



## **EZRENTAL POS DESIGN & IMPLEMENTATION**

## **TABLE OF CONTENTS**

<b>Executive Summary.....</b>	<b>3</b>
<b>Problem Statement &amp; Objectives.....</b>	<b>4</b>
<b>Project Management Methodology.....</b>	<b>5</b>
<b>Database Design Deliverable #1a - Application Business requirements.....</b>	<b>9</b>
<b>Database Design Deliverable #1b – Application Development Technical Requirements...</b>	<b>25</b>
<b>Application Physical &amp; Software Technical Architectures.....</b>	<b>35</b>
<b>Application Development Features and Functionalities (Agile Backlog).....</b>	<b>45</b>
<b>Database Management System Development Environment &amp; Physical Architecture.....</b>	<b>59</b>
<b>Project Roles &amp; Responsibilities.....</b>	<b>60</b>
<b>Database Design Deliverable #2 – ER/EER Conceptual Model Diagram.....</b>	<b>62</b>
<b>Database Design Deliverable #3 – Normalized Logical Model Diagram.....</b>	<b>65</b>
<b>Database Design Deliverable #4 – Physical Model Data Dictionary.....</b>	<b>70</b>
<b>Database Design Deliverable #5 – Physical Model Schema Design Diagram.....</b>	<b>78</b>
<b>Database Implementation Deliverable #6 – Development &amp; Implementation.....</b>	<b>80</b>
<b>Database Implementation Deliverable #7 – Implemented Physical Schema Diagram.....</b>	<b>81</b>
<b>Database Implementation Deliverable #8 – Database Validation Testing.....</b>	<b>82</b>
<b>Conclusion.....</b>	<b>83</b>

# **PROJECT 1 – EZRental Auto Rental POS Management System Database Design and Implementation**

## **Executive Summary**

The EZRentalAppDB was successfully implemented, tested, and validated using structured SQL operations in MySQL Workbench. The project focused on designing a normalized, relational database to support EZRental Inc.'s credit card processing, customer service operations, and business reporting needs. This involved defining primary and foreign key relationships across multiple entity tables including CREDITCARD, CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY, CREDITCARDNETWORKCOMPANY, CREDITCARDISSUINGBANK, and CREDITCARDCORPORATEMERCHANTBANK. Key objectives of the project included enforcing referential integrity, minimizing redundancy through normalization, and ensuring flexibility for future scalability. Using realistic sample data, each table was populated and verified through multiple SQL scripts. Business scenarios were tested via SQL SELECT, INSERT, UPDATE, and JOIN statements that reflected real-world use cases such as searching for credit card records by customer or location, updating cardholder information, changing addresses, and onboarding new payment processing merchants. Validation was documented through before-and-after queries, screenshots, and comprehensive testing of edge cases. The schema was designed to accommodate many-to-many relationships (e.g., customers owning multiple credit cards or shared cards), and each query confirmed the integrity and accessibility of related data. In addition to data integrity, the system was evaluated on how well it could support front-end application functionality, including transactional reporting, customer lookups, and administrative updates. Overall, the database accurately reflects business rules, supports efficient querying, and provides a strong foundation for integration with the EZRental customer service platform and reporting tools. Its design is modular, consistent, and ready to scale with additional business needs such as billing, analytics, or fraud prevention systems.

## **Project Statement & Objectives**

EZRental Inc., has hired NYC-Tech Solutions Inc., to design & implement a suite of Auto Rental Point-of-Sales Management System Application that include the following business modules: 1) EZRental 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 name EZRental.com intended for customers to make and manage reservations via the public internet.

The **EZRental Auto Rental Management System** features are designed to:

Allow customers, both retail and corporate customers, to reserve vehicles for renting, like other in-person or online car rental systems such as Avis, Hertz, Budget, etc.

The application needs to provide the required functionalities for our Customer Service representatives and other front-line workers in our rental agencies to service in-person customers for renting and reservation processes.

Provide the features for our business users in our corporate offices who need to create reports, perform analytics and other business functionalities related to the management of the reservations and rental of our vehicles, via our INTRENET PORTAL.

Finally, features to allow customers to make & manage vehicle reservations, profile, account etc., via the public internet.

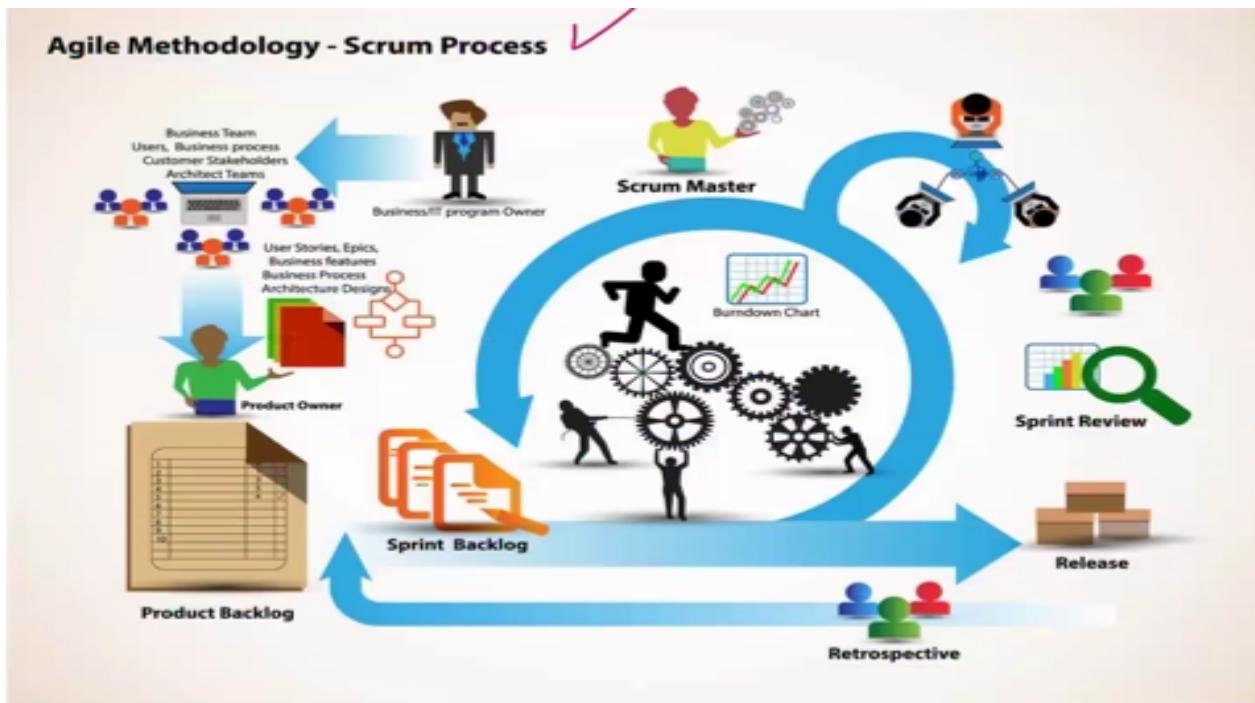
The application must be designed to support dozens of major cities around the world. In addition, provide a great user experience both in the rental agencies as well as the online systems with the best competitive pricing available in the market.

The company currently has rental agency branches in US, Canada, Mexico, United Kingdom, Japan & Australia and looking to expand further globally into other markets in Asia, Africa, and the Mediterranean.

## **Project Management Methodology**

The Agile Scrum methodology was selected after careful consideration due to its numerous advantages:

- The full list of features is divided into smaller, manageable units called SPRINTS, following an iterative process. At the end of each SPRINT, the completed features are delivered to the user.
- This approach allows for flexibility, enabling changes to requirements or new feature requests to be incorporated into future SPRINTS as needed.
- Users can start using the application as early as SPRINT #1 since features are released incrementally.
- As additional SPRINTS are completed, users gain access to more functionalities, ensuring continuous improvement and usability.



The project utilized a hybrid approach, integrating both Waterfall and Agile methodologies. The strategy for embedding Agile within Waterfall was as follows:

Waterfall phases were executed within Agile SPRINTS, ensuring a structured yet flexible development process

Each phase of database development and implementation was broken down into Agile SPRINT deliverables.

The overall Waterfall methodology was divided into four SPRINTS, spanning a total of five weeks.

A diagram and a detailed SPRINT description table are provided below to illustrate the breakdown.

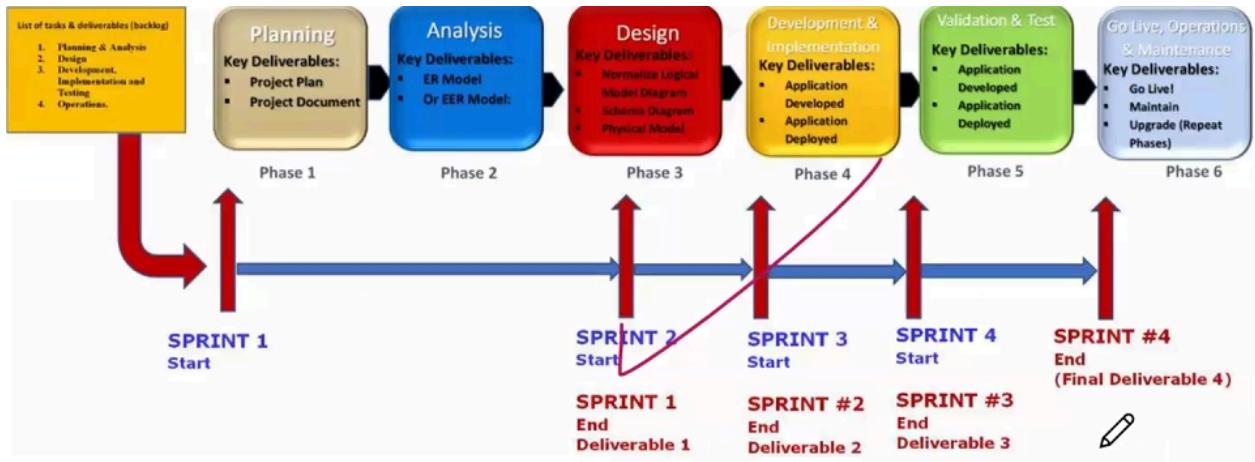
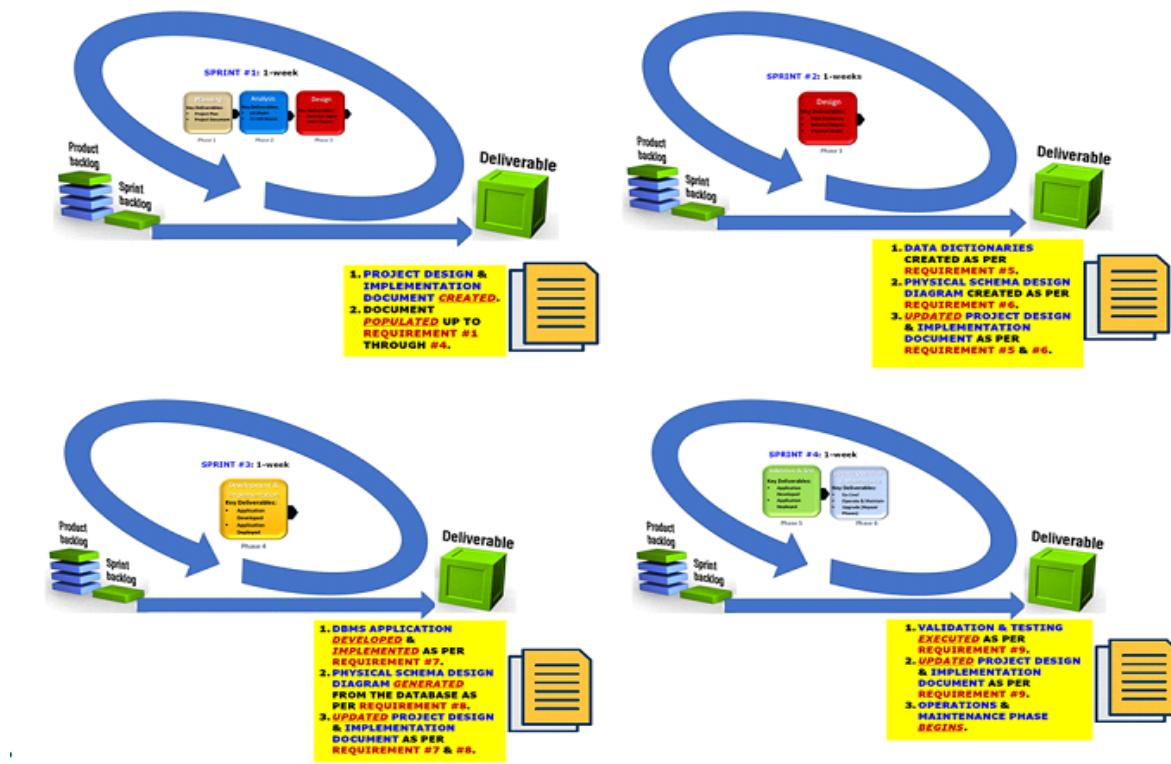


Table below shows Agile Sprints phases in detail.

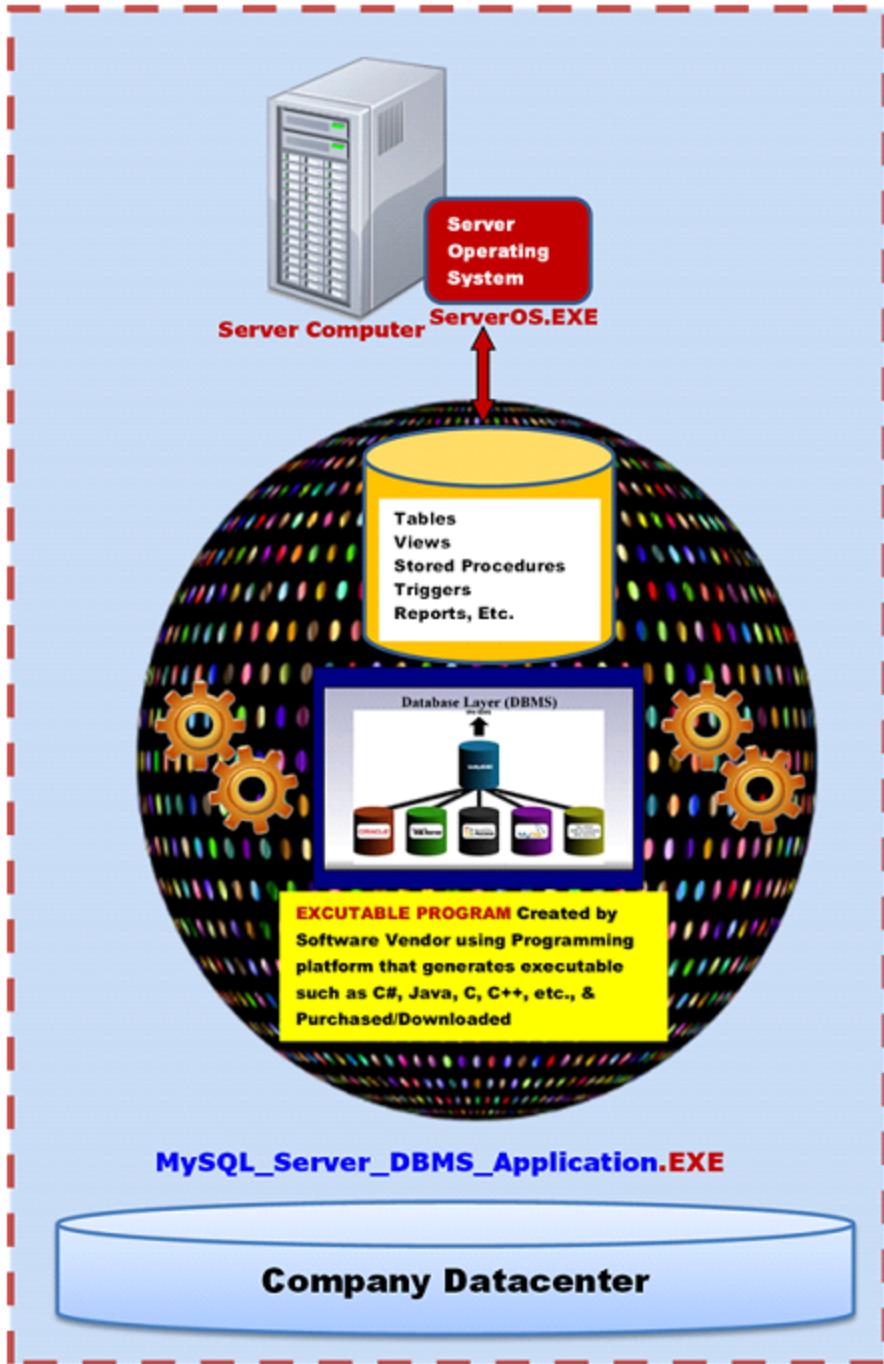
AGILE SPRINT #	WATERFALL PHASE	Output Deliverable
SPRINT #1	Planning	1. <b>Create Project Document</b> – Formatted and populated as per requirements. 2. <b>Business Requirements (Included in Project Document)</b> – List of Business & Technical Requirements from customer.
	Analysis	3. <b>ER/EER Conceptual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

Each Sprint was included throughout the Agile Methodology Process for a grand total of four sprints going over the course of five weeks.



Outcome being a fully designed and implemented MS SQL Server Database Management System for all three Auto Rental Management System Client/Server Applications

- **Desktop Window Two- Tiered Windows-Client Client/Server Application** in the Rental Agencies.
- **Web Portal Three-Tiered Web-based Client/Server** for corporate offices.
- **Web Portal Three-Tiered Web-based Client/Server** for customers via the Internet.



## **Database Design Deliverable #1a – Application Business requirements**

Abel Rodriguez hired a Database Business Analyst to interview EZRental Inc.'s project and business stakeholders. The purpose of these interviews was to gather and compile a comprehensive list of business data requirements for the application, collectively referred to as the Business Requirements.

These Business Requirements served as the foundation for the Database Analyst/Architect to initiate the first step of database design: Data Modeling. Based on these requirements, one of the fundamental database design diagrams was created:

- **Conceptual Entity-Relationship Diagram (ER)**
- **Conceptual Enhanced Entity-Relationship Diagram (EER)**

The following pages detail the Business Requirements captured by the Business Analyst, Khalid Mustapha.

Cont. next page ==>

## 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 Rental Agencies:**

A **rental agency** is the location where customers visit to pick up and drop off rental vehicles. Each **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.

### **Requirements for All Customers (Retail & Corporate Customers)**

To run our business, the application must store the following customer information for both types of **customers (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. Note that in the USA the format for the driver license number can be under 15 characters, but in other countries it could go up to 22 to 25 characters) *driver license expiration Date* and *driver license state*. In addition, note the following **business rule** policy regarding 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 main 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**. For our credit card processing and transactions, we need to capture the following *credit card* components: *credit card number* that uniquely identifies the credit card and is a 16-character number digits. We also need to capture the *credit card owner name*, in addition, credit card processing attributes such as *credit card issuing bank code*, *credit card issuing bank name*, *credit card network company code*, *credit card network company name*, *credit card processing merchant service company code*, *credit card merchant service company name*, *credit card corporate merchant bank code*, and *credit card corporate merchant bank name*. Important – further details on these credit card processing attributes will be provided in sections to follow. Is important that these attributes are clearly understood to correctly design the system.
- 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*, *credit card limit* (which is the maximum amount of money a customer can charge on their credit card), *credit card available credit* (which is how much you have left to spend with your credit card or unused amount within your limit). Note that we will capture the credit card limit to a maximum of \$999,999.99, since we don't expect our customers to have a credit limit of \$1 Million dollars. Finally, *activation status* (which is true if the credit card is active and can be used, or false when the credit card is not active or disabled).
- During the interview with business stakeholders provided the following ***Business Rules*** related to a credit card:
  1. You cannot reserve or rent one of our vehicles without a credit card.
  2. A customer can have many credit cards they can use to pay for rental transactions.
  3. 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.

### The Credit Card processing Workflow

Processing of the credit card transactions is a key part of this business and is important that we store data for each step of the credit card processing process. Therefore, the credit card processing attributes discussed in previous section need to be further analyzed and clearly understood. We will now provide the definition and detailed information on each of these credit card processing attributes: *credit card issuing bank code*, *credit card issuing bank name*, *credit card network company code*, *credit card network company name*, *credit card processing merchant service company code*, *credit card merchant service company name*, *credit card corporate merchant bank code*, and *credit card corporate merchant bank name*:

- **Credit Card Processing Merchant Service Company** – In credit card processing the merchant is the retailer where a customer purchased the goods and services and pay using a credit card. EZRental Inc., is the merchant in this scenario. The *Credit Card Processing Merchant Service Company* is the institution which works directly with the merchant (EZRental Inc.) to provide handle the credit card processing services and handles all the complexity of credit card processing and interactions with the other financial entities involved in the credit card processing process on behalf of the merchant (EZRental Inc.). The *Credit Card Processing Merchant Service Company* provides the Merchant or Business with the hardware which the customer swipes or inserts to pay for goods and services with their credit card. As part of the credit card processing cycle, the first financial institution which the *Credit Card Processing Merchant Service Company* interacts with is the *Credit Card Network Company* (which we will cover next). The *Credit Card Processing Merchant Service Company* ensure the merchant (EZRental Inc.) is connected to the right *Credit Card Network Company*. We will describe the *Credit Card Network Company* next, nevertheless, we need to capture the following information for the *Credit Card Processing Merchant Service Company*:
  - **Credit Card Processing Merchant Service Company Code** – In our business, we use and store a number code used to identify the *Credit Card Processing Merchant Service Company*. This code has business meaning and appears in reports and discussed by users. Therefore, a *Credit Card Processing Merchant Service Company* can be identified by this code.
  - **Credit Card Processing Merchant Service Company Name** – In our business, we also use and store the name of the *Credit Card Processing Merchant Service Company*. This name also has business meaning and appears in reports and discussed by users. Therefore, a *Credit Card Processing Merchant Service Company* can be identified by its name.
  - Below is a listing of the values/instances of the *Credit Card Processing Merchant Service Company Code* and *Credit Card Processing Merchant Service Company Name* we use at EZRental Inc.

Credit Card Processing Merchant Service Company Code	Credit Card Processing Merchant Service Company 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

## Business Requirements

### Our Customers & Credit Card Processing (Cont.):

- **Credit Card Network Company** – In credit card processing the objectives of the *Credit Card Network Company* is to process transactions between the *Credit Card Issuing Bank* (which we will cover next) and the *Credit Card Processing Merchant Service Company*. Covered previously. The *Credit Card Network Company* act like bridges between the *Credit Card Issuing Bank* that issue credit card and the *Credit Card Processing Merchant Service Company* that handles the transaction from the merchant (EZRental Inc.). The *Credit Card Network Company* that interacts with the *Credit Card Issuing Bank* to determine whether to approve or deny the transaction. And then the *Credit Card Network Company* notifies the merchant (EZRental Inc.) if the purchase was approved or denied. The *Credit Card Network Company* is a digital infrastructure that facilitates credit card transactions and prepares the transaction for the *Credit Card Issuing Bank*. We will describe the *Credit Card Issuing Bank* next, nevertheless, we need to capture the following information for the *Credit Card Issuing Bank*:
  - **Credit Card Network Company Code** – We use and store a number code to identify the *Credit Card Network Company*. This code has business meaning and appears in reports and discussed by users. Therefore, a *Credit Card Network Company* can be identified by this code.
  - **Credit Card Network Company Name** – In our business, we also use and store the name of the *Credit Card Network Company*. This name also has business meaning and appears in reports and discussed by users. Therefore, a *Credit Card Network Company* can be identified by its name.
  - Below is a listing of the values/instances of the *Credit Card Network Company Code* and *Credit Card Network Company Name* we use at EZRental Inc.:

<i>Credit Card Network Company Code</i>	<i>Credit Card Network Company Name</i>
1	American Express
2	Visa
3	Mastercard
4	Discover
5	Diners Club
6	Interlink
7	Star
8	Accel
9	Interac
10	Visa ReadyLink
11	Pulse
12	JCB (Japan Credit Bureau)
13	Rupay

- **Credit Card Issuing Bank** – The *Credit Card Issuing Bank* is the financial or lending institution that offers the Credit Card and pays for the goods and services until the customer pays back the credit/loan. These are Banks, Lending Institutions, Credit Unions, Fintech companies, etc. These institutions issues/provides the credit for the customer. The cardholder borrows money from the credit card issuing bank each time they make a purchase, and when they pay their credit card bill, they're paying the *Credit Card Issuing Bank*. We need to capture the following information for the *Credit Card Issuing Bank*:
  - **Credit Card Issuing Bank Code** – In our business, we use and store a number code used to identify the *Credit Card Issuing Bank*. This code has business meaning and appears in reports and discussed by users. Therefore, a *Credit Card Issuing Bank* can be identified by this code.
  - **Credit Card Issuing Bank Name** – In our business, we also use and store the name of the *Credit Card Issuing Bank*. This name also has business meaning and appears in reports and discussed by users. Therefore, a *Credit Card Issuing Bank* can be identified by its name.
  - Below is a listing of the values/instances of the *Credit Card Issuing Bank Code* and *Credit Card Issuing Bank Name* we use at EZRental Inc.:

<i>Credit Card Issuing Bank Code</i>	<i>Credit Card Issuing Bank Name</i>
1	American Express
2	Bank of America
3	Barclays
4	Capital One
5	Chase
6	Citi
7	Discover
8	Synchrony Bank
9	U.S. Bank
10	Wells Fargo

## Business Requirements

### *Our Customers & Credit Card Processing (Cont.):*

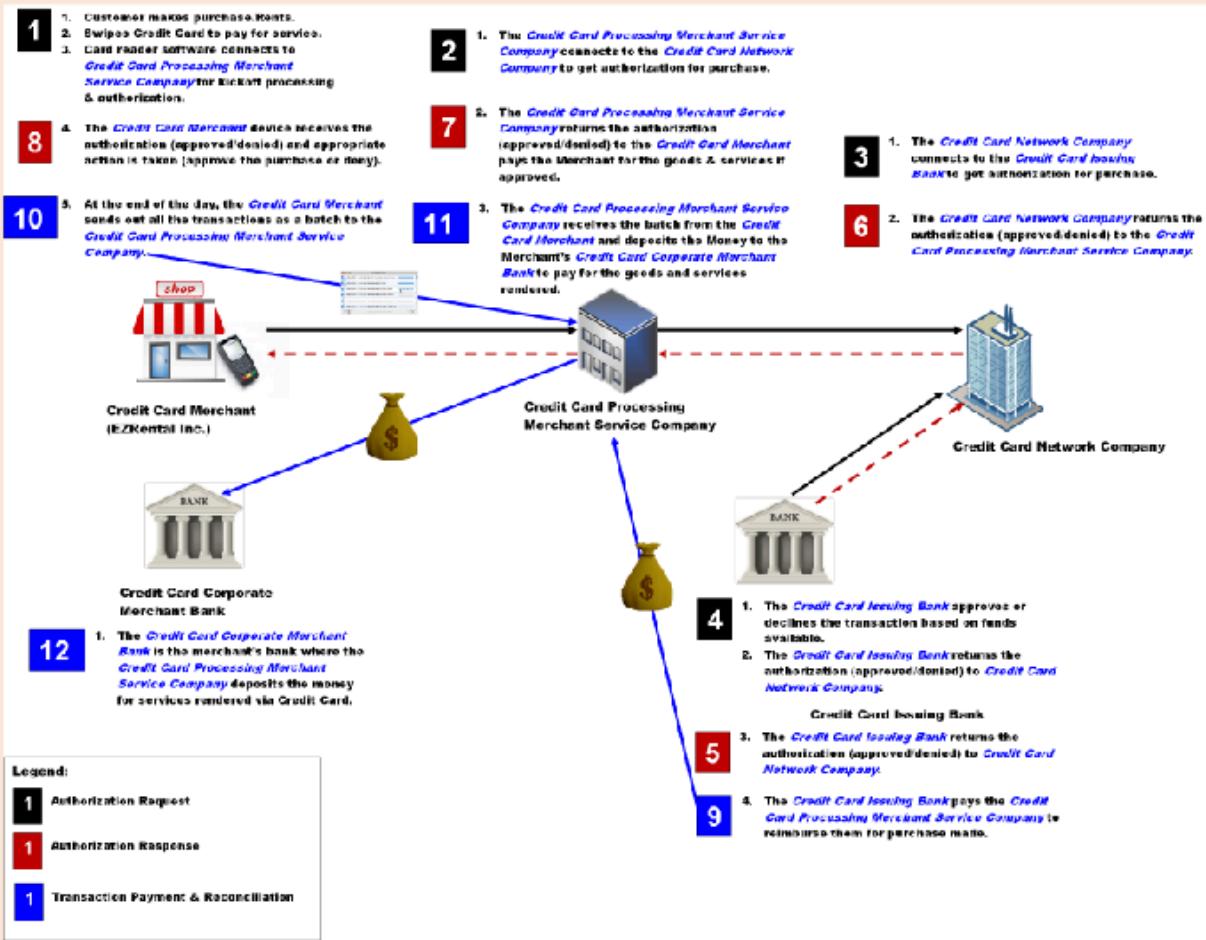
- **Credit Card Corporate Merchant Bank** – In credit card processing, the *Credit Card Corporate Merchant Bank* is the bank used by the merchant EZRental Inc., to handle credit card processing money transactions, payments, etc., between EZRental Inc., and the *Credit Card Processing Merchant Service Company* that handles the Credit Card Processing on behalf of EZRental Inc. In short, it is the bank that has the bank account used by EZRental Inc., to handle the accounting for Credit Card Processing Transactions. EZRental Inc., currently uses three banks to handle all their banking transactions. They don't expect to add many Corporate Merchant Banks throughout the next 5+ years, but business is everchanging, nevertheless, expectations is that there are no plans to add any new banks no time soon.
- We need to capture the following information for the *Credit Card Corporate Merchant Bank*:
  - **Credit Card Corporate Merchant Bank Code** – In our business, we use and store a number code used to identify the *Credit Card Corporate Merchant Bank*. This code has business meaning and appears in reports and discussed by users. Therefore, a *Credit Card Corporate Merchant Bank* can be identified by this code.
  - **Credit Card Corporate Merchant Bank Name** – In our business, we also use and store the name of the *Credit Card Corporate Merchant Bank*. This name also has business meaning and appears in reports and discussed by users. Therefore, a *Credit Card Corporate Merchant Bank* can be identified by its name.
  - Below is a listing of the values/instances of the *Credit Card Corporate Merchant Bank Code* and *Credit Card Corporate Merchant Bank Name* we use at EZRental Inc.:

<i>Credit Card Corporate Merchant Bank Code</i>	<i>Credit Card Corporate Merchant Bank Name</i>
1	Chase
2	Citi
3	Capital One

## Business Requirements

### Our Customers & Credit Card Processing (Cont.):

Below, is a pictorial representation of the interaction between the credit card processing entities **Merchant (EZRental Inc.)**, **credit card processing merchant service company**, **credit card network company** and **credit card issuing bank**:



In summary, we need to capture the data for the credit card processing attributes: **credit card issuing bank code**, **credit card issuing bank name**, **credit card network company code**, **credit card network company name**, **credit card processing merchant service company code**, **credit card merchant service company name**, **credit card corporate merchant bank code**, and **credit card corporate merchant bank name**.

