2021

EZRentalPOS Project Design & Implementation

JEFFERSON ROSA

PROJECT 1 – EZRental POS Project Objectives, Definition & Database Design, and Implementation

Executive summary

Throughout this document you will see how the creation of the EZRental POS system was made along with the reasoning for such conclusions. Everything was first derived from deciding the methodology that was used in the project being the Waterfall methodology. Afterword it is all followed from Planning to Analysis to Design to Development, Implementation then finally Operation & Maintenance.

Problem Statement & Objectives

In project statement & objectives is what we plan to achieve by the end of EZRental POS project. The EZRental POS System was designed to allow customers, both retail and corporate, to reserve vehicles for renting like existing in-person or online car rental systems such as Avis, Hertz, Budget, etc. The application was designed to support dozens of major cities around the world. In addition, providing a great user experience both in the physical rental agencies as well as online system 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 Roles & Responsibilities

Without careful planning a project will usually result in failure, thus projects must be planed carefully to be successful. Without every key member doing their part the project will not be able to move forward. Those key members being a Program Manager & Project Manager, Business & Database Analyst, Database Developer, Database Administrator, Object-Oriented-Programming Architect, Full Stack Application Developer, and Full Stack Web Developer.

Project Roles and Responsibilities

- The roles and individuals involved in this project are shown below.

Person	Role	Description
Consultant #1	**Program Manager & Project Manager**	▪ Owner of the project and liaison to Manage the EZRental Inc., the customer.
Consultant #2	**Business & Database Analyst**	▪ A Business/Database Analyst was hired to interview the stakeholders at EZRental Inc. to create the Business Requirements that is the foundation to the database design & implementation.
Consultant #3	**Database Developer**	▪ Using the Normalized Logical Model created by consultant #2 the Data Dictionary, Physical Schema Diagram, and Implement the Database Application for the Auto Rental System was created.
Consultant #4	**Database Administrator**	▪ As the DB Admin, install the DBMS, maintain, and operate the DBMS throughout its lifetime.
Consultant #5	**Object-Oriented-Programming Architect**	▪ An Object-Oriented-Programming Architect was hired by consultant #1 to interview the stakeholders at EZRental Inc., from a client application programming prospective and in addition design the Class/Object model based on interview and analysis results.

Person	Role	Description
Consultant #6	**Full Stack Application Developer**	▪ Object-Oriented-Programming developer implemented the Windows Client application using java technologies & on the database side, implementing stored procedures and supporting the databased team as needed.
Consultant #7	**Full Stack Web Developer**	▪ Object-Oriented-Programming developer & Web Developer implemented the Web-based application using java and other technology to be determine in future & on the database side, implemented stored procedures and supporting the databased team as needed.

Application Business requirements

- A Business Analyst was hired to interview EZRental Inc., project stakeholders and compile the list of the business data requirements for the or *business requirements* in order to gather the necessary data required for the application and database design.

- Below are the *business requirements* captured by the Business Analyst:

Business Requirements

About Us:

EZ-Car Rental is an auto rental company that rents vehicles such as cars, SUVs, minivans & cargo vans. In addition, specialized vehicles such as trucks, motorcycles, etc. We operate in several countries with rental agency locations in the US, Canada, Mexico, UK, Japan & Australia. In each country we operate, multiple rental agencies can exist in a city. For example, New York City has 2 rental agencies in Manhattan, one in Brooklyn and two in Queens, one in each airport. With multiple rental agencies in cities, 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). Also very important the current system is 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 our business new requirements, scalable, and elastic to adopt to business trends and seasonal market changes. Elasticity is very important since we are also faced with a new type of competition; small rental companies that are nimble and can quickly adopt to market changes thus able to provide new offerings that are appealing to customers thus affecting our profits. These smaller competitors are using new technologies that enable them to be nimble and elastic. Figurative speaking "*they are eating our lunch*".

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

Our Agencies:

A rental agency is identified by a unique number *rental agency ID*, *agency name*, *address* that is composed of the following elements: *address line1, address line 2*, *city*, *state*, *zip code* & *country*. In addition, we also need to capture the agency's *phone number*, and *email*.

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 and get special corporate rates for their employees. On the other hand, retail customers are consumers not associated with a company.

To run our business, the application must store the following information for both type of customers (retail & corporate):

- 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*, *city*, *state*, *zip code* & *country*.
- Agency *phone number* & *email* which is required to rent. In addition, the unique *driver license number* and *driver license expiration date*.
- Another very important attribute we need to capture for every customer is the *credit card*. You cannot rent one of our vehicles without a credit card. A *credit card* includes the following components: *credit card number* that uniquely identifies the credit card, credit card *owner name*, *merchant name*, *expiration date*, *billing address* composed of *address line1, address line 2*, *city*, *state*, *zip code* & *country*. Other attributes of credit card are *credit card balance*, *credit card limit* & *activation status* which is true if the credit card is active and can be used or false when disabled.
- Business rules related to a credit card are:
    - A customer can have many credit cards they can use to pay for rental transactions.
    - A credit card can be co-owned by many individuals such a family member or corporate entity the customer works for.

Business Requirements

Our Customers (Cont.):

For our corporate customers only, we must store the following properties: unique *company ID* (we have an ID number for each company), *company name*, company address which includes the elements: *address line1, address line 2*, *city*, *state*, *zip code* & *country*, in addition, *company contact* which is composed of *contact name*, *contact phone number* & *contact email*. Finally, we need to store the *company's daily rental rate* or rate applied to the corporate customers rentals.

Retail customers can opt-in to enrolled in the EZPlus rewards program where they earn points every time they rent and can redeem these points for future rentals. Note that the EZPlus program is optional for retail customers & points are earned only when they rent a vehicle. In addition, retail customers are eligible for special promotional discounts or coupons they can obtain from other businesses and organizations. Therefore, data unique to a retail customer that we need to capture for the promotional discount are: unique random number *discount ID* to uniquely identify a discount, a unique *discount code* or coupon code. and *discount code description*. For the EZPlus rewards program we need to store: unique random *EZPlus ID*, the unique *Ezplus rewards code*, *EZPlus rewards earned points*. Examples of common *discount ID*, *discount code*, *discount code description*, *EZPlus ID*, *EZPlus rewards Code* and *EZPlus earned points* are:

Discount ID	*Discount Code*	*Discount Code Description*
1234..	AAA99700	**AAA Membership Discount - 25% off base rate plus 10% donated for breast cancer research.**
5678..	GOV87569	**Government Employee Discount - 30% off base rate**
9101..	*STA34156*	**State Employee Discount for 25% off base rate**
1213..	VET20551	**Veteran Discount 35% off base rate Plus 10% donation to veteran's family fund.**
Etc..	Etc..	**Etc..**

EZPlus ID	*EZPlus Rewards Code*	*EZPlus Rewards Earned Points*
1234..	EZP90098	**10000**
5678..	EZP10001	**500**
9101..	*EZP64932*	**159000**
1213..	*EZP20051*	**23000**
	Etc..	**Etc..**

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

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

Business Requirements (Cont.)

Our Vehicles:

EZ-Car Rental needs a system to manage their vehicles for renting, maintenance, selling, etc. Vehicles are classified into 4 main types: cars, SUVs, minivans, and cargo vans. These are the vehicles most rented and available at every rental agency. Nevertheless, there are other categories of vehicles available only certain locations such as recreational vehicle, motorcycles etc. No matter what type of vehicle, all vehicle types of vehicles 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*. 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* (e.g., Manual, automatic, Continuously variable transmission (e.g. CVT), Semi-automatic & dual-clutch) and *seat capacity*.
- All vehicles also have a special identifier 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, such as reserved, rented, available, maintenance, not available, transferred, etc. Below Is the list of vehicle status IDs we are currently using and their descriptions:

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

In addition to these attributes shared by all vehicles, the unique characteristics for each of the 4 vehicle types available in all agencies 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 number of luggage they are carrying. For example, a luxury Mercedes E class car has a trunk capacity of 18.5 cubic ft.
- An SUV has a *towing capacity* in pounds and the option of being *All-Wheel-Drive* or not.
- A Minivan has the option of having a *disability option package* or not.
- Finally, a Cargo Van, has a *cargo capacity* in cubic feet volume and *maximum payload* it can hold in pounds.

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. In addition, a reservation or rental can only be for one of these four categories of vehicles not a combination. 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.

In our business, we have the following business rules for our vehicles:

- 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.
- 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.

Business Requirements (Cont.)

Reservation Process:

A vehicle must first be reserved before the vehicle can be rented. 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.

We have the following rules for reserving a vehicle:

- A reservation is not made for a specific vehicle, but to a vehicle rental category. Rental category examples are economy, intermediate, full size, luxury.
- 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.

A 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 cost we use:

Vehicle Rental Category ID	*Vehicle Rental Category Name*	*Category Daily Rental Rate*
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**

We have the following business rule relate to a vehicle and a vehicle rental category:

- A vehicle is a member of a vehicle rental category.
- 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 rules apply to a reservation:

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

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

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

Business Requirements (Cont.)

Reservation Process (Cont.):

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

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

For a reservation we must adhere to the following rules:

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

The Rental Process:

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

For the rental process, the following rules apply:

- A customer rents a vehicle at a rental agency. This means the rental process requires the customer, vehicle, and & rental agency for a rental to be complete.
- During the rental process we may have any of the following 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. Or finally, a vehicle can be picked up at **ANOTHER** rental agency other than what was indicated by the reservation and dropped off at **ANOTHER** rental agency of the reservation.

- ❖ Note that for scenarios 3 & 4, we cannot guarantee that the vehicle rental category of the reservation will be available at the agency other than what was agreed in the reservation. We will do our best to accommodate the change during these scenarios or find another vehicle that will be closed to the original reservation.

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

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

- The *rental agreement ID* that uniquely identifies the rental transaction, *rental pick up date*, *rental pick up time*, *rental drop off date* and *rental drop off time*, *rental pick up odometer value*, *rental drop off odometer value* & *rental total cost* which can be calculated based on selected fuel option, insurance option, vehicle rental category price and other factors.

Business Requirements (Cont.)

The Rental Process (Cont.):

- In addition to the above, customers receive a vehicle with a full tank of gas and customers have the option to return the car on a full tank of gas they purchase or pay a charge to return the car as is, therefore, we need to capture the unique *rental fuel option ID*, *rental fuel option description* and *rental fuel option additional cost*. We currently use the following fuel option IDs, descriptions, and cost:

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.**	**Calculated during car return and based on the current price of a gallon of gas. Price will vary.**
2	**Pay for full tank in advanced at time of rental, return car empty. No refund for unused gas.**	**Calculated during time of car rental and based on current price of a gallon of gas. Price will vary.**

- 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	**Lability 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.):

- Finally, we need to capture the *rental credit card deposit* for a rental. The rental credit card deposit value is calculated based on the **rental period + 25% of the rental period** for any damage or other charges. This deposit is refunded to the customer's credit card when the vehicle is returned in the condition in which it was rented.

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:

- A reservation is made for a rental and the opposite holds true; a rental is based on a reservation.
- 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.
- 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 employees consist of customer service agents who interact with our customer to reserve and rent vehicles. In addition, we have auto specialists who work in our services centers servicing our vehicles. In addition, 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*, *zip code* & *country*. Also, *employee phone*, *employee job title* and *employee email.*

Security & Access:

To access our systems proper security and authentication is required. Only authorized users can have access our agencies Point-0f-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-0f-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, and finally the *employee password* alpha-numeric that is known only to the user, An employee can own one employee user account only, and an employee user account can only be owned by one employee only since the user account represents the identify of that one employee.

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, and finally, the *customer password* that is an alpha-numeric known only to the customer. A customer can own one customer user account only, and a customer user account can only be owned by one customer. For a period, we will need to register customers into our business but the **EZRental.com** web portal may be incomplete, therefore creating a customer user account for a new customer can be optional. We will force the creation of customer user accounts when they login to our portal for the first time.

Conclusion:

The business data listed in this business requirements document is what we need to capture for our business to operate. As our business evolve, additional data will be required. 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 is that the design is modular and scalable for future growth.

Application Development & Technical Requirements

A Business Analyst was hired to interview EZRental Inc., project stakeholders and compile the list of the business data requirements for the or business requirements in order to gather the necessary data required for the application and database design.

- Below are the business requirements captured by the Business Analyst:

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 our 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 is 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 **Java**. For any Web related development, we support JavaScript, React, NodeJs and other standard Web Technologies. We have aligned **Java** & **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-0f-Sales with appropriate conditional access.

Application Development & Technical Requirements (Cont.)

