

EZRental Inc.

Abel Angel Rodriguez
(Program Manager & Project Manager)

Hasnain Zafar
(Database Developer & Database Administrator)



PROJECT 1 – EZRental

POS Project Objectives,

Definition & Database

Design and

Implementation

Executive Summary

The EZRental database is created on Oracle 18c XE edition. Entities of the database are created well suitable for the EZRental business. In this document all aspects of the EZRental are documented, with mentioning different roles of people in creating the database. This document defines each columns of the database with its data type and why is it being used here. Also, by the end of the document it contains the creating and testing part of the database which clearly shows how is it working by capturing the result of the queries.

Problem Statement & Objectives

EZRental Inc., has hired our team, to design & implement a Client/Server Application Auto Rental Point-of-Sales Management System named EZRental POS, which includes an e-commerce site name EZRental.com.

Basic objectives and architecture that are being targeted are following:

- The EZRental POS System is designed to allow customers, both retail and corporate, to reserve vehicles for renting similar to existing in-person or online car rental systems such as Avis, Hertz, Budget, etc.
- The application is to be designed to support dozens of major cities around the world. In addition, provide 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

Following is the information about Project Manager, Consultant and the roles they are handling. Namely Hasnain Zafar is the Database Developer and Database Administrator of the project which is managed by Abel Angel Rodriguez. Below is the information about manager and consultants in more depth.

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

		<ul style="list-style-type: none"> • Roles include but not limited to: <ol style="list-style-type: none"> 1. Engage in discovery activities & Interview the stakeholders at EZRental Inc. 2. From the interview and discovery 1) Design/Architect the ObjectOriented-Programming Class/Object Model. 3. From the interview and discovery 1) Design a high-level Graphical User-Interface (GUID) wireframe, & 2) front-end features & functionality.
Consultant #5	Full Stack Application Developer	<ul style="list-style-type: none"> • Object-Oriented-Programming developer to implement the Windows Client application using JAVA technologies & 2) on the database side, implement stored procedures and support the databased team as needed. • Roles include but not limited to: <ol style="list-style-type: none"> 1. As OOP developer, 1) Object-Oriented-Programming of Class/Object Model designed by Consultant #4 of the Windows Client/Server Application Client using JAVA. 2. In addition, 2) Database Stored Procedures and other development requirements in the Back-end DBMS.
Consultant #6	Full Stack Web Developer	<ul style="list-style-type: none"> • Object-Oriented-Programming developer & Web Developer to implement the Webbased application using JAVA and other technology to be determine in future & 2) on the database side, implement stored procedures and support the databased team as needed. • Roles include but not limited to: <ol style="list-style-type: none"> 1. Web Developer, 1) Object-Oriented-Programming of Class/Object Model designed by Consultant #4 of the Web Client/Server Application Client using JAVA & Other Web Technology to be determine. 2. In addition, 2) Database Stored Procedures and other development requirements in the Back-end DBMS.

Application Business Requirements

A **Business Analyst** is hired by **Abel Angel Rodriguez** (Project Manager) to interview EZRental Inc. project stakeholders and compile the list of the business data requirements for the **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:

<i>Discount ID</i>	<i>Discount Code</i>	<i>Discount Code Description</i>
1234..	AAA99700	AAA Membership Discount - 25% off base Rewards Code
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
	Etc..	Etc..
1213..	VET20551	Veteran Discount 35% off base rate Plus 10% donation to veteran's family fund.

<i>EZPlus ID</i>	<i>EZPlus</i>	<i>EZPlus Rewards</i>
1234..	EZP90098	10000
5678..	EZP10001	500
9101..	EZP64932	159000
1213..	EZP20051	23000

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-automatic & dual-

- clutchAll vehicles also have a special identifier we use to track the vehicle status named **statusID**) and **seat capacity** . **Reserved.Rented.**

vehicle status ID. This is a unique number that identifies the status of a vehicle, which works in conjunction with **3**

vehicle status descriptionAvailable. which describes the status, such as reserved, rented, available, maintenance, not available, transferred, etc. **4** Below Is the list of vehicle status IDs we are currently using and their descriptions: **Not available**

5

Maintenance

6



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

<i>Reservation Status ID</i>	<i>Reservation Status Description</i>
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:
 - A vehicle can be picked up at the **SAME** rental agency as indicated by the reservation and dropped off at the **SAME** rental agency.
 - Or a vehicle can be picked up at the **SAME** rental agency as indicated by the reservation and dropped off at **ANOTHER** rental agency.
 - Or a vehicle can be picked up at **ANOTHER** rental agency other than what was indicated by the reservation and dropped off at **SAME** rental agency of the reservation.
 - Finally, a vehicle can be picked up at **ANOTHER** rental agency other than what was indicated by the reservation and dropped off at **ANOTHER** rental agency of the reservation.
- Note that for scenarios 3 & 4, we cannot guarantee that the vehicle rental category of the reservation will be available at the agency other than what was agreed in the reservation. We will do our best to accommodate the change during these scenarios or find another vehicle that will be closed to the original reservation.
- 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 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	Liability Extended Protection – Agency provides renter with third party liability protection up to \$1 Million per accident for bodily injury or death or property damage to others.	\$89.99
5	Roadside Assistance Plus – 24/7 roadside assistance, replacement for lost keys, flat tire service, fuel delivery, etc.	\$15.99

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

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

Business Requirements (Cont.)

The Rental Process (Cont.):

- 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-Of-Sales & Back-End Management systems. In addition to our **EZRental.com** portal by our customers. Therefore, due to security and regulatory compliance purpose, we want to separate the employee access data from the customer access data by using two separate user accounts:

- Employee user accounts
- Customer user accounts

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

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

An employee user account should store the user *employee user account ID* a unique identifier alpha-numeric string that identifies the employee user account, *employee username* another unique alpha-numeric that identifies each individual user, 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 identity 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 evolves, 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

An **Application Analyst/Architect** is hired by Mr. Rodriguez to interview **EZRental Inc.**, project Business Decision Makers (BDMs), stakeholders and Information System Technical Decision Makers (TDMs) to compile the list of the application **Application Development & Technical Requirements** in order to develop the design & identify the technical requirements for the application.

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

Application Development & Technical Requirements

Introduction & Current Challenges

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

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

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

Rental Agencies Application & Technical Requirements:

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

Rental Agency Application Architecture Requirements:

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

1. Client application processing, transaction and response must be fast to minimize service time for a customer.
2. All transaction processing should be done in the user's computer or desktop for fast processing and response.
3. Application Architecture must be reusable and scalable to support future updates and new feature enhancements, without a long development lifecycle.
4. Depending on the architecture NYC-Tech Solutions Inc., decides for the application in the rental agencies (Desktop client or Web client). the primary Application Development Platform we use is **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-of-Sales with appropriate conditional access.

Application Development & Technical Requirements (Cont.)

Rental Agency Application Features and Functionalities Requirements:

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

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

Rental Agency Application Graphical User Interface Requirements:

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

Application Development & Technical Requirements

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

- **EZPlus Rewards Codes:**

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

- **Company Name:**

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

- **Vehicle Status:**

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

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

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

- **Rental Agency:**

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

Application Development & Technical Requirements

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

- **Rental Status:**

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

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

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

- **Rental Fuel Option:**

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

<i>Rental Fuel Option ID</i>	<i>Rental Fuel Option Description</i>	<i>Rental Fuel Option Additional Cost</i>
1	Return with a full tank or on return, pay for gas that is missing.	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:

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

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

Application Development & Technical Requirements (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, operate and update the application once NYCTech consultants complete the project and development of this system.
4. Web Portal Security Authentication System – Proper security and authentication must be implemented to make sure only the customer can access the **EZRental.com** website for his or her profile home page.

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

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

Web Portal Application Web Pages User Interface Requirements:

The web pages graphical UI requirements are listed below:

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

Application Development & Technical Requirements

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

- **EZPlus Rewards Codes:**

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

- **Rental Agency:**

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

- **Vehicle Rental Category:**

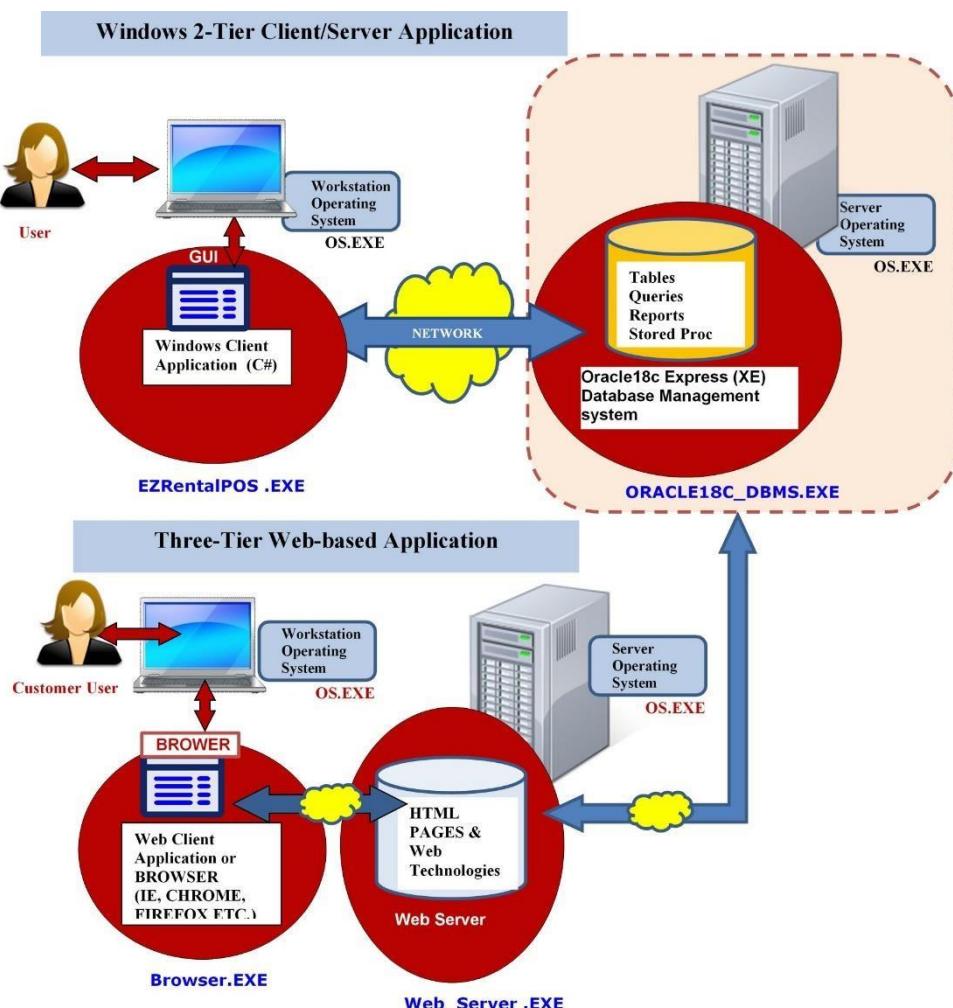
- Web pages that require the use of the VEHICLE RENTAL CATEGORY fields, must be prepopulated with the list of vehicle rental categories. Based on the business requirements, the current list of vehicle rental categories is as follows:

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

Application Physical & Technical Architecture

Following is the explanation of **2-Tier Client/Server Application**:

- User in the business office uses a WINDOWS Client Application in their workstation or laptop.
- The Client Application connects via the company's network to a DATABASE SERVER Application which is the Second Component of the Application.
- In the company's DATACENTER, The Database Management System or Database Server application manages the Client Application's Data on behalf of the Client Application.
- The Database Server enables the WINDOWS Client Application to retrieve, update, insert & delete its Data from the Database Server.



Following is the explanation of **Three-Tier Web-based Application**:

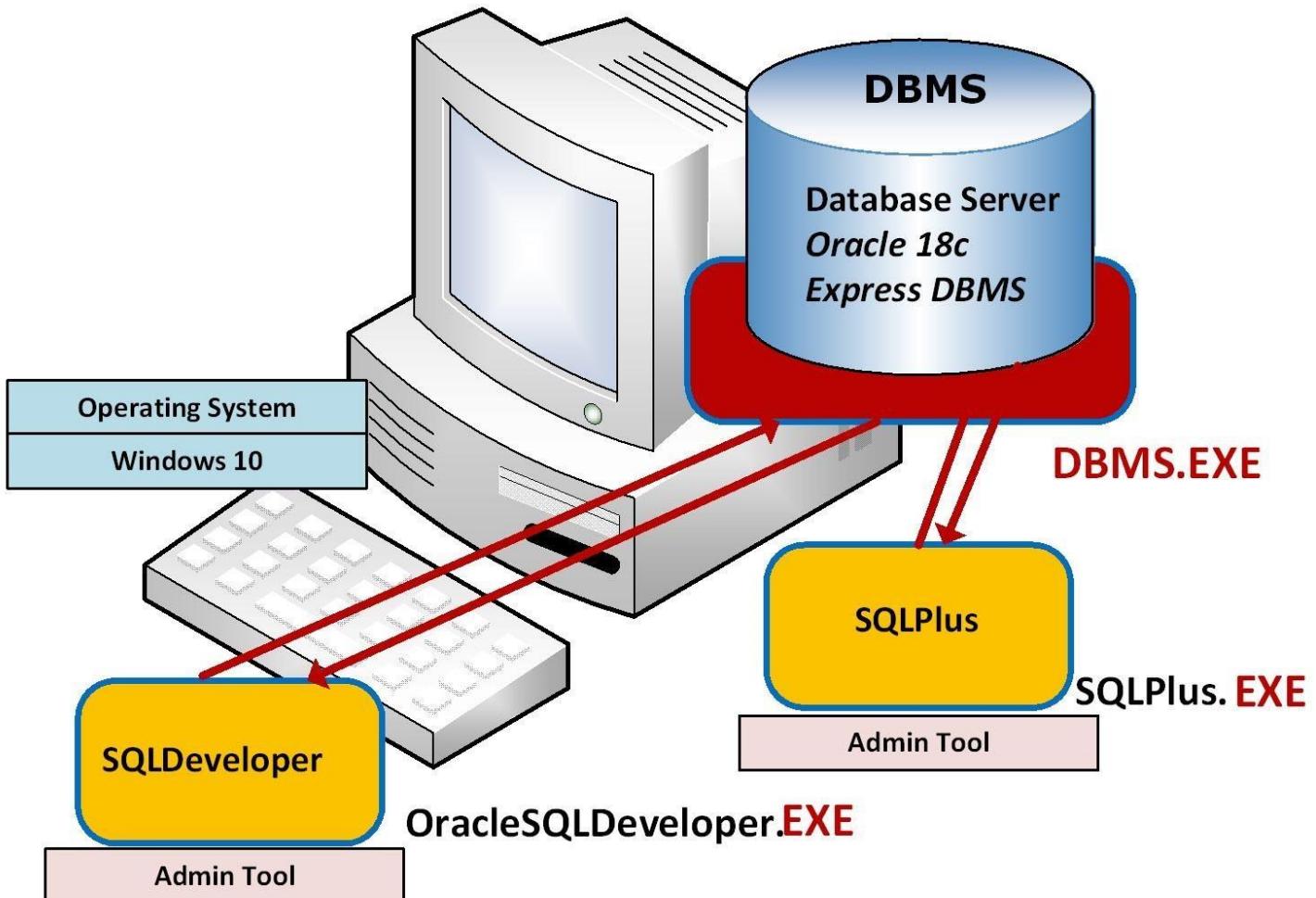
- User in the business office uses a BROWSER Client Application in their workstation or laptop.
- The BROWSER Client Application connects via the company's network to a WEB SERVER Application which is the Second Component of the Application.
- In the company's DATACENTER, the WEB SERVER Application connects via the company's network to a DATABASE SERVER Application which is the Third Component of the Application.

- In the company's DATACENTER, The Database Management System or Database Server application manages the Client Application's Data on behalf of the BROWSER Client Application.
- The Database Server enables the WEB SERVER Application to retrieve, update, insert & delete its Data from the Database Server on behalf of the BROWSER Client Application.

Database Management Physical Architecture

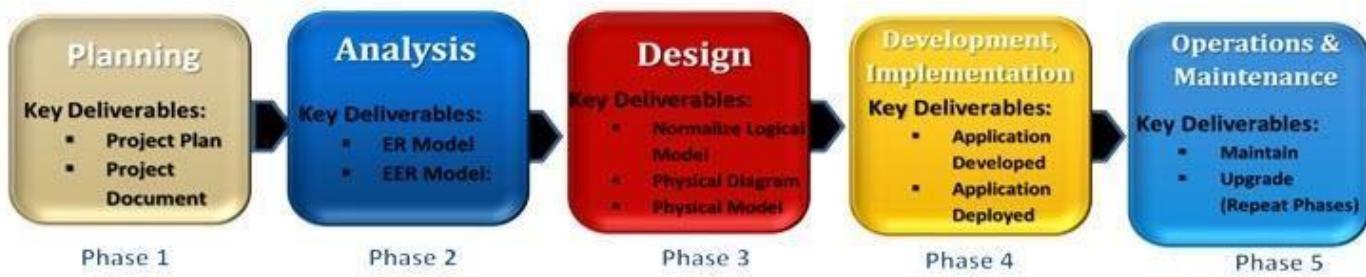
EZRENTAL Inc. is going to use Oracle Server 18c Express Edition as their database management system. We will install Oracle Server 18c Express Edition on **EZRENTAL Inc.** along with Oracle SQL Developer to create the following database development center.

