UNIVERSITY OF INFORMATION TECHNOLOGY

FACULTY OF INFORMATION SYSTEMS

LAB REPORT

AMBULANCE MANAGEMENT SYSTEMS

Class	CS5423.O21.CTTT	
Subject	Principle of database systems	
Lecture	Ph.D. Nguyen Thanh Binh	
Student	Pham Quang Dai Phuc	22521137
	Le Minh Sang	22521240

Thu Duc City, 2024

TABLE OF CONTENT

WORK DIVISION 1

I. Describe the problem 2

II. Describe the system's functions 2

III. ERD Model 4

IV. Relation Schema 5

V. SQL query 6

VI. Experimentalize 7

WORK DIVISION

No	Student	Task
1	Pham Quang Dai Phuc	1. Design ERD 2. Write a description of the problems 3. Write a description of the function's system
2	Le Minh Sang	1. Design ERD 2. Design RD 3. Write SQL Query 4. Write Experimentalize

I. Describe the problem

The company meticulously maintains records of its customers, assigning each a unique identifier (CustomerID) to ensure tracking accuracy. Customer details, such as **name**, **email**, **address**, and **phone number** are recorded for efficient communication and management.

To facilitate its operations, the company operates a diverse fleet of vehicles, each tracked with a unique identifier (VehicleID). Records for each vehicle including **license plate**, **model**, **capacity**, and current availability **status**, ensuring optimal allocation and utilization of resources.

For each customer request, staff will create a booking, each uniquely identified by a BookingID. These bookings capture crucial details like the involved customer and **vehicle**, **booking date**, and the **type of booking**, whether it's a one-time trip or a daily arrangement, streamlining the reservation process.

For every booking, the company generates an invoice, each tagged with a unique identifier (InvoiceID). These invoices provide comprehensive breakdowns of charges, including the **booking reference**, **issue date, total amount** due, and **price** per unit.

The company offers various package options, each identified by a unique PackageID. These packages delineate **unit types** ('Km' or 'Day'), **package types** ('Trip' or 'Daily'), **unit** amounts, and associated **package prices**, providing customers with flexible and cost-effective booking options.

To ensure organizational efficiency, the company is structured into distinct departments, each meticulously assigned a unique identifier (DepartmentID). Department records encompass vital information like department **names and locations**, facilitating seamless coordination and management across different divisions.

The company's workforce comprises employees, each assigned a unique EmployeeID and belonging to a department. Employee records include essential details such as **names, roles, phone numbers, and emails**.

Assets crucial to the company's operations, including vehicles and medical equipment, are meticulously tracked with unique identifiers (AssetID). Asset records encompass key details such as **names, types, values**, and **purchase dates**, and are managed by a department.

To maintain financial transparency, the company records its expenses, each tagged with a unique ExpenseID and linked to related assets. Expense details, including types ('Fixed' or 'Variable'), **descriptions, amounts, and dates**, are recorded, facilitating accurate financial reporting and analysis.

Finally, the company manages schedules for its employees and vehicles to ensure efficient utilization and coordination. Each schedule is assigned a unique ScheduleID and records essential information such as dates, start and end times, and descriptions, with references to the EmployeeID and VehicleID involved, ensuring seamless operation and resource allocation.

II. Describe the system's functions

Staff management:

- *Search for staff*: search for full details about the information of the company, including name, phone, address, and role.
- *Add new staff member:* Add information for the staff who join the company, which includes name, email, address, and phone number.

- *Delete staff member*: For people who left the organization, this system allows to deletion of all the information of this employee.
- *Update Information*: This system can update the information of the company's employees, including name, address, role, and phone number.

Booking and Schedule Management:

- *Search for booking*: search for full details about who makes this booking, booking day and the type of booking, etc.
- *Update Information*: Update new information, including the destination, phone,
- *Delete trip*: delete the information of trips that are canceled.
- *Add trip*: When booking is completed, a trip with some information will be added to the system.

Financial Management:

- *Calculate the total cost*: Calculate the total cost of each type of booking.
- *Calculate the expense*: Calculate the sum of money to maintain the company, including fixed costs like gas and the incurred cost.

