

EZRental POS Design and Implementation Document

Pavel Drozdov

10/18/2022

Table of Content

1. Executive Summary -- p3
2. Problem Statement & Objectives -- p4
3. Project Management Methodology -- p5
4. Application Business requirements -- p7
5. Application Development & Technical Requirements -- p18
6. Application Physical Technical Architecture -- p28
7. Application Development Features and Functionalities (Agile Backlog) -- p33
8. Database Management System Development Environment & Physical Architecture -- p47
9. Project Roles & Responsibilities -- p48
10. Database Design Deliverable #1 – ER/EER Conceptual Model Diagram -- p50
11. Database Design Deliverable #2 – Normalized Logical Model Diagram -- p51
12. Database Implementation Deliverable #3 Physical Model Data Dictionary -- p53
13. Database Implementation Deliverable #4 Physical Model Scheme Design Diagram -- p59
14. Database Implementation Deliverable #5 Development & Implementation – p60
15. Database Implementation Deliverable #6 Implemented Physical Scheme Diagram – p65
16. Database Implementation Deliverable #7 Database Validation Testing – p66
17. Conclusion – p88

EZRenatal Auto Rental POS Management System Database Design and Implementation

Executive Summary

Problem Statement & Objectives

EZRental Inc., has hired **NYC-Tech Solutions Inc.**, with the goal to design & implement a suite of **Auto Rental Point-of-Sales Management System Application** that include the following business modules: **1) EZRenatal Point-of-Sales (POS) system** intended for **Customer Service Representative** and other **employees** in the **rental agencies**, such as Maintenance Personnel, Vehicle Inventory Team, Transport Drivers etc. **2) A Corporate INTRANET Website** named **EZRentalCorp.com** intended for business employees in the corporate offices, and Rental Agencies, and finally, **3) an e-commerce INTERNET Website** named **EZRental.com** intended for customers to make and manage reservations via the public internet.

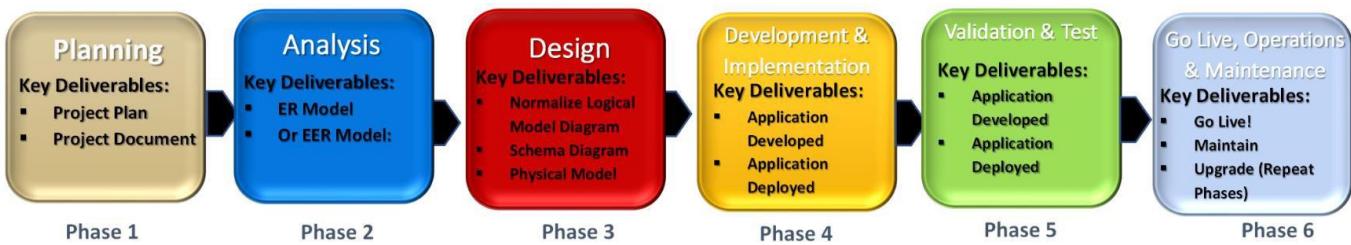
The application is designed to support to support dozens of major cities around the world in counties such as US, Canada, Mexico, UK, Japan, and Australia. In addition, it provides a great user experience both in the rental agencies as well as the online systems with the best competitive pricing available in the market.

Project Management Methodology

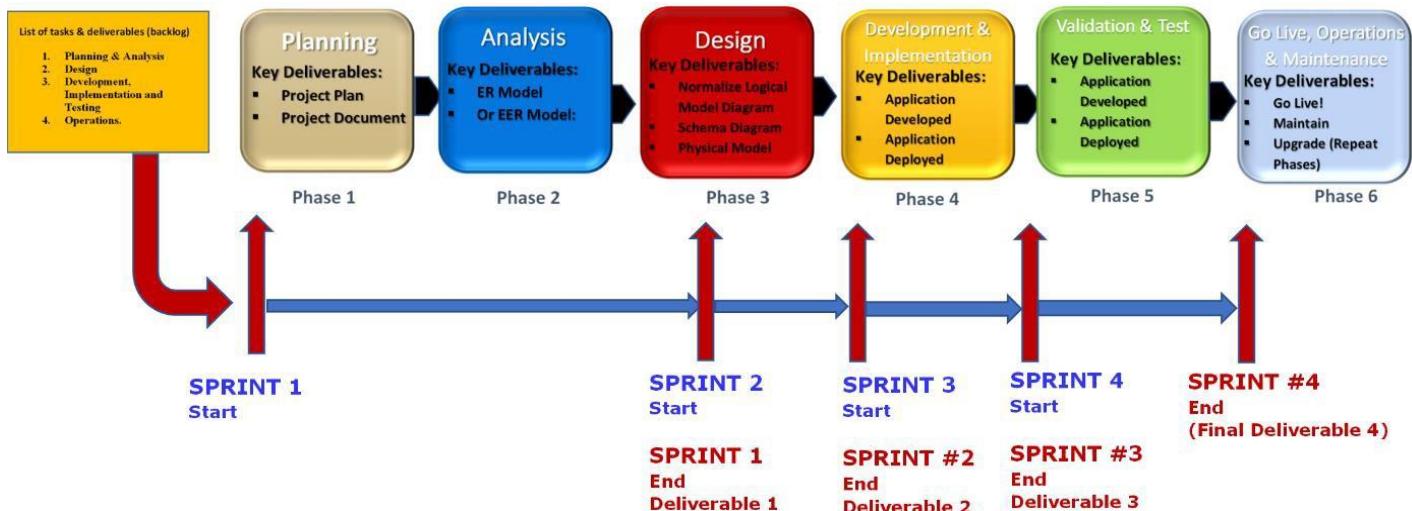
To design, develop, and implement the **DBMS Server Application** for the Auto Rental Management System ER-Rental POS, Two-Tired Client/Server & Three-Tired Web Client/Server Application was used **Waterfall Project Management Methodology** combined with **Agile Project Management Methodology**.

The Waterfall Project Management with 6-phases was decided by the project architectures and project manager.

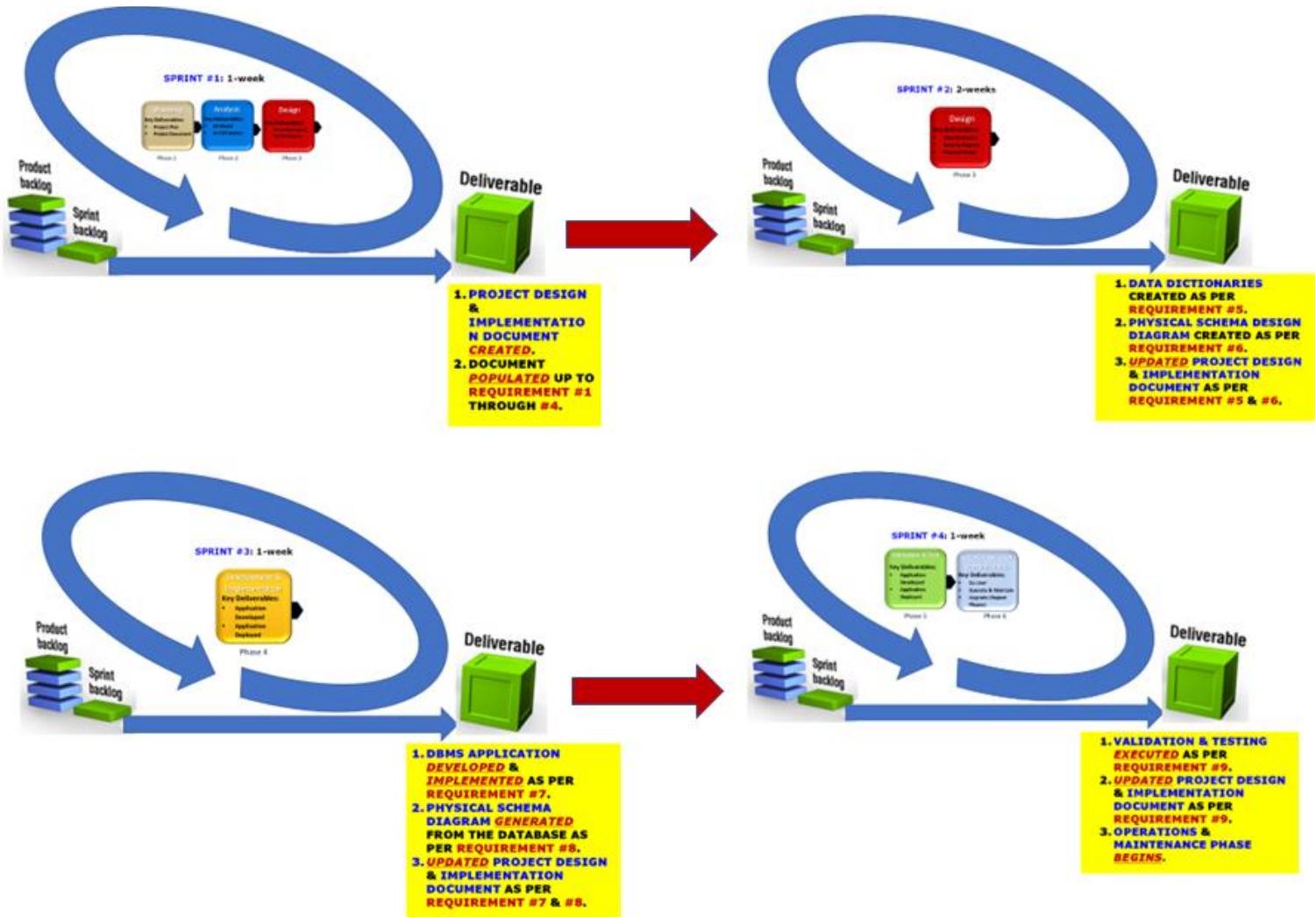
The illustration below shows the **6 phases** and the main deliverables for each phase.



Then, the **Waterfall Approach** was divided into **4 Sprints** for a period of **5 weeks** using **Agile Methodology**. The **diagram** and **sprints** description table is shown below:



AGILE SPRINT #	WATERFALL PHASE	Output Deliverable
SPRINT #1	Planning	1. Create Project Document – Formatted and populated as per requirements. 2. Business Requirements (Included in Project Document) – List of Business & Technical Requirements from customer.
	Analysis	3. ER/EER Conceptual Model Diagram
	Design Phase (Part 1)	4. Normalized Logical Model Diagram
SPRINT #2	Design Phase (Part 2)	5. Data Dictionary matrix 6. Physical Schema Design Diagram – from Normalize Logical Model + DataDictionary combination.
SPRINT #3	Development & Implementation	7. Database application developed & implemented – This includes the Database application installed, setup and configured 8. Generate the actual Physical Schema Diagram – from the Database & compared to the Physical Schema Design Diagram – to validate the design.
SPRINT #4	Validation & Testing	9. Unit & Integration testing.
	Operations	10. Operations – or keep database running. Keeping the lights on!



Application Business requirements

A Database Business Analyst interviewed EZRental Inc., Project & Business Stakeholders to gather & compile a list of the business requirements needed for the application.

The following pages contain the **Business Requirements** captured by the **Business Analyst**:

Business Requirements

About Us:

EZ-Car Rental is an auto rental company that rents vehicles such as cars, SUVs, minivans & cargo vans to customers. In addition, other specialized vehicles such as trucks, motorcycles, boats, mobile homes, etc. We operate in several countries with rental agency locations in the US, Canada, Mexico, UK, Japan & Australia. Within each country we own and operate rental agencies located in cities, regions and state. For example, New York City has 2 rental agencies in Manhattan, one in Brooklyn and two in Queens located at each airport. With multiple rental agencies in cities, states etc., a customer can pick up a vehicle in one location and drop it off at another.

Current Challenges:

Our current rental system is outdated, with a poor user-experience, inefficient (breaks often thus expensive to operate), does not meet our business requirements, and is not scalable (cannot be easily updated with new features). Another very important shortcoming of the current system, is the lack of elasticity since it does not give us the flexibility to scale-up or scale-down resources during business trends and seasonal changes in the market.

We want to invest in modernizing our business with a new vehicle management system that can meet these challenges and delivers a great user-experience, meet our new business requirements, scalable, and elastic to adopt to business trends and seasonal market changes. Elasticity is very important since recently we have been faced with a new type of competition; small rental companies that are nimble and can quickly adopt to market changes thus able to provide new offerings that are appealing to customers thus affecting our profits. These smaller competitors are using new technologies that enable them to be nimble and elastic. Figurative speaking “*they are eating our lunch*”.

We look forward to your proposed architecture & implementation of this new system. Below are our business requirements.

Our Agencies:

A **rental agency** is identified by a unique **rental agency ID** number, **agency name**, **address** that is composed of the following elements: **address line1**, **address line 2** (which is optional and used for apartment number, suite or any additional address information required), **city**, **state code** (which is the two-character code for a state in the US), **zip code** & **country**. In addition, we also need to capture the agency's **phone number**, and **email** which is unique for all agencies as all emails are.

Our Customers:

EZ-Car Rental offer their services to two types of **Customers: Corporate Customers & Retail Customers**. **Corporate Customers** are individuals whose corporation have a contract with us to use our services with special corporate rate for their employee's rental services. On the other hand, **Retail Customers** are consumers not associated with a company and engaging in personal rental.

All Customers (Retail & Corporate Customers)

To run our business, the application must store the following customer information for both types of **customer** (retail & corporate) so this data is common to both types of customers:

- A **Customer ID** number which uniquely identifies the customer, **customer name** which is composed of: **first name**, **last name**.
- **Birth date**, **Age**, **Address** which includes the elements: **address line1**, **address line 2** (which is optional and used for apartment number, suite or any additional address information required), **city**, **state code** (which is the two-character code for a state in the US), **zip code** & **country**.
- Customer **phone number** & **email** (unique like all emails and required to rent).
- In addition, a driver license is required to reserve and rent a vehicle. Therefore, we need to capture the unique **driver license number** (**an alpha numeric character string containing numbers & characters**), **driver license expiration Date** and **driver license state**. In addition, note the following business rule on the business importance of the **driver license number**:

1. **The driver license number is used throughout the business to identify a customer for searching, reporting etc.**
2. **Therefore, the driver license number is the unique ID for a customer to be identified and managed from a business perspective.**

Business Requirements

Our Customers (Cont.):

- A very important attribute we need to capture for every customer is the **credit card**. A credit card includes the following attributes: **credit card number** that uniquely identifies the credit card and is a 16-character number digits, **credit card owner name**, **credit card issuing company name** (such as American Express, Visa, MasterCard, Capital One, etc.), **merchant Code & merchant name** which is the credit card payment processing company that acts as an intermediary between our business and the customers' credit card companies or bank. The merchant handles the interaction between the purchase of a rental and the credit card company etc., validating credit card transaction. This merchant Code & Name attributes have business meaning and used throughout the business using a digit code for **merchant Code** and the name of the merchant associated with the code or **merchant name**. We currently use the following **merchant code** and **merchant names** throughout the world to handle our credit card processing:

Merchant Code	Merchant Name
1	Stax by Fattmerchant
2	Helcim
3	Dharma Merchant Services
4	Payment Depot
5	National Processing
6	Block
7	Intuit Quickbooks
8	PayPal
9	Stripe
10	Flagship Merchant Services
11	Clover

- Other attributes of credit card are **expiration date**, **billing address** composed of **address line1**, **address line 2** (which is optional and used for apartment number, suite or any additional address information required), **city**, **state code** (which is the two-character code for a state in the US), **zip code & country**.
- In addition, **credit card limit**, **credit card balance & activation status** which is true if the credit card is active and can be used or false when disabled.
- During the interview with business stakeholders we captured the following **Business Rules** related to a credit card:

- You cannot reserve or rent one of our vehicles without a credit card**
- A customer can have many credit cards they can use to pay for rental transactions.**
- A credit card can be owned by the one customer or co-owned by other individuals such a family member or corporate entity the customer works for. Therefore, many customers can own the same credit card and a credit card can be owned by many customers.**

Business Requirements

Our Customers (Cont.):

Corporate Customers

Corporate Customers are customers who are renting vehicle during business travel and their company have a contract with **EZRental Inc.** These companies get special corporate rate for their employee's rental services. Therefore, for our **corporate customers only**, we must store the following attributes/properties: unique **company ID** (we have a unique ID number for each company doing business with us), **company name, company address** which contains the elements: **address line1, address line 2** (which is optional and used for apartment number, suite or any additional address information required), **city, state code, zip code** (which is the two-character code for a state in the US) & **country**, in addition, **company contact** which is composed of **company representative name, contact phone number & contact email** (unique as all email addresses). And finally, we need to store the **company discount percentage rate** which is the discounted percentage applied to a corporate customers rental. The company Discount percentage rate is stored in the database as a decimal percentage value, for example 20% is stored as 0.20, 30% as 0.30. 50% as 0.50 etc. This discount percentage (0.ox) is applied to the **Vehicle Rental Categories** which determines the price of each category to determine the total discount. Therefore, when a corporate customer rents a vehicle from a vehicle category (such as economic, compact, standard etc.), this discount percentage is applied to each of the categories during the rental/reservation process. Note that every company has a different percentage rating depending on their contract with **EZ-Rentals Inc.** For example, some companies have 20% discount towards their rentals, which would be stored as 0.20 in the database, some have 30% (0.30) etc. Vehicle Rental Categories are discussed in more details later in these requirements.

Retail Customers

Retail Customers can (but don't have to) leverage promotional **discounts** or coupons obtain from other businesses, internet, magazine, organizations, etc., to save money on their rentals. Therefore, data unique to a retail customer that we need to capture for the promotional discount is unique random number **discount ID** which uniquely identifies a discount, a unique **discount code** or the coupon code itself used to redeem the coupon, which is an alphanumeric code **10-characters** long. This code is generated by our marketing team and published to magazines, newspapers, internet e-commerce sites, etc. Finally, the last attribute is **discount code description** or description of the discount. Examples of currently used **discount ID, discount code, discount code description** are shown in table below:

Discount ID	Discount Code	Discount Code Description
1234..	AAA9970054	AAA Membership Discount - 25% off base rate plus 10% donated for breast cancer research.
5678..	GOV8756921	Government Employee Discount - 30% off base rate
9101..	STA3415632	State Employee Discount for 25% off base rate
1213..	VET2055179	Veteran Discount 35% off base rate Plus 10% donation to veteran's family fund.
Etc..	Etc..	Etc..

Retail customers can opt-in to enrolled in the **EZPlus Rewards Program** where they earn points every time they rent and these points can be redeemed for future rentals. Note that the **EZPlus Rewards Program** is **optional** for retail customers & points are earned only when they rent vehicles. For the **EZPlus Rewards Program** we need to store unique random number **EZPlus ID**, the unique **Ezplus rewards code** which is the code used in the business when managing the **EZPlus Rewards Program**. This random code is generated and assigned to a **Retail Customer** by the client application. The number starts with the 3-characters EZP and a 10-digit number e.g., **EZP9999999999**, and the final attribute is the **EZPlus rewards earned points**, which is an integer that indicates the number of rewards points earned that accumulated after all the rentals and can be used to save on future rentals. **Examples of currently used EZPlus ID, EZPlus rewards Code and EZPlus earned points that we currently use are:**

EZPlus ID	EZPlus Rewards Code	EZPlus Rewards Earned Points
1234..	EZP9009854637	10000
5678..	EZP1000192461	500
9101..	EZP6493238865	159000
1213..	EZP2005135627	23000
Etc..	Etc..	Etc..

In this business, we have the following rules for our customers:

3. **We only have two types of customers retail customer or corporate customers. No other type of customer exists.**
4. **A customer cannot be a retail & corporate customer at the same time. A customer can only rent as a retail customer or as a corporate and these transactions must be separate. We don't want our customers to be able to combine both retail customer discounts, rewards program and corporate rates at the same time.**

Business Requirements (Cont.)

Our Vehicles:

EZ-Car Rental needs a system to manage their vehicles for renting, maintenance, selling, etc. Vehicles are classified into 4 main types: **CAR**, **SUV**, **MINIVAN**, and **CARGO VAN**. These are the vehicles most rented and available at every rental agency. Nevertheless, there are other categories of vehicles available only certain rental agency locations such as **RECREATIONAL VEHICLES**, **MOTORCYCLES**, **MOBILE HOMES**, etc. No matter what type of vehicle being rented, all vehicle types share the following common characteristics:

- Each vehicle is identified by the random number **vehicle ID**. In addition, each vehicle is also identified by the alpha-numeric **vehicle VIN number**. Note the following business rule on a **vehicle VIN number**:
1. **The vehicle VIN number is used throughout the business to identify a vehicle for searching, reporting etc.**
 2. **Therefore, the vehicle VIN number is the unique ID for a vehicle to be identified and managed from a business perspective.**
- Other attributes include the **vehicle name** composed of **make**, **model** & **year**. Additional attributes are **color**, also the **license plate** composed of the following components: **license plate number**, **license plate state**.
 - More attributes are **mileage**, **transmission type** of the vehicle. The Transmission Type attribute has business value thus used in reports and in the business processes. The values used for **transmission type** and a **transmission type description** as follows:

Transmission Type	Transmission Type Description
1	Manual Transmission
2	Automatic Transmission
3	Continuously Variable Transmission (e.g., CVT).
4	Semi-automatic Transmission
5	Dual-clutch Transmission
6	Transaxle Transmission

- **seat capacity** attribute, which is the number of seats in the vehicle. Vehicles such as *cars* have a seat capacity of 5 passengers (2 in front and 3 in the back), *SUVs* have 7 or 8 passengers. Cargo Vans have only 2 passenger seat capacity, Minivan have 8 to 9 passengers, special vehicles such as passenger van hold 12 passenger seat capacity, a shuttles bus can hold 16 to 20 passengers, mini-buses 30 to 40 passengers and large busses can hold 70 passengers.
- All vehicles also have a special code and description that we use to track the vehicle status named **vehicle status ID**. This is a unique number that identifies the status of a vehicle, which works in conjunction with **vehicle status description** which describes the status represented by the **Vehicle Status ID**, such as **reserved**, **rented**, **available**, **maintenance**, **not available**, **transferred**, etc. Below Is the list of vehicle status IDs we are currently using and their descriptions:

Vehicle Status ID	Vehicle Status Description
1	Available
2	Reserved
3	Rented
4	Not available
5	Maintenance (Not available)
6	Dropped off and located at another agency
7	In Transport to Owning Agency
8	No Longer available for rental

Business Requirements (Cont.)

Our Vehicles (Cont.):

In addition to these attributes shared by all vehicles, there are 4 main categories of vehicle which share unique characteristics than the other types of vehicles found in our agencies. These 4 types are as follows:

- A **Car** is a vehicle whose **trunk capacity** (measured in cubic feet volume) is advertised to our customers. Customers can decide which vehicles better fits their needs based on the trunk capacity and number of luggage they are carrying etc. For example, a *luxury Mercedes E class* car has a trunk capacity of 18.5 cubic ft., which has a large trunk capacity.
 - An **SUV** is a vehicle with a **towing capacity** attribute in pounds. Towing capacity is a single number in pound or could also be a decimal number in pounds. For example, some of our SUV have a maximum towing capacity of 3,000 pounds etc. Another attribute of SUV is an attribute classification if the SUV is **All-Wheel-Drive**, which stores a Boolean value of **YES/NO** or **TRUE/FALSE**.
 - A **Minivan** has the option of **having a disability package**, which is also a Boolean value of **YES/NO** or **TRUE/FALSE**.
 - Finally, a **Cargo Van**, has a **cargo capacity** in cubic feet volume. For example, the typical volume of our Vans is 245 cubic feet (cu.ft.). Cargo Vans also have a **maximum payload** attribute that determines how much weight in pound it can hold. Our cargo vans have typically a maximum payload of 3,880 lbs.
-
- As stated previously, there are other types of vehicles of interest that in some location we may want to store data on other than car, SUV minivans and cargo van.
 - Note that the following Business Rules were identified by the business stakeholders on the vehicles:

1. **A reservation/rental can only be for one of these four categories of Vehicles or other vehicle types, not a combination.**
2. **This means, you can only rent either a car, SUV minivans, cargo van or other for a reservation or rental, not a combination such as a car & SUV at the same time. Each reservation is unique to one vehicle.**

Below are additional business rules for our vehicles and agency ownership:

1. **Every vehicle is owned by one agency. The vehicle can be pick-up and dropped-off at any agency, but only one agency is the vehicle's owning agency. An agency can own many vehicles, but a vehicle can only be owned by one agency.**
2. **A vehicle can currently be located at any agency depending on where it was dropped-off after a rental. We need to track the current agency where the vehicle is located, to arrange a transfer or a rental that will ultimately direct the vehicle to the owning agency.**

Reservation Process:

A vehicle must be reserved if a customer wants to guarantee the vehicle will be available for rental. There is a distinction between a reservation and a rental. A reservation guarantees a vehicle will be ready for you to be pick-up and rented. A rental means a customer complied with the reservation and rented the vehicle. On the other hand, a customer can walk into an agency and rent without reservation but only vehicles that are available at the time and not reserved.