## 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 (0.0x) 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 **EZRental 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 Customer Discounts

**Retail Customers** can (but don't have to) leverage promotional **discounts** or multiple coupons obtain from other businesses, internet, magazine, organizations, etc., to save money on their rentals. Therefore, we need to capture specific data for the promotional discounts used by a retail customer. A **Promotional Discount** is composed of the following attributes: **discount ID**, a unique random number which uniquely identifies a discount, another 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..

The following business rules were identified regarding the retail customer discount program:

1. Only a Retail Customer can use Discounts, no other type of customer, e.g., Corporate Customer, can apply a discount.
2. Discounts are applied during reservation of a vehicle or during the actual rental process only.
3. A Retail Customer can apply multiple Discounts throughout their lifetime as an EZRental Inc., customer, NEVERTHELESS, ONLY ONE Discount can be applied for a reservation or rental instance. You cannot apply multiple discounts to a reservation or rental.
4. A Discount can be used by many Retail Customers and many Retail Customers can use a Discount.
5. When a Discount is used by a Retail Customer, we need to capture the **Discount Submitted Date** which is the date the customer provided the discount and the **Discount Redeemed Date** which is the date the customer used the discount for a rental. In this business a customer can submit a discount during registration on one date but use it in a future date when they are renting.

## Business Requirements

*Our Customers (Cont.):*

### Retail Customer EZPlus Rewards Program

**Retail customers** can opt-in to enrolled in the EZPlus Rewards Program where they earn points for every rental of a vehicle. These rewards 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..

The following business rules apply to the EZPlus Rewards Program:

1. Only a Retail Customer can leverage the [EZPlus Rewards Program](#), no other type of customer such as a Corporate Customer can join the EZPlus program.
2. The [EZPlus Rewards Program](#) is OPTIONAL. A Retail Customer can join the [EZPlus Rewards Program](#) during registration or any other time after or not join at all.
3. A Retail Customer can drop out of the EZPlus Rewards program at any time.
4. Every time a Retail Customer that is member of the [EZPlus Rewards Program](#) rents a vehicle, they earn **1000 EZPlus Rewards Points**. When a Retail Customer member of the [EZPlus Rewards Program](#) earns **10,000 EZPlus Rewards Points** they earn a **FREE RENTAL**.

As an incentive for our retail customers to join the [EZPlus Rewards Program](#) during registration to become a customer, we offer a **EZPLus sign-up rewards points of 1000 EZPlus Rewards Earned Points**. Also note that the maximum number of *EZPlus Rewards Earned Points* is capped at 50,000 points. A Retail Customer cannot accumulate more than 50,000 points in the [EZPlus Rewards](#) program.

## Business Requirements (Cont.)

### **Our Customers (Cont.):**

In this business, we have the following business rules for our customers (*Retail* or *Corporate*):

1. We only have two types of customers retail customer or corporate customers. No other type of customer exists.
2. The minimum age to be a customer and rental our vehicles is 21 years old. This rule must be enforced.
3. 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.

### **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:

<i>Transmission Type</i>	<i>Transmission Type Description</i>
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:

<i>Vehicle Status ID</i>	<i>Vehicle Status Description</i>
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 have the following business rule relate to a vehicle and a vehicle rental category:

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.
5. For reporting and analytics, we need to capture all changes to the reservation's pick-up and drop-off agency, such as all changes by the customer for pick-up agency & drop-off agency later after the original reservation. This means we need to store all history on all reservation pick-and drop-off agency changes.

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 to store the *reservation schedule* composed of *reservation pick up date*, *reservation pick up time*, *reservation drop off date* and *reservation drop off time*, also the *reservation estimated rental cost*. A reservation can have multiple schedule changes during the lifetime of the reservation since customers can make changes to the reservation and we need to track this history of changes for analytical purposes.

## Business Requirements (Cont.)

### Reservation Process (Cont.):

We have the following business rule relate to the *reservation schedule*:

1. For reporting and analytics, we need to capture all changes to the reservation schedule, such as all changes by the customer for pick-up date & time and drop-off date & time. This means we need to store all history on all reservation schedule changes.
2. A customer can have MANY reservation schedules based on changes to the reservation, but a schedule can only belong to ONE customer.

- 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 we must adhere to the following business rules:

1. A customer can make none, one or many reservations for a vehicle rental category at a rental agency.
2. A rental category can be reserved by none, one or many customers at a rental agency.
3. A rental agency can get many or no reservations for a vehicle rental category by a customer.
4. A reservation can only have one pick-up rental agency location, but a rental agency can have many reservation pick-ups happening.
5. 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:

1. 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.
2. 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.
3. During the rental process we may have any of the following business rules/scenarios:
  - 1) A vehicle can be picked up at the **SAME** rental agency as indicated by the reservation and dropped off at the **SAME** rental agency.
  - 2) Or a vehicle can be picked up at the **SAME** rental agency as indicated by the reservation and dropped off at **ANOTHER** rental agency.
  - 3) 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.
  - 4) 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.
  - 5) For reporting and analytics, we need to capture all changes to the rental pick-up and drop-off agency, such as all changes by the customer for pick-up agency & drop-off agency later after the original reservation. This means we need to store all history on all rental pick-and drop-off agency changes.
- ❖ 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:

1. A rental can only be for one pick-up rental agency, but a rental agency can have many rental pick-ups taking place.
2. 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, descriptions, and cost:

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 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*.

The following business rules related to employees must be followed:

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., therefore, the EmployeeID is the unique ID for an employee to be identified and managed from a business perspective.*
4. *The minimum age to be an employee of our company 18 years old. This rule must be enforced.*

## 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 identity 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:

<i>Transport Reason ID</i>	<i>Transport Reason Description</i>
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 to and from Rental Agency is executed by an employee who is part of a transportation team or 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:

<i>Transport Status ID</i>	<i>Transport Status Description</i>
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 to be able to know where our vehicles are located at any time and their status.

**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.

## Database Design Deliverable #1b – Application Development Technical Requirements

### Application Development & Technical Requirements Overview

Abel Rodriguez hired an **Application Analyst/Architect** to interview EZRental Inc.'s **Business Decision Makers (BDMs)** and **Information System Technical Decision Makers (TDMs)**. The goal was to gather and compile the **Application Development & Technical Requirements** necessary for designing and developing the application.

The **Technical Requirements** were established collaboratively by the **Object-Oriented Application Architect** and the **UX/User-Interface Architect** to support the following design and development aspects:

#### Derived Diagrams & Design Components:

- **Application Physical Architecture Design**
- **Application Software Design & Programming**
- **Application Feature List (AGILE Backlog)**
- **Other Application-Related Implementations**
- **User Interface (UI) Wireframe Design & Programming**

These elements contribute to the **Object-Oriented Object Model Design**, **User Interface Wireframing**, and **UI Design**, ensuring a structured and efficient approach to application development.

Cont. next page ⇒

## 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, Node.js 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	<b>EZRental</b> Rental Agency Point-of-Sales (POS) System	<ul style="list-style-type: none"> <li>Car Rental, Car Return, New Customer Registration &amp; Search/Print Customer Information, Customer Update, Customer Deletion, Customer Listing operations etc.</li> </ul>
2	<b>EZRental</b> Rental Agency Back-Office Vehicle Inventory Management System	<ul style="list-style-type: none"> <li>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.</li> <li>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.</li> <li>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.</li> </ul>
3	<b>EZRental</b> Rental Agency Back-Office Credit Card Management System	<ul style="list-style-type: none"> <li>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 &amp; Delete credit card information in the database</li> </ul>
4	<b>EZRental</b> Rental Agency Back-Office Employee & Customer User Account Management System	<ul style="list-style-type: none"> <li>The EZRental Customer &amp; Employee User Account Management System is a Back-end system meant for IT ADMINISTRATOR Employees to manage both Employee &amp; Customer USER ACCOUNTS.</li> </ul>
5	<b>EZRental</b> Rental Agency Desktop Application Security Authentication System	<ul style="list-style-type: none"> <li>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.</li> </ul>

### 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.):

- o **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.

- o **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.

- o **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:

<i>Vehicle Status ID</i>	<i>Vehicle Status Description</i>
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.

- o **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.):

- o **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.

- o **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.):

- o **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.

- o **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

- o **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

- o **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:

<i>Transport Reason ID</i>	<i>Transport Reason Description</i>
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.

- o **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:

<i>Transport Status ID</i>	<i>Transport Status Description</i>
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	<b>EZRental.com</b> Customer Web Portal	<ul style="list-style-type: none"> <li>▪ Front-end WEB INTERFACE SCREENS &amp; features used by customers via our web portal EZRentalCar.com to reserve a vehicle for rental and manage their account online.</li> <li>▪ Features include search &amp; reserve a car for rental, register as a new customer, search/view their account information, update their account etc.</li> </ul>
2	<b>EZRental.com</b> Customer Web Portal Application Security Authentication System	<ul style="list-style-type: none"> <li>▪ Proper security and authentication must be implemented to make sure only our customer can access the web portal to use the application.</li> </ul>

#### 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.):

- o **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.

- o **Rental Agency:**

- Web pages that required adding a RENTAL AGENCY field should be prepopulated with the list of rental agencies in our company.

- o **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 Ctgory 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 & Software Technical Architectures

### EZRental POS Application - Physical Architecture Overview

During design meetings with the application architects and full-stack developers, a comprehensive review of business and technical requirements led to key decisions regarding the physical architecture of the EZRental POS application. The architecture is structured to optimize performance, enhance user experience, and support the diverse needs of rental agency employees, corporate staff, and customers.

### Application Requirements & Architecture Overview

#### 1. Rental Agency System (Windows Desktop Client/Server - Two-Tier Architecture)

The system used at rental agencies must provide **high performance** and **fast response times** to efficiently serve customers, especially in high-traffic locations such as airports. The interface must also be user-friendly to facilitate quick execution of tasks by frontline employees, including customer service representatives, inventory managers, and support staff.

#### Key Features:

- **POS Customer Management:**
  - Customer Search & Print
  - New Customer Registration
  - Customer Updates & Deletion
  - Customer Listing
- **POS Vehicle Reservation, Rental & Return Management:**
  - Vehicle Reservations
  - Vehicle Rentals
  - Vehicle Returns
- **POS Vehicle Inventory Management:**
  - Search, Add, Update, Delete, Print, and List vehicles (Cars, SUVs, Mini-Vans, Cargo Vans, etc.)
- **POS Credit Card Management:**

- Credit Card Search & Print
- New Credit Card Registration
- Credit Card Updates & Deletion
- Credit Card Listing

Since customer interactions at rental agencies require **real-time processing**, a **Windows-based client-server architecture** is best suited to ensure minimal wait times and maximum efficiency.

## 2. Corporate Office System (Three-Tier Web-Based Client/Server - Intranet Web Portal)

The corporate office system serves employees who manage business operations, rentals, and customer accounts through an **intranet-based web portal**. While performance is important, the system does not require the same level of responsiveness as the rental agency application.

Key Features:

- **Enterprise Resource Planning (ERP) Integration:**
  - Customer Credit Card Management
  - Vehicle Inventory Management
  - Customer Relationship Management (CRM)
  - Human Resource Management (HRM)
  - Finance & Operations
  - Marketing & Customer Service
- **Web-Based EZRental POS Corporate Management System:**
  - Customer Profile Search & Management
  - Customer Registration, Updates, and Deletion
  - Customer Listing & Account Management
  - Vehicle Reservations & Rental Management

This system is designed to be accessed through a **browser-based intranet portal**, providing corporate employees with the necessary tools to manage daily business operations efficiently.

### 3. Customer Self-Service System (Three-Tier Web-Based Client/Server - Internet Web Portal)

Customers should be able to make and manage their reservations online from anywhere in the world via the **EZRental.com** web portal. The system must offer an intuitive user interface and smooth performance to enhance the online rental experience.

Key Features:

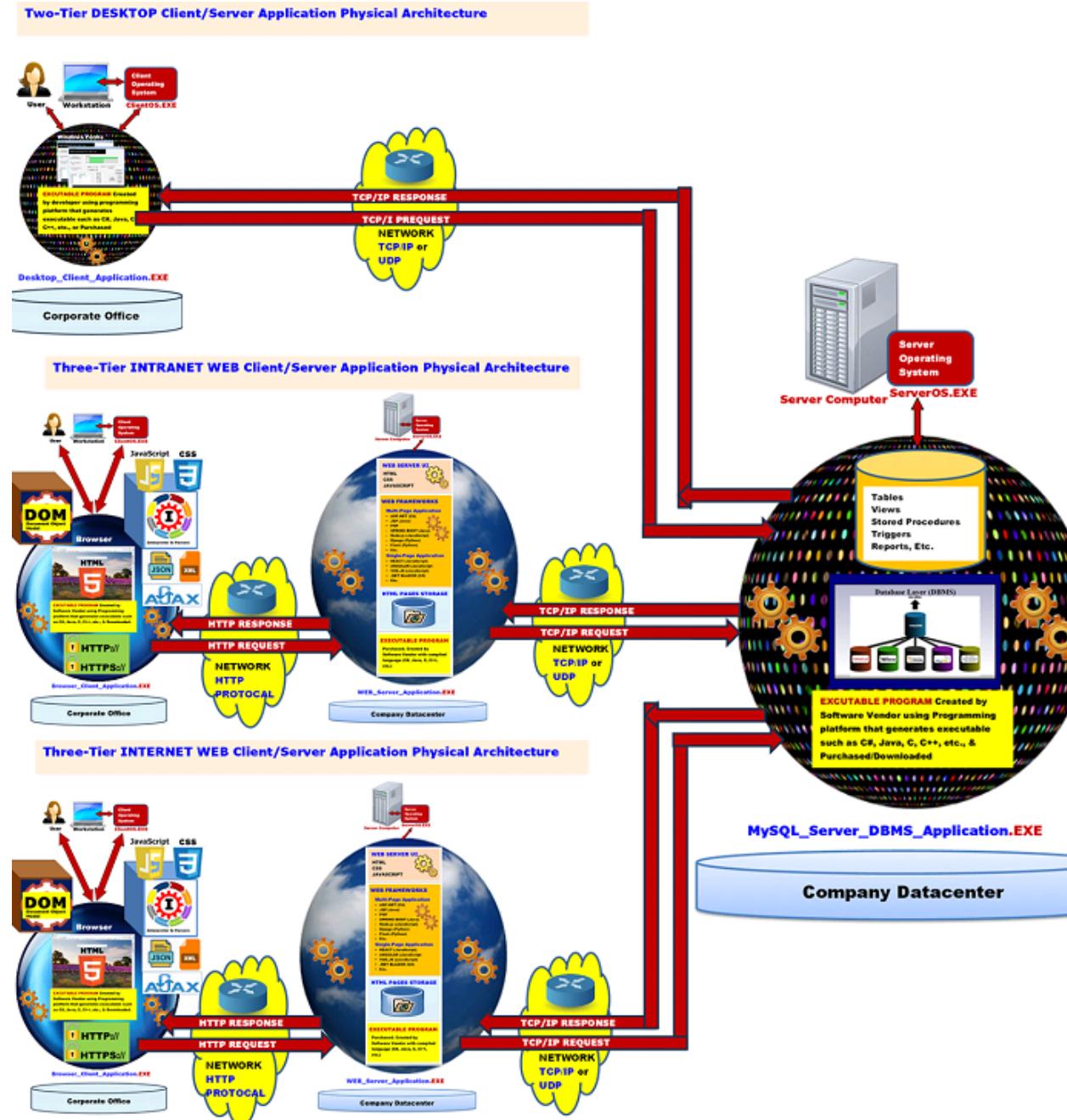
- **Web-Based EZRental POS for Customers:**
  - Customer Profile Search & Management
  - Customer Registration, Updates, and Deletion
  - Customer Listing & Account Management
  - Vehicle Reservations & Rental Management

This application is implemented as a **three-tier web-based system**, ensuring scalability and accessibility for customers using a web browser.

Database Architecture

All three applications—**Rental Agency (Windows Client/Server)**, **Corporate Office (Web Intranet)**, and **Customer Internet Portal (Web Internet)**—will share a **common database tier**. This unified database ensures data consistency across the entire EZRental ecosystem while supporting real-time transactions and reporting.

The chosen architecture balances **performance, scalability, and user experience** across all levels of the EZRental POS system. By leveraging a **two-tier architecture** for high-performance agency operations and **three-tier web-based systems** for corporate and customer-facing applications, EZRental ensures efficient and seamless management of rental operations across all platforms.



## Physical Architecture Hardware & Software Inventory

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

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

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

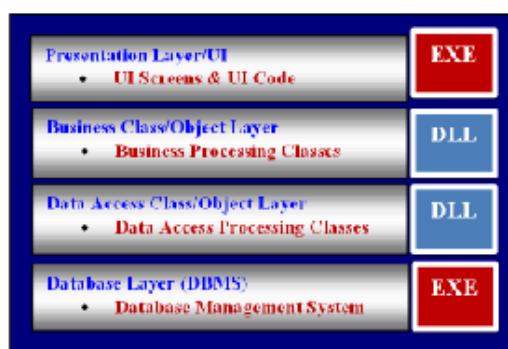
## Application Software Architecture Overview

- During the design meetings with the application architects and full-stack developers, from the **physical application architecture** decisions, a decision was also made on the **software application architecture** to be used to **design** and **program** the **EZRental POS** application for each of the **physical architectures**.
  - **Rental Agency Two-Tiered Windows Desktop Client/Server Application.**
  - **Corporate Office Three-Tiered Web-based Client/Server.**
  - **Customer Facing Internet Three-Tiered Web-based Client/Server.**
  - All sharing the **Database Tier supporting all Three Applications (Rental Agency, Corporate Office & Customer Internet).**

### Rental Agency Two-Tiered Windows Desktop Client/Server Application

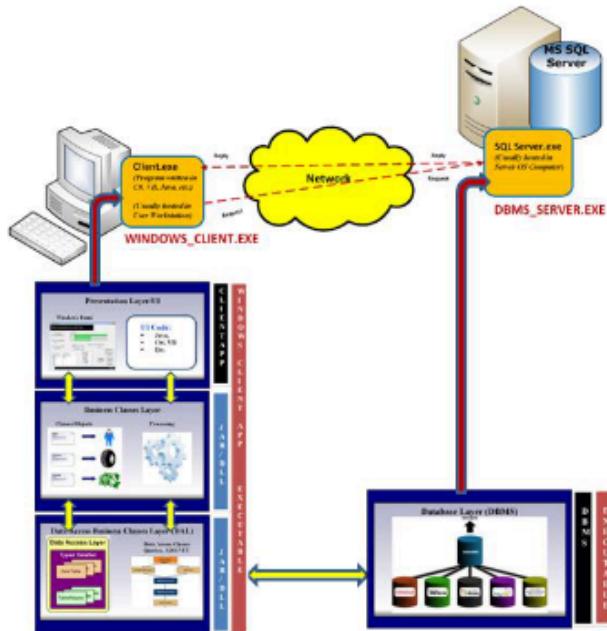
- For the Rental Agencies **Two-Tiered Windows Desktop Client/Server Application** we will use the scalable **Four-Layer Windows Client/Server Software Programming Architecture**:

4 Tiers Windows Client/Server Application Architecture

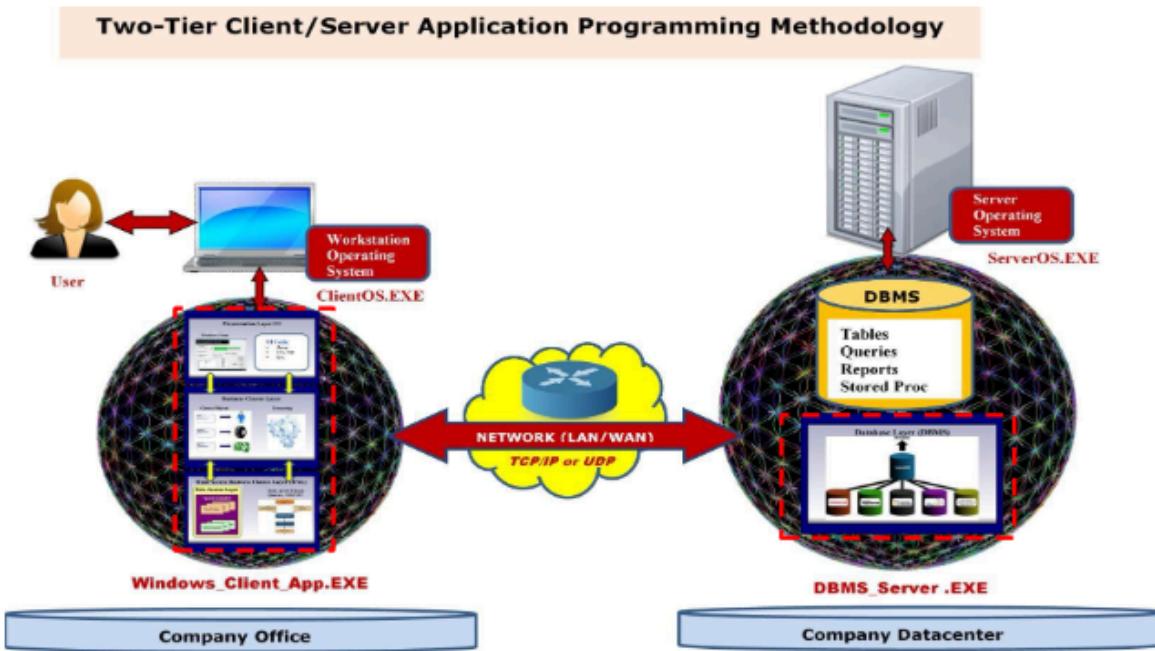


- The layers will be programmed as shown below as a **Windows Desktop FAT CLIENT Application** requesting database services from the **Database Management System Application**:

- High-level view of software architecture embedded within the physical architecture:



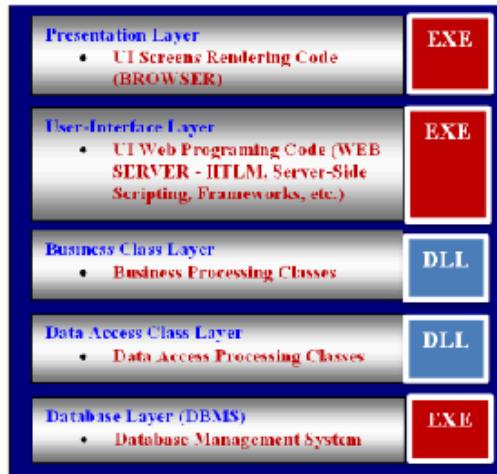
- Detailed view of software architecture embedded within the physical architecture:



### Corporate Office Three-Tiered Web-based Client/Server Application

- For the **Corporate Offices INTRANET (internal corporate network) Portal Three-Tiered Web-based Client/Server Application** we will use the scalable **Five-Layer Web Client/Server Software Programming Architecture**:

**5 Tiers Web Client/Server Application Architecture**

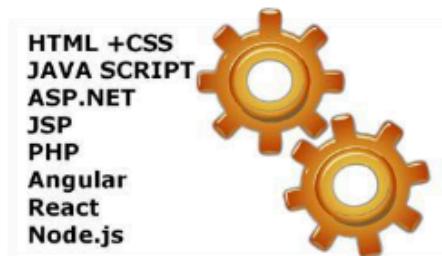


- **IMPORTANT!** Note the following:

