Auto Rental Management System

EZRental

Table of Contents

PROJECT 1 – EZRental Auto Rental POS Management System Database Design and Implementation 3
Executive Summary 3
Problem Statement & Objectives 4
Project Management Methodology 5
Database Design Deliverable #1a – Application Business requirements 11
Database Design Deliverable #1b – Application Development Technical Requirements 22
Application Physical Technical Architecture 31
Application Development Features and Functionalities (Agile Backlog) 38
Database Management System Development Environment & Physical Architecture 53
Project Roles & Responsibilities 54
Database Design Deliverable #2 – ER/EER Conceptual Model Diagram 56
Database Design Deliverable #3 – Normalized Logical Model Diagram 57
Database Design Deliverable #4 – Physical Model Data Dictionary 60
Database Design Deliverable #5 – Physical Model Schema Design Diagram 68
Database Implementation Deliverable #6 – Development & Implementation 69
Database Implementation Deliverable #7 – Implemented Physical Schema Diagram 73
Database Implementation Deliverable #8 – Database Validation Testing 74
Conclusion 106

PROJECT 2 – EZRental POS Operational & Strategic Decision Making via Business Reports 107
Project #2 Objectives 107
Business Reports Queries & Summary 108
Business Reports Stored Procedures & Summary 117
Conclusion 132

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

Executive Summary

In PROJECT 1 aims to design and implement a Database Management System. The following deliverables are included in the project:

- Application Business requirements
- Application Development Technical Requirements
- ER/EER Conceptual Model Diagram
- Normalized Logical Model Diagram
- Physical Model Data Dictionary
- Physical Model Schema Design Diagram
- Development & Implementation
- Implemented Physical Schema Diagram
- Database Validation Testing

The main objective of the project is to ensure that the DBMS functions correctly and meets the requirements of the system it is being used for. During the testing phase, various scenarios will be tested to ensure that the DBMS functions as intended, including testing for data consistency, accuracy, and performance. The successful implementation of the DBMS for EZ-Car Rental will lead to improved efficiency and accuracy in the company's operations, ultimately leading to better customer experiences and increased profitability.

After the successful implementation of PROJECT1, EZRental has come back with NEW REQUIREMENTS to upgrade the database tier to the Auto Rental Point-of-Sale Management Application. In PROJECT 2, the goal is to UPGRADE the EZRental POS by adding new features to the Auto Rental Point-of-Sale Management System for VERSION 2.0.

In PROJECT2, we will also adhere to the Project Management Methodology, utilizing the Database Application Development Lifecycle (including PLANNING, ANALYSIS, DESIGN, DEVELOPMENT/IMPLEMENTATION, and OPERATIONS/MAINTENANCE), to upgrade the initial version by accomplishing the following tasks:

1. Designing and implementing BUSINESS REPORT QUERIES that will be utilized by decision-makers across the organization. These queries will provide them with the necessary data to make crucial BUSINESS DECISIONS.

2. Packaging these BUSINESS REPORT QUERIES within STORED PROCEDURES to enhance the back-end processing. This approach will improve the overall design and performance of the system.

Problem Statement & Objectives

- The EZRental Auto Rental Management System features are designed to allow customers, both retail and corporate customers, to reserve vehicles for renting, like other in-person or online car rental systems such as Avis, Hertz, Budget, etc.

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

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

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

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

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

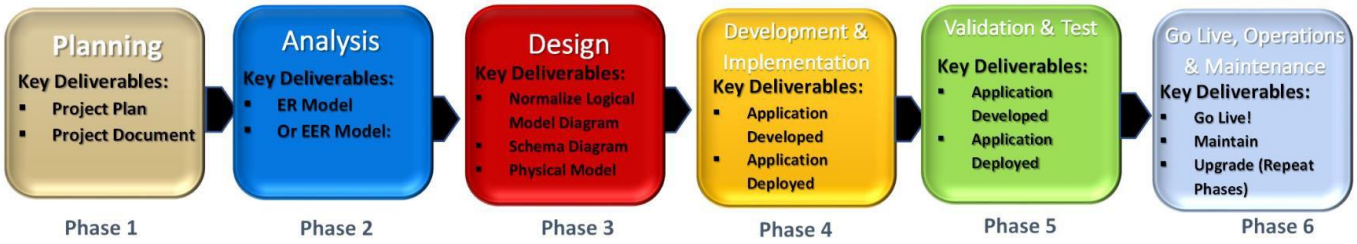
Project Management Methodology

Waterfall Methodology:

- A sequential project management design process where a project is divided into several phases.
- Each stage is executed in sequence.
- A developer cannot move on to the next phase until the current phase is completed.
- Only when all stages or phases are completed is the project considered done.

The 6 Phase Waterfall Methodology we will use for the Auto Rental Management System Application

- This section addresses how the Waterfall Project Management Methodology will be used to design, develop, and implement the DBMS Server Application for the **Auto Rental Management System ER-Rental POS (Point of Sales) Two-Tiered Client/Server & Three-Tiered Web Client/Server Applications**.
- **The Waterfall Project Management Methodology decided on by the project architects and project manager, contains 6-phase**.
- The illustration below shows the **6 PHASES** and the main deliverable for each phase, followed by table with a high-level overview of each phase:



WATEFALL PHASE	DESCRIPTION
Phase 1 – Planning & Discovery	▪ Plan the entire project & gather requirements from business. **Perform the necessary discovery via interviews, assessments, etc**., to properly plan, update the methodology and execute the project. **Gather Business Requirements**.
Phase 2 – Analysis	▪ Deep analysis of the identified **business requirements and other discovered data from interview and assessments**. This results in the foundation to derive design strategy.
Phase 3 – Design	▪ Design the solution. Organize all the requirements and produce a detailed specification of all data elements, required web forms, reports, displays, and processing or business rules. Design database models, webforms and application connectivity.
Phase 4 – Development & Implementation	▪ Implement the solution. Develop the DBMS Server Application (ER model, logical model, physical model, tables, view, stored procedures etc.).
Phase 5 – Testing and Validation	▪ Test and validate the app to confirm it meets the requirements & works.
Phase 6 – Go Live and Operations	▪ Open the application to all users. Monitor, backup & maintain the system. ▪ Database application component needs to be ready for future upgrade, therefore, have a disaster/recovery plan in place so if your computer crashes you can recover your application

- **Agile methodology** is a rapid development results-focused approach. It models our rapidly changing world.

- Agile methodology follows an incremental approach:

  - **Agile Methodology:**
    - Methodology:

      - An ***incremental*** approach to implementing a Project/Application by implementing subsets of **FEATURES** in pieces!
      - The entire Project/Application is ***DIVIDED*** and ***EXECUTED*** in increments or subset called a **SPRINT**.
      - Each **SPRINT** duration is **1, 2, 3 or 4 Weeks**.
      - How is done?

        - **Step #1** –The entire list of **FEATURES** to implement for the application called the **PRODUCT BACKLOG** is gathered and listed in a document.
        - **Step #2** – In **Agile Methodology**, we ***DON'T*** implement the entire **BACKLOG** to create the application ***as in Waterfall Methodology***, but instead, divided the **BACKLOG** into smaller subsets called **SPRINT BACKLOGS**. Each subset or **SPRINT BACKLOGS** is executed in **1, 2, 3 or 4 Weeks SPRINT**.
        - **Step #3** – At the ***END*** of each **SPRINT**, the **SPRINT FEATURE BACKLOG feature(s) are released to the USERS for use**, thus application starts to be used by the users immediately after the first **SPRINT** unlike ***Waterfall where the application is used at the end when all phases are executed!***
        - **Step #4** – **ONLY when *ALL* SPRINT BACKLOGS are executed then the application development is completed! Nevertheless, the application has been in use by the USERS from the start of the project!**

- **Scrum & Kanban are the most popular Agile Methodologies, and you will most likely come across these in the industry.**
- **We will focus on Scrum.**

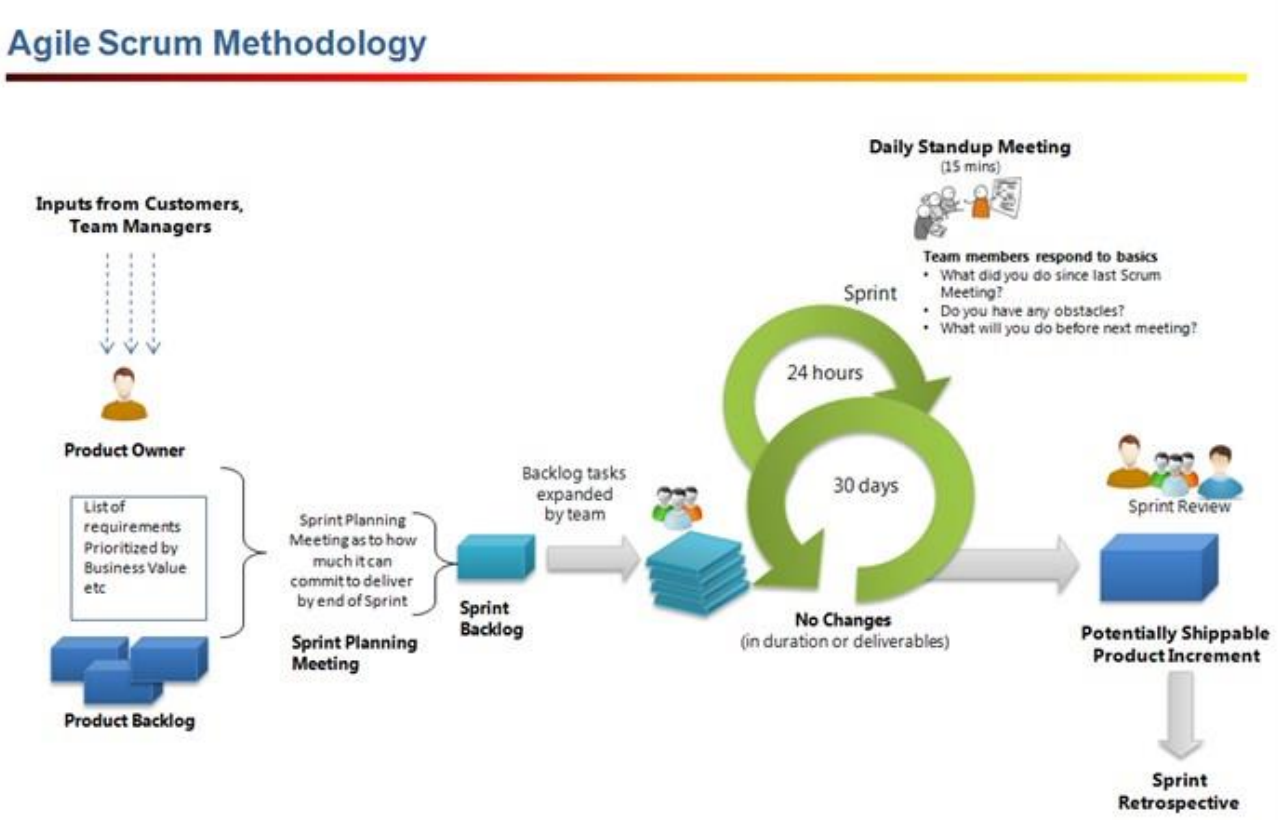
Agile Scrum Project Management in a Nutshell

- **What is Agile SCRUM Methodololgy in a nutshell?**

  - **Agile is the evolution of Waterfall**. Another Project Management Methodology that considers unexpected changes during the lifecycle of a project to ensure the project is delivered within the expected timelines.
  - **SCRUM** – is one of many Agile Methodology and the most popular.

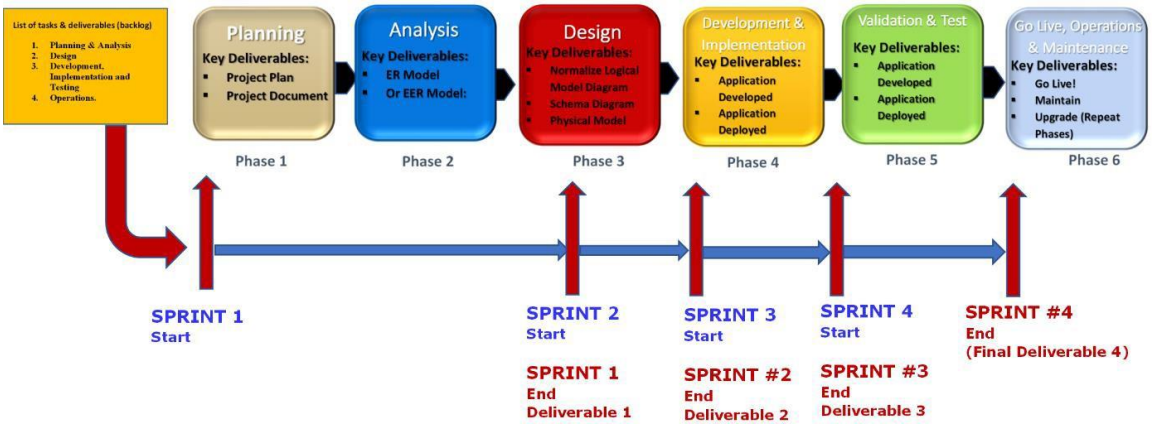
Agile	Waterfall Comparison
Business Teams, User representative, IT Project Architects, and any stakeholders, met to plan the project and assign the various **SCRUM Team** **(program owner, product owner, scrum master, scrum dev team, etc.) is assembled and roles defined, etc.**	▪ **Same as waterfall.**
In the meeting, project requirements are discussed and listed bythe business or stakeholders. Examples of an application that requires 20 features to be implemented at the end of the project.This full feature list is called a PRODUCT BACKLOG in SCRUM.	▪ **Same as waterfall.**
The project is broken into units called SPRINTS. A sprintduration is short about 2 to 4 weeks.	▪ **Like Waterfall which uses phases.**
The total feature list or BACKLOG is broken into sub-unitscalled SPRINT BACKLOGS and each SPRINT BACKLOGS to be executed by SPRINT.	▪ **NOT like Waterfall. In waterfall all features are implemented in every phase until last phase is complete. Project is driven by the list of features.**
Each **SPRINT BACKLOGS** is executed and implemented sequentially within a **SPRINT**. **IMPORTANT!** At the **END** of that **SPRINT** a finished product with the features of the **SPRINTBACKLOGS** is deployed to all the application users.	▪ **NOT like Waterfall. In waterfall all features are implemented in every phase not just a subset, and ALLFEATURES ARE DEPLOYED at the end of the last phase.**
The **2-to-4-week SPRINT** process: 1. ***SPRINT BACKLOG PLANNING*** 2. ***SPRINT EXECUTION:*** a) ***DESIGN*** b) ***DEVELOPMENT*** c) ***Daily 15-minute Scrum meetings*** d) ***DEPLOY SPRINT BACKLOG FEATURES TOUSERS*** e) ***RETROSPECTIVE MEETING TO EVALUATE SUCCESS OF SPRINT & NEEDED PROCESS IMPROVEMENTS ETC.*** 3. ***REPEAT AND INCREMENT TO NEXT SPRINT BACKLOG.***	▪ **Execution NOT like Waterfall. In waterfall, design, development is done for all features NOT a SPRINT.** ▪ **NO 15 minutes daily meetings, usually end of week status meetings because the scope is large since is ALL FEATURES not a subset that is being managed and created, team needs more time during the week toget work done.**
Process is repeated until ALL SPRINT BACKLOGS are executed, and entire PRODUCT BACKLOG is delivered tousers.	▪ **NOT like Waterfall where ALL FEATURES are deployedat the end of the last phase NOT in increments.**