We have the following business rules for reserving a vehicle reservation:

1. **A reservation is NOT made for a specific vehicle, but to a vehicle rental category. Rental category examples are economy, intermediate, full size, luxury.**
2. **Thus, a customer makes a reservation of a vehicle rental category at a rental agency. Therefore, the reservation process involves a customer a vehicle rental category and the rental agency where the vehicle will be picked up.**

Business Requirements (Cont.)

Reservation Process (Cont.):

A **Vehicle Rental Category** contains a list of vehicles depending on the vehicle type: Car (economy, intermediate, full size, luxury), SUV (standard, full size etc.), or Cargo Van etc. Each of these categories have a different price range. Therefore, for a vehicle rental category we need to capture the unique **vehicle rental category ID** that identifies the category of the vehicle being reserved or rented, **category name** and finally **category daily rental rate** for the category. We used a specific code for our vehicle rental category ID, category name & daily rental rate. The table below shows the ID, category names and rate we currently using in our business:

<i>Vehicle Rental Category ID</i>	<i>Vehicle Rental Category Name</i>	<i>Category Daily Rental Rate</i>
1	Car-Economic	\$113.99
2	Car-Compact	\$115.99
3	Car-Intermediate	\$116.67
4	Car-Standard	\$119.99
5	Car-Full Size	\$121.99
6	Car-Premium	\$127.79
7	Car-Luxury	\$139.99
8	SUV-Intermediate	\$127.99
9	SUV-Standard	\$128.99
10	SUV-Standard Elite	\$135.99
11	SUV-Full Size	\$148.99
12	SUV-Premium	\$157.99
13	Minivan-Standard	\$152.99
14	Van-Cargo Van	\$19.95
15	Pick Up-Mid Size	\$69.95
16	Pick Up-Full Size	\$105.99
17	Motorcycle-Touring	\$19.95
18	Motorcycle-Cruiser	\$199.99
19	Motorcycle-Scooter	\$79.95
20	Passenger Van (12 passengers)	\$161.00
21	Passenger Shuttle (16 passengers)	\$180.00
22	Passenger Shuttle (20 passengers)	\$220.00
23	Passenger Mini-Bus (30 passengers)	\$250.00
24	Passenger Mini-Bus (40 passengers)	\$280.00
25	Passenger Large-Bus (80 passengers)	\$300.00

We

1. A vehicle is a member of a vehicle rental category.
2. A vehicle rental category can have one, none or many vehicles belonging to that category at any given time, nevertheless, a vehicle can only belong to one vehicle rental category.

As stated previously, a **customer makes a reservation of a vehicle rental category at a rental agency**. Therefore, the reservation process requires the **customer, vehicle rental category & rental agency** for a reservation to be made. The following business rules apply to a reservation:

1. A vehicle can be reserved to be picked up at the INDICATED rental agency and dropped off at the SAME rental agency.
2. A vehicle can be reserved to be picked up at the INDICATED rental agency and dropped off at a DIFFERENT rental agency.
3. A reservation is made only for one pick-up rental agency, but a rental agency can have many reservations for pick-ups taking place.
4. A reservation can only be for one drop-off rental agency, but a rental agency can have many reservations drop-offs taking place.

When a customer reserves a vehicle rental category for a specific rental agency, we wish to capture the following:

- A unique **reservation ID** which is used by the business to manage and track reservations, the **rental agency ID** where the vehicle will be picked up, and the target **reservation drop-off rental agency**.
- In addition, we need **reservation pick up date**, **reservation pick up time**, **reservation drop off date** and **reservation drop off time**, also the **reservation estimated rental cost**.

Business Requirements (Cont.)

Reservation Process (Cont.):

- Finally, we need to store the unique **reservation status ID** which is a unique number we use to indicate the status of a reservation and **reservation status description** which describe each of the status such as: **confirmed**, **cancelled**, **completed** etc. Below is an example of the **reservation status ID** and **status description** we currently use in our business.

Reservation Status ID	Reservation Status Description
1	Confirmed
2	Modified & reconfirmed
3	Cancelled
4	Fulfilled & closed
Etc..	Etc..

For a reservation \

- A customer can make none, one or many reservations for a vehicle rental category at a rental agency.**
- A rental category can be reserved by none, one or many customers at a rental agency.**
- A rental agency can get many or no reservations for a vehicle rental category by a customer.**
- A reservation can only have one pick-up rental agency location, but a rental agency can have many reservation pick-ups happening.**
- Each reservation has a drop-off rental agency (may be different than pick-up rental agency). A reservation can only have one drop-off rental agency location, but a rental agency can have many reservation drop-offs taking place.**

The Rental Process:

Once a vehicle has been reserved, the vehicle can be rented (picked up/dropped off) as per the scheduled of the reservation agreement. A rental means a customer complied and fulfilled the reservation and rented the vehicle.

For the rental process, the following business rules apply:

- A customer rents a vehicle Rental Category at a rental agency. This means the rental process requires the customer, vehicle rental category, and & rental agency for a rental to be complete.**
 - A Rental includes a specific Vehicle of the vehicle rental category. A vehicle can be rented many times, but a rental is only for one vehicle only. You cannot rent multiple vehicles in one rental contract.**
 - During the rental process we may have any of the following business rules/scenarios:**
 - A vehicle can be picked up at the SAME rental agency as indicated by the reservation and dropped off at the SAME rental agency.**
 - Or a vehicle can be picked up at the SAME rental agency as indicated by the reservation and dropped off at ANOTHER rental agency.**
 - Or a vehicle can be picked up at ANOTHER rental agency other than what was indicated by the reservation and dropped off at SAME rental agency of the reservation.**
 - Finally, a vehicle can be picked up at ANOTHER rental agency other than what was indicated by the reservation and dropped off at ANOTHER rental agency of the reservation.**
- ❖ Note that for scenarios 3 & 4, we cannot guarantee that the vehicle rental category of the reservation will be available at the agency other than what was agreed in the reservation. We will do our best to accommodate the change during these scenarios or find another vehicle that will be closed to the original reservation.

For the rental process, the following business rules also apply:

- A rental can only be for one pick-up rental agency, but a rental agency can have many rental pick-ups taking place.**
- A rental can only be to one drop-off rental agency, but a rental agency can have many rental drop-offs taking place.**

When a customer rents a vehicle at the rental agency, we need to capture the following information about the rental:

- The **rental agreement ID** that uniquely identifies the rental transaction, **rental pick up date**, **rental pick up time**, **rental drop off date** and **rental drop off time**, **rental pick up odometer value** and **rental drop off odometer value**.

Business Requirements (Cont.)

The Rental Process (Cont.):

- In addition, customers receive a vehicle with a full tank of gas and customers are expected to return the car on a full tank of gas otherwise they must pay a penalty upon return. Since we understand our customers are busy and may forget to return the car with a full tank of gas, we offer our customers with the option to pay in advance for a full tank of gas at our rates and don't have to worry about returning the vehicle with a full tank of gas. Therefore, we need to capture the unique ***rental fuel option ID*** or option chosen by the customer, ***rental fuel option description*** and ***rental fuel option additional cost***. We currently use the following fuel option IDs, descriptions, and example of each of the additional cost for the fuel option:

Rental Fuel Option ID	Rental Fuel Option Description	Rental Fuel Option Additional Cost
1	Return with a full tank or on return, pay for gas that is missing.	\$13.97 <i>(Important, this Decimal value of \$13.97 is just an example, since the value is calculated during car return process and is based on the current price for a gallon of gas etc. therefore price will vary.)</i>
2	Pay for full tank in advanced at time of rental, return car empty. No refund for unused gas.	\$45.99 <i>(Important, this Decimal value of \$45.99 is just an example, since the value is calculated during car return process and is based on the current price for a gallon of gas etc. therefore price will vary.)</i>

- Also, we give customer options for car insurance & protection, therefore we need to capture the unique ***insurance option ID***, ***insurance option description*** and ***insurance option additional cost***. We currently use the following insurance option IDs,

Rental Insurance Option ID	Rental Insurance Option Description	Rental Insurance Option Additional Cost per Day
1	No insurance. Opt-out.	\$0.00
2	Collision Damage Waiver Max - Agency will pay for damage, lost or stolen vehicle.	\$49.99
3	Collision Damage Waiver 3000 - Agency will pay for first \$3,000 of loss or damage, renter pays all loss & damage after \$3,000.	\$39.99
4	Liability Extended Protection – Agency provides renter with third party liability protection up to \$1 Million per accident for bodily injury or death or property damage to others.	\$89.99
5	Roadside Assistance Plus – 24/7 roadside assistance, replacement for lost keys, flat tire service, fuel delivery, etc.	\$15.99

- Other attributes required for the rental that we need to capture are the unique ***rental status ID*** & ***rental status description***. We currently use the following rental status IDs & descriptions:

Rental Status ID	Rental Status Description
1	Picked up as scheduled.
2	Dropped off as scheduled.
3	Returned late
4	In progress.
5	Roadside assistance in progress.
7	Unknown

Business Requirements (Cont.)

The Rental Process (Cont.):

- Other attribute we need to capture the **rental deposit** for a rental. The rental deposit value is calculated based on the **rental period + 25% of the rental period** and for any damage or other charges that were incurred during the rental period. This deposit is refunded to the customer's credit card when the vehicle is returned in the condition in which it was rented.
- Finally another attribute we need to capture is the **rental total cost** or total cost that needs to be paid by the customer. This value is calculated based on selected *fuel option, insurance option, vehicle rental category* price and other factor such as duration of the rental etc.

We need to be able to associate a reservation to a rental and vice versa, therefore we maintain the following additional business rules for our rental & reservation:

1. **A reservation is made for a rental and the opposite holds true; a rental is based on a reservation.**
2. **But NOT all rentals are based on a reservation. We allow a customer to walk into a rental agency and rent a vehicle without a reservation.**
3. **When a reservation is made for a rental, then it must be for only one rental, and a rental can be for a reservation but not mandatory since a customer can walk into an agency and rent a vehicle without a reservation.**

Our Employees:

EZ-Car Rental currently has 5,500 employees across the world. We do expect to grow as we move into new markets such as Asia, Africa, and the Mediterranean. But our business does not require a large workforce, therefore, we don't expect to grow more than 12,000 in the next 10+ years. Our employees consist of customer service agents in the Rental Agencies & online support who interact with our customer to reserve and rent vehicles. In addition, back-office inventory personnel, auto specialists who work in our services centers servicing our vehicles, drivers to transport our vehicles from one agency to another and maintenance personnel who maintain our agencies and finally our business team that handles the day-to-day business activities in our agencies and other roles. For now, we are only interested in storing the following data for all these types of employees:

- An **Employee ID** which uniquely identifies the employee, **employee name** which is composed of: **first name, last name**, also **employee address** which includes the components: **address line1, address line 2, city, state code, zip code & country**. Also, **employee phone, employee job title** and **employee email**. In addition, we need to capture the employee **social security number**. Below are some business rules and usage for the **EmployeeID** and the **social security number**.

1. The employee **social security number** needs to be protected and secured as per federal regulations. All security measures such as encryption, etc., need to be taken to protect the **social security number**; therefore, the full **social security number** **cannot** be seen by employees, reports, and other business processes.
2. In special cases where the **social security number** needs to be displayed, only the last 4 digits will be shown using the following format ******_**_1234**. Nevertheless, the goal is **NOT** to display the **social security number** as much as possible, and it should only be used internally within the application for processing but not displaying.
3. The **EmployeeID** number is what is used throughout the business to identify an employee for searching, reporting, business processing, etc.
4. Therefore, the **EmployeeID** is the unique ID for an employee to be identified and managed from a business perspective.

Security & Application Access:

To access our systems proper security and authentication is required. Only authorized users can have access our agencies Point-of-Sales & Back-End Management systems. In addition to our **EZRental.com** portal by our customers. Therefore, due to security and regulatory compliance purpose, we want to separate the employee access data from the customer access data by using two separate user accounts:

- Employee user accounts
- Customer user accounts

Security Access for Employees to Computer Systems in our Agencies (Employee User Accounts):

For our authorized employees & customer service employees to access the agencies Point-of-Sales & Back-End Management systems they need to log in by entering a username & password for access to the application. This means every employee owns an employee user account.

An employee user account should store the user **employee user account ID** a unique identifier alpha-numeric string that identifies the employee user account, **employee username** another unique alpha-numeric that identifies each individual user, the **employee password** alpha-numeric that is known only to the user, and finally the employee **email** to map the user-account to an Employee. Note the following business rule:

1. An employee can own one employee user account only, and an employee user account can only be owned by one employee only since the user account represents the identify of that one employee.

Business Requirements (Cont.)

Security Access for our Customers who register for our EZ-CarRental.com web site (Customer User Accounts):

Customer who accesses our online portal to reserve and rent our vehicles also need a username and password to access our system, therefore each customer owns a customer user account.

A customer user account should store the user **customer user account ID** a unique alpha-numeric string identifier that identifies the customer user account, **customer username** another unique alpha-numeric value that identifies each customer, the **customer password** that is an alpha-numeric known only to the customer, and finally, the customer **email** to map the customer user-account to a customer. Note the following business rule:

1. A customer can own one customer user account only, and a customer user account can only be owned by one customer.
2. For a period of time, we will need to register customers into our **EZRental.com** business, nevertheless the web portal may NOT be implemented or completed when new customers are registering at this time, therefore, for period of time, creating a customer user account when registering a new customer is optional until the Web Portal Application is created. But is important in the future, that we force the creation of customer user accounts when a new customer is registered once the Web Portal Application is ready. It is the responsibility of the database architect(s) and full-stack developers to update this feature when the appropriate time comes.

Vehicle Transportation:

We need to know where our vehicles are located at all times, such as at the Rental Agency that owns the vehicle, another Rental Agency that does not own the vehicle, being transported from one Rental Agency to another as a result of a vehicle transfer after a rental to the owning rental agency, being transported as a new delivery to a Rental Agency from our distribution center, being transported for maintenance, or currently being rented by a customer. Vehicles need to be tracked or location status known. At this time, we are only interested in tracking when a vehicle is transported from one Rental Agency to another Rental Agency under the following scenarios:

- Vehicle can be located at a Rental Agency that does not own the vehicle after a rental dropping off at a different location than the picked up owning Rental Agency, thus vehicle eventually needs to be transported and delivered to the owning agency.
- Another non-owning Rental Agency requests support from other Rental Agency(s) for loans of vehicle(s) to borrow due to an unexpected busy period and requesting agency is short on inventory. After the first agency is done with the loaner vehicles, these vehicles need to be returned to the borrowed owning Rental Agency(s).
- In our current process & systems we currently use the following reason IDs and reason descriptions:

Transport Reason ID	Transport Reason Description
1	Rental Drop off at different location
2	Vehicle Loaned to another Agency
3	Pick up from Distribution Center
4	Drop off to Distribution Center
5	Vehicle sent for maintenance
7	Unknown

Note that transportation events involve multiple agencies, drivers, and drivers. Therefore, when an employee executes a transport request of a vehicle to and from Rental Agencies, we need to capture the following information:

- **Transport pickup agency ID, Transport drop-off agency ID, Driver departure date, driver departure time, vehicle pick up date, vehicle pick up time, transport completed arrival date, transport completed arrival time, estimated arrival date, estimated arrival time, & actual transport time to completion.**
- In addition, we need to know at any time the transport status and transport status description of the transfer, such as: transfer completed, on route to pick up location, on route from pick up location, etc. Therefore, we need to capture the **Transport Status ID** or unique number that identifies a status and the **Transport Status Description**, or description of each status ID. Currently we track a transportation event using the following ID and description:

Transport Status ID	Transport Status Description
1	Transport completed
2	On route to pick up location.
3	On route from pick up location
4	At pickup location. In progress (Loading etc.)
5	Pickup location delay
7	Unknown

The goal again is

Business Requirements (Cont.)

Conclusion:

The business data listed in this business requirements document is what we need to capture for our business to operate. As our business evolves, additional data will be required in the future. We will address these new requirements in future versions of the application. For example, invoice processing & employee management at our rental agencies are features on our roadmap. Therefore, our expectations are that the design is modular and scalable for future growth.

Application Development & Technical Requirements

An **Application Analyst/Architect** interviewed **EZRental Inc.**, project **Business Decision Makers (BDMs)**, **stakeholders** and **Information System Technical Decision Makers (TDMs)** to compile the list of the **Application Development & Technical Requirements** in order to design and develop the application.

Below are the **Application Development & Technical Requirements** captured by the Application Analyst/Architect:

Application Development & Technical Requirements

Introduction & Current Challenges

As described in the Business Requirements, the current rental system is outdated, with a poor user-experience, breaks often thus expensive to operate, does not meet our business requirements, and is not scalable so it cannot be easily updated with new features etc. Also, not elastic since it does not give us the flexibility to scale-up or scale-down based on business trends and seasonal changes in the market. We want to invest in modernizing our business with a new vehicle management system that can meet these challenges and give us a great user-experience, meet new business requirements, scalable, and elastic to adopt to business trends and seasonal market changes.

We have an outdated IT infrastructure in our datacenter and there is a current initiative to modernize our datacenter and also leverage cloud technology in a hybrid environment to save on cost, streamline our operations and drive innovation.

We look forward to your proposed architecture & implementation of this new system that will meet these requirements. Next sections contain the results of our application development & technical requirements.

Rental Agencies Application & Technical Requirements:

The rental agencies are location where customers both Retail & Corporate will engage our *Customer Service Representatives* to engage in rental/return activities in addition to other transactions such as registering, searching & updating customer information etc. Therefore, the application in the rental agencies is vital to the user-experience for both our *Customer Service Representatives* as well as our *customers*.

We are forecasting that in some locations such as major city centers and airports, there will be many customers engaging throughout the day thus increasing the risk of a poor customer experience in addition to the work overload and poor experience for our *Customer Service Representatives*. We want our *Customers* to be serviced quickly and efficiently with a great experience, and our *Customer Service Representatives* to be able to process each *Customer* easily and effectively. With these criteria in mind, the application at our rental agencies must adhere to the following requirements:

Rental Agency Application Architecture Requirements:

Below are the requirements for the application used in our rental agencies by our customer service representatives, inventory team, service personnel and other employees working in our agencies:

1. Client application processing, transaction and response must be fast to minimize service time for a customer.
2. All transaction processing should be done in the user's computer or desktop for fast processing and response.
3. Application Architecture must be reusable and scalable to support future updates and new feature enhancements, without a long development lifecycle.
4. Depending on the architecture NYC-Tech Solutions Inc., decides for the application in the rental agencies (Desktop client or Web client), the primary Application Development Platform we use is **C# & .NET technologies**. For any Web related development, we support JavaScript, React, NodeJs and other standard Web Technologies. We have aligned **C#.NET & ASP.NET Web developers** that have been assigned to assist, support and update the application once NYCTech consultants complete the project and development of this system.
5. Rental Agency Desktop Application Security Authentication System – Proper security and authentication must be implemented to make sure only authorized customer service representative and other rental office employees can access the Point-of-Sales with appropriate conditional access.

Application Development & Technical Requirements (Cont.)

Rental Agency Application Features and Functionalities Requirements:

The list of features and functionalities that we have compiled for the rental agencies' application are listed in the table below:

No.	Feature	Functionalities
1	EZRental Rental Agency Point-of-Sales (POS) System	<ul style="list-style-type: none"> ▪ Car Rental, Car Return, New Customer Registration & Search/Print Customer Information, Customer Update, Customer Deletion, Customer Listing operations etc.
2	EZRental Rental Agency Back-Office Vehicle Inventory Management System	<ul style="list-style-type: none"> ▪ Back-office system meant for employees to perform bulk IN-MEMORY inventory processing or management tasks on vehicles such as adding vehicles to the system, searching for vehicles, updating vehicles etc. ▪ This system is NOT meant for Point-of-Sales, but for the inventory management employees who need to search, add, remove etc., a large/bulk number of vehicles or employees during a session. ▪ Back-office vehicle Management features – Allows inventory personnel and employees to bulk-manage Cars, SUVs, Mini-Vans, Cargo Vans to be searched, added, removed, printed, listed etc.
3	EZRental Rental Agency Back-Office Credit Card Management System	<ul style="list-style-type: none"> ▪ The EZRental Credit Card Management System is a Back-office system meant for the Credit Card Department Employees to manage Credit Card Information. These uses can Search/Print, Add, Edit & Delete credit card information in the database
4	EZRental Rental Agency Back-Office Employee & Customer User Account Management System	<ul style="list-style-type: none"> ▪ The EZRental Customer & Employee User Account Management System is a Back-end system meant for IT ADMINISTRATOR Employees to manage both Employee & Customer USER ACCOUNTS.
5	EZRental Rental Agency Desktop Application Security Authentication System	<ul style="list-style-type: none"> ▪ Proper security and authentication must be implemented to make sure only authorized employees can access the Point-Of-Sales, Back-End Management system or any other access to the applications.

Rental Agency Application Graphical User Interface Requirements:

- Graphical User-Interface should be fast rendering and user-friendly workflow.
- Visual screens or forms should be rich in color and appearance and navigation flow should be flexible and easy.
- The following UI controls or data field need to be pre-populated in GUI Screens:
 - **Addresses**
 - Any forms/UI which contains addresses, the STATE & COUNTRY fields should be automatically populated with a list of STATES or COUNTRIES, so the user does not have to manually enter a state or a country and simply select from drop-down list etc.
 - **Discount Codes:**
 - UI screens with customer's DISCOUNT CODE fields should be prepopulated with discount codes. The idea is the user should be able to select the discount to apply to a customer entry from a drop-down list/Combo Box etc. Note that this may or may not include the Discount Code Description on the UI screen as well.
 - Also note that the DISCOUNT CODE VALUES are generated by our Marketing Team and need to be pre-populated in the database before a code can be used. Therefore, the discount codes are prepopulated in the database.
 - Currently, when the Marketing Team generates a new code, they make the request to the database administrator to manually enter an update any new Discount Codes.
 - In the future, we want the application to have the necessary features for the Marketing Team to be able to manage the discount codes. This is not an immediate requirement out of the gate but should be targeted as part of a future upgrade.

