

Car Rental Final Report

INTRODUCTION

To handle customer data, rental transactions, and vehicle availability, the automobile rental sector needs an effective database system. The main goal of our project is to create and implement a structured database that guarantees smooth operations, accurate data retrieval, and peak performance. The design process, implementation details, and overall assessment of our database system are all thoroughly examined in this report.

Background of the organization: Rentalcars.com

Rentalcars.com is a globally recognized leader in the car rental industry and a key subsidiary of Booking Holdings Inc., which also owns major travel platforms such as Kayak, Booking.com, and Priceline. Founded in 2004 and headquartered in Manchester, UK, the company has grown into one of the most trusted car rental aggregators, serving customers in over 160 countries through partnerships with more than 60,000 rental locations. Rentalcars.com simplifies the car rental process by offering a transparent, user-friendly booking platform with competitive pricing, multilingual customer support, and financial protection. The platform caters to a wide range of travelers, from budget-conscious tourists to business professionals seeking premium vehicles, ensuring a seamless rental experience worldwide.

Objective

The primary objective of Rentalcars.com is to deliver a hassle-free, cost-effective, and reliable car rental service that meets the diverse needs of global customers. The platform aims to provide an extensive selection of vehicles, including economy cars, SUVs, luxury models, and specialty vehicles, with clear pricing and no hidden fees. By enabling advance bookings with flexible cancellation policies, Rentalcars.com ensures convenience and peace of mind for travelers. Additionally, the company prioritizes customer security through verified payment methods, insurance options, and fraud protection. To enhance accessibility, the platform supports multiple languages and offers 24/7 customer assistance. Another critical goal is maintaining real-time inventory tracking to prevent overbooking and optimize fleet management, ensuring that customers always have access to available vehicles. Through these efforts, Rentalcars.com strives to uphold its reputation as the leading car rental aggregator in the travel industry.

Scope

The proposed database system for Rentalcars.com will streamline operations and enhance customer experience with secure user authentication, allowing customers to register, save payment details, and input rental preferences like locations, dates, and vehicle type. It generates real-time search results with dynamic pricing, enabling a

smooth checkout process where users review terms, select insurance, and complete payments automatically sending confirmations and reminders. The backend monitors fleet availability in real time, prevents overbooking, tracks maintenance, and flags low inventory. A feedback module collects reviews, while loyalty rewards and analytics tools track trends and supplier performance, ensuring data driven decisions. This integrated system delivers a seamless experience for users and administrators, supporting Rentalcars.com's commitment to efficient, high quality service.

Functional requirements of the application

The Rentalcars.com database system was designed to fulfill three core functional requirements: search/retrieve, book/purchase, and manage operations. For search and retrieval, the system enables users to query real-time vehicle inventory based on location, date, and vehicle type while displaying dynamic rental pricing and accessing customer profile information. The booking and purchase functionality features a secure checkout process with integrated payment processing, optional insurance add-ons, and automated email/SMS confirmation notifications to ensure a seamless transaction experience. On the administrative side, the management capabilities allow Rentalcars.com staff to monitor fleet availability across all locations, track maintenance schedules, and analyze customer feedback through comprehensive analytics. Together, these interconnected functions create a robust platform that serves both customer needs and business operations efficiently.

DATABASE MODELLING

External Views

Below are some external views of the site, rentalcars.com as we provide images of the entire booking process. We start with selecting where and when a customer would like to book, for this example we selected Toronto Pearson Airport, from April 3rd to April 5th:

The screenshot shows the Rentalcars.com by Booking.com homepage. The search bar at the top has 'Toronto Pearson International Airport (YYZ), Toronto, ...' entered. Below the search bar, there are five input fields: 'Pick-up location' (selected), 'Pick-up date' (Wed, Apr 2), 'Time' (10:00 a.m.), 'Drop-off date' (Sat, Apr 5), and 'Time' (10:00 a.m.). A green 'Search' button is to the right. Below these fields are two checkboxes: 'Drop car off at different location' (unchecked) and 'Driver aged between 30 - 65?' (checked). The background features a scenic view of a beach and ocean.

Result of the above search:

The screenshot shows the search results page for the selected parameters. At the top, it displays the pick-up and drop-off locations and dates. Below this, a map shows the location of the airport with a pin. A summary states '196 cars available'. There is a 'Sort by: Recommended' dropdown. A grid of car listings is shown, each with a thumbnail, model name, price, and deal details. The first listing is for a Ford Focus with a price of CAD 58 for 3 days. The second listing is for a Toyota Corolla with a price of CAD 58 for 3 days.

Car Model	Price (CAD)	Offer Details
Ford Focus	CAD 58	Top Pick, 10% off, Price for 3 days, Free cancellation, Pay at pick-up
Toyota Corolla	CAD 58	10% off, Ideal for Families, Price for 3 days, Free cancellation, Pay at pick-up

All available vehicle filtering options:

Filter [Clear all filters](#)

Location

- Airport (in terminal) 104
- Airport (shuttle) 92

Popular filters

- Pay at pick-up 154
- Unlimited mileage 156
- Free cancellation 196

Price per day

- CAD 0 - CAD 50 53
- CAD 50 - CAD 100 94
- CAD 100 - CAD 150 38
- CAD 150 - CAD 200 4
- CAD 200 + 7

Car specs

- Air Conditioning 194
- 4+ doors 163

Electric cars

- Fully electric 0
- Hybrid 0
- Plug-in hybrid 0

Mileage/Kilometres

- Limited 40
- Unlimited 156

Transmission

- Automatic 195
- Manual 1

Review score

- Superb: 9+ 59
- Very good: 8+ 89
- Good: 7+ 113

Car category

- Small 13
- Medium 59
- Large 137
- Premium 40
- People carriers 18
- SUVs 83

Fuel policy

- Like for like 42

Fuel policy

- Like for like 42

Deposit required at pick-up

- CAD 0 - CAD 375 5
- CAD 375 - CAD 750 3
- CAD 750 + 34

Supplier

- Ace 4
- ACE Rent A Car 11
- Alamo 23
- Avis 15
- Budget 15
- Dollar 7
- Economy 21
- Enterprise 18
- Fox 11
- Green Motion 15
- Hertz 3
- National 18
- Payless 6
- Routes 3
- Sixt 21
- Thrifty 5

Here we have selected the vehicle we would like to rent:

Toronto Pearson International Airport > Toronto Pearson International Airport
Wed 2 Apr 2025, 10:00 a.m. Sat 5 Apr 2025, 10:00 a.m.