- Because we need very tight control of all programming features for customization, **we will avoid using third-party programming frameworks where possible**.
- Nevertheless, **due to the nature of Web Development and advances in Web Technology Frameworks**, we will consider leveraging **Third-Party Frameworks** for the following Layer in the **Web Server Application**:

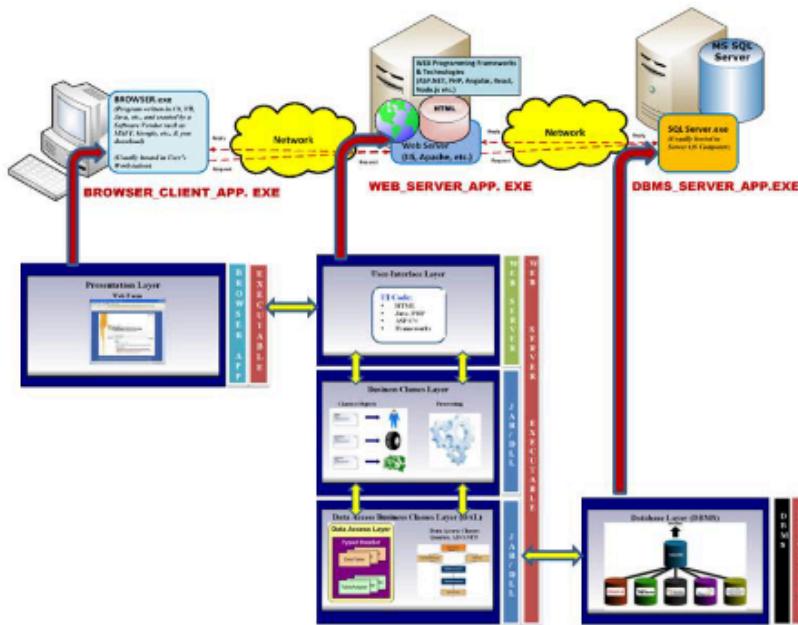


- These can include any of the following **frameworks and technologies**:

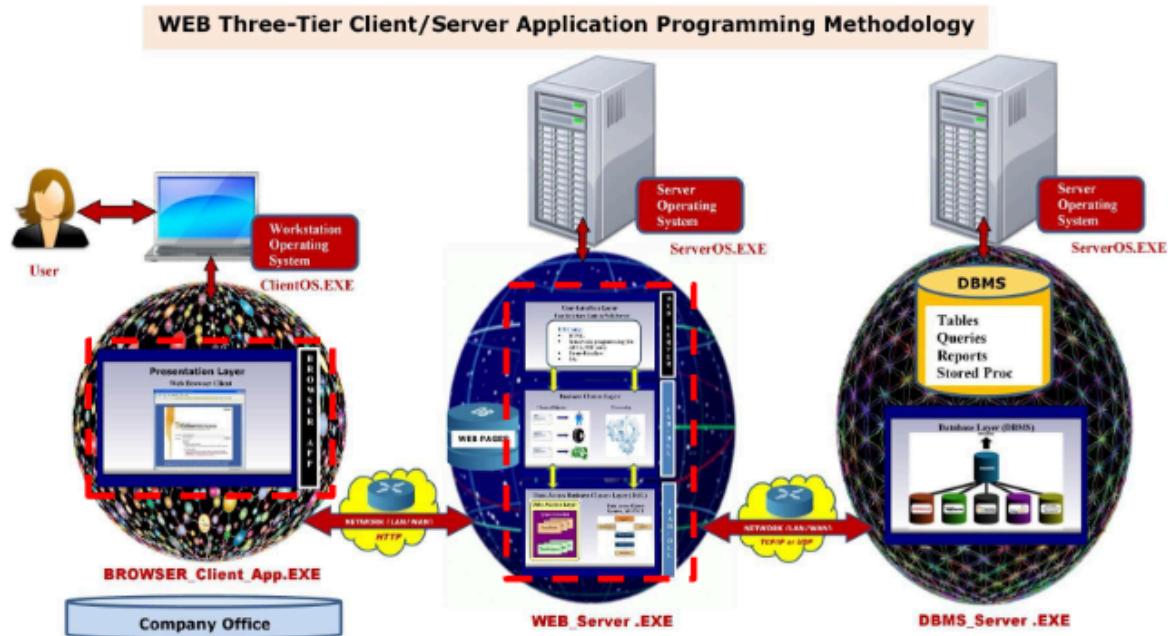


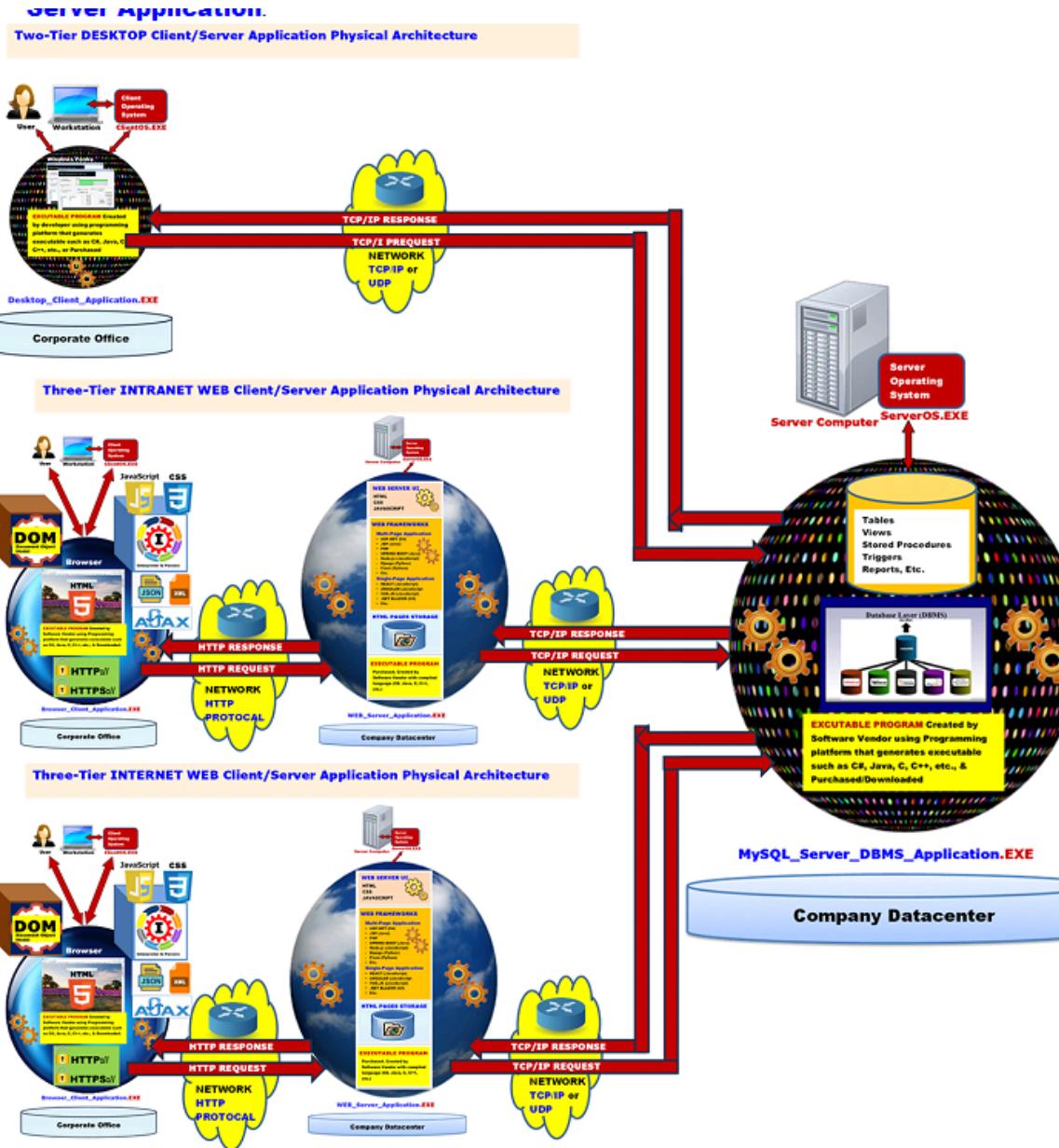
- **In summary, we will be developing all the layers of the application but leveraging Web Third-Party Frameworks to implement our Web Pages in the User-Interface Layer in the Web Server.**

- The layers will be programmed as shown below with a **THIN CLIENT BROWSER Application** requesting web services from a **Web Server Application** and the **Web Server Application** requesting database services from the **Database Management System Application**:
  - High-level view of software architecture embedded within the physical architecture:



- Detailed view of software architecture embedded within the physical architecture:





## Application Development Features and Functionalities (Agile Backlog)

The **Application Development Features and Functionalities (Agile Backlog)** serve as a structured list of prioritized tasks, enhancements, and requirements necessary for the development of the application. In an Agile environment, the backlog evolves continuously, incorporating user stories, functional and non-functional requirements, bug fixes, and technical debt items. Each backlog item is detailed with descriptions, acceptance criteria, and priority levels, ensuring that development aligns with business goals and user needs. Features may include user authentication, data processing, reporting, UI/UX enhancements, API integrations, and performance optimizations. The backlog is refined through regular grooming sessions, ensuring that development remains iterative, flexible, and responsive to stakeholder feedback, ultimately delivering a high-quality application.

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

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

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

Feature #	Feature Description
<b>FEATURE #4A</b>	<p><b>FEATURE #4A – EZRental Rental Agency Point-of-Sales (POS) BACK-OFFICE DESKTOP APPLICATION CREDIT CARD MANAGEMENT SYSTEM:</b></p> <ul style="list-style-type: none"> <li>▪ <b>DESKTOP APPLICATION POINT-OF SALES SYSTEM – WINDOWS UI FORM(S)</b> <i>FRONT-END APPLICATION &amp; OOP PROGRAMMING</i> features is a back-end system used by <b>customer service representative &amp; other employees</b> via the <i>Point-of-Sales computer</i> machine in the <i>Rental Agencies</i> to service customer's <b>CREDIT CARD MANAGEMENT</b>, or transactions required when servicing customers.</li> <li>▪ The following are features and functionalities required for this application with features such as: <ul style="list-style-type: none"> <li>○ <b>POS Credit Card Management Feature:</b> <i>POS Customer Credit Card Management features</i> such as <i>Credit Card Search &amp; Print, New Credit Card Registration, Credit Card Update, Credit Card Deletion, &amp; Credit Card Listing functionalities:</i> <ul style="list-style-type: none"> <li>- <b>Feature UI Form Requirements:</b> <i>Design &amp; programming</i> of required <i>User-Interface Forms &amp; GUI Controls</i> to support this feature.</li> <li>- <b>Feature Processing Requirements:</b> <i>Design &amp; programming</i> of required <i>Object-Oriented (OOP) Processing &amp; Logic</i> to support this feature.</li> </ul> </li> </ul> </li> <li>▪ This feature is designed only to be used by customer service agents and other employees using the <b>DESKTOP Two-Tiered Client/Server Application</b> in the <b>Rental Agencies</b>.</li> </ul>
<b>FEATURE #4B</b>	<p><b>FEATURE #4B – IN-SCOPE FOR CST4708 CLASS IN PROJECT #1</b></p> <p><b>FEATURE #4B – EZRental Rental Agency Point-of-Sales (POS) Credit Card Management System Back-end Database Design &amp; Implementation to support this feature:</b></p> <ul style="list-style-type: none"> <li>▪ <b>DATABASE SERVER BACK-END SYSTEM – BACK-END DATABASE DESIGN &amp; FEATURES</b> (Create the Tables &amp; Relationships, pre-populated tables, stored procedures, views, indexes etc..) to support this feature.</li> </ul>

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

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

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

Feature #	Feature Description
<b>FEATURE #7A</b>	<p><b>FEATURE #7A – IN-SCOPE FOR CST4708 CLASS IN PROJECT #2</b></p> <p><b>FEATURE #7A – EZRental CORPORATE OFFICES &amp; RENTAL AGENCIES CUSTOMER SERVICE AGENTS VIA THE INTRANET CORPORATE POINT-OF-SALES (POS) BACK-OFFICE CREDIT CARD MANAGEMENT SYSTEM WEB APPLICATION:</b></p> <ul style="list-style-type: none"> <li>▪ This <b>INTRANET (NOT THE PUBLIC INTERNET) Web Portal EZRentaCorp.com</b>, is a <b>Web-based Three-Tiered Client/Server physical &amp; Software Layered Development Architecture</b> used by <b>CORPORATE EMPLOYEES &amp; RENTAL AGENCIES EMPLOYEES</b> to <i>execute</i> <b>Enterprise Resource Planning Systems (ERP) Applications &amp; other Corporate Applications</b> online intranet via a <b>WEB BROWSER</b>.</li> <li>▪ <b>WEB APPLICATION CREDIT CARD MANAGEMENT SYSTEM ERP &amp; CORPORATE APPLICATION SYSTEM – WEB UI FORM/WEB PAGES FRONT-END &amp; OOP PROGRAMMING BACKEND Enterprise Resource Planning Systems (ERP) Credit Card Management System Application</b> used by <b>Corporate Customer Service &amp; Rental Agencies Customer Service Employees</b> to perform <b>Credit Card Management</b> processing via the <b>CORPORATE INTRANET PORTAL</b> and <b>NOT</b> the Desktop Two-Tiered Client/Server Application of the Rental Agencies.</li> <li>▪ The following are features and functionality that are required for this <b>Credit Card Management System Application INTRANET WEB APPLICATION</b>: <ul style="list-style-type: none"> <li>○ <b>POS Credit Card Management Feature:</b> POS Customer Credit Card Management features such as <i>Credit Card Search &amp; Print, New Credit Card Registration, Credit Card Update, Credit Card Deletion, &amp; Credit Card Listing</i> functionalities: <ul style="list-style-type: none"> <li>- <b>Feature WEB UI Requirements:</b> <i>Design &amp; programming</i> of required <i>User-Interface Forms &amp; GUI Controls</i> to support this feature.</li> <li>- <b>Feature Processing Requirements:</b> <i>Design &amp; programming</i> of required <i>Object-Oriented (OOP) Processing &amp; Logic</i> to support this feature.</li> </ul> </li> </ul> </li> <li>▪ This feature is designed only to be used by customer service agents in and other employees in the Corporate Offices or Rental Agencies using the <b>WEB Three-Tiered Client/Server Application</b> in the <b>Rental Agencies</b>.</li> </ul>

<b>FEATURE #7B</b>	<b>FEATURE #7B – IN-SCOPE FOR CST4708 CLASS IN PROJECT #1 (Note this requirement is the same as Requirement #4B)</b>  <b>FEATURE #7B – EZRental Rental Agency Point-of-Sales (POS) Credit Card Management System Back-end Database Design &amp; Implementation to support this feature:</b>  <ul style="list-style-type: none"><li>• <b>DATABASE SERVER BACK-END SYSTEM – BACK-END DATABASE DESIGN &amp; FEATURES</b> (Create the Tables &amp; Relationships, pre-populated tables, stored procedures, views, indexes etc.,) to support this feature.</li></ul>
--------------------	---

Feature #	Feature Description
<b>FEATURE #8A</b>	<p><b>FEATURE #8A – EZRental INTERNAL CORPORATE EMPLOYEE &amp; RENTAL AGENCIES EMPLOYEES INTRANET CORPORATE BUSINESS APPLICATIONS WEB PORTAL TO IMPLEMENT REMAINING ERP SYSTEM APPLICATIONS &amp; POINT-OF-SALES APPLICATION USING A WEB INTRANET APPLICATION:</b></p> <ul style="list-style-type: none"> <li>▪ This <b>INTRANET (NOT THE PUBLIC INTERNET) Web Portal</b> <a href="http://EZRentalHub.com">EZRentalHub.com</a>, is a <b>Web-based Three-Tiered Client/Server physical &amp; Software Layered Development Architecture</b> used by <b>CORPORATE &amp; OTHER EMPLOYEES</b> to <i>execute</i> <b>Enterprise Resource Planning Systems (ERP) Applications &amp; other Corporate Applications</b> online intranet via a <b>BROWSER</b>.</li> <li>▪ <b>BROWSER/INTRANET WEB ERP &amp; CORPORATE APPLICATION SYSTEM – WEB UI FORM(S) FRONT-END APPLICATION &amp; OOP PROGRAMMING Enterprise Resource Planning Systems (ERP) Applications &amp; other Corporate Applications Features</b> used by <b>Corporate Employees</b> via the <b>CORPORATE INTRANET PORTAL</b>.</li> <li>▪ The following are features and functionality that are required for this <b>INTRANET WEB APPLICATION</b>: <ul style="list-style-type: none"> <li>○ <b>Corporate Business Application Intranet Web Portal Features:</b> <ul style="list-style-type: none"> <li>- <b>Intranet Web Enterprise Resource Planning Systems (ERP) Portal Feature 7A-1</b> – Provides access to Enterprise Resource Planning Systems (ERP) Applications such as: <i>Vehicle Inventory Management System, Customer Relationship Management (CRM), Human Resource Management System, &amp; Finance &amp; Operations System, Marketing System, Customer &amp; Field Service System etc.</i></li> <li>- <b>Web EZRental Point-of-Sales Corporate Management Feature 7A-2</b> – Allows Employees to <i>manage &amp; execute</i> Point-of-Sales (POS) transaction via the <b>Intranet Web Portal</b> such as: <i>Search Customer Profile Information, Customer Account Management, Customer Registration, Customer Update, Customer Delete, &amp; Customer Listing functionalities, Manage &amp; Make Reservations of a Vehicle, Manage an existing Rental, etc.</i></li> </ul> </li> <li>○ Programming includes: <ul style="list-style-type: none"> <li>- <b>Feature UI Form Requirements:</b> <i>Design &amp; programming</i> of required <b>INTRANET WEB User-Interface Forms &amp; GUI Controls</b> to support this feature.</li> <li>- <b>Feature Processing Requirements:</b> <i>Design &amp; programming</i> of required <b>INTRANET WEB TECHNOLOGY, Object-Oriented (OOP) Processing &amp; Logic PROCESSING</b> to support this feature.</li> </ul> </li> </ul> </li> <li>▪ This feature is designed only to be used by <b>CORPORATE EMPLOYEES</b> in the Corporate Offices &amp; <b>RENTAL AGENCIES EMPLOYEES</b> via the <b>INTRANET</b> using the <b>Web Three-Tiered Client/Server Application</b>.</li> </ul>

**FEATURE #8B**

**FEATURE #8B – 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
<b>FEATURE #9A</b>	<p><b>FEATURE #9A – EZRental EXTERNAL CUSTOMER SELF-SERVICE INTERNET CUSTOMER FACING POINT-OF-SALES (POS) WEB PORTAL APPLICATION:</b></p> <ul style="list-style-type: none"> <li>- This Web Portal EZRental.com, is a <b>Web-based Three-Tiered Client/Server physical &amp; Software Development Architecture</b> used by <b>CUSTOMERS</b> to <i>manage &amp; make reservations</i> online via a <b>BROWSER</b>.</li> <li>- <b>BROWSER/WEB CUSTOMER POINT-OF SALES SYSTEM – WEB UI FORM(S) FRONT-END APPLICATION &amp; OOP PROGRAMMING Point-of-Sales (POS) Management Features</b> used by <b>customers</b> via the <b>INTERNET Point-of-Sales PORTAL</b> via their computers/laptop/tables/Mobile to <b>MAKE RESERVATIONS ONLINE &amp; MANAGE THEIR RENTAL</b>.</li> <li>- The following are features and functionality that are required for this <b>WEB APPLICATION</b> feature: <ul style="list-style-type: none"> <li>o <b>POS Reservation &amp; Management Features:</b> <ul style="list-style-type: none"> <li>- <b>Web POS Authentication System Feature 8A-1</b> – Proper <b>security and authentication</b> 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.</li> <li>- <b>Web POS Customer Self-Service Management Feature 8A-2</b> – Allows POS Customer (Retail Customer &amp; Corporate Customer) Self-Service Management of their account such as: <i>Customer Profile Information, Customer Account &amp; Login Registration, Customer Update Profile, Customer Delete Profile, &amp; Customer Listing functionalities such as listing of Reservations &amp; Rental History etc.</i></li> <li>- <b>Web POS Customer Self-Service Point-of-Sales Management Feature 8A-3</b> – Allows POS Customer (Retail Customer &amp; 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></li> <li>- <b>Web POS User Account Management Feature 8A-4</b> – Allows POS Customer (Retail Customer &amp; Corporate Customer) Self-Service features to enable customer to manage its <b>User Account</b> and perform the following operations: <i>Reset Username &amp; Reset Password</i>.</li> </ul> </li> <li>o Programming includes: <ul style="list-style-type: none"> <li>- <b>Feature UI Form Requirements:</b> <i>Design &amp; programming</i> of required <b>WEB User-Interface Forms &amp; GUI Controls</b> to support this feature.</li> <li>- <b>Feature Processing Requirements:</b> <i>Design &amp; programming</i> of required <b>WEB TECHNOLOGY, Object-Oriented (OOP) Processing &amp; Logic PROCESSING</b> to support this feature.</li> </ul> </li> </ul> </li> <li>- This feature is designed only to be used by <b>CUSTOMERS</b> via the <b>INTERNET</b> using the <b>Web Three-Tiered Client/Server Application</b> from their <b>Personal Computer/Mobile Devices via the INTERNET</b>.</li> </ul>

Feature #	Feature Description
<b>FEATURE #9B</b>	<p><b>FEATURE #9B – EZRental Customer Self-Service internet Customer facing Point-of-Sales (POS) WEB PORTAL System Back-end Database Design &amp; Implementation to support this feature:</b></p> <ul style="list-style-type: none"> <li>▪ <b>DATABASE SERVER BACK-END SYSTEM – BACK-END DATABASE DESIGN &amp; FEATURES</b> (Create the Tables &amp; Relationships, pre-populated tables, stored procedures, views, indexes etc.,) to support this feature.</li> </ul>

Feature #	Feature Description
<b>FEATURE #10A</b>	<p><b>FEATURE #10A – EZRental Customer Point-of-Sales (POS) System CUSTOMER FACING WEB PORTAL SECURITY LOGIN AUTHENTICATION SYSTEM:</b></p> <ul style="list-style-type: none"> <li>▪ Proper <i>security and authentication</i> must be implemented to make sure only authorized <b>customers</b> can access their <b>Self-Service-Point-Of-Sales Web Portal systems</b> when they <b>login</b> into the <b>Web Three-Tiered Customer Client/Server Application</b> via the <b>INTERNET</b>.</li> <li>▪ <b>CUSTOMER SELF-SERVICE POINT-OF SALES SYSTEM – WEB UI FORM(S) FRONT-END APPLICATION &amp; OOP PROGRAMMING</b> for <b>Customer Login Authentication features</b> used by <b>customer service representative &amp; IT Administrator employees</b> to service <b>CUSTOMER LOGIN AUTHENTICATION SYSTEM</b>.</li> <li>▪ The following are features and functionality that are required for this application such as: <ul style="list-style-type: none"> <li>○ <b>Customer Self-Service Web Portal Security Login Authentication System.</b> Self-Service Web Portal Login Authentication Access for Customer features such as: <ul style="list-style-type: none"> <li>- <b>Customer Authentication Feature 9A-1</b> – To have access to their Self-Service Web Portal Application, a customer must provide a <b>username &amp; password</b>. This feature is required to be <u>designed</u> &amp; <u>programmed</u> into the application.</li> <li>- <b>Customer Authentication Feature 9A-2 – Design &amp; programming</b> of required <b>Web User-Interface Forms &amp; GUI Controls</b> to support the <b>Web Portal Authentication System</b> feature!</li> </ul> </li> <li>○ Programming includes: <ul style="list-style-type: none"> <li>- <b>Feature Web UI Form Requirements:</b> <i>Design &amp; programming</i> of required <b>User-Interface Forms &amp; GUI Controls</b> to support this feature.</li> <li>- <b>Feature Web Processing Requirements:</b> <i>Design &amp; programming</i> of required <b>Object-Oriented (OOP) Processing &amp; Logic</b> to support this feature.</li> </ul> </li> </ul> </li> <li>▪ This feature is designed only to be used by customers wishing access to their <b>Self-Service Web Portal Three-Tiered Client/Server Application</b> via the <b>INTERNET</b>.</li> </ul>
<b>FEATURE #10B</b>	<p><b>FEATURE #10B – EZRental Customer Point-of-Sales (POS) System CUSTOMER FACING WEB PORTAL SECURITY LOGIN AUTHENTICATION SYSTEM Back-end Database Design &amp; Implementation to support this feature:</b></p> <ul style="list-style-type: none"> <li>▪ <b>DATABASE SERVER BACK-END SYSTEM – BACK-END DATABASE DESIGN &amp; FEATURES</b> (Create the Tables &amp; Relationships, pre-populated tables, stored procedures, views, indexes etc.,) to support the <b>CUSTOMER</b> facing authentication features.</li> </ul>

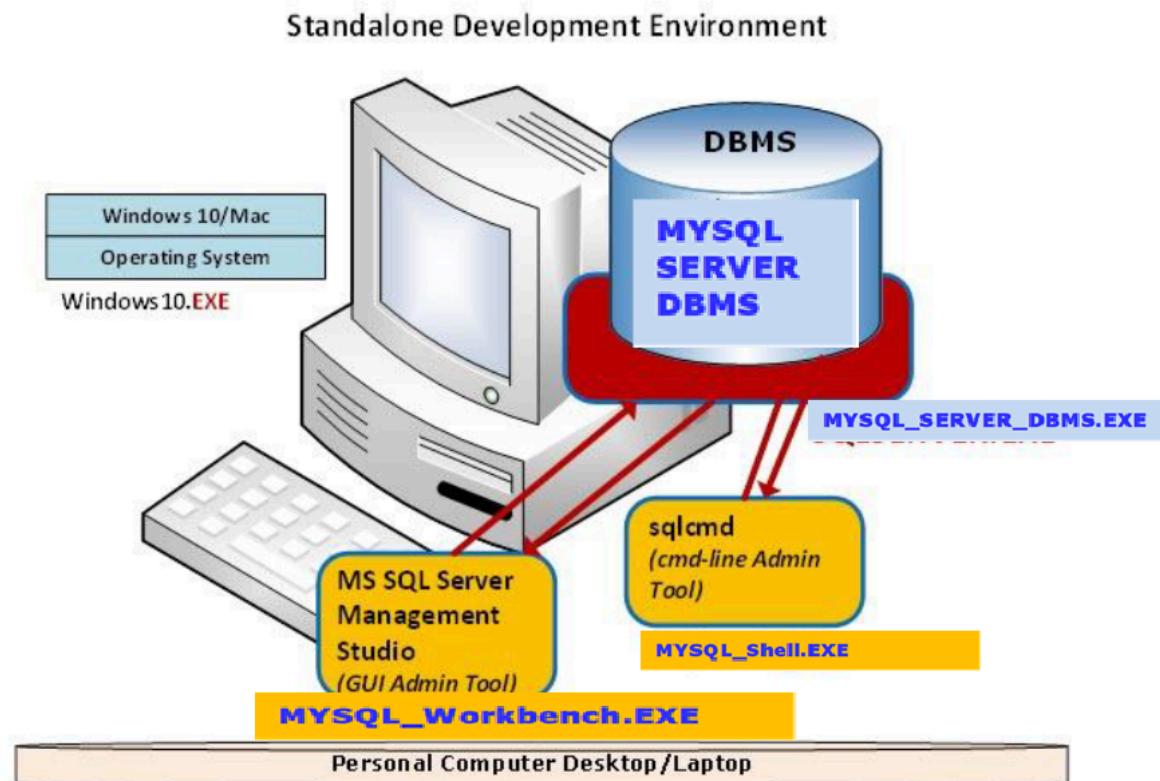
Feature #	Feature Description
<b>FEATURE #11A</b>	<p><b>FEATURE #11A – EZRental INTERNAL CORPORATE EMPLOYEE &amp; RENTAL AGENCIES EMPLOYEES INTRANET BACK-OFFICE VEHICLE TRANSPORT MANAGEMENT SYSTEM WEB PORTAL:</b></p> <ul style="list-style-type: none"> <li>▪ This <b>INTRANET Web Portal</b> <a href="#">EZRentalHub.com</a>, is a <b>Web-based Three-Tiered Client/Server physical &amp; Software Development Architecture</b> used by <b>CORPORATE &amp; AGENCY EMPLOYEES</b> to <i>manage</i> <b>Transportation of Vehicles by Employee Drivers to and from Rental Agencies, Vehicle Distribution Centers, and other Locations</b> via a <b>BROWSER</b>.</li> <li>▪ <b>WEB TRANSPORT MANAGEMENT SYSTEM APPLICATION – WEB UI FORM(S) FRONT-END APPLICATION &amp; OOP PROGRAMMING</b> <b>Transport Management Features</b> used by <b>Vehicle Transportation Managers &amp; Drivers Employees</b> to handle the day-to-day vehicle transportation process via the <b>CORPORATE INTRANET PORTAL</b>.</li> <li>▪ The following are features and functionality that are required for this <b>INTRANET Transport Management WEB APPLICATION</b>: <ul style="list-style-type: none"> <li>○ <b>Corporate Vehicle Transport Application Intranet Web Portal Features:</b> <ul style="list-style-type: none"> <li>- <b>Transport Scheduling Feature</b> – handle the day-to-day creating &amp; scheduling of a pic-up &amp; delivery (Any vehicle type) such as: <i>Creation of NEW Vehicle Transport Request, Vehicle Pick-up, Vehicle Drop-off &amp; Vehicle Transport Status etc.</i></li> </ul> </li> <li>○ Programming includes: <ul style="list-style-type: none"> <li>- <b>Feature UI Form Requirements:</b> <i>Design &amp; programming</i> of required <b>INTRANET WEB User-Interface Forms &amp; GUI Controls</b> to support this feature.</li> <li>- <b>Feature Processing Requirements:</b> <i>Design &amp; programming</i> of required <b>INTRANET WEB TECHNOLOGY, Object-Oriented (OOP) Processing &amp; Logic PROCESSING</b> to support this feature.</li> </ul> </li> </ul> </li> <li>▪ This feature is designed only to be used by <b>CORPORATE EMPLOYEES</b> in the Corporate Offices &amp; <b>RENTAL AGENCIES EMPLOYEES</b> via the <b>INTRANET</b> using the <b>Web Three-Tiered Client/Server Application</b>.</li> </ul>

## Database Management System Development Environment & Physical Architecture

This section outlines the database system used for the POS at EZRental Inc., detailing its usage and architectural structure.