Application Development & Technical Requirements

Rental Agency Application Graphical User Interface Requirements (Cont.):

- **EZPlus Rewards Codes:**
 - The EZPlus Reward UI screens with customer's EZPLUS REWARDS CODE fields should be prepopulated with the EZPlus Rewards code for the customer is being applied to. The idea is the user should be able to select the EZPLUS REWARD CODE to apply to a customer entry from a drop-down list/Combo Box etc. or be handled by the back-end database.
 - **Important:** The EZPLUS REWARDS CODE VALUES are NOT generated by a business entity in our organization, but AUTOMATICALLY GENERATED by the application on the fly when registering a new customer. This is a different approach compared to the DISCOUNT CODE which are generated by Marketing Team. In this case, the EZPlus Rewards Code values are generated by the application and available via the UI screen to be used or some other method of generation.
 - To finalize this requirement, the idea is the EZPlus Rewards Code should be automatically generated and either appear in the UI Screen or automatically generated in the database.
- **Company Name:**
 - UI screens with corporate customer's COMPANY NAME fields should be prepopulated with the list of corporations that are members of our corporate program, which enables our users to avoid having to manually enter the company name. Note that this may or may not include the Company ID in the UI Screen which is a unique number with business value that we assign to each company.
 - Note that the company names. Company ids and other company data are managed by our Corporate Sales Team and need to be pre-populated in the database before any corporate customer processing can be made. Therefore, the company information is prepopulated in the database.
 - Currently, when the Corporate Sales Team adds a new corporation or company into the program, they make the request to the database administrator to manually enter and add the new company to the database.
 - In the future we want the application to have the necessary features for the Corporate Sales Team to have the functionality to manage the data of our corporate companies via the application. This is not an immediate requirement out of the gate but should be targeted as part of a future upgrade.
- **Vehicle Status:**
 - UI screens for vehicle inventory management, VEHICLE STATUS field should be prepopulated with the list of vehicle status. Based on the business requirements, the current list of vehicle status is listed in table below:

Vehicle Status ID	Vehicle Status Description
1	Reserved.
2	Rented.
3	Available.
4	Not available
5	Maintenance
6	Transferred to another agency

 - Currently populating the database with a vehicle status record is handled manually by the database administrator. In the future we would like the application to have the necessary features for our business to be able to manage the vehicle status data. This is not an immediate requirement out of the gate but should be targeted as part of a future upgrade.
- **Rental Agency:**
 - UI screens that required adding or managing a RENTAL AGENCY field should be prepopulated with the list of rental agencies in our company.
 - Currently populating the database with a rental agency record is handled manually by the database administrator. In the future we would like the application to have the necessary features for our business to be able to manage the rental agency data. This is not an immediate requirement out of the gate but should be targeted as part of a future upgrade.

Application Development & Technical Requirements

Rental Agency Application Graphical User Interface Requirements (Cont.):

- **Vehicle Rental Category:**

- UI screens that require the use of the VEHICLE RENTAL CATEGORY fields, must be prepopulated with the list of vehicle rental categories. Based on the business requirements, the current list of vehicle rental categories is as follows:

<i>Vehicle Rental Category ID</i>	<i>Vehicle Rental Category Name</i>	<i>Category Daily Rental Rate</i>
1	Car-Economic	\$113.99
2	Car-Compact	\$115.99
3	Car-Intermediate	\$116.67
4	Car-Standard	\$119.99
5	Car-Full Size	\$121.99
6	Car-Premium	\$127.79
7	Car-Luxury	\$139.99
8	SUV-Intermediate	\$127.99
9	SUV-Standard	\$128.99
10	SUV-Standard Elite	\$135.99
11	SUV-Full Size	\$148.99
12	SUV-Premium	\$157.99
13	Minivan-Standard	\$152.99
14	Van-Passenger Van (12 passengers)	\$161.00
15	Van-Cargo Van	\$19.95
16	Pick Up-Mid Size	\$69.95
17	Pick Up-Full Size	\$105.99
18	Motorcycle-Touring	\$19.95
19	Motorcycle-Cruiser	\$199.99
20	Motorcycle-Scooter	\$79.95

- Currently populating the database with vehicle rental category records is handled manually by the database administrator. In the future we would like the application to have the necessary features for our business to be able to manage the vehicle rental categories data. This is not an immediate requirement out of the gate but should be targeted as part of a future upgrade.

- **Reservation Status:**

- UI screens that require the use of the RESERVATION STATUS field, must be prepopulated with the list of reservation status data. Based on the business requirements, the current list of reservation status is as follows:

<i>Reservation Status ID</i>	<i>Reservation Status Description</i>
1	Confirmed.
2	Modified & reconfirmed.
3	Cancelled & Closed.
4	Fulfilled & Closed.
Etc..	Etc..

- Currently populating the database with a reservation status record is handled manually by the database administrator. In the future we would like the application to have the necessary features for our business to be able to manage the reservation status data. This is not an immediate requirement out of the gate but should be targeted as part of a future upgrade.

Application Development & Technical Requirements

Rental Agency Application Graphical User Interface Requirements (Cont.):

- **Rental Status:**

- UI screens that require the use of the RENTAL STATUS field, must be prepopulated with the list of rental status data. Based on the business requirements, the current list of rental status is as follows:

<i>Rental Status ID</i>	<i>Rental Status Description</i>
1	Picked up as scheduled.
2	Dropped off as scheduled.
3	Returned late
4	In progress.
5	Roadside assistance in progress.
7	Unknown

- Currently populating the database with a rental status record is handled manually by the database administrator. In the future we would like the application to have the necessary features for our business to be able to manage the rental status data. This is not an immediate requirement out of the gate but should be targeted as part of a future upgrade.

- **Rental Fuel Option:**

- UI screens that require the use of the RENTAL FUEL OPTION field, must be prepopulated with the list of rental fuel options data. Based on the business requirements, the current list of rental fuel option is as follows:

<i>Rental Fuel Option ID</i>	<i>Rental Fuel Option Description</i>	<i>Rental Fuel Option Additional Cost</i>
1	Return with a full tank or on return, pay for gas that is missing.	\$13.97 <i>(Important, this Decimal value of \$13.97 is just an example, since the value is calculated during car return process and is based on the current price for a gallon of gas etc. therefore price will vary.)</i>
2	Pay for full tank in advanced at time of rental, return car empty. No refund for unused gas.	\$45.99 <i>(Important, this Decimal value of \$45.99 is just an example, since the value is calculated during car return process and is based on the current price for a gallon of gas etc. therefore price will vary.)</i>

- Currently populating the database with a rental fuel option record is handled manually by the database administrator. In the future we would like the application to have the necessary features for our business to be able to manage the rental fuel option data. This is not an immediate requirement out of the gate but should be targeted as part of a future upgrade.

Application Development & Technical Requirements

- **Rental Insurance Option:**

- UI screens that require the use of the RENTAL INSURANCE OPTION field, must be prepopulated with the list of rental insurance options data. Based on the business requirements, the current list of rental insurance option is as follows:

<i>Rental Insurance Option ID</i>	<i>Rental Insurance Option Description</i>	<i>Rental Insurance Option Additional Cost per Day</i>
1	No insurance. Opt-out.	\$0.00
2	Collision Damage Waiver Max - Agency will pay for damage, lost or stolen vehicle.	\$49.99
3	Collision Damage Waiver 3000 - Agency will pay for first \$3,000 of loss or damage, renter pays all loss & damage after \$3,000.	\$39.99
4	Liability Extended Protection – Agency provides renter with third party liability protection up to \$1 Million per accident for bodily injury or death or property damage to others.	\$89.99
5	Roadside Assistance Plus – 24/7 roadside assistance, replacement for lost keys, flat tire service, fuel delivery, etc.	\$15.99

- Currently populating the database with a rental insurance option record is handled manually by the database administrator. In the future we would like the application to have the necessary features for our business to be able to manage the rental insurance option data. This is not an immediate requirement out of the gate but should be targeted as part of a future upgrade.

Application Development & Technical Requirements

- **Transportation Reason Option:**

- UI screens that require the user to populate the TRANSPORTATION OPTIONS field, must be prepopulated with the list of transportation reason options as shown in the table below:

Transport Reason ID	Transport Reason Description
1	Rental Drop off at different location
2	Vehicle Loaned to another Agency
3	Pick up from Distribution Center
4	Drop off to Distribution Center
5	Vehicle sent for maintenance
7	Unknown

- Currently populating the database with a transportation reason option record is handled manually by the database administrator. In the future we would like the application to have the necessary features for our business to be able to manage the transportation reason option data. This is not an immediate requirement out of the gate but should be targeted as part of a future upgrade.

- **Transportation Reason Option:**

- UI screens that require the user to populate the TRANSPORTATION STATUS field, must be prepopulated with the list of transportation status options as shown in the table below:

Transport Status ID	Transport Status Description
1	Transport completed
2	On route to pick up location.
3	On route from pick up location
4	At pickup location. In progress (Loading etc.)
5	Pickup location delay
7	Unknown

- Currently populating the database with a transportation status option record is handled manually by the database administrator. In the future we would like the application to have the necessary features for our business to be able to manage the transportation status option data. This is not an immediate requirement out of the gate but should be targeted as part of a future upgrade

Application Development & Technical Requirements (Cont.)

Customer Facing Self-Service Web-Portal Application Architecture Requirements:

We now address architecture requirements for the application used in customers via the public internet to make reservations to rent a vehicle, modify their personal account, profile etc.:

1. Customer will use a secure and standard Web Application via a Browser to access our self-service portal in the internet. We need a website to support all customer self-service related transactions.
2. Web Application Architecture must be reusable and scalable to support future updates and new feature enhancements, without a long development lifecycle.
3. For this web development, we support **JavaScript, React, NodeJS** and other standard Web Technologies. In addition, the primary Application Development Platform we use is **C# & .NET technologies**. We have aligned **C# & .NET** & **Web** developers that have been assigned to assist, support, operate and update the application once NYCTech consultants complete the project and development of this system.
4. Web Portal Security Authentication System – Proper security and authentication must be implemented to make sure only the customer can access the **EZRental.com** website for his or her profile home page.

Customer Facing Self-Service Web-Portal Features and Functionalities Requirements:

No.	Feature	Functionalities
1	EZRental.com Customer Web Portal	<ul style="list-style-type: none">▪ Front-end WEB INTERFACE SCREENS & features used by customers via our web portal EZRentalCar.com to reserve a vehicle for rental and manage their account online.▪ Features include search & reserve a car for rental, register as a new customer, search/view their account information, update their account etc.
2	EZRental.com Customer Web Portal Application Security Authentication System	<ul style="list-style-type: none">▪ Proper security and authentication must be implemented to make sure only our customer can access the web portal to use the application.

Web Portal Application Web Pages User Interface Requirements:

The web pages graphical UI requirements are listed below:

- The GUI requirements for the web pages are like those functionalities of the Rental Agency Application that are found on the web site for example Search & reserve a car for rental, register as a new customer, search/view their account information, update their account etc.
- The design and graphics of the application should be appealing to customers and a smooth and fluent workflow.
- The following UI controls or data field need to be pre-populated in GUI Screens:
 - **Addresses**
 - Any web-page UI which contains addresses, the STATE & COUNTRY fields should be automatically populated with a list of STATES or COUNTRIES, so the user does not have to manually enter a state or a country and simply select from drop-down list etc.
 - **Discount Codes:**
 - Web pages with customer's DISCOUNT CODE fields should be a text box that allows the customer to ADD/APPLY the discount codes to redeem the coupon.

Application Development & Technical Requirements

Rental Agency Application Graphical User Interface Requirements (Cont.):

- **EZPlus Rewards Codes:**
 - The EZPlus Reward web page screens with customer's EZPLUS REWARDS CODE fields should be prepopulated with the EZPlus Rewards code for the customer is being applied to. The idea is the user should be able to select the EZPLUS REWARD CODE to apply to a customer entry from a drop-down list/Combo Box etc. or be handled by the back-end database.
 - **Important:** The EZPLUS REWARDS CODE VALUES are NOT generated by a business entity in our organization, but AUTOMATICALLY GENERATED by the application on the fly when registering a new customer. The EZPlus Rewards Code values are generated by the application and available via the UI screen to be used or some other method of generation.
 - To finalize this requirement, the idea is the EZPlus Rewards Code should be automatically generated and either appear in the UI Screen or automatically generated in the database.
- **Rental Agency:**
 - Web pages that required adding a RENTAL AGENCY field should be prepopulated with the list of rental agencies in our company.
- **Vehicle Rental Category:**
 - Web pages that require the use of the VEHICLE RENTAL CATEGORY fields, must be prepopulated with the list of vehicle rental categories. Based on the business requirements, the current list of vehicle rental categories is as follows:

<i>Vehicle Rental Category ID</i>	<i>Vehicle Rental Category Name</i>	<i>Category Daily Rental Rate</i>
1	Car-Economic	\$113.99
2	Car-Compact	\$115.99
3	Car-Intermediate	\$116.67
4	Car-Standard	\$119.99
5	Car-Full Size	\$121.99
6	Car-Premium	\$127.79
7	Car-Luxury	\$139.99
8	SUV-Intermediate	\$127.99
9	SUV-Standard	\$128.99
10	SUV-Standard Elite	\$135.99
11	SUV-Full Size	\$148.99
12	SUV-Premium	\$157.99
13	Minivan-Standard	\$152.99
14	Van-Passenger Van (12 passengers)	\$161.00
15	Van-Cargo Van	\$19.95
16	Pick Up-Mid Size	\$69.95
17	Pick Up-Full Size	\$105.99
18	Motorcycle-Touring	\$19.95
19	Motorcycle-Cruiser	\$199.99
20	Motorcycle-Scooter	\$79.95

Application Physical Technical Architecture

After a design meeting with the architects and full-stack developers a decision was made on the application architecture for the **EZRental POS** application.

After a thorough review of both the **business requirements** and **technical requirements** by the project team, the resultant decisions on architecture(s) were based on the following:

▪ **Rental Agency Employees:**

- The system in our agencies used by the customer service representatives or front-line workers, must be able to quickly respond and execute the necessary requests such as
 - **POS Customer Management (Retail Customer & Corporate Customer) features** such as *Customer Search & Print, New Customer Registration, Customer Update, Customer Deletion, & Customer Listing functionalities*
 - **POS Vehicle Reservation, Rental & Return Management Feature** such as *Vehicle Reservations, Vehicle Rental & Vehicle Return functionalities*.
 - **POS Vehicle Inventory Management Feature** allows inventory personnel and employees to bulk-manage vehicles such as **Cars, SUVs, Mini-Vans, Cargo Vans**, and other vehicles to be *searched, added, updated, deleted, printed, listed* etc.
 - **POS Credit Card Management Feature** such as *Credit Card Search & Print, New Credit Card Registration, Credit Card Update, Credit Card Deletion, & Credit Card Listing functionalities*.
- customer reservations, rentals, returns, customer management etc., therefore fast response and performance is required to quickly service a customer and minimize the wait. This is more important in Airports and other high-traffic locations.
- We also want to provide our customer service agents with a rich user-interface experience.
- The system in the agencies is also used by other back-end personnel such as vehicle inventory managers and administrators, service personnel, vehicle transport drivers, etc. Therefore, the system needs to also perform well.

▪ **Corporate Offices:**

- The corporate offices are where our business operations are managed by our business employees & employees at the rental agencies via the INTRANET Web Portal.
- These features include:
 - **Intranet Web Enterprise Resource Planning Systems (ERP) Portal Feature** such as providing access to Enterprise Resource Planning Systems (ERP) Applications such as: *Customer Credit Card Management System, Vehicle Inventory Management System, Customer Relationship Management (CRM), Human Resource Management System, & Finance & Operations System, Marketing System, Customer & Field Service System etc.*
 - **Web EZRental Point-of-Sales Corporate Management Feature** which allows employees to *manage & execute* Point-of-Sales (POS) transaction via the **Intranet Web Portal** such as: *Search Customer Profile Information, Customer Account Management, Customer Registration, Customer Update, Customer Delete, & Customer Listing functionalities, Manage & Make Reservations of a Vehicle, Manage an existing Rental, etc.*

- The system should also perform well, but the performance requirements are not as stringent as our rental agencies which the Corporate Web Intranet meets these requirements.

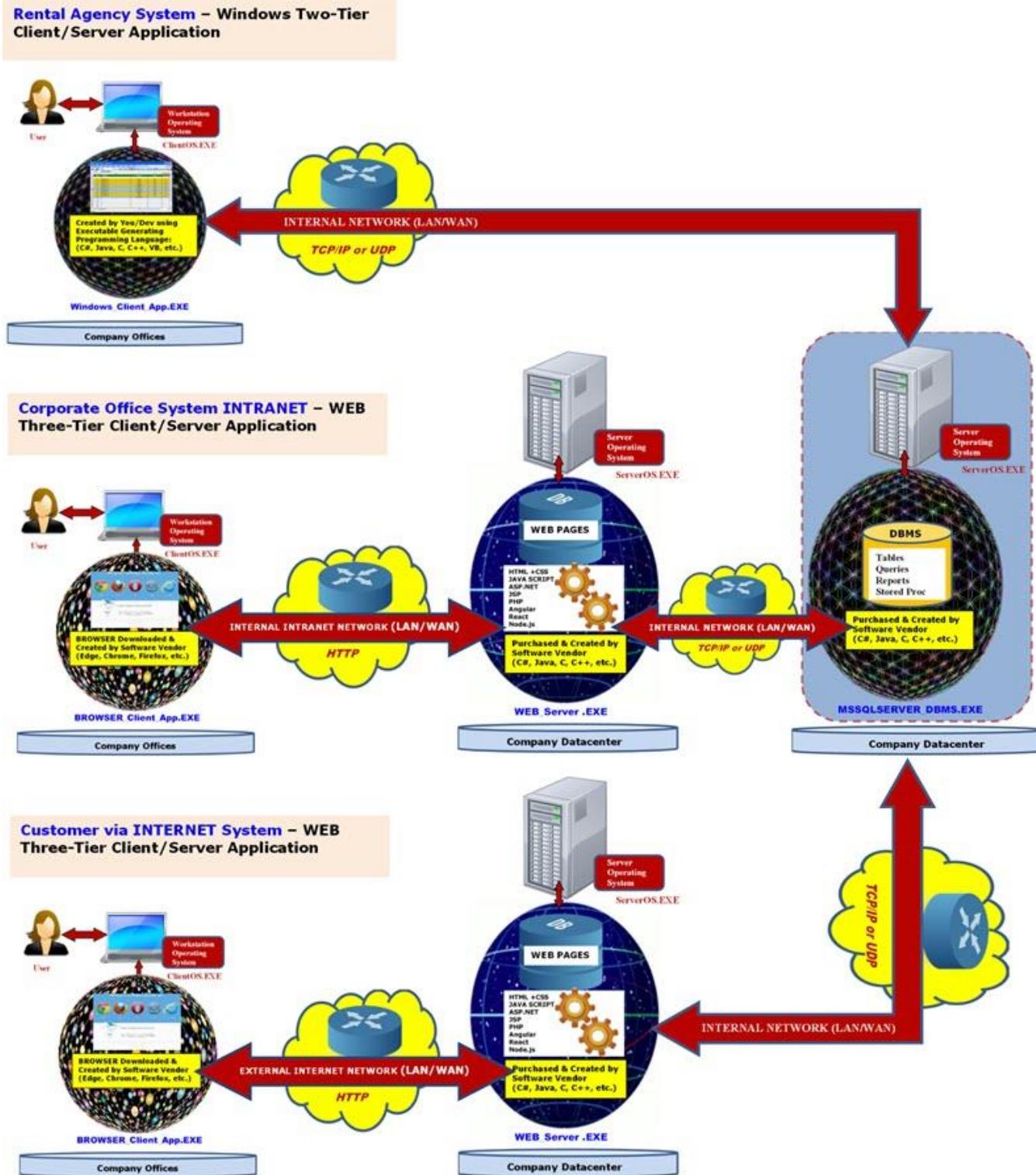
- **Customer self-service:**

- Customers who wish to make reservations and manage their reservations and rentals online via the internet, should be able to do so from anywhere in the world via our web portal.
- This also includes good user-experience.

Target applications architecture and components:

- **Rental Agency Two-Tiered Windows-Client Client/Server Application** – Front-line workers such as customer service desk in store branches, airports etc., in addition to other support personnel such as service centers employees, inventory etc., are to use this Windows-based client application for speed and performance.
- **Corporate Office Three-Tiered Web-based Client/Server** – This Web Application named EZRentalCorp.com, targeted for corporate business users in the corporate offices to manage the day-to-day business activities of our business and office workers personnel via a Browser Application.
- **Customer Internet Three-Tiered Web-based Client/Server** – This Web Application named EZRental.com, targeted for customers who will reserve vehicles online via a Browser Application.
- **Database Tier supporting all Three Applications (Rental Agency, Corporate Office & Customer Internet)** – Using **MS SQL Server for CST4708 class** and **Oracle DBMS for CST3604 class**. All the front-end applications (*Two-Tier Window for agencies, Three-tiered Web for Corporate Offices, and Three-Tier Web for Customers Internet application*) will **SHARE** the same **DATABASE TIER**. More information on the database scope will be provided in sections to follow.

Below is a pictorial diagram of this multi-component client/server architecture. Note that both the **Windows Client Application**, the **Corporate Office Browser Web Client Applications**, and the **Customer Internet Browser Web Client Application** are all sharing the same **Microsoft SQL Server DBMS Server Application**:



The list of hardware and software required for purchasing/downloading is shown in the table below.

Architecture Component	Hardware Purchase & Inventory	Software Purchase & Inventory
Rental Agency Windows Client/Server Infrastructure	<p>User Desktop/Laptop Computer:</p> <ul style="list-style-type: none"> ▪ As needed, purchase/upgrade Memory, Processor, Hard disk etc., only if PCs need to be upgraded to support the Windows Client application. <p>Network Hardware:</p> <ul style="list-style-type: none"> ▪ Required Switches, Routers & other network peripherals required to support the networking requirements of the application. <p>Office Desktop/Laptop Installation:</p> <ul style="list-style-type: none"> ▪ Assemble team to install the Windows Client Application to user's computers or use Computer Management Tool to package and deploy the applications to the user's computers. 	<p>User Desktop/Laptop Operating System:</p> <ul style="list-style-type: none"> ▪ Target OS – Windows 10, MAC Os etc. <p>Application Development & Framework:</p> <ul style="list-style-type: none"> ▪ Purchase/download required Application Development Tools such as Visual Studio, Eclipse, NetBeans etc., to develop the Windows Client Application. ▪ Download/Purchase any required framework for developing the Windows Client Application.
Corporate Office Intranet Web Client/Server Infrastructure	<p>Physical Server:</p> <ul style="list-style-type: none"> ▪ 1 Physical Server Computer with required specifications. <p>Network Hardware:</p> <ul style="list-style-type: none"> ▪ Required Switches, Routers & other network peripherals. <p>Datacenter Hardware Installation:</p> <ul style="list-style-type: none"> ▪ Required racks, cooling, electrical and other hardware to support installation of physical servers, etc. 	<p>Server Operating System:</p> <ul style="list-style-type: none"> ▪ Target OS – Windows Server, Linux, Unix, etc. <p>Web Server Application:</p> <ul style="list-style-type: none"> ▪ Purchase/download target Web Server Application such as MSFT IIS, Apache, other. <p>Web Development Framework:</p> <ul style="list-style-type: none"> ▪ Purchase/download required Web Development Tools ▪ Purchase/download required web development framework such as React, Angular, ASP.NET etc.

<p>Customer Facing Internet Web Client/Server Infrastructure</p>	<p>Physical Server:</p> <ul style="list-style-type: none"> ▪ 1 Physical Server Computer with required specifications. <p>Network Hardware:</p> <ul style="list-style-type: none"> ▪ Required Switches, Routers & other network peripherals. <p>Datacenter Hardware Installation:</p> <ul style="list-style-type: none"> ▪ Required racks, cooling, electrical and other hardware to support installation of physical servers, etc. 	<p>Server Operating System:</p> <ul style="list-style-type: none"> ▪ Target OS – Windows Server, Linux, Unix, etc. <p>Web Server Application:</p> <ul style="list-style-type: none"> ▪ Purchase/download target Web Server Application such as MSFT IIS, Apache, other. <p>Web Development Framework:</p> <ul style="list-style-type: none"> ▪ Purchase/download required Web Development Tools ▪ Purchase/download required web development framework such as React, Angular, ASP.NET etc.
---	--	--

<p>Shared Database Management System Infrastructure for Agency Two-Tier Client Server, Office, and Customer Web Portals</p>	<p>Physical Server:</p> <ul style="list-style-type: none"> ▪ 1 Physical Server Computer with required specifications. <p>Network Hardware:</p> <ul style="list-style-type: none"> ▪ Required Switches, Routers & other network peripherals. <p>Datacenter Hardware Installation:</p> <ul style="list-style-type: none"> ▪ Required racks, cooling, electrical and other hardware to support installation of physical servers, etc. 	<p>Server Operating System:</p> <ul style="list-style-type: none"> ▪ Target OS – Windows Server, Linux, Unix, etc. <p>Database Management System Application:</p> <ul style="list-style-type: none"> ▪ Purchase/download target DBMS Server Application, which in this case is either Microsoft SQL Server or Oracle 18c Express Edition. ▪ Purchase/download the DBMS Dev & Admin tools such as MS SQL Server Management Studio for Microsoft DBMS, or Oracle SQL Developer for Oracle DBMS etc.
--	--	---

Application Development Features and Functionalities (Agile Backlog)

During analysis and meetings, it was decided that the application is to deliver the following **10 features & functionality**.