Rental Agency Application Features and Functionalities Requirements:

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

No.	Feature	Functionalities
1	**EZRental** Rental Agency Point-of-Sales (POS) System	▪ Car Rental, Car Return, New Customer Registration & Search Customer Information, Customer Update, Customer Deletion, Customer Listing operations etc.
2	**EZRental** Rental Agency Back-Office Vehicle Inventory Management System	▪ Back-office system meant for employees to perform bulk IN-MEMORY inventory processing or management tasks on vehicles such as adding vehicles to the system, searching for vehicles, updating vehicles etc. ▪ This system is NOT meant for Point-of-Sales, but for the inventory management employees who need to search, add, remove etc., a large/bulk number of vehicles or employees during a session. ▪ Back-office vehicle Management features – Allows inventory personnel and employees to bulk-manage Cars, SUVs, Mini-Vans, Cargo Vans to be searched, added, removed, printed, listed etc.
3	**EZRental** Rental Agency Back-Office Credit Card Management System	▪ 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, Add, Edit & Delete credit card information in the database
4	**EZRental** Rental Agency Back-Office Employee & Customer User Account Management System	▪ The EZRental Customer & Employee User Account Management System is a Back-end system meant for IT ADMINISTRATOR Employees to manage both Employee & Customer USER ACCOUNTS.
5	**EZRental** Rental Agency Desktop Application Security Authentication System	▪ Proper security and authentication must be implemented to make sure only authorized employees can access the Point-0f-Sales, Back-End Management system or any other access to the applications.

Rental Agency Application Graphical User Interface Requirements:

- Graphical User-Interface should be fast rendering and user-friendly workflow.
- Visual screens or forms should be rich in color and appearance and navigation flow should be flexible and easy.
- The following UI controls or data field need to be pre-populated in GUI Screens:

  - **Addresses**

    - Any forms/UI which contains addresses, the STATE & COUNTRY fields should be automatically populated with a list of STATES or COUNTRIES, so the user does not have to manually enter a state or a country and simply select from drop-down list etc.

  - **Discount Codes:**

    - UI screens with customer's DISCOUNT CODE fields should be prepopulated with discount codes. The idea is the user should be able to select the discount to apply to a customer entry from a drop-down list/Combo Box etc. Note that this may or may not include the Discount Code Description on the UI screen as well.
    - Also note that the DISCOUNT CODE VALUES are generated by our Marketing Team and need to be pre-populated in the database before a code can be used. Therefore, the discount codes are prepopulated in the database.
    - Currently, when the Marketing Team generates a new code, they make the request to the database administrator to manually enter an update any new Discount Codes.
    - In the future, we want the application to have the necessary features for the Marketing Team to be able to manage the discount codes. This is not an immediate requirement out of the gate but should be targeted as part of a future upgrade.

Application Development & Technical Requirements

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

- **EZPlus Rewards Codes:**

  - The EZPlus Reward UI screens with customer's EZPLUS REWARDS CODE fields should be prepopulated with the EZPlus Rewards code for the customer is being applied to. The idea is the user should be able to select the EZPLUS REWARD CODE to apply to a customer entry from a drop-down list/Combo Box etc., or be handled by the back-end database.
  - **Important:** The EZPLUS REWARDS CODE VALUES are NOT generated by a business entity in our organization, but AUTOMATICALLY GENERATED by the application on the fly when registering a new customer. This is a different approach compared to the DISCOUNT CODE which are generated by Marketing Team. In this case, the EZPlus Rewards Code values are generated by the application and available via the UI screen to be used or some other method of generation.
  - To finalize this requirement, the idea is the EZPlus Rewards Code should be automatically generated and either appear in the UI Screen or automatically generated in the database.

- **Company Name:**

  - UI screens with corporate customer's COMPANY NAME fields should be prepopulated with the list of corporations that are members of our corporate program, which enables our users to avoid having to manually enter the company name. Note that this may or may not include the Company ID in the UI Screen which is a unique number with business value that we assign to each company.
  - Note that the company names. Company ids and other company data are managed by our Corporate Sales Team and need to be pre-populated in the database before any corporate customer processing can be made. Therefore, the company information is prepopulated in the database.
  - Currently, when the Corporate Sales Team adds a new corporation or company into the program, they make the request to the database administrator to manually enter and add the new company to the database.
  - In the future we want the application to have the necessary features for the Corporate Sales Team to have the functionality to manage the data of our corporate companies via the application. This is not an immediate requirement out of the gate but should be targeted as part of a future upgrade.

- **Vehicle Status:**

  - UI screens for vehicle inventory management, VEHICLE STATUS field should be prepopulated with the list of vehicle status. Based on the business requirements, the current list of vehicle status is listed in table below:

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

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

- **Rental Agency:**

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

Application Development & Technical Requirements

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

- **Vehicle Rental Category:**

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

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

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

- **Reservation Status:**

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

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

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

Application Development & Technical Requirements

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

- **Rental Status:**

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

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**

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

- **Rental Fuel Option:**

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

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.**	**Calculated during car return and based on the current price of a gallon of gas. Price will vary.**
2	**Pay for full tank in advanced at time of rental, return car empty. No refund for unused gas.**	**Calculated during time of car rental and based on current price of a gallon of gas. Price will vary.**

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

Application Development & Technical Requirements

- **Rental Insurance Option:**

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

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	**Lability 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 (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 **Java**. We have aligned **Java** & **Web** developers that have been assigned to assist, support, operated 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:

The list of features and functionalities that we have compiled for the customer serf-service Web Portal are listed in the table below:

No.	Feature	Functionalities
1	**EZRental.com** Customer Web Portal	▪ Front-end WEB INTERFACE SCREENS & features used by customers via our web portal EZRentalCar.com to reserve a vehicle for rental and manage their account online. ▪ Features include: Search & reserve a car for rental, register as a new customer, search/view their account information, update their account etc.
2	**EZRental.com** Customer Web Portal Application Security Authentication System	▪ Proper security and authentication must be implemented to make sure only our customer can access the web portal to use the application.

Web Portal Application Web Pages User Interface Requirements:

The web pages graphical UI requirements are listed below:

- The GUI requirements for the web pages are like those functionalities of the Rental Agency Application that are found on the web site for example Search & reserve a car for rental, register as a new customer, search/view their account information, update their account etc.
- The design and graphics of the application should be appealing to customers and a smooth and fluent workflow.
- The following UI controls or data field need to be pre-populated in GUI Screens:

  - **Addresses**

    - Any web-page UI which contains addresses, the STATE & COUNTRY fields should be automatically populated with a list of STATES or COUNTRIES, so the user does not have to manually enter a state or a country and simply select from drop-down list etc.

  - **Discount Codes:**

    - Web pages with customer's DISCOUNT CODE fields should be a text box that allows the customer to ADD/APPLY the discount codes to redeem the coupon.

Application Development & Technical Requirements

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

- **EZPlus Rewards Codes:**

  - The EZPlus Reward web page screens with customer's EZPLUS REWARDS CODE fields should be prepopulated with the EZPlus Rewards code for the customer is being applied to. The idea is the user should be able to select the EZPLUS REWARD CODE to apply to a customer entry from a drop-down list/Combo Box etc., or be handled by the back-end database.
  - **Important:** The EZPLUS REWARDS CODE VALUES are NOT generated by a business entity in our organization, but AUTOMATICALLY GENERATED by the application on the fly when registering a new customer. The EZPlus Rewards Code values are generated by the application and available via the UI screen to be used or some other method of generation.
  - To finalize this requirement, the idea is the EZPlus Rewards Code should be automatically generated and either appear in the UI Screen or automatically generated in the database.

- **Rental Agency:**

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

- **Vehicle Rental Category:**

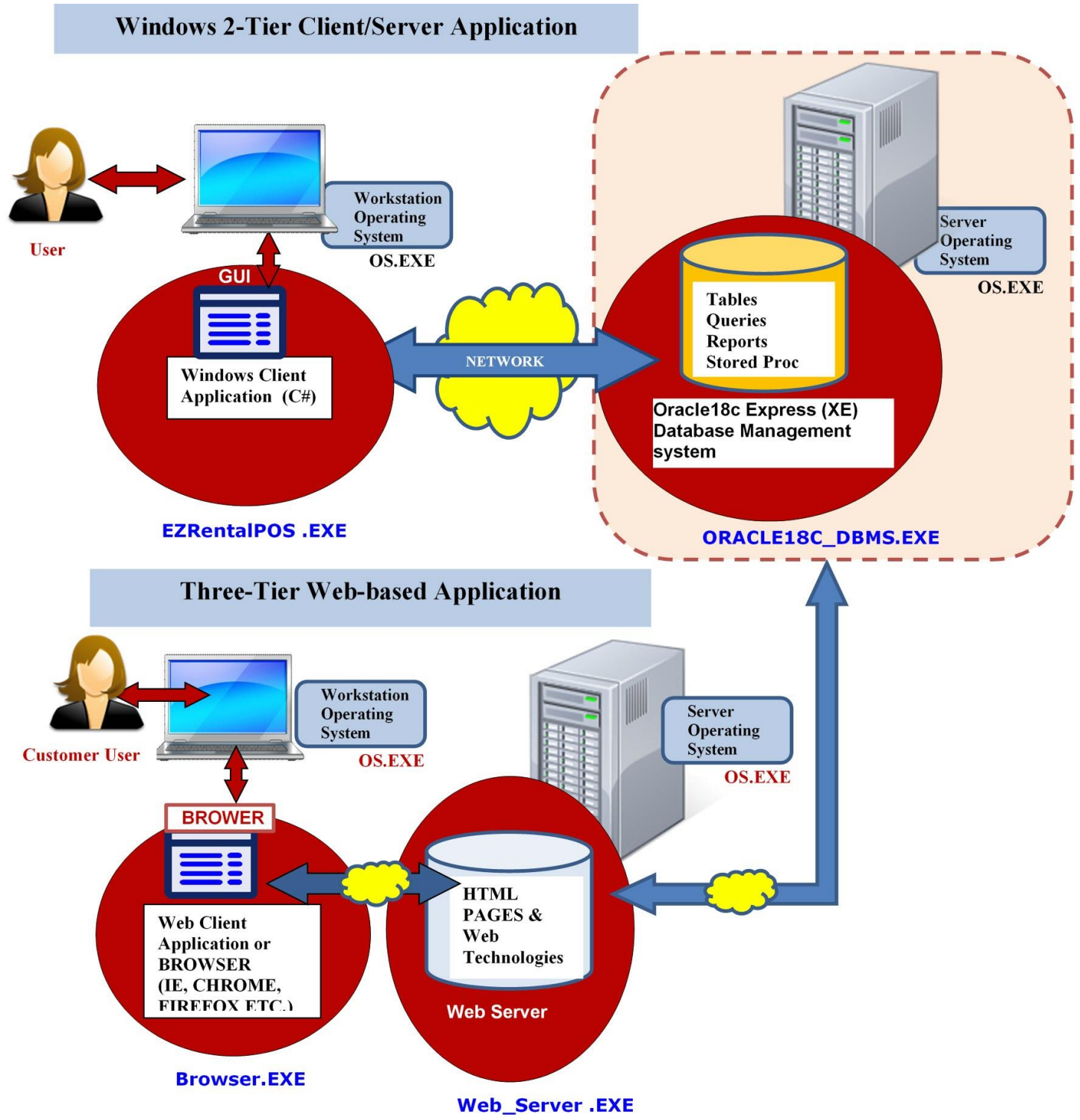
  - Web pages that require the use of the VEHICLE RENTAL CATEGORY fields, must be prepopulated with the list of vehicle rental categories. Based on the business requirements, the current list of vehicle rental categories is as follows:

