,'2022-04-03',3);

INSERT INTO Invoice (InvID ,InvDate ,CustomerID)VALUES(4,'2022-04-04',4);

INSERT INTO Invoice (InvID ,InvDate ,CustomerID)VALUES(5,'2022-04-06',5);

INSERT INTO Invoice (InvID ,InvDate ,CustomerID)VALUES(6,'2022-04-06',5);

-- InvoiceLine

INSERT INTO InvoiceLine (InvLineID ,InvID ,MessionID ,Duration ,DurationCost ,Millage ,MillageCost ,TaxID)

```
VALUES(1,1,1,2,1000,5000,1000,1);


INSERT INTO InvoiceLine (InvLineID ,InvID ,MessionID ,Duration ,DurationCost ,Millage ,MillageCost
,TaxID )

VALUES(2,1,7,1,300,7100,3000,2);


INSERT INTO InvoiceLine (InvLineID ,InvID ,MessionID ,Duration ,DurationCost ,Millage ,MillageCost
,TaxID )

VALUES(3,2,3,5,300,7500,3500,1);


INSERT INTO InvoiceLine (InvLineID ,InvID ,MessionID ,Duration ,DurationCost ,Millage ,MillageCost
,TaxID )

VALUES(4,3,4,1,300,2500,2000,3);


INSERT INTO InvoiceLine (InvLineID ,InvID ,MessionID ,Duration ,DurationCost ,Millage ,MillageCost
,TaxID )

VALUES(5,4,5,1,300,1000,800,3);


-- Payment


INSERT INTO Payments (PaymentID ,PayDate ,CustomerID ,InvoiceID,Amount )

VALUES(1,'2022-04-10',1,1,500);

INSERT INTO Payments (PaymentID ,PayDate ,CustomerID ,InvoiceID ,Amount  )

VALUES(2,'2022-04-10',1,2,2000);

INSERT INTO Payments (PaymentID ,PayDate ,CustomerID ,InvoiceID,Amount  )

VALUES(3,'2022-04-10',3,3,600);

INSERT INTO Payments (PaymentID ,PayDate ,CustomerID ,InvoiceID ,Amount  )

VALUES(4,'2022-04-10',4,4,700);

INSERT INTO Payments (PaymentID ,PayDate ,CustomerID ,InvoiceID,Amount  )

VALUES(5,'2022-04-10',5,5,1200);

INSERT INTO Payments (PaymentID ,PayDate ,CustomerID ,InvoiceID,Amount  )

VALUES(6,'2022-04-10',5,5,200);
```

```
INSERT INTO Payments (PaymentID ,PayDate ,CustomerID ,InvoiceID,Amount  )
VALUES(7,'2022-04-11',6,5,100);
INSERT INTO Payments (PaymentID ,PayDate ,CustomerID ,InvoiceID,Amount  )
VALUES(8,'2022-04-12',7,5,100);
INSERT INTO Payments (PaymentID ,PayDate ,CustomerID ,InvoiceID,Amount  )
VALUES(9,'2022-04-13',8,5,300);
INSERT INTO Payments (PaymentID ,PayDate ,CustomerID ,InvoiceID,Amount  )
VALUES(10,'2022-04-14',9,5,500);

-- CashPayment

Insert INTO cashPayment (PayID, PaymentID) VALUES (1,1);
Insert INTO cashPayment (PayID, PaymentID) VALUES (2,2);
Insert INTO cashPayment (PayID, PaymentID) VALUES (3,3);
Insert INTO cashPayment (PayID, PaymentID) VALUES (4,4);
Insert INTO cashPayment (PayID, PaymentID) VALUES (5,5);

-- CreditPayments

INSERT INTO creditPayment (PayID, PaymentID, CardNo) Values (1,1,256656);
INSERT INTO creditPayment (PayID, PaymentID, CardNo) Values (2,2,2123559);
INSERT INTO creditPayment (PayID, PaymentID, CardNo) Values (3,3,153657);
INSERT INTO creditPayment (PayID, PaymentID, CardNo) Values (4,6,123537);
INSERT INTO creditPayment (PayID, PaymentID, CardNo) Values (5,7,365367);
INSERT INTO creditPayment (PayID, PaymentID, CardNo) Values (6,9,555557);

-- ChequePayments

INSERT INTO chequePayment (PayID, PaymentID, ChqNo, Bank ) Values (1,3,53535,'abc');
INSERT INTO chequePayment (PayID, PaymentID, ChqNo, Bank ) Values (2,4,353653,'abc');
INSERT INTO chequePayment (PayID, PaymentID, ChqNo, Bank ) Values (3,5,899653,'abc');
```

```
INSERT INTO chequePayment (PayID, PaymentID, ChqNo, Bank ) Values (4,6,999993,'xyz');

INSERT INTO chequePayment (PayID, PaymentID, ChqNo, Bank ) Values (5,8,888853,'xyz');

INSERT INTO chequePayment (PayID, PaymentID, ChqNo, Bank ) Values (6,10,77753,'xyz');
```