[Back to Search results](#)

Your deal
[Next... Protection options](#)

Free cancellation - pay nothing until pick-up

Top Pick **10% off**

Ford Focus or similar large car



- 5 seats
- Automatic
- 2 Large bags ①
- 1 Small bag ①
- Unlimited mileage

Toronto Pearson International Airport
Shuttle Bus

 Pay at pick-up

ACE **Passable**
86 reviews

Great choice!

✓ Customer rating: 5.4 / 10 ✓ Free Cancellation



Real RC - ACE Rent A Car customers say...

...in genuine reviews of RC - ACE Rent A Car at Toronto Pearson International Airport.

Pick-up and drop-off

- Wed 2 Apr - 10:00
Toronto Pearson International Airport
[View pick-up instructions](#)
- Sat 5 Apr - 10:00
Toronto Pearson International Airport
[View drop-off instructions](#)

Car price breakdown

Car hire charge	CAD 50.88
10% off car rental price*	-CAD 6.45
Taxes and charges	CAD 13.62
Price for 3 days:	CAD 58.05

This car is costing you just CAD 58.05 – a real bargain...

At that time of year, the average large car at Toronto Pearson International Airport costs CAD 270.95!

Available options to add to the vehicle reservation before checkout:

Add extras, complete your trip

Additional driver

CAD 675 each per rental

If you also want other people to drive

— 0 +

Car seat

CAD 102.90 each per rental

For small children: 9–18 kg/20–40 lbs (about 1–3 years old)

— 0 +

We'll request your extras with RC - Dollar and you'll pay for them at pick-up. Pricing and availability can't be guaranteed until you arrive.

Show more extras ▾

Protection... for peace of mind

ⓘ **FREE cancellation** Full refund (CAD 60) if you cancel your plan anytime before pick-up

Collision Damage Protection from RentalCover.com: for CAD 60, you'll get CAD 50,000 of primary protection with no deductible (the protection price you see includes all applicable taxes and fees).

This protection includes insurance and non-insurance assistance services. Terms & conditions and standard exclusions apply. Please read:

Protection Terms

What is covered	No additional protection	Collision Damage Protection
✓ CAD 50,000 primary protection	✗	✓
✓ Covers damage and theft	✗	✓
✓ Covers all authorized drivers	✗	✓
	No protection price CAD 0	Total protection price CAD 60

ⓘ **Please note:** Your own car insurance is unlikely to cover hire cars.

Go to book
Without Collision Damage Protection

Go to book
With Collision Damage Protection

Pick-up and drop-off

○ Wed 2 Apr · 10:00
Toronto Pearson International Airport

[View pick-up instructions](#)

○ Sat 5 Apr · 10:00
Toronto Pearson International Airport

[View drop-off instructions](#)

Fast and reliable

Over 97% of claims paid out

Car price breakdown

CAR HIRE CHARGE	CAD 50.88
10% OFF CAR RENTAL PRICE*	-CAD 6.45
TAXES AND CHARGES	CAD 13.62

PRICE FOR 3 DAYS: CAD 58.05

This car is costing you just CAD 58.05 – a real bargain...

At that time of year, the average large car at Toronto Pearson International Airport costs CAD 270.95!

Necessary information the customer must fill out during the booking process:

Main driver's details

As they appear on driving licence

Email address *

So we can send the confirmation email and voucher

First name *

Last name *

Contact number *

So we can call if any problems come up

Country of residence *

▼

Flight number (optional)

Just in case the flight is delayed

Our [Privacy Statement](#) explains how we use and protect your personal information.

After booking the vehicle, we can see the confirmation and booking details:

Rentalcars.com by Booking.com

Lama, your car hire for Toronto Pearson International Airport is confirmed!

You're all set! Here is your booking confirmation and rental voucher.
You'll need to show this email to the counter staff at Toronto Pearson International Airport. You can either use a digital version (on your phone), or print it out.
You can check everything else you need to take with you in the Dollar terms.

[Manage Booking](#)

Booking number	5081302722168114
 Kia Rio or similar	
	5 seats Automatic
Pick-up instructions	Wednesday, April 02 2025 - 10:00 AM 6301 Silver Dart Drive, Toronto, L5P 1B2 Contact number: +19056769127
Drop-off instructions	Thursday, May 22 2025 - 10:00 AM 6301 Silver Dart Drive, Toronto, L5P 1B2 Contact number: +19056769127
Driver details	Lama Abdelfattah
Supplied by	dollar. Dollar Confirmation number: L1764885368

Price Breakdown

To pay at pick-up	
Car hire charge	CA\$1,350.00
PREM LOC SRG FEE	CA\$210.00
AIRCONDITON SRG REC	CA\$50.00
TAX	CA\$211.12
CUST TRANSACT CHG	CA\$14.00
Price for 20 days:	CA\$1,835.12

Terms and conditions
For details of what's included in your rental, the driver and licence requirements, information on your deposit, accepted payment methods, your fuel policy, and the car's excess, just check the T&Cs.

[Dollar rental terms](#)

Need some help?
 If you're running late to pick-up, have any problems with your car once you're on the road, or just need some assistance, please call Dollar on +19056769127.

Business Rules and Assumptions

Key Simplifying Assumptions for Database Modelling:

1. All rental transactions are assumed to be one-way, with vehicles being picked up and dropped off at different branches of the same rental company (e.g., Enterprise, Avis, Hertz).
2. Each vehicle is assigned to a single rental company and maintains a “current location” that updates after every rental.
3. Vehicle details (e.g., make, model, transmission) and extra options (e.g., waivers, car seats) are considered static for the purpose of the reservation period.
4. Vehicle categories (e.g., premium, SUVs, minivan) are predefined and a vehicle can belong to multiple categories.

Other Rules/Assumptions:

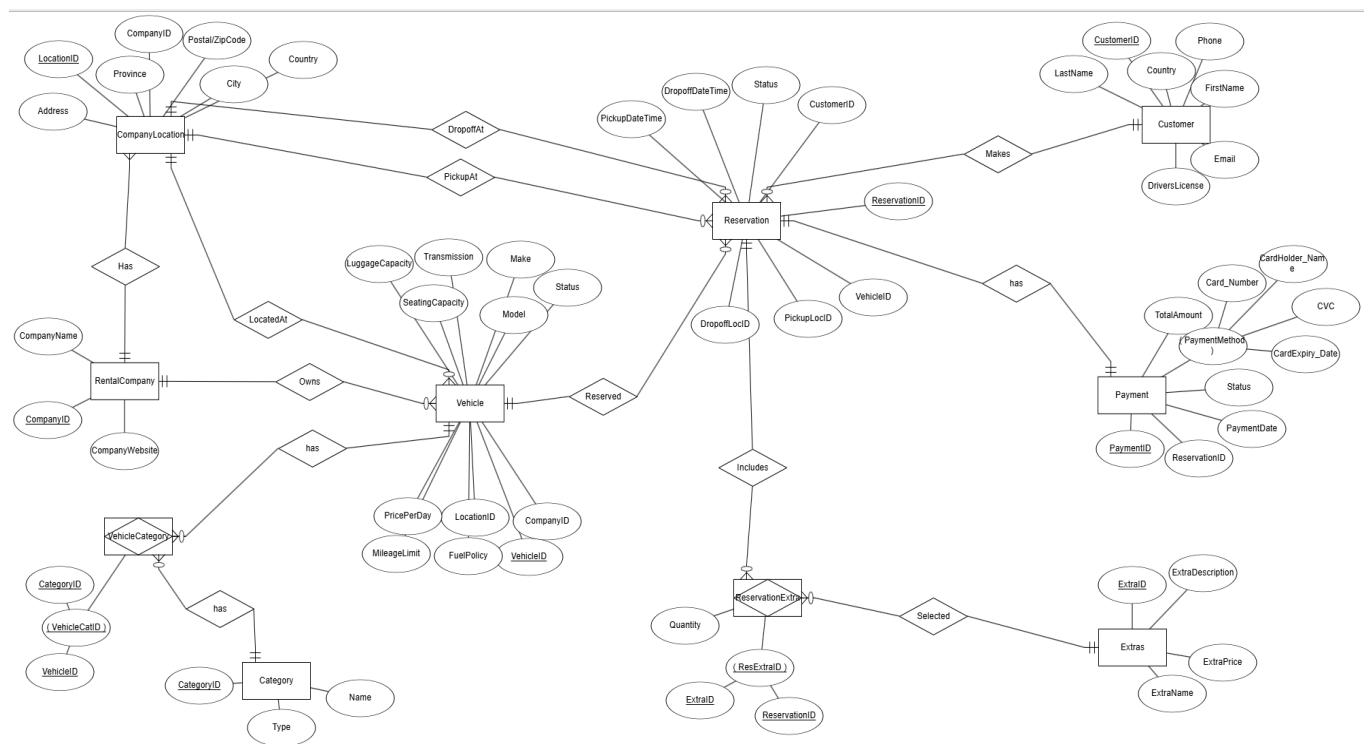
1. A reservation must include a valid customer, a selected vehicle, a pickup location, a dropoff location, and both pickup and dropoff date/time values.
2. A vehicle is available for rental only if its status is “available” and its current location matches the chosen pickup location.
3. One customer can create multiple reservations over time.
4. A single reservation can include multiple extra options, each with a specified quantity (e.g., multiple car seats).
5. Payment details are directly tied to a reservation and must be completed for the reservation to be confirmed.
6. The only Payment Method assumed here is Credit Card.
7. Rental pricing is determined by the vehicle’s price per day and may include additional charges for selected extra options.
8. Company locations are distinct and exclusively associated with a single rental company; thus, pickup and dropoff locations must belong to the same company.
9. All key attributes (such as CompanyID, VehicleID, CustomerID, etc.) are unique and non-null to ensure data integrity.
10. The system does not model vehicle maintenance or repair events as part of the rental process.
11. Transactions (reservations and payments) are recorded as separate audit trails for operational reporting and data integrity.
12. Date and time information for pickup and dropoff is critical for scheduling and is assumed to be provided in a consistent format.
13. Vehicle availability is updated in real-time based on active reservations.

Relationship between Entities:

1. A Rental Company owns one or more Company Locations, with each Company Location associated with exactly one Rental Company. (**One-to-Many**)
2. A Rental Company may have multiple vehicles or none at all, but each vehicle is assigned to only one Rental Company. (**One-to-Many**)
3. A vehicle can be classified under multiple categories, and each category can include multiple vehicles. (**Many-to-Many**)
4. A Company Location may have several vehicles or none, while each vehicle is stationed at exactly one Company Location. (**One-to-Many**)
5. A vehicle can have multiple reservations or none, whereas each reservation is linked to a single vehicle. (**One-to-Many**)
6. A Company Location may facilitate numerous reservations for both pickup and drop-off, but each reservation is tied to only one Company Location for pickup or drop-off. (**One-to-Many**)
7. A reservation can include multiple extras, and an extra may be associated with multiple reservations. (**Many-to-Many**)
8. A customer may make several reservations or none, while each reservation is linked to a single customer. (**One-to-Many**)
9. Each reservation is associated with exactly one payment, and each payment corresponds to only one reservation. (**One-to-One**)

DATABASE MODELLING

E-R Model



In the feedback on our Project Progress Report A, it was suggested that "Payment" should be modelled as a weak entity. However, the professor later confirmed via email that "Payment" is inherently a strong entity; therefore, we did not make any changes in this regard.

Additionally, we received feedback stating that the relationship between "Vehicle" and "Category" is not many-to-many, implying that a separate entity called "VehicleCategory" is unnecessary. However, upon revisiting the site, we observed that their relationship is indeed many-to-many, a finding we subsequently verified with the professor in class. To further

support this, we have attached screenshots demonstrating that a single category can include multiple cars, and the same car can belong to multiple categories.

The image consists of three separate screenshots from a car rental platform, each showing a list of available vehicles categorized by type. In each screenshot, a specific car model is highlighted with a blue oval, and the category it belongs to is circled in blue.

Screenshot 1 (Large Car Category):

- Large Car:** Vauxhall Grandland (highlighted), Renault Kadjar, Renault Kadjar, Vauxhall Insignia.
- Medium car:** Jeep Avenger.
- SUVs:** Vauxhall Grandland.
- Minivan:** None.

Screenshot 2 (SUVs Category):

- SUVs:** Jeep Avenger (highlighted).
- Large car:** Vauxhall Grandland.
- Medium car:** Renault Kadjar.
- Small car:** None.
- Minivan:** None.