- Pictorial illustration of **Agile SCRUM** Project Management Methodology:



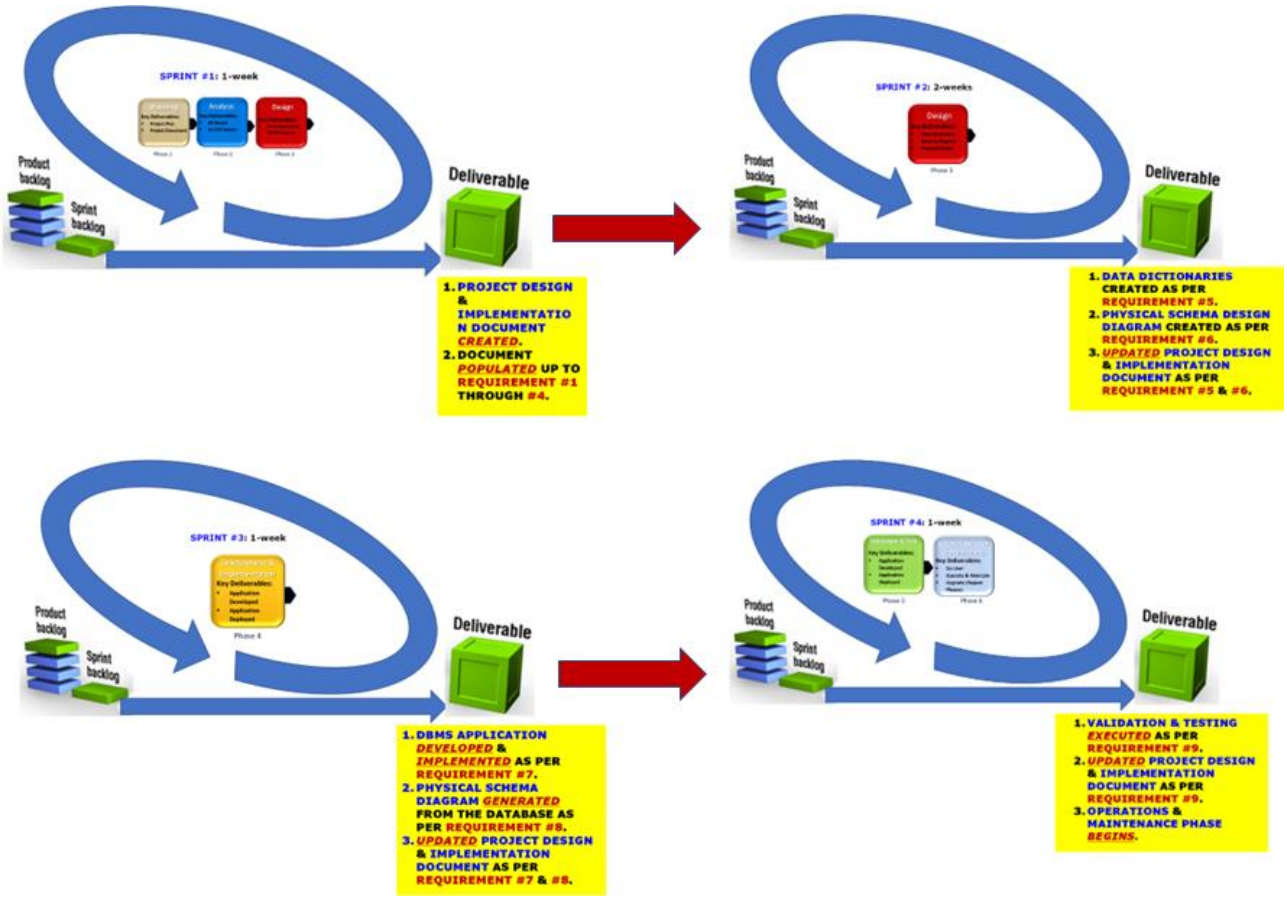
Combination of Waterfall & Agile Methodologies used in the Auto Rental Management System Application

- The strategy to embed the Agile methodology within the Waterfall methodology is as follows:
  - Agile can be customized to best fit the needs. There is no official methodology of combining Agile & Waterfall, nevertheless, Agile is sort of Waterfall executed by SPRINTS. There are key differences, but from a high-level.
  - For this project, the approach will be as follows:
    1. Use Waterfall as the primary project management methodology.
    2. Break up each of the database development and implementation waterfall phases as SPRINTS deliverables.
  - This is done because a standard database development lifecycle is based on waterfall and makes sense for this project. But we will add on top of that the SPRINT approach of Agile and consider the end of each Waterfall phase as a deliverable.
  - Future projects in the course will be based on Agile only
- We will do it by first dividing the **Waterfall Methodology Phases** into **SPRINTS** for **4 Sprints** for a period of **5-Weeks**. Diagram and **SPRINTS** descriptions table shown below:

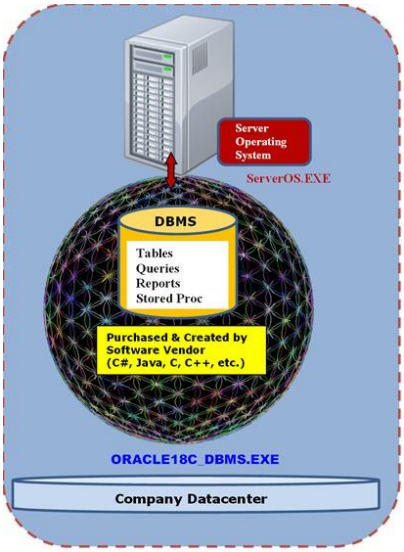


AGILE SPRINT #	WATERFALL PHASE	Output Deliverable
SPRINT #1	Planning	**1. Create Project Document – Formatted and populated as per requirements.** **2. Business Requirements (Included in Project Document – List of Business & Technical Requirements from customer.**
	Analysis	**3. ER/EER Conceptual Model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

- Second, we will break up the Waterfall Phased into **SPRINT** through the **Agile Methodology Process** for a total of **4 Sprints** with a total of **5-Weeks** shown in diagram below:



- **Outcome**:
  - Fully Designed and Implemented **ORACLE SERVER Database Management System** component **for the entire Application**:



Database Design Deliverable #1a – Application Business requirements

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

Business Requirements

About Us:

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

Current Challenges:

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

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

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

Our Agencies:

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

Our Customers:

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

All Customers (Retail & Corporate Customers)

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

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

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

Business Requirements

Our Customers (Cont.):

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

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

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

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

Business Requirements

Our Customers (Cont.):

Corporate Customers

Corporate Customers are customers who are renting vehicle during business travel and their company have a contract with **EZRental Inc**. These companies get special corporate rate for their employee's rental services. Therefore, for our **corporate customers only**, we must store the following attributes/properties: unique ***company ID*** (we have a unique ID number for each company doing business with us), ***company name***, ***company address*** which contains the elements: ***address line1, address line 2*** (which is optional and used for apartment number, suite or any additional address information required), ***city***, ***state code***, ***zip code*** (which is the two-character code for a state in the US) & ***country***, in addition, ***company contact*** which is composed of ***company representative name***, ***contact phone number*** & ***contact email*** (unique as all email addresses). And finally, we need to store the ***company discount percentage rate*** which is the discounted percentage applied to a corporate customers rental. The company Discount percentage rate is store in the database as a decimal percentage value, for example 20% is stored as 0.20, 30% as 0.30. 50% as 0.50 etc. This discount percentage (0.0x) is applied to the **Vehicle Rental Categories** which determines the price of each category to determine the total discount. Therefore, when a corporate customer rents a vehicle from a vehicle category (such as economic, compact, standard etc.), this discount percentage is applied to each of the categories during the rental/reservation process. Note that every company has a different percentage rating depending on their contract with **EZ-Rentals Inc**. For example, some companies have 20% discount towards their rentals, which would be stored as 0.20 in the database, some have 30% (0.30) etc. Vehicle Rental Categories are discussed in more details later in these requirements.

Retail Customers

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

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

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

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

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

3. *We only have two types of customers retail customer or corporate customers. No other type of customer exists.*

4. *A customer cannot be a retail & corporate customer at the same time. A customer can only rent as a retail customer or as a corporate and these transactions must be separate. We don't want our customers to be able to combine both retail customer discounts, rewards program and corporate rates at the same time.*

Business Requirements (Cont.)

Our Vehicles:

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

- Each vehicle is identified by the random number ***vehicle ID***. In addition, each vehicle is also identified by the alpha-numeric ***vehicle VIN number***. Note the following business rule on a ***vehicle VIN number***:

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

- Other attributes include the ***vehicle name*** composed of ***make***, ***model*** & **year**. Additional attributes are ***color***, also the ***license plate*** composed of the following components: ***license plate number***, ***license plate state***.
- More attributes are ***mileage***, ***transmission type*** of the vehicle. The Transmission Type attribute has business value thus used in reports and in the business processes. The values used for ***transmission type*** and a ***transmission type description*** as follows:

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

- ***seat capacity*** attribute, which is the number of seats in the vehicle. Vehicles such as ***cars*** have a seat capacity of 5 passengers (2 in front and 3 in the back), *SUVs* have 7 or 8 passengers. Cargo Vans have only 2 passenger seat capacity, Minivan have 8 to 9 passengers, special vehicles such as passenger van hold 12 passenger seat capacity, a shuttles bus can hold 16 to 20 passengers, mini-buses 30 to 40 passengers and large busses can hold 70 passengers.

- All vehicles also have a special code and description that we use to track the vehicle status named ***vehicle status ID***. This is a unique number that identifies the status of a vehicle, which works in conjunction with ***vehicle status description*** which describes the status represented by the *Vehicle Status ID*, such as **reserved**, **rented**, **available**, **maintenance**, **not available**, **transferred**, etc. Below Is the list of vehicle status IDs we are curren

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

Business Requirements (Cont.)

Our Vehicles (Cont.):

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

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

- As stated previously, there are other types of vehicles of interest that in some location we may want to store data on other than car, SUV minivans and cargo van.
- Note that the following Business Rules were identified by the business stakeholders on the vehicles:

1. *A reservation/rental can only be for one of these four categories of Vehicles or other vehicle types, not a combination.*

2. *This means, you can only rent either a car, SUV minivans, cargo van or other for a reservation or rental, not a combination such as a car & SUV at the same time. Each reservation is unique to one vehicle.*

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

1. *Every vehicle is owned by one agency. The vehicle can be pick-up and dropped-off at any agency, but only one agency is the vehicle's owning agency. An agency can own many vehicles, but a vehicle can only be owned by one agency.*

2. *A vehicle can currently be located at any agency depending on where it was dropped-off after a rental. We need to track the current agency where the vehicle is located, to arrange a transfer or a rental that will ultimately direct the vehicle to the owning agency.*

Reservation Process:

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

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

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

2. *Thus, a* ***customer*** *makes a reservation of a* ***vehicle rental category*** *at a* ***rental agency****. Therefore, the reservation process involves a* ***customer*** *a* ***vehicle rental category*** *and the* ***rental agency*** where the vehicle will be picked up.

Business Requirements (Cont.)

Reservation Process (Cont.):

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

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-Cargo Van	$19.95
15	Pick Up-Mid Size	$69.95
16	Pick Up-Full Size	$105.99
17	Motorcycle-Touring	$19.95
18	Motorcycle-Cruiser	$199.99
19	Motorcycle-Scooter	$79.95
20	Passenger Van (12 passengers)	$161.00
21	Passenger Shuttle (16 passengers)	$180.00
22	Passenger Shuttle (20 passengers)	$220.00
23	Passenger Mini-Bus (30 passengers)	$250.00
24	Passenger Mini-Bus (40 passengers)	$280.00
25	Passenger Large-Bus (80 passengers)	$300.00

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

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

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

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

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

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

Business Requirements (Cont.)

Reservation Process (Cont.):

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

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

For a reservation we must adhere to the following business rules:

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

The Rental Process:

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

For the rental process, the following business rules apply:

1. *A customer rents a vehicle Rental Category at a rental agency. This means the rental process requires the **customer, vehicle rental category,** and & **rental agency** for a rental to be complete.*
2. *A Rental includes a specific Vehicle of the vehicle rental category. A vehicle can be rented many times, but a rental is only for one vehicle only. You cannot rent multiple vehicles in one rental contract.*
3. *During the rental process we may have any of the following business rules/scenarios:*

    1) *A vehicle can be picked up at the **SAME** rental agency as indicated by the reservation and dropped off at the **SAME** rental agency.*
    2) *Or a vehicle can be picked up at the **SAME** rental agency as indicated by the reservation and dropped off at **ANOTHER** rental agency.*
    3) *Or a vehicle can be picked up at **ANOTHER** rental agency other than what was indicated by the reservation and dropped off at **SAME** rental agency of the reservation..*
    4) *Finally, a vehicle can be picked up at **ANOTHER** rental agency other than what was indicated by the reservation and dropped off at **ANOTHER** rental agency of the reservation.*

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

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

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

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

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

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

Business Requirements (Cont.)

The Rental Process (Cont.):

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

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

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

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

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

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

Business Requirements (Cont.)

The Rental Process (Cont.):

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

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

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

Our Employees:

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

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

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

Security & Application Access:

To access our systems proper security and authentication is required. Only authorized users can have access our agencies Point-0f-Sales & Back-End Management systems. In addition to our **EZRental.com** portal by our customers. Therefore, due to security and regulatory compliance purpose, we want to separate the employee access data from the customer access data by using two separate user accounts:

- Employee user accounts
- Customer user accounts

Security Access for Employees to Computer Systems in our Agencies (Employee User Accounts):
For our authorized employees & customer service employees to access the agencies Point-0f-Sales & Back-End Management systems they need to log in by entering a username & password for access to the application. This means every employee owns an employee user account.