Standalone Development Environment



Project Management Methodology

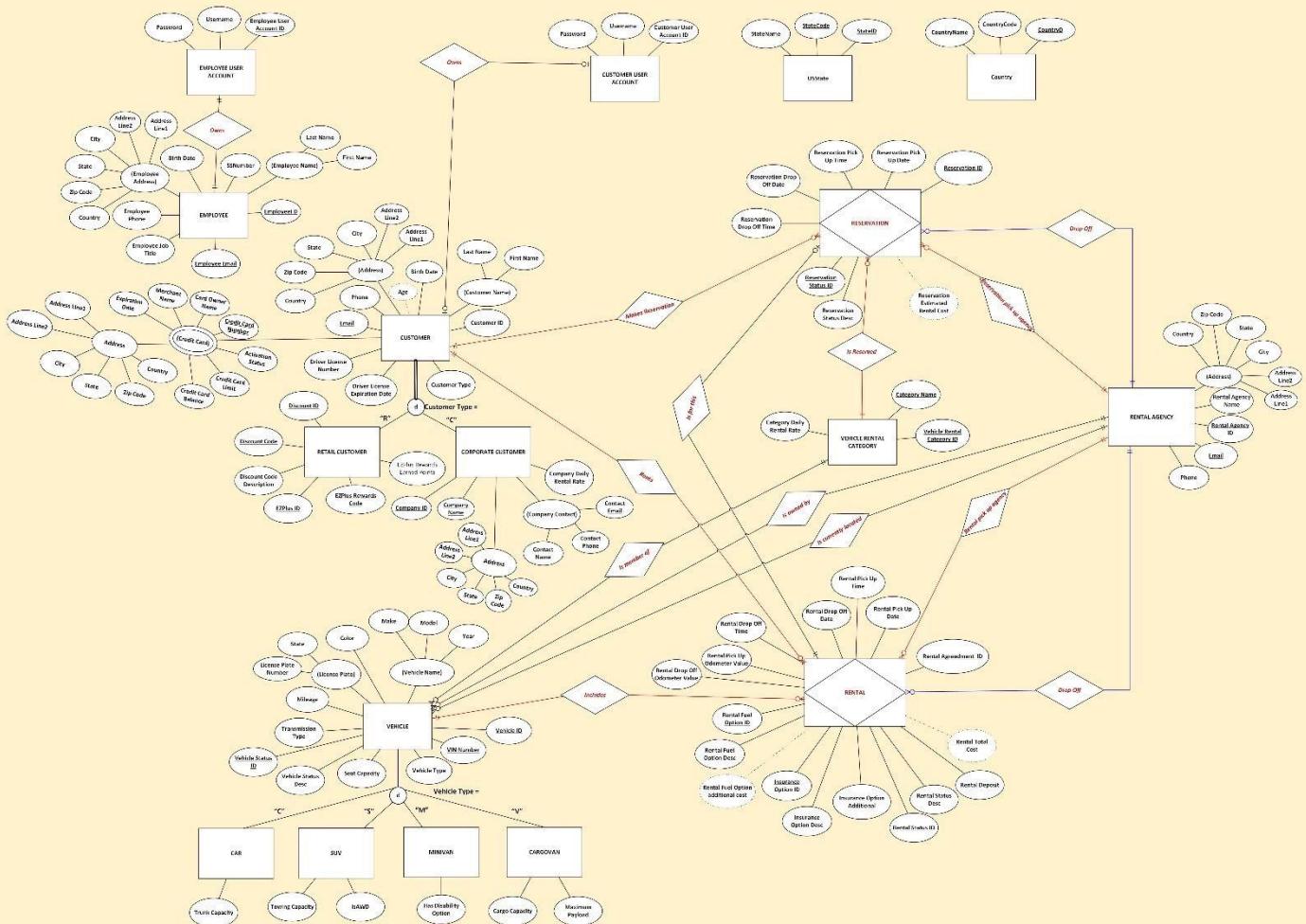
For **EZRental Inc.** we are going to use Waterfall methodology whose diagram you can below. Following this methodology, we are going to plan the entire project and gather all requirements for the project. In the next phase we will perform deep analysis of the requirements. In next Design phase we will start organizing all requirements also in the same phase we will buy hardware such as servers and get software from vendors in our case from Oracle. In next implementation phase we will create the databases and testing, documentation will happen in this phase. The last phase is Maintenance\Administration which means we will look after the system in case of a data loss or any other disaster.



ER/EER Conceptual Model

In the following diagram EZRental Inc. has all his entities, attributes and the cardinalities. This EER Conceptual Model is going to help us with Databases Development in the 18c Oracle.

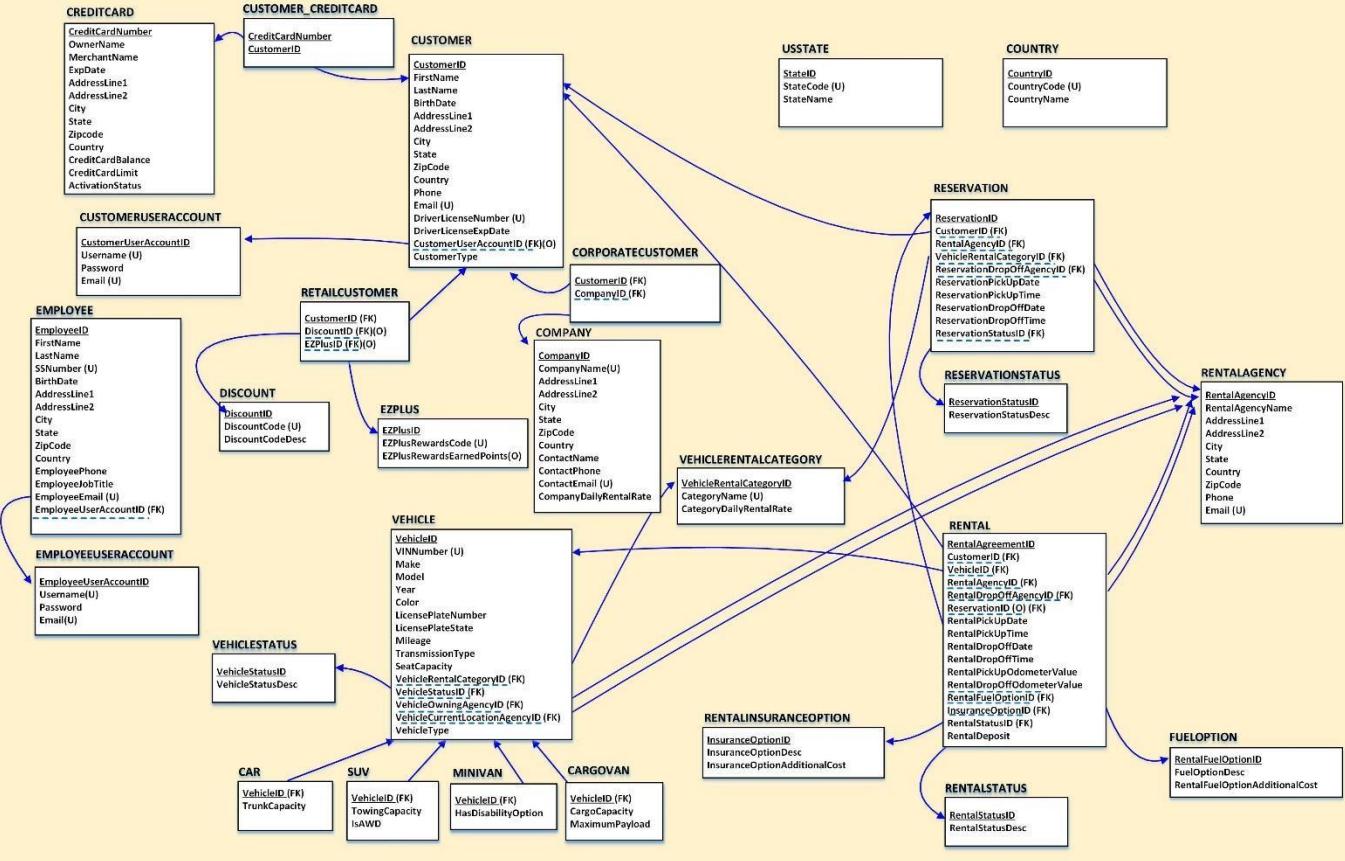
Auto Rental System EER Conceptual Model With Associative Entity Conversions



Normalized Logical Model

In the following diagram data redundancy is removed by creating each table with unique columns (fields) in it and when data will be stored in the columns it will match the column data type. Also, Normalized Logical Model help us creating databases with data consistency.

Auto Rental Management System Normalized Logical Model



Physical Model Data Dictionary

Following are the Data Dictionary tables for EZRental Inc. It contains the name of the columns with the data types.

Customer						
Attribute/Column Name	General Data Type Name	Oracle Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/purpose
CustomerID	IDENTITY	IDENTITY	YES		PRIMARY KEY	Unique identifier for a customer instance. This Primary Key has no Business meaning.
FirstName	String	VARCHAR2(50)	YES	50	NOT NULL	First name of customer
LastName	String	VARCHAR2(50)	YES	50	NOT NULL	Last name of customer

BirthDate	DATE	DATE	YES		MM/DD/YY	Date of birth
AddressLine1	String	VARCHAR2(50)	YES	50	NOT NULL	Home Address Street #1
AddressLine2	String	VARCHAR2(50)	YES	50	NOT NULL	Home Address Street #2
City	String	VARCHAR2(25)	YES	25	NOT NULL	City of customer
StateCode	String	CHAR(2)	YES	2	NOT NULL	2 letter abbreviation of state in US.
ZipCode	String	VARCHAR2(10)	YES	10	NOT NULL	Area zip code of customer in US
Country	String	VARCHAR2(100)	YES	100	NOT NULL	Country name with international scope
Phone	String	VARCHAR2(20)	YES	20	NOT NULL	Phone scope is international
Email	String	VARCHAR2(100)	YES	100	Unique NOT NULL	Customer Email with international scope
DriverLicenseNumber	String	VARCHAR2(15)	YES	15	Unique NOT NULL	Customer License Number
DriverLicenseExpDate	Date	Date	Yes		NOT NULL	Customer License Expiration Date
CustomerUserAccountID	BINARY DATA	RAW(16)	YES	16	Optional Foreign Key (Can be NULL)	Customer User Account ID
CustomerType	String	CHAR(1)	YES	1	NOT NULL	Customer Type

CREDIT CARD

Attribute/Column Name	General Data Type Name	Oracle Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/ purpose
CreditCardNumber	String	VARCHAR2(16)	YES	16	PRIMARY Key	For a credit card we will use a string of character size 16.
OwnerName	String	VARCHAR2(25)	YES	25	NOT NULL	Credit Card Owner Name
MerchantName	String	VARCHAR2(25)	YES	25	NOT NULL	Name of the merchant
ExpDate	DATE	DATE	YES		MM/DD/YY	Exp Date
AddressLine1	String	VARCHAR2(50)	YES	50	NOT NULL	Home Address Street #1
AddressLine2	String	VARCHAR2(50)	YES	50	NOT NULL	Home Address Street #2
City	String	VARCHAR2(25)	YES	25	NOT NULL	City of customer
StateCode	String	CHAR(2)	YES	2	NOT NULL	2 letter abbreviation of state in US.
ZipCode	String	VARCHAR2(10)	YES	10	NOT NULL	Area zip code of customer in US
Country	String	VARCHAR2(100)	YES	100	NOT NULL	Country name with international scope
CreditCardBalance	NUMBER	NUMBER(X,Y)	YES	X=8 Y=2	NOT NULL	Balance of credit card

CreditCardLimit	NUMBER	NUMBER(X,Y)	YES	X=8 Y=2	NOT NULL	Limit of credit card
ActivationStatus	String	CHAR(1)	YES	1	NOT NULL	Status of Credit Card 'A' for active 'D' for Deactive

CUSTOMER CREDIT CARD						
Attribute/Column Name	General Data Type Name	Oracle Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/ purpose
CreditCardNumber	String	VARCHAR2(16)	YES	16	Foreign Key	For a credit card we will use a STRING OF CHARACTERS size 16.
CustomerID	NUMBER	NUMBER(8)	YES	8	Foreign Key	CustomerID is foreign key from customer table.

CUSTOMER USER ACCOUNT						
Attribute/Column Name	General Data Type Name	Oracle Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/ purpose
CustomerUserAccountID	BINARY DATA	RAW(16) SYS_GUID()	YES	16	Primary Key	Customer Account ID. Since it has no business meaning in this table.
Username	STRING	VARCHAR2(25)	YES	25	UNIQUE NOT NULL	Customer username for the account
Password	STRING	VARCHAR2(25)	YES	25	NOT NULL	Password of the customer account
Email	STRING	VARCHAR2(100)	YES	100	Unique NOT NULL	Customer Email with international scope

DISCOUNT						
Attribute/Column Name	General Data Type Name	Oracle Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/ purpose
DiscountID	IDENTITY	IDENTITY	YES		Primary Key	DiscountID is identity data type since it has no business meaning.
DiscountCode	String	VARCHAR2(25)	YES	25	Unique NOT NULL	Discount Code
DiscountCodeDesc	String	VARCHAR2(25)	YES	25	NOT NULL	Discount Count in descending order.

EZPLUS						
Attribute/Column Name	General Data Type Name	Oracle Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/ purpose
EZPlusID	IDENTITY	IDENTITY	YES		Primary Key	EZPlusID is Identity since it has no business meaning.
EZPlusRewardsCode	STRING	VARCHAR2(25)	YES	25	Unique NOT NULL	Rewards Code
EZPlusRewardsEarnedPoints	NUMBER	NUMBER(10)	NO	10	Optional (Can be Null)	Points earned from using EZPlus.

RETAILCUSTOMER						
Attribute/Column Name	General Data Type Name	Oracle Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/ purpose
CustomerID	NUMBER	NUMBER(8)	YES	8	Foreign Key	CustomerID is foreign key from customer table.
DiscountID	NUMBER	NUMBER(8)	NO	8	Optional Foreign Key (Can be Null)	DiscountID is foreign key from discount table.
EZPlusID	NUMBER	NUMBER(5)	NO	5	Optional Foreign Key (Can be Null)	EZPlusID is foreign key from EZPLUS table .

CORPORATECUSTOMER						
Attribute/Column Name	General Data Type Name	Oracle Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/ purpose
CustomerID	NUMBER	NUMBER(8)	YES	8	Foreign Key	CustomerID is foreign key from customer table.
CompanyID	NUMBER	NUMBER(10)	YES	10	Foreign Key	CompanyID is foreign key from company table.