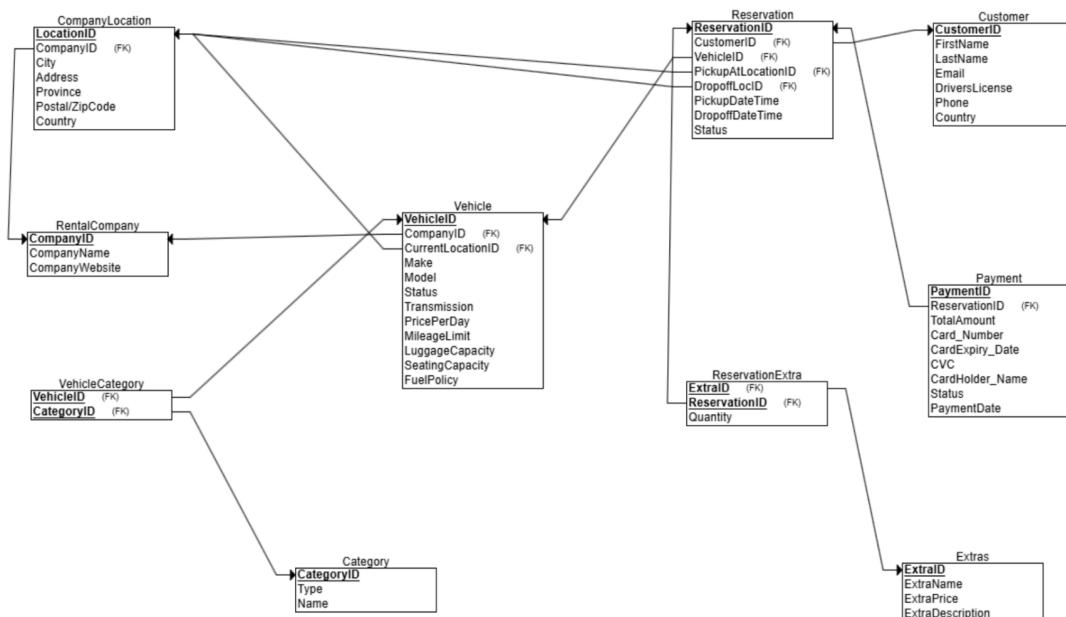
Screenshot 3 (Medium Car Category):

- Medium car:** Jeep Avenger (highlighted).
- Large car:** Vauxhall Grandland.
- SUVs:** Vauxhall Grandland.
- Minivan:** None.

As observed in all three images, a car, such as the "Vauxhall Grandland," belongs to multiple categories, including "Large Car," "SUVs," and "Medium Car". Similarly, the "Renault Kadjar" is classified under both the "Large Car" and "SUVs" categories. Additionally, it is evident that each category contains multiple cars. This confirms that the

relationship between "Vehicle" and "Category" is many-to-many.

Relational Schema



DATABASE DESIGN AND IMPLEMENTATION

Data Integrity Control

Table Name	Column Name	Key	Data Type	Allow Nulls	Foreign Key
RentalCompany	CompanyID	PK	INT	No	-
	CompanyName		VARCHAR(255)	No	-
	CompanyWebsite		VARCHAR(255)	No	-
CompanyLocation	LocationID	PK	INT	No	-
	CompanyID		INT	No	RentalCompany(CompanyID)
	City		VARCHAR(100)	No	-
	Address		VARCHAR(255)	No	-
	Province		VARCHAR(100)	No	-
	PostalZipCode		VARCHAR(50)	No	-

Vehicle	VehicleID	PK	INT	No	-
	CompanyID		INT	No	RentalCompany(CompanyID)
	CurrentLocationID		INT	No	CompanyLocation(LocationID)
	Make		VARCHAR(50)	No	-
	Model		VARCHAR(50)	No	-
	Status		VARCHAR(50)	No	-
	Transmission		VARCHAR(50)	No	-
	PricePerDay		DECIMAL(10,2)	No	-
	MileageLimit		INT	Yes	-
	LuggageCapacity		INT	No	-
	SeatingCapacity		INT	No	-
	FuelPolicy		VARCHAR(50)	No	-
Extras	ExtraID	PK	INT	No	-
	ExtraName		VARCHAR(255)	No	-
	ExtraPrice		DECIMAL(10,2)	No	-
	ExtraDescription		TEXT	Yes	-

Customer	CustomerID	PK	INT	No	-
	FirstName		VARCHAR(100)	No	-
	LastName		VARCHAR(100)	No	-
	Email		VARCHAR(255)	No	-
	DriversLicense		VARCHAR(50)	No	-
	Phone		VARCHAR(50)	No	-
	Country		VARCHAR(50)	No	-
Category	CategoryID	PK	INT	No	-
	Type		VARCHAR(50)	No	-
	Name		VARCHAR(100)	No	-
VehicleCategory	VehicleID	PK, FK	INT	No	Vehicle(VehicleID)
	CategoryID	PK, FK	INT	No	Category(CategoryID)
Reservation	ReservationID	PK	INT	No	-
	CustomerID		INT	No	Customer(CustomerID)
	VehicleID		INT	No	Vehicle(VehicleID)
	PickupLocID		INT	No	CompanyLocation(LocationID)

	DropoffLocID	INT	No	CompanyLocation(LocationID)
	PickupDateTime	DATETIME	No	-
	DropoffDateTime	DATETIME	No	-
	Status	VARCHAR(50)	No	-
Payment	PaymentID	PK	INT	No -
	ReservationID	INT	No	Reservation(ReservationID)
	TotalAmount	DECIMAL(10,2)	No	-
	PaymentMethod	VARCHAR(50)	No	-
	Status	VARCHAR(50)	No	-
	PaymentDate	DATETIME	No	-
	Card_Number	VARCHAR(19)	No	
	CVC	VARCHAR(4)	No	
	CardExpiry_Date	VARCHAR(5)	No	
	Card_Holder_Name	VARCHAR(50)	No	
ReservationExtra	ExtraID	PK, FK	INT	No Extras(ExtraID)
	ReservationID	PK, FK	INT	No Reservation(ReservationID)

Quantity	INT	No	-
----------	-----	----	---

Table Design

```

1  CREATE TABLE RentalCompany (
2      CompanyID INT NOT NULL,
3      CompanyName VARCHAR(255) NOT NULL,
4      CompanyWebsite VARCHAR(255) NOT NULL,
5      PRIMARY KEY (CompanyID)
6  );

1  CREATE TABLE CompanyLocation (
2      LocationID INT NOT NULL,
3      CompanyID INT NOT NULL,
4      City VARCHAR(100) NOT NULL,
5      Address VARCHAR(255) NOT NULL,
6      Province VARCHAR(100) NOT NULL,
7      PostalZipCode VARCHAR(50) NOT NULL,
8      Country VARCHAR(50) NOT NULL,
9      PRIMARY KEY (LocationID),
10     FOREIGN KEY (CompanyID) REFERENCES RentalCompany(CompanyID)
11 );
12

1  CREATE TABLE Vehicle (
2      VehicleID INT NOT NULL,
3      CompanyID INT NOT NULL,
4      CurrentLocationID INT NOT NULL,
5      Make VARCHAR(50) NOT NULL,
6      Model VARCHAR(50) NOT NULL,
7      Status VARCHAR(50) NOT NULL,
8      Transmission VARCHAR(50) NOT NULL,
9      PricePerDay DECIMAL(10,2) NOT NULL,
10     MileageLimit INT,
11     LuggageCapacity INT NOT NULL,
12     SeatingCapacity INT NOT NULL,
13     FuelPolicy VARCHAR(50) NOT NULL,
14     PRIMARY KEY (VehicleID),
15     FOREIGN KEY (CompanyID) REFERENCES RentalCompany(CompanyID),
16     FOREIGN KEY (CurrentLocationID) REFERENCES CompanyLocation(LocationID)
17 );