The chosen **Database Management System (DBMS)** for this project is **MS SQL Server Community Edition**, as it is the standard DBMS used by EZRental Inc. The goal is to set up and configure MS SQL Server Community Edition on a personal computer to establish a proper development environment.

The setup process includes installing **MS SQL Server Community Edition**, which comes with **SQLCMD**, a command-line tool for database administration. Additionally, **MS SQL Server Management Studio (SSMS)** will be installed as a graphical interface to efficiently manage and interact with the database. This setup will provide a robust environment for database development, allowing for structured data management, query execution, and system administration.



## Project Roles & Responsibilities

This section outlines the roles and responsibilities of the team involved in developing the POS database system.

The **Business/Database Analyst**, hired by Mr. Rodriguez, is responsible for assembling a qualified database development team. The following table provides a breakdown of each role within the team and the individuals responsible for carrying out these tasks:

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

### Summary of DBMS Development Roles and Responsibilities

Mr. Rodriguez has appointed a **Business/Database Analyst** to form a skilled database development team for the POS system. The team consists of key roles, each with specific responsibilities:

- The **Business/Database Analyst** ensures the database meets business requirements.
- The **Database Administrator (DBA)** manages performance, security, and backups.

- The **Database Developer** designs the structure and writes SQL queries.
- The **System Architect** ensures seamless integration with the POS system.
- The **Application Developer** connects the database to the POS application.
- The **QA Tester** verifies functionality and ensures the system meets requirements.

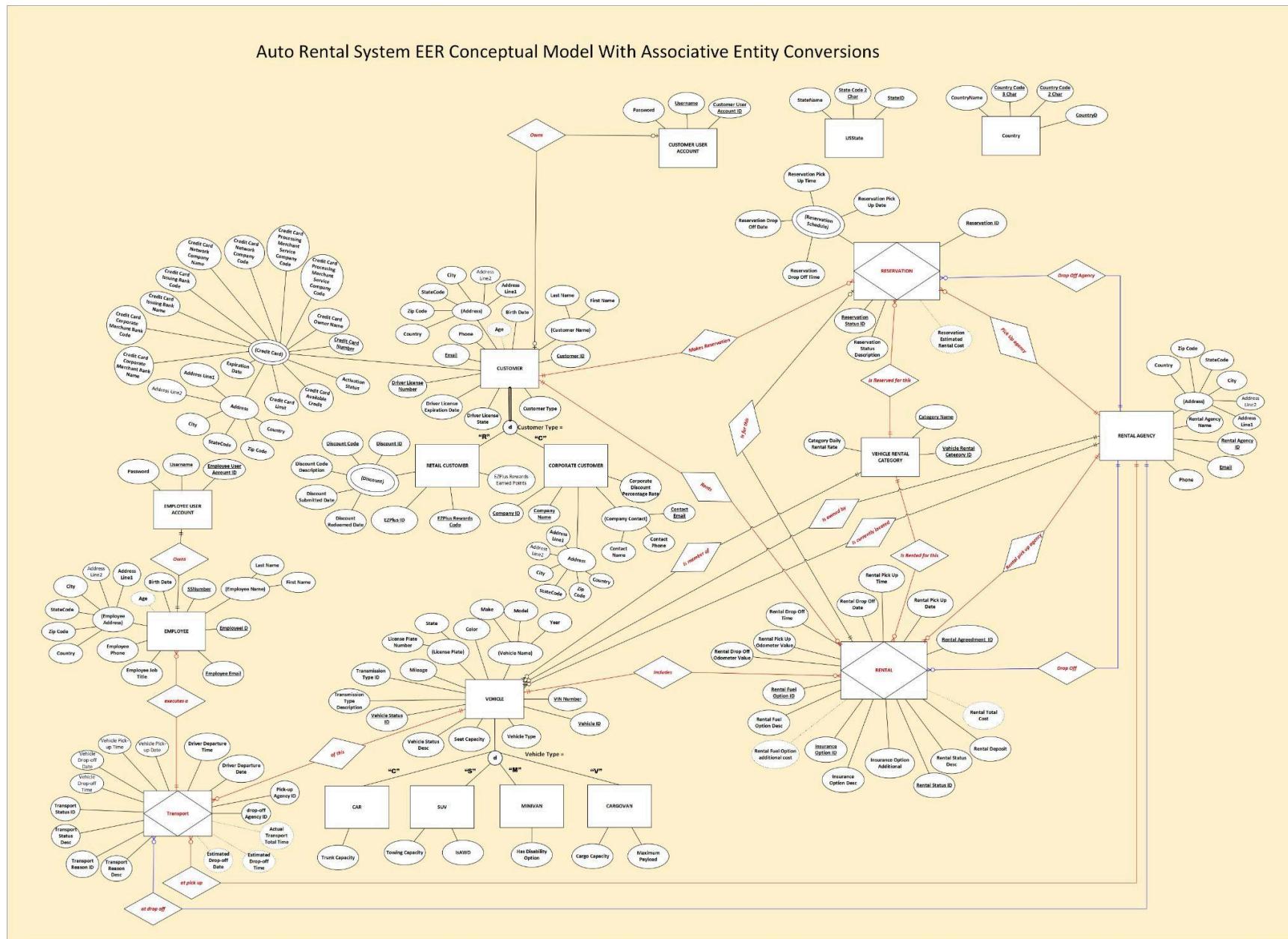
Each team member plays a vital role in the successful creation and maintenance of the database system.

## Database Design Deliverable #2 – ER/EER Conceptual Model Diagram

This deliverable represents the second phase of the database design process, following the **waterfall project management methodology** during the analysis stage. It includes conceptual diagrams created by the **Database Analyst/Architect**, based on the business requirements of the application.

The **Enhanced Entity-Relationship (EER) Model** is designed using **associative entities**, ensuring that complex relationships between data elements are accurately represented. This model is structured using **CHEN notation**, which provides a clear and standardized visual representation of entities, their attributes, and relationships.

The EER model serves as a blueprint for understanding how different entities within the POS system interact, allowing for efficient database development and ensuring the system meets business requirements.

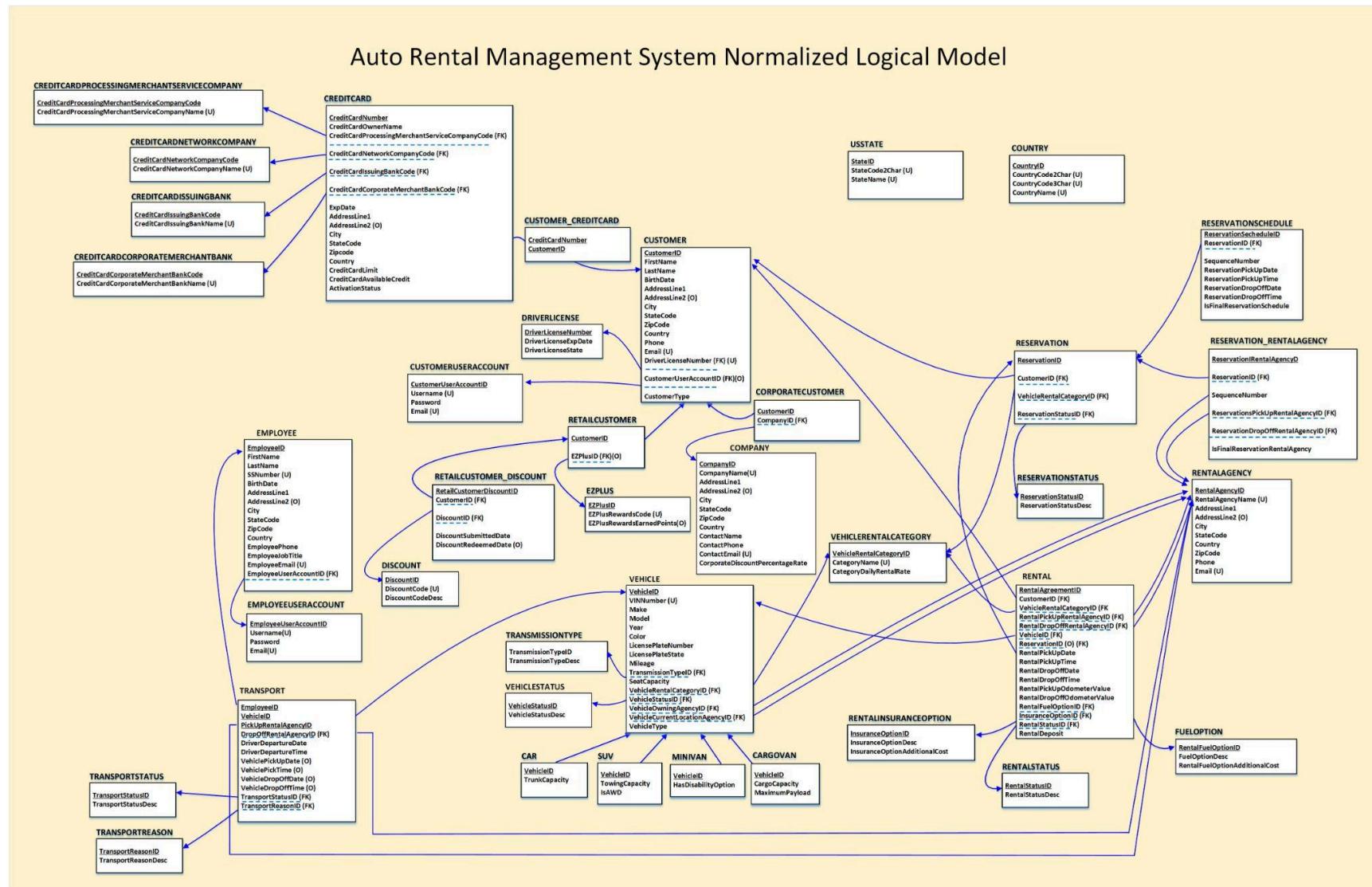


### **Database Design Deliverable #3 – Normalized Logical Model Diagram**

This deliverable represents the third stage of the database design process, building upon the **EER conceptual model** to introduce a **Normalized Logical Model** for the **Auto Management System**.

The **Normalized Logical Model** is derived from the **EER diagram** created in the previous phase. It refines and structures the database by applying **normalization principles**, reducing redundancy and improving data integrity. This ensures that the database is efficient, scalable, and optimized for performance.

The architect has designed this model to include **39 logical tables**, each representing distinct entities and relationships necessary for the system's functionality. This structured approach lays the foundation for the **physical database implementation**, ensuring that the system aligns with business and technical requirements.



The diagram illustrates how **credit card information** is structured within the auto rental system, ensuring that transactions and payments are properly managed. The model follows a **normalized database design**, which improves **data integrity, reduces redundancy, and ensures efficient data retrieval**.

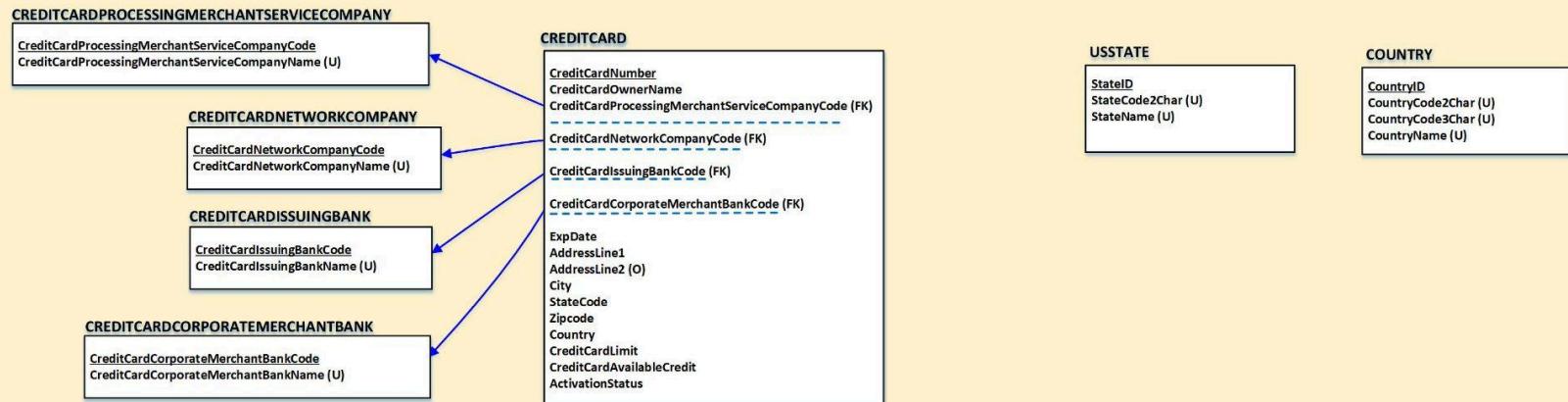
### Key Components of the Diagram

1. **CREDITCARD Table** – Stores information related to customer credit cards, including:
  - Credit card number, owner name, expiration date, billing address, and financial details.
  - Foreign keys (FK) linking to other tables to normalize data.
2. **Related Financial Institutions** – Separate tables for:
  - **Credit Card Processing Merchant Service Company**
  - **Credit Card Network Company**
  - **Credit Card Issuing Bank**
  - **Credit Card Corporate Merchant Bank**
  - These entities are connected to the **CREDITCARD** table via **foreign keys**, ensuring that each transaction is linked to valid financial institutions.
3. **USSTATE and COUNTRY Tables** – Used for:
  - **Geographic normalization** (storing states and countries separately).
  - Avoiding data redundancy and ensuring **consistent location-based information**.

### Why is this Model Important?

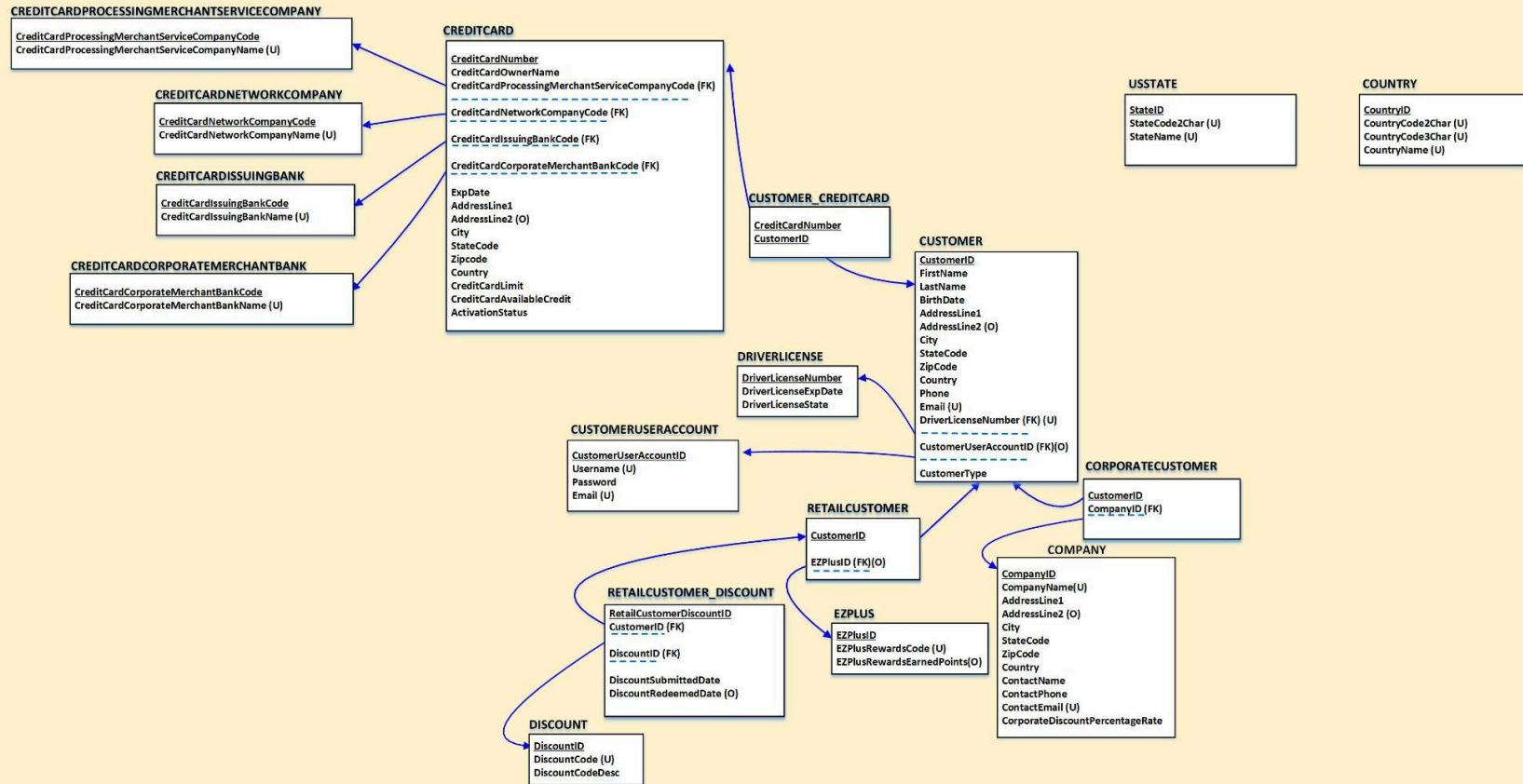
- **Improves efficiency** by structuring financial data in a way that minimizes duplication.
- **Enhances security** by linking financial transactions to verified institutions.
- **Supports scalability**, making it easier to expand the system with more financial institutions or regions.
- **Ensures consistency** in handling payments within the auto rental system.

## Auto Rental Management System Normalized Logical Model Proof-of-Concept (POC) Development & Deployment



# Auto Rental Management System Normalized Logical Model

## LIMITED PILOT Development & Deployment



**Database Design Deliverable #4 – Physical Model Data Dictionary**

The Physical Model Data Dictionary is the Second Component of the Waterfall Methodology design phase physical model design. The Physical Model Date Dictionary - Is a TABULAR LISTING (Column Name, Column Database Data Type, Column Constraints & Column Description) for all the columns of every table in the Normalized Logical Model Diagram.

CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY								
Column Num.	Attribute/Column Name	Generic Data Type Name	MySQL SERVER Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/ purpose	
1.	<b>CreditCardProcessingMerchantServiceCompanyCode</b>	Number	TINYINT	Yes	20	<b>PRIMARY KEY CHECK( CreditCard Processing MerchantService CompanyCode between 1 and 20)</b>	<b>Unique Identifier or Primary Key</b> for this table. This <b>Primary Key HAS Business Meaning</b> .	
2.	<b>CreditCardProcessingMerchantServiceCompany Name (U)</b>	String	VARCHAR(60)	Yes	25	<b>NOT NULL</b>	Stores the <b>unique</b> name of the Credit Card Processing Merchant Service Company.	

CREDITCARDNETWORKCOMPANY								
Colu mn Num .	Attribute/Column Name	Generic Data Type Name	MySQL SERVER Data Type Name	Is it Required?	Length/Siz e /Format	Constraints	Description/ purpose	

1.	<b>CreditCardNetwork CompanyCode</b>	Number	TINYINT		Yes	25	<b>PRIMARY KEY CHECK( CreditCard Network CompanyCode between 1 and 25)</b>	<b>Unique Identifier or Primary Key</b> for this table. This <b>Primary Key HAS Business Meaning</b> .
2.	<b>CreditCardNetwork CompanyName (U)</b>	String	VARCHAR(25)		Yes	50	<b>NOT NULL UNIQUE</b>	Stores the <b>unique</b> name of the Credit Card Network Company.

CREDITCARDISSUINGBANK								
Column Num.	Attribute/Column Name	Generic Data Type Name	MySQL Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/ purpose	
1.	<b>CreditCardIssuingBankCode</b>	Number	TINYINT	Yes	25	<b>PRIMARY KEY CHECK( CreditCard IssuingBankCode between 1 and 25)</b>	<b>Unique Identifier or Primary Key</b> for this table. This <b>Primary Key HAS Business Meaning</b> .	
2.	<b>CreditCardIssuingBankName (U)</b>	String	VARCHAR(50)	Yes	50	<b>UNIQUE NOT NULL</b>	Stores the <b>unique</b> name of the Credit Card Processing Merchant Service Company.	

CREDITCARDCORPORATEMERCHANTBANK								
Column Num.	Attribute/Column Name	Generic Data Type Name	MySQL SERVER Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/ purpose	
1.	<b>CreditCard Corporate MerchantBankCode</b>	Number	TINYINT	Yes	10	<b>PRIMARY KEY CHECK( CreditCard CorporateMerchantBank Code between 1 and 25)</b>	<b>Unique Identifier or Primary Key</b> for this table. This <b>Primary Key HAS Business Meaning</b> .	
2.	<b>CreditCard Corporate MerchantB</b>	String	VARCHAR(30)	Yes	30		Stores the <b>unique</b> name of the Credit Card Corporate Merchant Bank.	

	<b>ankName (U)</b>							
--	------------------------	--	--	--	--	--	--	--

<b>CREDITCARD</b>							
Column Num.	Attribute/Column Name	Generic Data Type Name	MySQL SERVER Data Type Name	Is it Required?	Length /Size /Format	Constraints	Description/ purpose
1.	<b>CreditCardNumber</b>	<b>String</b>	<b>VARCHAR(16 )</b>	<b>Yes</b>	<b>16</b>	<b>PRIMARY KEY</b>	<b>Unique Identifier or Primary Key</b> for the CreditCard table. This <b>Primary Key HAS Business Meaning</b> .
2.	<b>CreditCardOwner Name</b>	<b>String</b>	<b>VARCHAR(10 0)</b>	<b>Yes</b>	<b>100</b>	<b>NOT NULL</b>	Credit Card Owner first, last and middle initial if included in name
3.	<b>CreditCardProcessingMerchantServiceCompanyCode (FK)</b>	<b>Number</b>	<b>TINYINT</b>	<b>Yes</b>	<b>20</b>	<b>PRIMARY KEY CHECK</b> (CreditCard ProcessingMerchantServiceCompanyCode <b>between 1 and 20</b> ) <b>NOT NULL</b>	Column stores the Credit Card Processing Merchant Service Company unique code. <b>Foreign Key</b> to <i>CreditCardProcessingMerchantServiceCompany</i> Table.
4.	<b>CreditCardNetworkCompanyCode (FK)</b>	<b>Number</b>	<b>TINYINT</b>	<b>Yes</b>	<b>25</b>	<b>PRIMARY KEY CHECK</b> (CreditCard NetworkCompanyCode <b>between 1 and 25</b> )	Stores the Credit Card Network Company unique code. <b>Foreign Key</b> to <i>CreditCardNetworkCompany</i> Table.