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

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

Business Requirements (Cont.)

Security Access for our Customers who register for our EZ-CarRental.com web site (Customer User Accounts):
Customer who accesses our online portal to reserve and rent our vehicles also need a username and password to access our system, therefore each customer owns a customer user account.

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

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

Vehicle Transportation:

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

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

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

Note that transportation to and from Rental Agency is executed by an employee who is part of a transportation team or drivers. Therefore, when an employee executes a transport request of a vehicle to and from Rental Agencies, we need to capture the following information:

- *Transport pickup agency ID, Transport drop-off agency ID, Driver departure date, driver departure time, vehicle pick up date, vehicle pick up time, transport completed arrival date, transport completed arrival time, estimated arrival date, estimated arrival time,* & actual *transport time to completion.*

- In addition, we need to know at any time the transport status and transport status description of the transfer, such as: transfer completed, on route to pick up location, on route from pick up location, etc. Therefore, we need to capture the ***Transport Status ID*** or unique number that identifies a status and the ***Transport Status Description***, or description of each status ID. Currently we track a transportation event using the following ID and description:

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

The goal again is to be able to know where our vehicles are located at any time and their status.

Business Requirements (Cont.)

Conclusion:

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

Database Design Deliverable #1b – Application Development Technical Requirements

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

Application Development & Technical Requirements

Introduction & Current Challenges

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

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

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

Rental Agencies Application & Technical Requirements:

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

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

Rental Agency Application Architecture Requirements:

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

1. Client application processing, transaction and response must be fast to minimize service time for a customer.
2. All transaction processing should be done in the user's computer or desktop for fast processing and response.
3. Application Architecture must be reusable and scalable to support future updates and new feature enhancements, without a long development lifecycle.
4. Depending on the architecture NYC-Tech Solutions Inc., decides for the application in the rental agencies (Desktop client or Web client), the primary Application Development Platform we use is ***C# & .NET technologies***. For any Web related development, we support JavaScript, React, NodeJs and other standard Web Technologies. We have aligned ***C#.NET*** & **ASP.NET Web developers** that have been assigned to assist, support and update the application once NYCTech consultants complete the project and development of this system.
5. Rental Agency Desktop Application Security Authentication System – Proper security and authentication must be implemented to make sure only authorized customer service representative and other rental office employees can access the Point-0f-Sales with appropriate conditional access.

Application Development & Technical Requirements (Cont.)

Rental Agency Application Features and Functionalities Requirements:

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

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

Rental Agency Application Graphical User Interface Requirements:

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

  - **Addresses**

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

  - **Discount Codes:**

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

Application Development & Technical Requirements

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

- **EZPlus Rewards Codes:**

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

- **Company Name:**

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

- **Vehicle Status:**

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

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

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

- **Rental Agency:**

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

Application Development & Technical Requirements

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

- **Vehicle Rental Category:**

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

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

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

- **Reservation Status:**

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

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

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

Application Development & Technical Requirements

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

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

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

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

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

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

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

Application Development & Technical Requirements

- **Rental Insurance Option:**

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

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

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

Application Development & Technical Requirements

- **Transportation Reason Option:**

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

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

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

- **Transportation Reason Option:**

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

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

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

Application Development & Technical Requirements (Cont.)

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

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

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

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

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

Web Portal Application Web Pages User Interface Requirements:

The web pages graphical UI requirements are listed below:

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

Application Development & Technical Requirements

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

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

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

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

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

Application Physical Technical Architecture

Application Physical Architecture Overview

- After a design meeting with the architects and full-stack developers a decision was made on the application architecture for the **EZRental POS** application.
- After a thorough review of both the ***business requirements*** and ***technical requirements*** by the project team, the resultant decisions on architecture (s) were based on the following:

- **Rental Agency Employees:**
  - The system in our agencies used by the customer service representatives or front-line workers, must be able to quickly respond and execute the necessary requests such as

    - **POS Customer Management (Retail Customer & Corporate Customer) features** such as ***Customer Search & Print, New Customer Registration***, ***Customer Update***, ***Customer Deletion***, & ***Customer Listing functionalities***
    - **POS Vehicle Reservation, Rental & Return Management Feature** such as ***Vehicle Reservations, Vehicle Rental & Vehicle Return functionalities.***
    - **POS Vehicle Inventory Management Feature** allows inventory personnel and employees to bulk-manage vehicles such as **Cars**, **SUVs, Mini-Vans**, ***Cargo Vans,*** and other vehicles to be ***searched***, ***added***, ***updated***, ***deleted***, ***printed***, ***listed*** etc.
    - **POS Credit Card Management Feature** such as ***Credit Card Search & Print, New Credit Card Registration***, ***Credit Card Update***, ***Credit Card Deletion***, & ***Credit Card Listing functionalities.***

  - customer reservations, rentals, returns, customer management etc., therefore fast response and performance is required to quickly service a customer and minimize the wait. This is more important in Airports and other high-traffic locations.
  - We also want to provide our customer service agents with a rich user-interface experience.
  - The system in the agencies is also used by other back-end personnel such as vehicle inventory managers and administrators, service personnel, vehicle transport drivers, etc. Therefore, the system needs to also perform well.

- **Corporate Offices:**
  - The corporate offices are where our business operations are managed by our business employees & employees at the rental agencies via the INTRANET Web Portal.
  - These features include:

    - **Intranet Web Enterprise Resource Planning Systems (ERP) Portal Feature** such as providing access to **Enterprise Resource Planning Systems (ERP) Applications** such as: ***Customer Credit Card Management System***, ***Vehicle Inventory Management System***, ***Customer Relationship Management (CRM)***, ***Human Resource Management System***, & ***Finance & Operations System, Marketing System, Customer & Field Service System etc.***
    - **Web EZRental Point-of-Sales Corporate Management Feature** which allows employees to ***manage*** & ***execute*** **Point-of-Sales (POS)** transaction via the **Intranet Web Poral** such as: ***Search Customer Profile Information***, ***Customer Account Management***, ***Customer Registration***, ***Customer Update***, ***Customer Delete***, & ***Customer Listing functionalities, Manage & Make Reservations of a Vehicle***, ***Manage an existing Rental***, ***etc.***

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

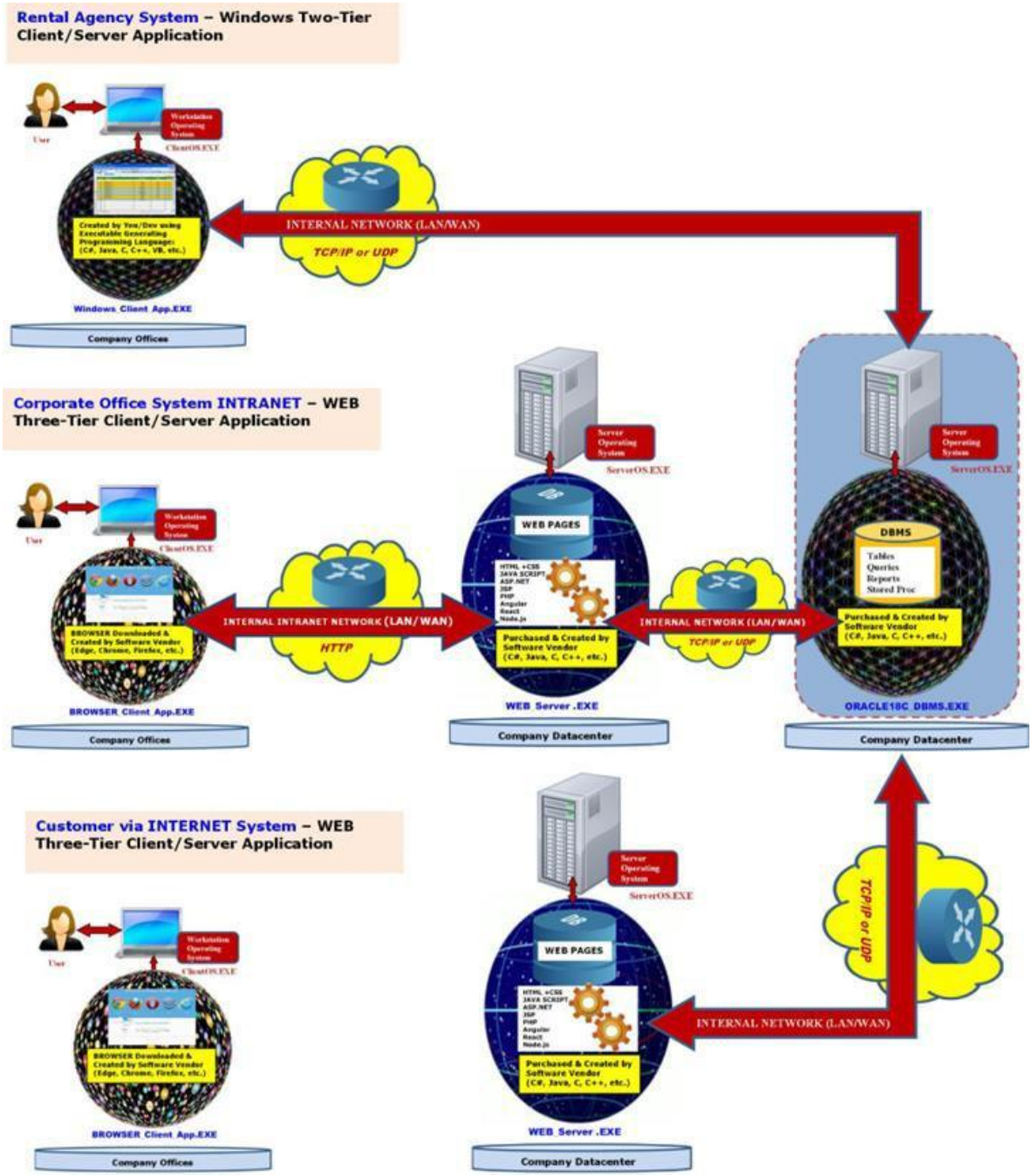
- **Customer self-service:**
  - Customers who wish to make reservations and manage their reservations and rentals online via the internet, should be able to do so from anywhere in the world via our web portal.
  - This also includes good user-experience.

- The target applications architecture and components are as follows:

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

- **GOAL IN THIS PROJECT IS TO DESIGN & IMPLEMENT THE DBMS SERVER APPLICATION FOR THE THREE APPLICATIONS which includes the Two-Tiered Windows-Client Client/Server Application for Rental Agencies, the Three-Tiered Web-based Client/Server for Corporate Offices & Three-Tiered Web-based Client/Server for Customer Internet Application!**

- Below is a pictorial diagram of this multi-component client/server architecture. Note that both the **Windows Client Application**, the **Corporate Office Browser Web Client Applications**, and the **Customer Internet Browser Web Client Application** are all sharing the same **Oracle 18c Server Express DBMS Server Application**:



Physical Architecture Hardware & Software Inventory

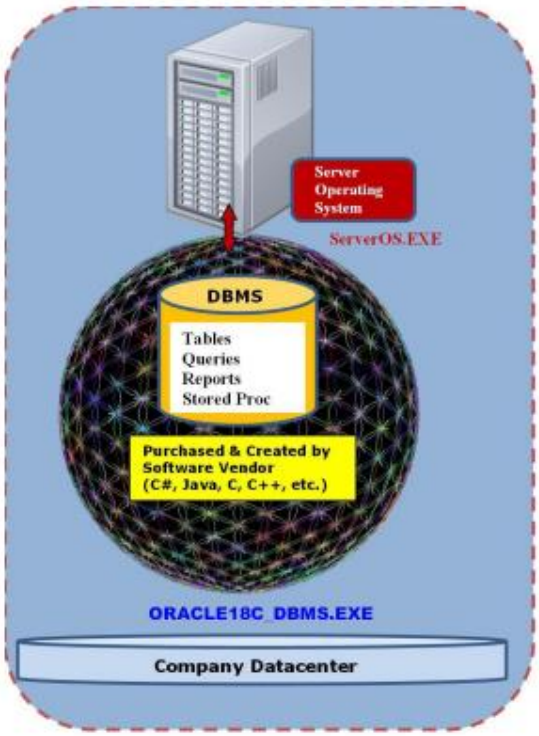
- The list of hardware and software required for purchasing/downloading is shown in the table below.

Architecture Component	Hardware Purchase & Inventory	Software Purchase & Inventory
Rental Agency Windows Client/Server Infrastructure	**User Desktop/Laptop Computer:** ▪ As needed, purchase/upgrade Memory, Processor, Hard disk etc., only if PCs need to be upgraded to support the Windows Client application. **Network Hardware:** ▪ Required Switches, Routers & other network peripherals required to support the networking requirements of the application. **Office Desktop/Laptop Installation:** ▪ Assemble team to install the Windows Client Application to user's computers or use Computer Management Tool to package and deploy the applications to the user's computers.	**User Desktop/Laptop Operating System:** ▪ Target OS – Windows 10, MAC Os etc. **Application Development & Framework:** ▪ Purchase/download required Application Development Tools such as Visual Studio, Eclipse, NetBeans etc., to develop the Windows Client Application. ▪ Download/Purchase any required framework for developing the Windows Client Application.
Corporate Office Intranet Web Client/Server Infrastructure	**Physical Server:** ▪ 1 Physical Server Computer with required specifications. **Network Hardware:** ▪ Required Switches, Routers & other network peripherals. **Datacenter Hardware Installation:** ▪ Required racks, cooling, electrical and other hardware to support installation of physical servers, etc.	**Server Operating System:** ▪ Target OS – Windows Server, Linux, Unix, etc. **Web Server Application:** ▪ Purchase/download target Web Server Application such as MSFT IIS, Apache, other. **Web Development Framework:** ▪ Purchase/download required Web Development Tools ▪ Purchase/download required web development framework such as React, Angular, ASP.NET etc.