```

```

1 CREATE TABLE Extras (
2     ExtraID INT NOT NULL,
3     ExtraName VARCHAR(255) NOT NULL,
4     ExtraPrice DECIMAL(10,2) NOT NULL,
5     ExtraDescription TEXT,
6     PRIMARY KEY (ExtraID)
7 );|_
8
9
10 CREATE TABLE Customer (
11     CustomerID INT NOT NULL,
12     FirstName VARCHAR(100) NOT NULL,
13     LastName VARCHAR(100) NOT NULL,
14     Email VARCHAR(255) NOT NULL,
15     DriversLicense VARCHAR(50) NOT NULL,
16     Phone VARCHAR(50) NOT NULL,
17     Country VARCHAR(50) NOT NULL,
18     PRIMARY KEY (CustomerID)
19 );|_
20
21 CREATE TABLE Category (
22     CategoryID INT NOT NULL,
23     Type VARCHAR(50) NOT NULL,
24     Name VARCHAR(100) NOT NULL,
25     PRIMARY KEY (CategoryID)
26 );|_
27
28 CREATE TABLE VehicleCategory (
29     VehicleID INT NOT NULL,
30     CategoryID INT NOT NULL,
31     PRIMARY KEY (VehicleID, CategoryID),
32     FOREIGN KEY (VehicleID) REFERENCES Vehicle(VehicleID),
33     FOREIGN KEY (CategoryID) REFERENCES Category(CategoryID)
34 );|_
35
36 CREATE TABLE Reservation (
37     ReservationID INT NOT NULL,
38     CustomerID INT NOT NULL,
39     VehicleID INT NOT NULL,
40     PickupLocID INT NOT NULL,
41     DropoffLocID INT NOT NULL,
42     PickupDateTime DATETIME NOT NULL,
43     DropoffDateTime DATETIME NOT NULL,
44     Status VARCHAR(50) NOT NULL,
45     PRIMARY KEY (ReservationID),
46     FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),
47     FOREIGN KEY (VehicleID) REFERENCES Vehicle(VehicleID),
48     FOREIGN KEY (PickupLocID) REFERENCES CompanyLocation(LocationID),
49     FOREIGN KEY (DropoffLocID) REFERENCES CompanyLocation(LocationID)
50 );
51
52

```

```

1  CREATE TABLE Payment (
2    PaymentID INT NOT NULL,
3    ReservationID INT NOT NULL,
4    TotalAmount DECIMAL(10,2) NOT NULL,
5    PaymentMethod VARCHAR(50) NOT NULL,
6    Status VARCHAR(50) NOT NULL,
7    PaymentDate DATETIME NOT NULL,
8    Card_Number VARCHAR(19) NOT NULL,
9    CVC VARCHAR(4) NOT NULL,
10   CardExpiry_Date VARCHAR(5) NOT NULL,
11   Card_Holder_Name VARCHAR(50) NOT NULL,
12   PRIMARY KEY (PaymentID),
13   FOREIGN KEY (ReservationID) REFERENCES Reservation(ReservationID)
14 );
1
1  CREATE TABLE ReservationExtra (
2    ExtraID INT NOT NULL,
3    ReservationID INT NOT NULL,
4    Quantity INT NOT NULL,
5    PRIMARY KEY (ExtraID, ReservationID),
6    FOREIGN KEY (ExtraID) REFERENCES Extras(ExtraID),
7    FOREIGN KEY (ReservationID) REFERENCES Reservation(ReservationID)
8 );

```

*Please ignore the underlined tables, all queries ran successfully

Entity Integrity

Entity integrity ensures that each table has a unique identifier as primary key. Each table has a unique identifier such as:

- CompanyID for RentalCompany Table.
- LocationID for CompanyLocation Table..
- VehicleID for Vehicle entity Table.
- ExtraID for Extras Table.
- CustomerID for Customer Table.
- CategoryID for Category Table.
- A composite key(VehicleID, CategoryID) for VehicleCategory Table.
- ReservationID for Reservation Table.
- PaymentID for Payment Table.
- A composite key (ExtraID, ReservationID) for ReservationExtra Table.

Reference Integrity

Reference integrity ensures valid relationships between tables by connecting them with Foreign Keys.

- In CompanyLocation Table:

- CompanyID is a foreign key referencing RentalCompany.CompanyID, ensuring every location belongs to a valid company.
- In Vehicle Table:
 - CompanyID is a foreign key referencing RentalCompany.CompanyID, ensuring every vehicle belongs to a valid company.
 - CurrentLocationID is a foreign key referencing CompanyLocation.LocationID, ensuring vehicles are associated with existing locations.
- In VehicleCategory Table:
 - VehicleID is a foreign key referencing Vehicle.VehicleID, ensuring only valid vehicles are categorized.
 - CategoryID is a foreign key referencing Category.CategoryID, ensuring categories exist.
- In Reservation Table:
 - CustomerID is a foreign key referencing Customer.CustomerID, ensuring each reservation belongs to a valid customer.
 - VehicleID is a foreign key referencing Vehicle.VehicleID, ensuring only valid vehicles are reserved.
 - PickupLocID and DropoffLocID are foreign keys referencing CompanyLocation.LocationID, ensuring valid pickup/drop-off locations.
- In Payment Table:
 - ReservationID is a foreign key referencing Reservation.ReservationID, ensuring payments correspond to valid reservations
- In ReservationExtra Table:
 - ExtraID is a foreign key referencing Extras.ExtraID, ensuring extra services are valid.
 - ReservationID is a foreign key referencing Reservation.ReservationID, ensuring extras belong to valid reservations.