5.	<b>CreditCardIssuingBankCode (FK)</b>	<b>Number</b>	<b>TINYINT</b>	<b>Yes</b>	<b>25</b>	<b>PRIMARY KEY CHECK(CreditCardReadingCompanyCode between 1 and 10)</b>	Stores the Credit Card Issuing Bank unique code. <b>Foreign Key</b> to <i>CreditCardIssuingBank</i> Table.
6.	<b>CreditCardCorporateMerchantBankCode (FK)</b>	<b>Number</b>	<b>TINYINT</b>	<b>Yes</b>	<b>10</b>	<b>PRIMARY KEY CHECK(CreditCardCorporateMerchantCompanyCode between 1 and 10)</b>	Stores the Credit Card Corporate Merchant Bank unique code. <b>Foreign Key</b> to <i>CreditCardCorporateMerchantBank</i> Table.
7.	<b>ExpDate</b>	<b>Date</b>	<b>DATE</b>	<b>Yes</b>	<b>MM-DD-YYYY</b>	<b>NOT NULL</b>	The date the credit card expires.
8.	<b>AddressLine1</b>	<b>String</b>	<b>VARCHAR(50)</b>	<b>Yes</b>	<b>50</b>	<b>NOT NULL</b>	Stores house/building number & street (part 1 of address).
9.	<b>AddressLine2 (O)</b>	<b>String</b>	<b>VARCHAR(50)</b>	<b>No</b>	<b>50</b>	<b>NULL</b>	Optional value that stores remaining part of address such as apartment number, or other address information.
10.	<b>City</b>	<b>String</b>	<b>VARCHAR(50)</b>	<b>Yes</b>	<b>50</b>	<b>NOT NULL</b>	Stores the city name
11.	<b>StateCode</b>	<b>Characters</b>	<b>CHAR(2)</b>	<b>Yes</b>	<b>2</b>	<b>NOT NULL</b>	Stores the U.S. State 2-character code. E.g., NY, NJ, CT, etc.
12.	<b>Zipcode</b>	<b>String</b>	<b>VARCHAR(10)</b>	<b>Yes</b>	<b>10</b>	<b>NOT NULL</b>	Stores US Zip Code/Postal Code. Support format: xxxx-xxxx.
13.	<b>Country</b>	<b>String</b>	<b>VARCHAR(100)</b>	<b>Yes</b>	<b>100</b>	<b>NOT NULL</b>	Stores the name of the country.
14.	<b>CreditCardLimit</b>	<b>Number</b>	<b>DECIMAL(8,2)</b>	<b>Yes</b>	<b>X = 8 Y = 2</b>	<b>NOT NULL</b>	The maximum amount of dollars that can be charged to the credit card. Assumes that the credit card has the full limit available. The format is <b>DECIMAL(X,Y)</b> , where X = The total number of digits and Y = total

							number of digits to the right of the decimal point. We assume the maximum number that can be stored is 999999.99 for a maximum limit amount \$999,999.99, since we don't expect a customer to have a credit limit of \$1 Million dollars, we cap it at \$999,999.99.
15.	<b>CreditCardAvailableCredit</b>	<b>Number</b>	<b>DECIMAL(8,2)</b>	<b>Yes</b>	<b>X = 8 Y = 2</b>	<b>NOT NULL</b>	Stores the current remaining credit available for charging.
16.	<b>ActivationStatus</b>	<b>Boolean</b>	<b>BIT</b>	<b>Yes</b>	<b>1</b>	<b>NOT NULL</b>	Stores a Boolean value indicating <b>True</b> if credit card is active or <b>False</b> otherwise. The <b>MS SQL SERVER DBMS</b> has a Data Type named <b>BIT</b> that will store <b>1</b> to represent <b>True</b> and <b>0</b> to represent <b>False</b> .

## #6:

USSTATE							
Column Number	Attribute/Column Name	Generic Data Type Name	MySQL SERVER Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/ purpose

	<b>StateID</b>	Number	<b>TINYINT</b>	<b>Yes</b>	75	<b>PRIMARY KEY</b> <b>CHECK</b> (CreditCardStateID between 1 and 75)	<b>Unique Identifier or Primary Key</b> for this table. This <b>Primary Key HAS Business Meaning</b> .
	<b>StateCode 2Char (U)</b>	Characters	<b>CHAR(2)</b>	<b>Yes</b>	2	NOT NULL, UNIQUE	Stores the U.S. State 2-character code. E.g., NY, NJ, CT etc.
	<b>StateName (U)</b>	String	<b>VARCHAR(50)</b>	<b>Yes</b>	50	NOT NULL, UNIQUE	Stores the unique full name of the U.S. State.

#7:

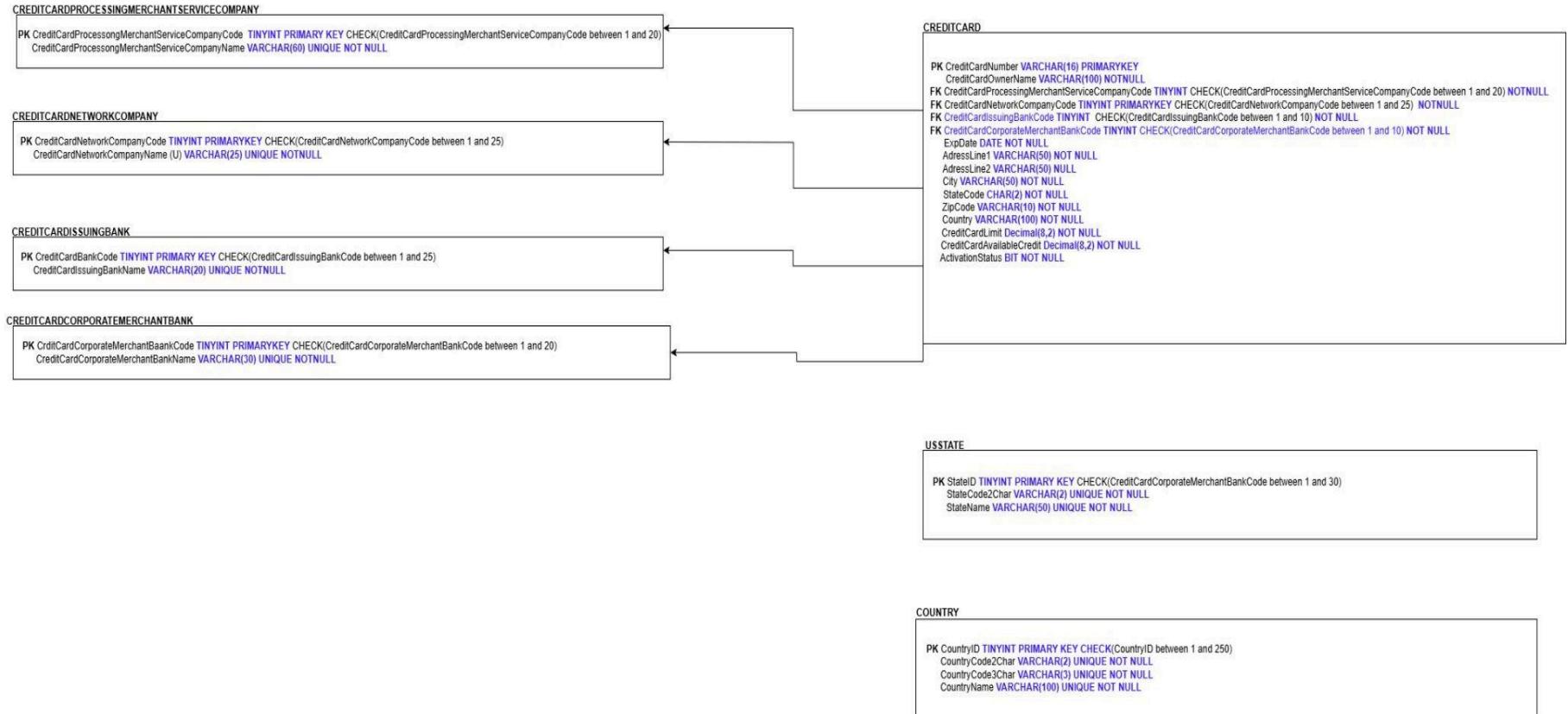
COUNTRY								
Colu mn Num.	Attribute/Column Name		Generic Data Type Name	<b>MySQL SERVER</b> Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/ purpose
1.	<b>CountryID</b>		<b>Number</b>	<b>TINYINT</b>	<b>Yes</b>	<b>250</b>	<b>PRIMARY KEY CHECK</b> (Cred itCardCountry IDbetween 1 and 75)	<b>Unique Identifier or Primary Key</b> for this table. This <b>Primary Key HAS Business Meaning</b> .
2.	<b>CountryCode2Char (U)</b>		<b>Charac ters</b>	<b>CHAR(2)</b>	<b>Yes</b>	<b>2</b>	NOT NULL, UNIQUE	Stores the country 2-character code. E.g., US, GB, etc.
3.	<b>CountryCode3Char (U)</b>		<b>Charac ters</b>	<b>CHAR(3)</b>	<b>Yes</b>	<b>3</b>	NOT NULL, UNIQUE	Stores the country 3-character code. E.g., USA, GBR, etc.

4.	<b>CountryName (U)</b>		<b>String</b>	<b>VARCHAR(100)</b>	<b>Yes</b>	<b>100</b>	<b>NOT NULL, UNIQUE</b>	Stores the unique full name of the country.
----	------------------------	--	---------------	---------------------	------------	------------	-----------------------------	---

**Database Design Deliverable #5 – Physical Model Schema Design Diagram**

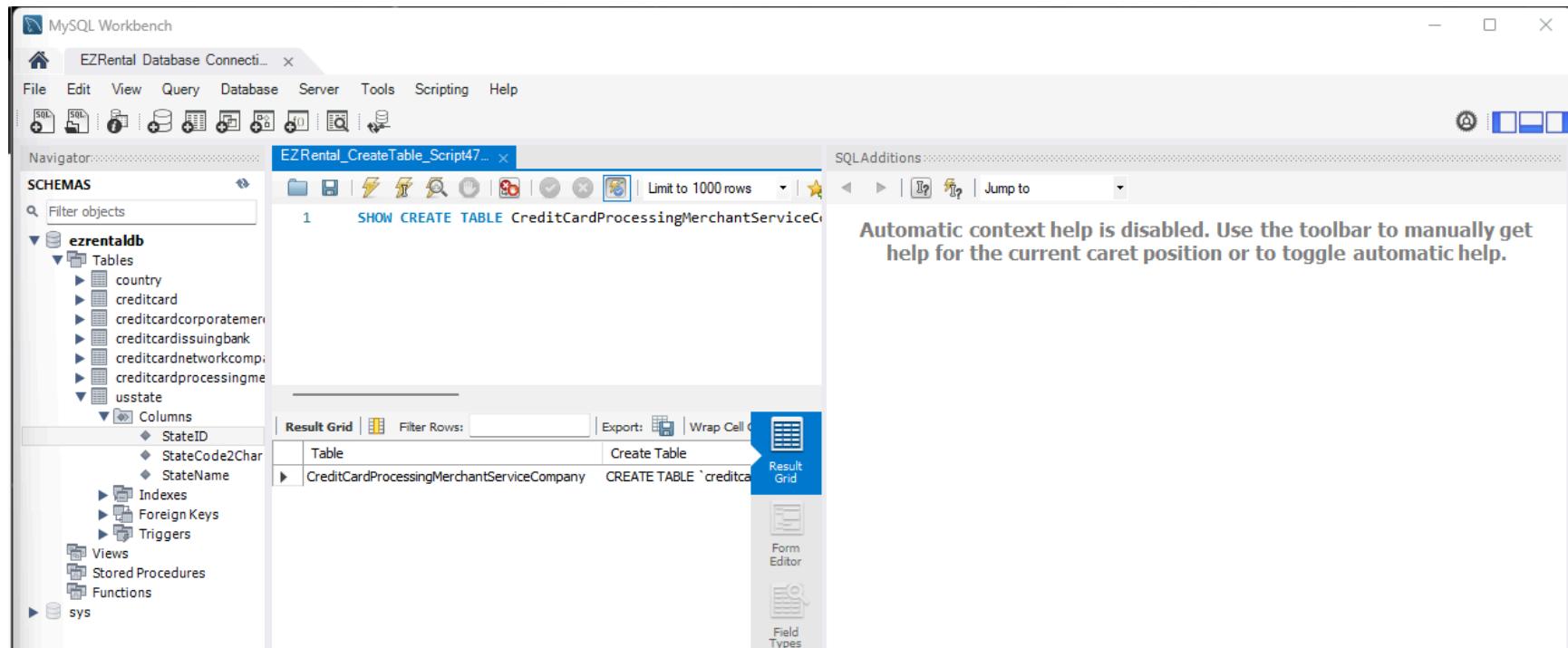
The Physical Model Schema Design Diagram is the DESIGN of what the PHYSICAL Database or DBMS tables and their relationships will look like. This diagram is created by combining the Normalized Logical Model Diagram with the Physical Model Data Dictionary to create a NEW DIAGRAM called the Physical Model Schema Design Diagram.

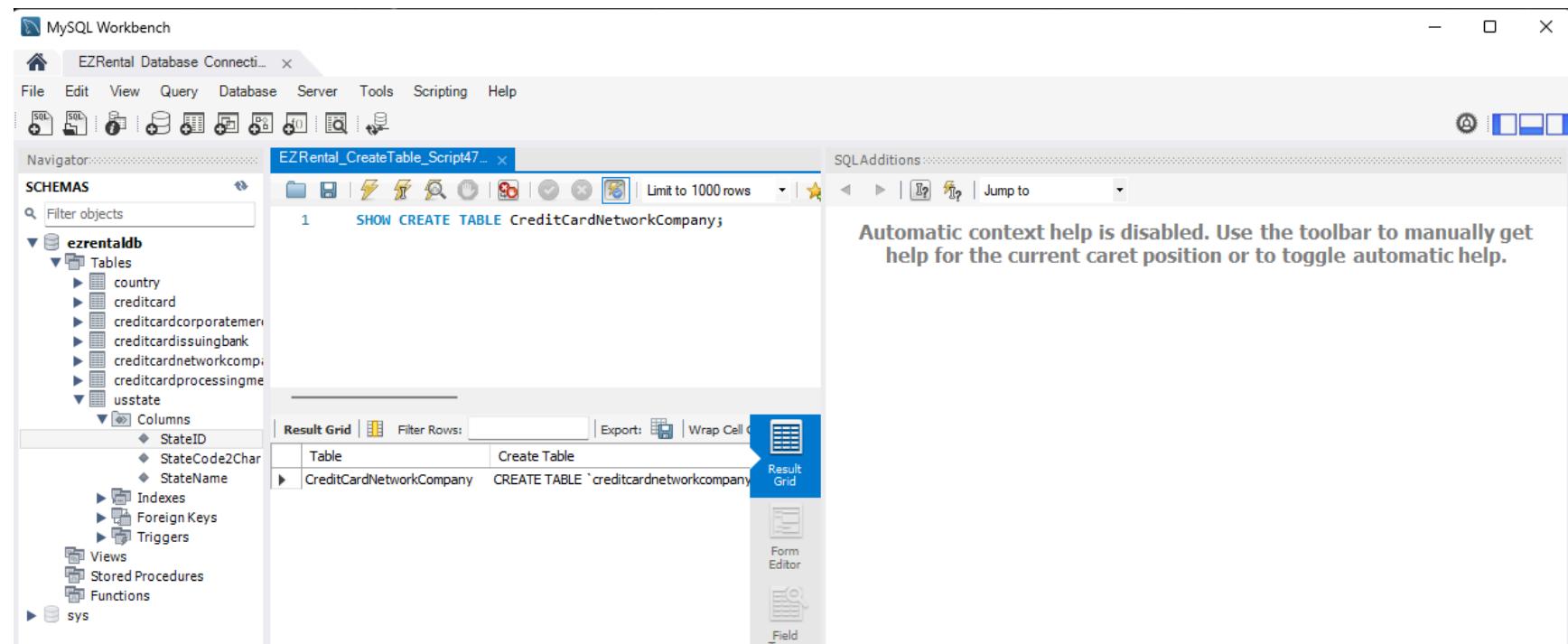
The Physical Model Schema Design Diagram is the DIAGRAM USED TO IMPLEMENT THE DATABASE in the DBMS APPLICATION. THIS DIAGRAM SUMMARIZES THE TABLES & RELATIONSHIPS REQUIRED TO IMPLEMENT THE DATABASE in the DBMS APPLICATION.

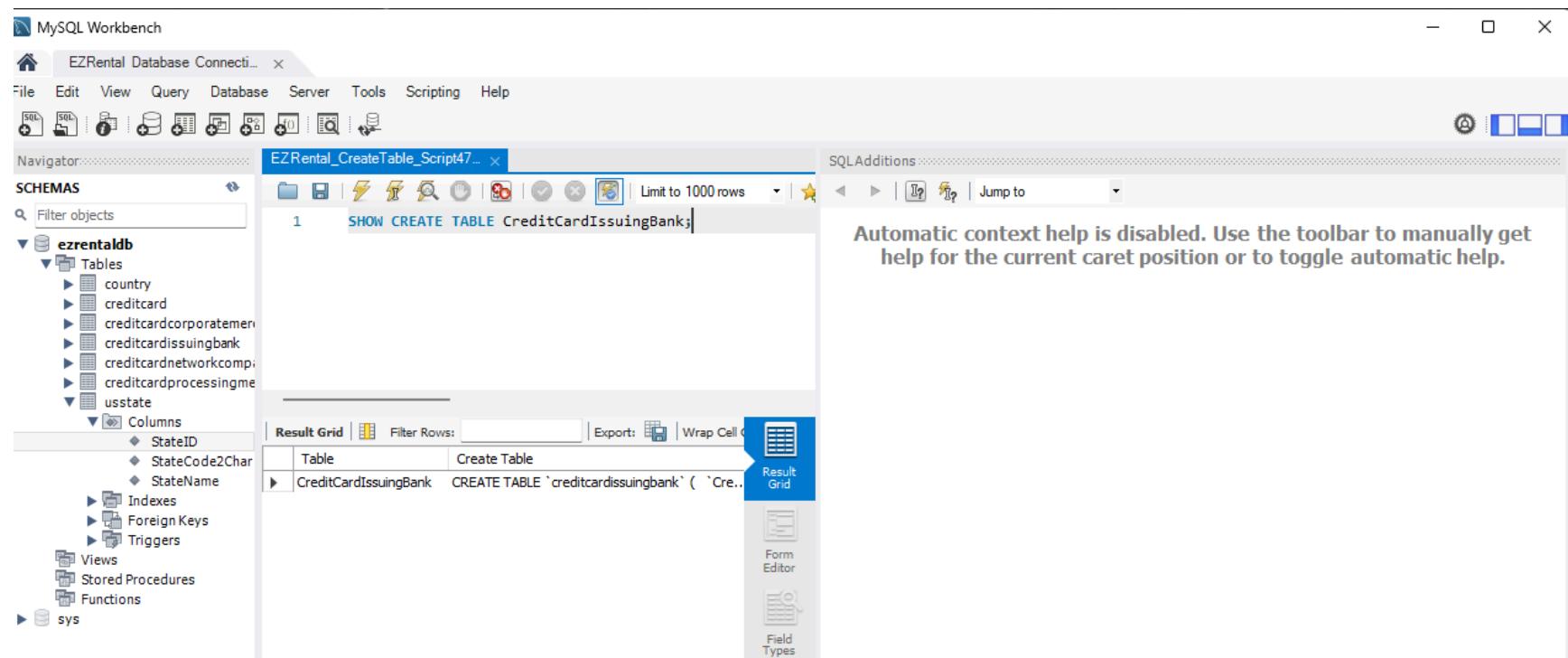


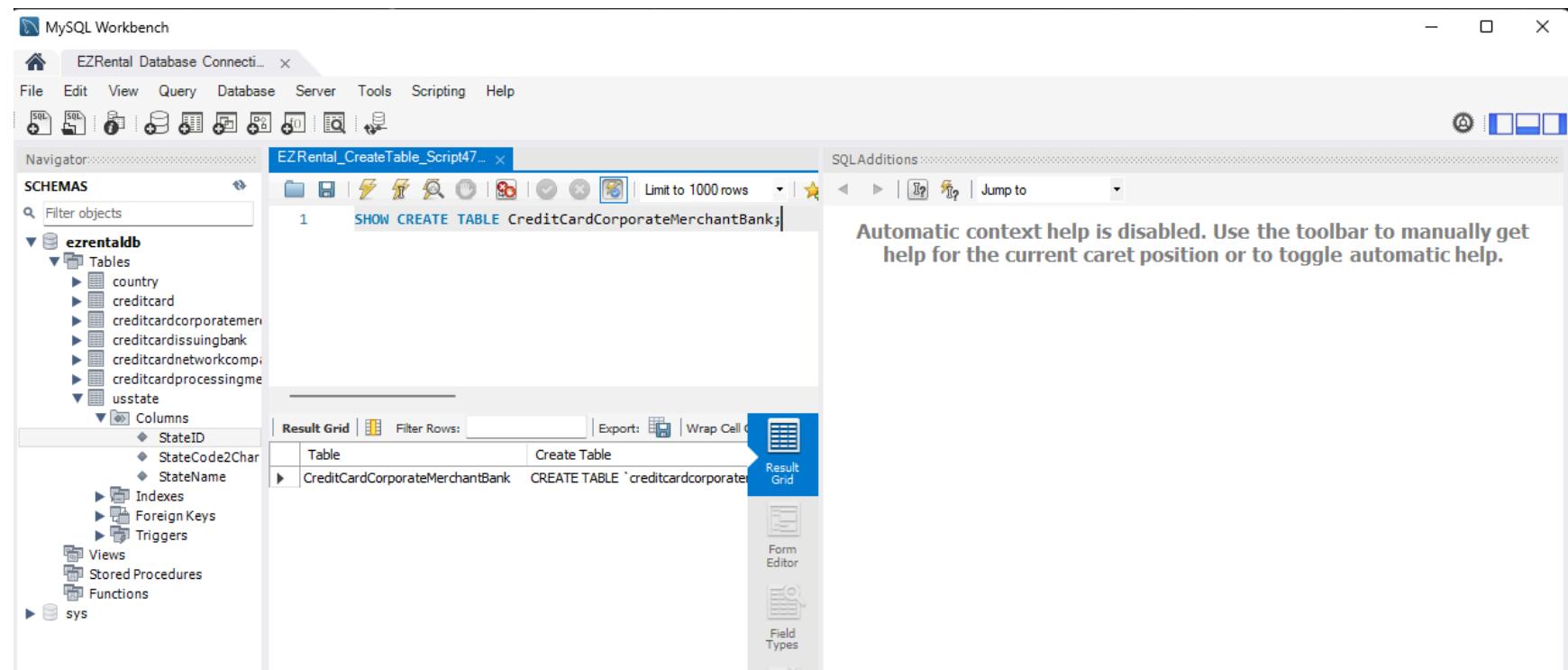
## **Database Implementation Deliverable #6 – Development & Implementation**

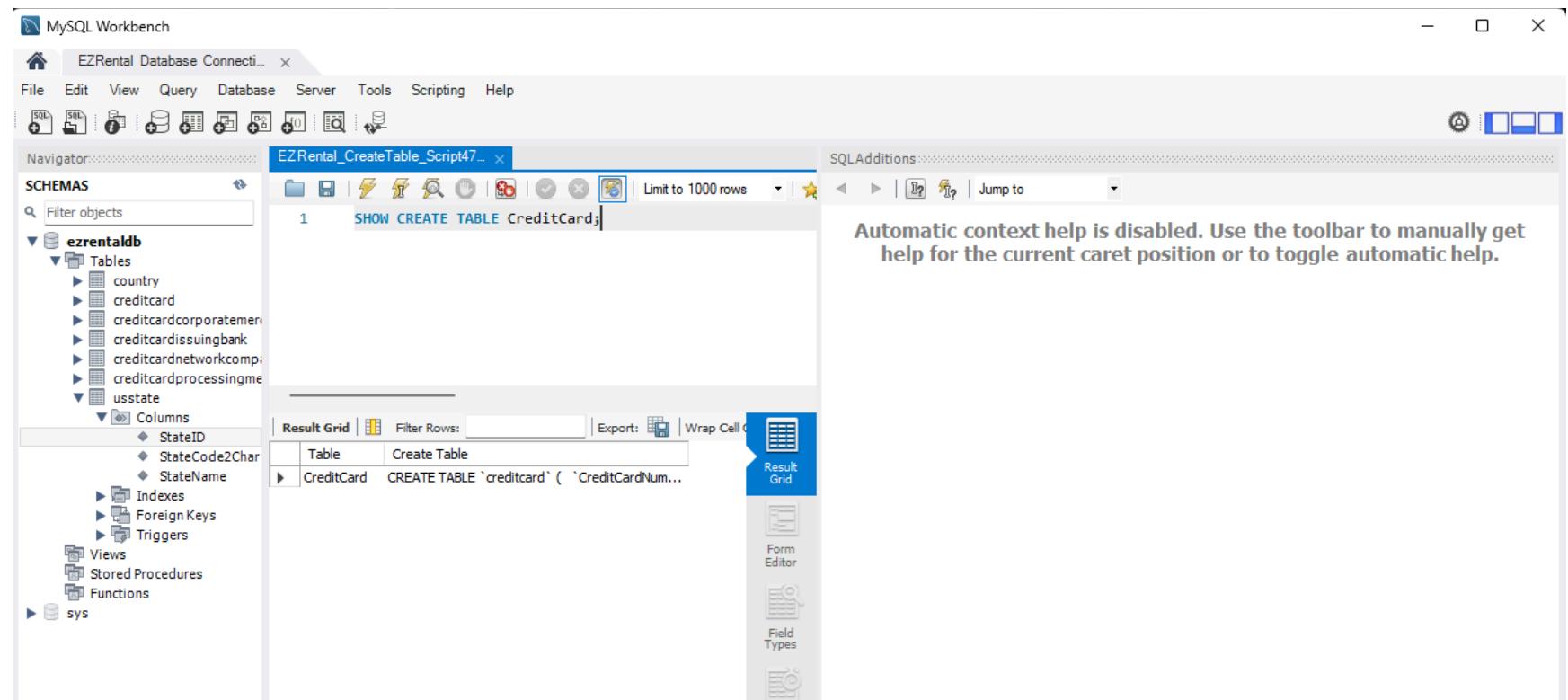
As part of Database Implementation Deliverable #6 – Development & Implementation, the EZRental database schema was successfully developed and implemented using MySQL Workbench. This process included the creation of all required tables listed in the Physical Model Schema Design Diagram of Sprint #2, along with their corresponding primary keys, data types, value constraints, and foreign key relationships. Referential integrity was enforced using FOREIGN KEY constraints with ON DELETE CASCADE and ON UPDATE CASCADE to ensure consistent handling of related data across the system. Key tables such as CreditCard, Country, USState, CreditCardProcessingMerchantServiceCompany, CreditCardNetworkCompany, CreditCardIssuingBank, and CreditCardCorporateMerchantBank were created in proper sequence, allowing seamless integration of lookup data and transactional records. The schema was also reverse-engineered into an ER diagram using MySQL Workbench to visually verify and document the system's relational structure. This deliverable marks the complete and accurate implementation of the database layer for the EZRental application, preparing it for future data population, query development, and system integration.