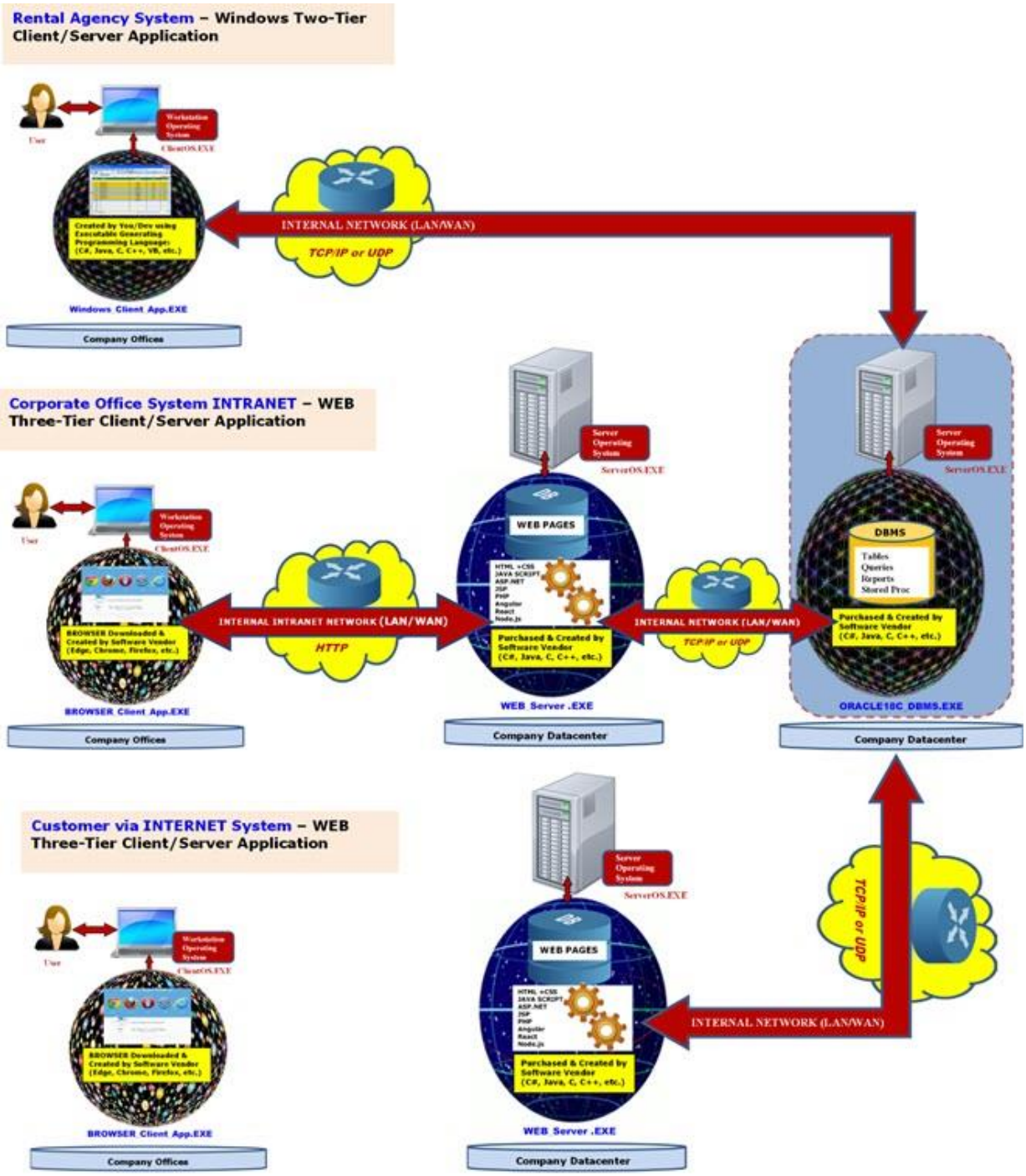
Customer Facing Internet Web Client/Server Infrastructure	**Physical Server:** ▪ 1 Physical Server Computer with required specifications. **Network Hardware:** ▪ Required Switches, Routers & other network peripherals. **Datacenter Hardware Installation:** ▪ Required racks, cooling, electrical and other hardware to support installation of physical servers, etc.	**Server Operating System:** ▪ Target OS – Windows Server, Linux, Unix, etc. **Web Server Application:** ▪ Purchase/download target Web Server Application such as MSFT IIS, Apache, other. **Web Development Framework:** ▪ Purchase/download required Web Development Tools ▪ Purchase/download required web development framework such as React, Angular, ASP.NET etc.

Shared Database Management System Infrastructure for Agency Two-Tier Client Server, Office, and Customer Web Portals	**Physical Server:** ▪ 1 Physical Server Computer with required specifications. **Network Hardware:** ▪ Required Switches, Routers & other network peripherals. **Datacenter Hardware Installation:** ▪ Required racks, cooling, electrical and other hardware to support installation of physical servers, etc.	**Server Operating System:** ▪ Target OS – Windows Server, Linux, Unix, etc. **Database Management System Application:** ▪ Purchase/download target DBMS Server Application, which in this case is either **Microsoft SQL Server** or **Oracle 18c Express Edition**. ▪ Purchase/download the **DBMS Dev & Admin tools** such as **MS SQL Server Management Studio** for **Microsoft DBMS**, or **Oracle SQL Developer** for **Oracle DBMS** etc.

❑ Below is the BIG PICTURE or illustration of the Oracle 18c Express Edition DBMS Server Application you are creating in this Project #1 after design, development & implementation:



- Another view or **BIG PICTURE** of the **Oracle 18c Express Edition DBMS Server Application** is creating in **EZRental Auto Rental Management System** to *HOST* the **DATA** for **The THREE APPLICATIONS which includes the Two-Tiered Windows-Client Client/Server Application for Rental Agencies, the Three-Tiered Web-based Client/Server for Corporate Offices & Three-Tiered Web-based Client/Server for Customer Internet Application**:



Application Development Features and Functionalities (Agile Backlog)

Pilot Deployment Features & Functionality Overview

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

  - A pilot is a small, limited deployment that consists of rolling out the new software to a select group of users in the organization.
  - A pilot is part of project management methodology that ensures an application is stable and reliable.
  - Deploying a new application to the entire user population that will use the application is risky. No matter how much planning and effort is taken to make sure the application is stable and reliable, there are always unexpected challenges that arise which a development team may not have planned for.
  - By first deploying a pilot deployment, a development team can flush out these unexpected challenges during the pilot and identify/address them before deploying to the entire population.

- A decision was made for the **DEPLOYMENT PILOT** to include **ONLY** the following features:

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

Features & Functionality Overview

- During analysis and meetings, it was decided that the application is to deliver the following **10 features** & **functionality**. These **10 features** together make up the **AGILE BACKLOG**. Note that features highlighted in **BLUE** & **ORANGE** are in scope for **Auto Rental Management System**. Features highlighted in **GREY** **are out of scope for PROJECT #1**.

Feature #	Feature Description
FEATURE #1A	**FEATURE #1A** **FEATURE #1A – EZRental Rental Agency Point-of-Sales (POS) System CUSTOMER SERVICE – CUSTOMER MANAGEMENT SYSTEM:** ▪ **WINDOWS CLIENT POINT-OF SALES SYSTEM** – ***WINDOWS UI FORM(S) FRONT-END APPLICATION & OOP PROGRAMMING*** features used by ***customer service representative employees*** via the ***Point-of-Sales computer*** machine in the ***Rental Agencies*** to service customer's **CUSTOMER MANAGEMENT** requests or transactions. ▪ The following are features and functionality that are required for this application feature: ○ **POS Customer Management Feature:** **POS Customer Management (Retail Customer & Corporate Customer) features** such as ***Customer Search & Print, New Customer Registration***, ***Customer Update***, ***Customer Deletion***, & ***Customer Listing functionalities***. ○ **Note that each transaction is saved to database immediately after execution:** - **Feature UI Form Requirements:** ***Design & programming*** of required ***User-Interface Forms & GUI Controls*** to support this feature. - **Feature Processing Requirements:** ***Design & programming*** of required ***Object-Oriented (OOP) Processing & Logic*** to support this feature. ▪ This feature is designed only to be used by customer service agents and other employees using the **Windows Two-Tiered Client/Server Application** in the **Rental Agencies**.
FEATURE #1B	**FEATURE #1B** **FEATURE #1B – EZRental Rental Agency Point-of-Sales (POS) Customer Management System Back-end Database Design & Implementation to support this feature:** ▪ **DATABASE SERVER BACK-END SYSTEM – *BACK-END DATABASE DESIGN & FEATURES*** **(Create the Tables & Relationships, pre-populated tables, stored procedures, views, indexes etc.,) to support this feature.**

Feature #	Feature Description
FEATURE #2A	**FEATURE #2A** **FEATURE #2A – EZRental Rental Agency Point-of-Sales (POS) System CUSTOMER SERVICE VEHICLE RESERVATION, RENTAL & RETURN FEATURE MANAGEMENT:** ▪ **WINDOWS CLIENT POINT-OF SALES SYSTEM** – ***WINDOWS UI FORM(S) FRONT-END APPLICATION & OOP PROGRAMMING*** features used by ***customer service representative employees*** via the ***Point-of-Sales computer*** machine in the ***Rental Agencies*** to service customer's **CUSTOMER VEHICLE RESERVATION, RENTAL & RETURN MANAGEMENT,** or transactions. ▪ The following are features and functionality are required for this application feature: ○ **POS Vehicle Reservation, Rental & Return Management Feature: POS Customer Vehicle Reservation, Rental & Return Management (Retail Customer & Corporate Customer) features** such as ***Vehicle Reservations, Vehicle Rental & Vehicle Return functionalities:*** - **Feature UI Form Requirements:** ***Design & programming*** of required ***User-Interface Forms & GUI Controls*** to support this feature. - **Feature Processing Requirements:** ***Design & programming*** of required ***Object-Oriented (OOP) Processing & Logic*** to support this feature. ▪ This feature is designed only to be used by customer service agents and other employees using the **Windows Two-Tiered Client/Server Application** in the **Rental Agencies**.
FEATURE #2B	**FEATURE #2B** **FEATURE #2B – EZRental Rental Agency Point-of-Sales (POS) Customer Vehicle Reservation, Rental & Return Management System Back-end Database Design & Implementation to support this feature:** ▪ **DATABASE SERVER BACK-END SYSTEM – *BACK-END DATABASE DESIGN & FEATURES*** **(Create the Tables & Relationships, pre-populated tables, stored procedures, views, indexes etc.,) to support this feature.**

Feature #	Feature Description
FEATURE #3A	**FEATURE #3A** **FEATURE #3A – EZRental Internal Back-Office Agency BACK-OFFICE VEHICLE INVENTORY MANAGEMENT SYSTEM (NOT A CUSTOMER FACING APPLICATION):** ▪ **WINDOWS CLIENT POINT-OF SALES SYSTEM** – ***WINDOWS UI FORM(S) FRONT-END APPLICATION & OOP PROGRAMMING*** features used by **Back-Office Inventory Team employees** via a **computer machine** in the ***Rental Agencies*** to service inventory needs for **VEHICLE INVENTORY MANAGEMENT,** or transactions. ▪ This is a unique Back-end system meant for inventory team employees to perform bulk **IN-MEMORY** inventory processing or management tasks on vehicles such as: ***adding*** vehicles to the system, ***searching*** for vehicles, ***updating*** vehicles, ***deleting*** vehicles, etc. The idea is that the employee can perform all these features on their computer **in-memory** repeatedly for several vehicles in one session saving to database after each transaction but managed in-memory using a **collection** or **other data structure** to manage it locally. When user is done with all inventory transactions, all transactions have been saved to database but, is still locally in the collection or other data structure and can be updated as needed. By keeping it locally in memory, the operations are faster. ▪ The following are features and functionality are required for this application feature: o **POS Vehicle Inventory Management Feature:** **POS Vehicle Inventory Management features** allows inventory personnel and employees to bulk-manage vehicles such as **Cars**, **SUVs, Mini-Vans**, ***Cargo Vans,*** and other vehicles to be ***searched***, ***added***, ***updated***, ***deleted***, ***printed***, ***listed*** etc. - **Feature UI Form Requirements:** ***Design & programming*** of required ***User-Interface Forms & GUI Controls*** to support this feature. - **Feature Processing Requirements:** ***Design & programming*** of required ***Object-Oriented (OOP) Processing & Logic*** to in the to support this feature. ▪ This back-office features is not designed to be used by customers and not available via the Web and implemented using the **Windows Two-Tiered Client/Server Application** in the **Rental Agencies**.
FEATURE #3B	**FEATURE #3B** **FEATURE #3B – EZRental Rental Agency Vehicle Inventory Management System Back-end Database Design & Implementation to support this feature:** ▪ **DATABASE SERVER BACK-END SYSTEM – *BACK-END DATABASE DESIGN & FEATURES*** **(Create the Tables & Relationships, pre-populated tables, stored procedures, views, indexes etc.,) to support this feature.**

Feature #	Feature Description
FEATURE #4A	**FEATURE #4A** **FEATURE #4A – EZRental Rental Agency Point-of-Sales (POS) BACK-OFFICE CREDIT CARD MANAGEMENT SYSTEM:** ▪ **WINDOWS CLIENT POINT-OF SALES SYSTEM** – ***WINDOWS UI FORM(S) FRONT-END APPLICATION & OOP PROGRAMMING*** features is a **back-end system** used by ***customer service representative & other employees*** via the ***Point-of-Sales computer*** machine in the ***Rental Agencies*** to service customer's **CREDIT CARD MANAGEMENT,** or transactions required when servicing customers. ▪ The following are features and functionality are required for this application with features such as: ○ **POS Credit Card Management Feature: POS Customer Credit Card Management features** such as ***Credit Card Search & Print, New Credit Card Registration***, ***Credit Card Update***, ***Credit Card Deletion***, & ***Credit Card Listing functionalities:*** - **Feature UI Form Requirements:** ***Design & programming*** of required ***User-Interface Forms & GUI Controls*** to support this feature. - **Feature Processing Requirements:** ***Design & programming*** of required ***Object-Oriented (OOP) Processing & Logic*** to support this feature. ▪ This feature is designed only to be used by customer service agents and other employees using the **Windows Two-Tiered Client/Server Application** in the **Rental Agencies**.
FEATURE #4B	**FEATURE #4B** **FEATURE #4B – EZRental Rental Agency Point-of-Sales (POS) Credit Card Management System Back-end Database Design & Implementation to support this feature:** ▪ **DATABASE SERVER BACK-END SYSTEM – *BACK-END DATABASE DESIGN & FEATURES* (Create the Tables & Relationships, pre-populated tables, stored procedures, views, indexes etc.,) to support this feature.**