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

Application Physical Technical Architecture

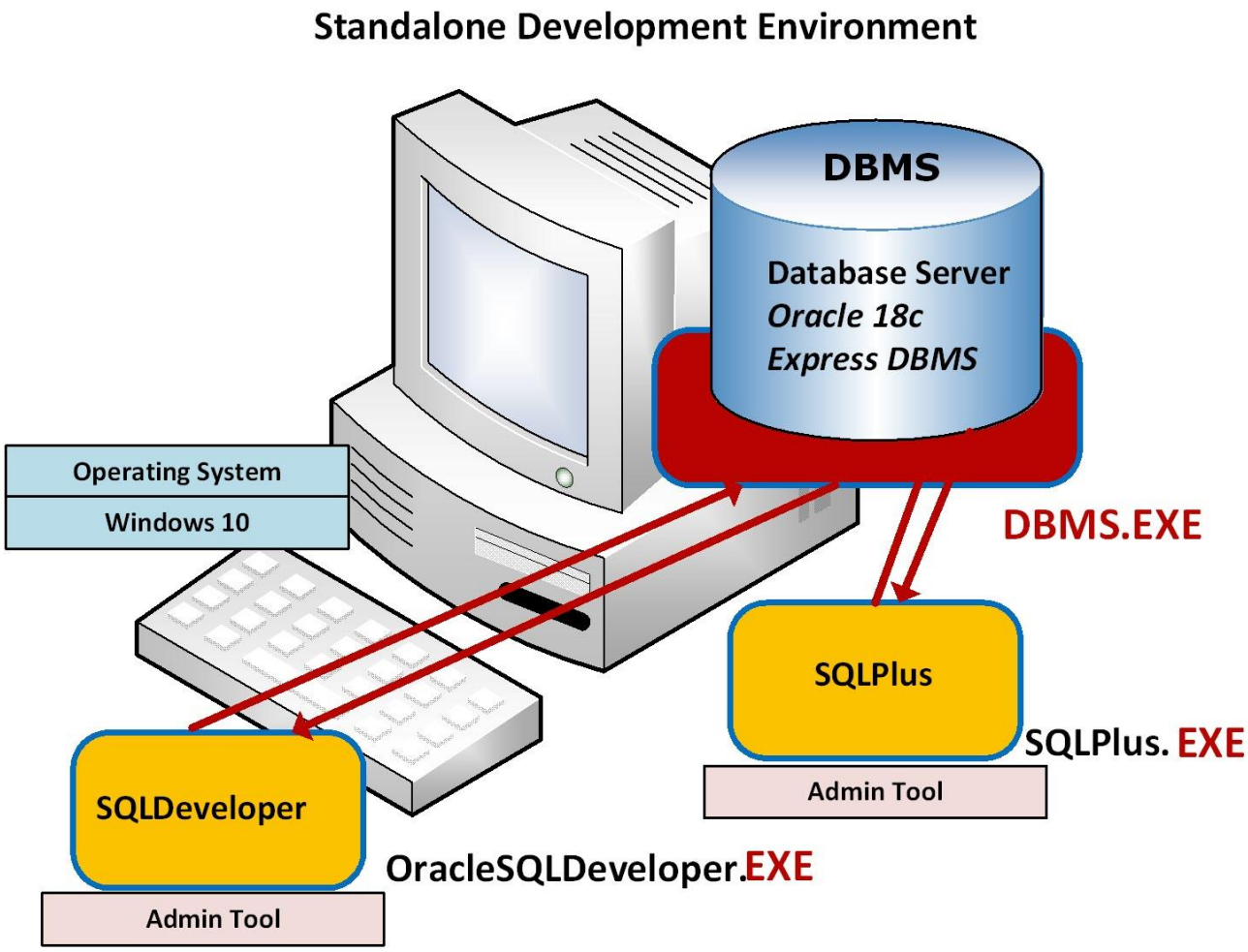
Two-Tiered Windows-Client Client/Server Application Front-line workers such as customer service desk in store branches, airports etc., in addition to other support personnel such as service centers employees, inventory etc., used this Windows-based client application for speed and performance. The Three-Tiered Web-based Client/Server –Web Application named EZRentalCar.com, targeted customers who rented a car online, in addition to the day-to-day activities of our business and office workers personnel via a Browser Application.

Below is a pictorial diagram of this multi-component client/server architecture:



Database Management Physical Architecture

The Database Management System (DBMS) in scope is Oracle Server 18c Express Edition (EX) since this is the standard DBMS used at EZRental Inc. Oracle Server 18c Express Edition (EX) was installed into personal computer along with Oracle SQL Developer to create the following database development Environment:

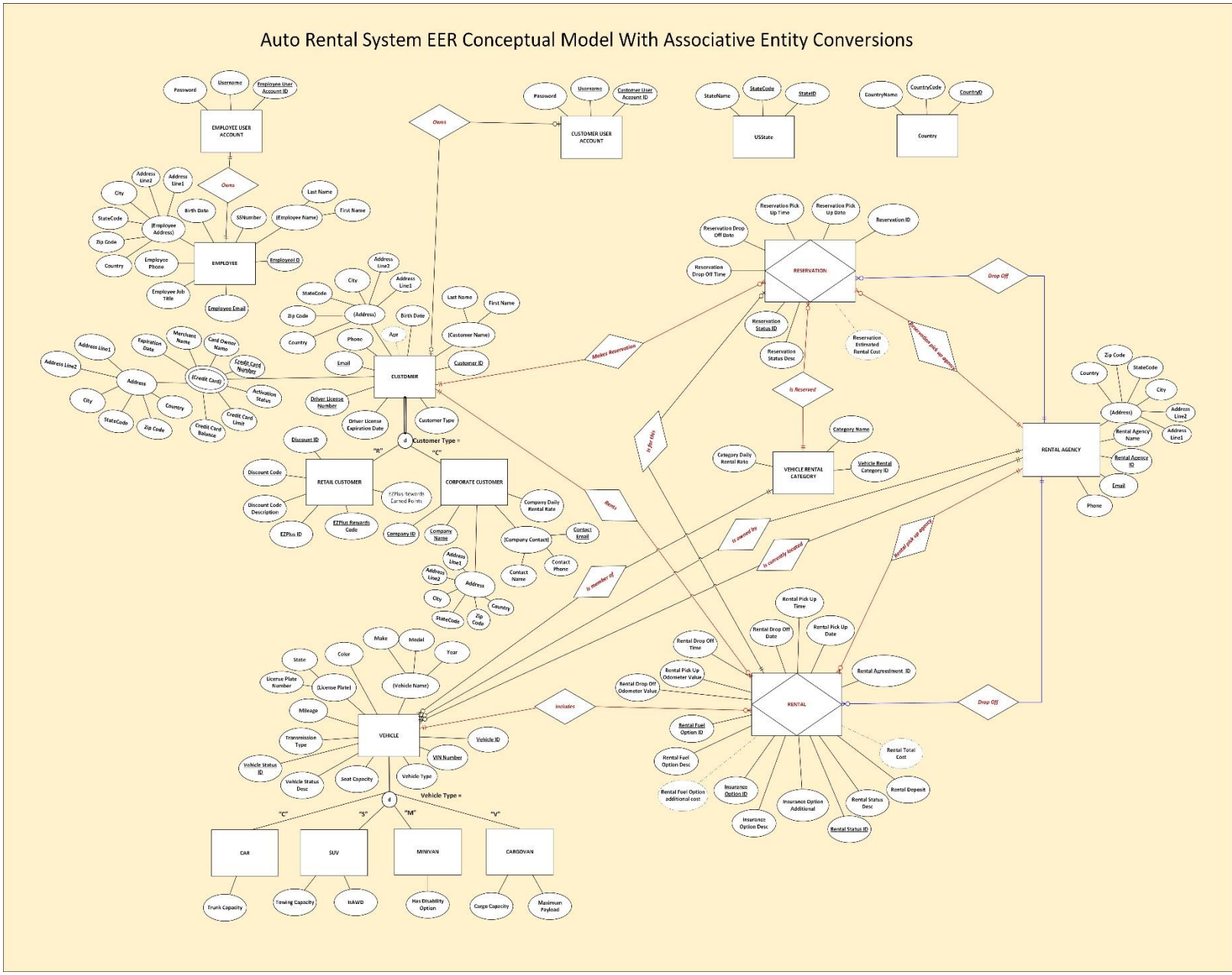


Project Management Methodology

A combination of Waterfall Project Management Methodology & Agile Project Management Methodology was used for the Auto Rental Management System. A project management is used to have a structured method, approach, process, technique, tools etc., to accomplish a project successfully, being flexible to unexpected changes and delivered within expected timelines.

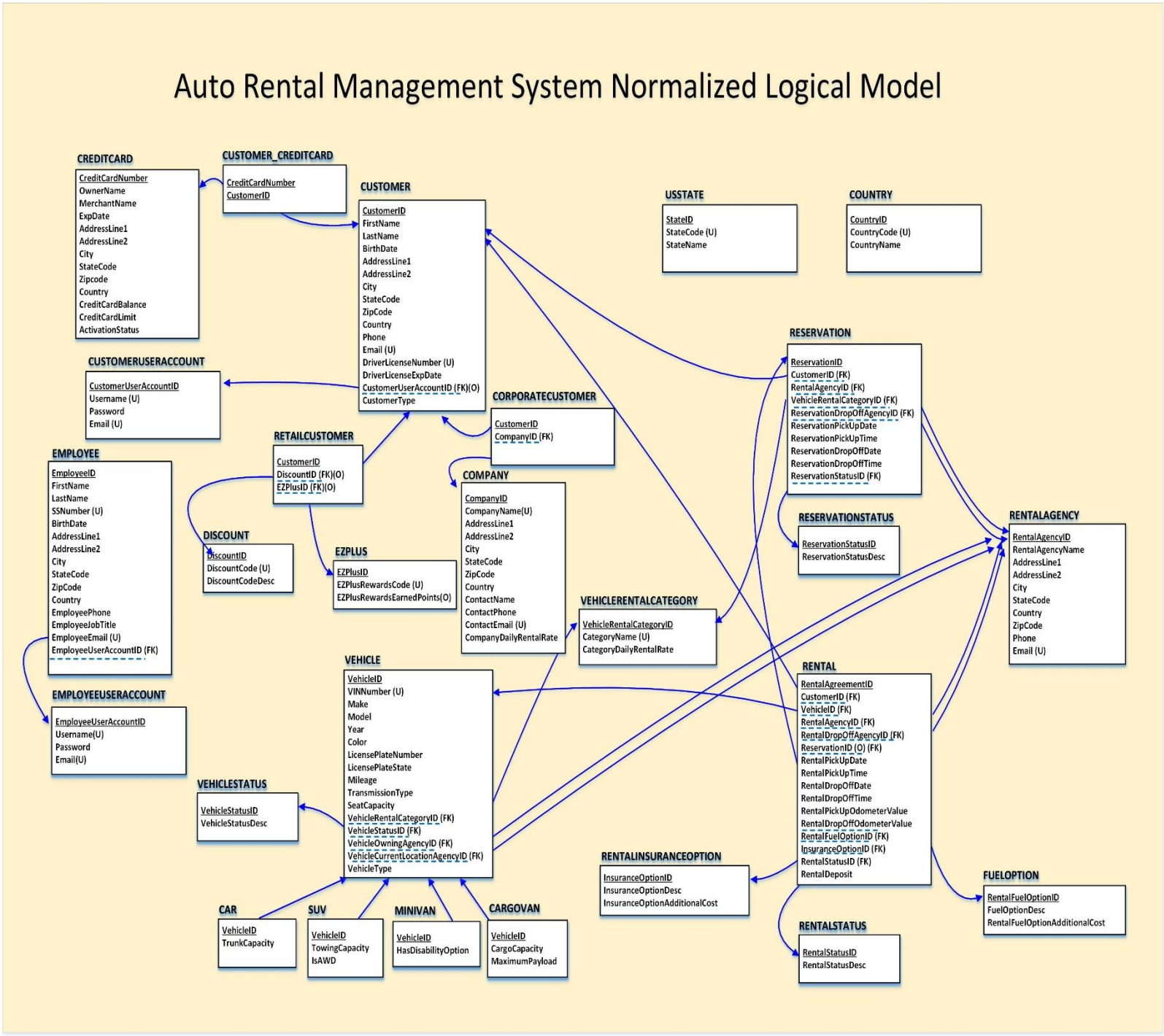
ER/EER Conceptual Model

Below is the rendition of the EER Model of the Business Requirements, with standard CHEN notation and including Associative entities.



Normalized Logical Model

Below is the rendition of the Normalized Logical Model based on EER diagram of the previous section.



Physical Model Data Dictionary

Below you will find the data dictionary for all 27 tables.

CUSTOMER CREDITCARD						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
CardNumber	String	Varchar2(16)	Y	16	PRIMARY KEY	Primary Key referencing CreditCard Table, CardNumber
CustomerID	Number	NUMBER(6)	Y	6	PRIMARY KEY	Primary Key That will reference CustomerID in Customer Table

CUSTOMER						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
CustomerID	NUMBER	NUMBER GENERATED ALWAYS AS IDENTITY	Y	Default size of data type	PRIMARY KEY	Auto-generated number IDENTITY primary key starting at 1. Size is maximum value of NUMBER data type.
FirstName	STRING	VARCHAR2(50)	Y	50	NOT NULL	First name of customer
LastName	STRING	VARCHAR2(50)	Y	50	NOT NULL	Last name of customer
BirthDate	DATE	DATE	Y	N/A	NOT NULL	Customer's birth date
AddressLine1	STRING	VARCHAR2(50)	Y	50	NOT NULL	Customer's 1st Address Line
AddressLine2	STRING	VARCHAR2(50)	Y	50	NOT NULL	Customer's 2nd Address Line
City	STRING	VARCHAR2(30)	Y	30	NOT NULL	Customer's City
StateCode	CHARACTER	CHAR(2)	Y	2	NOT NULL	Customer's State
ZipCode	STRING	VARCHAR2(10)	Y	10	NOT NULL	Customer's zipcode
Country	STRING	VARCHAR2(50)	Y	50	NOT NULL	Customer's country
Phone	STRING	VARCHAR2(20)	Y	20	NOT NULL	Customer's Phone
Email (U)	STRING	VARCHAR2(100)	Y	100	UNIQUE NOT NULL	Customer's email
DriverLicenseNumber (U)	STRING	VARCHAR2(16)	Y	16	UNIQUE NOT NULL	Customer's driver license number
DriverLicenseExpDate	DATE	DATE	Y	N/A	NOT NULL	Customer's driver license experiation date
CustomerUserAccountID (FK)	SYNTAX	RAW(16)	Y	16	NULL FORIEGN KEY	Foreign Key Referencing CustomerUserAccount table UserAccountID
CustomerType	CHARACTER	CHAR(1)	Y	1	NOT NULL	What type of customer is it retail or corporate