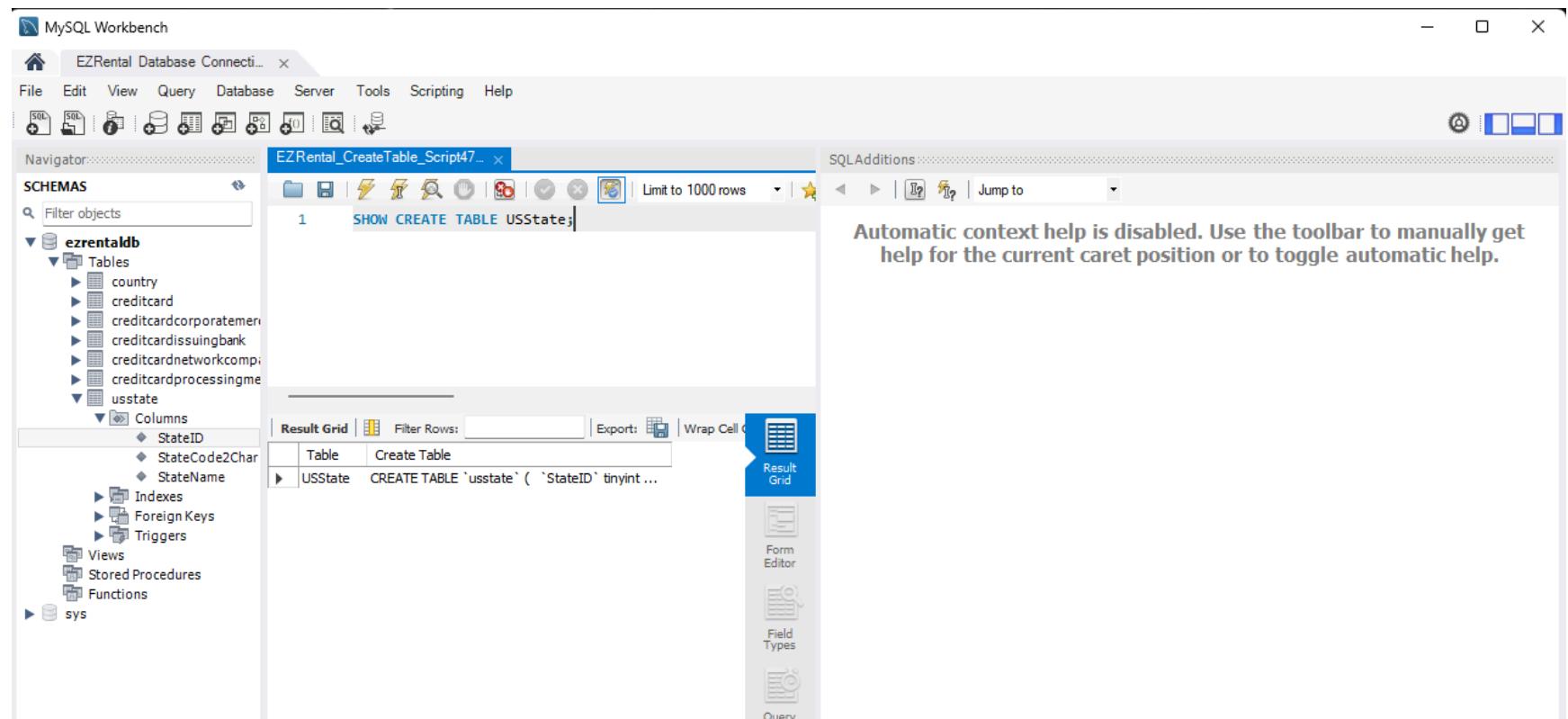


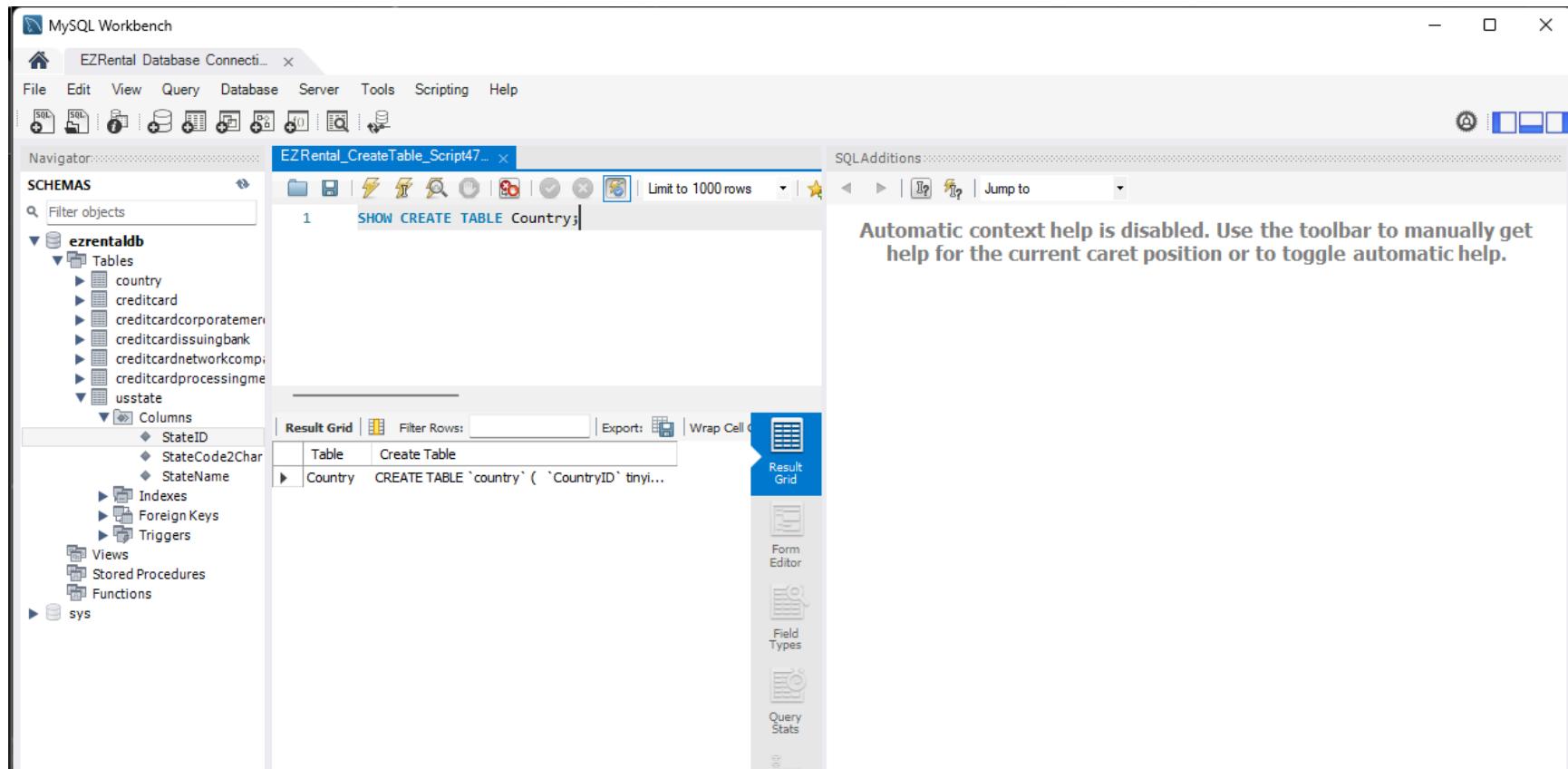


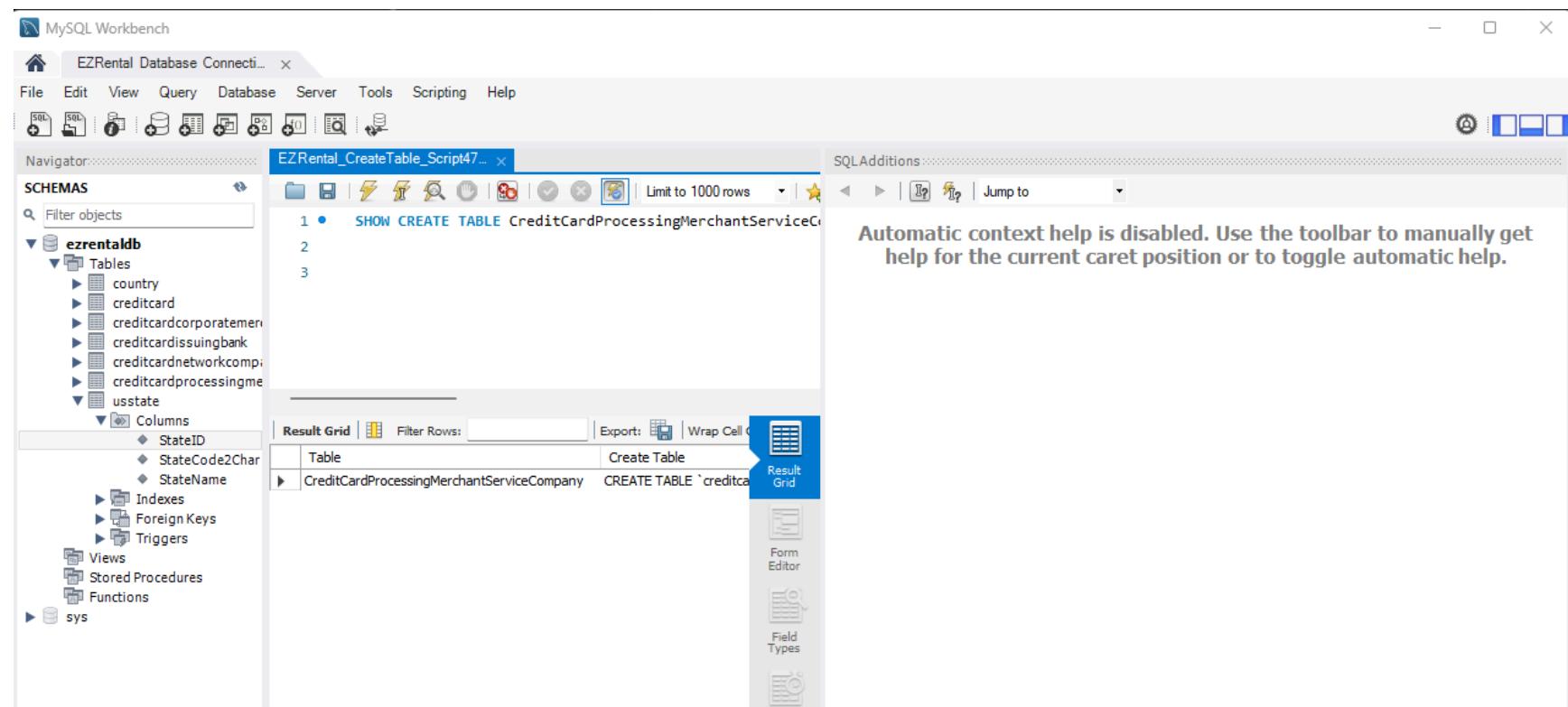


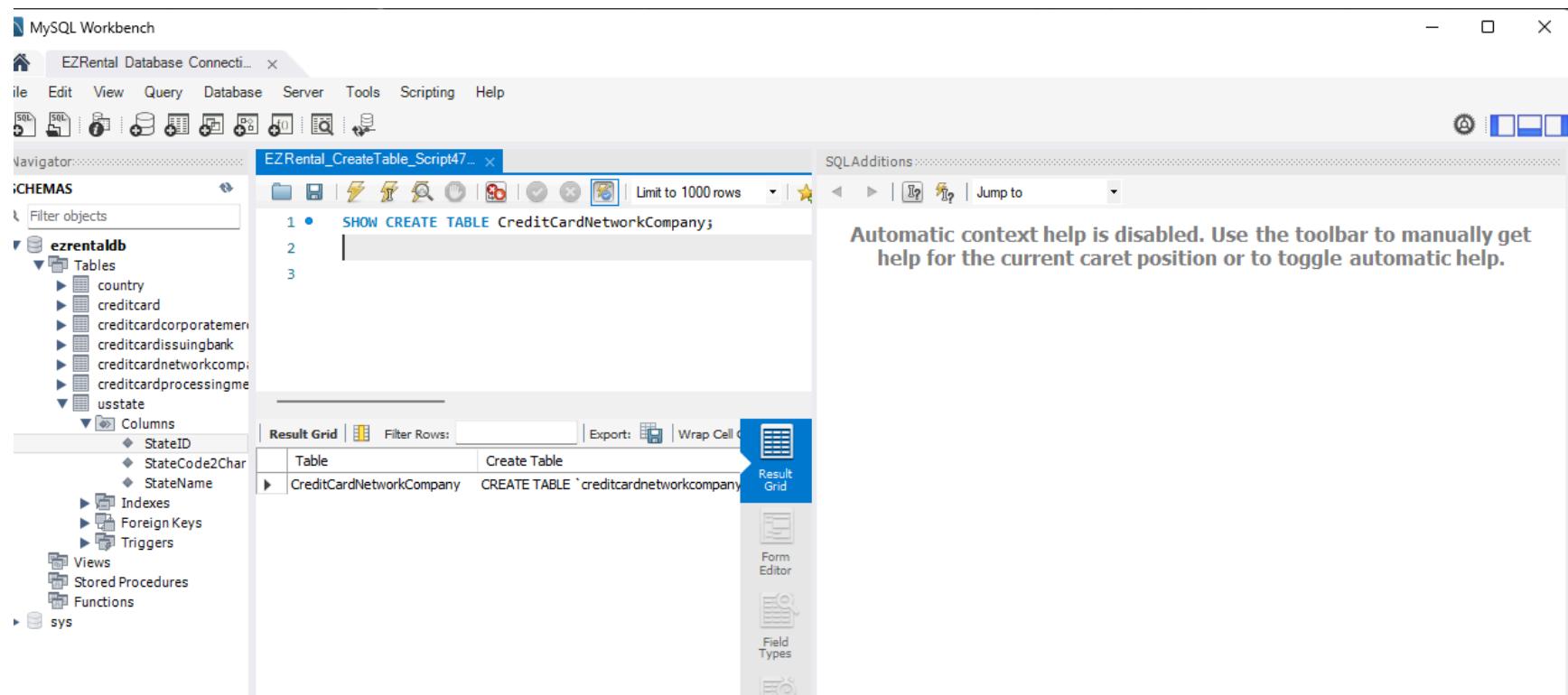


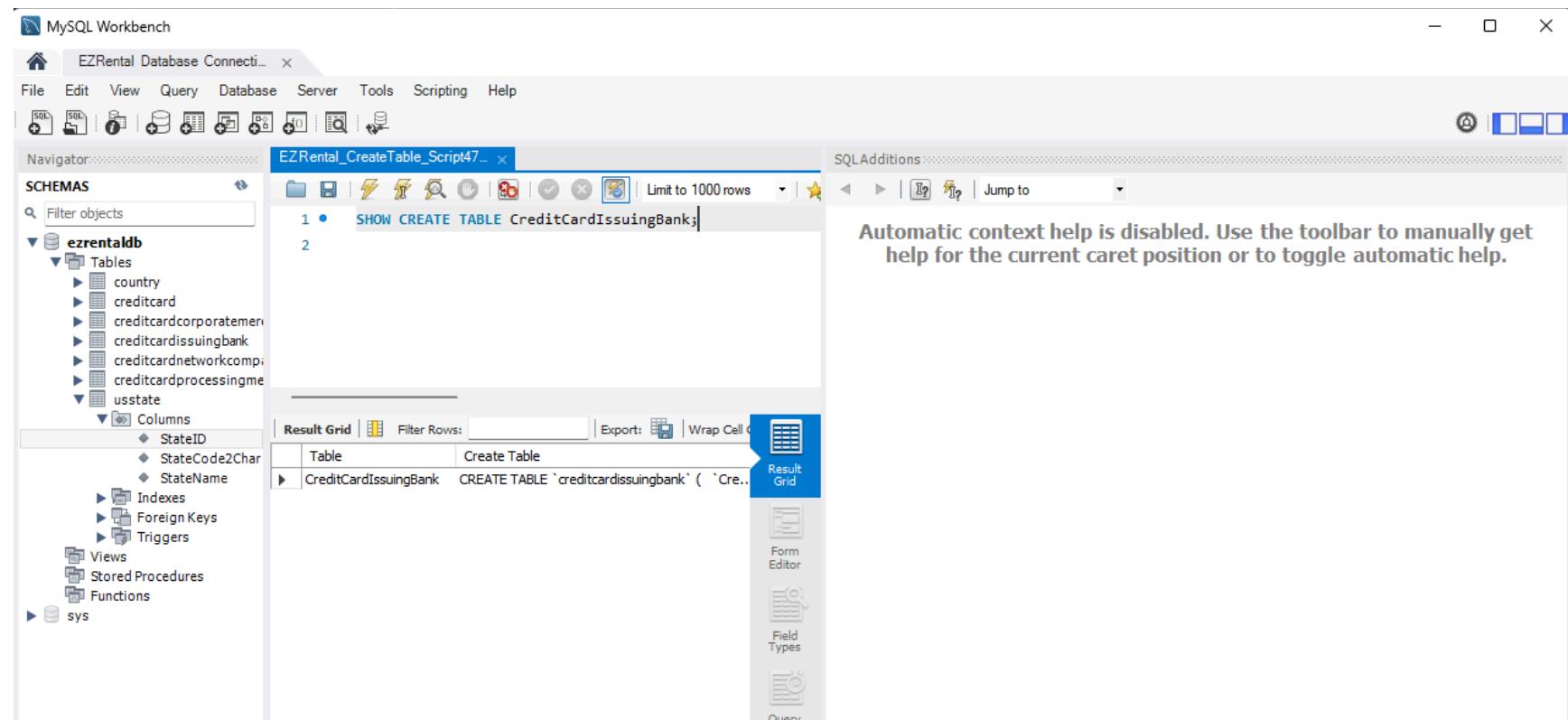


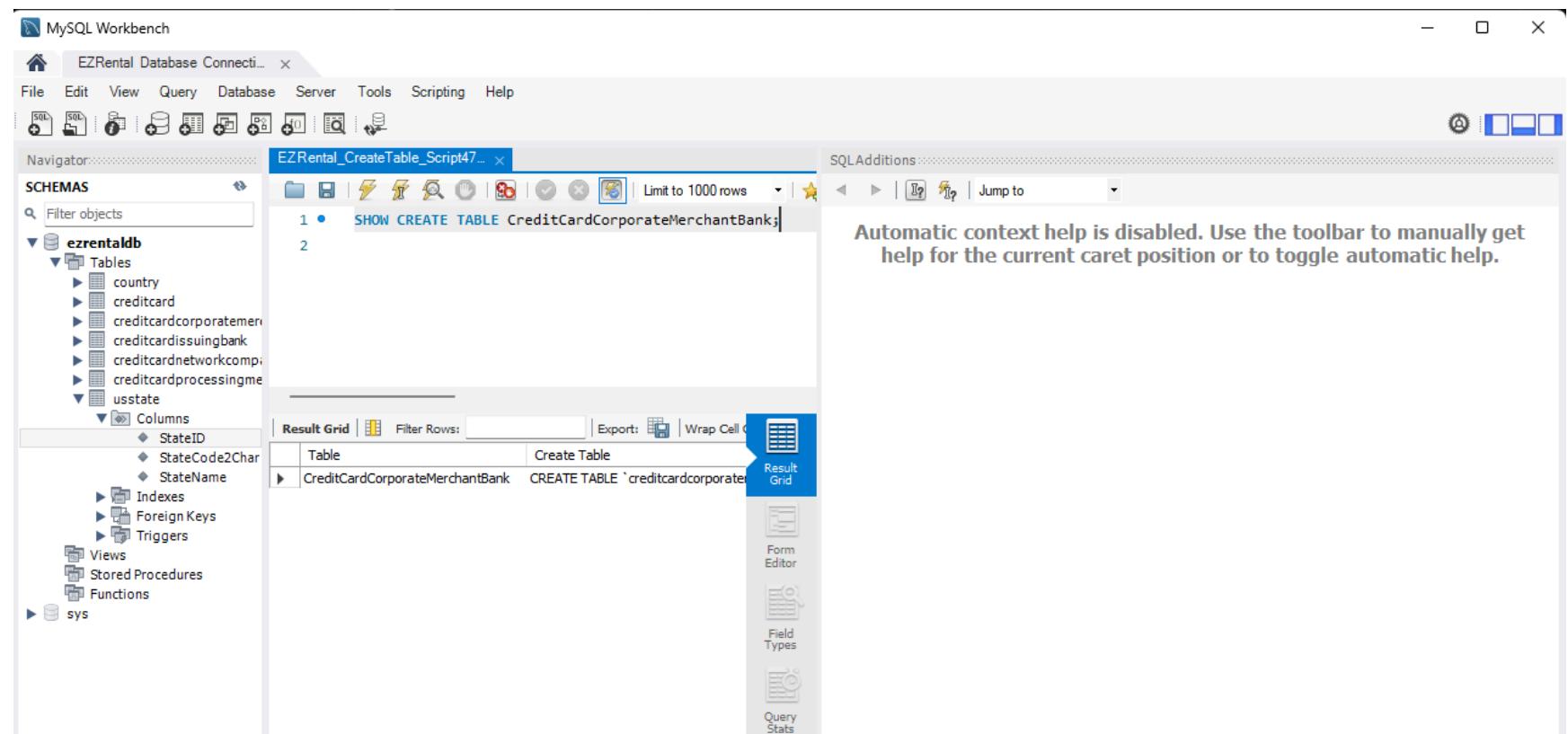


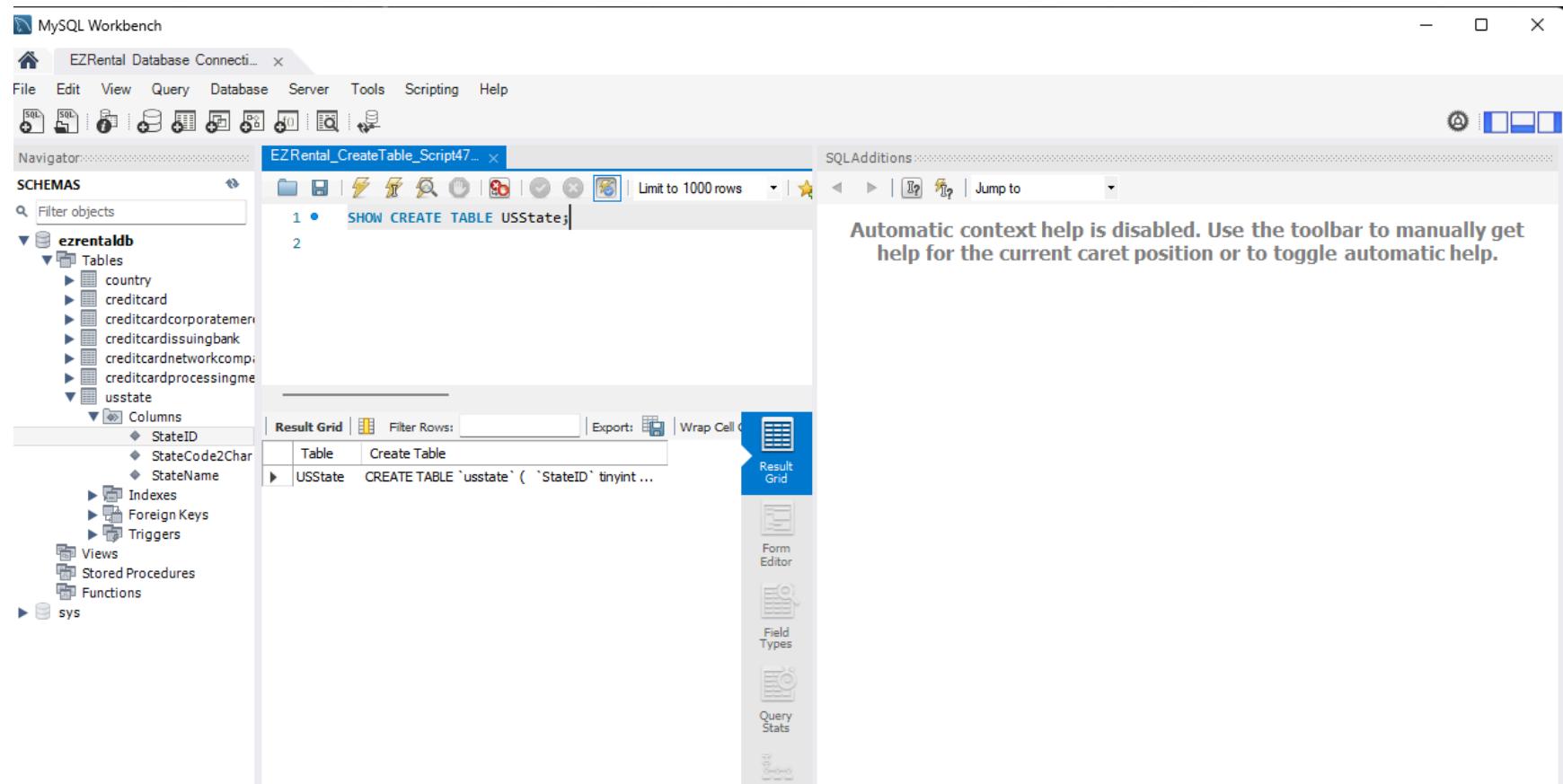


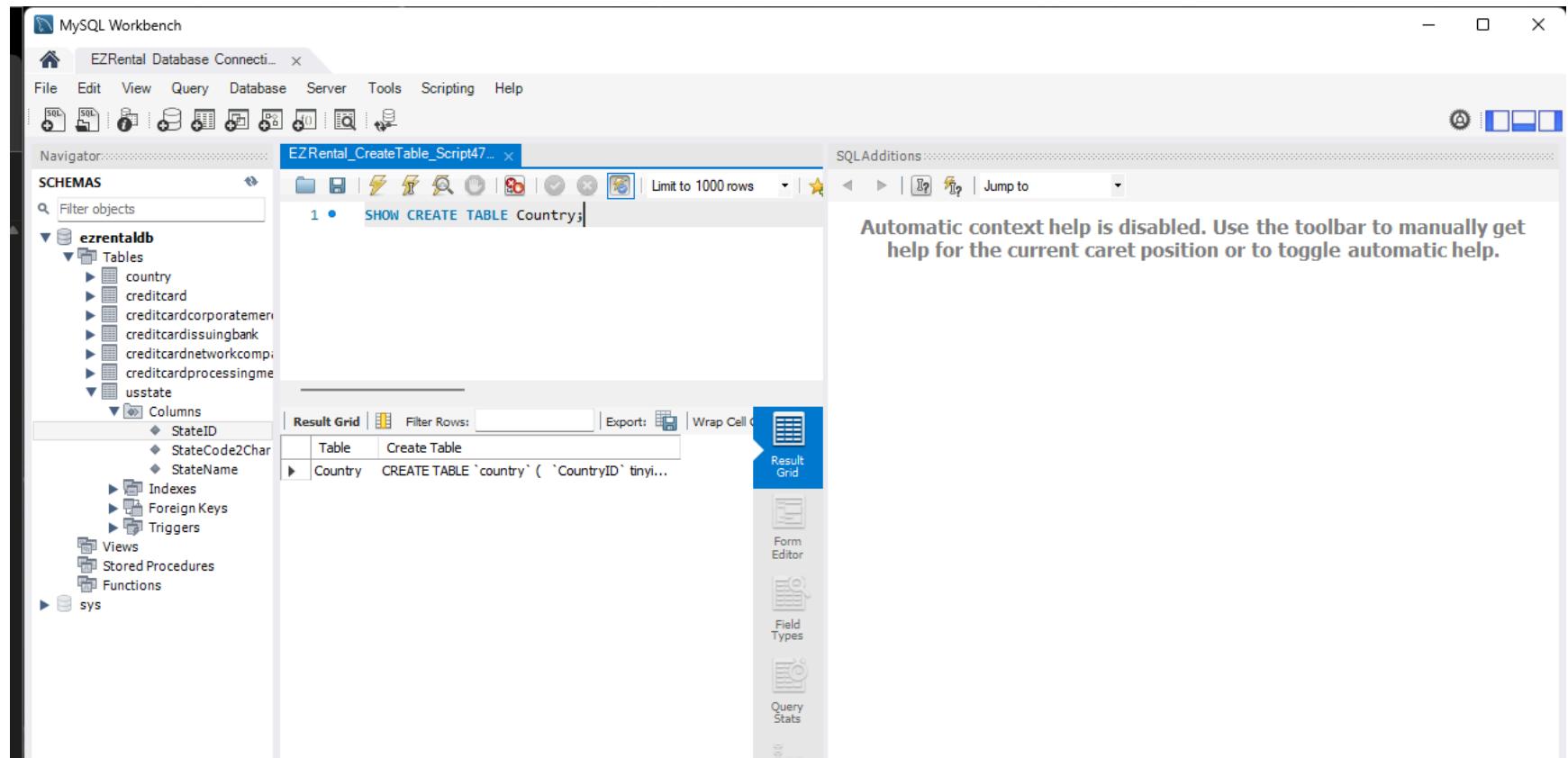


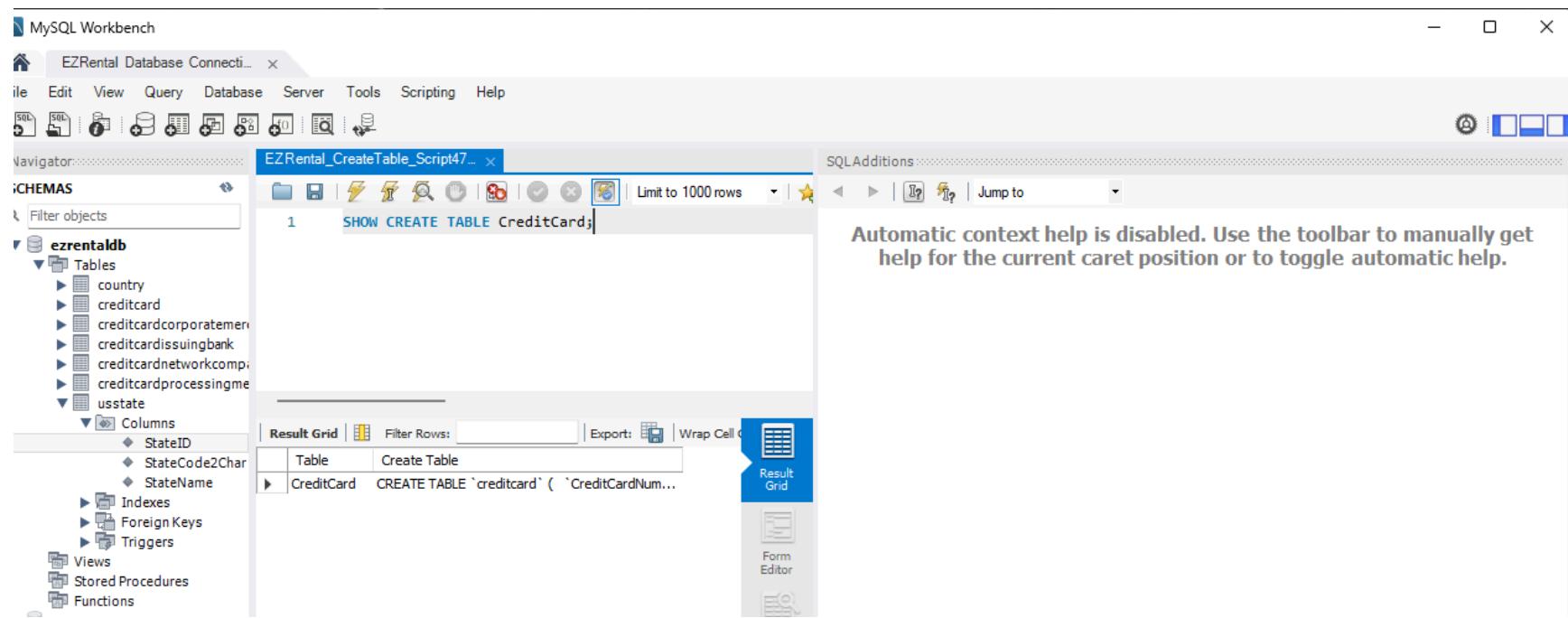


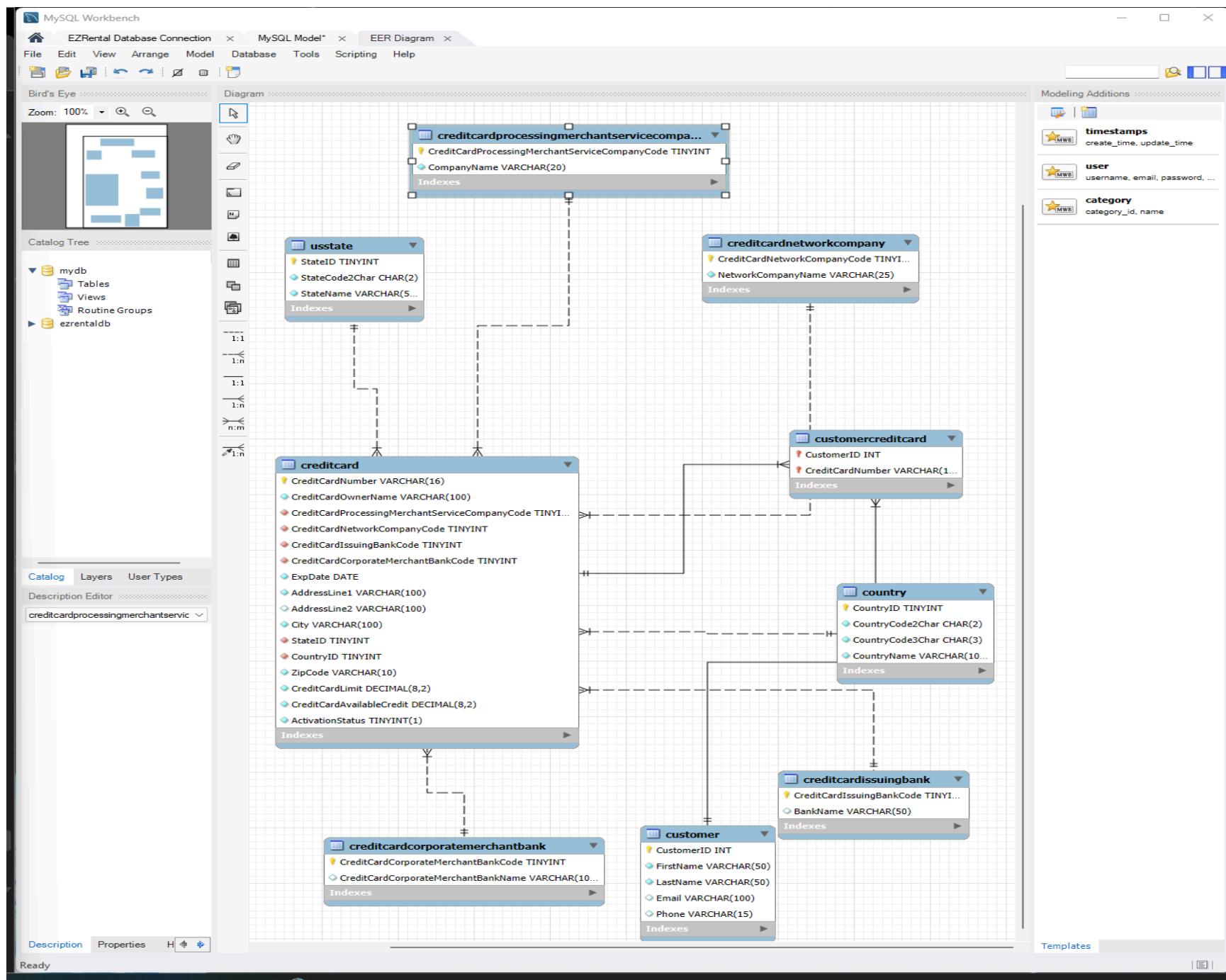








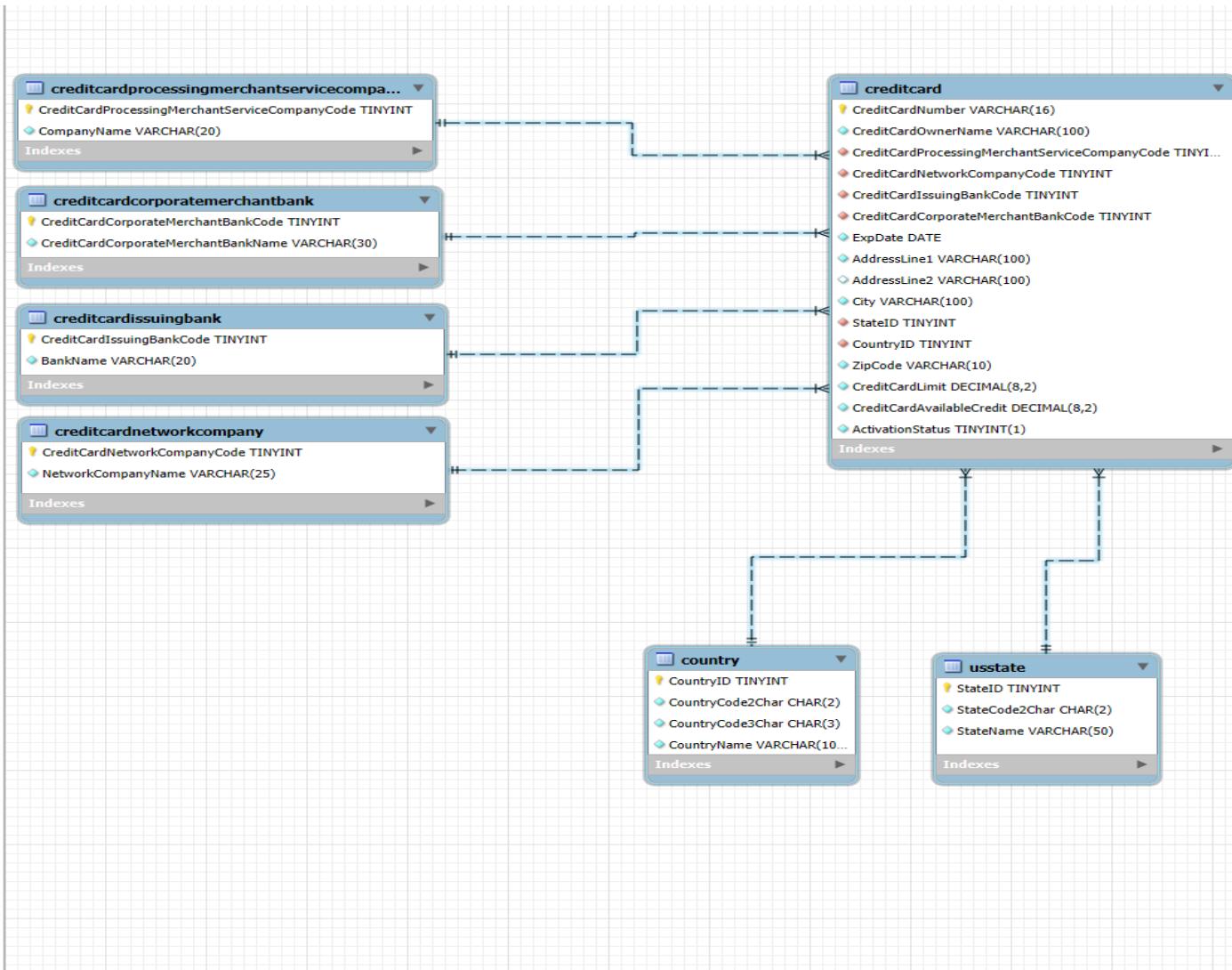






## **Database Implementation Deliverable #7 – Implemented Physical Schema Diagram**

For Database Implementation Deliverable #7 – Implemented Physical Schema Diagram, the finalized EZRental database structure was visually represented using the MySQL Workbench modeling tool. After successfully developing and executing the DDL scripts to create all required tables and relationships, the schema was reverse-engineered to generate a complete Entity-Relationship (ER) diagram. This diagram accurately reflects the implemented physical schema, showcasing the primary and foreign key relationships between core tables such as CreditCard, Country, USState, CreditCardProcessingMerchantServiceCompany, CreditCardNetworkCompany, CreditCardIssuingBank, and CreditCardCorporateMerchantBank. All relational constraints, cascading behaviors, and normalized structures are clearly displayed, ensuring alignment with the design specifications from Sprint #2. The diagram was then formatted, organized for readability, and exported as a high-resolution image file to serve as visual documentation of the system's underlying data architecture. Diagram will be showed below.



(Again, it didn't let me copy and paste, for some reason the code copy and pasted onto my doc.)



**Database Implementation Deliverable #8 – Database Validation Testing**

MySQL Workbench

EZRental Database Connection

File Edit View Query Database Server Tools Scripting Help

Navigator: EZRental\_Statement\_Script4713\* EZRental\_CreateTable\_Script47...

SCHEMAS: Filter objects ezrentaldb Tables: country creditcard creditcardcorporatemem... creditcardissuingbank creditcardnetworkcomp... creditcardprocessingme... usstate Views Stored Procedures Functions sys

SQL Additions: Limit to 1000 rows SQL Editor: SELECT \* FROM CREDITCARD; Jump to Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid: CreditCardNumber CreditCardOwnerName CreditCardProcessingMe...

CreditCardNumber	CreditCardOwnerName	CreditCardProcessingMe...
1111111111111111	Ashley Banks	1
2222222222222221	Carlton Banks	2
2222222222222222	Carlton Banks	3
2222222222222223	Carlton Banks	4
3333333333333333	Shared Card	5
4444444444444444	Hilary Banks	6
5555555555555555	Geoffrey Butler	7
NULL	NULL	NULL

Form Editor

Field Types

Query Stats

Execution Plan

Administration Schemas Information Schema: ezrentaldb

Output: Action Output

#	Time	Action	Message	Duration / Fetch
40	16:13:37	INSERT INTO USState (StateID, StateCode2Char, StateName) VALUES (32, 'NY', ...)	1 row(s) affected	0.000 sec
41	16:13:53	INSERT INTO CREDITCARD ( CreditCardNumber, CreditCardOwnerName, ... )	1 row(s) affected	0.000 sec
42	16:14:37	INSERT INTO CREDITCARD ( CreditCardNumber, CreditCardOwnerName, ... )	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.000 sec
43	16:14:48	INSERT INTO CREDITCARD ( CreditCardNumber, CreditCardOwnerName, ... )	1 row(s) affected	0.000 sec
44	16:15:00	INSERT INTO CREDITCARD ( CreditCardNumber, CreditCardOwnerName, ... )	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0	0.000 sec
45	16:21:30	SELECT * FROM CREDITCARD LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec

Object Info Session 4:33 PM 5/9/2025

```
INSERT INTO CREDITCARD (
    CreditCardNumber, CreditCardOwnerName,
    CreditCardProcessingMerchantServiceCompanyCode, CreditCardNetworkCompanyCode,
    CreditCardIssuingBankCode, CreditCardCorporateMerchantBankCode,
    ExpDate, AddressLine1, AddressLine2,
    City, StateID, Zipcode, CountryID,
    CreditCardLimit, CreditCardAvailableCredit, ActivationStatus
) VALUES
('1111111111111111', 'Ashley Banks', 1, 1, 1, 1, '2027-01-31', '123 Maple St', NULL, 'Brooklyn', 32, '11201', 1, 5000.00, 5000.00, 1),
('222222222222221', 'Carlton Banks', 2, 2, 2, 2, '2026-08-15', '456 Oak Ave', NULL, 'Queens', 32, '11377', 1, 4000.00, 4000.00, 1),
('222222222222222', 'Carlton Banks', 3, 3, 3, 3, '2026-09-30', '456 Oak Ave', NULL, 'Queens', 32, '11377', 1, 3500.00, 3500.00, 1),
('222222222222223', 'Carlton Banks', 4, 4, 4, 1, '2027-04-25', '456 Oak Ave', NULL, 'Queens', 32, '11377', 1, 3000.00, 2800.00, 1),
('333333333333333', 'Shared Card', 5, 5, 5, 2, '2028-12-01', '789 Pine Blvd', NULL, 'Bronx', 32, '10451', 1, 7000.00, 6900.00, 1),
('444444444444444', 'Hilary Banks', 6, 6, 6, 3, '2025-11-30', '300 Fashion Ln', NULL, 'Manhattan', 32, '10001', 1, 6500.00, 6200.00,
1),
('555555555555555', 'Geoffrey Butler', 7, 7, 7, 3, '2029-07-01', '1 Butler Rd', NULL, 'Brooklyn', 32, '11205', 1, 6000.00, 6000.00, 1);
```

The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The left sidebar displays the Navigator with the schema 'ezrentaldb' selected, showing tables like country, creditcard, and CreditCardProcessingMerchantServiceCompany. The main area contains two tabs: 'EZRental\_Statement\_Script4713\*' and 'EZRental\_CreateTable\_Script47...'. The first tab contains a SQL query:

```
1 •   SELECT * FROM CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY
```

The results grid shows the following data:

	CreditCardProcessingMerchantServiceCompanyCode	CompanyName
1		Stripe
2		Square
3		PayPal
4		Authorize.Net
5		Adyen
6		Worldpay
7		Fiserv
8		Elavon
9		Global Payments
10		BlueSnap
11		Amazon Pay
*	NULL	NULL

The right side of the interface includes a toolbar with various icons and a message: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help." Below the toolbar are buttons for Result Grid, Form Editor, Field Types, Query Stats, and Execution Plan.

INSERT INTO CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY (  
CreditCardProcessingMerchantServiceCompanyCode, CompanyName

) VALUES  
(1, 'Stripe'),  
(2, 'Square'),  