COMPANY

Attribute/Column Name	General Data Type Name	Oracle Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/purpose
CompanyID	NUMBER	NUMBER(10)	YES	10	PRIMARY Key	Unique identifier for a customer instance. This Primary Key has Business meaning.
CompanyName	String	VARCHAR2(25)	YES	25	Unique NOT NULL	Name of the Company
AddressLine1	String	VARCHAR2(50)	YES	50	NOT NULL	Home Address Street #1
AddressLine2	String	VARCHAR2(50)	YES	50	NOT NULL	Home Address Street #2
City	String	VARCHAR2(25)	YES	25	NOT NULL	City of customer
StateCode	String	CHAR(2)	YES	2	NOT NULL	2 letter abbreviation of state in US.
ZipCode	String	VARCHAR2(10)	YES	10	NOT NULL	Area zip code of customer in US
Country	String	VARCHAR2(100)	YES	100	NOT NULL	Country name with international scope
ContactName	String	VARCHAR2(25)	YES	25	NOT NULL	Name of the person to contact
ContactPhone	String	VARCHAR2(20)	YES	20	NOT NULL	Phone scope is international
ContactEmail	String	VARCHAR2(100)	YES	100	Unique NOT NULL	Customer Email with international scope
CompanyDailyRentalRate	NUMBER	DEC(X,Y)	YES	X=5 Y=2	NOT NULL	Rental Rate

CUSTOMER USER ACCOUNT						
Attribute/Column Name	General Data Type Name	Oracle Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/purpose
CustomerUserAccountID	IDENTITY	IDENTITY	YES		Primary Key	Customer Account ID
Username	STRING	VARCHAR2(25)	YES	25	UNIQUE NOT NULL	Customer username for the account
Password	STRING	VARCHAR2(25)	YES	25	NOT NULL	Password of the customer account
Email	STRING	VARCHAR2(100)	YES	100	Unique NOT NULL	Customer Email with international scope

EMPLOYEE						
Attribute/Column Name	General Data Type Name	Oracle Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/purpose

EmployeeID	IDENTITY	IDENTITY	YES		PRIMARY KEY	Unique identifier for a employee instance. This Primary Key has no Business meaning.
FirstName	STRING	VARCHAR2(50)	YES	50	NOT NULL	First name of employee
LastName	STRING	VARCHAR2(50)	YES	50	NOT NULL	Last name of employee
SSNumber	STRING	CHAR(9)	YES	9	Unique NOT NULL	Social security number of an employee.
BirthDate	DATE	DATE	YES		MM/DD/YY	Date of birth of employee
AddressLine1	STRING	VARCHAR2(50)	YES	50	NOT NULL	Home Address Street #1
AddressLine2	STRING	VARCHAR2(50)	YES	50	NOT NULL	Home Address Street #2
City	STRING	VARCHAR2(25)	YES	25	NOT NULL	City of customer
StateCode	STRING	CHAR(2)	YES	2	NOT NULL	2 letter abbreviation of state in US.
ZipCode	STRING	VARCHAR2(10)	YES	10	NOT NULL	Area zip code of customer in US
Country	STRING	VARCHAR2(100)	YES	100	NOT NULL	Country name with international scope
EmployeePhone	STRING	VARCHAR2(20)	YES	20	NOT NULL	Phone scope is international
EmployeeJobTitle	STRING	VARCHAR2(25)	YES	25	NOT NULL	Job title of Employee
EmployeeEmail	STRING	VARCHAR2(100)	YES	100	Unique NOT NULL	Employee email with international scope
EmployeeUserAccountID	NUMBER	NUMBER(8)	YES		Foreign Key	Employee User Account ID

EMPLOYEE USER ACCOUNT						
Attribute/Column Name	General Data Type Name	Oracle Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/purpose
EmployeeUserAccountID	IDENTITY	IDENTITY	YES		Primary Key	Employee Account ID
Username	STRING	VARCHAR2(25)	YES	25	UNIQUE NOT NULL	Employee username for the account
Password	STRING	VARCHAR2(25)	YES	25	NOT NULL	Password of the employee account
Email	STRING	VARCHAR2(100)	YES	100	Unique NOT NULL	Employee mail with international scope

VEHICLE						
Attribute/Column Name	General Data Type Name	Oracle Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/ purpose
VehicleID	IDENTITY	IDENTITY	YES		PRIMARY Key	the Vehicle ID has no business meaning. So, the data type will be identity.
VINNumber	STRING	CHAR(17)	YES	17	Unique NOT NULL	Unique identifier for a car.
Make	STRING	VARCHAR2(25)	YES	25	NOT NULL	Company name
Model	STRING	VARCHAR2(25)	YES	25	NOT NULL	Car Model name
Year	STRING	CHAR(4)	YES	4	NOT NULL	(YYYY) Year the car was made.
Color	STRING	VARCHAR2(15)	YES	15	NOT NULL	Color of the car
LicensePlateNumber	STRING	VARCHAR2(10)	YES	10	NOT NULL	Plate number of a car
LicensePlateState	STRING	CHAR(2)	YES	2	NOT NULL	2 letter abbreviation of state in US.
Mileage	NUMBER	NUMBER(10)	YES	10	NOT NULL	Car Mileage
TransmissionType	STRING	VARCHAR(25)	YES	25	NOT NULL	Transmission type of a car
SeatCapacity	STRING	VARCHAR2(3)	YES	3	NOT NULL	Seat capacity of a car.
VehicleRentalCategoryID	NUMBER	NUMBER(2)	YES	2	Foreign Key	Vehicle Rental CategoryID.
VehicleStatusID	NUMBER	NUMBER(1)	YES	1	Foreign Key	Vehicle status id.
VehicleOwningAgencyID	NUMBER	NUMBER(3)	YES	3	Foreign Key	Vehicle Owning AgencyID
VehicleCurrentlocationAgentID	STRING	VARCHAR2(4)	YES	4	Foreign Key	Vehicle Current Location AgentID
VehicleType	STRING	CHAR(1)	YES	1	NOT NULL	Vehicle type like Car,SUV, Minivan,Cargo Van

VEHICLE STATUS						
Attribute/Column Name	General Data Type Name	Oracle Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/ purpose

VehicleStatusID	NUMBER	NUMBER(1)	YES	1	Primary Key	Vehicle status id.
VehicleStatusDesc	STRING	VARCHAR2(100)	YES	25	NOT NULL	Description of a vehicle.

CAR						
Attribute/Column Name	General Data Type Name	Oracle Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/purpose
VehicleID	NUMBER	NUMBER(8)	YES	8	Foreign Key	VehicleID is foreign key from vehicle table.
TrunkCapacity	NUMBER	NUMBER(5,2)	YES	X = 5 Y = 2	NOT NULL	Trunk capacity in a cubic foot

SUV						
Attribute/Column Name	General Data Type Name	Oracle Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/purpose
VehicleID	NUMBER	NUMBER(8)	YES	8	Foreign Key	VehicleID is foreign key from vehicle table.
TowingCapacity	NUMBER	NUMBER(5)	YES	5	NOT NULL	Towing capacity in a cubic foot
IsAWD	STRING	VARCHAR2(2)	YES	2	NOT NULL	How many wheeler is the SUV.

MINIVAN						
Attribute/Column Name	General Data Type Name	Oracle Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/purpose
VehicleID	NUMBER	NUMBER(8)	YES	8	Foreign Key	VehicleID is foreign key from vehicle table.
HasDisabilityOption	STRING	CHAR(1)	YES	1	NOT NULL	Y/N if the vehicle contains a disability chair

CARGOVAN						
Attribute/Column Name	General Data Type Name	Oracle Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/purpose

VehicleID	NUMBER	NUMBER(8)	YES	8	Foreign Key	VehicleID is foreign key from vehicle table.
CargoCapacity	NUMBER	NUMBER(4)	YES	4	NOT NULL	Cargo Capacity in cubic foot.
MaximumPayLoad	NUMBER	NUMBER(10)	YES	10	NOT NULL	Weight in pounds

VEHICLE RENTAL CATEGORY						
Attribute/Column Name	General Data Type Name	Oracle Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/purpose
VehicleRentalCategoryID	NUMBER	NUMBER(2)	YES	2	Primary Key	Vehicle Rental CategoryID.
CategoryName	STRING	VARCHAR2(25)	YES	25	NOT NULL	Vehicle category name
CategoryDailyRentalRate	DECIMAL	DECIMAL(5,2)	YES	X=5 Y=2	NOT NULL	Rental rate

US STATE						
Attribute/Column Name	General Data Type Name	Oracle Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/purpose
StateID	NUMBER	NUMBER(2)	YES	2	Primary Key	StateID in integer foam
StateCode	STRING	CHAR(2)	YES	2	Unique NOT NULL	2 letter abbreviation of state in US.
StateName	STRING	VARCHAR2(25)	YES	25	NOT NULL	US state full 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(2)	YES	2	Primary Key	Country ID in integer foam
CountryCode	STRING	CHAR(2)	YES	2	Unique NOT NULL	Country Code
CountryName	STRING	VARCHAR2(100)	YES	25	NOT NULL	Country Name

RESERVATION STATUS						
Attribute/Column Name	General Data Type Name	Oracle Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/purpose

ReservationStatusID	NUMBER	NUMBER(8)	YES	8	Primary Key	ReservationStatusID has a business meaning. So, it is an NUMBER data type.
ReservationStatusDesc	STRING	VARCHAR2(100)	YES	25	NOT NULL	Description of a vehicle.

RENTAL AGENCY						
Attribute/Column Name	General Data Type Name	Oracle Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/purpose
RentalAgencyID	NUMBER	NUMBER(8)	YES	8	PRIMARY KEY	Unique identifier for a rental agency instance. This Primary Key has no Business meaning.
RentalAgencyName	STRING	VARCHAR2(50)	YES	50	NOT NULL	Agency name
AddressLine1	STRING	VARCHAR2(50)	YES	50	NOT NULL	Rental Agency Address Street #1
AddressLine2	STRING	VARCHAR2(50)	YES	50	NOT NULL	Rental Agency Address Street #2
City	STRING	VARCHAR2(25)	YES	25	NOT NULL	City of Rental Agency
StateCode	STRING	CHAR(2)	YES	2	NOT NULL	2 letter abbreviation of state in US.
ZipCode	STRING	VARCHAR2(10)	YES	10	NOT NULL	Area zip code of agency in US
Country	STRING	VARCHAR2(100)	YES	100	NOT NULL	Country name with international scope
Phone	STRING	VARCHAR2(20)	YES	20	NOT NULL	Phone scope is international
Email	STRING	VARCHAR2(100)	YES	100	Unique NOT NULL	Agency email with international scope

RESERVATION						
Attribute/Column Name	General Data Type Name	Oracle Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/purpose
ReservationID	NUMBER	NUMBER(8)	YES	8	Primary Key	ReservationID has a business meaning. So, the data type is NUMBER.
CustomerID	NUMBER	NUMBER(8)	YES	8	Foreign Key	CustomerID is foreign key from customer table.
RentalAgencyID	NUMBER	NUMBER(8)	YES	8	Foreign KEY	Foreign Key from Rental Agency table

VehicleRentalCategoryID	NUMBER	NUMBER(2)	YES	2	Foreign Key	Vehicle Rental CategoryID.
ReservationDropOffAgencyID	NUMBER	NUMBER(8)	YES	8	Foreign Key	Reservaton AgencyID
ReservationPickUpDate	DATE	DATE	YES		NOT NULL	Date of car pick up.
ReservationPickUpTime	TIME	TIMESTAMP	YES		NOT NULL	Reservation pick up time
ReservationDropOffDate	DATE	DATE	YES		NOT NULL	Date of car drop off.
ReservationDropOffTime	TIME	TIMESTAMP	YES		NOT NULL	Reservation drop of time
ReservationStatusID	NUMBER	NUMBER(8)	YES	8	Foreign Key	ReservationStatusID from reservation status table.

RENTAL STATUS						
Attribute/Column Name	General Data Type Name	Oracle Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/purpose
RentalStatusID	NUMBER	NUMBER(1)	YES	1	Primary Key	RentalStatusID is a single value.
RentalStatusDesc	STRING	VARCHAR2(100)	YES	25	NOT NULL	Description of a rental status

RENTAL INSURANCE OPTION						
Attribute/Column Name	General Data Type Name	Oracle Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/purpose
InsuranceOptionID	NUMBER	NUMBER(1)	YES	1	Primary Key	InsuranceOptionID is a single value
InsuranceOptionDesc	STRING	VARCHAR2(100)	YES	25	NOT NULL	Description of a insurance option
InsuranceOptionAdditionalCost	NUMBER	NUMBER(4,2)	YES	X=4 Y=2	NOT NULL	Additional Cost of insurance

FUEL OPTION						
Attribute/Column Name	General Data Type Name	Oracle Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/purpose
RentalFuelOptionID	NUMBER	NUMBER(1)	YES	1	Primary Key	RentalFuelOptionID is a single value
FuelOptionDesc	STRING	VARCHAR2(100)	YES	25	NOT NULL	Description of fuel option

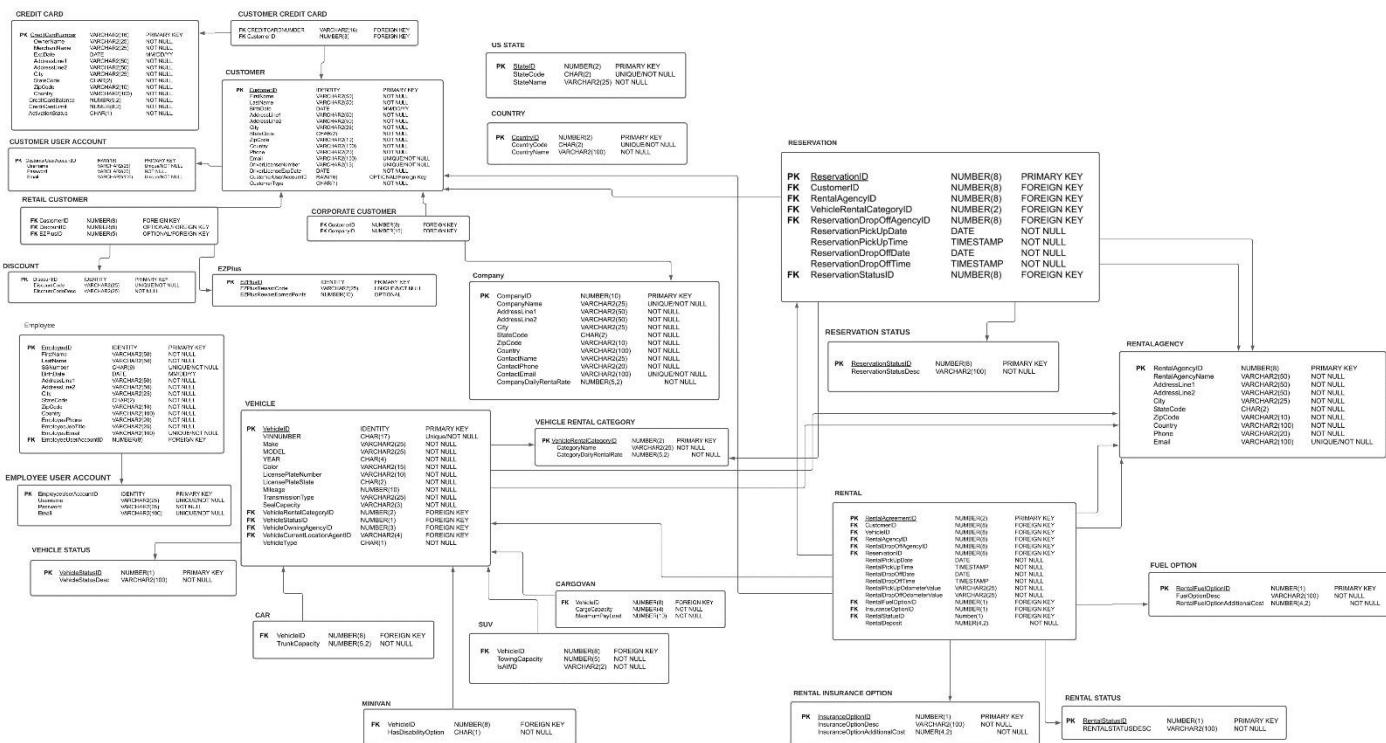
RentalFuelOptionAdditionalCost	NUMBER	NUMBER(4,2)	YES	X=4 Y=2	NOT NULL	Additional Cost of fuel
--------------------------------	--------	-------------	-----	------------	----------	-------------------------

RENTAL						
Attribute/Column Name	General Data Type Name	Oracle Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/purpose
RentalAgreementID	NUMBER	NUMBER(2)	YES	2	Primary Key	RentalAgreementID has a business meaning
CustomerID	NUMBER	NUMBER(8)	YES	8	Foreign Key	CustomerID is foreign key from customer table.
VehicleID	NUMBER	NUMBER(8)	YES	8	Foreign Key	Vehicle ID is foreign key from vehicle table.
RentalAgencyID	NUMBER	NUMBER(8)	YES	8	Foreign KEY	Foreign Key from Rental Agency table
RentalDroppOffAgencyID	NUMBER	NUMBER(8)	YES	8	Foreign KEY	Foreign Key from Rental Agency table
ReservationID	NUMBER	NUMBER(8)	YES	8	Foreign Key	Foreign Key from Reservation table.
RentalPickUpDate	DATE	DATE	YES		NOT NULL	Date of rental pick up.
RentalPickUpTime	TIME	TIMESTAMP	YES		NOT NULL	Rental pick up time.
RentalDropOffDate	DATE	DATE	YES		NOT NULL	Date of rental drop off.
RentalDropOffTime	TIME	TIMESTAMP	YES		NOT NULL	Rental drop off time.
RentalPickUpOdometerValue	STRING	VARCHAR2(25)	YES	25	NOT NULL	Rental car milage traveled before pick up.
RentalDropOffOdemeterValue	STRING	VARCHAR2(25)	YES	25	NOT NULL	Rental car milage traveled after drop off.
RentalFuelOptionID	NUMBER	NUMBER(1)	YES	1	Foreign Key	RentalFuelOptionID is a single value and foreign key from fuel option.
InsuranceOptionID	NUMBER	NUMBER(1)	YES	1	Foreign Key	InsuranceOptionID is a single value and foreign key from rental insurance option.