Feature #	Feature Description
FEATURE #1A	FEATURE #1A – EZRental Rental Agency Point-of-Sales (POS) System CUSTOMER SERVICE – CUSTOMER MANAGEMENT SYSTEM: <ul style="list-style-type: none">▪ WINDOWS CLIENT POINT-OF SALES SYSTEM – <i>WINDOWS UI FORM(S) FRONT-END APPLICATION & OOP PROGRAMMING</i> features used by customer service representative employees via the <i>Point-of-Sales computer</i> machine in the <i>Rental Agencies</i> to service customer's CUSTOMER MANAGEMENT requests or transactions.▪ The following are features and functionality that are required for this application feature:<ul style="list-style-type: none">○ POS Customer Management Feature: <i>POS Customer Management (Retail Customer & Corporate Customer) features</i> such as <i>Customer Search & Print, New Customer Registration, Customer Update, Customer Deletion, & Customer Listing functionalities</i>.○ Note that each transaction is saved to database immediately after execution:<ul style="list-style-type: none">- Feature UI Form Requirements: <i>Design & programming</i> of required <i>User-Interface Forms & GUI Controls</i> to support this feature.- Feature Processing Requirements: <i>Design & programming</i> of required <i>Object-Oriented (OOP) Processing & Logic</i> to support this feature.▪ This feature is designed only to be used by customer service agents and other employees using the Windows Two-Tiered Client/Server Application in the Rental Agencies.
FEATURE #1B	FEATURE #1B – EZRental Rental Agency Point-of-Sales (POS) Customer Management System Back-end Database Design & Implementation to support this feature: <ul style="list-style-type: none">▪ DATABASE SERVER BACK-END SYSTEM – <i>BACK-END DATABASE DESIGN & FEATURES</i> (Create the Tables & Relationships, pre-populated tables, stored procedures, views, indexes etc.,) to support this feature.

Feature #	Feature Description
FEATURE #2A	<p>FEATURE #2A – EZRental Rental Agency Point-of-Sales (POS) System</p> <p>CUSTOMER SERVICE VEHICLE RESERVATION, RENTAL & RETURN FEATURE MANAGEMENT:</p> <ul style="list-style-type: none"> ▪ WINDOWS CLIENT POINT-OF SALES SYSTEM – WINDOWS UI FORM(S) FRONT-END APPLICATION & OOP PROGRAMMING features used by customer service representative employees via the <i>Point-of-Sales computer</i> machine in the <i>Rental Agencies</i> to service customer's CUSTOMER VEHICLE RESERVATION, RENTAL & RETURN MANAGEMENT, or transactions. ▪ The following are features and functionality are required for this application feature: <ul style="list-style-type: none"> ○ POS Vehicle Reservation, Rental & Return Management Feature: POS Customer Vehicle Reservation, Rental & Return Management (Retail Customer & Corporate Customer) features such as <i>Vehicle Reservations, Vehicle Rental & Vehicle Return functionalities</i>: - Feature UI Form Requirements: <i>Design & programming</i> of required User-Interface Forms & GUI Controls to support this feature. - Feature Processing Requirements: <i>Design & programming</i> of required Object-Oriented (OOP) Processing & Logic to support this feature. ▪ This feature is designed only to be used by customer service agents and other employees using the Windows Two-Tiered Client/Server Application in the Rental Agencies.
FEATURE #2B	<p>FEATURE #2B – EZRental Rental Agency Point-of-Sales (POS) Customer Vehicle Reservation, Rental & Return Management System Back-end Database Design & Implementation to support this feature:</p> <ul style="list-style-type: none"> ▪ DATABASE SERVER BACK-END SYSTEM – BACK-END DATABASE DESIGN & FEATURES (Create the Tables & Relationships, pre-populated tables, stored procedures, views, indexes etc.,) to support this feature.

Feature #	Feature Description
FEATURE #3A	<p>FEATURE #3A – EZRental Internal Back-Office Agency BACK-OFFICE VEHICLE INVENTORY MANAGEMENT SYSTEM (NOT A CUSTOMER FACING APPLICATION):</p> <ul style="list-style-type: none"> ▪ WINDOWS CLIENT POINT-OF SALES SYSTEM – WINDOWS UI FORM(S) FRONT-END APPLICATION & OOP PROGRAMMING features used by Back-Office Inventory Team employees via a computer machine in the <i>Rental Agencies</i> to service inventory needs for VEHICLE INVENTORY MANAGEMENT, or transactions. ▪ This is a unique Back-end system meant for inventory team employees to perform bulk IN-MEMORY inventory processing or management tasks on vehicles such as: <i>adding</i> vehicles to the system, <i>searching</i> for vehicles, <i>updating</i> vehicles, <i>deleting</i> vehicles, etc. The idea is that the employee can perform all these features on their computer in-memory repeatedly for several vehicles in one session saving to database after each transaction but managed in-memory using a collection or other data structure to manage it locally. When user is done with all inventory transactions, all transactions have been saved to database but, is still locally in the collection or other data structure and can be updated as needed. By keeping it locally in memory, the operations are faster. ▪ The following are features and functionality are required for this application feature: <ul style="list-style-type: none"> ○ POS Vehicle Inventory Management Feature: POS Vehicle Inventory Management features allows inventory personnel and employees to bulk-manage vehicles such as Cars, SUVs, Mini-Vans, Cargo Vans, and other vehicles to be <i>searched, added, updated, deleted, printed, listed</i> etc. <ul style="list-style-type: none"> - Feature UI Form Requirements: <i>Design & programming</i> of required User-Interface Forms & GUI Controls to support this feature. - Feature Processing Requirements: <i>Design & programming</i> of required Object-Oriented (OOP) Processing & Logic to support this feature. ▪ This back-office features is not designed to be used by customers and not available via the Web and implemented using the Windows Two-Tiered Client/Server Application in the Rental Agencies.
FEATURE #3B	<p>FEATURE #3B – EZRental Rental Agency Vehicle Inventory Management System Back-end Database Design & Implementation to support this feature:</p> <ul style="list-style-type: none"> ▪ DATABASE SERVER BACK-END SYSTEM – BACK-END DATABASE DESIGN & FEATURES (Create the Tables & Relationships, pre-populated tables, stored procedures, views, indexes etc.,) to support this feature.

Feature #	Feature Description
FEATURE #4A	<p>FEATURE #4A – EZRental Rental Agency Point-of-Sales (POS) BACK-OFFICE CREDIT CARD MANAGEMENT SYSTEM:</p> <ul style="list-style-type: none"> ▪ WINDOWS CLIENT POINT-OF SALES SYSTEM – WINDOWS UI FORM(S) FRONT-END APPLICATION & OOP PROGRAMMING features is a back-end system used by customer service representative & other employees via the <i>Point-of-Sales computer</i> machine in the <i>Rental Agencies</i> to service customer's CREDIT CARD MANAGEMENT, or transactions required when servicing customers. ▪ The following are features and functionality are required for this application with features such as: <ul style="list-style-type: none"> ○ POS Credit Card Management Feature: POS Customer Credit Card Management features such as <i>Credit Card Search & Print, New Credit Card Registration, Credit Card Update, Credit Card Deletion, & Credit Card Listing functionalities:</i> <ul style="list-style-type: none"> - Feature UI Form Requirements: <i>Design & programming</i> of required <i>User-Interface Forms & GUI Controls</i> to support this feature. - Feature Processing Requirements: <i>Design & programming</i> of required <i>Object-Oriented (OOP) Processing & Logic</i> to support this feature. ▪ This feature is designed only to be used by customer service agents and other employees using the Windows Two-Tiered Client/Server Application in the Rental Agencies.
FEATURE #4B	<p>FEATURE #4B – EZRental Rental Agency Point-of-Sales (POS) Credit Card Management System Back-end Database Design & Implementation to support this feature:</p> <ul style="list-style-type: none"> ▪ DATABASE SERVER BACK-END SYSTEM – BACK-END DATABASE DESIGN & FEATURES (Create the Tables & Relationships, pre-populated tables, stored procedures, views, indexes etc.,) to support this feature.

Feature #	Feature Description
FEATURE #5A	<p>FEATURE #5A – EZRental Rental Agency Point-of-Sales (POS) System BACK-OFFICE EMPLOYEE & CUSTOMER USER-ACCOUNT MANAGEMENT SYSTEM:</p> <ul style="list-style-type: none"> ▪ WINDOWS CLIENT POINT-OF SALES SYSTEM – WINDOWS UI FORM(S) BACK-OFFICE APPLICATION & OOP PROGRAMMING User-Accounts Management Features used by customer service representative & IT Administrator employees via the <i>Point-of-Sales computer</i> machine in the <i>Rental Agencies</i> to service customer's CUSTOMER & EMPLOYEE USER ACCOUNT MANAGEMENT requests or transactions. ▪ Employee User Accounts – These are the user accounts used by IT Administrators, Customer Service Employees, back-office employees, and any employee who qualifies for access to the system. ▪ Customer User Accounts – These are the user accounts used by IT Administrators, Customer Service Employees, back-office employees and any employee who has access to the system to manage the Customer User Accounts for login into the Customer Web Portal. ▪ The following are features and functionality that are required for this application feature: <ul style="list-style-type: none"> ○ POS User Account (Employee & Customer) Management Feature: POS User Account Management (Employee & Customer) features such as: <ul style="list-style-type: none"> - Employee User Account Feature 5A-1 – Allows IT Administrators, Customer Service Employees, back-office employees, and any employee who qualifies to manage employee user accounts that allow employees to login into the POS System. And perform the following tasks: Employee User Account Search by username, New Employee User Account Registration, Employee User Account Update by username, Employee User Account Deletion by username, & Employee User Account Listing functionalities. IMPORTANT! Note that the password is NEVER DISPLAYED or LISTED, only the username! - Customer User Account Feature 5A-2 – Allows IT Administrators, Customer Service Employees, back-office employees, and any employee who qualifies to manage customer user accounts that allow customers to login into the Customer Web Portal System. And perform the following tasks: Customer User Account Search by username, New Customer User Account Registration, Customer User Account Update by username, Customer User Account Deletion deletion by username, & Customer User Account Listing functionalities. IMPORTANT! Note that the password is NEVER DISPLAYED or LISTED, only the username! ○ Note that each transaction is saved to database immediately after execution: <ul style="list-style-type: none"> - Feature UI Form Requirements: <i>Design & programming</i> of required User-Interface Forms & GUI Controls to support this feature. - Feature Processing Requirements: <i>Design & programming</i> of required Object-Oriented (OOP) Processing & Logic to support this feature. ▪ This feature is designed only to be used by IT Administrations and other employees who qualify to use this system to manage both employees & customers user accounts using the Windows Two-Tiered Client/Server Application in the Rental Agencies.

FEATURE #5B	FEATURE #5B – EZRental Rental Agency Point-of-Sales (POS) User EMPLOYEE User Account Management System Back-end Database Design & Implementation to support this feature: <ul style="list-style-type: none"> ▪ DATABASE SERVER BACK-END SYSTEM – BACK-END DATABASE DESIGN & FEATURES (Create the Tables & Relationships, pre-populated tables, stored procedures, views, indexes etc.,) to support this feature to store and manage EMPLOYEE USER ACCOUNTS.
FEATURE #5C	FEATURE #5C – EZRental Rental Agency Point-of-Sales (POS) User CUSTOMER User Account Management System Back-end Database Design & Implementation to support this feature: <ul style="list-style-type: none"> ▪ DATABASE SERVER BACK-END SYSTEM – BACK-END DATABASE DESIGN & FEATURES (Create the Tables & Relationships, pre-populated tables, stored procedures, views, indexes etc.,) to support this feature to store and manage CUSTOMER USER ACCOUNTS.

Feature #	Feature Description
FEATURE #6A	<p>FEATURE #6A – EZRental Rental Agency Point-of-Sales (POS) System</p> <p>EMPLOYEES BACK-OFFICE SECURITY LOGIN AUTHENTICATION SYSTEM:</p> <ul style="list-style-type: none"> ▪ Proper <i>security and authentication</i> must be implemented to make sure only authorized employees can access the Point-Of- Sales & Back-End Management systems when they login into the Windows Two-Tiered Client/Server Application & Web Three-Tiered Corporate Client/Server Application. ▪ WINDOWS CLIENT POINT-OF SALES SYSTEM – WINDOWS UI FORM(S) BACK-OFFICE APPLICATION & OOP PROGRAMMING Login Authentication features used by customer service representative & IT Administrator employees via the <i>Point-of-Sales computer</i> machine in the <i>Rental Agencies</i> to service employee LOGIN AUTHENTICATION SYSTEM. ▪ The following are features and functionality that are required for this application such as: <ul style="list-style-type: none"> ○ POS Employee Back-Office Security Login Authentication System: POS Login Authentication Access for Employees features such as: <ul style="list-style-type: none"> - Employee Authentication Feature 6A-1 – To have access to the application, an employee (Customer Service Reps, Back-office employee etc.) must provide a username & password. This feature is required to be <i>designed & programmed</i> into the application. - Employee Authentication Feature 6A-2 – Design & programming of required User-Interface Forms & GUI Controls to support the Authentication System feature! ○ Programming includes: <ul style="list-style-type: none"> - Feature UI Form Requirements: Design & programming of required <i>User-Interface Forms & GUI Controls</i> to support this feature. - Feature Processing Requirements: Design & programming of required Object-Oriented (OOP) Processing & Logic to support this feature. ▪ This feature is designed only to be used by all employees wishing access to the Windows Two-Tiered Client/Server Application POS System in the Rental Agencies.
FEATURE #6B	<p>FEATURE #6B – EZRental Rental Agency Point-of-Sales (POS) Employee Back-Office Security Login Authentication System Back-end Database Design & Implementation to support this feature:</p> <ul style="list-style-type: none"> ▪ DATABASE SERVER BACK-END SYSTEM – BACK-END DATABASE DESIGN & FEATURES (Create the Tables & Relationships, pre-populated tables, stored procedures, views, indexes etc.,) to support this feature.

Feature #	Feature Description
FEATURE #7A	<p>FEATURE #7A – EZRental INTERNAL CORPORATE EMPLOYEE & RENTAL AGENCIES EMPLOYEES INTRANET CORPORATE BUSINESS APPLICATIONS WEB PORTAL:</p> <ul style="list-style-type: none"> ▪ This INTRANET (NOT THE PUBLIC INTERNET) Web Portal EZRentalHub.com, is a Web-based Three-Tiered Client/Server physical & Software Development Architecture used by CORPORATE & OTHER EMPLOYEES to <i>execute Enterprise Resource Planning Systems (ERP) Applications & other Corporate Applications</i> online intranet via a BROWSER. ▪ WEB UI FORM(S) FRONT-END APPLICATION & OOP PROGRAMMING Enterprise Resource Planning Systems (ERP) Applications & other Corporate Applications Features used by Corporate Employees via the CORPORATE INTRANET PORTAL. ▪ The following are features and functionality that are required for this INTRANET WEB APPLICATION: <ul style="list-style-type: none"> ○ Corporate Business Application Intranet Web Portal Features: <ul style="list-style-type: none"> - Intranet Web Enterprise Resource Planning Systems (ERP) Portal Feature 7A-1 – Provides access to Enterprise Resource Planning Systems (ERP) Applications such as: <i>Customer Credit Card Management System, Vehicle Inventory Management System, Customer Relationship Management (CRM), Human Resource Management System, & Finance & Operations System, Marketing System, Customer & Field Service System etc.</i> - Web EZRental Point-of-Sales Corporate Management Feature 7A-2 – Allows Employees to <i>manage & execute</i> Point-of-Sales (POS) transaction via the Intranet Web Portal such as: <i>Search Customer Profile Information, Customer Account Management, Customer Registration, Customer Update, Customer Delete, & Customer Listing functionalities, Manage & Make Reservations of a Vehicle, Manage an existing Rental, etc.</i> ○ Programming includes: <ul style="list-style-type: none"> - Feature UI Form Requirements: <i>Design & programming</i> of required INTRANET WEB User-Interface Forms & GUI Controls to support this feature. - Feature Processing Requirements: <i>Design & programming</i> of required INTRANET WEB TECHNOLOGY, Object-Oriented (OOP) Processing & Logic PROCESSING to support this feature.

FEATURE #7B

FEATURE #7B – EZRental Corporate Employee & Rental Agencies Employees INTRANET WEB PORTAL System Back-end Database Design & Implementation to support this feature:

- **DATABASE SERVER BACK-END SYSTEM – BACK-END DATABASE DESIGN & FEATURES** (Create the Tables & Relationships, pre-populated tables, stored procedures, views, indexes etc.,) to support this feature.

Feature #	Feature Description
FEATURE #8A	<p>FEATURE #8A – EZRental EXTERNAL CUSTOMER SELF-SERVICE INTERNET CUSTOMER FACING POINT-OF-SALES (POS) WEB PORTAL:</p> <ul style="list-style-type: none"> ▪ This Web Portal EZRental.com, is a Web-based Three-Tiered Client/Server physical & Software Development Architecture used by CUSTOMERS to <i>manage</i> & <i>make reservations</i> online via a BROWSER. ▪ BROWSER/WEB CUSTOMER POINT-OF SALES SYSTEM – WEB UI FORM(S) FRONT-END APPLICATION & OOP PROGRAMMING Point-of-Sales (POS) Management Features used by customers via the INTERNET Point-of-Sales PORTAL via their computers/laptop/tables/Mobile to MAKE RESERVATIONS ONLINE & MANAGE THEIR RENTAL. ▪ The following are features and functionality that are required for this WEB APPLICATION feature: <ul style="list-style-type: none"> ○ POS Reservation & Management Features: <ul style="list-style-type: none"> - Web POS Authentication System Feature 8A-1 – Proper security and authentication must be implemented to make sure only the authorized customer can access to its Point-Of-Sales portal and login and out of their profile website. - Web POS Customer Self-Service Management Feature 8A-2 – Allows POS Customer (Retail Customer & Corporate Customer) Self-Service Management of their account such as: <i>Customer Profile Information, Customer Account & Login Registration, Customer Update Profile, Customer Delete Profile, & Customer Listing functionalities such as listing of Reservations & Rental History etc.</i> - Web POS Customer Self-Service Point-of-Sales Management Feature 8A-3 – Allows POS Customer (Retail Customer & Corporate Customer) Self-Service features to make reservations and manage rentals such as: <i>Make Reservations of a Vehicle, Manage an existing Rental, etc.</i> - Web POS User Account Management Feature 8A-4 – Allows POS Customer (Retail Customer & Corporate Customer) Self-Service features to enable customer to manage its User Account and perform the following operations: <i>Reset Username & Reset Password.</i> ○ Programming includes: <ul style="list-style-type: none"> - Feature UI Form Requirements: <i>Design & programming</i> of required WEB User-Interface Forms & GUI Controls to support this feature. - Feature Processing Requirements: <i>Design & programming</i> of required WEB TECHNOLOGY, Object-Oriented (OOP) Processing & Logic PROCESSING to support this feature. ▪ This feature is designed only to be used by CUSTOMERS via the INTERNET using the Web Three-Tiered Client/Server Application from their Personal Computer/Mobile Devices via the INTERNET.

FEATURE #8B

FEATURE #8B – EZRental Customer Self-Service Internet Customer facing Point-of-Sales (POS) WEB PORTAL System Back-end Database Design & Implementation to support this feature:

- **DATABASE SERVER BACK-END SYSTEM – BACK-END DATABASE DESIGN & FEATURES** (Create the Tables & Relationships, pre-populated tables, stored procedures, views, indexes etc.,) to support this feature.

Feature #	Feature Description
FEATURE #9A	<p>FEATURE #9A – EZRental Customer Point-of-Sales (POS) System CUSTOMER FACING WEB PORTAL SECURITY LOGIN AUTHENTICATION SYSTEM:</p> <ul style="list-style-type: none"> ▪ Proper <i>security and authentication</i> must be implemented to make sure only authorized customers can access their Self-Service-Point-Of-Sales Web Portal systems when they login into the Web Three-Tiered Customer Client/Server Application via the INTERNET. ▪ CUSTOMER SELF-SERVICE POINT-OF SALES SYSTEM – WEB UI FORM(S) FRONT-END APPLICATION & OOP PROGRAMMING for Customer Login Authentication features used by customer service representative & IT Administrator employees to service CUSTOMER LOGIN AUTHENTICATION SYSTEM. ▪ The following are features and functionality that are required for this application such as: <ul style="list-style-type: none"> ○ Customer Self-Service Web Portal Security Login Authentication System. <i>Self-Service Web Portal Login Authentication Access for Customer</i> features such as: <ul style="list-style-type: none"> - Customer Authentication Feature 9A-1 – To have access to their Self-Service Web Portal Application, a customer must provide a username & password. This feature is required to be <u>designed</u> & <u>programmed</u> into the application. - Customer Authentication Feature 9A-2 – Design & programming of required Web User-Interface Forms & GUI Controls to support the Web Portal Authentication System feature! ○ Programming includes: <ul style="list-style-type: none"> - Feature Web UI Form Requirements: Design & programming of required <i>User-Interface Forms & GUI Controls</i> to support this feature. - Feature Web Processing Requirements: Design & programming of required <i>Object-Oriented (OOP) Processing & Logic</i> to support this feature. ▪ This feature is designed only to be used by customers wishing access to their Self-Service Web Portal Three-Tiered Client/Server Application via the INTERNET.
FEATURE #9B	<p>FEATURE #9B – EZRental Customer Point-of-Sales (POS) System CUSTOMER FACING WEB PORTAL SECURITY LOGIN AUTHENTICATION SYSTEM Back-end Database Design & Implementation to support this feature:</p> <ul style="list-style-type: none"> ▪ DATABASE SERVER BACK-END SYSTEM – BACK-END DATABASE DESIGN & FEATURES (Create the Tables & Relationships, pre-populated tables, stored procedures, views, indexes etc.,) to support the CUSTOMER facing authentication features.

Feature #	Feature Description
FEATURE #10A	<p>FEATURE #10A – EZRental INTERNAL CORPORATE EMPLOYEE & RENTAL AGENCIES EMPLOYEES INTRANET BACK-OFFICE VEHICLE TRANSPORT MANAGEMENT SYSTEM WEB PORTAL:</p> <ul style="list-style-type: none"> ▪ This INTRANET Web Portal EZRentalHub.com, is a Web-based Three-Tiered Client/Server physical & Software Development Architecture used by CORPORATE & AGENCY EMPLOYEES to <i>manage Transportation of Vehicles by Employee Drivers to and from Rental Agencies, Vehicle Distribution Centers, and other Locations</i> via a BROWSER. ▪ WEB TRANSPORT MANAGEMENT SYSTEM APPLICATION – WEB UI FORM(S) FRONT-END APPLICATION & OOP PROGRAMMING Transport Management Features used by Vehicle Transportation Managers & Drivers Employees to handle the day-to-day vehicle transportation process via the CORPORATE INTRANET PORTAL. ▪ The following are features and functionality that are required for this INTRANET Transport Management WEB APPLICATION: <ul style="list-style-type: none"> ○ Corporate Vehicle Transport Application Intranet Web Portal Features: <ul style="list-style-type: none"> - Transport Scheduling Feature – handle the day-to-day creating & scheduling of a pic-up & delivery (Any vehicle type) such as: <i>Creation of NEW Vehicle Transport Request, Vehicle Pick-up, Vehicle Drop-off & Vehicle Transport Status etc.</i> ○ Programming includes: <ul style="list-style-type: none"> - Feature UI Form Requirements: <i>Design & programming</i> of required INTRANET WEB User-Interface Forms & GUI Controls to support this feature. - Feature Processing Requirements: <i>Design & programming</i> of required INTRANET WEB TECHNOLOGY, Object-Oriented (OOP) Processing & Logic PROCESSING to support this feature. ▪ This feature is designed only to be used by CORPORATE EMPLOYEES in the Corporate Offices & RENTAL AGENCIES EMPLOYEES via the INTRANET using the Web Three-Tiered Client/Server Application.