Feature #	Feature Description
FEATURE #5A	**FEATURE #5A**

FEATURE #5A – EZRental Rental Agency Point-of-Sales (POS) System BACK-OFFICE EMPLOYEE & CUSTOMER USER-ACCOUNT MANAGEMENT SYSTEM:

- **WINDOWS CLIENT POINT-OF SALES SYSTEM** – ***WINDOWS UI FORM(S) BACK-OFFICE APPLICATION & OOP PROGRAMMING*** **User-Accounts Management Features** used by ***customer service representative & IT Administrator employees*** via the ***Point-of-Sales computer*** machine in the ***Rental Agencies*** to service customer's **CUSTOMER & EMPLOYEE USER ACCOUNT MANAGEMENT** requests or transactions.
- **Employee User Accounts –** These are the user accounts used by ***IT Administrators***, ***Customer Service Employees***, ***back-office employees***, and ***any employee who qualifies*** for access to the system.
- **Customer User Accounts –** These are the user accounts used by ***IT Administrators***, ***Customer Service Employees***, ***back-office employees*** and any ***employee*** who has access to the system to ***manage*** the **Customer User Accounts** for login into the **Customer Web Portal**.
- The following are features and functionality that are required for this application feature:
  - **POS User Account (Employee & Customer) Management Feature:** **POS User Account Management (Employee & Customer) features** such as:
    - **Employee User Account Feature 5A-1 –** Allows ***IT Administrators***, ***Customer Service Employees***, ***back-office employees***, and ***any employee who qualifies*** to ***manage*** employee user accounts that allow employees to login into the POS System. And perform the following tasks: ***Employee User Account Search by username, New Employee User Account Registration***, ***Employee User Account Update by username***, ***Employee User Account Deletion by username***, & ***Employee User Account Listing functionalities***. **IMPORTANT!** Note that the **password** is ***NEVER DISPLAYED or LISTED***, only the **username**!
    - **Customer User Account Feature 5A-2 –** Allows ***IT Administrators***, ***Customer Service Employees***, ***back-office employees***, and ***any employee who qualifies*** to ***manage*** customer user accounts that allow customers to login into the **Customer Web Portal System**. And perform the following tasks: ***Customer User Account Search by username, New Customer User Account Registration***, ***Customer User Account Update by username***, ***Customer User Account Deletion deletion by username***, & ***Customer User Account Listing functionalities***. **IMPORTANT!** Note that the **password** is ***NEVER DISPLAYED or LISTED***, only the **username**!
  - ***Note that each transaction is saved to database immediately after execution:***
    - **Feature UI Form Requirements:** ***Design & programming*** of required ***User-Interface Forms & GUI Controls*** to support this feature.
    - **Feature Processing Requirements:** ***Design & programming*** of required ***Object-Oriented (OOP) Processing & Logic*** to support this feature.
- This feature is designed only to be used by ***IT Administrations*** and other employees who qualify to use this system to manage both employees & customers user accounts using the **Windows Two-Tiered Client/Server Application** in the **Rental Agencies**.

FEATURE #5B	FEATURE #5B
	FEATURE #5B – EZRental Rental Agency Point-of-Sales (POS) User EMPLOYEE User Account Management System Back-end Database Design & Implementation to support this feature: ▪ **DATABASE SERVER BACK-END SYSTEM – *BACK-END DATABASE DESIGN & FEATURES* (Create the Tables & Relationships, pre-populated tables, stored procedures, views, indexes etc.,) to support this feature to store and manage EMPLOYEE USER ACCOUNTS.**
FEATURE #5C	**FEATURE #5C**
	FEATURE #5C – EZRental Rental Agency Point-of-Sales (POS) User CUSTOMER User Account Management System Back-end Database Design & Implementation to support this feature: ▪ **DATABASE SERVER BACK-END SYSTEM – *BACK-END DATABASE DESIGN & FEATURES* (Create the Tables & Relationships, pre-populated tables, stored procedures, views, indexes etc.,) to support this feature to store and manage CUSTOMER USER ACCOUNTS.**

Feature #	Feature Description
FEATURE #6A	**FEATURE #6A** **FEATURE #6A – EZRental Rental Agency Point-of-Sales (POS) System EMPLOYEES BACK-OFFICE SECURITY LOGIN AUTHENTICATION SYSTEM:** ▪ Proper ***security and authentication*** must be implemented to make sure only authorized **employees** can access the **Point-0f- Sales & Back-End Management systems** when they **login** into the **Windows Two-Tiered Client/Server Application** & **Web Three-Tiered Corporate Client/Server Application**. ▪ **WINDOWS CLIENT POINT-OF SALES SYSTEM** – ***WINDOWS UI FORM(S) BACK-OFFICE APPLICATION & OOP PROGRAMMING*** **Login Authentication features** used by ***customer service representative & IT Administrator employees*** via the ***Point-of-Sales computer*** machine in the ***Rental Agencies*** to service employee **LOGIN AUTHENTICATION SYSTEM**. ▪ The following are features and functionality that are required for this application such as: ○ **POS Employee Back-Office Security Login Authentication System:** **POS Login Authentication Access for Employees features** such as: - **Employee Authentication Feature 6A-1 –** To have access to the application, an employee (Customer Service Reps, Back-office employee etc.) must provide a **username & password**. This feature is required to be ***designed*** & ***programmed*** into the application. - **Employee Authentication Feature 6A-2 –** ***Design & programming*** of required **User-Interface Forms** & **GUI Controls** to support the **Authentication System feature**! ○ Programming includes: - **Feature UI Form Requirements:** ***Design & programming*** of required ***User-Interface Forms & GUI Controls*** to support this feature. - **Feature Processing Requirements:** ***Design & programming*** of required ***Object-Oriented (OOP) Processing & Logic*** to support this feature. ▪ This feature is designed only to be used by all employees wishing access to the **Windows Two-Tiered Client/Server Application POS System** in the **Rental Agencies**.
FEATURE #6B	**FEATURE #6B** **FEATURE #6B – EZRental Rental Agency Point-of-Sales (POS) Employee Back-Office Security Login Authentication System Back-end Database Design & Implementation to support this feature:** ▪ **DATABASE SERVER BACK-END SYSTEM – *BACK-END DATABASE DESIGN & FEATURES* (Create the Tables & Relationships, pre-populated tables, stored procedures, views, indexes etc.,) to support this feature.**

Feature #	Feature Description
FEATURE #7A	**FEATURE #7A**

FEATURE #7A – EZRental INTERNAL CORPORATE EMPLOYEE & RENTAL AGENCIES EMPLOYEES INTRANET CORPORATE BUSINESS APPLICATIONS WEB PORTAL:

- This **INTRANET (NOT THE PUBLIC INTERNET) Web Portal** EZRentalHub.com, is a **Web-based Three-Tiered Client/Server physical & Software Development Architecture** used by **CORPORATE & OTHER EMPLOYEES** to ***execute*** **Enterprise Resource Planning Systems (ERP) Applications & other Corporate Applications** online intranet via a **BROWSER**.
- **BROWSER/INTRANET WEB ERP & CORPORATE APPLICATION SYSTEM** – ***WEB UI FORM(S) FRONT-END APPLICATION & OOP PROGRAMMING*** **Enterprise Resource Planning Systems (ERP) Applications & other Corporate Applications Features** used by ***Corporate Employees*** via the **CORPORATE INTRANET** *PORTAL*.
- The following are features and functionality that are required for this **INTRANET WEB APPLICATION**:

  - **Corporate Business Application Intranet Web Portal Features:**

    - **Intranet Web Enterprise Resource Planning Systems (ERP) Portal Feature 7A-1 –** Provides access to **Enterprise Resource Planning Systems (ERP) Applications** such as: ***Customer Credit Card Management System***, ***Vehicle Inventory Management System***, ***Customer Relationship Management (CRM)***, ***Human Resource Management System***, & ***Finance & Operations System, Marketing System, Customer & Field Service System etc***.
    - **Web EZRental Point-of-Sales Corporate Management Feature 7A-2 –** Allows Employees to ***manage*** & ***execute*** **Point-of-Sales (POS)** transaction via the **Intranet Web Poral** such as: ***Search Customer Profile Information***, ***Customer Account Management***, ***Customer Registration***, ***Customer Update***, ***Customer Delete***, & ***Customer Listing functionalities, Manage & Make Reservations of a Vehicle***, ***Manage an existing Rental***, ***etc.***

  - Programming includes:

    - **Feature UI Form Requirements:** ***Design & programming*** of required ***INTRANET WEB*** *User-Interface Forms & GUI Controls* to support this feature.
    - **Feature Processing Requirements:** ***Design & programming*** of required ***INTRANET WEB TECHONLOGY***, ***Object-Oriented (OOP) Processing & Logic PROCESSING*** to support this feature.

- This feature is designed only to be used by ***CORPORATE EMPLOYEES*** in the **Corporate Offices** & ***RENTAL AGENCIES EMPLOYEES*** via the **INTRANET** using the **Web Three-Tiered Client/Server Application**.

FEATURE #7B	FEATURE #7B
	FEATURE #7B – EZRental Corporate Employee & Rental Agencies Employees INTRANET WEB PORTAL System Back-end Database Design & Implementation to support this feature: ▪ **DATABASE SERVER BACK-END SYSTEM – *BACK-END DATABASE DESIGN & FEATURES* (Create the Tables & Relationships, pre-populated tables, stored procedures, views, indexes etc.,) to support this feature.**

Feature #	Feature Description
FEATURE #8A	**FEATURE #8A**

FEATURE #8A – EZRental EXTERNAL CUSTOMER SELF-SERVICE INTERNET CUSTOMER FACING POINT-OF-SALES (POS) WEB PORTAL:

- This **Web Portal** EZRental.com, is a **Web-based Three-Tiered Client/Server physical & Software Development Architecture** used by **CUSTOMERS** to ***manage*** & ***make*** **reservations** online via a **BROWSER**.
- **BROWSER/WEB CUSTOMER POINT-OF SALES SYSTEM** – ***WEB UI FORM(S) FRONT-END APPLICATION & OOP PROGRAMMING*** **Point-of-Sales (POS) Management Features** used by ***customers*** via the **INTERNET** ***Point-of-Sales PORTAL*** via their computers/laptop/tables/Mobile to **MAKE RESERVATIONS ONLINE & MANAGE THEIR RENTAL**.
- The following are features and functionality that are required for this **WEB APPLICATION** feature:
  - **POS Reservation & Management Features:**
    - **Web POS Authentication System Feature 8A-1 –** Proper **security and authentication** must be implemented to make sure **only the authorized customer can access to its Point-0f-Sales portal and login and out of their profile website**.
    - **Web POS Customer Self-Service Management Feature 8A-2 –** Allows POS Customer (Retail Customer & Corporate Customer) Self-Service Management of their account such as: ***Customer Profile Information***, ***Customer Account & Login Registration***, ***Customer Update Profile***, ***Customer Delete Profile***, & ***Customer Listing functionalities such as listing of Reservations & Rental History etc***.
    - **Web POS Customer Self-Service Point-of-Sales Management Feature 8A-3 –** Allows POS Customer (Retail Customer & Corporate Customer) Self-Service features to make reservations and manage rentals such as: ***Make Reservations of a Vehicle***, ***Manage an existing Rental***, ***etc***.
    - **Web POS User Account Management Feature 8A-4 –** Allows POS Customer (Retail Customer & Corporate Customer) Self-Service features to enable customer to manage its **User Account** and perform the following operations: ***Reset Username*** & ***Reset Password.***
  - Programming includes:
    - **Feature UI Form Requirements:** ***Design & programming*** of required ***WEB*** *User-Interface Forms & GUI Controls* to support this feature.
    - **Feature Processing Requirements:** ***Design & programming*** of required ***WEB TECHONLOGY***, ***Object-Oriented (OOP) Processing & Logic PROCESSING*** to support this feature.
- This feature is designed only to be used by ***CUSTOMERS*** via the **INTERNET** using the **Web Three-Tiered Client/Server Application** from their **Personal Computer/Mobile Devices via the INTERNET**.

FEATURE #8B	FEATURE #8B
	FEATURE #8B – EZRental Customer Serf-Service internet Customer facing Point-of-Sales (POS) WEB PORTAL System Back-end Database Design & Implementation to support this feature: ▪ **DATABASE SERVER BACK-END SYSTEM – *BACK-END DATABASE DESIGN & FEATURES* (Create the Tables & Relationships, pre-populated tables, stored procedures, views, indexes etc.,) to support this feature.**

Feature #	Feature Description
FEATURE #9A	**FEATURE #9A** **FEATURE #9A – EZRental Customer Point-of-Sales (POS) System CUSTOMER FACING WEB PORTAL SECURITY LOGIN AUTHENTICATION SYSTEM:** ▪ Proper ***security and authentication*** must be implemented to make sure only authorized **customers** can access their **Self-Service-Point-Of-Sales Web Portal systems** when they **login** into the **Web Three-Tiered Customer Client/Server Application** via the **INTERNET**. ▪ **CUSTOMER SELF-SERVICE POINT-OF SALES SYSTEM** – ***WEB UI FORM(S) FRONT-END APPLICATION & OOP PROGRAMMING*** for **Customer Login Authentication features** used by ***customer service representative & IT Administrator employees*** to service **CUSTOMER LOGIN AUTHENTICATION SYSTEM**. ▪ The following are features and functionality that are required for this application such as: ○ **Customer Self-Service Web Portal Security Login Authentication System.** **Self-Service Web Portal Login Authentication Access for Customer features** such as: - **Customer Authentication Feature 9A-1 –** To have access to their Self-Service Web Portal Application, a customer must provide a **username & password**. This feature is required to be ***designed*** & ***programmed*** into the application. - **Customer Authentication Feature 9A-2 –** ***Design & programming*** of required **Web User-Interface Forms** & **GUI Controls** to support the **Web Portal Authentication System feature**! ○ Programming includes: - **Feature Web UI Form Requirements:** ***Design & programming*** of required ***User-Interface Forms & GUI Controls*** to support this feature. - **Feature Web Processing Requirements:** ***Design & programming*** of required ***Object-Oriented (OOP) Processing & Logic*** to support this feature. ▪ This feature is designed only to be used by customers wishing access to their **Self-Service Web Portal Three-Tiered Client/Server Application** via the **INTERNET**.
FEATURE #9B	**FEATURE #9B** **FEATURE #9B – EZRental Customer Point-of-Sales (POS) System CUSTOMER FACING WEB PORTAL SECURITY LOGIN AUTHENTICATION SYSTEM Back-end Database Design & Implementation to support this feature:** ▪ **DATABASE SERVER BACK-END SYSTEM – *BACK-END DATABASE DESIGN & FEATURES* (Create the Tables & Relationships, pre-populated tables, stored procedures, views, indexes etc.,) to support the CUSTOMER facing authentication features.**