CREDITCARD						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
CardNumber	String	VARCHAR2(16)	Y	16	PRIMARY KEY	CardNumber, Unique identifier for CustomerCreditCard. Has Business meaning. Primary Key
OwnerName	STRING	VARCHAR2(100)	Y	100	NOT NULL	Creditcard's owner name
MerchantName	STRING	VARCHAR2(100)	Y	100	NOT NULL	Creditcard's merchant name
ExpDate	DATE	DATE	Y	N/A	NOT NULL	Creditcard's expiration date
AddressLine1	STRING	VARCHAR2(50)	Y	50	NOT NULL	Creditcard's 1st Address Line
AddressLine2	STRING	VARCHAR2(50)	Y	50	NOT NULL	CreditCards's 2nd Address Line
City	STRING	VARCHAR2(30)	Y	30	NOT NULL	Creditcard's city
StateCode	CHARACTER	CHAR(2)	Y	2	NOT NULL	Creditcard's state
Zipcode	STRING	VARCHAR2(10)	Y	10	NOT NULL	Creditcard's zip code
Country	STRING	VARCHAR2(50)	Y	50	NOT NULL	Creditcard's country
CreditLimit	NUMBER	NUMBER(6)	Y	6	NOT NULL	Creditcard's credit limit
ActivationStatus	NUMER	NUMBER(1)	Y	1	NOT NULL	Activation status 1 for true 0 for false

CUSTOMERUSERACCOUNT						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
UserAccountID	SYNTAX	RAW(16)	Y	16	PRIMARY KEY	Unique identifier for customer user account
Username	STRING	VARCHAR2(64)	Y	64	UNIQUE NOT NULL	The customer's user account username
Password	STRING	VARCHAR2(100)	Y	100	NOT NULL	The password to the customer's suer account
Email	STRING	VARCHAR2(100)	Y	100	UNIQUE NOT NULL	The email tied to the account of the user account

CORPORATECUSTOMER						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
CustomerID(FK)	NUMBER	NUMBER(6)	Y	6	FOREIGN KEY NOT NULL	Foreign Key Referencing customer table CustomerID
CompanyID(FK)	NUMBER	NUMBER(6)	Y	6	FOREIGN KEY NOT NULL	Foreign Key Referencing Company table CompanyID

COMPANY						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
CompanyID	Number	NUMBER(6)	Y	6	PRIMARY KEY	CompanyID, Unique identifier for Company. Has Business meaning. Primary Key
CompanyName	STRING	VARCHAR2(80)	Y	80	NOT NULL	Company's Name
AddressLine1	STRING	VARCHAR2(50)	Y	50	NOT NULL	Company's 1st Address
AddressLine2	STRING	VARCHAR2(50)	Y	50	NOT NULL	Company's 2nd address
City	CHAR	VARCHAR2(30)	Y	30	NOT NULL	Company's City
StateCode		VARCHAR2(2)	Y	2	NOT NULL	Company's StateCode
ZipCode	STRING	VARCHAR2(10)	Y	10	NOT NULL	Company's ZipCode
Country	STRING	VARCHAR2(50)	Y	50	NOT NULL	Company's Country
ContactName	STRING	VARCHAR2(100)	Y	100	NOT NULL	Company's contact name
ContactPhone	STRING	VARCHAR2(20)	Y	20	NOT NULL	Company's phone number
ContactEmail	STRING	VARCHAR2(100)	Y	100	UNIQUE NOT NULL	Company's Email
CompanyDailyRentalRate	NUMBER	NUMBER(5,2)	Y	5	NOT NULL	Daily rental of rate for someone who is a customer but works in the company

DISCOUNT						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
DiscountID	NUMBER	NUMBER GENERATED ALWAYS AS IDENTITY	Y	Default size of data type	PRIMARY KEY	Auto-generated number IDENTITY primary key starting at 1. Size is maximum value of NUMBER data type.
DiscountCode	STRING	VARCHAR2(8)	Y	8	UNIQUE NOT NULL	A discount code for discounts
DiscountCodeDesc	STRING	VARCHAR2(30)	Y	30	NOT NULL	Description of the discount

EZPLUS						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
EZPlusID	NUMBER	NUMBER GENERATED ALWAYS AS IDENTITY	Y	Default size of data type	PRIMARY KEY	Auto-generated number IDENTITY primary key starting at 1. Size is maximum value of NUMBER data type.
EZPlusRewardsCode	STRING	VARCHAR2(8)	Y	8	UNIQUE NOT NULL	Unique EZPlusRewardsCode

EZPlusRewardsEarnedPoints	NUMBER	NUMBER(6)	Y	6	NULL	Amount of Ez plus reward points earned in your ezplus account

RETAILCUSTOMER						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
CustomerID (PK)	NUMBER	NUMBER(6)	Y	6	PRIMARY KEY NOT NULL	Primary Key, and Foreign Key Referencing customerID customer table
DiscountID (FK)	STRING	NUMBER(8)	Y	8	FOREIGN KEY NULL	Foreign key referencing discount table discountID
EZPlusID (FK)	NUMBER	NUMBER(6)	Y	6	FOREIGN KEY NULL	Foreign key referencing ezplus table ezplusId

EMPLOYEEUSERACCOUNT						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
UserAccountID	CHARACTER	CHAR(32)	Y	32	PRIMARY KEY NOT NULL	Employe User Account Unique Identifier
UserName	STRING	VARCHAR2(64)	Y	64	UNIQUE NOT NULL	Employee user account's name
Password	STRING	VARCHAR2(100)	Y	100	NOT NULL	Password of the employee user account
Email	STRING	VARCHAR2(100)	Y	100	UNIQUE NOT NULL	Unique Email of the Employee User Account

EMPLOYEE						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
EmployeeID	NUMBER	NUMBER GENERATED ALWAYS AS IDENTITY	Y	Default size of data type	PRIMARY KEY	Auto-generated number IDENTITY primary key starting at 1. Size is maximum value of NUMBER data type.
EmployeeFullName	STRING	VARCHAR2(50)	Y	50	NOT NULL	Employee's full name
EmployeeLastName	STRING	VARCHAR2(50)	Y	50	NOT NULL	Employee's last name
SSNumber	STRING	VARCHAR2(20)	Y	20	UNIQUE NOT NULL	Employee's Social security number
AddressLine1	STRING	VARCHAR2(50)	Y	50	NOT NULL	1st Address of employee
AddressLine2	STRING	VARCHAR2(50)	Y	50	NOT NULL	2nd Address of the employee
City	STRING	VARCHAR2(30)	Y	30	NOT NULL	City of employee
StateCode	CHARACTER	CHAR(2)	Y	2	NOT NULL	State of employee
Zipcode	STRING	VARCHAR2(10)	Y	10	NOT NULL	Zipcode of employee
Country	STRING	VARCHAR2(50)	Y	50	NOT NULL	Country of employee
EmployeePhone	STRING	VARCHAR2(20)	Y	20	NOT NULL	Employee's phone number
EmployeeJobTitle	STRING	VARCHAR2(25)	Y	25	NOT NULL	Employee's job title
EmployeeEmail (U)	STRING	VARCHAR2(100)	Y	100	UNIQUE NOT NULL	Employee's email address
EmployeeUserAccountID (FK)	CHARACTER	CHAR(32)	Y	32	FOREIGN KEY NOT NULL	Foreign key referencing employeeuseraccount table useraccountID

VEHICLESTATUS						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
VehicleStatusID	NUMBER	NUMBER(1)	Y	1	PRIMARY KEY	VehicleStatusID, Unique identifier for VehicleStatus. Has Business meaning. Primary Key
VehicleStatusDesc	STRING	VARCHAR2(20)	Y	20	NOT NULL	Description of the current vehicle status

VEHICLE						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
VehicleID	NUMBER	NUMBER GENERATED ALWAYS AS IDENTITY	Y	Default size of data type	PRIMARY KEY	Auto-generated number IDENTITY primary key starting at 1. Size is maximum value of NUMBER data type.
VINNumber (U)	STRING	VARCHAR2(17)	Y	17	UNIQUE NOT NULL	Vehilce Identification Number for a specific automobile
Make	STRING	VARCHAR2(40)	Y	40	NOT NULL	Brand of the vehicle
Model	STRING	VARCHAR2(40)	Y	40	NOT NULL	Model the name of the vehicle
Year	DATE	DATE	Y	N/A	NOT NULL	Year vehicle was made
Color	STRING	VARCHAR2(25)	Y	25	NOT NULL	Color of the vehicle
LicensePlateNumber	STRING	VARCHAR2(8)	Y	8	NOT NULL	License plate number of the vehicle
LicensePlateState	CHARACTER	Char(2)	Y	2	NOT NULL	Vehicles license plate stat
Mileage	NUMBER	NUMBER(9)	Y	9	NOT NULL	Mileage of the vehicle
TransmissionType	STRING	VARCHAR2(40)	Y	40	NOT NULL	Type of transmission type of the vehicle ex manual,automatic,contion sly variable transmission, ect.
SeatCapacity	NUMBER	NUMBER(1)	Y	1	NOT NULL	Number of seats in the vehicle
VehicleRentalCategoryID (FK)	NUMBER	NUMBER(2)	Y	2	NOT NULL FOREIGN KEY	VehicleRentalCategory foreign key referencing vehiclerentalcategory tabl vehiclerentalcategoryID
VehicleStatusID (FK)	NUMBER	NUMBER(1)	Y	1	NOT NULL FOREIGN KEY	Foreign key referencing vehiclestatus table vehiclestatusID
VehicleOwningAgencyID (FK)	NUMBER	NUMBER(6)	Y	6	NOT NULL FOREIGN KEY	Foreign key referencing rental agency table RentalAgencyID
VehicleCurrentLocationAgency ID (FK)	NUMBER	NUMBER(6)	Y	6	NOT NULL FOREIGN KEY	Foreign key referencing RentalAgency table RentalAgencyID
VehicleType	CHARACTER	CHAR(1)	Y	1	NOT NULL	Type of vehicle wether it is a Car, Suv, Minivan or CargoVan

RENTALAGENCY						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
RentalAgencyID	NUMBER	NUMBER(6)	Y	6	PRIMARY KEY	RentalAgencyID, Unique identifier for RentalAgency. Has Business meaning. Primary Key
RentalAgencyName	STRING	VARCHAR2(50)	Y	50	NOT NULL	Name of the rental agency
AddressLine1	STRING	VARCHAR2(50)	Y	50	NOT NULL	1st Address Line of the rental agency
AddressLine2	STRING	VARCHAR2(50)	Y	50	NOT NULL	2nd Address Line of the rental agency
City	STRING	VARCHAR2(30)	Y	30	NOT NULL	City of rental agency
StateCode	CHARACTER	CHAR(2)	Y	2	NOT NULL	State of rental agency
Country	STRING	VARCHAR2(50)	Y	50	NOT NULL	Country of rental agency
ZipCode	STRING	VARCHAR2(10)	Y	10	NOT NULL	ZipCode of rental agency
Phone	STRING	VARCHAR2(20)	Y	20	NOT NULL	Phone of rental Agency
Email	STRING	VARCHAR2(100)	Y	100	UNIQUE NOT NULL	Email of the Rental Agency

CAR						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
VehicleID(FK)	NUMBER	NUMBER(6)	Y	6	FOREIGN KEY NOT NULL	Foreign Key Referencing Vehicle table VehicleID
TrunkCapacity	NUMBER	NUMBER(4,2)	Y	4	NOT NULL	Capacity of the trunk of the car in cubit feet volume

SUV						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
VehicleID(FK)	NUMBER	NUMBER(6)	Y	6	NOT NULL FOREIGN KEY	Foreign Key Referencing Vehicle table VehicleID
TowingCapacity	NUMBER	NUMBER(4)	Y	4	NOT NULL	Towing capacity of SUV in pounds
IsAWD	NUMBER	NUMBER(1)	Y	1	NOT NULL	1 = True 0 = False for if the SUV is All wheel drive

MINIVAN						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
VehicleID(FK)	NUMBER	NUMBER(6)	Y	6	NOT NULL FOREIGN KEY	Foreign Key Referencing Vehicle table VehicleID

HasDisabilityOption	NUMBER	NUMBER(1)	Y	1	NOT NULL	1 = True 0 = False if the vehicle has a disability option.