-- 7- Queries

```
-- 1-List of customers that are businesses (Enterprises or Companies).

SELECT customers.CusName FROM customers INNER JOIN

customercompany on customers.CusID = customercompany.CustomerID;


-- 2-List of reservations whose reservation number is greater than 1.

SELECT * from reservation where RevID > 1;


-- 3- List of drivers and vehicles having participated in at least one mission.

SELECT EmpFirstName,EmpLastName, TruckDescripion FROM employees INNER JOIN mission

on mission.EmpID = employees.EmpID

INNER JOIN truck on Mission.TruckID = truck.TruckID

where employees.DepID = 2

GROUP BY EmpFirstName,EmpLastName

HAVING COUNT(employees.EmpID) >=1;


-- 4. List of missions between March 11, 2022 and March 18, 2022 as well

-- as the drivers and vehicles participating in these missions.

SELECT MissionID, employees.EmpFirstName,employees.EmpLastName, TruckDescripion from
mission inner join employees

on mission.EmpID = employees.EmpID

INNER JOIN truck on Mission.TruckID = truck.TruckID

 WHERE StartDate BETWEEN

'2022-03-11' and '2022-03-18';


-- 5. The list of customers who have not paid their invoices.
```

```
SELECT invoice.InvID from invoice LEFT JOIN payments
on invoice.InvID = payments.InvoiceID WHERE payments.InvoiceID is null;

-- 6. List of drivers who have driven 'GMC' brand vehicles.
SELECT distinct employees.EmpFirstName,employees.EmpLastName from employees
INNER JOIN mission on Mission.EmpID = employees.EmpID
INNER JOIN truck on Mission.TruckID = truck.TruckID
INNER JOIN brand on brand.BrandID = truck.BrandID
WHERE brand.BrandName = 'GMC';

-- 7. Which customers have invoices greater than $1000?
SELECT customers.CusName,SUM(invoiceline.MillageCost)'Cost' FROM invoice INNER JOIN
customers
on customers.CusID = invoice.CustomerID
INNER JOIN invoiceline on invoiceline.InvID = invoice.InvID
GROUP by customers.CusName
HAVING SUM(invoiceline.MillageCost) >1000;

-- 8. List of customers with their number of associated invoices.
SELECT customers.CusName,invoice.InvID FROM invoice INNER JOIN customers
on customers.CusID = invoice.CustomerID;

-- 9. What are the last names and first names of the drivers who have a
-- mission between the following dates: February 1, 2022 and March 31,
-- 2022 whose mileage (number of kilometers traveled) is more than 7000 km?

SELECT employees.EmpFirstName,employees.EmpLastName FROM employees
INNER JOIN mission on mission.EmpID = employees.EmpID
WHERE mission.StartDate BETWEEN '2022-02-01' and '2022-03-31'
HAVING SUM(Millage) > 7000;
```

Note: workbench was used to trial the queries.