RentalStatusID	NUMBER	NUMBER(1)	YES	1	Foreign Key	RentalStatusID is a single value and foreign key from rental status table.
RentalDeposit	NUMBER	NUMBER(4,2)	YES	X=4 Y=2	NOT NULL	Rental Deposit

Physical Model Schema Diagram

Following is physical model schema diagram drawn to show the names of the tables with columns names. Also, it shows the relationship between the tables.



Development & Implementation

Following is the sql script which is used to create 27 tables of EZRental Inc.

```
/* Create Customer User Account Table */

CREATE TABLE CustomerUserAccount
(
    CustomerUserAccountId           RAW(16)          DEFAULT
    SYS_GUID(),
    Username                        VARCHAR2(25)      NOT NULL  UNIQUE,
    Password                         VARCHAR2(25)      NOT NULL,
    Email                            VARCHAR2(100)     NOT NULL  UNIQUE,
    CONSTRAINT CustomerUserAccountId_PK PRIMARY KEY (CustomerUserAccountId)
);

/* Create Customer Table */

CREATE TABLE Customer
(
    CustomerID                      NUMBER           GENERATED ALWAYS AS IDENTITY,
    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(25)      NOT NULL,
    StateCode                        CHAR(2)          NOT NULL,
    ZipCode                          VARCHAR2(10)      NOT NULL,
    Country                          VARCHAR2(100)     NOT NULL,
    Phone                            VARCHAR2(20)      NOT NULL,
    Email                            VARCHAR2(100)     NOT NULL  UNIQUE,
    DriverLicenseNumber              VARCHAR2(15)      NOT NULL  UNIQUE,
    DriverLicenseExpDate            DATE             NOT NULL,
    CustomerUserAccountId           RAW(16)          NULL,
    CustomerType                     CHAR(1)          NOT NULL,
```

```
CONSTRAINT CustomerID_PK PRIMARY KEY (CustomerID),  
  
CONSTRAINT CustomerUserAccountID_FK  
FOREIGN KEY (CustomerUserAccountID)  
REFERENCES CustomerUserAccount(CustomerUserAccountID)  
);
```

```
/* Create Credit Card Table */
```

```
CREATE TABLE CreditCard  
(  
    CreditCardNumber      VARCHAR2(16)      NOT NULL,  
    OwnerName             VARCHAR2(25)      NOT NULL,  
    MerchantName          VARCHAR2(25)      NOT NULL,  
    ExpDate               DATE              NOT NULL,  
    AddressLine1          VARCHAR2(50)      NOT NULL,  
    AddressLine2          VARCHAR2(50)      NOT NULL,  
    City                  VARCHAR2(25)      NOT NULL,  
    StateCode             CHAR(2)           NOT NULL,  
    ZipCode               VARCHAR2(10)      NOT NULL,  
    Country               VARCHAR2(100)     NOT NULL,  
    CreditCardBalance     NUMBER(5,2)        NOT NULL,  
    CreditCardLimit       NUMBER(8,2)        NOT NULL,  
    ActivationStatus      CHAR(1)            NOT NULL,
```

```
CONSTRAINT CreditCard_PK PRIMARY KEY (CreditCardNumber)
```

```
);
```

```
/* Create US State Table */
```

```
CREATE TABLE US_State  
(  
    StateID               NUMBER(2)        NOT NULL,  
    StateCode              CHAR(2)          NOT NULL      UNIQUE,  
    StateName              VARCHAR2(25)     NOT NULL,
```

```
CONSTRAINT StateID_PK PRIMARY KEY (StateID)
```

```
);
```

```
/* Create Country Table */
```

```
CREATE TABLE Country
```

```
(  
    CountryID      NUMBER(2) NOT NULL,  
    CountryCode    CHAR(2)      NOT NULL UNIQUE,  
    CountryName   VARCHAR2(100) NOT NULL,  
    CONSTRAINT CountryID_PK PRIMARY KEY (CountryID)  
);
```

```
/* Create Customer Credit Card Table With 2 Foreign Keys from Customer  
and CreditCard Tables*/
```

```
CREATE TABLE CustomerCreditCard  
(  
    CreditCardNumber  VARCHAR2(16)      NOT NULL,  
    CustomerID       NUMBER(8) NOT NULL,  
  
    CONSTRAINT CreditCardNumber_FK  
    FOREIGN KEY (CreditCardNumber)  
    REFERENCES CreditCard(CreditCardNumber),  
  
    CONSTRAINT CustomerID_FK  
    FOREIGN KEY (CustomerID)  
    REFERENCES Customer(CustomerID)  
);
```

```
/* Create Discount Table */
```

```
CREATE TABLE Discount  
(  
    DiscountID      NUMBER          GENERATED ALWAYS AS IDENTITY,  
    DiscountCode    VARCHAR2(25)     NOT NULL UNIQUE,  
    DiscountCodeDesc VARCHAR2(25)     NOT NULL,  
  
    CONSTRAINT DiscountID_PK PRIMARY KEY (DiscountID)  
);
```

```
/* Create EZPlus Table */
```

```
CREATE TABLE EZPlus  
(  
    EZPlusID        NUMBER          GENERATED ALWAYS AS  
IDENTITY,  
    EZPlusRewardCode VARCHAR2(25)     NOT NULL UNIQUE,  
    EZPlusRewardsEarnedPoints NUMBER(10) NULL,
```

```

CONSTRAINT EZPlusID_PK PRIMARY KEY (EZPlusID)
);
/* Create Retail Customer Table With 3 Relationships */

CREATE TABLE RetailCustomer
(
    CustomerID      NUMBER(8)  NOT NULL,
    DiscountID     NUMBER(8)   NULL,
    EZPlusID       NUMBER(5)   NULL,
    CONSTRAINT RetailCustomer_CustomerID_FK
    FOREIGN KEY (CustomerID)
    REFERENCES Customer(CustomerID),
    CONSTRAINT DiscountID_FK
    FOREIGN KEY (DiscountID)
    REFERENCES Discount(DiscountID),
    CONSTRAINT EZPlusID_FK
    FOREIGN KEY (EZPlusID)
    REFERENCES EZPlus(EZPlusID)
);

/* Create Employee User Account Table */

CREATE TABLE EmployeeUserAccount
(
    EmployeeUserAccountId      NUMBER          GENERATED ALWAYS AS
IDENTITY,
    Username                  VARCHAR2(25) NOT NULL UNIQUE,
    Password                  VARCHAR2(25) NOT NULL,
    Email                     VARCHAR2(100) NOT NULL UNIQUE,
    CONSTRAINT EmployeeUserAccountId_FK PRIMARY KEY (EmployeeUserAccountId)
);

/* Create Employee Table With One Foreign Key*/

CREATE TABLE Employee
(
    EmployeeID      NUMBER          GENERATED ALWAYS AS IDENTITY,
    FirstName       VARCHAR2(50) NOT NULL,
    LastName        VARCHAR2(50) NOT NULL,
    SSNNumber       CHAR(9)        NOT NULL UNIQUE,

```

```

BirthDate           DATE          NOT NULL,
AddressLine1        VARCHAR2(50)   NOT NULL,
AddressLine2        VARCHAR2(50)   NOT NULL,
City                VARCHAR2(25)   NOT NULL,
StateCode           CHAR(2)       NOT NULL,
ZipCode             VARCHAR2(10)   NOT NULL,
Country             VARCHAR2(100)  NOT NULL,
EmployeePhone       VARCHAR2(20)   NOT NULL,
EmployeeJobTitle    VARCHAR2(25)   NOT NULL,
EmployeeEmail        VARCHAR2(100)  NOT NULL      UNIQUE,
EmployeeUserAccountID NUMBER(8)   NOT NULL,

```

CONSTRAINT EmployeeID_PK PRIMARY KEY (EmployeeID),

CONSTRAINT Employee_EmployeeUserAccountID_FK
FOREIGN KEY (EmployeeUserAccountID)
REFERENCES EmployeeUserAccount(EmployeeUserAccountID)

);

/* Create Vehicle Status Table */

CREATE TABLE VehicleStatus

(

```

VehicleStatusID      NUMBER(1)   NOT NULL,
VehicleStatusDesc    VARCHAR2(100) NOT NULL,

```

CONSTRAINT VehicleStatusID_PK PRIMARY KEY (VehicleStatusID)

);

/* Create Company Table */

CREATE TABLE Company

(

```

CompanyID           NUMBER(10)  NOT NULL,
CompanyName         VARCHAR2(25) NOT NULL      UNIQUE,
AddressLine1        VARCHAR2(50) NOT NULL,
AddressLine2        VARCHAR2(50) NOT NULL,
City                VARCHAR2(25) NOT NULL,
StateCode           CHAR(2)      NOT NULL,
ZipCode             VARCHAR2(10) NOT NULL,
Country             VARCHAR2(100) NOT NULL,
ContactName         VARCHAR2(25) NOT NULL,
ContactPhone        VARCHAR2(20) NOT NULL,
ContactEmail        VARCHAR2(100) NOT NULL      UNIQUE,

```

```
    CompanyDailyRentalRate      NUMBER(5,2)      NOT NULL,  CONSTRAINT  
CompanyID_PK PRIMARY KEY (CompanyID)  
);
```

```
/* Create Corporate Customer with two relationship (Foreign Key) */
```

```
CREATE TABLE CorporateCustomer  
(  
    CustomerID   NUMBER(8)  NOT NULL,  
    CompanyID    NUMBER(10) NOT NULL,  
  
    CONSTRAINT Corporate_Customer_CustomerID_FK  
    FOREIGN KEY (CustomerID)  
    REFERENCES Customer(CustomerID),  
  
    CONSTRAINT Corporate_Customer_CompanyID_FK  
    FOREIGN KEY (CompanyID)  
    REFERENCES Company(CompanyID)  
);
```

```
/* Create Vehicle Rental Category */
```

```
CREATE TABLE VehicleRentalCategory  
(  
    VehicleRentalCategoryID      NUMBER(2)  NOT NULL,  
    CategoryName                VARCHAR2(25)  NOT NULL,  
    CategoryDailyRentalRate     NUMBER(5,2) NOT NULL,  
  
    CONSTRAINT VehicleRentalCategoryID_PK PRIMARY KEY (VehicleRentalCategoryID)  
);
```

```
/* Create Fuel Option Table */
```

```
CREATE TABLE FuelOption  
(  
    RentalFuelOptionID          NUMBER(1)  NOT NULL,  
    FuelOptionDesc               VARCHAR2(100)  NOT NULL,  
    RentalFuelOptionAdditionalCost  NUMBER(4,2) NOT NULL,  
  
    CONSTRAINT RentalFuelOptionID_PK PRIMARY KEY (RentalFuelOptionID)  
);
```

```
/* Create Rental Status Table */
```

```

CREATE TABLE RentalStatus
(
    RentalStatusID      NUMBER(1) NOT NULL,
    RentalStatusDesc    VARCHAR2(100) NOT NULL,
    CONSTRAINT RentalStatusID_PK PRIMARY KEY (RentalStatusID)
);

/* Create Rental Insurance Option Table */

CREATE TABLE RentalInsuranceOption
(
    InsuranceOptionID      NUMBER(1) NOT NULL,
    InsuranceOptionDesc    VARCHAR2(100) NOT NULL,
    InsuranceOptionAdditionalCost NUMBER(4,2) NOT NULL,
    CONSTRAINT InsuranceOptionID_PK PRIMARY KEY (InsuranceOptionID)
);

/* Create Rental Agency Table */

CREATE TABLE RentalAgency
(
    RentalAgencyID      NUMBER(8) NOT NULL,
    RentalAgencyName    VARCHAR2(50) NOT NULL,
    AddressLine1        VARCHAR2(50) NOT NULL,
    AddressLine2        VARCHAR2(50) NOT NULL,
    City                VARCHAR2(25) NOT NULL,
    StateCode           CHAR(2) NOT NULL,
    ZipCode              VARCHAR2(10) NOT NULL,
    Country              VARCHAR2(100) NOT NULL,
    Phone                VARCHAR2(20) NOT NULL,
    Email                VARCHAR2(100) NOT NULL UNIQUE,
    CONSTRAINT RentalAgencyID_PK PRIMARY KEY (RentalAgencyID)
);

/* Create Reservation Status Table */

CREATE TABLE ReservationStatus
(
    ReservationStatusID  NUMBER(8) NOT NULL,
    ReservationStatusDesc VARCHAR2(100) NOT NULL,

```

```
CONSTRAINT ReservationStatusID_FK PRIMARY KEY (ReservationStatusID)
);
```

```
/* Create Reservation Table With Foreign Keys*/
```

```
CREATE TABLE Reservation
(
    ReservationID          NUMBER(8) NOT NULL,
    CustomerID             NUMBER(8)      NOT NULL,
    RentalAgencyID         NUMBER(8) NOT NULL,
    VehicleRentalCategoryID NUMBER(2)  NOT NULL,
    ReservationDropOffAgencyID NUMBER(8) NOT NULL,
    ReservationPickUpDate   DATE        NOT NULL,
    ReservationPickUpTime    TIMESTAMP NOT NULL,
    ReservationDropOffDate   DATE        NOT NULL,
    ReservationDropOffTime    TIMESTAMP NOT NULL,
    ReservationStatusID      NUMBER(8) NOT NULL,
```

```
CONSTRAINT ReservationID_PK PRIMARY KEY (ReservationID),
```

```
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_ReservationDropOffAgencyID_FK
FOREIGN KEY (ReservationDropOffAgencyID)
REFERENCES RentalAgency(RentalAgencyID),
```

```
CONSTRAINT Reservation_ReservationStatusID_FK
FOREIGN KEY (ReservationStatusID)
REFERENCES ReservationStatus(ReservationStatusID)
```

```
);
```

```
/* Create Vehicle Table With three Foreign Keys */
```

```
CREATE TABLE Vehicle
```

```

(
    VehicleID          NUMBER          GENERATED ALWAYS AS
IDENTITY,
    VinNumber         CHAR(17)        NOT NULL UNIQUE,
    Make              VARCHAR2(25)   NOT NULL,
    Model             VARCHAR2(25)   NOT NULL,
    Year              CHAR(4)         NOT NULL,
    Color             VARCHAR2(15)   NOT NULL,
    LicensePlateNumber VARCHAR2(10)  NOT NULL,
    LicensePlateState CHAR(2)        NOT NULL,
    Mileage           NUMBER(10)     NOT NULL,
    TransmissionType VARCHAR2(25)   NOT NULL,
    SeatCapacity      VARCHAR2(3)    NOT NULL,
    VehicleRentalCategoryID NUMBER(2) NOT NULL,
    VehicleStatusID   NUMBER(1)      NOT NULL,
    VehicleOwningAgencyID NUMBER(8) NOT NULL,
    VehicleCurrentLocationAgentID NUMBER(8) NOT NULL,
    VehicleType       CHAR(1)        NOT NULL,
)

CONSTRAINT VehicleID_PK PRIMARY KEY (VehicleID),

CONSTRAINT VehicleRentalCategoryID_FK
FOREIGN KEY (VehicleRentalCategoryID)
REFERENCES VehicleRentalCategory(VehicleRentalCategoryID),

CONSTRAINT VehicleStatusID_FK
FOREIGN KEY (VehicleStatusID)
REFERENCES VehicleStatus(VehicleStatusID),

CONSTRAINT VehicleOwningAgencyID_FK
FOREIGN KEY (VehicleOwningAgencyID)
REFERENCES RentalAgency(RentalAgencyID),

CONSTRAINT VehicleCurrentLocationAgentID_FK
FOREIGN KEY (VehicleCurrentLocationAgentID)
REFERENCES RentalAgency(RentalAgencyID)
);

```

/* Create Car Table with Foreign Key */

```

CREATE TABLE Car
(
    VehicleID      NUMBER(8)  NOT NULL,
    TrunkCapacity  NUMBER(5,2) NOT NULL,

```

```

CONSTRAINT VehicleID_FK
FOREIGN KEY (VehicleID)
REFERENCES Vehicle(VehicleID)
);

/* Create SUV Table */

CREATE TABLE SUV
(
    VehicleID          NUMBER(8) NOT NULL,
    TowingCapacity     NUMBER(5)  NOT NULL,
    IsAWD              VARCHAR2(2) NOT NULL,

CONSTRAINT Suv_VehicleID_FK
FOREIGN KEY (VehicleID)
REFERENCES Vehicle(VehicleID)
);

/* Create Minivan Table */

CREATE TABLE Minivan
(
    VehicleID          NUMBER(8) NOT NULL,
    HasDisabilityOption CHAR(1)      NOT NULL,

CONSTRAINT Minivan_VehicleID_FK
FOREIGN KEY (VehicleID)
REFERENCES Vehicle(VehicleID)
);

/* Create Rental Table */

CREATE TABLE RENTAL
(
    RentalAgreementID   NUMBER(2)      NOT NULL,
    CustomerID           NUMBER(8)      NOT NULL,
    VehicleID            NUMBER(8)      NOT NULL,
    RentalAgencyID       NUMBER(8)      NOT NULL,
    RentalDropOffAgencyID NUMBER(8) NOT NULL,
    ReservationID        NUMBER(8)      NOT NULL,
    RentalPickUpDate     DATE          NOT NULL,
    RentalPickUpTime     TIMESTAMP    NOT NULL,
    RentalDropOffDate    DATE          NOT NULL,
    RentalDropOffTime    TIMESTAMP    NOT NULL,