CARGOVAN						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
VehicleID(FK)	NUMBER	NUMBER(6)	Y	6	NOT NULL FOREIGN KEY	Foreign Key Referencing Vehicle table VehicleID
CargoCapacity	NUMBER	NUMBER(5,2)	Y	5	NOT NULL	Cargo capacity of vehicle in cubit feet
MaximumPayload	NUMBER	NUMBER(4)	Y	4	NOT NULL	Maximum payload of car in pounds

VEHICLERENTALCATEGORY						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
VehicleRentalCategoryID	NUMBER	NUMBER(2)	Y	2	PRIMARY KEY	VehicleRentalCategoryID, Unique identifier for VehicleRentalCategory. Has Business meaning. Primary Key
CategoryName	STRING	VARCHAR2(30)	Y	30	UNIQUE NOT NULL	UNIQUE Name of vehicle rental category such as car economic,car-compact,ect
CategoryDailyRentalRate	NUMBER	NUMBER(5,2)	Y	5	NOT NULL	Category daily rental rate

RESERVATIONSTATUS						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
ReservationStatusID	NUMBER	NUMBER(1)	Y	1	PRIMARY KEY	ReservationStatusID, Unique identifier for ReservationStatus. Has Business meaning. Primary Key
ReservationStatusDesc	STRING	VARCHAR2(20)	Y	20	NOT NULL	Description of reservation status

RESERVATION						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
ReservationID	NUMBER	NUMBER(6)	Y	6	PRIMARY KEY	ReservationID, Unique identifier for Reservation. Has Business meaning. Primary Key
CustomerID(FK)	NUMBER	NUMBER(6)	Y	6	FOREIGN KEY NOT NULL	Foreign Key Referencing Customer table Customer ID
RentalAgencyID(FK)	NUMBER	NUMBER(6)	Y	6	FOREIGN KEY NOT NULL	Foreign Key Referencing RentalAgency Table RentalAgencyID
VehicleRentalCategoryID(FK)	NUMBER	NUMBER(2)	Y	2	FOREIGN KEY NOT NULL	Foreign Key Referencing VehiclerentalCategory Table VehicleRentalCategoryID
ReservationDropOffAgency(FK)	NUMBER	NUMBER(6)	Y	6	FOREIGN KEY NOT NULL	Foreign Key referencing RentalAgency table RentalAgencyID
ReservationPickUplDate	DATE	DATE	Y	N/A	NOT NULL	Reservation pick of date of vehicle
ReservationPickUpTime	NUMBER	NUMBER(4)	Y	4	NOT NULL	Time of pick up of vehicle
ReservationDropOffDate	DATE	DATE	Y	N/A	NOT NULL	Date vehicle must be drop off
ReservationDropOffTime	NUMBER	NUMBER(4)	Y	4	NOT NULL	Drop of time of vehicle
ReservationStatusID(FK)	NUMBER	NUMBER(1)	Y	1	FOREIGN KEY NOT NULL	Foreign Key referencing ReservationStatus Table ReservationStatusID

RENTALINSURANCEOPTION						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
RentalInsuranceOptionID	NUMBER	NUMBER(1)	Y	1	PRIMARY KEY	RentalInsuranceOption ID, Unique identifier for RentalInsuranceOption . Has Business meaning. Primary Key
RentalInsuranceOptionDesc	STRING	VARCHAR2 (200)	Y	200	NOT NULL	Description of rental insurance

RentalInsuranceOptionAdditionalCost	NUMBER	NUMBER(4, 2)	Y	4	NOT NULL	Rental insurance option additional cost per day

RENTALSTATUS						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
RentalStatusID	NUMBER	NUMBER(1)	Y	1	PRIMARY KEY	RentalStatusID, Unique identifier for RentalStatus. Has Business meaning. Primary Key
RentalStatusDesc	STRING	VARCHAR2(50)	Y	50	NOT NULL	Description of the rental status. Such as if it was picked up on schedule and dropped off as scheduled.

FUELOPTION						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
FuelOptionID	NUMBER	NUMBER(1)	Y	1	PRIMARY KEY	FuelOptionID, Unique identifier for FuelOption. Has Business meaning. Primary Key
FuelOptionDesc	STRING	VARCHAR2(100)	Y	100	NOT NULL	Description of the fuel option

RENTAL						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
RentalAgreementID	NUMBER	NUMBER(6)	Y	6	PRIMARY KEY	RentalAgreementID, Unique identifier for Rental. Has Business meaning. Primary Key
CustomerID(FK)	NUMBER	NUMBER(6)	Y	6	NOT NULL FOREIGN KEY	Foreign Key referencing Customer Table CustomerID
VehicleID(FK)	NUMBER	NUMBER(6)	Y	6	NOT NULL FOREIGN KEY	Foreign Key referencing Vehicle Table VehicleID
RentalAgencyID (FK)	NUMBER	NUMBER(6)	Y	6	NOT NULL FOREIGN KEY	Foreign Key referencing RentalAgency table RentalAgencyID
RentalDropOffAgencyID (FK)	NUMBER	NUMBER(6)	Y	6	NOT NULL FOREIGN KEY	Foreign Key referencing RentalAgency table RentalAgencyID
ReservationID (FK)	NUMBER	NUMBER(6)	N	6	NULL FOREIGN KEY	Foreign Key referencing Reservation table ReservationID
RentalPickUpDate	DATE	DATE	Y	N/A	NOT NULL	Rental vehicle pick up date
RentalPickUpTime	NUMBER	NUMBER(4)	Y	4	NOT NULL	Rental vehicle pick up time
RentalDropOffDate	DATE	DATE	Y	N/A	NOT NULL	Drop off date of the vehicle
RentalDropOffTime	NUMBER	NUMBER(4)	Y	4	NOT NULL	Drop off time of vehicle
RentalPickUpOdometerValue	NUMBER	NUMBER(6,1)	Y	6	NOT NULL	Rental vehicle pick up odometer value
RentalDropOffOdometerValue	NUMBER	NUMBER(6,1)	Y	6	NOT NULL	Rental vehicle drop off odometer value
RentalFuelOptionID (FK)	NUMBER	NUMBER(1)	Y	1	NOT NULL FOREIGN KEY	Foreign Key referencing FuelOption table FuelOptionID
RentalIsuranceOptionID (FK)	NUMBER	NUMBER(1)	Y	1	NOT NULL FOREIGN KEY	Foreign Key referencing RentalInsuranceOption Table RentalInsuranceOptionID
RentalStatusID (FK)	NUMBER	NUMBER(1)	Y	1	NOT NULL FOREIGN KEY	Foreign Key referencing RentalStatus table RentalStatusID

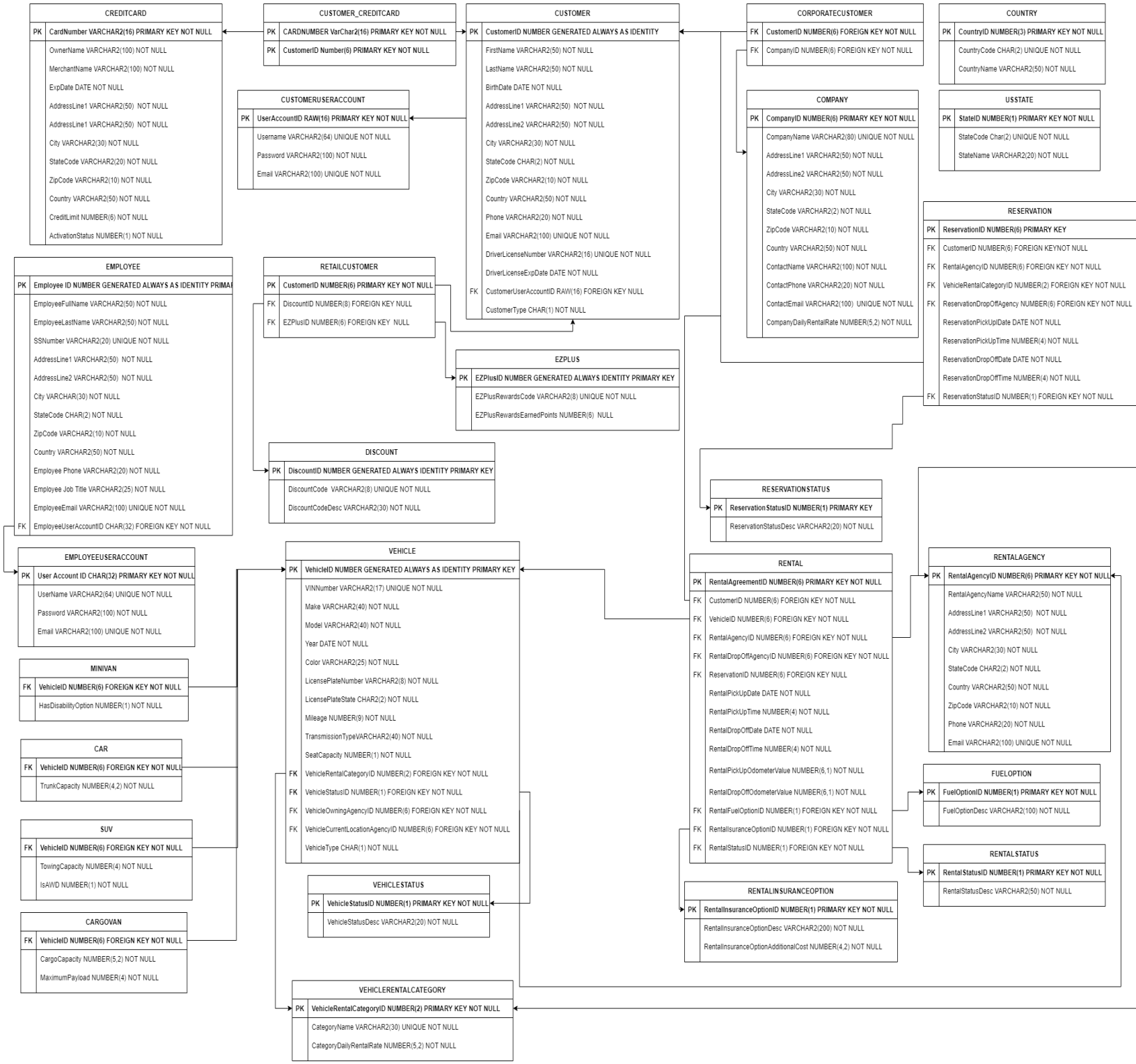
USSTATE

Attribute/Column Name	General Data Type Name	Oracle Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/ purpose
StateID	NUMBER	NUMBER(1)	Y	1	PRIMARY KEY NOT NULL	StateID, Unique identifier for USSTATE. Has Business meaning. Primary Key
StateCode	CHARACTER	Char(2)	Y	2	UNIQUE, NOT NULL	Code of the state
StateName	STRING	VARCHAR2(20)	Y	20	NOT NULL	State's name

COUNTRY						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
CountryID	NUMBER	NUMBER(3)	Y	3	PRIMARY KEY	CountryID, Unique identifier for Country. Has Business meaning. Primary Key
CountryCode	CHARACTER	CHAR(2)	Y	2	UNIQUE NOT NULL	Country's code
CountryName	STRING	VARCHAR2(50)	Y	50	NOT NULL	Country's name

Physical Model Schema Diagram

Below you will find the Physical model schema diagram of the 27 tables.