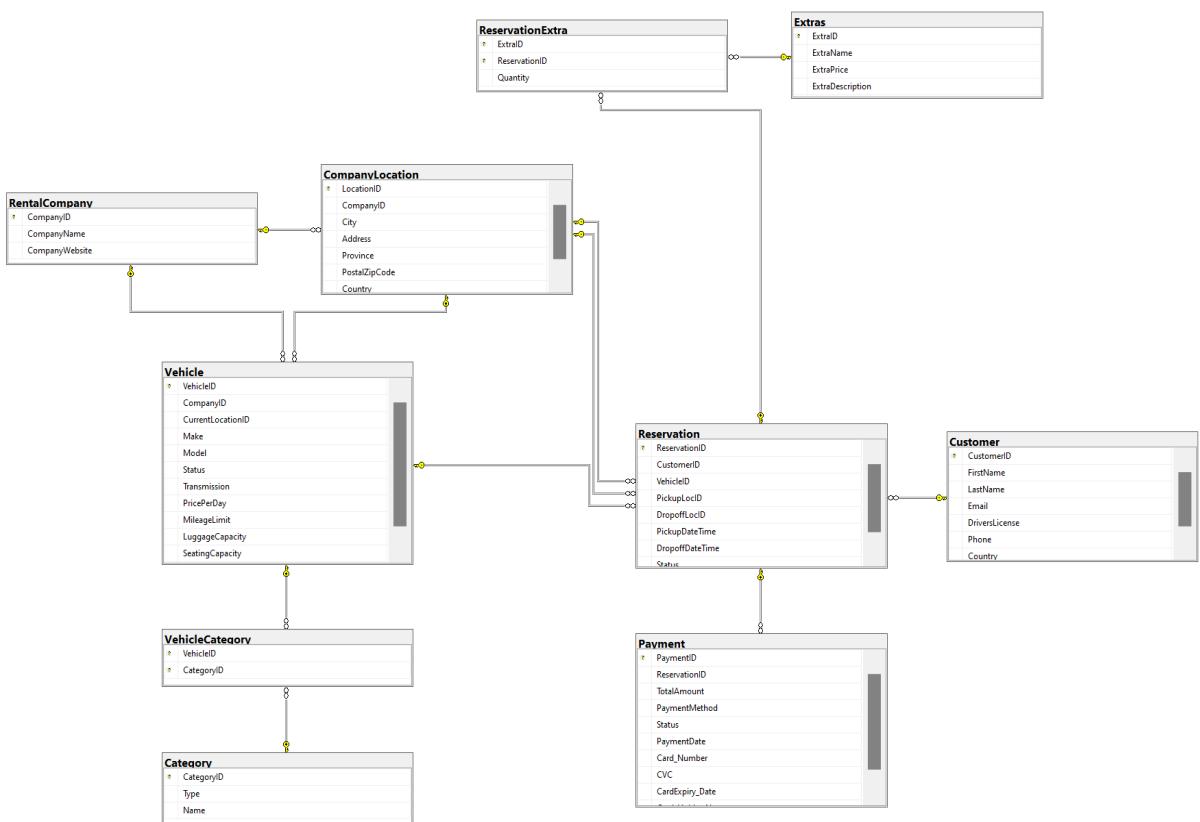
Domain Integrity

Domain integrity ensures that columns have appropriate data types and constraints to maintain data validity.

- CompanyName, Make, Model, Status, Transmission, FuelPolicy, PaymentMethod, Card_Holder_Name: Stored as VARCHAR, ensuring only text is entered.
- There is a NUMERIC constraint on PricePerDay DECIMAL(10,2) that ensures the maximum numbers should be 10 with 2 digits after the decimal.
- PickupDateTime, DropoffDateTime, PaymentDate: Stored as DATETIME, ensuring valid date formats.
- SeatingCapacity, MileageLimit, LuggageCapacity, Quantity: Stored as INT, ensuring numeric values. Where MileageLimit allows NULL to indicate unlimited mileage.

- CVC VARCHAR(4): Ensures valid CVC length is either 3 (for most credit cards) or 4(American express).
- Card_Number VARCHAR(19): Ensures card numbers fit standard lengths.
- CardExpiry_Date VARCHAR(5): Ensures expiry date follows MM/YY format.

Relational Table



*Not all the column names are visible due to space constraints.

Sample Data

RentalCompany

	CompanyID	CompanyName	CompanyWebsite
1	1	Avis	https://www.avis.com
2	2	Enterprise	https://www.enterprise.com
3	3	Hertz	https://www.hertz.com

CompanyLocation

	LocationID	CompanyID	City	Address	Province	PostalZipCode	Country
1	101	1	Toronto	123 Queen St West	Ontario	M5H 2N2	Canada
2	102	1	Ottawa	456 Rideau St	Ontario	K1N 5X3	Canada
3	201	2	Mississauga	789 Hurontario St	Ontario	L5B 2H2	Canada
4	202	2	Hamilton	1011 King St	Ontario	L8R 1S8	Canada
5	301	3	London	121 Oxford St	Ontario	N6A 1B4	Canada
6	302	3	Kitchener	202 King St E	Ontario	N2G 4H8	Canada

Vehicle

	VehicleID	CompanyID	CurrentLocationID	Make	Model	Status
1	1001	1	101	Toyota	Camry	Available
2	1002	1	102	Ford	Focus	Rented
3	1003	1	101	Nissan	Altima	Available
4	1004	1	101	Chrysler	Pacifica	Available
5	2001	2	201	Chevrolet	Malibu	Available
6	2002	2	202	Honda	Civic	Unavailable
7	2003	2	201	Toyota	Corolla	Available
8	2004	2	202	Toyota	Prius Prime	Available
9	3001	3	301	BMW	3 Series	Available
10	3002	3	302	Mercedes	C-Class	In Service
11	3003	3	301	Audi	A4	Rented

Transmission	PricePerDay	MileageLimit	LuggageCapacity	SeatingCapacity	FuelPolicy
Automatic	45.00	NULL	3	5	Full-to-Full
Manual	40.00	150	2	5	Prepaid
Automatic	47.00	NULL	3	5	Full-to-Full
Automatic	55.00	NULL	4	7	Full-to-Full
Automatic	50.00	NULL	3	5	Full-to-Full
Manual	42.00	180	2	5	Full-to-Full
Automatic	38.00	NULL	2	5	Full-to-Full
Automatic	60.00	NULL	2	5	Full-to-Full
Automatic	80.00	NULL	3	5	Full-to-Full
Automatic	85.00	300	3	5	Prepaid
Automatic	90.00	NULL	3	5	Prepaid

Extras

	ExtraID	ExtraName	ExtraPrice	ExtraDescription
1	1	GPS Navigation	10.00	In-car GPS navigation system
2	2	Child Seat	15.00	Child safety seat suitable for ages 1-5
3	3	Additional Driver	20.00	Allows an extra driver for the rental
4	4	Insurance Waiver	25.00	Reduces or eliminates insurance coverage costs