```

```
RentalPickUpOdometerValue    VARCHAR2(25)      NOT NULL,  
RentalDropOffOdometerValue   VARCHAR2(25)      NOT NULL,  
RentalFuelOptionID          NUMBER(1)         NOT NULL,  
InsuranceOptionID           NUMBER(1)         NOT NULL,  
RentalStatusID              NUMBER(1)         NOT NULL,  
RentalDeposit               NUMBER(4,2)        NOT NULL,
```

```
CONSTRAINT RentalAgreementID_PK PRIMARY KEY (RentalAgreementID),
```

```
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),
```

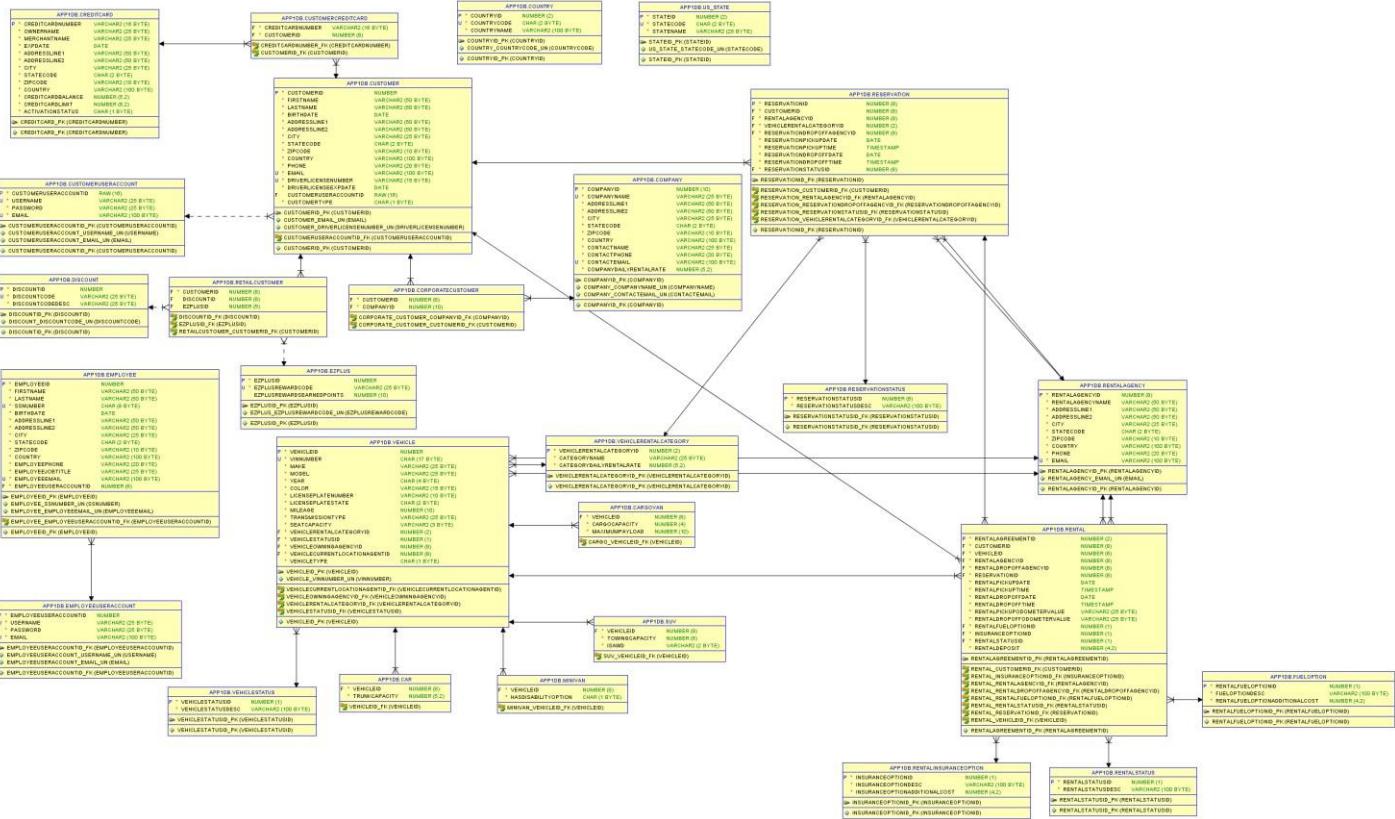
```
CONSTRAINT Rental_RentalFuelOptionID_FK  
FOREIGN KEY (RentalFuelOptionID)  
REFERENCES FuelOption(RentalFuelOptionID),
```

```
CONSTRAINT Rental_InsuranceOptionID_FK  
FOREIGN KEY (InsuranceOptionID)  
REFERENCES RentalInsuranceOption(InsuranceOptionID),
```

```
CONSTRAINT Rental_RentalStatusID_FK  
FOREIGN KEY (RentalStatusID)  
REFERENCES RentalStatus(RentalStatusID)
```

```
);
```

Development & Implementation Physical Schema Diagram



Database Development & Implementation Unit Testing

Insert Statements for Customer User Account

This insert statement contains the data for customer user account which is a foreign key in customer table. Also, it is an example of binary relationship.

```
/* INSERTING INTO CUSTOMERUSERACCOUNT */
/* BINARY ENTITY */
INSERT INTO CUSTOMERUSERACCOUNT
VALUES ('3568714689264256','johnsmith','johnsmith123','johnsmith123@gmail.com');

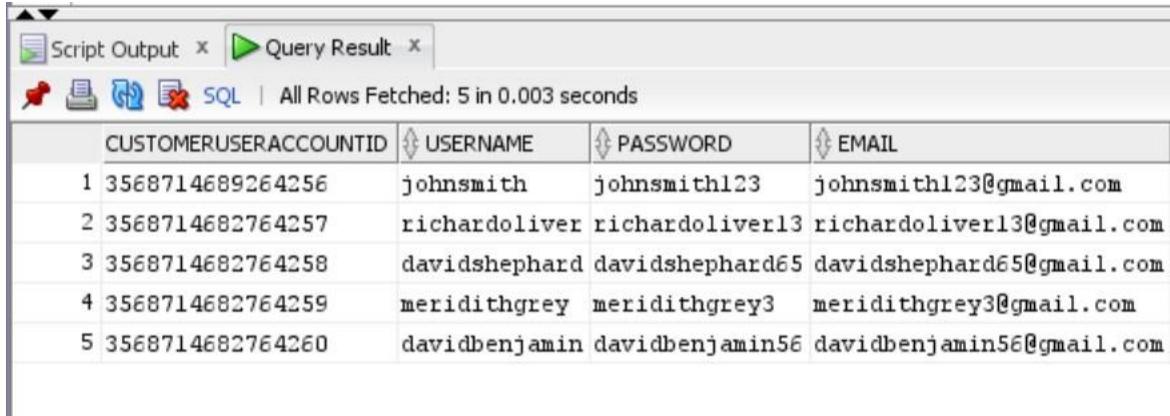
INSERT INTO CUSTOMERUSERACCOUNT
VALUES ('3568714682764257','richardoliver','richardoliver13','richardoliver13@gmail.com');

INSERT INTO CUSTOMERUSERACCOUNT
VALUES
('3568714682764258','davidshephard','davidshephard65','davidshephard65@gmail.com');

INSERT INTO CUSTOMERUSERACCOUNT
VALUES ('3568714682764259','meridithgrey','meridithgrey3','meridithgrey3@gmail.com');

INSERT INTO CUSTOMERUSERACCOUNT
VALUES
('3568714682764260','davidbenjamin','davidbenjamin56','davidbenjamin56@gmail.com');
```

Result of Customer User Account



	CUSTOMERUSERACCOUNTID	USERNAME	PASSWORD	EMAIL
1	3568714689264256	johnsmith	johnsmith123	johnsmith123@gmail.com
2	3568714682764257	richardoliver	richardoliver13	richardoliver13@gmail.com
3	3568714682764258	davidshephard	davidshephard65	davidshephard65@gmail.com
4	3568714682764259	meridithgrey	meridithgrey3	meridithgrey3@gmail.com
5	3568714682764260	davidbenjamin	davidbenjamin56	davidbenjamin56@gmail.com

Above is the query result of customer user account which contain customeruseraccountid, username, password and email.

Insert Statement for Customer Table

Below is the insert statement for customer table.

```
/* INSERTING INTO CUSTOMER */
```

```
INSERT INTO CUSTOMER
VALUES (DEFAULT,'JOHN','SMITH','24-APR-1990','3456 CONEY ISLAND
AVE','3546 CONEY ISLAND AVE',
'BROOKLYN','NY','11234','UNITED STATES','987-897-
0986','johnsmith23@gmail.com','987-234-454',
'24-APR-2023','3568714689264256','R');
```

```
INSERT INTO CUSTOMER
VALUES (DEFAULT,'RICHARD','OLIVER','29-JAN-1990','345 OCEAN AVE','546
OCEAN AVE',
'BROOKLYN','NY','11233','UNITED STATES','718-837-
0186','richardoliver13@gmail.com','900-254-514',
'29-JAN-2021','3568714682764257','C');
```

```
INSERT INTO CUSTOMER
VALUES (DEFAULT,'DAVID','SHEPHARD','29-MAY-1993','500 NEPTUNE
AVE','510 NEPTUNE AVE',
'BROOKLYN','NY','11235','UNITED STATES','918-637-
0446','davidshephard65@gmail.com','934-274-500',
'29-MAY-2025','3568714682764258','C');
```

```
INSERT INTO CUSTOMER
VALUES (DEFAULT,'MERIDITH','GREY','13-DEC-1989','520 BRIGHTON BEACH
AVE','525 BRIGHTON BEACH AVE',
'BROOKLYN','NY','11235','UNITED STATES','918-637-
0236','meridithgrey3@gmail.com','914-474-520',
'13-DEC-2023','3568714682764259','R');
```

```
INSERT INTO CUSTOMER
VALUES (DEFAULT,'DAVID','BENJAMIN','23-MAR-1999','230 GARRISON
STREET','130 GARRISON STREET',
'BROOKLYN','NY','11235','UNITED STATES','929-137-
5236','davidbenjamin56@gmail.com','944-574-510',
'19-MAR-2023','3568714682764260','C');
```

Result of Customer Table

The screenshot shows a database query results window with two tabs: 'Script Output' and 'Query Result'. The 'Query Result' tab is active, displaying a table with 5 rows of data. The columns are labeled: CUSTOMERID, FIRSTNAME, LASTNAME, BIRTHDATE, ADDRESSLINE1, ADDRESSLINE2, CITY, STATECODE, ZIPCODE, and COUNTRY. The data is as follows:

CUSTOMERID	FIRSTNAME	LASTNAME	BIRTHDATE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY
1	21 JOHN	SMITH	24-APR-90	3456 CONEY ISLAND AVE	3546 CONEY ISLAND AVE	BROOKLYN NY	11234	UNITED STATES	98
2	22 RICHARD	OLIVER	29-JAN-90	345 OCEAN AVE	546 OCEAN AVE	BROOKLYN NY	11233	UNITED STATES	71
3	23 DAVID	SHEPHARD	29-MAY-93	500 NEPTUNE AVE	510 NEPTUNE AVE	BROOKLYN NY	11235	UNITED STATES	91
4	24 MERIDITH	GREY	13-DEC-89	520 BRIGHTON BEACH AVE	525 BRIGHTON BEACH AVE	BROOKLYN NY	11235	UNITED STATES	91
5	26 DAVID	BENJAMIN	23-MAR-99	230 GARRISON STREET	130 GARRISON STREET	BROOKLYN NY	11235	UNITED STATES	92

Customer table contains the id, name and address of the customer.

Insert Statement for Credit Card

Below is the insert statement for credit card.

```
/* INSERT INTO CREDIT CARD TABLEM */
```

```
INSERT INTO CREDITCARD
```

```
VALUES ('0674851697523760','JOHN SMITH','CITY BANK','13-APR-2023','3456 CONEY  
ISLAND AVE',  
'3546 CONEY ISLAND AVE','BROOKLYN','NY','11234','UNITED  
STATES',200,2000,'Y');
```

```
INSERT INTO CREDITCARD
```

```
VALUES ('0674851697523762','RICHARD OLIVER','CHASE BANK','23-MAY-2025','345  
OCEAN AVE',  
'546 OCEAN AVE','BROOKLYN','NY','11233','UNITED STATES',280,1000,'Y');
```

```
INSERT INTO CREDITCARD
```

```
VALUES ('0674851697523764','DAVID SHEPHARD','DISCOVER','9-JUN-2022','500  
NEPTUNE AVE',  
'510 NEPTUNE AVE','BROOKLYN','NY','11235','UNITED STATES',980,3000,'Y');
```

```
INSERT INTO CREDITCARD
```

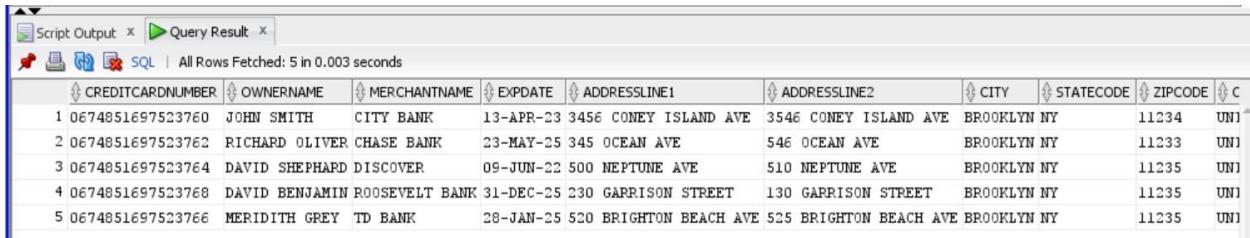
```
VALUES ('0674851697523766','MERIDITH GREY','TD BANK','28-JAN-2025','520  
BRIGHTON BEACH AVE',  
'525 BRIGHTON BEACH AVE','BROOKLYN','NY','11235','UNITED  
STATES',900,900,'N');
```

```
INSERT INTO CREDITCARD
```

```
VALUES ('0674851697523768','DAVID BENJAMIN','ROOSEVELT BANK','31-  
DEC2025','230 GARRISON STREET',  
'130 GARRISON STREET','BROOKLYN','NY','11235','UNITED
```

STATES',680,1000,'Y');

Result of Credit Card Table



The screenshot shows a SQL query result window with two tabs: 'Script Output' and 'Query Result'. The 'Query Result' tab is active, displaying the following data:

CREDITCARDNUMBER	OWNERNAME	MERCHANTNAME	EXPPDATE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	C
1 0674851697523760	JOHN SMITH	CITY BANK	13-APR-23	3456 CONEY ISLAND AVE	3546 CONEY ISLAND AVE	BROOKLYN NY	11234	UNI	
2 0674851697523762	RICHARD OLIVER	CHASE BANK	23-MAY-25	345 OCEAN AVE	546 OCEAN AVE	BROOKLYN NY	11233	UNI	
3 0674851697523764	DAVID SHEPARD	DISCOVER	09-JUN-22	500 NEPTUNE AVE	510 NEPTUNE AVE	BROOKLYN NY	11235	UNI	
4 0674851697523768	DAVID BENJAMIN	ROOSEVELT BANK	31-DEC-25	230 GARRISON STREET	130 GARRISON STREET	BROOKLYN NY	11235	UNI	
5 0674851697523766	MERIDITH GREY	TD BANK	28-JAN-25	520 BRIGHTON BEACH AVE	525 BRIGHTON BEACH AVE	BROOKLYN NY	11235	UNI	

Above is the query result of credit card.

Insert Statement for Customer Credit Card Table

Below is the associative entity of customer and credit card table.

```
/* INSERT INTO CUSTOMERCREDITCARD */
/* ASSOCIATIVE ENTITY */
```

```
INSERT INTO CUSTOMERCREDITCARD
VALUES ('0674851697523760','21');
```

```
INSERT INTO CUSTOMERCREDITCARD
VALUES ('0674851697523762','22');
```

```
INSERT INTO CUSTOMERCREDITCARD
VALUES ('0674851697523764','23');
```

```
INSERT INTO CUSTOMERCREDITCARD
VALUES ('0674851697523766','24');
```

```
INSERT INTO CUSTOMERCREDITCARD
VALUES ('0674851697523768','26');
```

Result of Customer Credit Card Table

	CREDITCARDNUMBER	CUSTOMERID
1	0674851697523760	21
2	0674851697523762	22
3	0674851697523764	23
4	0674851697523766	24
5	0674851697523768	26

Customer credit card table contains the 2 foreign keys from customer and credit card table, and it is an associative entity of customer and credit card table.