Development & Implementation

Below you will find the statements that were used to create the 27 tables found in the physical model schema diagram.

```
CREATE TABLE CUSTOMERUSERACCOUNT
(
UserAccountID Raw(16) DEFAULT SYS_GUID() PRIMARY KEY,
UserName VARCHAR(64) UNIQUE NOT NULL,
Password VARCHAR2(100) NOT NULL,
Email VARCHAR2(100) UNIQUE NOT NULL
);

CREATE TABLE CUSTOMER
(
CustomerID NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
FirstName Varchar2(50) NOT NULL,
LastName VARCHAR2(50) NOT NULL,
BirthDate DATE NOT NULL,
AddressLine1 VARCHAR2(50) NOT NULL,
AddressLine2 VARCHAR2(50) NOT NULL,
City VARCHAR2(30) NOT NULL,
StateCode Char(2) NOT NULL,
ZipCode VARCHAR2(10) NOT NULL,
Country VARCHAR2(50) NOT NULL,
Phone VARCHAR2(20) NOT NULL,
Email VARCHAR2(100) UNIQUE NOT NULL,
DriverLicenseNumber VARCHAR2(16) UNIQUE NOT NULL,
DriverLicenseExpDate DATE NOT NULL,
CustomerUserAccountID RAW(16) NULL,
CustomerType Char(1) NOT NULL,
```

```
CONSTRAINT CUSTOMER_CustomerUserAccountID_FK FOREIGN KEY (CustomerUserAccountID)
REFERENCES CUSTOMERUSERACCOUNT(UserAccountID)
);

CREATE TABLE CREDITCARD
(
CardNumber VARCHAR2(16) PRIMARY KEY NOT NULL,
OwnerName VARCHAR2(100) NOT NULL,
MerchantName VARCHAR2(100) NOT NULL,
ExpDate DATE NOT NULL,
AddressLine1 VARCHAR2(50) NOT NULL,
AddressLine2 VARCHAR2(50) NOT NULL,
City VARCHAR2(30) NOT NULL,
StateCode VARCHAR2(20) NOT NULL,
ZipCode VARCHAR2(10) NOT NULL,
Country VARCHAR2(50) NOT NULL,
CreditLimit NUMBER(6) NOT NULL,
ActivationStatus NUMBER(1) NOT NULL
);

CREATE TABLE CUSTOMER_CREDITCARD
(
CardNumber VARCHAR2(16) NOT NULL,
CustomerID NUMBER(6) NOT NULL,

CONSTRAINT Customer_CREDITCARD_PK PRIMARY KEY (CardNumber,CustomerID),

CONSTRAINT CardNumber_FK FOREIGN KEY (CardNumber)
REFERENCES CREDITCARD(CardNumber) ON DELETE CASCADE,

CONSTRAINT CustomerID_FK FOREIGN KEY (CustomerID)
REFERENCES CUSTOMER(CustomerID) ON DELETE CASCADE
```

```
);

CREATE TABLE DISCOUNT
(
DiscountID NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
DiscountCode VARCHAR2(8) UNIQUE NOT NULL,
DiscountDesc VARCHAR2(30) NOT NULL
);

CREATE TABLE EZPLUS
(
EZPlusID NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
EZPlusRewardsCode VARCHAR2(8) UNIQUE NOT NULL,
EZPlusRewardsEarnedPoints NUMBER(6) NULL
);

CREATE TABLE RETAILCUSTOMER
(
CustomerID NUMBER(6) NOT NULL,
DiscountID NUMBER(8) NULL,
EZPlusID NUMBER(6) NULL,

CONSTRAINT RETAILCUSTOMER_CustomerID_FK FOREIGN KEY (CustomerID)
REFERENCES CUSTOMER(CustomerID)ON DELETE CASCADE,

CONSTRAINT RETAILCUSTOMER_DiscountID_FK FOREIGN KEY (DiscountID)
REFERENCES DISCOUNT(DiscountID),

CONSTRAINT RETAILCUSTOMER_EZPlusID_FK FOREIGN KEY (EZPlusID)
REFERENCES EZPLUS(EZPlusID)
);
```

```
CREATE TABLE COMPANY
(
CompanyID NUMBER(6) PRIMARY KEY NOT NULL,
CompanyName VARCHAR2(80) NOT NULL,
AddressLine1 VARCHAR2(50) NOT NULL,
AddressLine2 VARCHAR2(50) NOT NULL,
City VARCHAR2(30) NOT NULL,
StateCode Char(2) NOT NULL,
ZipCode VARCHAR2(10) NOT NULL,
Country VARCHAR2(50) NOT NULL,
ContactName VARCHAR2(100) NOT NULL,
ContactPhone VARCHAR2(20) NOT NULL,
ContactEmail VARCHAR2(100) UNIQUE NOT NULL,
CompanyDailyRentalRate NUMBER(5,2) NOT NULL
);

CREATE TABLE CORPORATECUSTOMER
(
CustomerID NUMBER(6) NOT NULL,
CompanyID NUMBER(6) NOT NULL,

CONSTRAINT CORPORATECUSTOMER_CustomerID_FK FOREIGN KEY (CustomerID)
REFERENCES CUSTOMER(CustomerID) ON DELETE CASCADE,

CONSTRAINT CORPORATECUSTOMER_CompanyID_FK FOREIGN KEY (CompanyID)
REFERENCES COMPANY(CompanyID) ON DELETE CASCADE
);

CREATE TABLE RENTALAGENCY
(
RentalAGencyID NUMBER(6) PRIMARY KEY NOT NULL,
RentalAgencyName VARCHAR2(50) NOT NULL,
```

```
AddressLine1 VARCHAR2(50) NOT NULL,
AddressLine2 VARCHAR2(50) NOT NULL,
City VARCHAR2(30) NOT NULL,
StateCode CHAR(2) NOT NULL,
Country VARCHAR2(50) NOT NULL,
ZipCode VARCHAR2(10) NOT NULL,
Phone VARCHAR2(20) NOT NULL,
Email VARCHAR2(100) UNIQUE NOT NULL
);

CREATE TABLE RESERVATIONSTATUS
(
ReservationStatusID NUMBER(1) PRIMARY KEY,
ReservationStatusDesc VARCHAR2(20) NOT NULL
);

CREATE TABLE VEHICLERENTALCATEGORY
(
VehicleRentalCategoryID NUMBER(2) PRIMARY KEY NOT NULL,
CategoryName VARCHAR2(30) UNIQUE NOT NULL,
CategoryDailyRentalRate NUMBER(5,2) NOT NULL
);

CREATE TABLE RESERVATION
(
ReservationID NUMBER(6) PRIMARY KEY NOT NULL,
CustomerID NUMBER(6) NOT NULL,
RentalAgencyID NUMBER(6) NOT NULL,
VehicleRentalCategoryID NUMBER(2) NOT NULL,
ReservationDropOffAgency NUMBER(6) NOT NULL,
ReservationPickUpDate DATE NOT NULL,
ReservationPickUpTime NUMBER(4)NOT NULL,
```

```
ReservationDropOffDate DATE NOT NULL,
ReservationDropOffTime NUMBER(4)NOT NULL,
ReservationStatusID NUMBER(1) NOT NULL,

CONSTRAINT RESERVATION_CustomerID_FK FOREIGN KEY (CustomerID)
REFERENCES CUSTOMER(CustomerID),

CONSTRAINT RESERVATION_RentalAgencyID_FK FOREIGN KEY (RentalAgencyID)
REFERENCES RENTALAGENCY(RentalAgencyID),

CONSTRAINT RESERVATION_VehicleRentalCategoryID_FK FOREIGN KEY (VehicleRentalCategoryID)
REFERENCES VEHICLERENTALCATEGORY(VehicleRentalCategoryID),

CONSTRAINT RESERVATION_ReservationDropOffAgency_FK FOREIGN KEY (ReservationDropOffAgency)
REFERENCES RENTALAGENCY(RentalAgencyID),

CONSTRAINT RESERVATION_ReservationStatusID_FK FOREIGN KEY (ReservationStatusID)
REFERENCES RESERVATIONSTATUS(ReservationStatusID)
);

CREATE TABLE VEHICLESTATUS
(
VehicleStatusID Number(1) PRIMARY KEY NOT NULL,
HasDisabilityOPtion NUMBER(1) NOT NULL
);

CREATE TABLE VEHICLE
(
VehicleID NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
VINNUMBER VARCHAR2(17) UNIQUE NOT NULL,
Make VARCHAR2(40) NOT NULL,
Model VARCHAR2(40) NOT NULL,
```

```
Year Date NOT NULL,
Color VARCHAR2(25) NOT NULL,
LicensePlateNumber VARCHAR2(8) NOT NULL,
LicensePlateState Char(2) NOT NULL,
Mileage NUMBER(9) NOT NULL,
TransmissionType VARCHAR2(40) NOT NULL,
SeatCapacity NUMBER(1) NOT NULL,
VehicleRentalCategoryID NUMBER(2) NOT NULL,
VehicleStatusID NUMBER(1) NOT NULL,
VehicleOwningAgencyID NUMBER(6) NOT NULL,
VehicleCurrentLocationAgencyID NUMBER(6) NOT NULL,
VehicleType CHAR(1) NOT NULL,

CONSTRAINT VEHICLE_VehicleRentalCategoryID_FK FOREIGN KEY (VehicleRentalCategoryID)
REFERENCES VEHICLERENTALCATEGORY(VehicleRentalCategoryID),

CONSTRAINT VEHICLE_VehicleStatusID_FK FOREIGN KEY (VehicleStatusID)
REFERENCES VEHICLESTATUS(VehicleStatusID),

CONSTRAINT VEHICLE_VehicleOwningAgencyID_FK FOREIGN KEY (VehicleOwningAgencyID)
REFERENCES RENTALAGENCY(RentalAgencyID),

CONSTRAINT VEHICLE_VehicleCurrentLocationAgencyID_FK FOREIGN KEY (VehicleCurrentLocationAgencyID)
REFERENCES RENTALAGENCY(RentalAgencyID)
);

CREATE TABLE MINIVAN
(
VehicleID Number(6) NOT NULL,
HasDisabilityOption NUMBER(1) NOT NULL,

CONSTRAINT MINIVAN_VehicleID_FK FOREIGN KEY (VehicleID)
```

```
REFERENCES VEHICLE(VehicleID)
);

CREATE TABLE CAR
(
VehicleID Number(6) NOT NULL,
TrunkCapacity NUMBER(4,2) NOT NULL,

CONSTRAINT CAR_VehicleID_FK FOREIGN KEY (VehicleID)
REFERENCES VEHICLE(VehicleID)
);

CREATE TABLE SUV
(
VehicleID Number(6) NOT NULL,
TowingCapacity NUMBER(4) NOT NULL,
IsAWD NUMBER(1) NOT NULL,

CONSTRAINT SUV_VehicleID_FK FOREIGN KEY (VehicleID)
REFERENCES VEHICLE(VehicleID)
);

CREATE TABLE CARGOVAN
(
VehicleID Number(6) NOT NULL,
CargoCapacity NUMBER(5,2) NOT NULL,
MaximumPayload NUMBER(4) NOT NULL,

CONSTRAINT CARGOVAN_VehicleID_FK FOREIGN KEY (VehicleID)
REFERENCES VEHICLE(VehicleID)
);
```

```
CREATE TABLE RENTALINSURANCEOPTION
(
RentalInsuranceOptionID NUMBER(1) PRIMARY KEY NOT NULL,
RentalInsuranceOptionDesc VARCHAR2(200) NOT NULL,
RentalInsuranceOptionAdditionalCost NUMBER(4,2) NOT NULL
);

CREATE TABLE FUELOPTION
(
FuelOptionID NUMBER(1) PRIMARY KEY NOT NULL,
RentalStatusDesc VARCHAR2(50) NOT NULL
);

CREATE TABLE RENTALSTATUS
(
RentalStatusID NUMBER(1) PRIMARY KEY NOT NULL,
RentalStatusDesc VARCHAR2(50) NOT NULL
);

CREATE TABLE RENTAL
(
RentalAgreementID NUMBER(6) PRIMARY KEY NOT NULL,
CustomerID NUMBER(6) NOT NULL,
VehicleID NUMBER(6) NOT NULL,
RentalAgencyID NUMBER(6) NOT NULL,
RentalDropOffAgencyID NUMBER(6) NOT NULL,
ReservationID NUMBER(6) NULL,
RentalPickupDate DATE NOT NULL,
RentalPickUpTime NUMBER(4) NOT NULL,
RentalDropOffDate DATE NOT NULL,
RentalDropOffTime NUMBER(4) NOT NULL,
RentalPickUpOdometerValue NUMBER(6,1) NOT NULL,
```