Customer

	CustomerID	FirstName	LastName	Email	DriversLicense	Phone	Country
1	1	John	Doe	john.doe@example.com	D1234567	555-1234	Canada
2	2	Jane	Smith	jane.smith@example.com	S7654321	555-5678	Canada
3	3	Carlos	Gonzalez	carlos.gonzalez@example.com	G9876543	555-9012	Canada

Category

	CategoryID	Type	Name
1	1	Size	Small
2	2	Size	Medium
3	3	Size	Large
4	4	Electric	Hybrid
5	5	Tier	Premium
6	6	Electric	Fully-Electric
7	7	Type	SUV
8	8	Electric	Plug-in Hybrid
9	9	Type	Minivan

VehicleCategory

	VehicleID	CategoryID
1	1001	2
2	1002	1
3	1003	2
4	1003	4
5	1004	9
6	2001	2
7	2002	1
8	2003	1
9	2004	8
10	3001	3
11	3001	5
12	3002	3
13	3002	5
14	3003	3
15	3003	5

Reservation

	ReservationID	CustomerID	VehicleID	PickupLocID
1	1	1	1002	102
2	2	2	2001	201
3	3	3	3001	301
4	4	1	3002	302

DropoffLocID	PickupDateTime	DropoffDateTime	Status
101	2025-02-15 10:00:00.000	2025-02-20 10:00:00.000	Completed
202	2025-03-01 09:00:00.000	2025-03-05 09:00:00.000	Active
302	2025-04-10 08:00:00.000	2025-04-15 08:00:00.000	Pending
301	2025-05-01 12:00:00.000	2025-05-07 12:00:00.000	Cancelled

Payement

	PaymentID	ReservationID	TotalAmount	PaymentMethod	Status
1	1	1	200.00	Credit Card	Completed
2	2	2	200.00	Credit Card	Pending
3	3	3	400.00	Credit Card	Completed
4	4	4	510.00	Credit Card	Refunded

PaymentDate	Card_Number	CVC	CardExpiry_Date	Card_Holder_Name
2025-02-15 10:05:00.000	4111111111111111	123	12/28	John Doe
2025-03-01 09:10:00.000	4222222222222222	456	11/27	Jane Smith
2025-04-10 08:05:00.000	4333333333333333	789	10/26	Carlos Gonzalez
2025-05-01 12:05:00.000	4444444444444444	321	09/25	John Doe

ReservationExtra

	ExtraID	ReservationID	Quantity
1	1	1	1
2	1	4	1
3	2	2	1
4	3	2	1
5	4	3	1

Normalization Check:

1NF (First Normal Form):

Each column contains atomic (indivisible) values. Each row is uniquely identified by a primary key. There are no multiple values in a single column or nested tables within the tables.

Therefore, All tables are in 1NF.

2NF (Second Normal Form):

To be in 2NF, a table must be in 1NF and Have no partial dependency (i.e., no non-key column depends on just part of a composite key).

There are two such tables:

VehicleCategory (VehicleID, CategoryID)

Only includes foreign keys and no partial dependencies exist.

ReservationExtra (ExtraID, ReservationID)

Quantity depends on the whole composite key.

Therefore, All tables are in 2NF

3NF (Third Normal Form):

To be in 3NF, a table must be in 2NF and have no transitive dependencies (non-key columns must not depend on other non key columns)

None of the tables have transitive dependencies except the Payment Table.

Card Holder Name depends on a non key column Payment Method.

Sample SQL Queries

1. List Available Vehicles at a Specific Location and Date Range

This query finds vehicles that are available in a given city (in this example, Toronto) and that are not booked during a specific period.

```
1  SELECT v.VehicleID, v.Make, v.Model, v.Transmission, v.PricePerDay, cl.City
2  FROM Vehicle v
3  JOIN CompanyLocation cl ON v.CurrentLocationID = cl.LocationID
4  WHERE cl.City = 'Toronto'
5  AND v.Status = 'Available'
6  AND v.VehicleID NOT IN (
7      SELECT r.VehicleID
8      FROM Reservation r
9      WHERE (r.PickupDateTime < '2025-06-10 00:00:00' AND r.DropoffDateTime > '2025-06-01 00:00:00')
10 );
11
```

Results Messages

	VehicleID	Make	Model	Transmission	PricePerDay	City
1	1001	Toyota	Camry	Automatic	45.00	Toronto
2	1003	Nissan	Altima	Automatic	47.00	Toronto
3	1004	Chrysler	Pacifica	Automatic	55.00	Toronto

2. Retrieve All Reservations for a Specific Customer with Details

This query shows the reservations for a given customer (change the CustomerID as needed) along with vehicle and location details.

```

1  SELECT r.ReservationID,
2      r.PickupDateTime,
3      r.DropoffDateTime,
4      r.Status,
5      v.Make,
6      v.Model,
7      pl.City AS PickupCity,
8      dl.City AS DropoffCity
9  FROM Reservation r
10 JOIN Vehicle v ON r.VehicleID = v.VehicleID
11 JOIN CompanyLocation pl ON r.PickupLocID = pl.LocationID
12 JOIN CompanyLocation dl ON r.DropoffLocID = dl.LocationID
13 WHERE r.CustomerID = 1;
14

```

Results Messages

	ReservationID	PickupDateTime	DropoffDateTime	Status	Make	Model	PickupCity	DropoffCity
1	1	2025-02-15 10:00:00.000	2025-02-20 10:00:00.000	Completed	Ford	Focus	Ottawa	Toronto
2	4	2025-05-01 12:00:00.000	2025-05-07 12:00:00.000	Cancelled	Mercedes	C-Class	Kitchener	London

3. Search for Vehicles by Category (e.g., Premium)

This query retrieves all vehicles that belong to the Premium category.

```

1  SELECT v.VehicleID, v.Make, v.Model, v.Status, v.PricePerDay
2  FROM Vehicle v
3  JOIN VehicleCategory vc ON v.VehicleID = vc.VehicleID
4  JOIN Category c ON vc.CategoryID = c.CategoryID
5  WHERE c.Name = 'Premium';
6

```

Results Messages

	VehicleID	Make	Model	Status	PricePerDay
1	3001	BMW	3 Series	Available	80.00
2	3002	Mercedes	C-Class	In Service	85.00
3	3003	Audi	A4	Rented	90.00

4. Reservations that Include a Specific Extra (e.g., 'Child Seat')

This query retrieves reservations where the selected extra option is a Child Seat. It joins with customer details for context.

```

1  SELECT
2      r.ReservationID,
3      c.FirstName,
4      c.LastName,
5      e.ExtraName,
6      r.PickupDateTime,
7      r.DropoffDateTime
8  FROM Reservation r
9  JOIN ReservationExtra re ON r.ReservationID = re.ReservationID
10 JOIN Extras e ON re.ExtraID = e.ExtraID
11 JOIN Customer c ON r.CustomerID = c.CustomerID
12 WHERE e.ExtraName = 'Child Seat';
13