Asset Management:

- *Search for an asset*: search for details about the asset, including the amount of this asset which are ready to use.
- *Add and delete asset*: add and delete the asset, the information includes the name, purchase date, and the quantity of this asset.

III. ERD Model

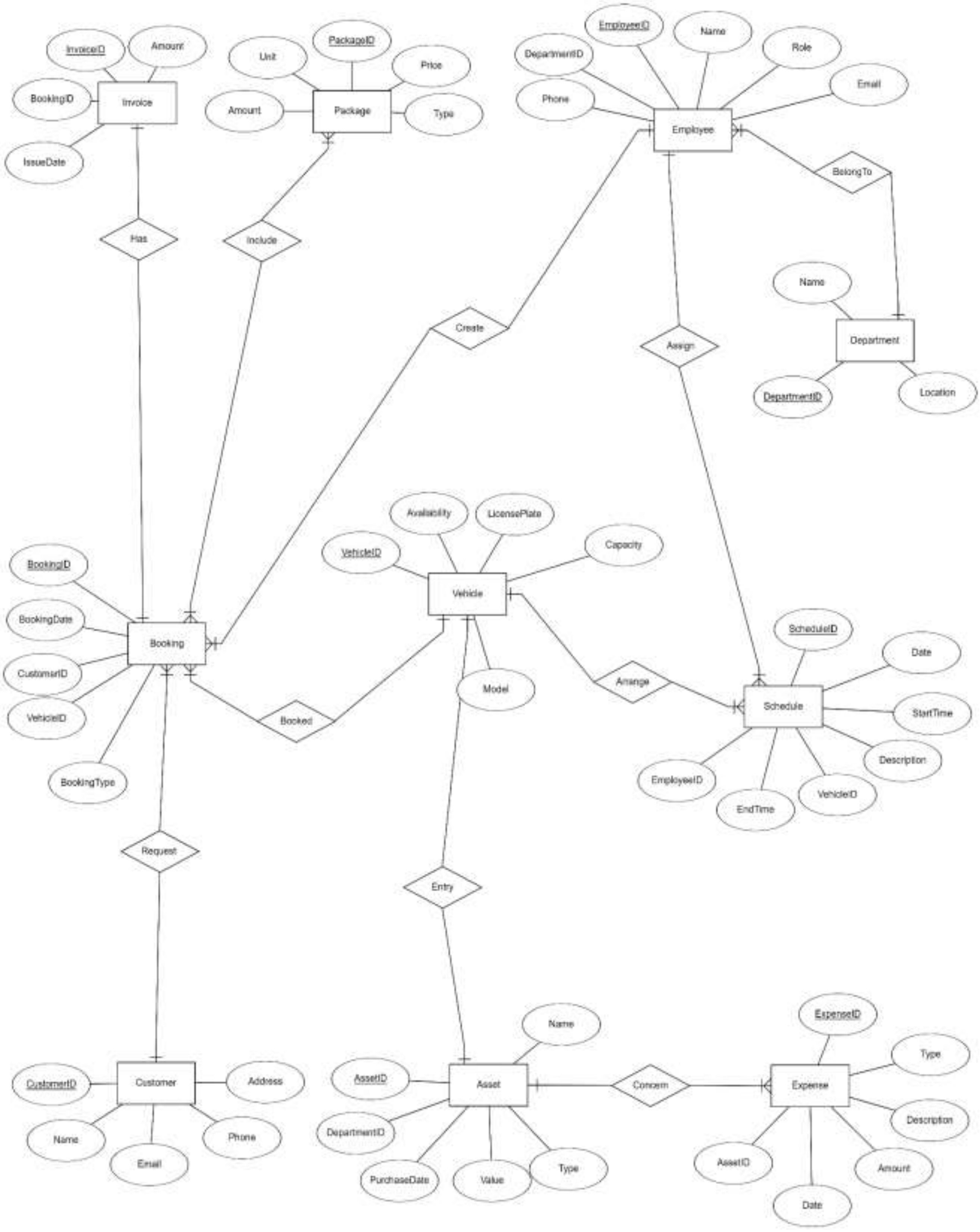
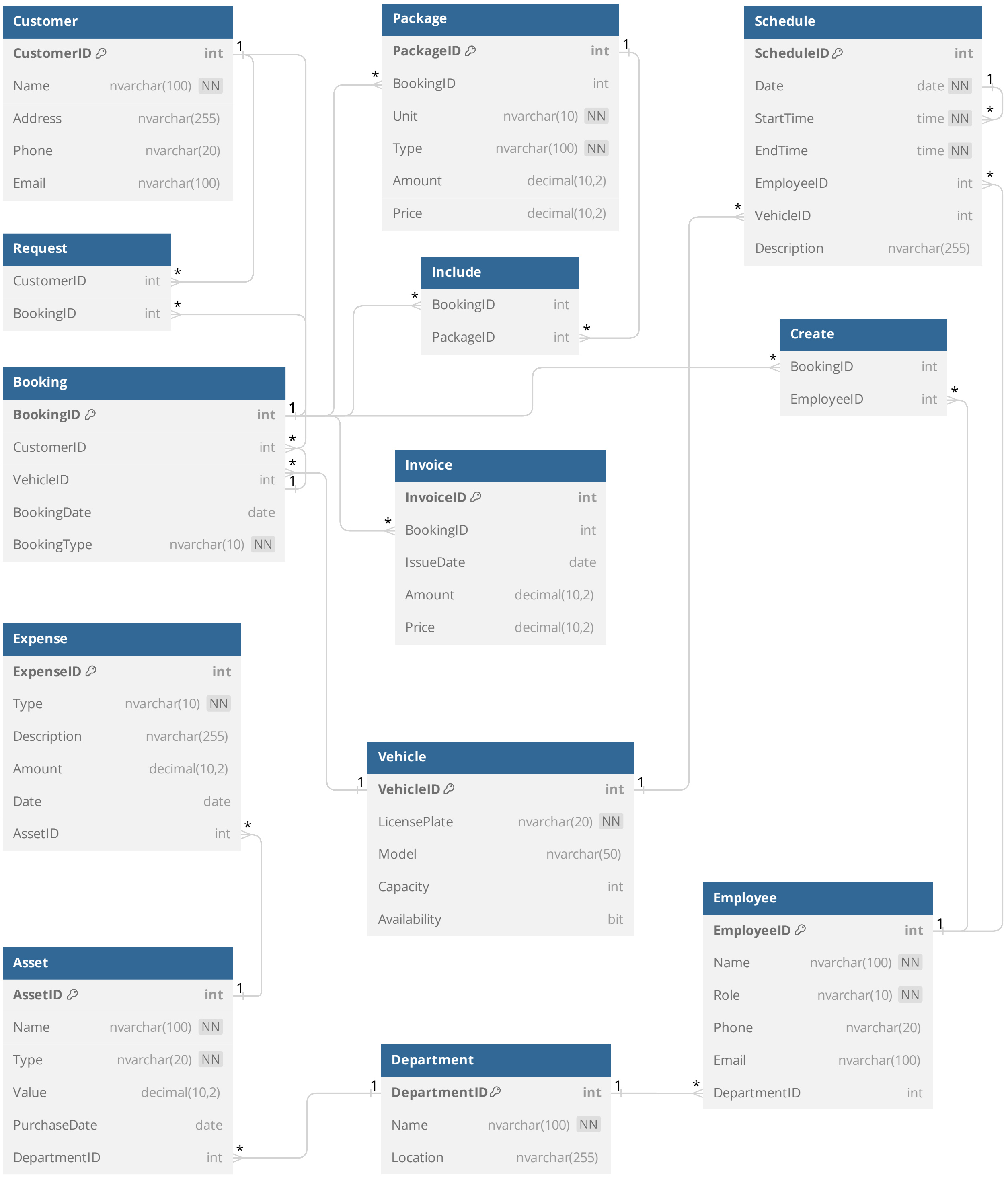