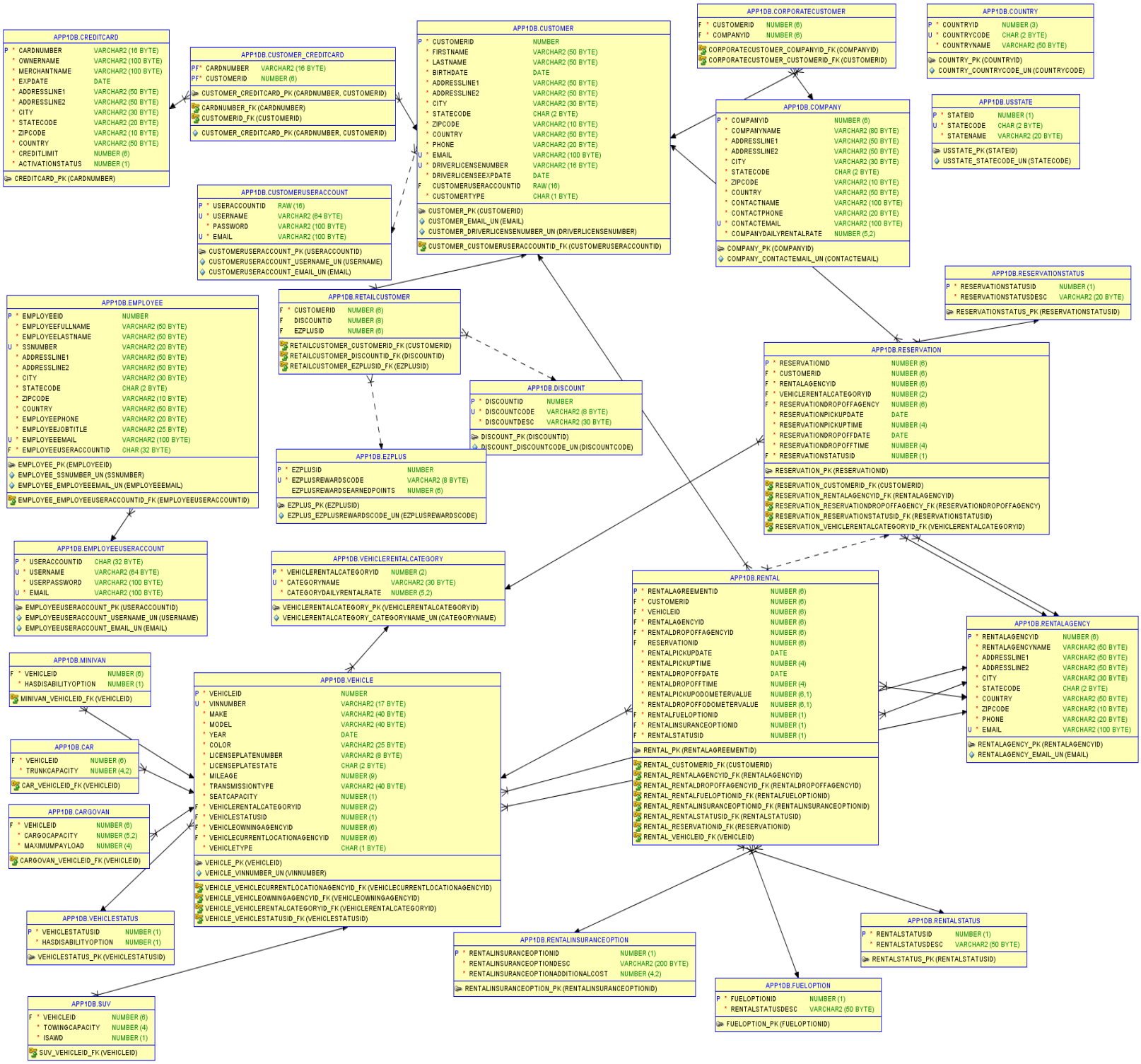
```
RentalDropOffOdometerValue NUMBER(6,1) NOT NULL,
RentalFuelOptionID NUMBER(1) NOT NULL,
RentalInsuranceOptionID NUMBER(1) NOT NULL,
RentalStatusID NUMBER(1) NOT NULL,

CONSTRAINT RENTAL_CustomerID_FK FOREIGN KEY (CustomerID)
REFERENCES CUSTOMER(CustomerID),

CONSTRAINT RENTAL_VehicleID_FK FOREIGN KEY (VehicleID)
REFERENCES VEHICLE(VehicleID),

CONSTRAINT RENTAL_RentalAgencyID_FK FOREIGN KEY (RentalAgencyID)
REFERENCES RENTALAGENCY(RentalAgencyID),

CONSTRAINT RENTAL_RentalDropOffAgencyID_FK FOREIGN KEY (RentalDropOffAgencyID)
REFERENCES RENTALAGENCY(RentalAgencyID),

CONSTRAINT RENTAL_ReservationID_FK FOREIGN KEY (ReservationID)
REFERENCES RESERVATION(ReservationID),

CONSTRAINT RENTAL_RentalFuelOptionID_FK FOREIGN KEY (RentalFuelOptionID)
REFERENCES FUELOPTION(FuelOptionID),

CONSTRAINT RENTAL_RentalInsuranceOptionID_FK FOREIGN KEY (RentalInsuranceOptionID)
REFERENCES RENTALINSURANCEOPTION(RentalInsuranceOptionID),

CONSTRAINT RENTAL_RentalStatusID_FK FOREIGN KEY (RentalStatusID)
REFERENCES RENTALSTATUS(RentalStatusID)
);

CREATE TABLE EMPLOYEEUSERACCOUNT
(
```

```
UserAccountID Char(32) PRIMARY KEY NOT NULL,
UserName VARCHAR2(64) UNIQUE NOT NULL,
Password VARCHAR2(100) NOT NULL,
Email VARCHAR2(100) UNIQUE NOT NULL
);

CREATE TABLE EMPLOYEE
(
EmployeeID NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
EmployeeFullName VARCHAR2(50) NOT NULL,
EmployeeLastName VARCHAR2(50) NOT NULL,
SSNumber VARCHAR(20)UNIQUE NOT NULL,
AddressLine1 VARCHAR2(50) NOT NULL,
AddressLine2 VARCHAR2(50) NOT NULL,
City VARCHAR(30) NOT NULL,
StateCode CHAR(2) NOT NULL,
ZipCode VARCHAR2(10) NOT NULL,
Country VARCHAR2(50) NOT NULL,
EmployeePhone VARCHAR2(20) NOT NULL,
EmployeeJobTitle VARCHAR2(25) NOT NULL,
EmployeeEmail VARCHAR2(100) UNIQUE NOT NULL,
EmployeeUserAccountID CHAR(32) NOT NULL,

CONSTRAINT EMPLOYEE_EmployeeUserAccountID_FK FOREIGN KEY (EmployeeUserAccountID)
REFERENCES EMPLOYEEUSERACCOUNT(UserAccountID)
);

CREATE TABLE COUNTRY
(
CountryID NUMBER(3) PRIMARY KEY NOT NULL,
CountryCode CHAR(2) UNIQUE NOT NULL,
CountryName VARCHAR2(50) NOT NULL
```

```
);

CREATE TABLE USSTATE
(
StateID NUMBER(1) PRIMARY KEY NOT NULL,
StateCode CHAR(2) UNIQUE NOT NULL,
StateName VARCHAR2(20) NOT NULL
);
```

Development & Implementation Physical Schema Diagram

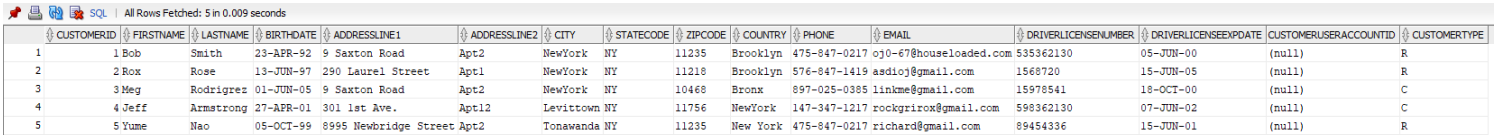
Below you see the Database diagram that was created which matches the physical model diagram.



The following are insert statements for CUSTOMER table:

```
Insert INTO CUSTOMER
VALUES(DEFAULT,'Bob','Smith','23-APR-1992','9 Saxton Road',
'Apt2','NewYork','NY','11235','Brooklyn','475-847-0217',
'oj0-67@houseloaded.com','535362130','05-JUNE-2000',NULL,'R');

Insert INTO CUSTOMER
VALUES(DEFAULT,'Rox','Rose','13-Jun-1997','290 Laurel Street',
'Apt1','NewYork','NY','11218','Brooklyn','576-847-1419',
'asdioj@gmail.com','1568720','15-JUNE-2005',NULL,'R');

Insert INTO CUSTOMER
VALUES(DEFAULT,'Meg','Rodrigrez','1-Jun-2005','9 Saxton Road','Apt2',
'NewYork','NY','10468','Bronx','897-025-0385',
'linkme@gmail.com','15978541','18-OCT-2000',NULL,'C');

Insert INTO CUSTOMER
VALUES(DEFAULT,'Jeff','Armstrong','27-APR-2001','301 1st Ave.',
'Apt12','Levittown','NY','11756','NewYork','147-347-1217',
'rockgrirox@gmail.com','598362130','07-JUNE-2002',NULL,'C');

Insert INTO CUSTOMER
VALUES(DEFAULT,'Yume','Nao','05-OCT-1999','8995 Newbridge Street',
'Apt2','Tonawanda','NY','11235','New York','475-847-0217',
'richard@gmail.com','89454336','15-JUNE-2001',NULL,'R');
```

The results of these following statements were that the rows were inserted successfully into the table. Below you will see the results of all the insertions made to the table.



SQL | All Rows Fetched: 5 in 0.009 seconds

	CUSTOMERID	FIRSTNAME	LASTNAME	BIRTHDATE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY	PHONE	EMAIL	DRIVERLICENSENUMBER	DRIVERLICENSEEXPDATE	CUSTOMERUSERACCOUNTID	CUSTOMERTYPE
1	1	Bob	Smith	23-APR-92	9 Saxton Road	Apt2	NewYork	NY	11235	Brooklyn	475-847-0217	oj0-67@houseloaded.com	535362130	05-JUN-00	(null)	R
2	2	Rox	Rose	13-JUN-97	290 Laurel Street	Apt1	NewYork	NY	11218	Brooklyn	576-847-1419	asdioj@gmail.com	1568720	15-JUN-05	(null)	R
3	3	Meg	Rodrigrez	01-JUN-05	9 Saxton Road	Apt2	NewYork	NY	10468	Bronx	897-025-0385	linkme@gmail.com	15978541	18-OCT-00	(null)	C
4	4	Jeff	Armstrong	27-APR-01	301 1st Ave.	Apt12	Levittown	NY	11756	NewYork	147-347-1217	rockgrirox@gmail.com	598362130	07-JUN-02	(null)	C
5	5	Yume	Nao	05-OCT-99	8995 Newbridge Street	Apt2	Tonawanda	NY	11235	New York	475-847-0217	richard@gmail.com	89454336	15-JUN-01	(null)	R

The following are insert statements for COMPANY table:

Insert INTO COMPANY

VALUES(1,'RentalBuy','931 Pearl Lane

','931 Pearl Lane','East Elmhurst','NY','11370','NewYork','Richard Richsinson',

'456-741-1546','RentalBuyinc@gmail.com',123.53);

Insert INTO COMPANY

VALUES(2,'Staple','65 Tanglewood Street

','65 Tanglewood Street','Brooklyn','NY','11212','NewYork','Copper Richsinson',

'369-741-1546','Staple@gmail.com',80.53);

Insert INTO COMPANY

VALUES(3,'EZTools','9538 Glenlake St.

','9538 Glenlake St.','New York','NY','10025','NewYork','Lenny Hawk',

'456-741-1546','EZTools@gmail.com',50.53);

Insert INTO COMPANY

VALUES(4,'BoxnBox','50 Military Drive

','50 Military Drive','Buffalo','NY','14224','NewYork','Randy Wrangler',

'369-258-1546','BoxnBox@gmail.com',23.53);

Insert INTO COMPANY

VALUES(5,'Woopers','129 Leeton Ridge St.

','129 Leeton Ridge St.','Jamaica','NY','11435','NewYork','Bob Smith',

'456-741-1546','Woopers@gmail.com',42.23);

The results of these following statements were that the rows were inserted successfully into the table. Below you will see the results of all the insertions made to the table.

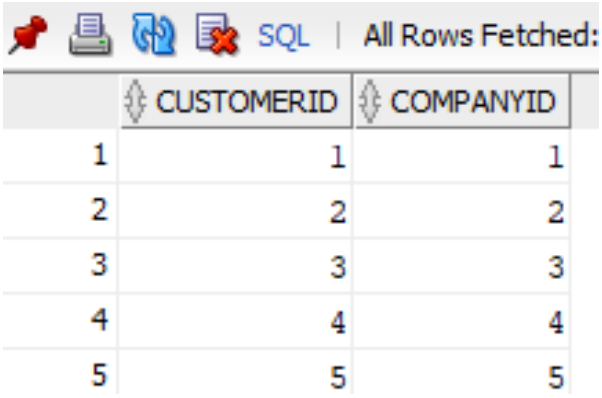
SQL | All Rows Fetched: 5 in 0.003 seconds

	COMPANYID	COMPANYNAME	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY	CONTACTNAME	CONTACTPHONE	CONTACTEMAIL	COMPANYDAILYRENTALRATE
1	1	RentalBuy	931 Pearl Lane	931 Pearl Lane	East Elmhurst	NY	11370	NewYork	Richard Richsinson	456-741-1546	RentalBuyinc@gmail.com	123.53
2	2	Staple	65 Tanglewood Street	65 Tanglewood Street	Brooklyn	NY	11212	NewYork	Richard Richsinson	369-741-1546	Staple@gmail.com	80.53
3	3	EZTools	9538 Glenlake St.	9538 Glenlake St.	New York	NY	10025	NewYork	Lenny Hawk	456-741-1546	EZTools@gmail.com	50.53
4	4	BoxnBox	50 Military Drive	50 Military Drive	Buffalo	NY	14224	NewYork	Randy Wrangler	369-258-1546	BoxnBox@gmail.com	23.53
5	5	Woopers	129 Leeton Ridge St.	129 Leeton Ridge St.	Jamaica	NY	11435	NewYork	Bob Smith	456-741-1546	Woopers@gmail.com	42.23

The following are insert statements for CORPORATECUSTOMER table:

```
INSERT INTO CORPORATECUSTOMER
VALUES(1,1);

INSERT INTO CORPORATECUSTOMER
VALUES(2,2);

INSERT INTO CORPORATECUSTOMER
VALUES(3,3);

INSERT INTO CORPORATECUSTOMER
VALUES(4,4);

INSERT INTO CORPORATECUSTOMER
VALUES(5,5);
```

The results of these following statements were that the rows were inserted successfully into the table. Below you will see the results of all the insertions made to the table.


SQL | All Rows Fetched:

	CUSTOMERID	COMPANYID
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5

The following are insert statements for CREDITCARD table:

Insert INTO CREDITCARD
VALUES (5411406697268879,'Box Smith','Amazon','03-APR-2030','9915 Monroe St.',
'Apt#5','Newburgh','NY','12550','New York',1000,1);

Insert INTO CREDITCARD
VALUES (5105903518925977,'Roxie Lex','Amazon','14-JUN-2025','41 Pin Oak St.',
'Apt#3','Yonkers','NY','10701','New York',500,1);

Insert INTO CREDITCARD
VALUES (5194176023813994,'Lenny Hawk','Wallmart','03-APR-2030','13 W. Surrey St.',
'Apt#6','Spring Valley','NY','10977','New York',250,1);

Insert INTO CREDITCARD
VALUES (5176620860731399,'Box Smith','Twitch','25-SEP-2026','46 Sunbeam St.',
'Apt#1','Astoria','NY','11105','New York',347,0);

Insert INTO CREDITCARD

VALUES (5447022440523615,'Box Smith','Macys','07-DEC-2027','17 2nd Street',

'Apt#23','New York','NY','10016','New York',574,1);

The results of these following statements were that the rows were inserted successfully into the table. Below you will see the results of all the insertions made to the table.