```

Results Messages

	ReservationID	FirstName	LastName	ExtraName	PickupDateTime	DropoffDateTime
1	2	Jane	Smith	Child Seat	2025-03-01 09:00:00.000	2025-03-05 09:00:00.000

EVALUATION AND CONCLUSION

Overview of Design and Implementation

To build a scalable and efficient car rental database system, we adopted a methodical development cycle involving requirements analysis, ER modeling, schema normalization, and SQL Server based implementation. Emphasis was placed on maintaining data integrity through primary and foreign keys and enforcing consistency via domain and entity constraints. Performance was enhanced through indexing on frequently queried attributes. The final schema supports all core operations such as real-time vehicle availability tracking, secure booking, and dynamic pricing, providing a foundation for a reliable and extensible rental platform.

Achievement of the project

The Rentalcars.com database project successfully delivered a fully functional system that streamlines the car rental process for global users. The database enables real-time vehicle searches based on location, date, and vehicle type, presenting customers with transparent pricing and availability across 60,000+ rental partners. Key achievements include a secure booking module with integrated payment processing, automated confirmation emails/SMS, and a user-friendly interface available in multiple languages. On the administrative side, the system tracks fleet inventory, prevents overbooking, and provides analytics on rental trends. By centralizing customer profiles, payment methods, and rental histories, the database significantly enhances operational efficiency while maintaining Rentalcars.com's commitment to hassle free service.

Further development and improvement

To build on this project, several enhancements could further elevate the platform. Future iterations might integrate partnerships with ride-sharing services (e.g., Uber or Lyft) to offer end-to-end travel solutions. Dynamic pricing algorithms could adjust rates based on demand fluctuations, local events, or fuel prices. Additional vehicle details such as real-time GPS tracking, emissions data, or user-submitted cleanliness ratings would help customers make informed choices. Expanding the loyalty program to include tiered rewards (e.g., free upgrades for frequent renters) and personalized discounts could boost retention. Finally, adding AI-driven features like predictive maintenance alerts for rental fleets or chatbot-assisted bookings would align with industry innovations.

Advantages of the Database Package (MICROSOFT SQL SERVER)

Microsoft SQL Server provided an ideal foundation for Rentalcars.com's database system due to its strong scalability, effortlessly accommodating growing global rental transactions and user traffic. The platform's relational constraints ensure data integrity across linked tables (e.g., customer bookings to vehicle inventory), preventing inconsistencies. As an industry-standard tool, SQL Server offered seamless integration with Rentalcars.com's front-end booking interface, while its intuitive design allowed student developers to contribute effectively. The cost-effective licensing (for educational use) and compatibility with cloud deployments aligned well with project budgets. However, limitations emerged during stress testing: concurrency delays occurred during peak booking simulations, and the base version required supplemental tools for end-to-end encryption of sensitive payment data. Performance also varied based on server hardware, and initial design iterations revealed redundancy risks in vehicle availability tables until normalization was refined.

Limitations of the Database Package

The database package we used, Microsoft SQL Server, comes with several limitations worth noting. It may encounter concurrency issues, leading to delays when multiple transactions are processed at the same time. The basic version also lacks some advanced, enterprise-level features, which could restrict functionality in larger-scale applications. From a security standpoint, the platform requires additional configuration or third-party tools to implement strong encryption and effective access control, especially for beginners. Its performance is highly dependent on the hardware it runs on, meaning lower-end systems may experience slowdowns. Lastly, the system does not provide warnings for poorly normalized data, which can lead to redundancy and inconsistent tables if the schema isn't carefully designed from the outset.

Experience Learned

Developing Rentalcars.com's database was a transformative learning opportunity. The team gained hands-on expertise in relational database design, constructing ER diagrams that mapped customer accounts, rental transactions, and fleet logistics. SQL query optimization became critical for instance, indexing frequently searched fields like pickup locations and vehicle types accelerated response times. By implementing primary/foreign keys, we enforced referential integrity (e.g., ensuring a booked vehicle couldn't be deleted from the inventory table). Practical challenges like concurrency management taught us to balance real-time bookings with system performance. Collaboratively, we divided tasks into one subgroup focused on table normalization to eliminate redundancy, while others designed sample queries for pricing algorithms. The project honed time management (meeting iterative deadlines) and debugging skills when testing edge cases (e.g., overlapping reservations). Most importantly, we learned to design for scalability structuring tables to support future features like dynamic pricing or AI-driven recommendations. These transferable skills will prove invaluable in professional database roles.

Contribution and Time Spent

- Introduction of the Project and company.
- Database Model - Business rules and assumptions.
- Developing E-R model- Company Location and Rental Company entity.
- Database Modeling Relational Schema.
- Table Design of Company Location and Rental Company.
- Implementation of Relational Table.
- Running Sample Query - List Available Vehicles at a Specific Location and Date Range.
- Evaluation and Conclusion - Further development and improvement

- Objective of the company.
- Data Modeling - Relationship between Entities.
- Developing E-R model- Vehicle category and Category entity.
- Database Design and Implementation of Data Integrity control.
- Table Design of Vehicle category and Category.
- Implementation of Relational Table.
- Running Sample Query - Retrieve All Reservations for a Specific Customer with Details

- Advantages of the Database Package (MICROSOFT SQL SERVER) and Limitations.
- Company scope.
- Database Model - External Views.
- Developing E-R model- Reservation and Reservation Extra entity.
- Database Modeling Relational Schema.
- Table Design of Category Reservation and Reservation Extra
- Inserting Sample Data in the table.
- Running Sample Query - Reservations that Include a Specific Extra.
- Evaluation and Conclusion - Achievement of the project.
- Functional requirements of the application.
- Data Modeling- Relationship between Entities.
- Developing E-R model- Vehicle and Extra Entity.
- Database Design and Implementation of Data Integrity control.
- Table Design of Vehicle and Extra.
- Inserting Sample Data in the table.
- Running Sample Query - Search for Vehicles by Category.
- Evaluation and Conclusion - Overview of Design and Implementation
- Functional requirements of the application.
- Database Model - External Views and Business rules and assumptions.
- Developing E-R model- Customer and Payment entity.
- Database Design and Implementation of Data Integrity control.
- Table design of Customer and Payment.
- Inserting Sample Data in the table.
- Normalisation Check for all 1,2 and 3 Normal Form.
- Evaluation and Conclusion - Experience Learned

THANK YOU!