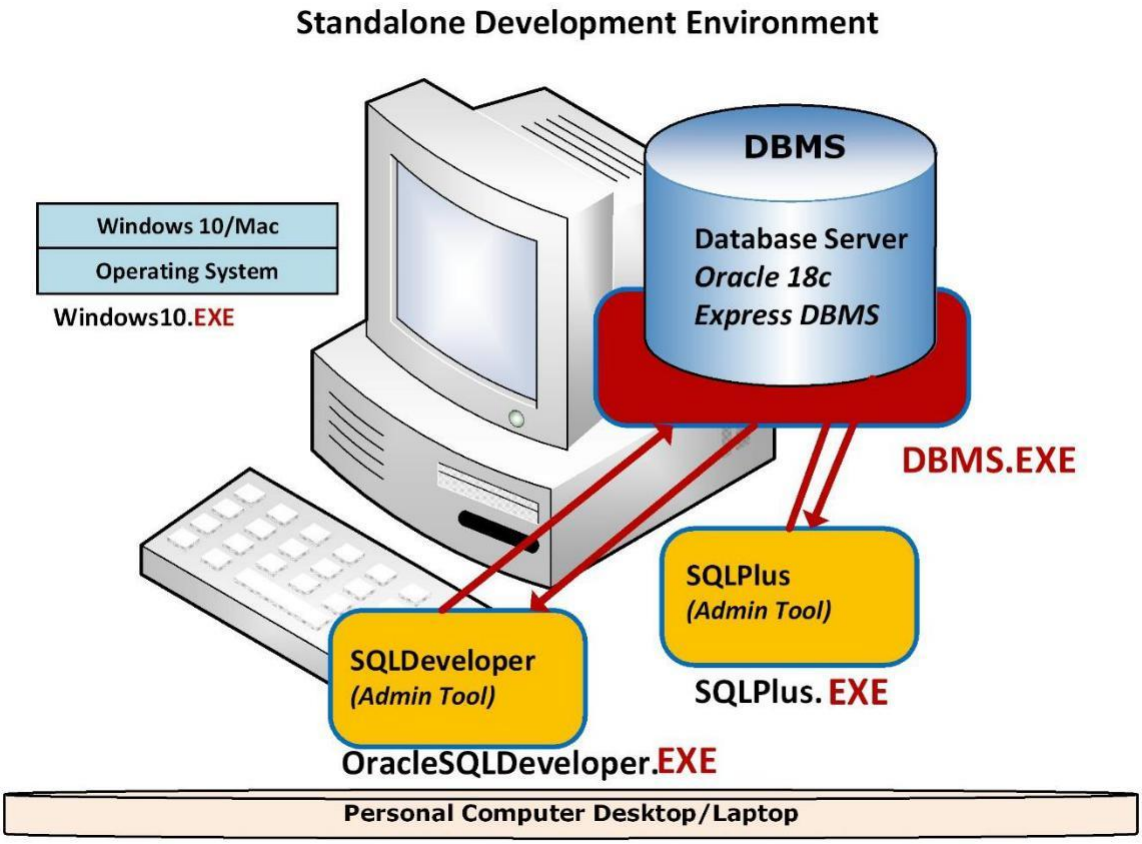
Feature #	Feature Description
FEATURE #10A	**FEATURE #10A** **FEATURE #10A – EZRental INTERNAL CORPORATE EMPLOYEE & RENTAL AGENCIES EMPLOYEES INTRANET BACK-OFFICE VEHICLE TANSPORT MANAGEMENT SYSTEM WEB PORTAL:** ▪ This **INTRANET Web Portal** EZRentalHub.com, is a **Web-based Three-Tiered Client/Server physical & Software Development Architecture** used by **CORPORATE & AGENCY EMPLOYEES** to *manage* **Transportation of Vehicles by Employee Drivers to and from Rental Agencies, Vehicle Distribution Centers, and other Locations** via a **BROWSER**. ▪ **WEB TRANSOPORT MANAGEMENT SYSTEM APPLICATION** – ***WEB UI FORM(S) FRONT-END APPLICATION & OOP PROGRAMMING*** **Transport Management Features** used by ***Vehicle Transportation Managers & Drivers Employees*** to handle the day-to-day vehicle transportation process via the **CORPORATE INTRANET** *PORTAL*. ▪ The following are features and functionality that are required for this **INTRANET Transport Management WEB APPLICATION**: o **Corporate Vehicle Transport Application Intranet Web Portal Features:** - **Transport Scheduling Feature –** handle the day-to-day creating & scheduling of a pic-up & delivery (Any vehicle type) such as: ***Creation of NEW Vehicle Transport Request***, ***Vehicle Pick-up***, ***Vehicle Drop-off*** & ***Vehicle Transport Status etc***. o Programming includes: - **Feature UI Form Requirements:** ***Design & programming*** of required ***INTRANET WEB*** *User-Interface Forms & GUI Controls* to support this feature. - **Feature Processing Requirements:** ***Design & programming*** of required ***INTRANET WEB TECHONLOGY***, ***Object-Oriented (OOP) Processing & Logic PROCESSING*** to support this feature. ▪ This feature is designed only to be used by ***CORPORATE EMPLOYEES*** in the **Corporate Offices** & ***RENTAL AGENCIES EMPLOYEES*** via the **INTRANET** using the **Web Three-Tiered Client/Server Application**.

FEATURE #10B	FEATURE #10B
	FEATURE #10B – EZRental Corporate Vehicle Transport Application Intranet Web Portal Features System Back-end Database Design & Implementation to support this feature: ▪ **DATABASE SERVER BACK-END SYSTEM – *BACK-END DATABASE DESIGN & FEATURES* (Create the Tables & Relationships, pre-populated tables, stored procedures, views, indexes etc.,) to support this feature.**

Database Management System Development Environment & Physical Architecture

- Database Tier – the Database Management System (DBMS) in scope **is Oracle Server 18c Express Edition (EX)** since this is the standard DBMS used at **EZRental Inc**.

- The objectives are to install **Oracle Server 18c Express Edition (EX)** which includes **SQL PLUS command-line admin tool** on your personal computer along with **Oracle SQL Developer graphical admin tool** to create the following database development Environment:



Project Roles & Responsibilities

- The **Business/Database Analyst hired by Mr. Rodriguez** will assemble the required database development team, and the table below describes each of the roles and the individual (s) that will execute the roles:

Prof. Rodriguez	**Program Manager, AgileScrum Master & ProjectManager**	▪ Owner of the project and liaison to Manage the **EZRental Inc**., the customer. ▪ Activities include but not limited to. 1. Owner of project responsible for the success of the project. 2. Project Management 3. Scrum Master ensures the project stays on time and moving in the right direction. Clear any obstacles impeding the team's progress etc.
Consultant #1: **Prof. Rodriguez**	**Business & DatabaseAnalyst**	▪ A **Business/Database Analyst was hired by Prof. Rodriguez** to interview the stakeholders at **EZRental Inc**. And create the ***Business Requirements*** that will be the foundation to the database design & implementation. ▪ Activities include but not limited to: 1. Engage in discovery activities & interview the stakeholders at **EZRental Inc**. 2. From the interview and discovery ***create*** **1) ER/EER Conceptual Data Model from the business requirements & 2) Normalized Logical Model**.
Consultant #2, 3, 4 & 5 **(Yuanlian Jiang)**	**Database Developers**	▪ This role uses the Normalized Logical Model created by consultant #2 to create the Data Dictionary, Physical Schema Diagram, and Implement the Database Application forthe Auto Rental System. ▪ Activities include but not limited to: 1. Use the ***Normalized Logical Model*** created by consultant #2 to do the **following:** 1) **Create Data Dictionary tables for eachlogical table targeting MS SQL Server and Oracle18c 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 #6 **(Yuanlian Jiang)**	**Database Administrator**	▪ The DB Admin, install the DBMS, maintain, and operate the DBMS throughoutits lifetime. ▪ Activities include but not limited to: 1. As DB Admin, you are to **1) Setup & install MS SQL Server and Oracle18c for DBMS.** **2) Administrative tools for target DBMS**. 2. Also, as DB Admin, you are to **3) Operate & Maintain the DBMS**.

Summary of the Windows Client & Browser Client Applications Development Project Roles and Responsibilities

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

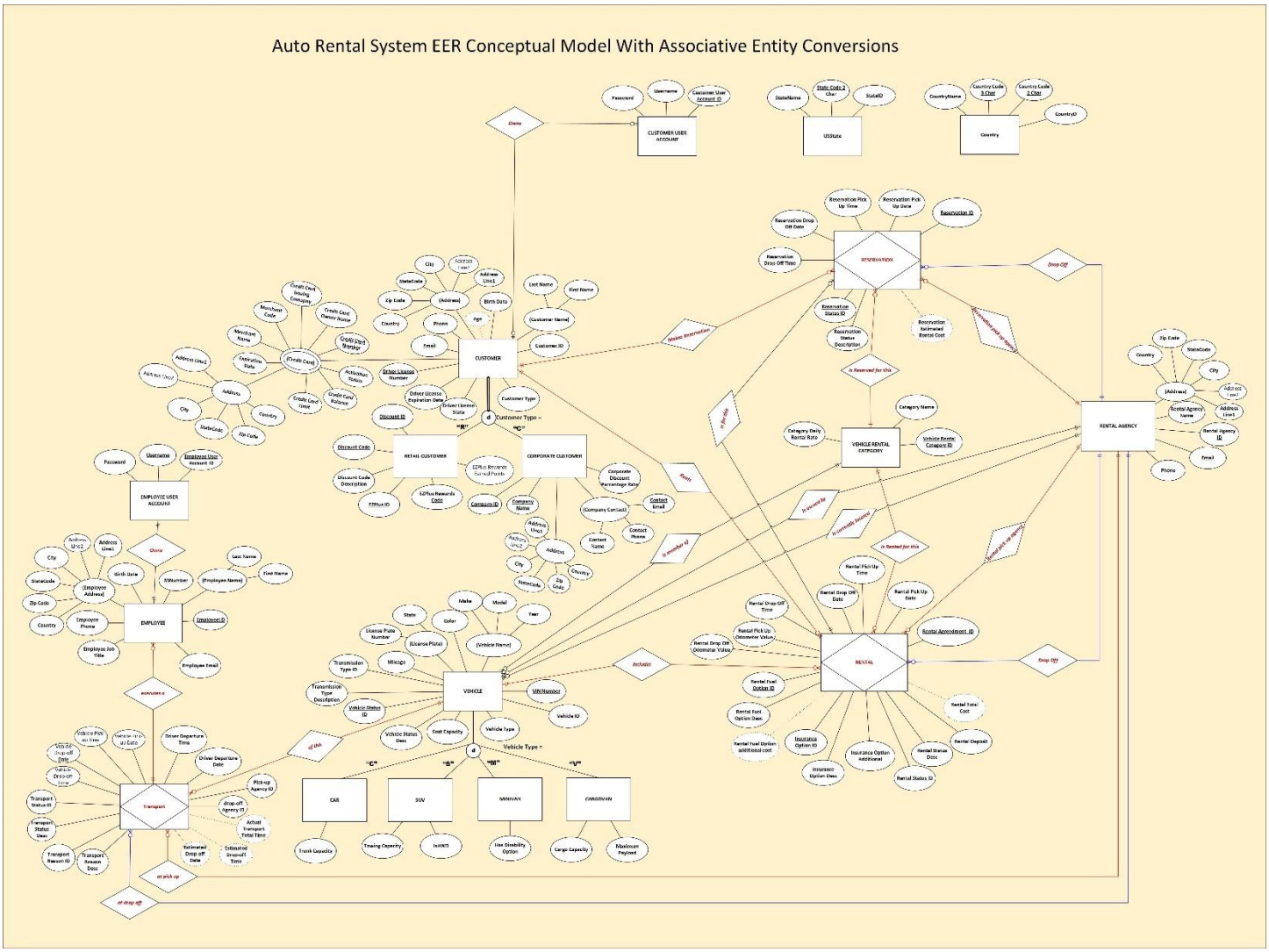
Person	Role	Description
Consultant #7 & 13 **Mr. Rodriguez**	**Full Stack Object-Oriented-Programming Architect**	▪ An Object-Oriented-Programming Architect was hired by Prof. Rodriguez to interview the stakeholders at **EZRental Inc**. and derive the **Application Technical Requirements** in addition to designing the **Class/Object Model Architecture**. This also includes the planning and designing both **Windows Client Application** and the **Web Browser Application**. ▪ Activities include but not limited to: 1. Engage in discovery activities & interview the stakeholders at **EZRental Inc**. 2. From the interview and discovery **1) Design/Architect the Object-Oriented-Programming Class/Object Model for the Windows Client Application**. 3. **Design/Architect the Object-Oriented-Programming Class/Object Model for the Web Browser Application**.
Consultants #8, 9, 10, 11 & 12	**Full Stack Windows Application Developers & UI/UX Client Application Developer**	▪ Object-Oriented-Programming developer to implement the **Windows Client Application** using **C# & .NET technologies** & on the database side, implement stored procedures and support the databased team as needed. ▪ Activities include but not limited to: 1. As full stack developer, **Programming & implementation of theObject-Oriented-Programming of Class/Object Model designed by consultant #7 for the Windows Client Application using C# & .NET Technologies**. 2. In addition, **Development of Database Stored Procedures,and other development requirements in the Back-end DBMS**. 3. From the technical requirements, **design a high-level GraphicalUser-Interface (GUID) wireframe**, & **implement the front-end UI Programming, features & functionality**
Consultant #14, 15, 16, 17 & 18	**Full Stack Web Application Developer & UI/UX Web Application Developer**	▪ Object-Oriented-Programming developer to implement the **Web Browser Application** using **C# & ASP.NET technologies**. ▪ Activities include but not limited to: 1. As full stack developer, **Programming & implementation of theObject-Oriented-Programming of Class/Object Model designed by consultant #7 for the Web Browser Client Application using C# & ASP.NET Technologies**. 2. From the technical requirements, **design a high-level Graphical User-Interface (GUID) wireframe**, & **implement the Webfront-end UI Programming, features & functionality in the Web Server Application**

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

- **DIAGRAM #1 –** A **Database Analyst/Architect** was hired by your sponsor Mr. Rodriguez, who derived the **EER Conceptual Model** for the **AUTO MANAGEMENT SYSTEM**, based on the ***Application Business Requirements*** as part of the **Analysis Phase** **deliverables**.