FEATURE #10B	<p>FEATURE #10B – EZRental Corporate Vehicle Transport Application Intranet Web Portal Features System Back-end Database Design & Implementation to support this feature:</p> <ul style="list-style-type: none"> ▪ DATABASE SERVER BACK-END SYSTEM – BACK-END DATABASE DESIGN & FEATURES (Create the Tables & Relationships, pre-populated tables, stored procedures, views, indexes etc.,) to support this feature.
-------------------------	---

During analysis and meetings between the customer stakeholders and **NYC Tech Solutions** Architects, a decision was made that before deploying the entire application with all **10 features & functionality** a **DEPLOYMENT PILOT** to a **LIMITED SUBSET OF USERS** or PILOT.

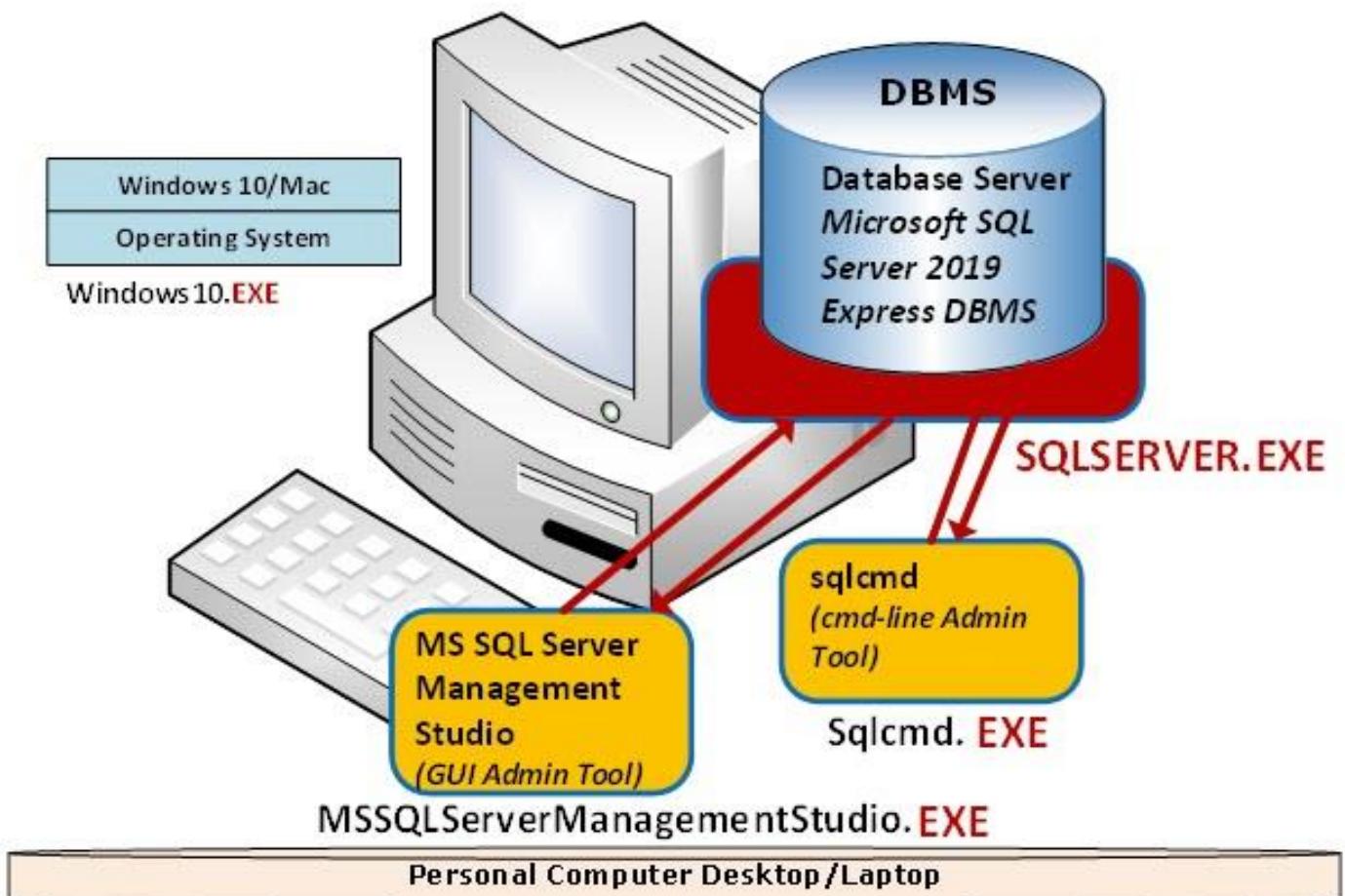
- A decision was made for the **DEPLOYMENT PILOT** to include **ONLY** the following features:
 - 1. Back-end Database Design and Implementation** – for the **EZRental Rental Agency Point-of-Sales (POS) Customer Management System** feature.
 - 2. Back-end Database Design and Implementation** – for the **EZRental Rental Agency Point-of-Sales (POS) Credit Card Management System** feature.
 - 3. Back-end Database Design and Implementation** – for the **EZRental Rental Agency Point-of-Sales (POS) User CUSTOMER User Account Management System** feature.
 - 4. Back-end Database Design and Implementation** – for the **EZRental Customer Point-of-Sales (POS) System CUSTOMER FACING WEB PORTAL SECURITY LOGIN AUTHENTICATION SYSTEM** feature.
 - 5. Front-end Client Application Design and Implementation** – for the **EZRental Rental Agency Point-of-Sales (POS) Customer Management System** feature.
 - 6. Front-end Client Application Design and Implementation** – for the **EZRental Rental Agency Point-of-Sales (POS) Credit Card Management System** feature.
 - 7. Front-end Client Application Design and Implementation** – for the **EZRental Rental Agency Point-of-Sales (POS) User CUSTOMER User Account Management System** feature.
 - 8. Front-end Web Application Design and Implementation** – for the **EZRental Customer Point-of-Sales (POS) System CUSTOMER FACING WEB PORTAL SECURITY LOGIN AUTHENTICATION SYSTEM** feature.

Database Management System Development Environment & Physical Architecture

Database Management System & Development Environment Physical Architecture Overview.

Database Tier – the Database Management System (DBMS) in scope is **MS SQL Server Community Edition** since this is the standard DBMS used at **EZRental Inc.**

Standalone Development Environment



Project Roles & Responsibilities

The **Business/Database Analyst** aligned the required database development team, and the table below describes each of the roles and the individual(s) that will execute the roles:

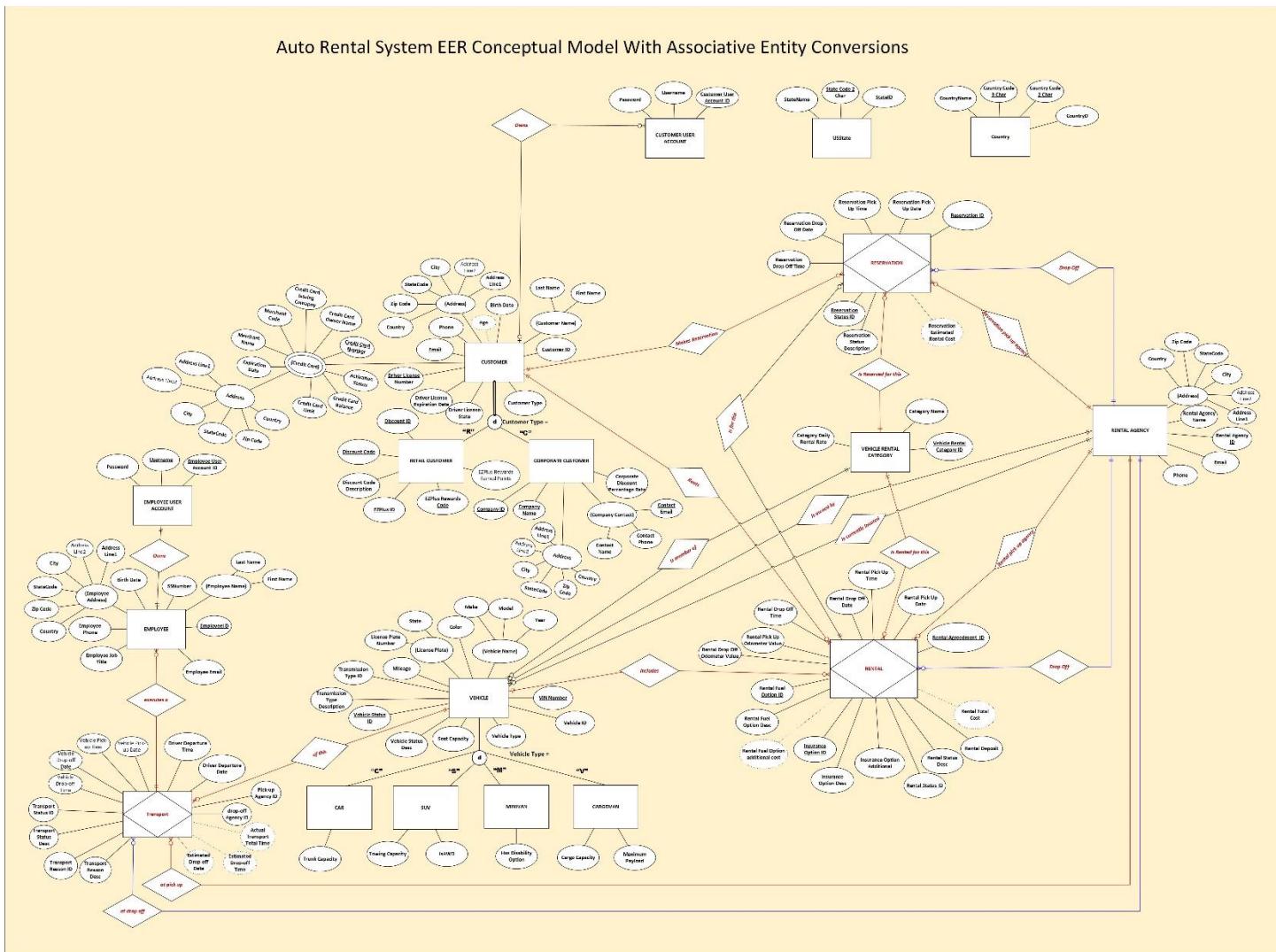
Person	Role	Description
Prof. Rodriguez	Program Manager, AgileScrum Master & ProjectManager	<ul style="list-style-type: none"> ▪ Owner of the project and liaison to Manage the EZRental Inc., the customer. ▪ Activities include but not limited to: <ol style="list-style-type: none"> 1. Owner of project responsible for the success of the project. 2. Project Management 3. Scrum Master ensures the project stays on time and moving in the right direction. Clear any obstacles impeding the team's progress etc.
Consultant #1: Prof. Rodriguez	Business & DatabaseAnalyst	<ul style="list-style-type: none"> ▪ A Business/Database Analyst was hired by Prof. Rodriguez to interview the stakeholders at EZRental Inc. And create the Business Requirements that will be the foundation to the database design & implementation. ▪ Activities include but not limited to: <ol style="list-style-type: none"> 1. Engage in discovery activities & interview the stakeholders at EZRental Inc. 2. From the interview and discovery create 1) ER/EER Conceptual Data Model from the business requirements & 2) Normalized Logical Model.
Consultant #2, 3, 4 & 5 Pavel Drozdov	Database Developers	<ul style="list-style-type: none"> ▪ This role uses the Normalized Logical Model created by consultant #2 to create the Data Dictionary, Physical Schema Diagram, and Implement the Database Application for the Auto Rental System. ▪ Activities include but not limited to: <ol style="list-style-type: none"> 1. Use the Normalized Logical Model created by consultant #2 to do the following: <ol style="list-style-type: none"> 1) Create Data Dictionary tables for eachlogical table targeting MS SQL Server 2) Create Physical Schema Diagram. 2. From these two deliverables, <ol style="list-style-type: none"> 1) implement the Database Application using Oracle 18c for the Auto Rental System.
Consultant #6 Pavel Drozdov	Database Administrator	<ul style="list-style-type: none"> ▪ The DB Admin, install the DBMS, maintain, and operate the DBMS throughout its lifetime. ▪ Activities include but not limited to: <ol style="list-style-type: none"> 1. As DB Admin, you are to <ol style="list-style-type: none"> 1) Setup & install MS SQL Server 2) Administrative tools for target DBMS. 2. Also, as DB Admin, you are to 3) Operate & Maintain the DBMS.

The **Application Full Stack OOP Architect/Analyst** aligned the required application development team and the table below describes each of the roles and the individual (s) that will execute the roles:

Person	Role	Description
Consultant #7 & 13 Mr. Rodriguez	Full Stack Object-Oriented-Programming Architect	<ul style="list-style-type: none"> ▪ An Object-Oriented-Programming Architect was hired by Prof. Rodriguez to interview the stakeholders at EZRental Inc. and derive the Application Technical Requirements in addition to designing the Class/Object Model Architecture. This also includes the planning and designing both Windows Client Application and the Web Browser Application. ▪ Activities include but not limited to: <ol style="list-style-type: none"> 1. Engage in discovery activities & interview the stakeholders at EZRental Inc. 2. From the interview and discovery 1) Design/Architect the Object-Oriented-Programming Class/Object Model for the Windows Client Application. 3. Design/Architect the Object-Oriented-Programming Class/Object Model for the Web Browser Application.
Consultants #8, 9, 10, 11 & 12 Pavel Drozdov	Full Stack Windows Application Developers & UI/UX Client Application Developer	<ul style="list-style-type: none"> ▪ Object-Oriented-Programming developer to implement the Windows Client Application using C# & .NET technologies & on the database side, implement stored procedures and support the databased team as needed. ▪ Activities include but not limited to: <ol style="list-style-type: none"> 1. As full stack developer, Programming & implementation of the Object-Oriented-Programming of Class/Object Model designed by consultant #7 for the Windows Client Application using C# & .NET Technologies. 2. In addition, Development of Database Stored Procedures, and other development requirements in the Back-end DBMS. 3. From the technical requirements, design a high-level Graphical User-Interface (GUID) wireframe, & implement the front-end UI Programming, features & functionality
Consultant #14, 15, 16, 17 & 18 Pavel Drozdov	Full Stack Web Application Developer & UI/UX Web Application Developer	<ul style="list-style-type: none"> ▪ Object-Oriented-Programming developer to implement the Web Browser Application using C# & ASP.NET technologies. ▪ Activities include but not limited to: <ol style="list-style-type: none"> 1. As full stack developer, Programming & implementation of the Object-Oriented-Programming of Class/Object Model designed by consultant #7 for the Web Browser Client Application using C# & ASP.NET Technologies. 2. From the technical requirements, design a high-level Graphical User-Interface (GUID) wireframe, & implement the Webfront-end UI Programming, features & functionality in the Web Server Application

Database Design Deliverable #1 – ER/EER Conceptual Model Diagram

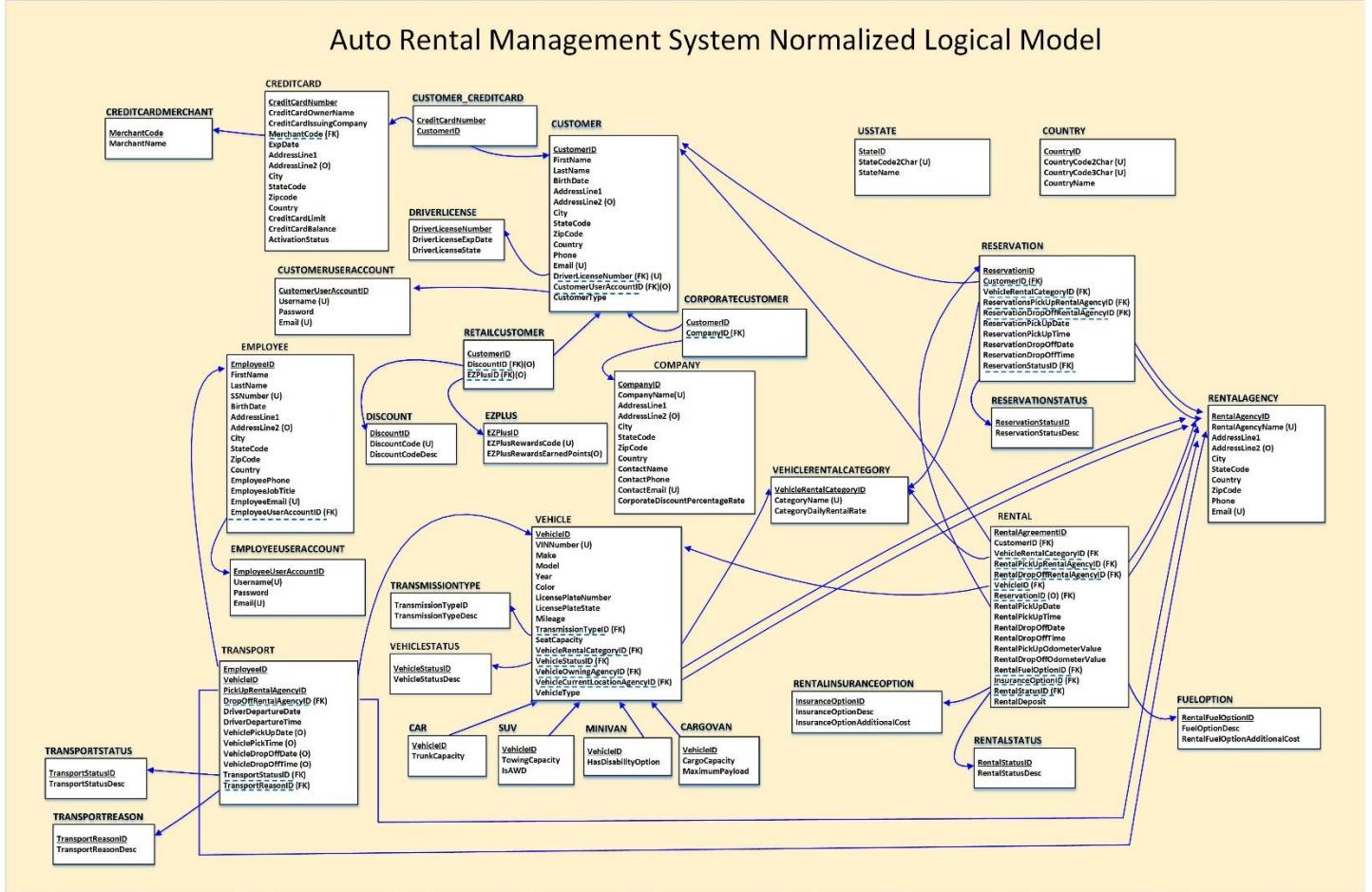
This **EER Conceptual Model Diagram**, which showed below, is the foundation of the **Database Design** for the **DBMS Auto Rental Management System Application**. It visualizes **relationships** between **entities** that are identified in the **Application Business Requirements**. By doing so, it forms a high-level picture or **diagram** of how the Application Key Business Data relate to each other.



Database Design Deliverable #2 – Normalized Logical Model Diagram

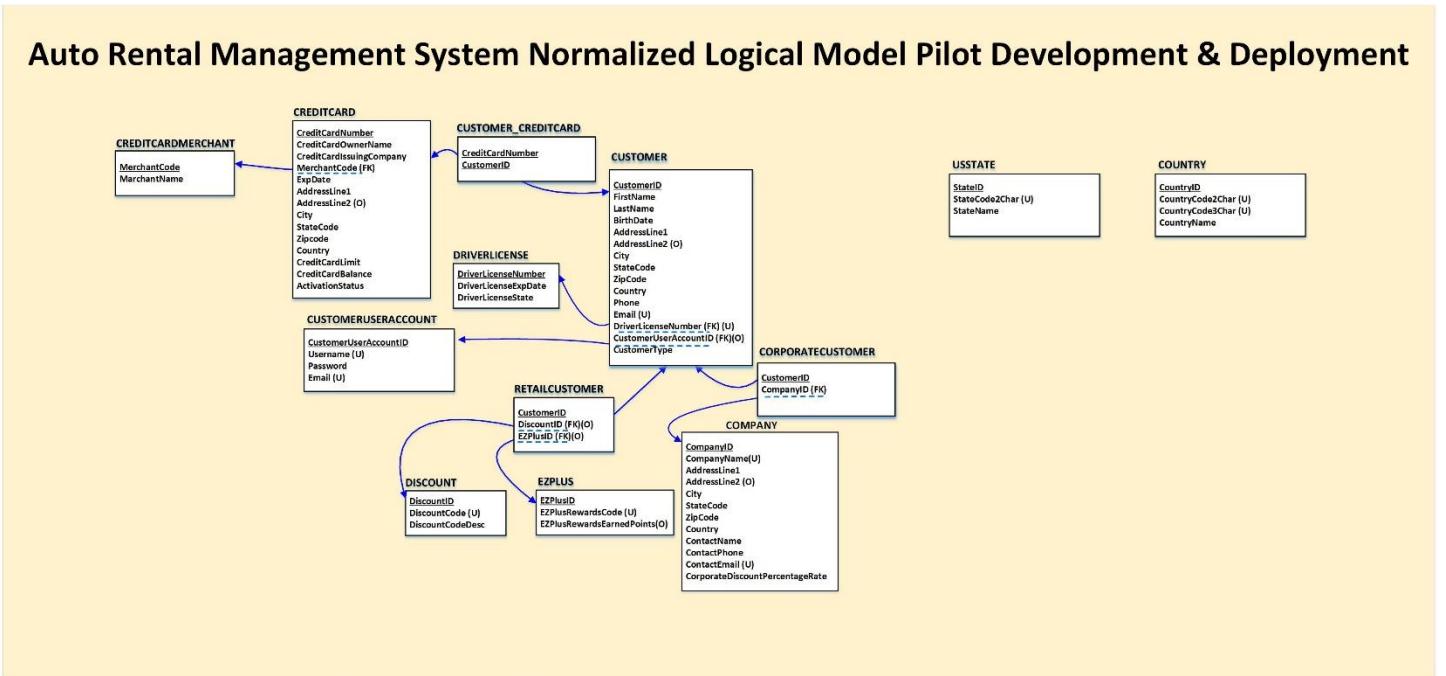
The main goal of **Normalized Logical Model Diagram** is to take the **EER Conceptual Model Diagram** and transform into actual **database TABLES** and their **relationships**. In other words, it shows **relationships** of **TABLES** that will be implemented into **DBMS**.

The **Normalized Logical Model Diagram**, which is shown below, illustrates **TABLES** with their content and the relationship between them.



The **Pilot** version of the **Normalized Logical Model** will be deployed to a few rental agencies only to test and prove the design with real users.

Pilot version is a true small production deployment for a small number of Pilot Users.



Database Design Deliverable #3 – Physical Model Data Dictionary

Tables below represent the Physical Model Data Dictionary, which contains the listing of all attributes, data types, and metadata, which shows the information inserted into database of the EZRental Auto Rental Management System.