	CARDNUMBER	OWNERNAME	MERCHANTNAME	EXPDATE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY	CREDITLIMIT	ACTIVATIONSTATUS
1	5411406697268879	Box Smith	Amazon	03-APR-30	9915 Monroe St.	Apt#5	Newburgh	NY	12550	New York	1000	1
2	5105903518925977	Roxie Lex	Amazon	14-JUN-25	41 Pin Oak St.	Apt#3	Yonkers	NY	10701	New York	500	1
3	5194176023813994	Lenny Hawk	Wallmart	03-APR-30	13 W. Surrey St.	Apt#6	Spring Valley	NY	10977	New York	250	1
4	5176620860731399	Box Smith	Twitch	25-SEP-26	46 Sunbeam St.	Apt#1	Astoria	NY	11105	New York	347	0
5	5447022440523615	Box Smith	Macys	07-DEC-27	17 2nd Street	Apt#23	New York	NY	10016	New York	574	1

The following are insert statements for CUSTOMER_CREDITCARD table:

Insert INTO CUSTOMER_CREDITCARD

VALUES(5411406697268879,1);

Insert INTO CUSTOMER_CREDITCARD

VALUES(5105903518925977,2);

Insert INTO CUSTOMER_CREDITCARD

VALUES(5194176023813994,3);

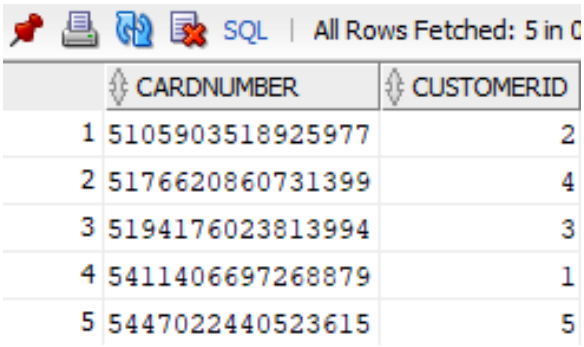
Insert INTO CUSTOMER_CREDITCARD

VALUES(5176620860731399,4);

Insert INTO CUSTOMER_CREDITCARD

VALUES(5447022440523615,5);

The results of these following statements were that the rows were inserted successfully into the table. Below you will see the results of all the insertions made to the table.


SQL | All Rows Fetched: 5 in 0

	CARDNUMBER	CUSTOMERID
1	5105903518925977	2
2	5176620860731399	4
3	5194176023813994	3
4	5411406697268879	1
5	5447022440523615	5

The following are insert statements for CUSTOMERUSERACCOUNT table:

Insert INTO CUSTOMERUSERACCOUNT

VALUES (DEFAULT,'Jeff','defi542@','Jeff@gmail.com');

Insert INTO CUSTOMERUSERACCOUNT

VALUES (DEFAULT,'Richard','h54dfs@','Richard@gmail.com');

Insert INTO CUSTOMERUSERACCOUNT

VALUES (DEFAULT,'Bob','5617@89@','bobys@gmail.com');

Insert INTO CUSTOMERUSERACCOUNT

VALUES (DEFAULT,'Link','cupcakes6','linkdabest@gmail.com');

Insert INTO CUSTOMERUSERACCOUNT

VALUES (DEFAULT,'Sora','pancakes12','Soracoralora@gmail.com');

The results of these following statements were that the rows were inserted successfully into the table. Below you will see the results of all the insertions made to the table.

SQL | All Rows Fetched: 5 in 0.003 seconds

	USERACCOUNTID	USERNAME	PASSWORD	EMAIL
1	71F1686BB7164F50AC58A36E60564012	Jeff	defi542@	Jeff@gmail.com
2	7D7BCFA9769242BCB904A91299EF6583	Richard	h54dfs@	Richard@gmail.com
3	A402B61AD2144386A25BC9A739C3B0EC	Bob	5617@89@	bobys@gmail.com
4	15E9374FC197472786E84F8EF488BD50	Link	cupcakes6	linkdabest@gmail.com
5	73C097AEDC6640709F368C2869897419	Sora	pancakes12	Soracoralora@gmail.com

The following is the select statement for CUSTOMER that will return one record that includes all columns based on the primary key:

Select * from CUSTOMER where CUSTOMERID = 1;

Below is the result of the previous select statement that is selecting all columns from CUSTOMER Table where the CUSTOMERID is equal to one.

SQL | All Rows Fetched: 1 in 0.004 seconds

	CUSTOMERID	FIRSTNAME	LASTNAME	BIRTHDATE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY	PHONE	EMAIL	DRIVERLICENSENUMBER	DRIVERLICENSEEXPDATE	CUSTOMERUSERACCOUNTID	CUSTOMERTYPE
1	1	Bob	Smith	23-APR-92	9 Saxton Road	Apt2	NewYork	NY	11235	Brooklyn	475-847-0217	oj0-67@houseloaded.com	535362130	05-JUN-00	(null)	R

The following is the select statement for CUSTOMER that will return multiple records that includes all columns based on the CustomerType:

Select * from CUSTOMER where CustomerType = ‘R’;

Below is the result of the previous select statement that is selecting all columns from CUSTOMER Table where the CustomerType is equal to ‘R’.

SQL | All Rows Fetched: 3 in 0.007 seconds

	CUSTOMERID	FIRSTNAME	LASTNAME	BIRTHDATE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY	PHONE	EMAIL	DRIVERLICENSENUMBER	DRIVERLICENSEEXPDATE	CUSTOMERUSERACCOUNTID	CUSTOMERTYPE
1	1	Bob	Smith	23-APR-92	9 Saxton Road	Apt2	NewYork	NY	11235	Brooklyn	475-847-0217	oj0-67@houseloaded.com	535362130	05-JUN-00	(null)	R
2	2	Rox	Rose	13-JUN-97	290 Laurel Street	Apt1	NewYork	NY	11218	Brooklyn	576-847-1419	asdioj@gmail.com	1568720	15-JUN-05	(null)	R
3	5	Yume	Nao	05-OCT-99	8995 Newbridge Street	Apt2	Tonawanda	NY	11235	New York	475-847-0217	richard@gmail.com	89454336	15-JUN-01	(null)	R

The following is the select statement for CORPORATECUSTOMER,CUSTOMER,COMPANY that will return multiple records that includes multiple columns based on the CompanyID and CustomerType:

Select cor.CompanyID,co.CompanyName,cu.FirstName,cu.LastName,cu.CustomerType

From CORPORATECUSTOMER cor,COMPANY co,CUSTOMER cu WHERE co.CompanyID = 1 AND cu.CustomerType = 'R';

Below is the result of the previous select statement that is selecting multiple columns from CORPORATECUSTOMER, CUSTOMER, AND COMPANY Table where the co.CompanyID is equal to '1' AND cu.CustomerType is equal to 'R'.

SQL | All Rows Fetched: 15 in 0.008 seconds

	COMPANYID	COMPANYNAME	FIRSTNAME	LASTNAME	CUSTOMERTYPE
1	1	BiliBili	Bob	Smith	R
2	1	BiliBili	Rox	Rose	R
3	1	BiliBili	Yume	Nao	R
4	2	BiliBili	Bob	Smith	R
5	2	BiliBili	Rox	Rose	R
6	2	BiliBili	Yume	Nao	R
7	3	BiliBili	Bob	Smith	R
8	3	BiliBili	Rox	Rose	R
9	3	BiliBili	Yume	Nao	R
10	4	BiliBili	Bob	Smith	R
11	4	BiliBili	Rox	Rose	R
12	4	BiliBili	Yume	Nao	R
13	5	BiliBili	Bob	Smith	R
14	5	BiliBili	Rox	Rose	R
15	5	BiliBili	Yume	Nao	R

The following is the UPDATE statement for COMPANY table that will update all columns except the primary key:

UPDATE COMPANY SET CompanyName = 'BiliBili', AddressLine1 = '2277 Oak Street Apt#2',

AddressLine2 = '2277 Oak Street Apt#2', City = 'Old Forge', StateCode = 'NY',ZipCode = '13420',

Country = 'Old Forge', ContactName = 'Annika M Ramos' ,ContactPhone = '315-369-5311' ,

ContactEmail = 'sjror3tvbg@temporary-mail.net',CompanyDailyRentalRate = 25.65

WHERE CompanyID = 1;

Below is the result of the previous update statement that updated all columns from Company Table where the CompanyID is equal to '1'.

SQL | All Rows Fetched: 5 in 0.002 seconds

	COMPANYID	COMPANYNAME	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY	CONTACTNAME	CONTACTPHONE	CONTACTEMAIL	COMPANYDAILYRENTALRATE
1	1	BiliBili	2277 Oak Street Apt#2	2277 Oak Street Apt#2	Old Forge	NY	13420	Old Forge	Annika M Ramos	315-369-5311	sjror3tvbg@temporary-mail.net	25.65
2	2	Staple	65 Tanglewood Street	65 Tanglewood Street	Brooklyn	NY	11212	NewYork	Richard Richsinson	369-741-1546	Staple@gmail.com	80.53
3	3	EZTools	9538 Glenlake St.	9538 Glenlake St.	New York	NY	10025	NewYork	Lenny Hawk	456-741-1546	EZTools@gmail.com	50.53
4	4	BoxnBox	50 Military Drive	50 Military Drive	Buffalo	NY	14224	NewYork	Randy Wrangler	369-258-1546	BoxnBox@gmail.com	23.53
5	5	Woopers	129 Leeton Ridge St.	129 Leeton Ridge St.	Jamaica	NY	11435	NewYork	Bob Smith	456-741-1546	Woopers@gmail.com	42.23

The following is the UPDATE statement for CUSTOMER_CREDITCARD table that will one column of an associate table:

UPDATE CUSTOMER_CREDITCARD SET CustomerID = 1 WHERE CustomerID = 2;

Below is the result of the previous update statement that updated the column from CUSTOMER_CREDITCARD Table where the CustomerID is equal to ‘2’.

SQL | All Rows Fetched: 5 in 0

	CARDNUMBER	CUSTOMERID
1	5105903518925977	1
2	5176620860731399	4
3	5194176023813994	3
4	5411406697268879	1
5	5447022440523615	5

The following is the DELETE statement for COMPANY table that will delete a column based on the primary key:

Delete FROM COMPANY where CompanyID = 1;

Below is the result of the previous delete statement that delete the column from COMPANY Table where the CompanyID is equal to ‘1’.

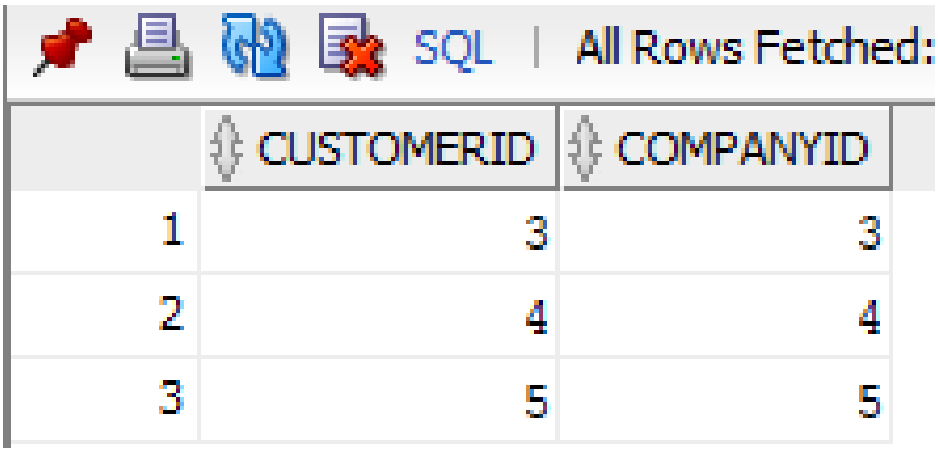
SQL | All Rows Fetched: 4 in 0.003 seconds

	COMPANYID	COMPANYNAME	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY	CONTACTNAME	CONTACTPHONE	CONTACTEMAIL	COMPANYDAILYRENTALRATE
1	2	Staple	65 Tanglewood Street	65 Tanglewood Street	Brooklyn	NY	11212	NewYork	Copper Richsinson	369-741-1546	Staple@gmail.com	80.53
2	3	EZTools	9538 Glenlake St.	9538 Glenlake St.	New York	NY	10025	NewYork	Lenny Hawk	456-741-1546	EZTools@gmail.com	50.53
3	4	BoxnBox	50 Military Drive	50 Military Drive	Buffalo	NY	14224	NewYork	Randy Wrangler	369-258-1546	BoxnBox@gmail.com	23.53
4	5	Woopers	129 Leeton Ridge St.	129 Leeton Ridge St.	Jamaica	NY	11435	NewYork	Bob Smith	456-741-1546	Woopers@gmail.com	42.23

The following is the DELETE statement for CORPORATECUSTOMER table that will delete the record from an associative entity table:

Delete FROM CORPORATECUSTOMER WHERE CompanyID = 2;

Below is the result of the previous delete statement that delete the record from CORPORATECUSTOMER Table where the CompanyID is equal to '2'.



	CUSTOMERID	COMPANYID
1	3	3
2	4	4
3	5	5

Conclusion

In conclusion the EZRentalPOS project was successful both in design and implementation with all of the requirements successfully.