Figure 1. ERD model of Ambulance Management Systems

IV. Relation Schema



Customer		
CustomerID	**int**	
Name	nvarchar(100)	NN
Address	nvarchar(255)	
Phone	nvarchar(20)	
Email	nvarchar(100)	

Request	
CustomerID	int
BookingID	int

Booking		
BookingID	**int**	
CustomerID	int	
VehicleID	int	
BookingDate	date	
BookingType	nvarchar(10)	NN

Expense		
ExpenseID	**int**	
Type	nvarchar(10)	NN
Description	nvarchar(255)	
Amount	decimal(10,2)	
Date	date	
AssetID	int	

Asset		
AssetID	**int**	
Name	nvarchar(100)	NN
Type	nvarchar(20)	NN
Value	decimal(10,2)	
PurchaseDate	date	
DepartmentID	int	

Package		
PackageID	**int**	
BookingID	int	
Unit	nvarchar(10)	NN
Type	nvarchar(100)	NN
Amount	decimal(10,2)	
Price	decimal(10,2)	

Include	
BookingID	int
PackageID	int

Invoice	
InvoiceID	**int**
BookingID	int
IssueDate	date
Amount	decimal(10,2)
Price	decimal(10,2)

Vehicle		
VehicleID	**int**	
LicensePlate	nvarchar(20)	NN
Model	nvarchar(50)	
Capacity	int	
Availability	bit	

Department		
DepartmentID	**int**	
Name	nvarchar(100)	NN
Location	nvarchar(255)	

Schedule		
ScheduleID	**int**	
Date	date	NN
StartTime	time	NN
EndTime	time	NN
EmployeeID	int	
VehicleID	int	
Description	nvarchar(255)	

Create	
BookingID	int
EmployeeID	int

Employee		
EmployeeID	**int**	
Name	nvarchar(100)	NN
Role	nvarchar(10)	NN
Phone	nvarchar(20)	
Email	nvarchar(100)	
DepartmentID	int	

Figure 2. RD of Ambulance Management Systems

V. SQL query

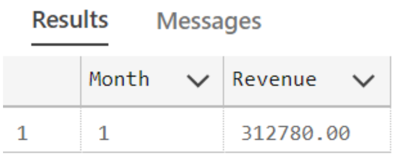
```
-- Create a new database called AmbulanceService
CREATE DATABASE AmbulanceService;
USE AmbulanceService;
-- Create the Customer table
CREATE TABLE Customer (
    CustomerID INT PRIMARY KEY IDENTITY(1,1),
    Name NVARCHAR(100) NOT NULL,
    Address NVARCHAR(255),
    Phone NVARCHAR(20),
    Email NVARCHAR(100) );
-- Create the Vehicle table
CREATE TABLE Vehicle (
    VehicleID INT PRIMARY KEY IDENTITY(1,1),
    LicensePlate NVARCHAR(20) NOT NULL,
    Model NVARCHAR(50),
    Capacity INT,
    Availability BIT );
-- Create the Booking table
CREATE TABLE Booking (
    BookingID INT PRIMARY KEY,
    CustomerID INT,
    VehicleID INT,
    BookingDate DATE,
    BookingType NVARCHAR(10) NOT NULL CHECK (BookingType IN ('Trip', 'Daily')),
    FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),
    FOREIGN KEY (VehicleID) REFERENCES Vehicle(VehicleID) );
-- Create the Invoice table
CREATE TABLE Invoice (
    InvoiceID INT PRIMARY KEY IDENTITY(1,1),
    BookingID INT,
    IssueDate DATE,
    Amount DECIMAL(10, 2),
    FOREIGN KEY (BookingID) REFERENCES Booking(BookingID) );
-- Create the Package table
CREATE TABLE Package (
    PackageID INT PRIMARY KEY IDENTITY(1,1),
    BookingID INT,
    Unit NVARCHAR(10) NOT NULL CHECK (Unit IN ('Km', 'Day')),
    Type NVARCHAR(100) NOT NULL CHECK (Type IN ('Trip', 'Daily')),
    Amount DECIMAL(10, 2),
    Price DECIMAL(10, 2),
    FOREIGN KEY (BookingID) REFERENCES Booking(BookingID) );
-- Create the Department table
CREATE TABLE Department (
    DepartmentID INT PRIMARY KEY IDENTITY(1,1),
    Name NVARCHAR(100) NOT NULL,
    Location NVARCHAR(255) );
-- Create the Employee table
CREATE TABLE Employee (
    EmployeeID INT PRIMARY KEY IDENTITY(1,1),
    Name NVARCHAR(100) NOT NULL,
    Role NVARCHAR(10) NOT NULL CHECK (Role IN ('Operator', 'Driver')),
    Phone NVARCHAR(20),
    Email NVARCHAR(100),
    DepartmentID INT,
    FOREIGN KEY (DepartmentID) REFERENCES Department(DepartmentID) );
-- Create the BookingEmployee table
CREATE TABLE BookingEmployee (
    BookingID INT,
    EmployeeID INT,
    PRIMARY KEY (BookingID, EmployeeID),
    FOREIGN KEY (BookingID) REFERENCES Booking(BookingID),
    FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID) );
-- Create the Asset table
CREATE TABLE Asset (
    AssetID INT PRIMARY KEY IDENTITY(1,1),
    Name NVARCHAR(100) NOT NULL,
    Type NVARCHAR(20) NOT NULL CHECK (Type IN ('Vehicle', 'MedicalEquipment')),
    Value DECIMAL(10, 2),
    PurchaseDate DATE,
    DepartmentID INT,
    FOREIGN KEY (DepartmentID) REFERENCES Department(DepartmentID) );
-- Create the Expense table
CREATE TABLE Expense (
    ExpenseID INT PRIMARY KEY IDENTITY(1,1),
    Type NVARCHAR(10) NOT NULL CHECK (Type IN ('Fixed', 'Variable')),
    Description NVARCHAR(255),
    Amount DECIMAL(10, 2),
    Date DATE,
    AssetID INT,
    FOREIGN KEY (AssetID) REFERENCES Asset(AssetID) );
-- Create the Schedule table
CREATE TABLE Schedule (
    ScheduleID INT PRIMARY KEY IDENTITY(1,1),
    Date DATE NOT NULL,
    StartTime TIME NOT NULL,
    EndTime TIME NOT NULL,
    EmployeeID INT,
    VehicleID INT,
    Description NVARCHAR(255),
    FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID),
    FOREIGN KEY (VehicleID) REFERENCES Vehicle(VehicleID) );
```