Insert Statement for Vehicle Status

Vehicle status is a foreign key in vehicle table. So it should be created before vehicle table.

```
/* INSERTING INTO VEHICLESTATUS TABLE */
```

```
INSERT INTO VEHICLESTATUS
VALUES (1,'RESERVED');
```

```
INSERT INTO VEHICLESTATUS
VALUES (2,'RENTED');
```

```
INSERT INTO VEHICLESTATUS
VALUES (3,'AVAILABLE');
```

```
INSERT INTO VEHICLESTATUS
VALUES (4,'NOT AVAILABLE');
```

```
INSERT INTO VEHICLESTATUS
VALUES (5,'MAINTENANCE');
```

```
INSERT INTO VEHICLESTATUS
VALUES (6, 'TRANSFERRED TO ANOTHER AGENCY');
```

Result of Vehicle Statement

The screenshot shows the MySQL Workbench interface with the 'Query Result' tab active. The results show six rows of data from the VEHICLESTATUS table:

VEHICLESTATUSID	VEHICLESTATUSDESC
1	1 RESERVED
2	2 RENTED
3	3 AVAILABLE
4	4 NOT AVAILABLE
5	5 MAINTENANCE
6	6 TRANSFERRED TO ANOTHER AGENCY

Above is the query result for vehicle table.

Insert Statement for Vehicle Rental Category

Below is the insert statement for vehicle rental category which is a foreign in vehicle table.

```
/* INSERT INTO VEHILCERENTALCATEGORY TABLE */
```

```
INSERT INTO VEHICLERENTALCATEGORY
VALUES (1,'CAR-ECONOMIC',133.56);
```

```
INSERT INTO VEHICLERENTALCATEGORY
VALUES (2,'CAR-INTER',133.56);
```

```
INSERT INTO VEHICLERENTALCATEGORY
VALUES (3,'CAR-PREMIUM',168.56);
```

```
INSERT INTO VEHICLERENTALCATEGORY
VALUES (4,'VAN-PASSENGER (12)',79.56);
```

```
INSERT INTO VEHICLERENTALCATEGORY
VALUES (5,'MOTORCYCLE-SCOOTER',161.56);
```

Result of Vehicle Rental Category

Below is the query result of vehicle rental category.

The screenshot shows a database query results window with two tabs: 'Script Output' and 'Query Result'. The 'Query Result' tab is active, displaying the following table:

VEHICLERENTALCATEGORYID	CATEGORYNAME	CATEGORYDAILYRENTALRATE
1	1 CAR-ECONOMIC	133.56
2	2 CAR-INTER	133.56
3	3 CAR-PREMIUM	168.56
4	5 MOTORCYCLE-SCOOTER	161.56
5	4 VAN-PASSENGER (12)	79.56

Insert Statement for Rental Agency Table

Rental agency table is a foreign key in vehicle table which came in twice, once as an vehicleowningagencyid and second time as an vehiclecurrentlocationagencyid.

```
/*INSERT INTO RENTALAGENCY */
```

```
INSERT INTO RENTALAGENCY
VALUES (98765456,'EZRENTAL INC','342 FLATBUSH AVE','457 FLATBUSH AVE',
'BROOKLYN', 'NY', '11234',
'UNITED STATES', '980-546-9898', 'ezrental@gmail.com');
```

```
INSERT INTO RENTALAGENCY
VALUES (98765457,'QUICKRENTAL INC','42 ROCKAWAY AVE','4557 ROCKAWAY
AVE', 'BROOKLYN', 'NY', '11244',
'UNITED STATES', '980-506-9814', 'quickrental@gmail.com');
```

```
INSERT INTO RENTALAGENCY
VALUES (98765458,'CARONROAD INC','24 FOSTER AVE','57 FOSTER AVE',
'BROOKLYN', 'NY', '11244',
'UNITED STATES', '718-906-0014', 'caronroad@gmail.com');
```

```
INSERT INTO RENTALAGENCY
VALUES (98765459,'RUSHCAR INC','78 AVE U','57 AVE U', 'BROOKLYN', 'NY', '11244',
'UNITED STATES', '718-069-0001', 'rushcar@gmail.com');
```

```
INSERT INTO RENTALAGENCY
VALUES (98765460,'CARATUNION INC','78 UNION SQUARE','79 UNION SQUARE',
'MANHATTAN', 'NY', '11287',
```

'UNITED STATES', '718-464-2101', 'caratunion@gmail.com');

Result of Rental Agency

Below is the result of rental agency.

	RENTALAGENCYID	RENTALAGENCYNAME	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY	PHONE	EMAIL
1	98765456	EZRENTAL INC	342 FLATBUSH AVE	457 FLATBUSH AVE	BROOKLYN	NY	11234	UNITED STATES	980-546-9898	ezrental@gmail.com
2	98765457	QUICKRENTAL INC	42 ROCKAWAY AVE	4557 ROCKAWAY AVE	BROOKLYN	NY	11244	UNITED STATES	980-506-9814	quickrental@gmail.com
3	98765458	CARONROAD INC	24 FOSTER AVE	57 FOSTER AVE	BROOKLYN	NY	11244	UNITED STATES	718-906-0014	caronroad@gmail.com
4	98765459	RUSHCAR INC	78 AVE U	57 AVE U	BROOKLYN	NY	11244	UNITED STATES	718-069-0001	rushcar@gmail.com
5	98765460	CARATUNION INC	78 UNION SQUARE	79 UNION SQUARE	MANHATTAN	NY	11287	UNITED STATES	718-464-2101	caratunion@gmail.com

Insert Statement for Vehicle (Supertype) Table

Below is the sql script of vehicle table which contains 4 foreign keys and, it is a parent table of 4 subtype table. It contains 20 rows 4 set of rows for each type of vehicle.

```
/* INSERT INTO VEHICLE TABLE */
```

```
INSERT INTO VEHICLE
```

```
VALUES (DEFAULT, 'JN3MS37A9PW202929', 'TOYOTA', 'RAV4', '2017', 'BLACK', 'JUY-212', 'NY',  
21234, '8-speed automatic', '5', 1, 1, 98765456, 98765456, 'S');
```

```
INSERT INTO VEHICLE
```

```
VALUES (DEFAULT, '5YJ3E1EAXHF000316', 'TESLA', 'MODEL S', '2019', 'WHITE', 'KDY-234',  
'NY',  
9000, '7-speed automatic', '5', 1, 1, 98765456, 98765456, 'C');
```

```
INSERT INTO VEHICLE
```

```
VALUES (DEFAULT, '8YJ323EAXHFF00318', 'INFINNTI', 'Q5', '2021', 'BLACK', 'XYL-314', 'NY',  
19000, '7-speed automatic', '5', 1, 1, 98765458, 98765458, 'M');
```

```
INSERT INTO VEHICLE
```

```
VALUES (DEFAULT, '0PJ123EAXHFF00419', 'TOYOTA', '2021 Toyota Sienna', '2021', 'SILVER',  
'ILL-514', 'NY',  
19700, 'Continuous variable', '6', 1, 3, 98765456, 98765456, 'M');
```

```
INSERT INTO VEHICLE
```

```
VALUES (DEFAULT, '2IN187EAXHFF00890', 'DODGE', 'Dodge Grand Caravan', '2020', 'BLACK',  
'OLI-553', 'NY',  
122700, '6-speed automatic', '8', 1, 4, 98765460, 98765456, 'M');
```

INSERT INTO VEHICLE
VALUES (DEFAULT, '8LP187OPMHFF00999', 'CHRYSLER', 'Chrysler Pacifica Hybrid', '2021',
'BLACK', 'GHL-003', 'NY',
85872, 'Continuous variable', '8', 1, 2, 98765458, 98765456, 'M');

INSERT INTO VEHICLE
VALUES (DEFAULT, '9HJ003EAXTVF00100', 'FORD', 'BRONCO', '2021', 'SILVER', 'DBL-204', 'NY',
19004, 'Sport utility vehicle', '5', 3, 2, 98765458, 98765459, 'C');

INSERT INTO VEHICLE
VALUES (DEFAULT, '10J013EAXTVF11100', 'MERCEDES', '2021 Mercedes E-Class', '2021',
'SILVER',
'CBL-104', 'NY', 2004, 'LUXURY CAR', '5', 3, 2, 98765460, 98765460, 'C');

INSERT INTO VEHICLE
VALUES (DEFAULT, '9G3113ETXTHF22100', 'FORD', '2020 Ford Transit-150', '2020', 'WHITE',
'VIL-114', 'NY', 20004, 'ECONOMIC', '10', 3, 2, 98765460, 98765460, 'M');

INSERT INTO VEHICLE
VALUES (DEFAULT, '9OP113EI8THF22189', 'NISSAN', '2021 Nissan Rogue', '2021', 'RED', 'JUY-
908',
'NY', 34532, 'sport utility vehicle', '5', 3, 3, 98765460, 98765460, 'S');

INSERT INTO VEHICLE
VALUES (DEFAULT, '20P115TI0THF29891', 'GMC', '2020 GMC Savana', '2020', 'WHITE', 'SDE-108',
'NY', 2032, '8-speed automatic', '12', 5, 2, 98765458, 98765458, 'V');

INSERT INTO VEHICLE
VALUES (DEFAULT, '10P115TI0THF29990', 'RAM', '2020 RAM ProMaster City', '2020', 'BLACK',
'JUI-008',
'NY', 2032, '9-speed automatic', '12', 5, 2, 98765456, 98765456, 'V');

INSERT INTO VEHICLE
VALUES (DEFAULT, '9YP550I0THF90765', 'FORD', 'Ford Transit-200', '2020', 'BLACK', 'OIR-096',
'NY', 10132, '10-speed automatic', '12', 5, 1, 98765456, 98765460, 'V');

INSERT INTO VEHICLE
VALUES (DEFAULT, '1Y515OI0TPF26570', 'NISSAN', '2020 Nissan NV200', '2020', 'BLACK', 'HTR-
896',
'NY', 20132, 'Continuous variable', '12', 5, 4, 98765456, 98765456, 'V');

INSERT INTO VEHICLE

```
VALUES (DEFAULT, '7YK5190I0HHF28740', 'MERCEDES', 'Mercedes-Benz Metris', '2020', 'BLACK',
'PUR-896',
'NY', 20132, '7-speed automatic', '12', 5, 2, 98765457, 98765456, 'V');
```

INSERT INTO VEHICLE

```
VALUES (DEFAULT, '0YK5190I0IOD28702', 'DODGE', 'Dodge Durango', '2021', 'BLACK', 'IUN-806',
'NY', 211102, '7-speed automatic', '5', 1, 2, 98765457, 98765457, 'C');
```

INSERT INTO VEHICLE

```
VALUES (DEFAULT, '0YK5190I0IOD28701', 'HYUNDAI', 'Hyundai Tucson', '2021', 'RED', 'MNM-
006',
'NY', 211102, '6-speed automatic', '5', 1, 5, 98765460, 98765456, 'C');
```

INSERT INTO VEHICLE

```
VALUES (DEFAULT, '1J5190I0I4Y28678', 'HONDA', 'Honda Pilot', '2021', 'BLACK', 'MOI-106',
'NY', 211132, 'Sport utility vehicle', '5', 1, 3, 98765460, 98765458, 'S');
```

INSERT INTO VEHICLE

```
VALUES (DEFAULT, '1IY5190I0I0P27502', 'MAZDA', 'Mazda CX-30', '2021', 'BLACK', 'PLH-246',
'NY', 21115, 'sport utility vehicle', '5', 1, 3, 98765460, 98765458, 'S');
```

INSERT INTO VEHICLE

```
VALUES (DEFAULT, '7YK5190I0IOD28231', 'SUBARU', 'Subaru Crosstrek', '2021', 'BLUE', 'IMM-
486',
'NY', 2112, 'sport utility vehicle', '5', 1, 5, 98765456, 98765456, 'S');
```

Result of Vehicle Table

Below is the result of few vehicle rows.

The screenshot shows a SQL developer interface with multiple tabs at the top: Query Result, Query Result 3, Query Result 4, Script Output, and Query Result 5. The Query Result tab is active, displaying the following table:

VEHICLEID	VINNUMBER	MAKE	MODEL	YEAR	COLOR	LICENSEPLATENUMBER	LICENSEPLATESTATE	MILEAGE	TRANSMISSIONTYPE
1	6110P11ST10THF29990	RAM	2020 RAM ProMaster City	2020	BLACK	JUI-008	NY	2032	9-speed automatic
2	647YK519010HHF28740	MERCEDES	Mercedes-Benz Metris	2020	BLACK	PUR-896	NY	20132	7-speed automatic
3	E71Y515010TPF26570	NISSAN	2020 Nissan NV200	2020	BLACK	HTR-896	NY	20132	Continuous variable
4	689YP55010THF90765	FORD	Ford Transit-150	2020	BLACK	OIR-096	NY	10132	10-speed automatic
5	7020P11ST10THF29891	GMC	2020 GMC Savana	2020	WHITE	SDE-108	NY	2032	8-speed automatic
6	710PJ123EAХHFF00419	TOYOTA	2021 Toyota Sienna	2021	SILVER	ILL-514	NY	19700	Continuous variable
7	722IM187EAХHFF00890	DODGE	Dodge Grand Caravan	2020	BLACK	OLI-553	NY	122700	6-speed automatic
8	738LP1870PMHFF00999	CHRYSLER	Chrysler Pacifica Hy...	2021	BLACK	GHL-003	NY	85872	Continuous variable
9	740YK519010I0D28701	HYUNDAI	Hyundai Tucson	2021	RED	MNM-006	NY	211102	6-speed automatic
10	780YK519010I0D28702	DODGE	Dodge Durango	2021	BLACK	IUN-806	NY	211102	7-speed automatic
11	791IY519010IOP27502	MAZDA	Mazda CX-30	2021	BLACK	PLH-246	NY	21115	sport utility vehicle
12	811J519010I4Y28678	HONDA	Honda Pilot	2021	BLACK	MOI-106	NY	211132	Sport utility vehicle
13	827YK519010I0D28231	SUBARU	Subaru Crosstrek	2021	BLUE	IMM-486	NY	2112	sport utility vehicle
14	21JN3MS37A9PW202929	TOYOTA	RAV4	2017	BLACK	JUY-212	NY	21234	8-speed automatic
15	249HJ003EAХTVF00100	FORD	BRONCO	2021	SILVER	DBL-204	NY	19004	Sport utility vehicle
16	2610J013EAХTVF11100	MERCEDES	2021 Mercedes E-Class	2021	SILVER	CBL-104	NY	2004	LUXURY CAR

Insert Statement for Car (Subtype Table)

Below is the insert statement for car table which is a child table of vehicle table.

```
/* INSERT INTO CAR TABLE */
/* INSERT INTO VEHICLE CHILD TABLE */
```

```
INSERT INTO CAR
VALUES (24, 82.97);
```

```
INSERT INTO CAR
VALUES (26, 64);
```

```
INSERT INTO CAR
VALUES (74, 78.2);
```

```
INSERT INTO CAR
VALUES (78, 98.2);
```

```
INSERT INTO CAR
VALUES (28, 60.2);
```

Result of Car Table

Below is the query result of car table.

The screenshot shows a MySQL Workbench interface with two tabs open: 'Query Result' and 'Query Result 3'. The 'Query Result' tab displays a table with columns 'VEHICLEID' and 'TRUNKCAPACITY' containing the following data:

VEHICLEID	TRUNKCAPACITY
1	74
2	78
3	24
4	26
5	28

The 'Query Result 3' tab displays a table with columns 'VEHICLEID' and 'TRUCKCAPACITY' containing the following data:

VEHICLEID	TRUCKCAPACITY
1	78.2
2	98.2
3	82.97
4	64
5	60.2

Insert Statement for SUV (Subtype Table)

Below is the query of SUV table which is a child table of vehicle.

```
/* INSERT INTO SUV TABLE */
/* SUV IS CHILD TABLE OF VEHICLE */
```

```
INSERT INTO SUV
VALUES (21, 1200,'4');
```

```
INSERT INTO SUV
VALUES (41, 1350, '4');
```

```
INSERT INTO SUV
VALUES (79, 1450, '4');
```

```
INSERT INTO SUV
VALUES (81, 1110, '4');
```

```
INSERT INTO SUV
VALUES (82, 1980, '4');
```

```
/* SELECT SUV DATA */
SELECT * FROM SUV;
```

Result of SUV Table

Following is the query result of SUV table.