CREDITCARD							
Column Num.	Attribute/Column Name	Generic Data Type Name	MS SQL SERVER Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/purpose
1.	<u>CreditCardNumber</u>	String	VARCHAR(16)	Yes	16	PRIMARY KEY	Primary key that has a business meaning. Credit card table identifier
2.	<u>CreditCardOwnerName</u>	String	VARCHAR(50)	Yes	50	NOT NULL	Credit card's owner name
3.	<u>CreditCardIssuingCompany</u>	String	VARCHAR(30)	Yes	30	NOT NULL	Credit card's issuing company
4.	<u>MerchantCode</u>	Number	TINYINT	Yes	DEFAULT	FOREIGN KEY NOT NULL CHECK(between 1 and 20)	Merchant code that identifies the merchant's name
5.	<u>ExpDate</u>	Date	DATE	Yes	MM/YY	NOT NULL	Expiration date
6.	<u>AddressLine1</u>	String	VARCHAR(50)	Yes	50	NOT NULL	Address of the owner of the credit card
7.	<u>AddressLine2</u>	String	VARCHAR(50)	No	50	NULL	Optional address of the owner of the credit card
8.	<u>City</u>	String	VARCHAR(30)	Yes	30	NOT NULL	Owner's city
9.	<u>StateCode</u>	Character	CHAR(2)	Yes	2	NOT NULL	Owner's state
10.	<u>Zipcode</u>	String	VARCHAR(10)	Yes	10	NOT NULL	Owner's zipcode
11.	<u>Country</u>	String	VARCHAR(50)	Yes	50	NOT NULL	Owner's country
12.	<u>CreditCardLimit</u>	Number	DECIMAL(p,s)	Yes	p=5, s=2	NOT NULL	Credit card limit
13.	<u>CreditCardBalance</u>	Number	DECIMAL(p,s)	Yes	p=5, s=2	NOT NULL	Credit card balance
14.	<u>ActivationStatus</u>	Boolean	BIT	YES	DEFAULT	NOT NULL	Activation Status (Y/N)

CREDITCARDMERCHANT							
Column Num.	Attribute/Column Name	Generic Data Type Name	MS SQL SERVER Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/purpose
1.	<u>MerchantCode</u>	Number	TINYINT	Yes	DEFAULT	PRIMARY KEY CHECK(between 1 and 20)	Merchant code that identifies the merchant's name
2.	<u>MerchantName</u>	String	VARCHAR(50)	Yes	50	NOT NULL	Name that describes merchant.

CUSTOMER_CREDITCARD							
Column Num.	Attribute/Column Name	Generic Data Type Name	MS SQL SERVER Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/purpose
1.	<u>CreditCardNumber</u>	String	VARCHAR(16)	Yes	16	PRIMARY KEY	Primary key that has a business meaning. Credit card table identifier
2.	<u>CustomerID</u>	Number	INT	Yes	DEFAULT	PRIMARY KEY	Customer ID has no business meaning. The ID number will be generated automatically starting from 1.

DRIVERLICENSE							
Column Num.	Attribute/Column Name	Generic Data Type Name	MS SQL SERVER Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/purpose
1.	<u>DriverLicenseNumber</u>	String	VARCHAR(25)	Yes	25	PRIMARY KEY	Driver license's number
2.	<u>DriverLicenseExpDate</u>	Date	DATE	Yes	MM/DD/YY	NOT NULL	Driver license's expiration date
3.	<u>DriverLicenseState</u>	Character	CHAR(2)	Yes	2	NOT NULL	Driver license's state

CUSTOMER

Column Num.	Attribute/Column Name	Generic Data Type Name	MS SQL SERVER Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/purpose
1.	<u>CustomerID</u>	Number	IDENTITY	Yes	DEFAULT	PRIMARY KEY	Customer ID has no business meaning. The ID number will be generated automatically starting from 1.
2.	FirstName	String	VARCHAR(50)	Yes	50	NOT NULL	Customer's First name
3.	LastName	String	VARCHAR(50)	Yes	50	NOT NULL	Customer's Last Name
4.	BirthDate	Date	DATE	Yes	MM/DD/YY	NOT NULL	Customer's Date of birth
5.	AddressLine1	String	VARCHAR(50)	Yes	50	NOT NULL	Customer's address
6.	AddressLine2	String	VARCHAR(50)	No	50	NULL	Customer's optional address
7.	City	String	VARCHAR(30)	Yes	30	NOT NULL	City Name
8.	StateCode	Character	CAHR(2)	Yes	2	NOT NULL	Customer's state
9.	ZipCode	String	VARCHAR(10)	Yes	10	NOT NULL	Customer's zipcode
10.	Country	String	VARCHAR(50)	Yes	50	NOT NULL	Country name
11.	Phone	String	VARCHAR(20)	Yes	20	NOT NULL	Phone number
12.	Email	String	VARCHAR(100)	Yes	100	NOT NULL UNIQUE	Unique email address
13.	DriverLicenseNumber	String	VARCHAR(25)	Yes	25	NOT NULL UNIQUE FOREIGN KEY	Driver license's number
14.	CustomerUserAccountID	GUID	UNIQUEIDENTIFIER	No	DEFAULT	FOREIGN KEY NULL	Optional customer user account ID
15.	CustomerType	Character	CHAR(1)	Yes	1	NOT NULL	Customer Type (C/R)

COMPANY							
Column Num.	Attribute/Column Name	Generic Data Type Name	MS SQL SERVER Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/ purpose
1.	<u>CompanyID</u>	Number	INT	Yes	DEFAULT	PRIMARY KEY CHECK(between 1 and 50,000)	Company ID to identify a corporate company
2.	CompanyName	String	VARCHAR(50)	Yes	50	NOT NULL UNIQUE	Company's name
3.	AddressLine1	String	VARCHAR(50)	Yes	50	NOT NULL	Company's address
4.	AddressLine2	String	VARCHAR(50)	No	50	NULL	Optional company's address
5.	City	String	VARCHAR(30)	Yes	30	NOT NULL	City where company is located
6.	StateCode	Character	CHAR(2)	Yes	2	NOT NULL	State where company is located
7.	ZipCode	String	VARCHAR(10)	Yes	10	NOT NULL	Company's zipcode
8.	Country	String	VARCHAR(50)	Yes	50	NOT NULL	Country where company is located
9.	CompanyRepName	String	VARCHAR(100)	Yes	100	NOT NULL	Company's representative first and last name
10.	ContactPhone	String	VARCHAR(20)	Yes	20	NOT NULL	Contact phone
11.	ContactEmail	String	VARCHAR(100)	Yes	100	NOT NULL UNIQUE	Contact email
12.	CorporateDiscountPercentageRate	Number	DECIMAL	Yes	p=2 s=2	NOT NULL	Company's discount percentage rate

CORPORATECUSTOMER							
Column Num	Attribute/Column Name	Generic Data Type Name	MS SQL SERVER Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/ purpose
1.	<u>CustomerID</u>	Number	INT	Yes	DEFAULT	PRIMARY KEY	Cuistomer's ID
2.	CompanyID	Number	INT	Yes	DEFAULT	FOREIGN KEY NOT NULL	Company's ID

CUSTOMERUSERACCOUNT

Column Num.	Attribute/Column Name	Generic Data Type Name	MS SQL SERVER Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/ purpose
1.	<u>CustomerUserAccountID</u>	GUID	UNIQUEIDENTIFIER	Yes	DEFAULT NEWID()	PRIMARY KEY	Customer's account ID
2.	Username	String	VARCHAR(50)	Yes	50	NOT NULL UNIQUE	Customer's user name
3.	Password	String	VARCHAR(20)	Yes	20	NOT NULL	Customer's password
4.	Email	String	VARCHAR(100)	Yes	100	NOT NULL UNIQUE	Customer's email

RETAILCUSTOMER

Column Num	Attribute/Column Name	Generic Data Type Name	MS SQL SERVER Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/ purpose
1.	<u>CustomerID</u>	Number	INT	Yes	DEFAULT	PRIMARY KEY	Customer's ID
2.	DiscountID	Number	INT	No	DEFAULT	FOREIGN KEY NULL	Discount ID
3.	EZPlusID	Number	INT	Yes	DEFAULT	FOREIGN KEY NULL	

DISCOUNT

Column Num	Attribute/Column Name	Generic Data Type Name	MS SQL SERVER Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/ purpose
1.	<u>DiscountID</u>	Number	IDENTITY	Yes	DEFAULT	PRIMARY KEY	Discount ID
2.	DiscountCode	Character	CHAR(10)	Yes	10	NOT NULL UNIQUE	Discount code for discount
3.	DiscountCodeDesc	String	VARCHAR(150)	Yes	150	NOT NULL	Description for discount

EZPLUS

Column Num	Attribute/Column Name	Generic Data Type Name	MS SQL SERVER Data Type Name	Is it Required?	Length/Siz e /Format	Constraints	Description/ purpose
1.	<u>EZPlusID</u>	Number	IDENTITY	Yes	DEFAULT	PRIMARY KEY	EZPlus ID
2.	EZPlusRewardsCode	String	VARCHAR(13)	Yes	13	NOT NULL UNIQUE	Reward code
3.	EZPlusRewardsEarnedPoi nts	Number	INT	No	DEFAULT	NULL UNIQUIE CHECK(<=999999)	Points earned for using EZPlus

USSTATE

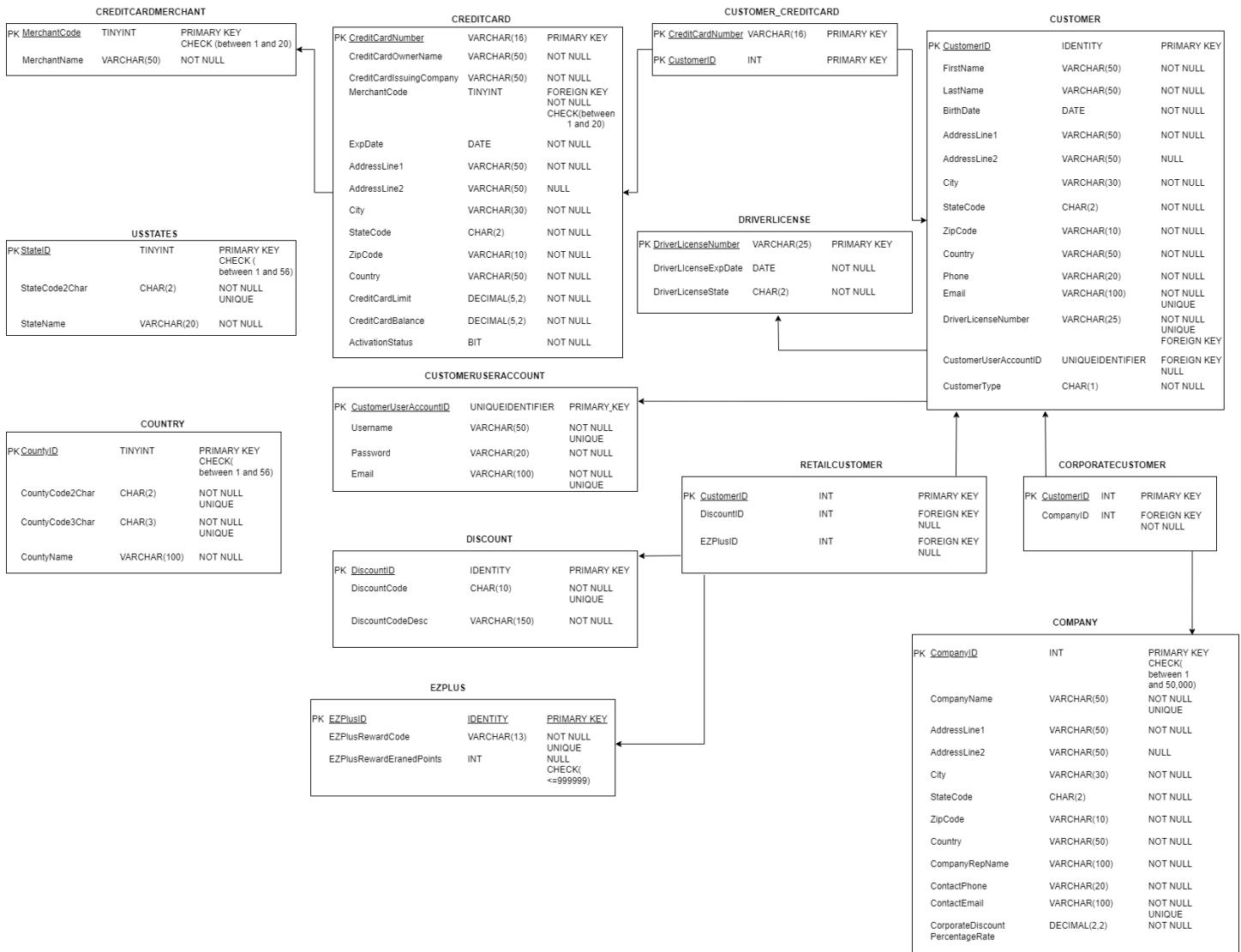
Column Num.	Attribute/Column Name	Generic Data Type Name	MS SQL SERVER Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/purpose
1.	<u>StateID</u>	Number	TINYINT	Yes	DEFAULT	PRIMARY KEY CHECK (between 1 and 56)	State ID
2.	StateCode2Char	Character	CHAR(2)	Yes	2	NOT NULL UNIQUE	2 character that represent state
3.	StateName	String	VARCHAR(20)	Yes	20	NOT NULL	Name of state

COUNTRY

Column Num.	Attribute/Column Name	Generic Data Type Name	MS SQL SERVER Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/purpose
1.	<u>CountryID</u>	Number	TINYINT	Yes	DEFAULT	PRIMARY KEY CHECK (between 1 and 200)	Country ID
2.	CountryCode2Char	Character	CHAR(2)	Yes	2	NOT NULL UNIQUE	2 letters representing name of the country
3.	CountryCode3Char	Character	CHAR(3)	Yes	3	NOT NULL UNIQUE	3 letters representing name of the country
4.	CountryName	String	VARCHAR(100)	Yes	100	NOT NULL	Name of the country

Database Design Deliverable #4 – Physical Model Scheme Design Diagram

Physical Data Model Diagram is a combination of the **normalized logical diagram** and **data dictionary table**. It is the design deliverable the database developer is to use to implement that DBMS Application. Basically, it is the final design component before implementation.



Database Implementation Deliverable #5 – Development & Implementation

The script below is creating a database in MS SQL Server called **EZRentalDB** with 13 tables from the **Pilot Logical Model**.

```
--creating database
CREATE DATABASE EZRentalDB;

--selecting database
USE EZRentalDB;

--creating DRIVERLICENSE table
CREATE TABLE DRIVERLICENSE(
    DriverLicenseNumber      VARCHAR(25)      PRIMARY KEY,
    DriverLicenseExpDate     DATE            NOT NULL,
    DriverLicenseState       CHAR(2)          NOT NULL
);

--creating CUSTOMERACCOUNT table
CREATE TABLE CUSTOMERACCOUNT(
    CustomerUserAccountId   UNIQUEIDENTIFIER PRIMARY KEY,
    Username                 VARCHAR(50)        UNIQUE NOT NULL,
    Password                 VARCHAR(20)        NOT NULL,
    Email                    VARCHAR(100)       UNIQUE NOT NULL
);
```

```
--creating CUSTOMER table
CREATE TABLE CUSTOMER(
    CustomerID          INT IDENTITY      PRIMARY KEY,
    FirstName           VARCHAR(50)       NOT NULL,
    LastName            VARCHAR(50)       NOT NULL,
    BirthDate           DATE             NOT NULL,
    AddressLine1        VARCHAR(50)       NOT NULL,
    AddressLine2        VARCHAR(50)       NULL,
    City                VARCHAR(30)       NOT NULL,
    StateCode           CHAR(2)          NOT NULL,
    ZipCode              VARCHAR(10)       NOT NULL,
    Country             VARCHAR(50)       NOT NULL,
    Phone               VARCHAR(20)       NOT NULL,
    Email               VARCHAR(100)      UNIQUE NOT NULL,
    DriverLicenseNumber VARCHAR(25)      UNIQUE NOT NULL,
    CustomerUserAccountID UNIQUEIDENTIFIER NULL,
    CustomerType        CHAR(1)          NOT NULL,
    CONSTRAINT fk_DriverLicenseNumber FOREIGN KEY (DriverLicenseNumber) REFERENCES
    DRIVERLICENSE(DriverLicenseNumber) ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT fk_CustomerUserAccountID FOREIGN KEY (CustomerUserAccountID) REFERENCES
    CUSTOMERACCOUNT(CustomerUserAccountID) ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
--creating CREDITCARDMERCHANT table
CREATE TABLE CREDITCARDMERCHANT(
    MerchantCode        TINYINT          PRIMARY KEY CHECK(MerchantCode>=1 AND MerchantCode<=20),
    MerchantName        VARCHAR(50)      NOT NULL
);
```

```

--creating CREDITCARD table
CREATE TABLE CREDITCARD(
    CreditCardNumber           VARCHAR(16)      PRIMARY KEY,
    CreditCardOwnerName        VARCHAR(50)      NOT NULL,
    CreditCardIssuingCompany   VARCHAR(50)      NOT NULL,
    MerchantCode               TINYINT         NOT NULL CHECK(MerchantCode>=1 AND
MerchantCode<=20),
    ExpDate                   DATE            NOT NULL,
    AddressLine1               VARCHAR(50)      NOT NULL,
    AddressLine2               VARCHAR(50)      NULL,
    City                       VARCHAR(30)      NOT NULL,
    StateCode                  CHAR(2)          NOT NULL,
    ZipCode                    VARCHAR(10)      NOT NULL,
    Country                    VARCHAR(50)      NOT NULL,
    CreditCardLimit            DECIMAL(5,2)     NOT NULL,
    CreditCardBalance          DECIMAL(5,2)     NOT NULL,
    ActivationStatus           BIT             NOT NULL,
    CONSTRAINT fk_MerchantCode FOREIGN KEY (MerchantCode) REFERENCES
CREDITCARDMERCHANT(MerchantCode) ON DELETE CASCADE ON UPDATE CASCADE
);

--creating CUSTOMER_CREDITCARD table
CREATE TABLE CUSTOMER_CREDITCARD(
    CreditCardNumber           VARCHAR(16)      NOT NULL,
    CustomerID                 INT              NOT NULL,
    CONSTRAINT pk_CreditCardNumber_CustomerID PRIMARY KEY (CreditCardNumber,CustomerID),
    CONSTRAINT fk_CreditCardNumber FOREIGN KEY (CreditCardNumber) REFERENCES
CREDITCARD(CreditCardNumber) ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT fk_CustomerID FOREIGN KEY (CustomerID) REFERENCES CUSTOMER(CustomerID) ON
DELETE CASCADE ON UPDATE CASCADE
);

--creating EZPLUS table
CREATE TABLE EZPLUS(
    EZPlusID                   INT IDENTITY    PRIMARY KEY,
    EZPLusRewardCode            VARCHAR(13)      UNIQUE NOT NULL,
    EZPlusRewardEarnedPoints   VARCHAR(150)     NOT NULL
);

--creating DISCOUNT table
CREATE TABLE DISCOUNT(
    DiscountID                 INT IDENTITY    PRIMARY KEY,
    DiscountCode                CHAR(10)         UNIQUE NOT NULL,
    DiscountCodeDesc            VARCHAR(150)     NOT NULL
);

```

```

--creating RETAILCUSTOMER table
CREATE TABLE RETAILCUSTOMER(
    CustomerID      INT      NOT NULL,
    DiscountID      INT      NULL,
    EZPlusID        INT      NULL,
    CONSTRAINT pk_Retail_CustomerID PRIMARY KEY (CustomerID),
    CONSTRAINT fk_Retail_CustomerID FOREIGN KEY (CustomerID) REFERENCES CUSTOMER(CustomerID)
ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT fk_DiscountID FOREIGN KEY (DiscountID) REFERENCES DISCOUNT(DiscountID) ON
DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT fk_EZPlusID FOREIGN KEY (EZPlusID) REFERENCES EZPLUS(EZPlusID) ON DELETE
CASCADE ON UPDATE CASCADE
);

--creating COMPANY table
CREATE TABLE COMPANY(
    CompanyID          INT          PRIMARY KEY CHECK(CompanyID>=1 AND
CompanyID<=50000),
    CompanyName        VARCHAR(50)   UNIQUE NOT NULL,
    AddressLine1       VARCHAR(50)   NOT NULL,
    AddressLine2       VARCHAR(50)   NULL,
    City               VARCHAR(30)   NOT NULL,
    StateCode          CHAR(2)      NOT NULL,
    ZipCode            VARCHAR(10)   NOT NULL,
    Country            VARCHAR(50)   NOT NULL,
    CompanyRepName    VARCHAR(100)  NOT NULL,
    ContactPhone      VARCHAR(20)   NOT NULL,
    ContactEmail       VARCHAR(100)   UNIQUE NOT NULL,
    CorporateDiscountPercentageRate DECIMAL(2,2)  NOT NULL
);

--creating CORPORATECUSTOMER table
CREATE TABLE CORPORATECUSTOMER(
    CustomerID      INT      NOT NULL,
    CompanyID       INT      NOT NULL CHECK(CompanyID>=1 AND CompanyID<=50000),
    CONSTRAINT pk_Corporate_CustomerID PRIMARY KEY (CustomerID),
    CONSTRAINT fk_Corporate_CustomerID FOREIGN KEY (CustomerID) REFERENCES
CUSTOMER(CustomerID) ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT fk_CompanyID FOREIGN KEY (CompanyID) REFERENCES COMPANY(CompanyID) ON DELETE
CASCADE ON UPDATE CASCADE
);

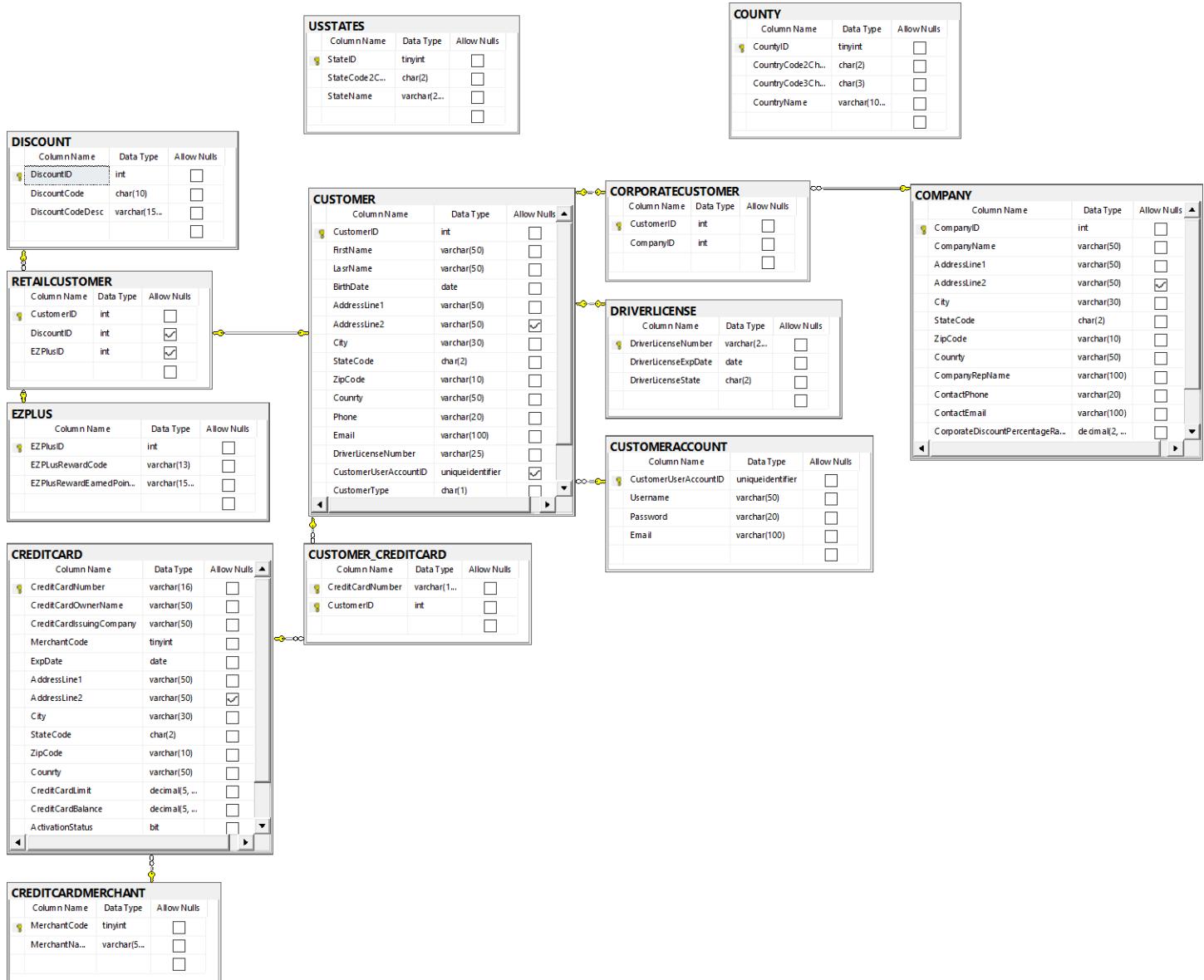
```

```
--creating USSTATES table
CREATE TABLE USSTATES(
    StateID      TINYINT          PRIMARY KEY CHECK(StateID>=1 AND StateID<=56),
    StateCode2Char CHAR(2)        UNIQUE NOT NULL,
    StateName     VARCHAR(20)      NOT NULL
);

--creating COUNTRY table
CREATE TABLE COUNTY(
    CountyID      TINYINT          PRIMARY KEY CHECK(CountyID>=1 AND CountyID<=200),
    CountryCode2Char CHAR(2)        UNIQUE NOT NULL,
    CountryCode3Char CHAR(3)        UNIQUE NOT NULL,
    CountryName    VARCHAR(100)      NOT NULL
);
```

Database Implementation Deliverable #6 – Implemented Physical Scheme Diagram

This is a generated scheme diagram by the MS SQL Server Manager of the 13 tables, that was shown above in the Pilot Logical Model.