- This **EER Conceptual Model Diagram** is the **foundation** of the ***Database Design*** for the **DBMS Auto Rental Management System Application**. Its goal is to take the ***entities*** and ***relationships*** identified in the ***Application Business Requirements*** and connect them together to build a DIAGRAM or high-level picture of how the ***Application key Business data*** relate to each other.

- Below is the **Database Analyst/Architect** rendition of the **EER Model of the Business Requirements** **for Auto Rental Management System** 1, with standard CHEN notation and using Associative entities.



Database Design Deliverable #3 – Normalized Logical Model Diagram

- The **Database Analyst/Architect** hired by your sponsor Mr. Rodriguez, Also ***derived & normalized*** the **Normalized Logical Model Diagram** for the **AUTO MANAGEMENT SYSTEM**, from the **EER Conceptual Model Diagram** as part of one of the **Design Phase** **deliverables**.

- This **Normalized Logical Model Diagram** is the **2nd foundation** of the ***Database Design*** for the **DBMS Auto Rental Management System Application**. Its goal is to take the **EER Conceptual Model Diagram**, which focuses on the ***entities*** and ***relationships*** of the required ***business data*** to the **actual database TABLES** and their **RELATIONSHIPS** that will be ***IMPLEMENTED*** in a database. In other words, it shows how the **TABLES** and **RELATIONSHIPS** will look like in a **Database (DBMS) when you implement it**. This is known as the **SCHEMA**!

- We will break up the into 3 diagrams for the project, based on our deployment strategy and methodology:

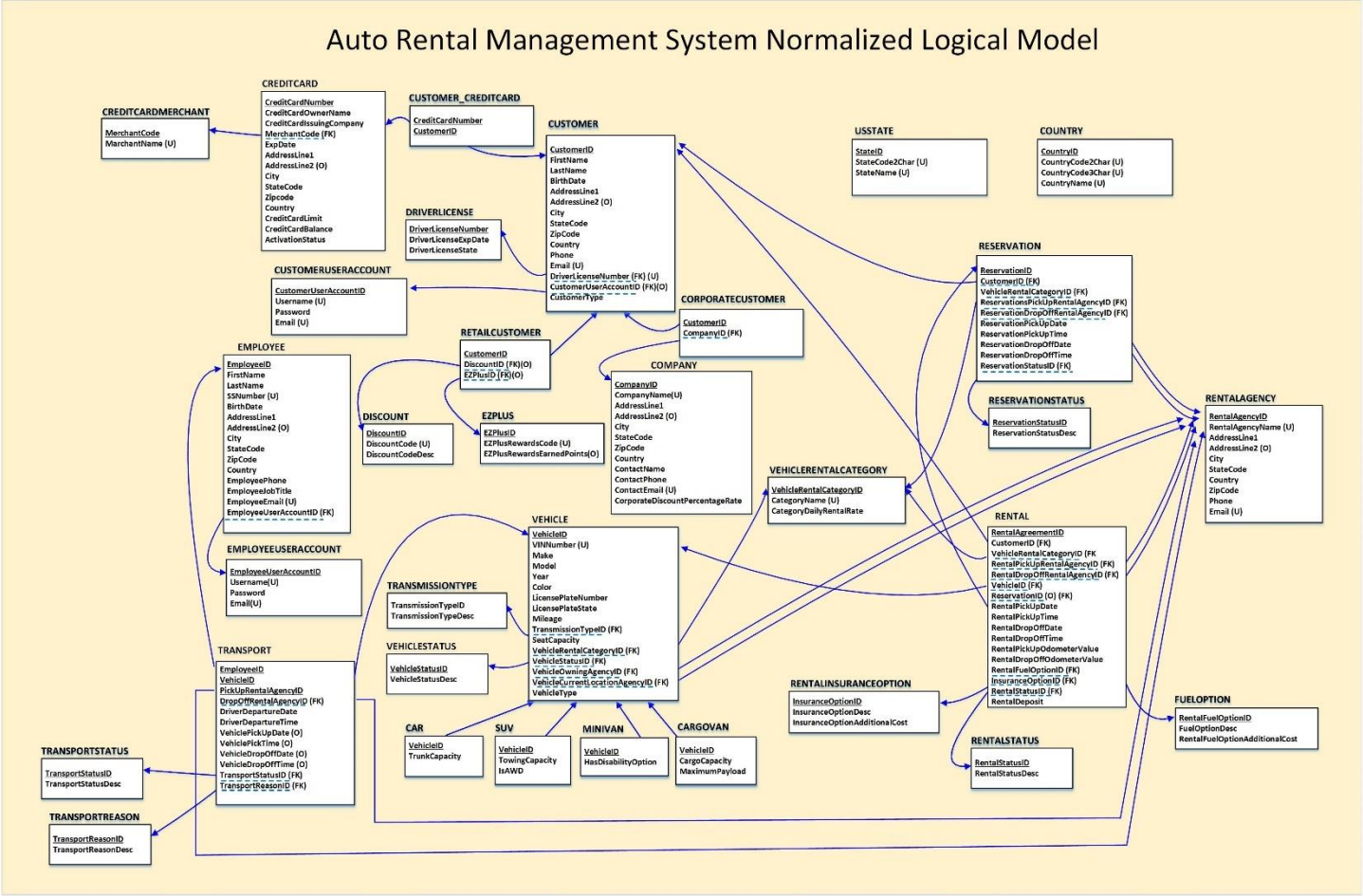
1) **DIAGRAM #1 –** The complete **Normalized Logical Model Diagram** for the **AUTO MANAGEMENT SYSTEM** deri ***derived*** from the **EER Conceptual Model Diagram**.

2) **DIAGRAM #2 –** The limited **PILOT Normalized Logical Model Diagram** that includes only **13 tables only** from the **full normalized model**. This will be the first production deployment rollout to test the application on a limited number of users in different locations to flush out any unforeseen errors in the **AUTO MANAGEMENT SYSTEM**.

3) **DIAGRAM #3 –** A limited **PROOF-OF-CONCEPT PROTOTYPE** portion of **Normalized Logical Model Diagram** that includes only **2 tables only** from the full normalized model. This will be used for a PROTOTYPE version of the **AUTO MANAGEMENT SYSTEM** to prove out the concept.

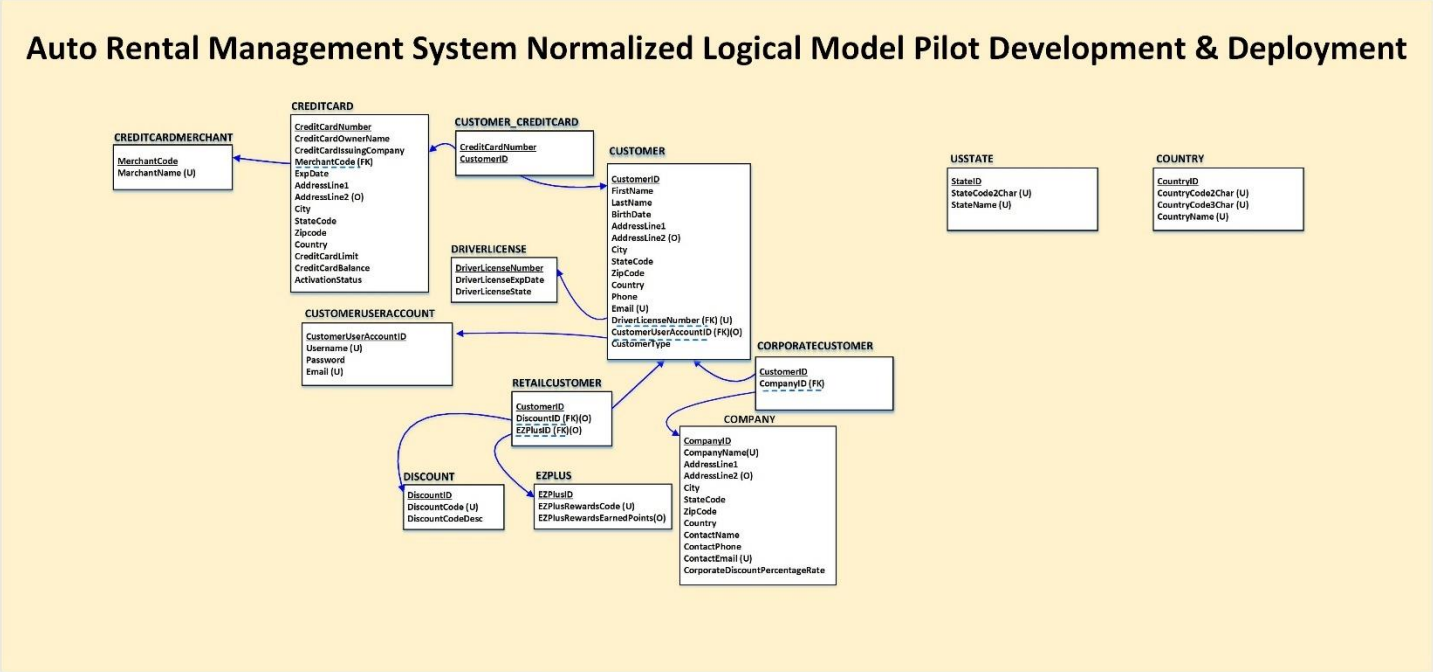
Production Normalized Logical Model

- Below is the architect's rendition of the **Normalized Logical Model** based on **EER diagram** in previous section.



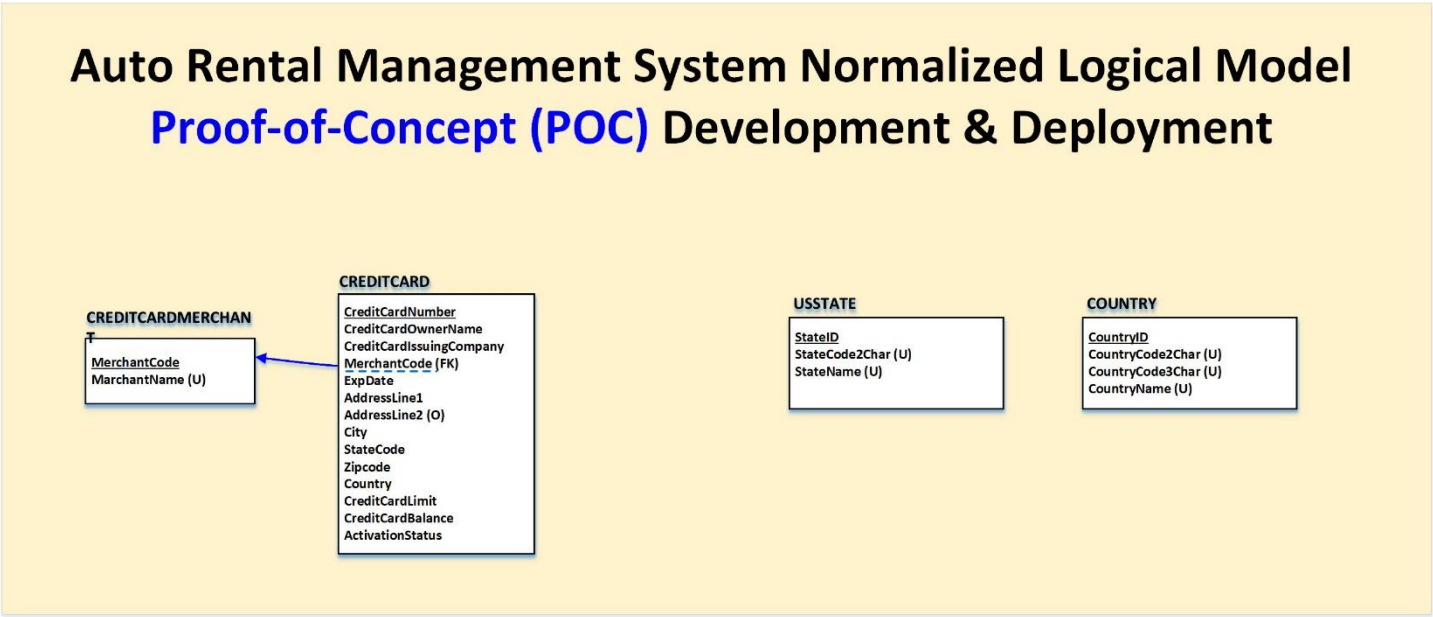
Normalized Logical Model Limited Pilot

- Below is the **LIMITED PILOT Normalized Logical Model** that will be deploy to a limited number of users throughout the organization as a final production test. This diagram only contains the **13 logical tables** in scope of the **PILOT**:



Normalized Logical Model PROOF-OF-CONCEPT PROTOTYTYPE

- Below is the **PROOF-OF-CONCEPT (POC) Normalized Logical Model** that will be used to create a PROTOTYPE of the application to demo to the business. This diagram only contains the **4 logical tables** in scope of the **POC**:



Database Design Deliverable #4 – Physical Model Data Dictionary

1. As stated in the previous ***sub-section***, the **PHYSICAL MODEL DESIGN PHASE** is divided into the following 3 components:

    a) The **Normalized Logical Model Diagram**.
    b) The **Physical Model Data Dictionary**.
    c) The **Physical Model Schema Design Diagram**.
    d) **Technical Specifications for performance, efficiency, data integrity, security, disaster recovery etc**.