(3, 'PayPal'),  
(4, 'Authorize.Net'),  
(5, 'Adyen'),  
(6, 'Worldpay'),  
(7, 'Fiserv'),  
(8, 'Elavon'),  
(9, 'Global Payments'),  
(10, 'BlueSnap'),  
(11, 'Amazon Pay');

MySQL Workbench

EZRental Database Connect... x

Edit View Query Database Server Tools Scripting Help

Navigator: EZRental\_Statement\_Script4713\* EZRental\_CreateTable\_Script47...

HMAS

ezrentaldb

Tables: country, creditcard, creditcardcorporatemerch, creditcardissuingbank, creditcardnetworkcomp, creditcardprocessingmerch, usstate

Views

Stored Procedures

Functions

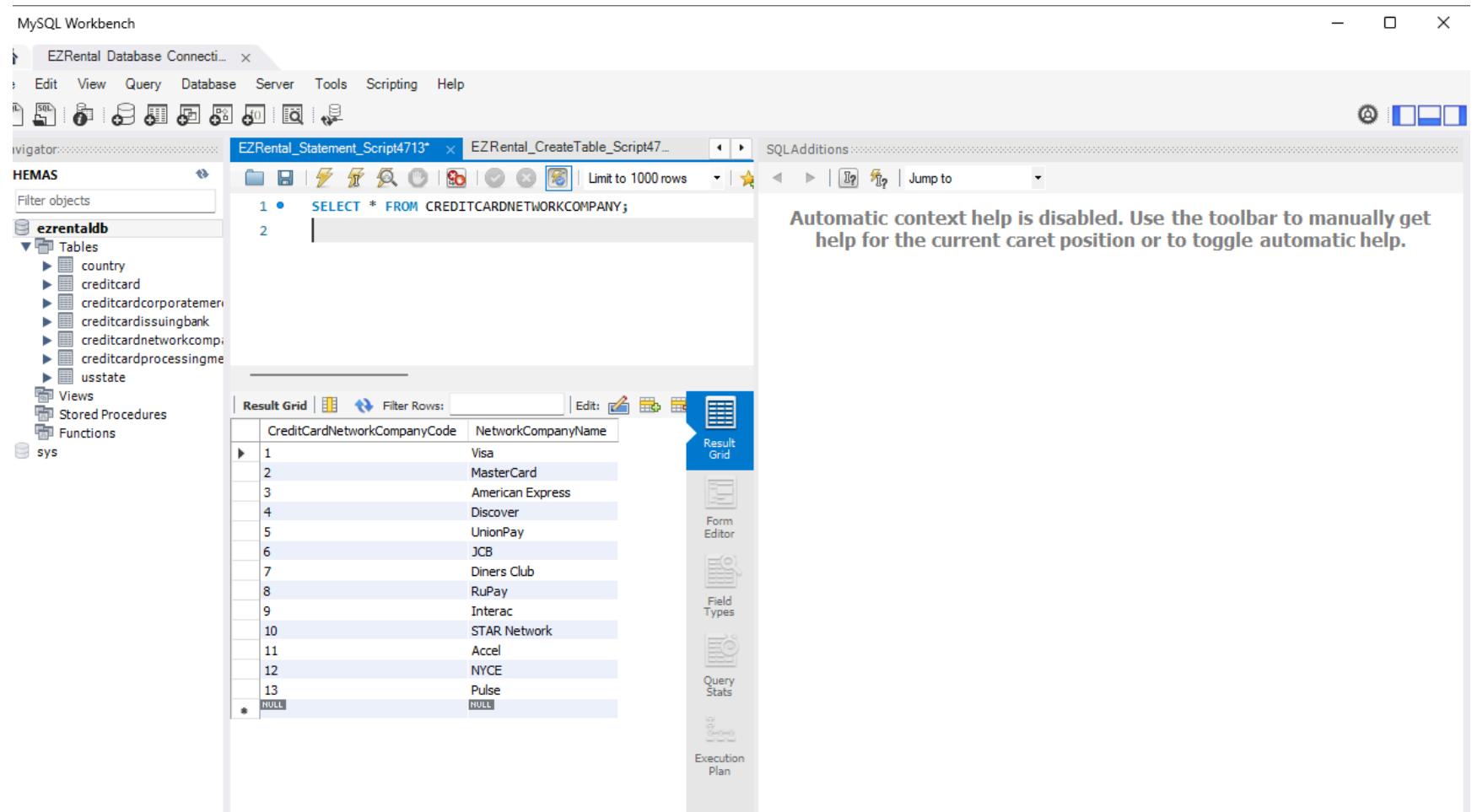
sys

SQL Additions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid | Filter Rows: Edit: Result Grid Form Editor Field Types Query Stats Execution Plan

	CreditCardNetworkCompanyCode	NetworkCompanyName
1		Visa
2		MasterCard
3		American Express
4		Discover
5		UnionPay
6		JCB
7		Diners Club
8		RuPay
9		Interac
10		STAR Network
11		Accel
12		NYCE
13		Pulse
*	NULL	NULL



```
INSERT INTO CREDITCARDNETWORKCOMPANY (
    CreditCardNetworkCompanyCode, NetworkCompanyName
) VALUES
(1, 'Visa'),
(2, 'MasterCard'),
```

(3, 'American Express'),  
(4, 'Discover'),  
(5, 'UnionPay'),  
(6, 'JCB'),  
(7, 'Diners Club'),  
(8, 'RuPay'),  
(9, 'Interac'),  
(10, 'STAR Network'),  
(11, 'Accel'),  
(12, 'NYCE'),  
(13, 'Pulse');

The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The toolbar below has icons for New Connection, Home, SQL Editor, Text Editor, Results Grid, Form Editor, Field Types, Query Stats, and Execution. The Navigator pane on the left lists Schemas (ezrentaldb, sys), Views, Stored Procedures, and Functions. The main area displays a query editor with the following code:

```
1 •  SELECT * FROM CREDITCARDISSUINGBANK;
```

The results grid shows the following data:

	CreditCardIssuingBankCode	BankName
1		Chase Bank
2		Bank of America
3		Wells Fargo
4		Citbank
5		Capital One
6		US Bank
7		PNC Bank
8		TD Bank
9		HSBC
10		American Express Bank
*	HULL	NULL

The status bar at the bottom right says: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help."

```
INSERT INTO CREDITCARDISSUINGBANK (
    CreditCardIssuingBankCode, BankName
) VALUES
(1, 'Chase Bank'),
(2, 'Bank of America'),
(3, 'Wells Fargo'),
```

```
(4, 'Citibank'),  
(5, 'Capital One'),  
(6, 'US Bank'),  
(7, 'PNC Bank'),  
(8, 'TD Bank'),  
(9, 'HSBC'),  
(10, 'American Express Bank');
```

The screenshot shows the MySQL Workbench interface. The title bar reads "MySQL Workbench". The main window has two tabs: "EZRental\_Statement\_Script4713\*" and "EZRental\_CreateTable\_Script47...". The "EZRental\_Statement\_Script4713\*" tab contains a SQL editor with the following query:

```
1 •  SELECT * FROM CREDITCARDCORPORATEMERCHANTBANK;
```

The results grid below shows the following data:

CreditCardCorporateMerchantBankCode	CreditCardCorporateMerchant
1	JPMorgan Corporate
2	Bank of America Merchant Services
3	Wells Fargo Corporate Banking
*	NULL

The "Navigator" pane on the left shows the database schema, including tables like "country", "creditcard", and "creditcardcorporatemerchantbank". The "SQLAdditions" pane on the right displays a message: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help."

```
INSERT INTO CREDITCARDCORPORATEMERCHANTBANK (
    CreditCardCorporateMerchantBankCode, CreditCardCorporateMerchantBankName
) VALUES
(1, 'JPMorgan Corporate'),
(2, 'Bank of America Merchant Services'),
(3, 'Wells Fargo Corporate Banking');
```

In this test, we simulate a front-end transaction where a customer is about to rent a reserved vehicle and provides their credit card. EZRental staff needs to retrieve all relevant **credit card information**, including **address details**, **credit limits**, and **associated financial institutions**. However, **only the names of the processing companies, banks, and networks are needed** — not their numeric codes.

This query helps validate how the front-end retrieves customer billing and processing information cleanly using JOINs on referenced tables.

```
SELECT
    c.CreditCardNumber,
    c.CreditCardOwnerName,
    c.ExpDate,
    c.AddressLine1,
    c.AddressLine2,
```

c.City,  
c.Zipcode,  
c.CreditCardLimit,  
c.CreditCardAvailableCredit,  
c.ActivationStatus,  
pm.CompanyName AS ProcessingCompanyName,  
nw.NetworkCompanyName,  
ib.BankName AS IssuingBankName,  
cb.CreditCardCorporateMerchantBankName AS CorporateBankName  
FROM CREDITCARD c  
JOIN CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY pm  
ON c.CreditCardProcessingMerchantServiceCompanyCode = pm.CreditCardProcessingMerchantServiceCompanyCode  
JOIN CREDITCARDNETWORKCOMPANY nw  
ON c.CreditCardNetworkCompanyCode = nw.CreditCardNetworkCompanyCode  
JOIN CREDITCARDISSUINGBANK ib  
ON c.CreditCardIssuingBankCode = ib.CreditCardIssuingBankCode  
JOIN CREDITCARDCORPORATEMERCHANTBANK cb

```
ON c.CreditCardCorporateMerchantBankCode = cb.CreditCardCorporateMerchantBankCode  
WHERE c.CreditCardNumber = '1111111111111111';
```

MySQL Workbench

EZRental Database Connect... x

File Edit View Query Database Server Tools Scripting Help

Navigator: EZRental\_Statement\_Script4713\* EZRental\_CreateTable\_Script47...

SCHEMAS

ezrentaldb

- Tables
  - country
  - creditcard
  - creditcardcorporatemer...
  - creditcardissuingbank
  - creditcardnetworkcomp...
  - creditcardprocessingme...
  - usstate
- Views
- Stored Procedures
- Functions

sys

SQLAdditions:

```

20     ON c.CreditCardNetworkCompanyCode = nw.CreditCardN
21     JOIN CREDITCARDISSUINGBANK ib
22         ON c.CreditCardIssuingBankCode = ib.CreditCardIssu
23     JOIN CREDITCARDCORPORATEMERCHANTBANK cb
24         ON c.CreditCardCorporateMerchantBankCode = cb.Cred
25 WHERE c.CreditCardNumber = '1111111111111111';
26

```

Result Grid | Filter Rows: \_\_\_\_\_ | Export: \_\_\_\_\_ | Wrap

	AddressLine1	City	Zipcode	CreditCardLimit	CreditCardOwnerName
he1	HULL	Brooklyn	11201	5000.00	5000.00

Result Grid

MySQL Workbench

EZRental Database Connect... x

File Edit View Query Database Server Tools Scripting Help

Navigator: EZRental\_Statement\_Script4713\* EZRental\_CreateTable\_Script47...