Database Implementation Deliverable #7 – Database Validation Testing

Insert Statements

Insert statements for DRIVERLICENSE table.

Insert statements contains the following information for the **DRIVERLICENSE** table:

1. Driver License Number
2. Driver License Expiration Date
3. Driver License State

Below is an empty table before any insert statements were executed.

```
Select * from DRIVERLICENSE;
```

The screenshot shows a SQL Server Management Studio (SSMS) interface with a results grid. The grid has three columns: 'DriverLicenseNumber', 'DriverLicenseExpDate', and 'DriverLicenseState'. There are no rows of data in the grid. The status bar at the bottom indicates 'Query executed successfully.' and '0 rows'.

DriverLicenseNumber	DriverLicenseExpDate	DriverLicenseState
Click to select all grid cells		

We created the INSERT QUERY for this table and the updated table now looks like this:

```
Insert into DRIVERLICENSE(DriverLicenseNumber, DriverLicenseExpDate, DriverLicenseState)
values ('1111111111', '01/15/2022', 'NY');
Insert into DRIVERLICENSE(DriverLicenseNumber, DriverLicenseExpDate, DriverLicenseState)
values ('2222222222', '02/15/2022', 'NY');
Insert into DRIVERLICENSE(DriverLicenseNumber, DriverLicenseExpDate, DriverLicenseState)
values ('3333333333', '03/15/2022', 'NJ');
Insert into DRIVERLICENSE(DriverLicenseNumber, DriverLicenseExpDate, DriverLicenseState)
values ('4444444444', '04/15/2022', 'NJ');
Insert into DRIVERLICENSE(DriverLicenseNumber, DriverLicenseExpDate, DriverLicenseState)
values ('5555555555', '05/15/2022', 'AL');
```

The screenshot shows a SQL Server Management Studio (SSMS) interface with a results grid. The grid has three columns: DriverLicenseNumber, DriverLicenseExpDate, and DriverLicenseState. The data is as follows:

	DriverLicenseNumber	DriverLicenseExpDate	DriverLicenseState
1	1111111111	2022-01-15	NY
2	2222222222	2022-02-15	NY
3	3333333333	2022-03-15	NJ
4	4444444444	2022-04-15	NJ
5	5555555555	2022-05-15	AL

Insert statements for CUSTOMERUSERACCOUNT table.

Insert statements contains the following information for the **CUSTOMERUSERACCOUNT** table:

1. Username
2. Password
3. Email

Note: Customer User Account ID will be generated automatically.

Below is an empty table before any insert statements were executed.

```
Select * from CUSTOMERUSERACCOUNT;
```

CustomerUserAccountID	Username	Password	Email

We created the INSERT QUERY for this table and the updated table now looks like this:

```
Insert into CUSTOMERUSERACCOUNT(Username, Password, Email)
values ('Cust111', 'Pas111', 'cust111@mail.com');
Insert into CUSTOMERUSERACCOUNT(Username, Password, Email)
values ('Cust222', 'Pas222', 'cust222@mail.com');
Insert into CUSTOMERUSERACCOUNT(Username, Password, Email)
values ('Cust333', 'Pas333', 'cust333@mail.com');
Insert into CUSTOMERUSERACCOUNT(Username, Password, Email)
values ('Cust444', 'Pas444', 'cust444@mail.com');
Insert into CUSTOMERUSERACCOUNT(Username, Password, Email)
values ('Cust555', 'Pas555', 'cust555@mail.com');
```

	CustomerUserAccountID	Username	Password	Email
1	ECB866C9-D0CA-44B9-9966-07B907C3B1AF	Cust222	Pas222	cust222@mail.com
2	65504AC121175-4F31-AE2B-0D75963EE5E	Cust555	Pas555	cust555@mail.com
3	99003FA5-9765-47CA-88B5-3007E621B052	Cust444	Pas444	cust444@mail.com
4	E0D33858-ECC-428E-9EA8-41D0F36450C0	Cust111	Pas111	cust111@mail.com
5	D209E432-042A-4339-BE7E-77DC95FC3F4	Cust333	Pas333	cust333@mail.com

Insert statements for CUSTOMER table.

Insert statements contains the following information for the **CUSTOMER** table:

1. First Name
2. Last Name
3. Birth Date
4. Address Line 1
5. Address Line 2
6. City
7. State Code
8. Zip Code
9. Country
10. Phone
11. Email
12. Driver License Number
13. Customer Account ID
14. Customer Type

Note: Customer ID will be generated automatically.

Below is an empty table before any insert statements were executed.

```
Select * from CUSTOMER;
```

The screenshot shows a SQL Server Management Studio (SSMS) interface. The title bar says "200 %". The main area is titled "Results" and contains a table structure for the CUSTOMER table. The columns listed are: CustomerID, FirstName, LastName, BirthDate, AddressLine1, AddressLine2, City, StateCode, ZipCode, County, Phone, Email, DriverLicenseNumber, CustomerUserAccountID, and CustomerType. Below the table structure, a yellow status bar at the bottom of the window displays the message "Query executed successfully." and other system information like the computer name and database name.

CustomerID	FirstName	LastName	BirthDate	AddressLine1	AddressLine2	City	StateCode	ZipCode	County	Phone	Email	DriverLicenseNumber	CustomerUserAccountID	CustomerType
Query executed successfully.														

We created the INSERT QUERY for this table and the updated table now looks like this:

```

Insert into CUSTOMER(FirstName, LastName, BirthDate, AddressLine1, AddressLine2, City, StateCode,
                     ZipCode, Country, Phone, Email, DriverLicenseNumber, CustomerUserAccountID,
                     CustomerType)
values('Johan', 'Wilks', '11/22/2022', '1111 Jay St.', '', 'Brooklyn', 'NY', '11209', 'USA',
       '(505) 111-1111', 'johanwilks@mail.com', '1111111111',
       'EC8556C9-DDC4-4689-9966-070597C8B1AF', 'R');

Insert into CUSTOMER(FirstName, LastName, BirthDate, AddressLine1, AddressLine2, City, StateCode,
                     ZipCode, Country, Phone, Email, DriverLicenseNumber, CustomerUserAccountID,
                     CustomerType)
values('Velma', 'Hulme', '11/22/2006', '5555 Jay St.', '', 'Brooklyn', 'NY', '11259', 'USA',
       '(505) 555-5555', 'Velma@mail.com', '5555555555',
       '6550AC12-1175-4FF3-AEEB-0A7596E0EE6E', 'R');

Insert into CUSTOMER(FirstName, LastName, BirthDate, AddressLine1, AddressLine2, City, StateCode,
                     ZipCode, Country, Phone, Email, DriverLicenseNumber, CustomerUserAccountID,
                     CustomerType)
values('Mikhaeel', 'Hussain', '11/22/1985', '4444 Jay St.', '', 'Manhattan', 'NY', '11249', 'USA',
       '(505) 444-4444', 'Mikhaeel@mail.com', '4444444444',
       '99869F5A-9765-47CA-8BB5-3007E6218052', 'C');

Insert into CUSTOMER(FirstName, LastName, BirthDate, AddressLine1, AddressLine2, City, StateCode,
                     ZipCode, Country, Phone, Email, DriverLicenseNumber, CustomerUserAccountID,
                     CustomerType)
values('Samual', 'Nicholls', '11/22/2000', '2222 Jay St.', '', 'Trenton', 'NJ', '11229', 'USA',
       '(505) 222-2222', 'Samual@mail.com', '2222222222',
       'ED033858-4ECC-42BE-9EA8-41D0F36450C0', 'C');

Insert into CUSTOMER(FirstName, LastName, BirthDate, AddressLine1, AddressLine2, City, StateCode,
                     ZipCode, Country, Phone, Email, DriverLicenseNumber, CustomerUserAccountID,
                     CustomerType)
values('Silas', 'Decker', '11/22/1996', '3333 Jay St.', '', 'Trenton', 'NJ', '11239', 'USA',
       '(505) 333-3333', 'Silas@mail.com', '3333333333',
       'D209E432-042A-4839-BE7E-77DCB5FCF3F4', 'C');

```

CustomerID	FirstName	LastName	BirthDate	AddressLine1	AddressLine2	City	StateCode	ZipCode	County	Phone	Email	DriverLicenseNumber	CustomerUserAccountID	CustomerType
1	Johan	Wilks	2022-11-22	1111 Jay St.		Brooklyn	NY	11209	USA	(505) 111-1111	johanwilks@mail.com	1111111111	EC8556C9-DDC4-4689-9966-070597C8B1AF	R
2	Velma	Hulme	2006-11-22	5555 Jay St.		Brooklyn	NY	11259	USA	(505) 555-5555	Velma@mail.com	5555555555	6550AC12-1175-4FF3-AEEB-0A7596E0EE6E	R
3	Mikhaeel	Hussain	1985-11-22	4444 Jay St.		Manhattan	NY	11249	USA	(505) 444-4444	Mikhaeel@mail.com	4444444444	99869F5A-9765-47CA-8BB5-3007E6218052	C
4	Samual	Nicholls	2000-11-22	2222 Jay St.		Trenton	NJ	11229	USA	(505) 222-2222	Samual@mail.com	2222222222	ED033858-4ECC-42BE-9EA8-41D0F36450C0	C
5	Silas	Decker	1996-11-22	3333 Jay St.		Trenton	NJ	11239	USA	(505) 333-3333	Silas@mail.com	3333333333	D209E432-042A-4839-BE7E-77DCB5FCF3F4	C

Query executed successfully.

Insert statements for CREDITCARDMERCHANT table.

Insert statements contains the following information for the CREDITCARDMERCHANT table:

1. Merchant Code
2. Merchant Name

Below is an empty table before any insert statements were executed.

```
Select * from CREDITCARDMERCHANT;
```

Results		Messages	
	MerchantCode	MerchantName	
Query executed successfully.			

We created the INSERT QUERY for this table and the updated table now looks like this:

```
Insert into CREDITCARDMERCHANT(MerchantCode, MerchantName)
values (1, 'Stax by Fattmerchant');
Insert into CREDITCARDMERCHANT(MerchantCode, MerchantName)
values (2, 'Helcim');
Insert into CREDITCARDMERCHANT(MerchantCode, MerchantName)
values (3, 'Dharma Merchant Services');
Insert into CREDITCARDMERCHANT(MerchantCode, MerchantName)
values (4, 'Payment Depot');
Insert into CREDITCARDMERCHANT(MerchantCode, MerchantName)
values (5, 'National Processing');
```

Results		Messages	
	MerchantCode	MerchantName	
1	1	Stax by Fattmerchant	
2	2	Helcim	
3	3	Dharma Merchant Services	
4	4	Payment Depot	
5	5	National Processing	
Query executed successfully.			

Insert statements for CREDITCARD table.

Insert statements contains the following information for the **CREDITCARD** table:

1. Credit Card Number
2. Credit Card Owner Name
3. Credit Card Issuing Company
4. Merchant Code
5. Expiration Date
6. Address Line 1
7. Address Line 2
8. City
9. State Code
10. Zip Code
11. Country
12. Credit Card Limit
13. Credit Card Balance
14. Activation Status

Below is an empty table before any insert statements were executed.

```
Select * from CREDITCARD;
```

CreditCardNumber	CreditCardOwnerName	CreditCardIssuingCompany	MerchantCode	ExpDate	AddressLine1	AddressLine2	City	StateCode	ZipCode	County	CreditCardLimit	CreditCardBalance	ActivationStatus
Query executed successfully.													

We created the INSERT QUERY for this table and the updated table now looks like this:

```

Insert into CREDITCARD(CreditCardNumber, CreditCardOwnerName, CreditCardIssuingCompany,
MerchantCode, ExpDate, AddressLine1, AddressLine2, City, StateCode, ZipCode, Country,
CreditCardLimit, CreditCardBalance, ActivationStatus)
values ('1111111111111111', 'Johan Wilks', 'American Express', 1, '01/01/2024', '1111 Jay St.',
'', 'Brooklyn', 'NY', '11209', 'USA', 10000.00, 256.21, 'True');
Insert into CREDITCARD(CreditCardNumber, CreditCardOwnerName, CreditCardIssuingCompany,
MerchantCode, ExpDate, AddressLine1, AddressLine2, City, StateCode, ZipCode, Country,
CreditCardLimit, CreditCardBalance, ActivationStatus)
values ('2222222222222222', 'Samual Nicholls', 'TD Bank', 1, '03/20/2024', '2222 Jay St.',
'', 'Trenton', 'NJ', '11229', 'USA', 8000.00, 4230.86, 'True');
Insert into CREDITCARD(CreditCardNumber, CreditCardOwnerName, CreditCardIssuingCompany,
MerchantCode, ExpDate, AddressLine1, AddressLine2, City, StateCode, ZipCode, Country,
CreditCardLimit, CreditCardBalance, ActivationStatus)
values ('3333333333333333', 'Silas Decker', 'American Express', 1, '05/22/2025', '3333 Jay St.',
'', 'Trenton', 'NJ', '11239', 'USA', 10000.00, 256.21, 'True');
Insert into CREDITCARD(CreditCardNumber, CreditCardOwnerName, CreditCardIssuingCompany,
MerchantCode, ExpDate, AddressLine1, AddressLine2, City, StateCode, ZipCode, Country,
CreditCardLimit, CreditCardBalance, ActivationStatus)
values ('4444444444444444', 'Mikhaeel Hussain', 'Chase Bank', 1, '01/01/2024', '4444 Jay St.',
'', 'Manhattan', 'NY', '11249', 'USA', 10000.00, 256.21, 'False');
Insert into CREDITCARD(CreditCardNumber, CreditCardOwnerName, CreditCardIssuingCompany,
MerchantCode, ExpDate, AddressLine1, AddressLine2, City, StateCode, ZipCode, Country,
CreditCardLimit, CreditCardBalance, ActivationStatus)
values ('5555555555555555', 'Velma Hulme', 'American Express', 1, '01/01/2024', '5555 Jay St.',
'', 'Brooklyn', 'NY', '11259', 'USA', 500.00, 256.21, 'True');
Insert into CREDITCARD(CreditCardNumber, CreditCardOwnerName, CreditCardIssuingCompany,
MerchantCode, ExpDate, AddressLine1, AddressLine2, City, StateCode, ZipCode, Country,
CreditCardLimit, CreditCardBalance, ActivationStatus)
values ('6666666666666666', 'Artur Smith', 'American Express', 3, '01/01/2024', '6666 Jay St.',
'', 'Brooklyn', 'NY', '11269', 'USA', 500.00, 0.00, 'True');

```

CreditCardNumber	CreditCardOwnerName	CreditCardIssuingCompany	MerchantCode	ExpDate	AddressLine1	AddressLine2	City	StateCode	ZipCode	Country	CreditCardLimit	CreditCardBalance	ActivationStatus
1111111111111111	Johan Wilks	American Express	1	2024-01-01	1111 Jay St.		Brooklyn	NY	11209	USA	10000.00	256.21	1
2222222222222222	Samual Nicholls	TD Bank	1	2024-03-20	2222 Jay St.		Trenton	NJ	11229	USA	8000.00	4230.86	1
3333333333333333	Silas Decker	American Express	1	2025-05-22	3333 Jay St.		Trenton	NJ	11239	USA	10000.00	256.21	1
4444444444444444	Mikhaeel Hussain	Chase Bank	1	2024-01-01	4444 Jay St.		Manhattan	NY	11249	USA	10000.00	256.21	0
5555555555555555	Velma Hulme	American Express	1	2024-01-01	5555 Jay St.		Brooklyn	NY	11259	USA	500.00	256.21	1

Insert statements for CUSTOMER_CREDITCARD table.

Insert statements contains the following information for the CUSTOMER_CREDITCARD table:

1. Credit Card Number
2. Customer ID

Below is an empty table before any insert statements were executed.

```
Select * from CUSTOMER_CREDITCARD;
```

CreditCardNumber	CustomerID
Query executed successfully.	
DESKTOP-9VFICMM\SQLEXPRESS... DESKTOP-9VFICMM\neichen... EZRentalDB 00:00:00 0 rows	

We created the INSERT QUERY for this table and the updated table now looks like this:

```
Insert into CUSTOMER_CREDITCARD(CreditCardNumber, CustomerID)
values('1111111111111111', 1);
Insert into CUSTOMER_CREDITCARD(CreditCardNumber, CustomerID)
values('2222222222222222', 4);
Insert into CUSTOMER_CREDITCARD(CreditCardNumber, CustomerID)
values('3333333333333333', 5);
Insert into CUSTOMER_CREDITCARD(CreditCardNumber, CustomerID)
values('4444444444444444', 3);
Insert into CUSTOMER_CREDITCARD(CreditCardNumber, CustomerID)
values('5555555555555555', 2);
```

CreditCardNumber	CustomerID
1111111111111111	1
2222222222222222	4
3333333333333333	5
4444444444444444	3
5555555555555555	2

Insert statements for DISCOUNT table.

Insert statements contains the following information for the **DISCOUNT** table:

1. Discount Code
2. Discount Code Description

Below is an empty table before any insert statements were executed.

```
Select * from DISCOUNT;
```

DiscountID	DiscountCode	DiscountCodeDesc
Query executed successfully.		

We created the INSERT QUERY for this table and the updated table now looks like this:

```
Insert into DISCOUNT(DiscountCode, DiscountCodeDesc)
values('AAA9970054', 'AAA Membership Discount - 25% off base
rate plus 10% donated for breast cancer research');
Insert into DISCOUNT(DiscountCode, DiscountCodeDesc)
values('GOV8756921', 'Government Employee Discount - 30% off
base rate');
Insert into DISCOUNT(DiscountCode, DiscountCodeDesc)
values('STA3415632', 'State Employee Discount for 25% off base
rate');
Insert into DISCOUNT(DiscountCode, DiscountCodeDesc)
values('VET2055179', 'Veteran Discount 35% off base rate Plus
10% donation to veteran's family fund');
Insert into DISCOUNT(DiscountCode, DiscountCodeDesc)
values('AAA9974432', 'AAA Membership Discount - 10% off base
rate plus 60% donated for breast cancer research');
```

DiscountID	DiscountCode	DiscountCodeDesc
1	2	AAA Membership Discount - 25% off base rate plu...
2	GOV8756921	Government Employee Discount - 30% off base rate
3	4	STA3415632 State Employee Discount for 25% off base rate
4	5	VET2055179 Veteran Discount 35% off base rate Plus 10% don...
5	6	AAA9974432 AAA Membership Discount - 10% off base rate plu...

Insert statements for EZPLUS table.

Insert statements contains the following information for the **EZPLUS** table:

1. EZPlus Reward Code
2. EZPlus Reward Earned Points

Below is an empty table before any insert statements were executed.

```
Select * from EZPLUS;
```

Results			
	EZPlusID	EZPlusRewardCode	EZPlusRewardEarnedPoints
1			
Query executed successfully.			

We created the INSERT QUERY for this table and the updated table now looks like this:

```
Insert into EZPLUS(EZPLusRewardCode, EZPlusRewardEarnedPoints)
values ('EZP9009854637', 10000);
Insert into EZPLUS(EZPLusRewardCode, EZPlusRewardEarnedPoints)
values ('EZP1000192461', 500);
Insert into EZPLUS(EZPLusRewardCode, EZPlusRewardEarnedPoints)
values ('EZP6493238865', 159000);
Insert into EZPLUS(EZPLusRewardCode, EZPlusRewardEarnedPoints)
values ('EZP2005135627', 23000);
Insert into EZPLUS(EZPLusRewardCode, EZPlusRewardEarnedPoints)
values ('EZP6842134756', 850000);
```

Results			
	EZPlusID	EZPlusRewardCode	EZPlusRewardEarnedPoints
1		EZP9009854637	10000
2	2	EZP1000192461	500
3	3	EZP6493238865	159000
4	4	EZP2005135627	23000
5	5	EZP6842134756	850000

Insert statements for EZPLUS table.

Insert statements contains the following information for the **EZPLUS** table:

1. EZPlus Reward Code
2. EZPlus Reward Earned Points

Below is an empty table before any insert statements were executed.

```
Select * from RETAILCUSTOMER;
```

CustomerID	DiscountID	EZPlusID
Query executed successfully.		

We created the INSERT QUERY for this table and the updated table now looks like this:

```
Insert into RETAILCUSTOMER(CustomerID, DiscountID, EZPlusID)
values (1, 2, 1);
Insert into RETAILCUSTOMER(CustomerID, DiscountID, EZPlusID)
values (2, 3, 2);
Insert into RETAILCUSTOMER(CustomerID, DiscountID, EZPlusID)
values (3, 4, 3);
Insert into RETAILCUSTOMER(CustomerID, DiscountID, EZPlusID)
values (4, 5, 4);
Insert into RETAILCUSTOMER(CustomerID, DiscountID, EZPlusID)
values (5, 6, 5);
```

CustomerID	DiscountID	EZPlusID
1	2	1
2	3	2
3	4	3
4	5	4
5	6	5

Insert statements for COMPANY table.

Insert statements contains the following information for the **COMPANY** table:

1. Company ID
2. Company Name
3. Address Line 1
4. Address Line 2
5. City
6. State Code
7. Zip Code
8. Country
9. Company Representative Name
10. Contact Phone
11. Contact Email
12. Corporate Discount Percentage Rate

Below is an empty table before any insert statements were executed.

```
Select * from COMPANY;
```

CompanyID	CompanyName	AddressLine1	AddressLine2	City	StateCode	ZipCode	Country	CompanyRepName	ContactPhone	ContactEmail	CorporateDiscountPercentageRate
Query executed successfully.											