VEHICLEID	TOWINGCAPACITY	ISAWD
1	79	1450 4
2	81	1110 4
3	82	1980 4
4	21	1200 4
5	41	1350 4

Insert into

Minivan Table

Below is the query of minivan which is a child table of vehicle.

```
/* INSERT INTO MINIVAN TABLE */
/* VAN IS VEHICLE CHILD TABLE */
```

```
INSERT INTO MINIVAN
VALUES (29, 'Y');
```

```
INSERT INTO MINIVAN
VALUES (30, 'N');
```

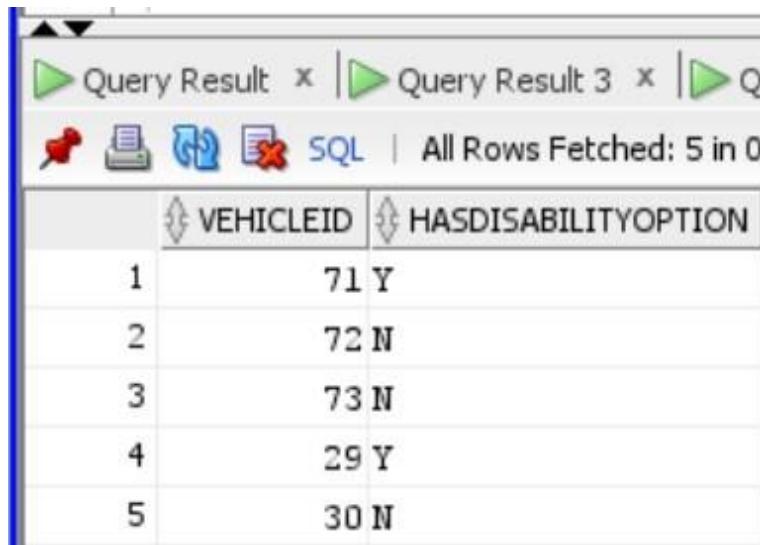
```
INSERT INTO MINIVAN
VALUES (71, 'Y');
```

```
INSERT INTO MINIVAN
VALUES (72, 'N');
```

```
INSERT INTO MINIVAN
VALUES (73, 'N');
```

Result of Minivan Table

Below is the result of minivan table.



The screenshot shows a MySQL Workbench interface with three tabs: 'Query Result', 'Query Result 3', and 'Query Result 4'. The 'Query Result' tab is active, displaying the results of a query. The results are presented in a table with two columns: 'VEHICLEID' and 'HASDISABILITYOPTION'. The data is as follows:

VEHICLEID	HASDISABILITYOPTION
1	71 Y
2	72 N
3	73 N
4	29 Y
5	30 N

Insert Statement of Cargovan (Subtype Table)

Below is the query of cargovan which is child table of vehicle table.

```
/*INSERT INTO CARGOVAN */  
/* CHILD OF VEHICLE TABLE */
```

```
INSERT INTO CARGOVAN  
VALUES(61,12,4100);
```

```
INSERT INTO CARGOVAN  
VALUES(64,12,4000);
```

```
INSERT INTO CARGOVAN  
VALUES(67,12,3600);
```

```
INSERT INTO CARGOVAN  
VALUES(68,12,4600);
```

```
INSERT INTO CARGOVAN  
VALUES(70,12,3227);
```

Result of Cargovan Table

Below is the query result of cargovan.

The screenshot shows a MySQL Workbench interface with three tabs: 'Query Result', 'Query Result 3', and 'Query Result 4'. The 'Query Result' tab is active, displaying the results of a query. The results are presented in a table with three columns: VEHICLEID, CARGOCAPACITY, and MAXIMUMPAYLOAD. The data is as follows:

VEHICLEID	CARGOCAPACITY	MAXIMUMPAYLOAD
1	61	12
2	64	12
3	67	12
4	68	12
5	70	12

Select Statement for Specific Row

Below is the query to select a specific row of vehicle table.

```
/* SELECTING SPECIFIC ROW FROM VEHICLE */
```

```
SELECT *
FROM VEHICLE
WHERE VEHICLEID = 71;
```

Result of Select Statement for Specific Row

Below is the result of select statement for a specific row.

The screenshot shows a MySQL Workbench interface with three tabs: 'Query Result', 'Query Result 3', and 'Query Result 4'. The 'Query Result' tab is active, displaying the results of a query. The results are presented in a table with ten columns: VEHICLEID, VINNUMBER, MAKE, MODEL, YEAR, COLOR, LICENSEPLATENUMBER, LICENSEPLATESTATE, MILEAGE, and TRANSMISSIONTYPE. The data is as follows:

VEHICLEID	VINNUMBER	MAKE	MODEL	YEAR	COLOR	LICENSEPLATENUMBER	LICENSEPLATESTATE	MILEAGE	TRANSMISSIONTYPE
1	71OPJ123EAХHFF00419	TOYOTA	2021 Toyota Sienna	2021	SILVER	ILL-514	NY	19700	Continuous variable 6

Select 2 Parent and 1 Associative Entity

Below is the select query of parent table credit card and customer and associative entity table.

```
/* SELECTING 2 PARENT AND ASSOCIATIVE ROW */
```

```

SELECT CC.MERCHANTNAME, CCC.CUSTOMERID, CCC.CREDITCARDNUMBER,
CM.FIRSTNAME, CM.LASTNAME
FROM ((CUSTOMERCREDITCARD CCC
INNER JOIN CREDITCARD CC
ON CC.CREDITCARDNUMBER = CCC.CREDITCARDNUMBER)
INNER JOIN CUSTOMER CM
ON CM.CUSTOMERID = CCC.CUSTOMERID);

```

Result of Select 2 Parent and 1 Associative Entity

Below is the query result of 2 parent table and associative entity

MERCHANTNAME	CUSTOMERID	CREDITCARDNUMBER	FIRSTNAME	LASTNAME
1 CITY BANK		21 0674851697523760	JOHN	SMITH
2 CHASE BANK		22 0674851697523762	RICHARD	OLIVER
3 DISCOVER		23 0674851697523764	DAVID	SHEPHARD
4 ROOSEVELT BANK		26 0674851697523768	DAVID	BENJAMIN
5 TD BANK		24 0674851697523766	MERIDITH	GREY

Update Statement for Table

Below is the query to update vehicle rental category.

```
/* UPDATING ALL COLUMNS EXCEPT A PRIMARY KEY */
```

```

UPDATE VEHICLERENTALCATEGORY
SET
    CATEGORYNAME = 'SPORT-RACE',
    CATEGORYDAILYRENTALRATE = 180.89
WHERE VEHICLERENTALCATEGORYID = 3;

```

Result of Update Statement for Table

Below is the result of update table which says that table has been updated.

A screenshot of a database management system interface. At the top, there are tabs for "Query Result" and "Script Output". Below the tabs, there are several icons. To the right of the icons, the text "Task completed in 0.033 seconds" is displayed. In the main area, the message "1 row updated." is shown.

UPDATE ASSOCIATIVE ENTITY

Below is the query to update a table which has 3 columns and all of them are foreign keys, but DISCOUNT table and EZPLUS table are optional.

```
/* UPDATING A ASSOCIATIVE ENTITY */
```

```
UPDATE RETAILCUSTOMER  
SET DISCOUNTID = 1,  
    EZPLUSID = 3  
WHERE CUSTOMERID = 26;
```

Result of Update Associative Entity

Below is the query result of updating associative entity.

A screenshot of a database management system interface. The main window shows a query editor with the following content:

```
/* UPDATING A ASSOCIATIVE ENTITY */  
UPDATE RETAILCUSTOMER  
SET DISCOUNTID = 1,  
    EZPLUSID = 3  
WHERE CUSTOMERID = 26;
```

Below the query editor, there are tabs for "Query Result", "Query Result 1", "Query Result 2", "Script Output", and "Query Result 3". The "Script Output" tab is active. To the right of the tabs, the text "Task completed in 0.024 seconds" is displayed. In the main area, the message "1 row updated." is shown.

Delete Statement for Table

Below is the query to delete a row from a table

```
/* DELETE ONE ROW FROM VEHICLERENTALCATEGORY */
```

```
DELETE VEHICLERENTALCATEGORY
```

```
WHERE VEHICLERENTALCATEGORYID = 2;
```

Result of Delete Statement for Table

Below is the result of delete row from table which says that row has been deleted.

```
1 row deleted.
```

Delete Statement for Associative Entity Table

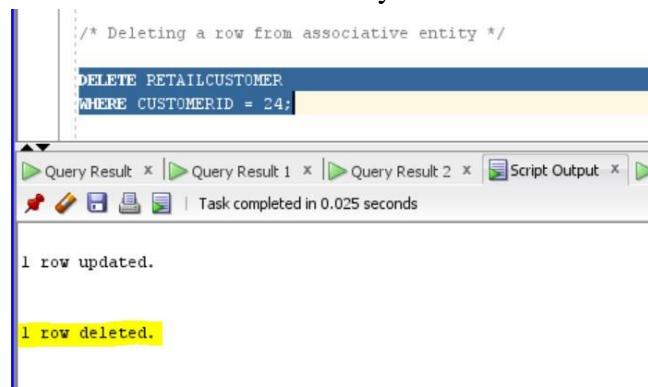
Below is the delete statement for associative entity table which deleted a row whose customerid is 24.

```
/* Deleting a row from associative entity */
```

```
DELETE RETAILCUSTOMER  
WHERE CUSTOMERID = 24;
```

Result of Delete Statement for Associative Entity Table

Below is the delete statement for an associative entity.



The screenshot shows a database interface with a query editor and several tabs at the bottom. The query editor contains the following code:

```
/* Deleting a row from associative entity */  
  
DELETE RETAILCUSTOMER  
WHERE CUSTOMERID = 24;
```

The tabs at the bottom are labeled: Query Result x, Query Result 1 x, Query Result 2 x, Script Output x. Below the tabs, it says "Task completed in 0.025 seconds".

The output pane shows the results of the query:

```
1 row updated.  
  
1 row deleted.
```

Conclusion

This document contains every aspect of the database that has been implemented. Starting from business requirements to creating and testing data in the database. During the development phase of the EZRental database every aspect of the database is well documented by taking screenshots of the queries and coping queries on this document. In the end, this document contains information about EZRental in a chronological order which explain every part of the database very deeply.

PROJECT 2 – EZRental

POS Operational &

strategic Decision

Making via Business

Reports

Upgrade Project Objectives

Business Reports Queries & Summary

Report #1

GUIDELINES	ANSWER
Q1 – What is the Business Scenario & Objectives?	A campaign to target customers whose birthday is today, and send them a message wishing Happy Birthday with promoting EZRENTAL Inc Brand.
Q2 – Target Persona/Decision Maker?	Marketing Manager & team
Q3 – What is the Decision to be made?	To send an email to customers whose Birthday is today, and Marketing Team will send them an email wishing Happy Birthday.
Q4 – What Data is required to make the Decision?	Customer first and last name, Phone number, Customer Birthdate, Discount Code and Description of Discount Code.
Q5 – Identify tables that contain the Data identified in Step 4?	Customer: Customer first and last name, phone, Birthdate. Retail Customer Discount: DISCOUNTCODE, DISCOUNTCODEDESC

SQL Query for Report #1

```
SELECT C.FIRSTNAME, C.LASTNAME, C.PHONE, C.EMAIL, C.BIRTHDATE, DISCOUNTCODE,  
DISCOUNTCODEDESC  
  
FROM CUSTOMER C, RETAILCUSTOMER R, DISCOUNT D  
  
WHERE C.CUSTOMERID = R.CUSTOMERID AND  
  
R.DISCOUNTID = D.DISCOUNTID AND  
  
EXTRACT(MONTH FROM SYSDATE) = EXTRACT(MONTH FROM BIRTHDATE) AND  
EXTRACT(DAY FROM SYSDATE) = EXTRACT(DAY FROM BIRTHDATE);
```

Screen Shot of Report #1

```


/*
Get customer first and last name, phone, email, birthdate, cutsmertype, discount code, discount
description and check if their birthday is today and offer them their discount.
*/

SELECT C.FIRSTNAME, C.LASTNAME, C.PHONE, C.EMAIL, C.BIRTHDATE, DISCOUNTCODE, DISCOUNTCODEDESC
FROM CUSTOMER C, RETAILCUSTOMER R, DISCOUNT D
WHERE C.CUSTOMERID = R.CUSTOMERID AND
R.DISCOUNTID = D.DISCOUNTID AND
EXTRACT(MONTH FROM SYSDATE) = EXTRACT(MONTH FROM BIRTHDATE) AND
EXTRACT(DAY FROM SYSDATE) = EXTRACT(DAY FROM BIRTHDATE);


```

Query Result x

SQL | All Rows Fetched: 1 in 0.06 seconds

	FIRSTNAME	LASTNAME	PHONE	EMAIL	BIRTHDATE	DISCOUNTCODE	DISCOUNTCODEDESC
1	DAVID	SHEPHARD	918-637-0446	davidshephard65@gmail.com	04-MAY-00	KI65REFR47	LINCOLN DAY DISCOUNT

Report #2

GUIDELINES	ANSWER
Q1 – What is the Business Scenario & Objectives?	A query to check if customer card is active and if it's active then check it shouldn't have balance more than \$900.
Q2 – Target Persona/Decision Maker?	Marketing, Finance, Accounting, HR etc.
Q3 – What is the Decision to be made?	To let Finance, Marketing and Accounting team know which customer credit card is not working and has balance more than \$900.
Q4 – What Data is required to make the Decision?	Customer first and last name, Phone number, Customer Email, Driver License Number, Credit Card Number, Credit Card Balance and Credit Card Activation Status
Q5 – Identify tables that contain the Data identified in Step 4?	Customer: Customer first and last name, phone, email, Driver license number Customer Credit Card Credit Card: CREDIT CARD NUMBER, CREDIT CARD BALANCE, ACTIVATION STATUS

SQL Query for Report #2

```

SELECT C.FIRSTNAME, C.LASTNAME, C.PHONE, C.EMAIL, C.DRIVERLICENSENUMBER,
CR.CREDITCARDNUMBER, CR.CREDITCARDBALANCE, CR.ACTIVATIONSTATUS
FROM CUSTOMER C
JOIN CUSTOMERCREDITCARD CCD
ON C.CUSTOMERID = CCD.CUSTOMERID

```

```

JOIN CREDITCARD CR
ON CR.CREDITCARDNUMBER = CCD.CREDITCARDNUMBER
WHERE CR.CREDITCARDBALANCE >= 900 OR
CR.ACTIVATIONSTATUS = 'N';

```

Screen Shot of Report #2

```

Report #2
Get customer first and last name, phone, email, Driver license #, credit card # to check if customer
card is active or if balance is more than $900 then customer should pay his balance to
rent car in future.

/*
SELECT C.FIRSTNAME, C.LASTNAME, C.PHONE, C.EMAIL, C.DRIVERLICENSENUMBER,
       CR.CREDITCARDNUMBER, CR.CREDITCARDBALANCE, CR.ACTIVATIONSTATUS
FROM CUSTOMER C
JOIN CUSTOMERCREDITCARD CCD
ON C.CUSTOMERID = CCD.CUSTOMERID
JOIN CREDITCARD CR
ON CR.CREDITCARDNUMBER = CCD.CREDITCARDNUMBER
WHERE CR.CREDITCARDBALANCE >= 900 OR
CR.ACTIVATIONSTATUS = 'N';

```

Query Result

All Rows Fetched: 2 in 0.006 seconds

	FIRSTNAME	LASTNAME	PHONE	EMAIL	DRIVERLICENSENUMBER	CREDITCARDNUMBER	CREDITCARDBALANCE	ACTIVATIONSTATUS
1	DAVID	SHEPHARD	918-637-0446	davidshephard65@gmail.com	934-274-500	0674851697523764	980	Y
2	MERIDITH	GREY	918-637-0236	meridithgrey3@gmail.com	914-474-520	0674851697523766	900	N

Report #3

GUIDELINES

ANSWER

Q1 – What is the Business Scenario & Objectives?	To get information about customer first and last name, phone, email, EZPLUSREWARDCODE, EZPLUSREWARDSEARNEDPOINTS to check customer has more than 2000 points if so, offer better rates than regular customers.
Q2 – Target Persona/Decision Maker?	Marketing, Finance, Accounting, HR etc.
Q3 – What is the Decision to be made?	To get customer names who have more than 2000 points and give them better rates.
Q4 – What Data is required to make the Decision?	Customer first and last name, Phone number, Customer Email, EZPLUSREWARDCODE, EZPLUSREWARDSEARNEDPOINTS
Q5 – Identify tables that contain the Data identified in Step 4?	Customer: Customer first and last name, phone, email, Driver license number Retail Customer EZPLUS: EZPLUSREWARDCODE, EZPLUSREWARDSEARNEDPOINTS

SQL Query Report #3

```
SELECT C.FIRSTNAME, C.LASTNAME, C.PHONE, C.EMAIL, EZ.EZPLUSREWARDCODE,  
EZ.EZPLUSREWARDSEARNEDPOINTS  
FROM CUSTOMER C  
JOIN RETAILCUSTOMER RC  
ON RC.CUSTOMERID = C.CUSTOMERID  
JOIN EZPLUS EZ  
ON EZ.EZPLUSID = RC.EZPLUSID  
WHERE EZ.EZPLUSREWARDSEARNEDPOINTS >= 2000;
```

Screen Shot of Report #3

The screenshot shows a SQL query editor with the following content:

```
/*
Report #3
Get customer first and last name, phone, email, EZPLUSREWARDCODE, EZPLUSREWARDSEARNEDPOINTS to
check customer has more than 2000 points if so offer better rates than regular customers.
*/
SELECT C.FIRSTNAME, C.LASTNAME, C.PHONE, C.EMAIL, EZ.EZPLUSREWARDCODE, EZ.EZPLUSREWARDSEARNEDPOINTS
FROM CUSTOMER C
JOIN RETAILCUSTOMER RC
ON RC.CUSTOMERID = C.CUSTOMERID
JOIN EZPLUS EZ
ON EZ.EZPLUSID = RC.EZPLUSID
WHERE EZ.EZPLUSREWARDSEARNEDPOINTS >= 2000;
```

Below the code, there is a "Query Result" window showing the following data:

	FIRSTNAME	LASTNAME	PHONE	EMAIL	EZPLUSREWARDCODE	EZPLUSREWARDSEARNEDPOINTS
1	JOHN	SMITH	987-897-0986	johnsmith23@gmail.com	LY67987U55	2000
2	DAVID	SHEPHARD	918-637-0446	davidshephard65@gmail.com	LY67987U57	20000
3	DAVID	BENJAMIN	929-137-5236	davidbenjamin56@gmail.com	LY67987U57	20000

Report #4

GUIDELINES	ANSWER
Q1 – What is the Business Scenario & Objectives?	A customer forgot his username or password for his customer user account he needs to access his account he could verify that a specific account belongs to him by providing his phone no or email address where he will code for verification.
Q2 – Target Persona/Decision Maker?	Front-desk Customer Service Team, Online Customer Service Team etc.