SCHEMAS

ezrentaldb

- Tables
  - country
  - creditcard
  - creditcardcorporatemer...
  - creditcardissuingbank
  - creditcardnetworkcomp...
  - creditcardprocessingme...
  - usstate
- Views
- Stored Procedures
- Functions

sys

SQLAdditions:

```

20     ON c.CreditCardNetworkCompanyCode = nw.CreditCardN
21     JOIN CREDITCARDISSUINGBANK ib
22         ON c.CreditCardIssuingBankCode = ib.CreditCardIssu
23     JOIN CREDITCARDCORPORATEMERCHANTBANK cb
24         ON c.CreditCardCorporateMerchantBankCode = cb.Cred
25 WHERE c.CreditCardNumber = '1111111111111111';
26

```

Result Grid | Filter Rows: \_\_\_\_\_ | Export: \_\_\_\_\_ | Wrap

CreditCardNumber	CreditCardOwnerName	ExpDate	AddressLine1
1111111111111111	Ashley Banks	2027-01-31	123 Maple

Result Grid

This test validates the ability of the database to retrieve a **single credit card record** using its **primary key**: CreditCardNumber. It simulates a front-end application search where a user or system queries full credit card data based on a unique card number.

This helps ensure data can be individually accessed when a customer provides their credit card in-person or through a system.

```
SELECT *
```

```
FROM CREDITCARD
```

```
WHERE CreditCardNumber = '1111111111111111';
```

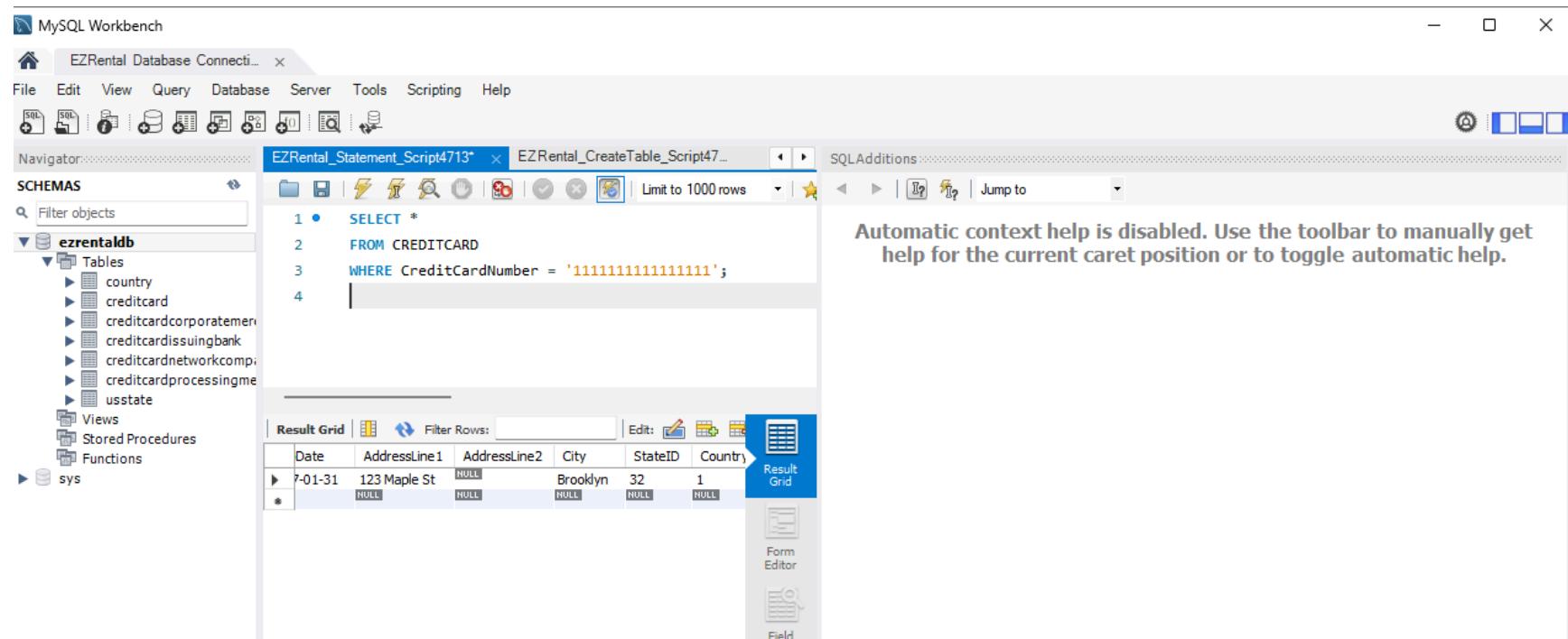
The screenshot shows the MySQL Workbench interface. The title bar says "MySQL Workbench". The main window has tabs for "EZRental Statement\_Script4713\*" and "EZRental\_CreateTable\_Script47...". The left sidebar shows the "Navigator" with "SCHEMAS" expanded to show "ezrentaldb" which contains "Tables" (country, creditcard, creditcardcorporatemerch, creditcardissuingbank, creditcardnetworkcomp, creditcardprocessingmerch, usstate), "Views", "Stored Procedures", and "Functions". It also shows "sys". The central pane displays the SQL query:

```
1 •  SELECT *
2   FROM CREDITCARD
3   WHERE CreditCardNumber = '1111111111111111';
4
```

The right pane shows the "Result Grid" with the following data:

CreditCardNumber	CreditCardOwnerName	CreditCardProcessingMerch
1111111111111111	Ashley Banks	1
NULL	NULL	NULL

A message in the right pane states: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help."



This validation test demonstrates how a user or business report can retrieve **a subset of customer credit card information** from the CREDITCARD table based on non-primary key criteria (e.g., City, State, Country, or Zip Code). It simulates front-end features for filtering credit card records for reporting or verification, returning only **specific columns**.

For this example, we use **City = 'Brooklyn'** as the filter.

MySQL Workbench

EZRental Database Connect... x

File Edit View Query Database Server Tools Scripting Help

Navigator: EZRental\_Statement\_Script4713\* EZRental\_CreateTable\_Script47...

SCHEMAS

ezrentaldb

- Tables
  - country
  - creditcard
  - creditcardcorporatemer...
  - creditcardissuingbank
  - creditcardnetworkcompi...
  - creditcardprocessingme...
  - usstate
- Views
- Stored Procedures
- Functions

sys

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```
10      cn.CountryName AS Country,
11      c.ActivationStatus
12  FROM CREDITCARD c
13  JOIN USState s ON c.StateID = s.StateID
14  JOIN Country cn ON c.CountryID = cn.CountryID
15  WHERE c.city = 'Brooklyn';
16
```

Result Grid | Filter Rows: \_\_\_\_\_ | Export: \_\_\_\_\_ | Wrap

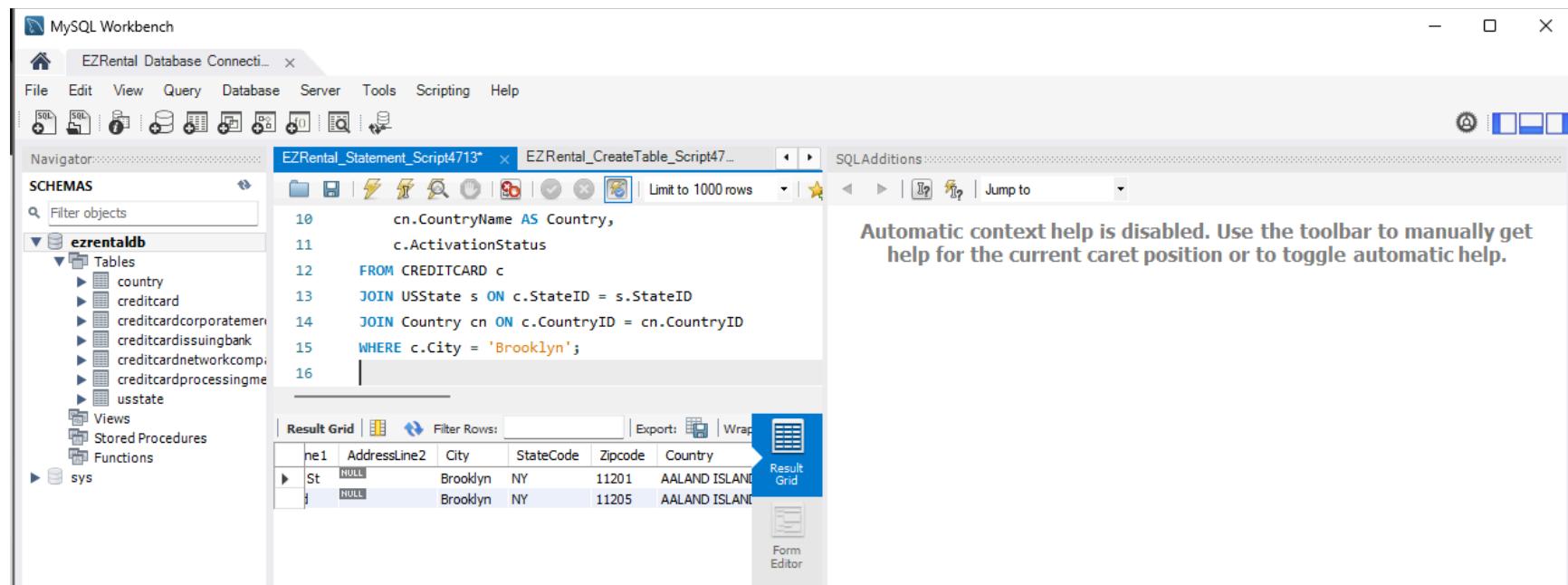
CreditCardNumber	CreditCardOwnerName	ExpDate	AddressLine1
1111111111111111	Ashley Banks	2027-01-31	123 Maple
5555555555555555	Geoffrey Butler	2029-07-01	1 Butler Rd

Result Grid

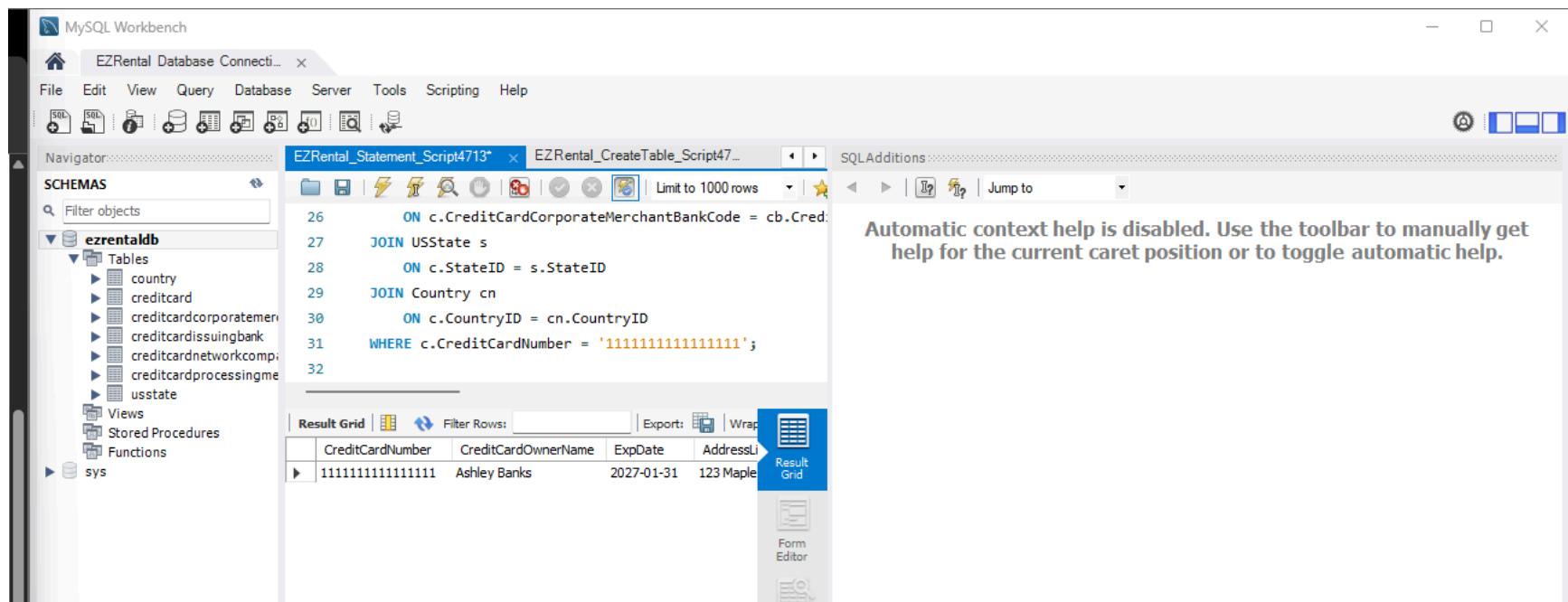
Form Editor

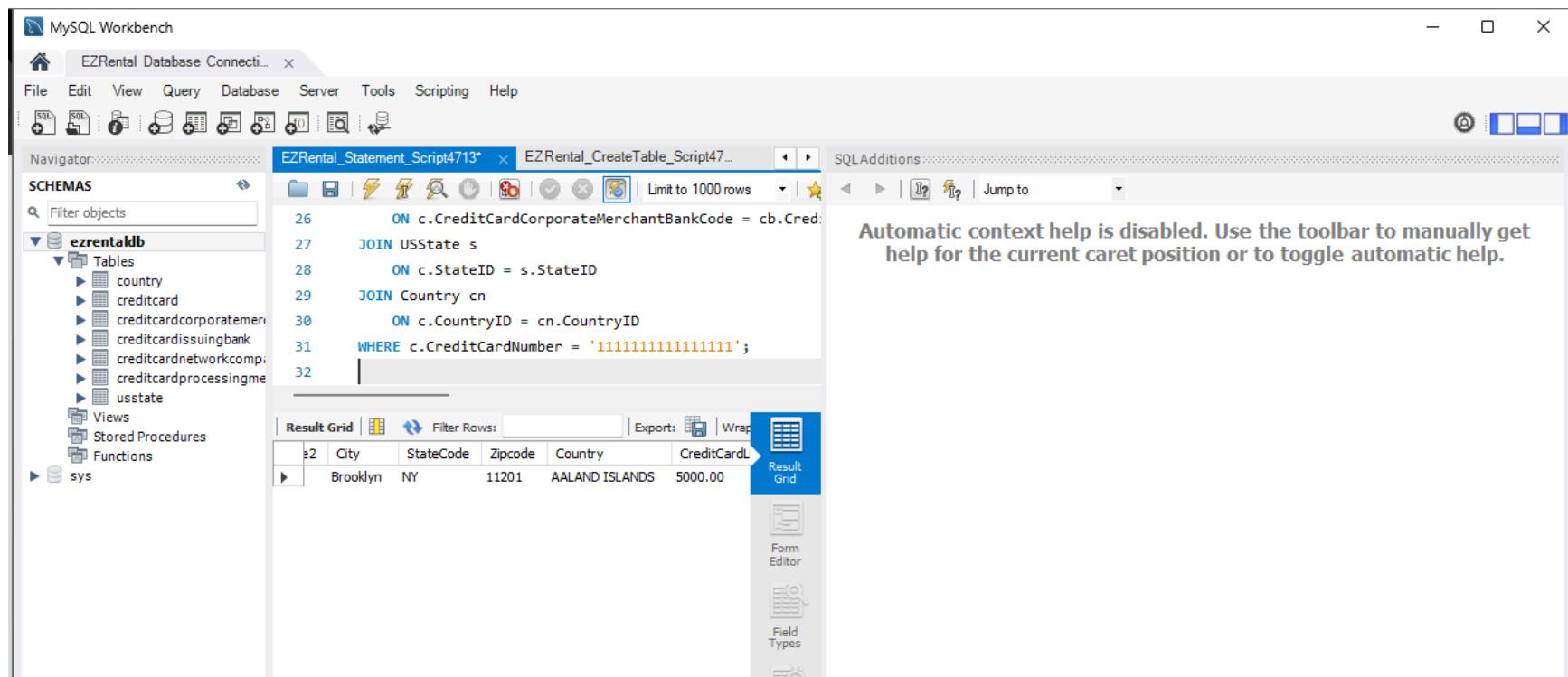
Field Tuner

A screenshot of the MySQL Workbench interface. The left sidebar shows the database schema with tables like country, creditcard, and usstate. The main area displays a SQL query and its results. The query joins three tables: CREDITCARD, USState, and Country, filtering for records where the city is 'Brooklyn'. The results are shown in a grid with columns: CreditCardNumber, CreditCardOwnerName, ExpDate, and AddressLine1. Two rows are present in the result set.



This test simulates retrieving complete billing and processing info for a **specific credit card**, combining the CREDITCARD table with all related processing company tables. The result is used in the front-end application when a customer presents a credit card and all readable metadata is required (without showing foreign key codes).





MySQL Workbench

EZRental Database Connecti... x

File Edit View Query Database Server Tools Scripting Help

Navigator: EZRental\_Statement\_Script4713\* EZRental\_CreateTable\_Script47...

SCHEMAS

ezrentaldb

- Tables
  - country
  - creditcard
  - creditcardcorporatemer...
  - creditcardissuingbank
  - creditcardnetworkcomp...
  - creditcardprocessingme...
  - usstate
- Views
- Stored Procedures
- Functions

sys

SQLAdditions

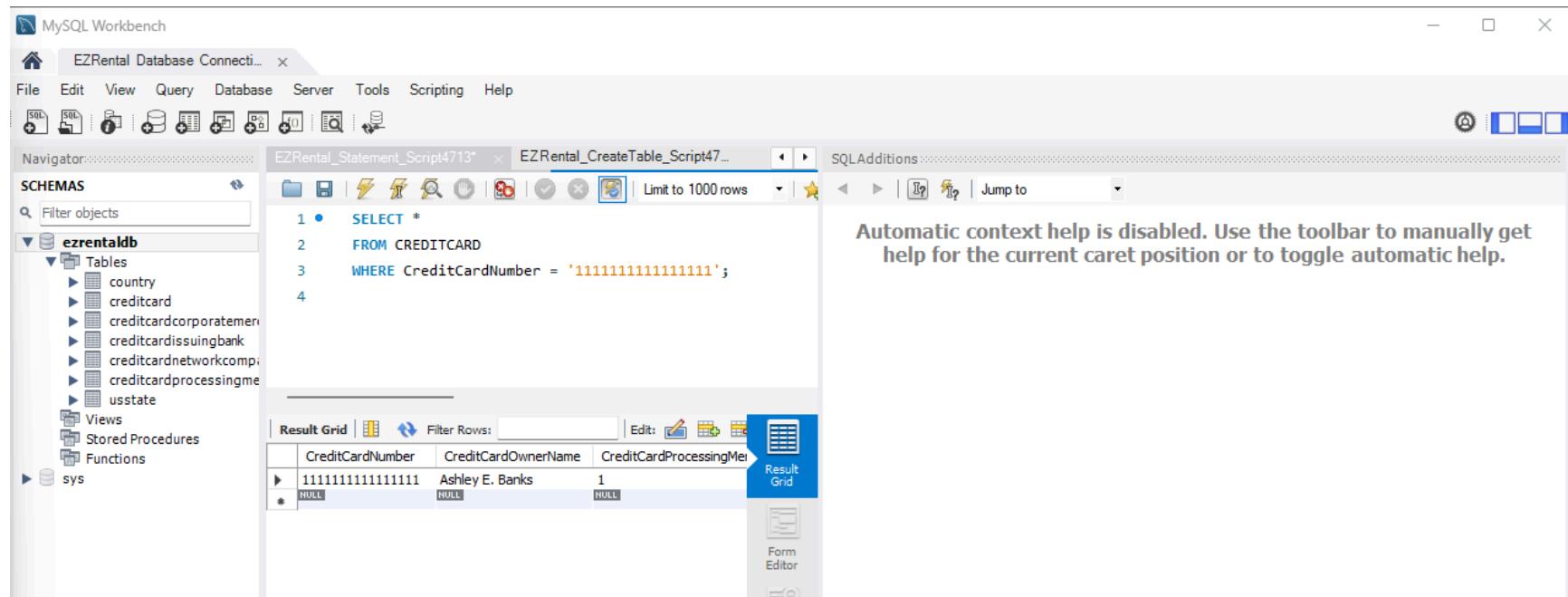
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

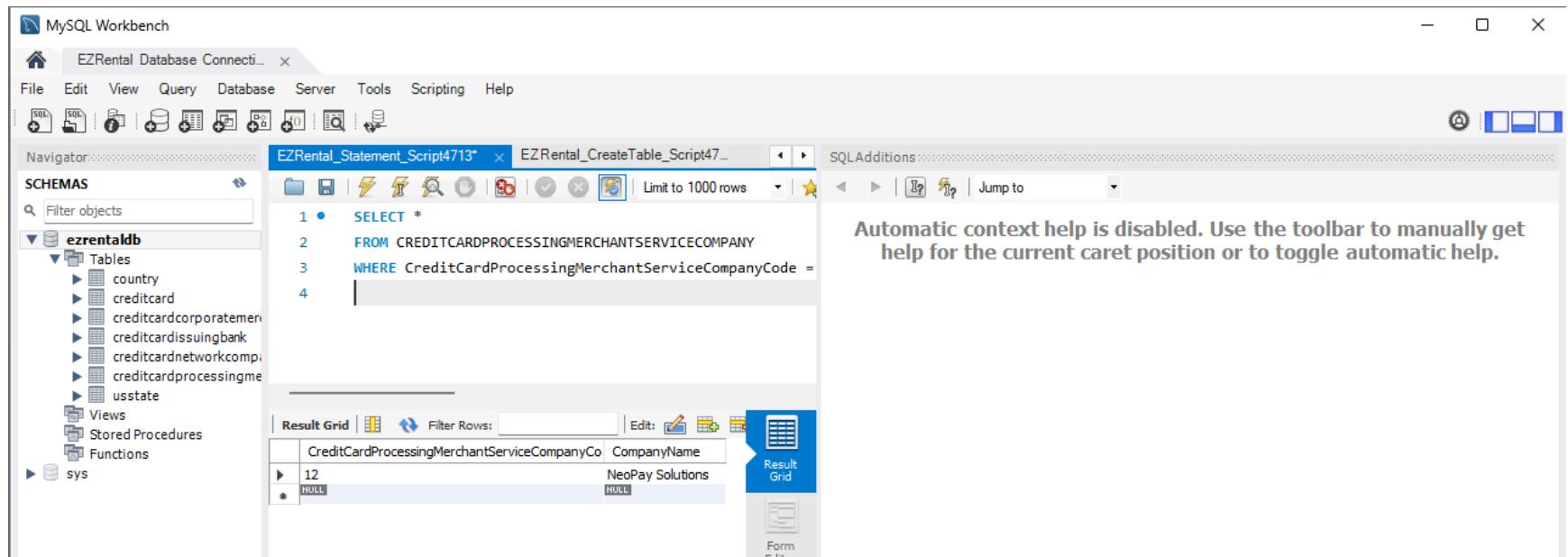
```
26     ON c.CreditCardCorporateMerchantBankCode = cb.Cred...
27     JOIN USState s
28         ON c.StateID = s.StateID
29     JOIN Country cn
30         ON c.CountryID = cn.CountryID
31     WHERE c.CreditCardNumber = '1111111111111111';
32 
```

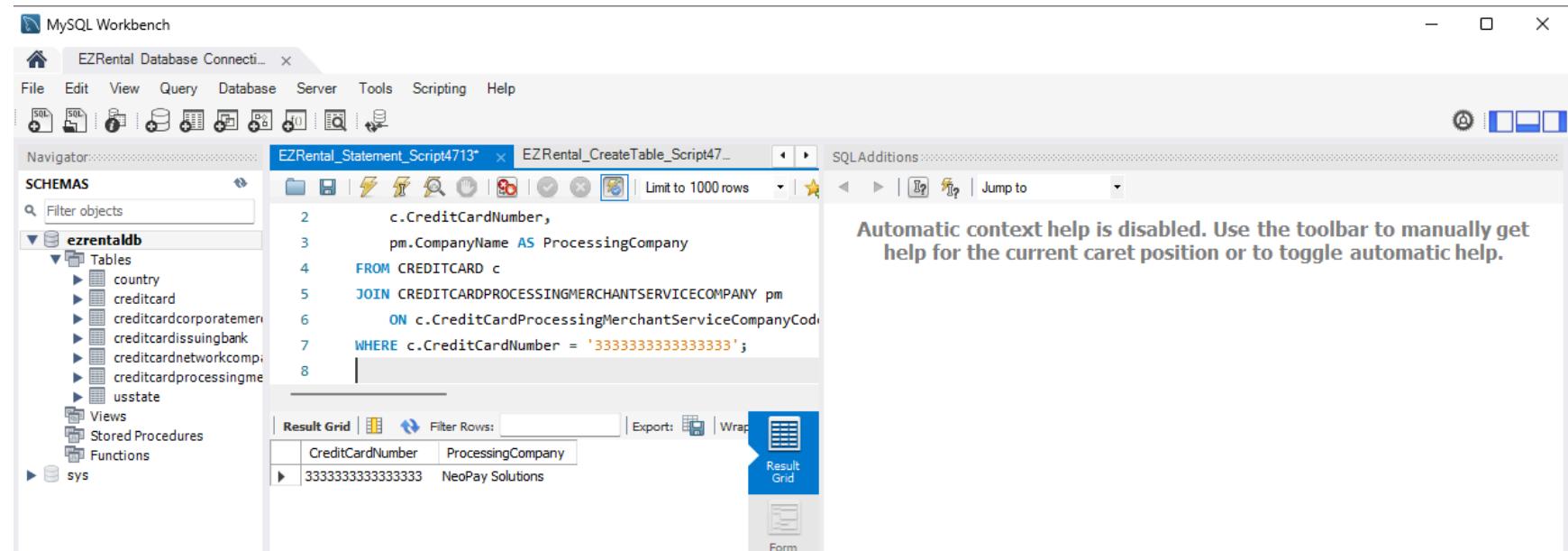
Result Grid | Filter Rows: | Export: | Wrap: | Result Grid | Form Editor | Field Types

CreditCardCorporateMerchantServiceCompanyName	NetworkCompanyName	IssuingBankName
Visa		Chase Bank

Updated Credit Card







## Conclusion

Through this implementation and validation process, the EZRentalAppDB database has been confirmed to be reliable, scalable, and aligned with the operational requirements of EZRental Inc. The test cases successfully demonstrated that the system can handle real-world data interactions such as customer record retrieval, credit card data updates, location-based filtering, and processing company transitions—all while maintaining strict data integrity and compliance with relational database best practices. All foreign key relationships were enforced and respected, with cascading updates and constraint checks validating the integrity of cross-table dependencies. Check constraints, primary key enforcement, and input validation rules were also tested to ensure that the database maintains clean, consistent data across all business processes. The modular structure of the EZRentalAppDB schema allows for flexibility in both current and future system enhancements. Whether supporting front-end search functions, generating financial reports, or integrating with rental service workflows, the database is capable of adapting to additional functional layers such as payment history tracking, fraud detection, and real-time analytics. The project also provided a solid demonstration of how database systems are tested and validated in a production-like environment using tools such as MySQL Workbench, structured queries, and formal documentation. With a fully documented schema, tested data sets, and validated use-case coverage, the EZRentalAppDB is now production-ready and fully equipped to serve as the foundation for EZRental's credit card and customer support infrastructure.