We created the INSERT QUERY for this table and the updated table now looks like this:

```
Insert into COMPANY(CompanyID, CompanyName, AddressLine1, AddressLine2,
City, StateCode, ZipCode, Country, CompanyRepName, ContactPhone,
ContactEmail, CorporateDiscountPercentageRate)
values(1, 'Company 1', '1111 Jay st.', '', 'Brooklyn', 'NY', '11209',
'USA', 'Jack Reacher', '(718) 111-1111', 'JackReacher@mail.com', 23.56);
Insert into COMPANY(CompanyID, CompanyName, AddressLine1, AddressLine2,
City, StateCode, ZipCode, Country, CompanyRepName, ContactPhone,
ContactEmail, CorporateDiscountPercentageRate)
values(2, 'Company 2', '2222 Jay st.', '', 'Brooklyn', 'NY', '11219',
'USA', 'Malachi Proctor', '(718) 222-2222', 'MalachiProctor@mail.com', 2.50);
Insert into COMPANY(CompanyID, CompanyName, AddressLine1, AddressLine2,
City, StateCode, ZipCode, Country, CompanyRepName, ContactPhone,
ContactEmail, CorporateDiscountPercentageRate)
values(3, 'Company 3', '3333 Jay st.', '', 'Brooklyn', 'NY', '11229',
'USA', 'Gilbert Dennis', '(718) 333-3333', 'GilbertDennis@mail.com', 42.13);
Insert into COMPANY(CompanyID, CompanyName, AddressLine1, AddressLine2,
City, StateCode, ZipCode, Country, CompanyRepName, ContactPhone,
ContactEmail, CorporateDiscountPercentageRate)
values(4, 'Company 4', '4444 Jay st.', '', 'Brooklyn', 'NY', '11239',
'USA', 'Jude Hudson', '(718) 444-4444', 'JudeHudson@mail.com', 20.35);
Insert into COMPANY(CompanyID, CompanyName, AddressLine1, AddressLine2,
City, StateCode, ZipCode, Country, CompanyRepName, ContactPhone,
ContactEmail, CorporateDiscountPercentageRate)
values(5, 'Company 5', '5555 Jay st.', '', 'Brooklyn', 'NY', '11249',
'USA', 'Frazer Downs', '(718) 555-5555', 'FrazerDowns@mail.com', 68.00);
Insert into COMPANY(CompanyID, CompanyName, AddressLine1, AddressLine2,
City, StateCode, ZipCode, Country, CompanyRepName, ContactPhone,
ContactEmail, CorporateDiscountPercentageRate)
values(6, 'Company 6', '6666 Jay st.', '', 'Brooklyn', 'NY', '11259',
'USA', 'Frazer Downs', '(718) 666-6666', 'FrazerDowns6@mail.com', 68.00);
```

The screenshot shows a SQL Server Management Studio (SSMS) results grid. The grid has a header row with columns: CompanyID, CompanyName, AddressLine1, AddressLine2, City, StateCode, ZipCode, Country, CompanyRepName, ContactPhone, ContactEmail, and CorporateDiscountPercentageRate. Below the header, there are five data rows, each corresponding to one of the six companies inserted into the table. The data is as follows:

CompanyID	CompanyName	AddressLine1	AddressLine2	City	StateCode	ZipCode	Country	CompanyRepName	ContactPhone	ContactEmail	CorporateDiscountPercentageRate
1	Company 1	1111 Jay st.		Brooklyn	NY	11209	USA	Jack Reacher	(718) 111-1111	JackReacher@mail.com	23.56
2	Company 2	2222 Jay st.		Brooklyn	NY	11219	USA	Malachi Proctor	(718) 222-2222	MalachiProctor@mail.com	2.50
3	Company 3	3333 Jay st.		Brooklyn	NY	11229	USA	Gilbert Dennis	(718) 333-3333	GilbertDennis@mail.com	42.13
4	Company 4	4444 Jay st.		Brooklyn	NY	11239	USA	Jude Hudson	(718) 444-4444	JudeHudson@mail.com	20.35
5	Company 5	5555 Jay st.		Brooklyn	NY	11249	USA	Frazer Downs	(718) 555-5555	FrazerDowns@mail.com	68.00

Insert statements for CORPORATECUSTOMER table.

Insert statements contains the following information for the **CORPORATECUSTOMER** table:

1. Customer ID
2. Company ID

Below is an empty table before any insert statements were executed.

```
Select * from CORPORATECUSTOMER;
```

Results		Messages
CustomerID	CompanyID	
 Query executed successfully.		
DESKTOP-9VFJCM\SQLEXPRESS ... DESKTOP-9VFJCM\irech... EZRentalDB 00:00:00 0 rows		

We created the INSERT QUERY for this table and the updated table now looks like this:

```
Insert into CORPORATECUSTOMER (CustomerID, CompanyID)
values (16, 1);
Insert into CORPORATECUSTOMER (CustomerID, CompanyID)
values (17, 2);
Insert into CORPORATECUSTOMER (CustomerID, CompanyID)
values (18, 3);
Insert into CORPORATECUSTOMER (CustomerID, CompanyID)
values (19, 4);
Insert into CORPORATECUSTOMER (CustomerID, CompanyID)
values (20, 5);
```

Results		Messages
CustomerID	CompanyID	
1	16	1
2	17	2
3	18	3
4	19	4
5	20	5
 Query executed successfully.		
DESKTOP-9VFJCM\SQLEXPRESS ... DESKTOP-9VFJCM\irech... EZRentalDB 00:00:00 5 rows		

Select statements

Select statement #1

Select statement that queries **CUSTOMER** table returning **only 1 record** that includes all columns based on the primary key.

```
Select * from CUSTOMER where CustomerID = 1;
```

The result of the query that has been executed, is all information that was previously populated and associated with the given primary key (in this case with CustomerID).

A screenshot of the SQL Server Management Studio (SSMS) interface. The title bar says 'DESKTOP-PVFCMM SQL EXPRESS ... DESKTOP-PVFCMM\rech... EZRentalDB 00:00:00 1 rows'. The main window shows a results grid with one row of data. The columns are: CustomerID, FirstName, LastName, BirthDate, AddressLine1, AddressLine2, City, StateCode, ZipCode, County, Phone, Email, DriverLicenseNumber, CustomerUserAccountID, and CustomerType. The data for the single row is: CustomerID 1, FirstName Johan, LastName Wilks, BirthDate 2022-11-22, AddressLine1 1111 Jay St., AddressLine2 Brooklyn, City NY, StateCode NY, ZipCode 11209, County USA, Phone (505) 111-1111, Email johanwilks@mail.com, DriverLicenseNumber 11111111, CustomerUserAccountID EC0556C9-0DC4-4689-9966-070597C881AF, and CustomerType R. Below the grid, a status bar says 'Query executed successfully.'

CustomerID	FirstName	LastName	BirthDate	AddressLine1	AddressLine2	City	StateCode	ZipCode	County	Phone	Email	DriverLicenseNumber	CustomerUserAccountID	CustomerType
1	Johan	Wilks	2022-11-22	1111 Jay St.		Brooklyn	NY	11209	USA	(505) 111-1111	johanwilks@mail.com	11111111	EC0556C9-0DC4-4689-9966-070597C881AF	R

Select statement #2

Select statement that queries **CUSTOMER** table returning **all record** that includes all columns based on the criteria.

```
Select * from CUSTOMER where StateCode = 'NY';
```

The result of the query that has been executed, is all rows and columns that was previously populated and associated with the given criteria in the **Customer** table (in this case all records that have **NY** in the State Code Column).

A screenshot of the SQL Server Management Studio (SSMS) interface. The title bar says 'DESKTOP-PVFCMM SQL EXPRESS ... DESKTOP-PVFCMM\rech... EZRentalDB 00:00:00 6 rows'. The main window shows a results grid with six rows of data. The columns are: CustomerID, FirstName, LastName, BirthDate, AddressLine1, AddressLine2, City, StateCode, ZipCode, County, Phone, Email, DriverLicenseNumber, CustomerUserAccountID, and CustomerType. The data for the six rows is: CustomerID 1, FirstName Johan, LastName Wilks, BirthDate 2022-11-22, AddressLine1 1111 Jay St., AddressLine2 Brooklyn, City NY, StateCode NY, ZipCode 11209, County USA, Phone (505) 111-1111, Email johanwilks@mail.com, DriverLicenseNumber 11111111, CustomerUserAccountID EC0556C9-0DC4-4689-9966-070597C881AF, and CustomerType R; CustomerID 2, FirstName Veena, LastName Hume, BirthDate 2006-11-22, AddressLine1 5555 Jay St., AddressLine2 Brooklyn, City NY, StateCode NY, ZipCode 11259, County USA, Phone (505) 555-6655, Email veena.hume@mail.com, DriverLicenseNumber 55555555, CustomerUserAccountID 6550AC12-1175-4F73-AE8B-0A7996E0E6E6, and CustomerType R; CustomerID 3, FirstName Michael, LastName Hussain, BirthDate 1985-11-22, AddressLine1 4444 Jay St., AddressLine2 Manhattan, City NY, StateCode NY, ZipCode 11249, County USA, Phone (505) 444-4444, Email mihkael@mail.com, DriverLicenseNumber 44444444, CustomerUserAccountID 9989975A-9763-47CA-AB85-3007E6210052, and CustomerType R; CustomerID 4, FirstName Johan, LastName Wilks, BirthDate 2022-11-22, AddressLine1 1111 Jay St., AddressLine2 Brooklyn, City NY, StateCode NY, ZipCode 11209, County USA, Phone (505) 111-1111, Email johanwilks@mail.com, DriverLicenseNumber 66666666, CustomerUserAccountID EC0556C9-0DC4-4689-9966-070597C881AF, and CustomerType C; CustomerID 5, FirstName Veena, LastName Hume, BirthDate 2006-11-22, AddressLine1 5555 Jay St., AddressLine2 Brooklyn, City NY, StateCode NY, ZipCode 11259, County USA, Phone (505) 555-5555, Email veena7@mail.com, DriverLicenseNumber 77777777, CustomerUserAccountID 6550AC12-1175-4F73-AE8B-0A7996E0E6E6, and CustomerType C; CustomerID 6, FirstName Michael, LastName Hussain, BirthDate 1985-11-22, AddressLine1 4444 Jay St., AddressLine2 Manhattan, City NY, StateCode NY, ZipCode 11249, County USA, Phone (505) 444-4444, Email mihkael@mail.com, DriverLicenseNumber 88888888, CustomerUserAccountID 9989975A-9763-47CA-AB85-3007E6210052, and CustomerType C. Below the grid, a status bar says 'Query executed successfully.'

CustomerID	FirstName	LastName	BirthDate	AddressLine1	AddressLine2	City	StateCode	ZipCode	County	Phone	Email	DriverLicenseNumber	CustomerUserAccountID	CustomerType
1	Johan	Wilks	2022-11-22	1111 Jay St.		Brooklyn	NY	11209	USA	(505) 111-1111	johanwilks@mail.com	11111111	EC0556C9-0DC4-4689-9966-070597C881AF	R
2	Veena	Hume	2006-11-22	5555 Jay St.		Brooklyn	NY	11259	USA	(505) 555-6655	veena.hume@mail.com	55555555	6550AC12-1175-4F73-AE8B-0A7996E0E6E6	R
3	Michael	Hussain	1985-11-22	4444 Jay St.		Manhattan	NY	11249	USA	(505) 444-4444	mihkael@mail.com	44444444	9989975A-9763-47CA-AB85-3007E6210052	R
4	Johan	Wilks	2022-11-22	1111 Jay St.		Brooklyn	NY	11209	USA	(505) 111-1111	johanwilks@mail.com	66666666	EC0556C9-0DC4-4689-9966-070597C881AF	C
5	Veena	Hume	2006-11-22	5555 Jay St.		Brooklyn	NY	11259	USA	(505) 555-5555	veena7@mail.com	77777777	6550AC12-1175-4F73-AE8B-0A7996E0E6E6	C
6	Michael	Hussain	1985-11-22	4444 Jay St.		Manhattan	NY	11249	USA	(505) 444-4444	mihkael@mail.com	88888888	9989975A-9763-47CA-AB85-3007E6210052	C

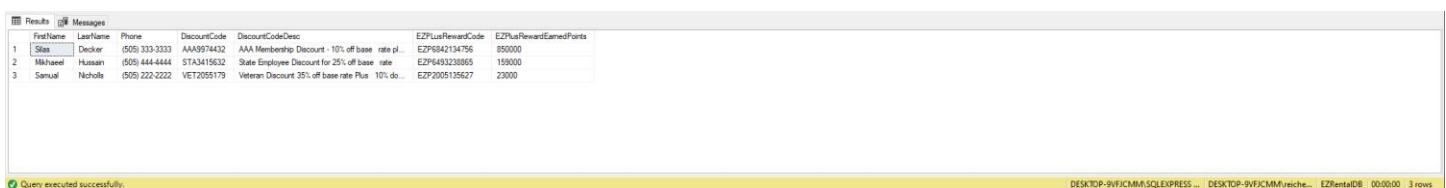
Select statement #3

Select statement that queries **three tables** (DISCOUNT, EZPLUS, and CUSTOMER) and returning **records** based on the criteria business scenario.

```
Select * from CUSTOMER where StateCode = 'NY';
```

The result of the query that has been executed, is First Name, Last Name, Phone number, Discount Code, Discount Code Description, EZPlus Reward Code, and EZPlus Reward Earned Points of all customers that have earned more than 20000 reward points.

```
Select c.FirstName, c.LastName, c.Phone, d.DiscountCode, d.DiscountCodeDesc,
       e.EZPlusRewardCode, e.EZplusRewardEarnedPoints
  from CUSTOMER as c, DISCOUNT as d, EZPLUS as e, RETAILCUSTOMER as r
 where r.CustomerID = c.CustomerID and d.DiscountID = r.DiscountID and
       e.EZPlusID = r.EZPlusID and e.EZplusRewardEarnedPoints > 20000
order by c.LastName;
```



A screenshot of a SQL query results window. The results grid shows three rows of data with the following columns: FirstName, LastName, Phone, DiscountCode, DiscountCodeDesc, EZPlusRewardCode, and EZplusRewardEarnedPoints. The data is as follows:

	FirstName	LastName	Phone	DiscountCode	DiscountCodeDesc	EZPlusRewardCode	EZplusRewardEarnedPoints
1	Steve	Decker	(509) 333-3333	AAA9974432	AAA Membership Discount - 10% off base rate plus	EZP942134756	85000
2	Mihaeel	Hussain	(509) 444-4444	ST43415632	State Employee Discount for 25% off base rate	EZP6453238865	159000
3	Samuel	Nichols	(509) 222-2222	VET2055179	Veteran Discount 35% off base rate plus 10% do...	EZP2005135627	23000

Query executed successfully.

DESKTOP-9VFJCM\SQLEXPRESS ... | DESKTOP-9VFJCM\reiche... | EzrentalDB | 00:00:00 | 3 rows

Update statements

Update statement #1

Update statement that updates **one record** of a parent to an associative child table.

The EZPLUS before executing an update statement:

	EZPlusID	EZPlusRewardCode	EZPlusRewardEarnedPoints
1	1	NEW9009854637	99999
2	2	EZP1000192461	500
3	3	EZP6493238865	199000
4	4	EZP2005135627	23000
5	5	EZP6842134756	850000

Update statement that overrides EZPlus Reward Code and EZPlus Reward Earned Points:

```
Update EZPLUS
set EZPlusRewardCode = 'NEW9009854637',
    EZPlusRewardEarnedPoints = 99999
where EZPlusID = 1;
```

	EZPlusID	EZPlusRewardCode	EZPlusRewardEarnedPoints
1	1	NEW9009854637	99999
2	2	EZP1000192461	500
3	3	EZP6493238865	199000
4	4	EZP2005135627	23000
5	5	EZP6842134756	850000

Update statement #2

Update statement that updates **one record** an associative child table.

The **CREDITCARD** before executing an update statement:

A screenshot of a SQL query results window titled "Results". It displays a table with 5 rows of data. The columns are: CreditCardNumber, CreditCardOwnerName, CreditCardIssuingCompany, MerchantCode, ExpDate, AddressLine1, AddressLine2, City, StateCode, ZipCode, Country, CreditCardLimit, CreditCardBalance, and ActivationStatus. The data is as follows:

	CreditCardNumber	CreditCardOwnerName	CreditCardIssuingCompany	MerchantCode	ExpDate	AddressLine1	AddressLine2	City	StateCode	ZipCode	Country	CreditCardLimit	CreditCardBalance	ActivationStatus
1	1111111111111111	Johan Wilks	American Express	1	2024-01-01	1111 Jay St.		Brooklyn	NY	11209	USA	10000.00	256.21	1
2	2222222222222222	Samuel Nichols	TD Bank	2	2024-09-20	2222 Jay St.		Trenton	NJ	11229	USA	8000.00	4230.86	1
3	3333333333333333	Slim Decker	American Express	3	2025-02-22	3333 Jay St.		Trenton	NJ	11239	USA	10000.00	256.21	1
4	4444444444444444	Mihamed Hussain	Chase Bank	4	2024-01-01	4444 Jay St.		Manhattan	NY	11249	USA	10000.00	256.21	0
5	5555555555555555	Veina Hulme	American Express	5	2024-01-01	5555 Jay St.		Brooklyn	NY	11259	USA	500.00	256.21	1

Query executed successfully.

Update statement that overrides Merchant Code, Credit Card Balance, and Activation Status in the **CREDITCARD** table:

Update CREDITCARD

```
set MerchantCode = '2', CreditCardBalance = 15632.26, ActivationStatus = 'False'  
where CreditCardNumber = '5555555555555555';
```

A screenshot of a SQL query results window titled "Results". It displays a table with 5 rows of data, identical to the one above but with updated values for MerchantCode, CreditCardBalance, and ActivationStatus. The data is as follows:

	CreditCardNumber	CreditCardOwnerName	CreditCardIssuingCompany	MerchantCode	ExpDate	AddressLine1	AddressLine2	City	StateCode	ZipCode	Country	CreditCardLimit	CreditCardBalance	ActivationStatus
1	1111111111111111	Johan Wilks	American Express	1	2024-01-01	1111 Jay St.		Brooklyn	NY	11209	USA	10000.00	256.21	1
2	2222222222222222	Samuel Nichols	TD Bank	2	2024-09-20	2222 Jay St.		Trenton	NJ	11229	USA	8000.00	4230.86	1
3	3333333333333333	Slim Decker	American Express	3	2025-02-22	3333 Jay St.		Trenton	NJ	11239	USA	10000.00	256.21	1
4	4444444444444444	Mihamed Hussain	Chase Bank	4	2024-01-01	4444 Jay St.		Manhattan	NY	11249	USA	10000.00	256.21	0
5	5555555555555555	Veina Hulme	American Express	2	2024-01-01	5555 Jay St.		Brooklyn	NY	11259	USA	500.00	15632.26	0

Query executed successfully.

Delete statements

Delete statement #1

Delete statement that deletes **one record** based on the primary key of one of the table that is **not** an associative entity.

The **COMPANY** before executing delete statement:

A screenshot of the SQL Server Management Studio (SSMS) interface. The title bar says 'Results' and 'Messages'. The main area shows a table named 'COMPANY' with columns: CompanyID, CompanyName, AddressLine1, AddressLine2, City, StateCode, ZipCode, County, CompanyRepName, ContactPhone, ContactEmail, CorporateDiscountPercentageRate. There are 6 rows of data. The last row, CompanyID 6, is selected. At the bottom, a yellow status bar says 'Query executed successfully.'

	CompanyID	CompanyName	AddressLine1	AddressLine2	City	StateCode	ZipCode	County	CompanyRepName	ContactPhone	ContactEmail	CorporateDiscountPercentageRate
1	1	Company 1	1111 Jay st.		Brooklyn	NY	11209	USA	Jack Reacher	(718) 111-1111	jackreacher@mail.com	23.95
2	2	Company 2	2222 Jay st.		Brooklyn	NY	11219	USA	Malachi Proctor	(718) 222-2222	malachiproctor@mail.com	2.50
3	3	Company 3	3333 Jay st.		Brooklyn	NY	11229	USA	Gilbert Dennis	(718) 333-3333	GilberDennis@mail.com	42.13
4	4	Company 4	4444 Jay st.		Brooklyn	NY	11239	USA	Jude Hudson	(718) 444-4444	JudeHudson@mail.com	20.35
5	5	Company 5	5555 Jay st.		Brooklyn	NY	11249	USA	Frazer Downes	(718) 555-5555	FrazerDowns@mail.com	68.00
6	6	Company 6	6666 Jay st.		Brooklyn	NY	11249	USA	John Hohn	(718) 666-6666	JohnHohn@mail.com	99.00

DESKTOP-9VFICMM\SQLEXPRESS... DESKTOP-9VFICMM\reiche... EZRentalDB | 00:00:00 | 6 rows

Delete statement that deletes all information of the specific company (in this case of the company that has a specific id)

```
Delete
from COMPANY
where CompanyID = 6;
```

A screenshot of the SQL Server Management Studio (SSMS) interface. The title bar says 'Results' and 'Messages'. The main area shows the same 'COMPANY' table as before, but now with only 5 rows of data. The last row, CompanyID 6, is still selected. At the bottom, a yellow status bar says 'Query executed successfully.'

	CompanyID	CompanyName	AddressLine1	AddressLine2	City	StateCode	ZipCode	County	CompanyRepName	ContactPhone	ContactEmail	CorporateDiscountPercentageRate
1	1	Company 1	1111 Jay st.		Brooklyn	NY	11209	USA	Jack Reacher	(718) 111-1111	jackreacher@mail.com	23.95
2	2	Company 2	2222 Jay st.		Brooklyn	NY	11219	USA	Malachi Proctor	(718) 222-2222	malachiproctor@mail.com	2.50
3	3	Company 3	3333 Jay st.		Brooklyn	NY	11229	USA	Gilbert Dennis	(718) 333-3333	GilberDennis@mail.com	42.13
4	4	Company 4	4444 Jay st.		Brooklyn	NY	11239	USA	Jude Hudson	(718) 444-4444	JudeHudson@mail.com	20.35
5	5	Company 5	5555 Jay st.		Brooklyn	NY	11249	USA	Frazer Downes	(718) 555-5555	FrazerDowns@mail.com	68.00

DESKTOP-9VFICMM\SQLEXPRESS... DESKTOP-9VFICMM\reiche... EZRentalDB | 00:00:00 | 5 rows

Delete statement #2

Delete statement that deletes **one record** of an associative entity table.

The **CREDITCARD** before executing delete statement:

	CreditCardNumber	CreditCardOwnerName	CreditCardIssuingCompany	MerchantCode	ExpDate	AddressLine1	AddressLine2	City	StateCode	ZipCode	Country	CreditCardLimit	CreditCardBalance	ActivationStatus
1	1111111111111111	Johan Wilks	American Express	1	2024-01-01	1111 Jay St.		Brooklyn	NY	11209	USA	10000.00	256.21	1
2	2222222222222222	Samuel Nichols	TD Bank	2	2024-03-20	2222 Jay St.		Trenton	NJ	11229	USA	8000.00	4230.86	1
3	3333333333333333	Silas Decker	American Express	3	2025-05-22	3333 Jay St.		Trenton	NJ	11239	USA	10000.00	256.21	1
4	4444444444444444	Mikhail Hussain	Chase Bank	4	2024-01-01	4444 Jay St.		Manhattan	NY	11249	USA	10000.00	256.21	0
5	5555555555555555	Vetna Hulme	American Express	2	2024-01-01	5555 Jay St.		Brooklyn	NY	11259	USA	500.00	15632.26	0
6	6666666666666666	Arut Smith	American Express	3	2024-01-01	6666 Jay St.		Brooklyn	NY	11269	USA	500.00	0.00	1

DESKTOP-9VFICMM\SQLEXPRESS ... DESKTOP-9VFICMM\reiche... EZRentalDB | 00:00:00 | 6 rows

Delete statement that deletes all records of credit cards that have balance of 0:

```
Delete
from CREDITCARD
where CreditCardBalance = 0;
```

	CreditCardNumber	CreditCardOwnerName	CreditCardIssuingCompany	MerchantCode	ExpDate	AddressLine1	AddressLine2	City	StateCode	ZipCode	Country	CreditCardLimit	CreditCardBalance	ActivationStatus
1	1111111111111111	Johan Wilks	American Express	1	2024-01-01	1111 Jay St.		Brooklyn	NY	11209	USA	10000.00	256.21	1
2	2222222222222222	Samuel Nichols	TD Bank	2	2024-03-20	2222 Jay St.		Trenton	NJ	11229	USA	8000.00	4230.86	1
3	3333333333333333	Silas Decker	American Express	3	2025-05-22	3333 Jay St.		Trenton	NJ	11239	USA	10000.00	256.21	1
4	4444444444444444	Mikhail Hussain	Chase Bank	4	2024-01-01	4444 Jay St.		Manhattan	NY	11249	USA	10000.00	256.21	0
5	5555555555555555	Vetna Hulme	American Express	2	2024-01-01	5555 Jay St.		Brooklyn	NY	11259	USA	500.00	15632.26	0

DESKTOP-9VFICMM\SQLEXPRESS ... DESKTOP-9VFICMM\reiche... EZRentalDB | 00:00:00 | 5 rows

Conclusion