Q3 – What is the Decision to be made?	To assist customer who forgot their password of EZRENTAL Inc application.
Q4 – What Data is required to make the Decision?	Customer username and password
Q5 – Identify tables that contain the Data identified in Step 4?	Customer: CustomerUserAccount: username and password

SQL Query Report #4

```
SELECT CA.USERNAME, CA.PASSWORD
FROM CUSTOMERUSERACCOUNT CA, CUSTOMER C
WHERE CA.CUSTOMERUSERACCOUNTID = C.CUSTOMERUSERACCOUNTID AND
C.PHONE = '718-837-0186' AND C.EMAIL = 'richardoliver13@gmail.com';
```

Screen Shot of Query#4

The screenshot shows the SQL Worksheet interface. The query is:

```
GROUP BY RESERVATIONDROPPOFFAGENCYID;

/* REPORT #4
A customer forgot his username or password for his customeruseraccout he need to
verify that a specific account belong to him by providing his phone no or email
access his account he could
address where he will code for
verification */

SELECT CA.USERNAME, CA.PASSWORD
FROM CUSTOMERUSERACCOUNT CA, CUSTOMER C
WHERE CA.CUSTOMERUSERACCOUNTID = C.CUSTOMERUSERACCOUNTID AND
C.PHONE = '718-837-0186' AND C.EMAIL = 'richardoliver13@gmail.com';
```

The results are:

USERNAME	PASSWORD
richardoliver	richardoliver13

Query Result | All Rows Fetched: 1 in 0.057 seconds

Report #5

Q1 – What is the Business Scenario & Objectives?	Get the information about Reservation drop off Agency ID and the number of vehicle where most of the customer drop off their vehicle so the maintenance people will be sent to busiest agency.
Q2 – Target Persona/Decision Maker?	Vehicle Maintenance (people who manage and execute any repairs of the vehicles etc.)
Q3 – What is the Decision to be made?	To get information about busiest agency holding vehicle.
Q4 – What Data is required to make the Decision?	RESERVATIONDROPOFFAGENCYID
Q5 – Identify tables that contain the Data identified in Step 4?	Reservation: RESERVATIONDROPOFFAGENCYID

SQL Query Report #5

```

SELECT * FROM (
    SELECT RESERVATIONDROPOFFAGENCYID AS RENTAL_AGENCY_ID, COUNT(*) AS
    NUMBER_OF_VEHICLE
    FROM RESERVATION
    GROUP BY RESERVATIONDROPOFFAGENCYID
    ORDER BY NUMBER_OF_VEHICLE DESC)
    FETCH FIRST 1 ROWS ONLY;

```

Screen Shot of Report #5

The screenshot shows an SQL Worksheet window with the following details:

- Toolbar:** Includes icons for Run, Stop, Refresh, Save, and others.
- Tab Bar:** Shows "Worksheet" and "Query Builder".
- Query Editor:**

```

/* REPORT #5
Get the information about Reservation drop off Agency ID and the number of vehicle where most of the customer
drop off their vehicle so the maintenance people will be send to busiest agency.
*/
SELECT * FROM (
    SELECT RESERVATIONDROPOFFAGENCYID AS RENTALAGENCY, COUNT(*) AS NUMBER_OF_VEHICLE
    FROM RESERVATION
    GROUP BY RESERVATIONDROPOFFAGENCYID
    ORDER BY NUMBER_OF_VEHICLE DESC)
    FETCH FIRST 1 ROWS ONLY;

```
- Query Result:**

All Rows Fetched: 1 in 0.049 seconds

RENTALAGENCY	NUMBER_OF_VEHICLE
98765456	3

Report #6

GUIDELINES	ANSWER
Q1 – What is the Business Scenario & Objectives?	Get the information about vehicle rental category which was used by customer the most and report to the car inventory warehouse to have that vehicle rental category enough for customers.
Q2 – Target Persona/Decision Maker?	Vehicle Inventory managers – People who manage the inventory of rental vehicles and manage the fleet.
Q3 – What is the Decision to be made?	To get information about vehicle rental category which is used by customer the most.
Q4 – What Data is required to make the Decision?	RESERVATIONDROPOFFAGENCYID
Q5 – Identify tables that contain the Data identified in Step 4?	Reservation: RESERVATIONDROPOFFAGENCYID

SQL Query Report #6

```

SELECT VEHICLERENTALCATEGORYID ,COUNT(*) AS TIMES_ORDERED
FROM RESERVATION
GROUP BY VEHICLERENTALCATEGORYID
ORDER BY TIMES_ORDERED DESC
FETCH FIRST 1 ROWS ONLY;

```

Screen Shot of Report #6

```

/* 
REPORT #6
Get the information about vehicle rental category which was used by customer the most and report to the
car inventory warehouse to have that vehicle rental category enough for customers.
*/

SELECT VEHICLERENTALCATEGORYID ,COUNT(*) AS TIMES_ORDERED
FROM RESERVATION
GROUP BY VEHICLERENTALCATEGORYID
ORDER BY TIMES_ORDERED DESC
FETCH FIRST 1 ROWS ONLY

```

Query Result | All Rows Fetched: 1 in 0.006 seconds

VEHICLERENTALCATEGORYID	TIMES_ORDERED
1	3

Report #7

GUIDELINES	ANSWER
Q1 – What is the Business Scenario & Objectives?	Get the information of customer who made reservation but cancelled, send them an email for a survey if it's because EZRENTAL Inc rates were high, did they find any fault in EZRENTAL Inc application which they would love to see improved.
Q2 – Target Persona/Decision Maker?	Marketing and Human relations
Q3 – What is the Decision to be made?	To get information about customer who cancelled the reservation.
Q4 – What Data is required to make the Decision?	Customer first and last name, phone, email, reservationid, reservationstatusdesc
Q5 – Identify tables that contain the Data identified in Step 4?	Customer: first and last name, Phone number, Customer email Reservation: Reservationid Reservationstatus: Reservationstatusdesc

SQL Query Report #7

```

SELECT C.FIRSTNAME, C.LASTNAME, C.EMAIL, C.PHONE, R.RESERVATIONID,
RS.RESERVATIONSTATUSDESC
FROM CUSTOMER C
INNER JOIN RESERVATION R

```

```

ON C.CUSTOMERID = R.CUSTOMERID
INNER JOIN RESERVATIONSTATUS RS
ON R.RESERVATIONSTATUSID = RS.RESERVATIONSTATUSID
WHERE RS.RESERVATIONSTATUSID = 3;

```

Screen Shot of Report #7

The screenshot shows a SQL Worksheet interface. The top menu bar includes 'SQL Worksheet' and 'History'. Below the menu is a toolbar with various icons. The main area has two tabs: 'Worksheet' and 'Query Builder'. The 'Worksheet' tab is active, displaying a query and its results. The query is:

```

Get the information of customer who made reservation but cancelled, send them an email for a survey if it's because EZRENTAL Inc rates were high, did they find any fault in EZRENTAL Inc application which they would love to see improved.
*/
SELECT C.FIRSTNAME, C.LASTNAME, C.EMAIL, C.PHONE, R.RESERVATIONID, RS.RESERVATIONSTATUSDESC
FROM CUSTOMER C
INNER JOIN RESERVATION R
ON C.CUSTOMERID = R.CUSTOMERID
INNER JOIN RESERVATIONSTATUS RS
ON R.RESERVATIONSTATUSID = RS.RESERVATIONSTATUSID
WHERE RS.RESERVATIONSTATUSID = 3;

```

Below the query, the 'Query Result' tab is selected, showing the following table:

	FIRSTNAME	LASTNAME	EMAIL	PHONE	RESERVATIONID	RESERVATIONSTATUSDESC
1	MERIDITH	GREY	meridithgrey3@gmail.com	918-637-0236	87854329	CANCELLED

Report #8

GUIDELINES	ANSWER
Q1 – What is the Business Scenario & Objectives?	Get the information for a customer who call the front desk to get the information about his reservation and reservationstatus and the rentalagency to pick his car from whenever they forgot.
Q2 – Target Persona/Decision Maker?	Front-desk Customer Service Team, Online Customer Service Team
Q3 – What is the Decision to be made?	To get information about customer who forgot from which rentalagency they are picking up car.

Q4 – What Data is required to make the Decision?	Customer first and last name, phone, email, reservationid, reservationstatusdesc, rentalagencyid, rentalagencyname
Q5 – Identify tables that contain the Data identified in Step 4?	Customer: first and last name, Phone number, email Reservation: Reservationid Reservationstatus: Reservationstatusdesc RentalAgency: Rentalagencyid, Rentalagencyname

SQL Query Report #8

```

SELECT C.FIRSTNAME, C.LASTNAME, C.PHONE, C.EMAIL,
       R.RESERVATIONID, R.RESERVATIONSTATUSID,
       RS.RESERVATIONSTATUSDESC,
       RA.RENTALAGENCYID, RA.RENTALAGENCYNAME
  FROM CUSTOMER C
 INNER JOIN RESERVATION R
    ON C.CUSTOMERID = R.CUSTOMERID
 INNER JOIN RESERVATIONSTATUS RS
    ON R.RESERVATIONSTATUSID = RS.RESERVATIONSTATUSID
 INNER JOIN RENTALAGENCY RA
    ON R.RENTALAGENCYID = RA.RENTALAGENCYID
 WHERE C.CUSTOMERID = '23';

```

Screen Shot of Report #8

The screenshot shows the SQL Server Management Studio interface. The top pane displays the SQL query:

```

/*
REPORT #8
Get the information for a customer who call the front desk to get the information about his
reservation and reservationstatus and the rentalagency to pick his car from whenever they forget.
*/

SELECT C.FIRSTNAME, C.LASTNAME, C.PHONE, C.EMAIL,
       R.RESERVATIONID, R.RESERVATIONSTATUSID,
       RS.RESERVATIONSTATUSDESC,
       RA.RENTALAGENCYID, RA.RENTALAGENCYNAME
  FROM CUSTOMER C
 INNER JOIN RESERVATION R
    ON C.CUSTOMERID = R.CUSTOMERID
 INNER JOIN RESERVATIONSTATUS RS
    ON R.RESERVATIONSTATUSID = RS.RESERVATIONSTATUSID
 INNER JOIN RENTALAGENCY RA
    ON R.RENTALAGENCYID = RA.RENTALAGENCYID
 WHERE C.CUSTOMERID = '23';

```

The bottom pane shows the results of the query:

FIRSTNAME	LASTNAME	PHONE	EMAIL	RESERVATIONID	RESERVATIONSTATUSID	RESERVATIONSTATUSDESC	RENTALAGENCYID
DAVID	SHEPHARD	918-637-0446	davidshephard65@gmail.com	87854327	1	CONFIRMED	98765457

Below the table, a message says "Activate Windows".

Report #9

GUIDELINES	ANSWER
Q1 – What is the Business Scenario & Objectives?	A customer called the EZRENTAL Inc with a query regarding a vehicle which would be a Minivan with Disability option and the location to pick it from should be NY because customer lives there and wants to pick it from there.
Q2 – Target Persona/Decision Maker?	Customers who rent vehicle
Q3 – What is the Decision to be made?	To get information about vehicle.
Q4 – What Data is required to make the Decision?	VehicleID, Vehicle make, model, licenseplatenumber Minivan Disability Option
Q5 – Identify tables that contain the Data identified in Step 4?	Vehicle: VehicleID, Vehicle make, model, licenseplatenumber Minivan: Minivan Disability Option

SQL Query Report #9

```
SELECT V.VEHICLEID, V.MAKE, V.MODEL, V.LICENSEPLATENUMBER,
       M.HASDISABILITYOPTION
  FROM VEHICLE V
 INNER JOIN MINIVAN M
    ON V.VEHICLEID = M.VEHICLEID
   WHERE M.HASDISABILITYOPTION = 'Y' AND
        V.LICENSEPLATESTATE = 'NY';
```

Screen Shot of Report #9

```

/*
REPORT #9
A customer called the EZRENTAL Inc with a query regarding a vehicle which would be a Minivan with
Disability option and also the location to pick it from should be NY because customer lives there
and wants to pick it from there.
*/

SELECT V.VEHICLEID, V.MAKE, V.MODEL, V.LICENSEPLATENUMBER,
       M.HASDISABILITYOPTION
  FROM VEHICLE V
 INNER JOIN MINIVAN M
    ON V.VEHICLEID = M.VEHICLEID
   WHERE M.HASDISABILITYOPTION = 'Y' AND
        V.LICENSEPLATESTATE = 'NY';

```

Query Result x | Script Output x | Query Result 1 x | Query Result 2 x

SQL | All Rows Fetched: 1 in 0.004 seconds

	VEHICLEID	MAKE	MODEL	LICENSEPLATENUMBER	HASDISABILITYOPTION
1	29	INFINNITI	Q5	XYL-314	Y

Report #10

GUIDELINES	ANSWER
Q1 – What is the Business Scenario & Objectives?	A vehicle whose status is fulfilled came to the cleaning staff and they noticed that someone lost their wallet now write query for a customer whose staff is lost.
Q2 – Target Persona/Decision Maker?	Customers and management & staff
Q3 – What is the Decision to be made?	To get information about customer.
Q4 – What Data is required to make the Decision?	Customer first, last name, email and phone
Q5 – Identify tables that contain the Data identified in Step 4?	Customer: Customer first, last name, email and phone

SQL Query Report #10

```

SELECT C.FIRSTNAME, C.LASTNAME, C.PHONE, C.EMAIL
  FROM CUSTOMER C
 INNER JOIN RESERVATION R
    ON C.CUSTOMERID = R.CUSTOMERID

```

```
WHERE RESERVATIONID = '87854328';
```

Screen Shot of Report #10

The screenshot shows a SQL query editor interface. At the top, there is a script pane containing a multi-line comment block and a SELECT statement. The SELECT statement is highlighted with a yellow background. Below the script pane is a toolbar with icons for Query Result, Script Output, and two Query Result tabs. The main area displays the results of the query in a table format.

```
/*  
REPORT #10  
A vehicle whose status is fulfilled came to the cleaning staff and they noticed that someone lost  
their wallet now write query for a customer whose staff is lost.  
*/  
  
SELECT C.FIRSTNAME, C.LASTNAME, C.PHONE, C.EMAIL  
FROM CUSTOMER C  
INNER JOIN RESERVATION R  
ON C.CUSTOMERID = R.CUSTOMERID  
WHERE RESERVATIONID = '87854328';
```

FIRSTNAME	LASTNAME	PHONE	EMAIL
1 RICHARD	OLIVER	718-837-0186	richardoliver13@gmail.com

Business Reports Stored Procedures & Summary

Conclusion