VI. Experimentalize

There are some example queries to extract important information from the Ambulance Service database:

- Find the month with the highest revenue:

```
SELECT TOP 1 MONTH(IssueDate) AS Month, SUM(Amount) AS Revenue
FROM Invoice
GROUP BY MONTH(IssueDate)
ORDER BY SUM(Amount) DESC;
```

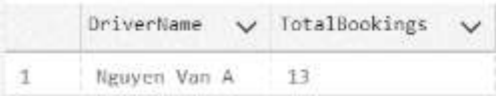


	Month	Revenue
1	1	312780.00

Figure 3. The month has the highest revenue

- Find the driver who has the highest total fare

```
SELECT TOP 1 E.Name AS DriverName, COUNT(BE.BookingID) AS TotalBookings
FROM Employee E
INNER JOIN BookingEmployee BE ON E.EmployeeID = BE.EmployeeID
GROUP BY E.Name
ORDER BY TotalBookings DESC
```

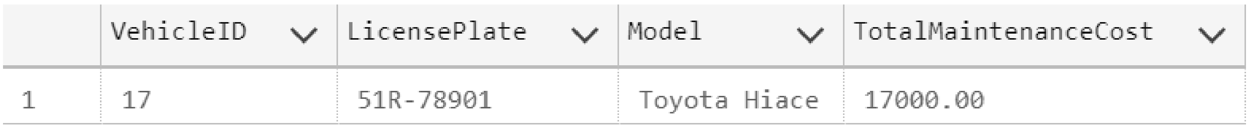


	DriverName	TotalBookings
1	Nguyen Van A	13

Figure 4. The driver has the highest total fare

- Find the vehicle that has the highest total maintenance cost

```
SELECT TOP 1 V.VehicleID, V.LicensePlate, V.Model, SUM(E.Amount) AS TotalMaintenanceCost
FROM Vehicle V
INNER JOIN Expense E ON V.VehicleID = E.AssetID
WHERE E.Description = 'Maintenance'
GROUP BY V.VehicleID, V.LicensePlate, V.Model
ORDER BY TotalMaintenanceCost DESC
```



	VehicleID	LicensePlate	Model	TotalMaintenanceCost
1	17	51R-78901	Toyota Hiace	17000.00

Figure 5. The vehicle has the highest total maintenance cost