2. This ***sub-section*** describes the **FOURTH DATABASE DESIGN DELIVERABLE** the **Physical Model Data Dictionary**. The **Physical Model Data Dictionary** is the ***Second Component*** of our ***Waterfall Methodology*** **DESIGN PHASE PHYSICAL MODEL DESIGN!**

3. The 13 Data Dictionary Tabular Listings in Scope
    - Below are the 13 Data Dictionary Tabular tables.

#1:

CREDITCARD							
Column Num.	**Attribute/Column Name**	**Generic Data Type Name**	**Oracle SERVER Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
1.	**CreditCardNumber**	String	VARCHAR2(16)	Y	16	PRIMARY KEY	Credit Card Number of customer. This PRIMARY KEY has business meaning.
2.	**CreditCardOwnerName**	String	VARCHAR2(50)	Y	50	NOT NULL	Owner name of the Credit Card.
3.	**CreditCardIssuingCompany**	String	VARCHAR2(50)	Y	50	NOT NULL	Credit card issuing company name.
4.	**MerchantCode**	Number	NUMBER(2)	Y	Default size of the data type	NOT NULL **CHECK(MerchantCode between 1 and 20)**	A Digit Code that associated with Merchant Name.
5.	**ExpDate**	Date	DATE	Y	MM/DD/YY	NOT NULL	Expiration Date.
6.	**AddressLine1**	String	VARCHAR2(50)	Y	50	NOT NULL	House number & Street part 1.
7.	**AddressLine2**	String	VARCHAR2(50)	N	50	NULL	House number & Street part 2 (OPTIONAL).
8.	**City**	String	VARCHAR2(30)	Y	30	NOT NULL	City name.
9.	**StateCode**	Character	CHAR(2)	Y	2	NOT NULL	Two-character code for a state in the US.
10.	**Zipcode**	String	VARCHAR2(10)	Y	10	NOT NULL	US Zip Code.
11.	**Country**	String	VARCHAR2(100)	Y	100	NOT NULL	Country column with international scope.
12.	**CreditCardLimit**	Number	NUMBER(8, 2)	Y	X=8 Y=2	NOT NULL	The maximum amount a customer can charge on their credit card.
13.	**CreditCardBalance**	Number	NUMBER(8, 2)	Y	X=8 Y=2	NOT NULL	The balance remaining on the credit card after you have made charges and owe money on the credit card.
14.	**ActivationStatus**	Number	NUMBER(1)	Y	1	NOT NULL **CHECK(ActivationStatus IN ('0', '1'))**	Is the credit card active? (1 for active or 0 for not active.)

#2:

CREDITCARDMERCHANT							
Column Num.	**Attribute/Column Name**	**Generic Data Type Name**	**Oracle SERVER Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
1.	**MerchantCode**	Number	NUMBER(2)	Y	2	PRIMARY KEY **CHECK(MerchantCode between 1 and 20)**	A Digit Code that associated with Merchant Name.
2.	**MerchantName**	String	VARCHAR2(50)	Y	50	UNIQUE NOT NULL	Merchant Name.

#3:

CUSTOMER							
Column Num.	**Attribute/Column Name**	**Generic Data Type Name**	**Oracle SERVER Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
1.	**CustomerID**	Number	NUMBER(10) GENERATED ALWAYS AS IDENTITY	Y	10	PRIMARY KEY	CustomerID stores an auto-generated INTERGER NUMBER IDENTITY primary key starting at 1 to 9,999,999,999. This PRIMARY KEY has no business meaning.
2.	**FirstName**	String	VARCHAR2(50)	Y	50	NOT NULL	First Name of customer.
3.	**LastName**	String	VARCHAR2(50)	Y	50	NOT NULL	Last Name of customer.
4.	**BirthDate**	Date	DATE	Y	MM/DD/YY	NOT NULL	Date of Birth.
5.	**AddressLine1**	String	VARCHAR2(50)	Y	50	NOT NULL	House number & Street part 1.
6.	**AddressLine2**	String	VARCHAR2(50)	N	50	NULL	House number & Street part 2.
7.	**City**	String	VARCHAR2(30)	Y	30	NOT NULL	City name.
8.	**StateCode**	Character	CHAR(2)	Y	2	NOT NULL	Two-character code for a state in the US.
9.	**ZipCode**	String	VARCHAR2(10)	Y	10	NOT NULL	US Zip Code.
10.	**Country**	String	VARCHAR2(100)	Y	100	NOT NULL	Country column with international scope.
11.	**Phone**	String	VARCHAR2(20)	Y	20	NOT NULL	Phone – scope is international.
12.	**Email**	String	VARCHAR2(100)	Y	100	UNIQUE NOT NULL	Email – Considering international scope. (Unique and required)
13.	**DriverLicenseNumber**	String	VARCHAR2(25)	Y	25	UNIQUE NOT NULL	Driver License Number.
14.	**CustomerUserAccountID**	Binary	RAW(16)	N	16	NULL	Customer User Account ID.
15.	**CustomerType**	Character	CHAR(1)	Y	1	NOT NULL	Customer type. (“R” for Retail or “C” for Corporate)

#4:

CUSTOMER_CREDITCARD							
Column Num.	**Attribute/Column Name**	**Generic Data Type Name**	**Oracle SERVER Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
3.	**CreditCardNumber**	String	VARCHAR2(16)	Y	16	PRIMARY KEY	Derived from CREDITCARD table. Composited key of the CUSTOMER_CREDITCARD table.
4.	**CustomerID**	Number	NUMBER(10)	Y	10	PRIMARY KEY	Derived from CUSTOMER table. Composited key of the CUSTOMER_CREDITCARD table.

#5:

RETAILCUSTOMER							
Column Num	**Attribute/Column Name**	**Generic Data Type Name**	**Oracle SERVER Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
1.	**CustomerID**	Number	NUMBER(10)	Y	10	PRIMARY KEY	Derived from CUSTOMER table.
2.	**DiscountID**	Number	NUMBER	N	Default size of data type	NULL	Derived from DISCOUNT table.
3.	**EZPlusID**	Number	NUMBER(10)	N	10	NULL	Derived from EZPLUS table.

#6:

DISCOUNT							
Column Num	**Attribute/Column Name**	**Generic Data Type Name**	**Oracle SERVER Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
1.	**DiscountID**	Number	NUMBER GENERATED ALWAYS AS IDENTITY	Y	Default size of data type	PRIMARY KEY	Discount ID stores an auto-generated INTERGER NUMBER IDENTITY. This PRIMARY KEY has no business meaning.
2.	**DiscountCode**	Character	CHAR(10)	Y	10	UNIQUE NOT NULL	Coupon code that 10-characters long.
3.	**DiscountCodeDesc**	String	VARCHAR2(200)	Y	200	NOT NULL	Discount code description.

#7:

EZPLUS							
Column Num	**Attribute/Column Name**	**Generic Data Type Name**	**Oracle SERVER Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
1.	**EZPlusID**	Number	NUMBER(10) GENERATED ALWAYS AS IDENTITY	Y	10	PRIMARY KEY	EZPlusID generate automatically. This PRIMARY KEY has no business meaning.
2.	**EZPlusRewardsCode**	Charater	CHAR(13)	Y	15	UNIQUE NOT NULL	Ezplus rewards code
3.	**EZPlusRewardsEarnedPoints**	Number	NUMBER(6)	N	6	NULL	The number of rewards points earned that accumulated after all the rentals and can be used to save on future rentals.

#8:

COPRORATECUSTOMER							
Column Num	**Attribute/Column Name**	**Generic Data Type Name**	**Oracle SERVER Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
1.	**CustomerID**	Number	NUMBER(10)	Y	5	PRIMARY KEY	**CustomerID**. This PRIMARY KEY has no business meaning.
2.	**CompanyID**	Number	NUMBER(5)	Y	5	NOT NULL **CHECK(CompanyID between 1 and 20000)**	CompanyID.

#9:

COMPANY							
Column Num.	**Attribute/Column Name**	**Generic Data Type Name**	**Oracle SERVER Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
1.	**CompanyID**	Number	NUMBER(5)	Y	5	PRIMARY KEY **CHECK(CompanyID between 1 and 20000)**	Unique identifier for a company instance. This PRIMARY KEY has business meaning.
2.	**CompanyName**	String	VARCHAR2(50)	Y	50	UNIQUE NOT NULL	Company Name.
3.	**AddressLine1**	String	VARCHAR2(50)	Y	50	NOT NULL	House number & Street part 1.
4.	**AddressLine2**	String	VARCHAR2(50)	N	50	NULL	House number & Street part 2.
5.	**City**	String	VARCHAR2(30)	Y	30	NOT NULL	City name.
6.	**StateCode**	Character	CHAR(2)	Y	2	NOT NULL	US state code.
7.	**ZipCode**	String	VARCHAR2(10)	Y	10	NOT NULL	US zip code.
8.	**Country**	String	VARCHAR2(100)	Y	100	NOT NULL	Country column with international scope.
9.	**CompanyRepName**	String	VARCHAR2(50)	Y	50	NOT NULL	Company representative name.
10.	**ContactPhone**	String	VARCHAR2(20)	Y	20	NOT NULL	Phone number– scope is international.
11.	**ContactEmail**	String	VARCHAR2(100)	Y	100	UNIQUE NOT NULL	Email – Considering international scope.
12.	**CorporateDiscountPercentageRate**	Number	Number(3,2)	Y	X=3 Y=2	NOT NULL	Corporate Discount Percentage Rate.

#10:

DRIVERLICENSE							
Column Num.	**Attribute/Column Name**	**Generic Data Type Name**	**Oracle SERVER Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
1.	**DriverLicenseNumber**	String	VARCHAR2(25)	Y	25	PRIMARY KEY	The driver license number is used throughout the business to identify a customer for searching, reporting etc. It is the unique ID for a customer to be identified and managed from a business perspective.
2.	**DriverLicenseExpDate**	Date	DATE	Y	MM/DD/YY	NOT NULL	Driver license expiration date.
3.	**DriverLicenseState**	Character	CHAR(2)	Y	2	NOT NULL	Driver license state.

#11:

CUSTOMERUSERACCOUNT							
Column Num.	**Attribute/Column Name**	**Generic Data Type Name**	**Oracle SERVER Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
1.	**CustomerUserAccountID**	Binary	RAW(16)	Y	DEFAULT SYS_GUID()	PRIMARY KEY	The primary key CustomerUserAccountID is targeted to STORE a GLOBAL UNIQUE IDENTIFIER (GUID) string generated by the DATABASE.
2.	**Username**	String	VARCHAR2(50)	Y	50	UNIQUE NOT NULL	Username.
3.	**Password**	String	VARCHAR2(50)	Y	50	NOT NULL	Password.
4.	**Email**	String	VARCHAR(100)	Y	100	UNIQUE NOT NULL	Email Address.

\

#12:

USSTATE							
Column Num.	**Attribute/Column Name**	**Generic Data Type Name**	**Oracle SERVER Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
1.	**StateID**	Number	NUMBER(2)	Y	2	PRIMARY KEY **CHECK(StateID between 1 and 75)**	State ID.
2.	**StateCode2Char**	Character	CHAR(2)	Y	2	UNIQUE NOT NULL	US State Code.
3.	**StateName**	String	VARCHAR2(50)	Y	50	UNIQUE NOT NULL	US State Name.

#13:

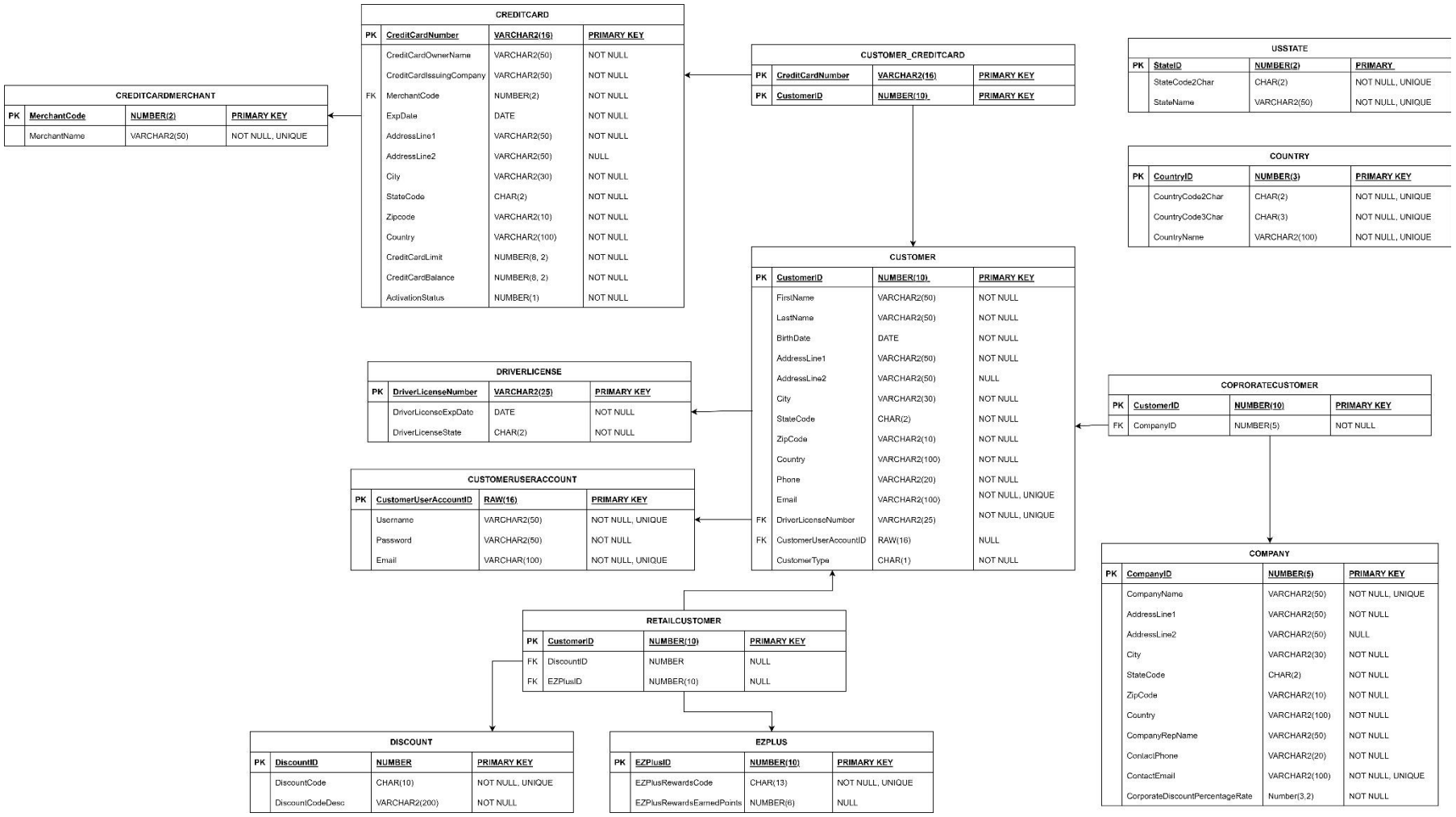
COUNTRY							
Column Num.	**Attribute/Column Name**	**Generic Data Type Name**	**Oracle SERVER Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
1.	**CountryID**	Number	NUMBER(3)	Y	3	PRIMARY KEY **CHECK(CountryID between 1 and 250)**	Country ID.
2.	**CountryCode2Char**	Character	CHAR(2)	Y	2	UNIQUE NOT NULL	Country Code with 2 character.
3.	**CountryCode3Char**	Character	CHAR(3)	Y	3	UNIQUE NOT NULL	Country Code with 3 character.
4.	**CountryName**	String	VARCHAR2(100)	Y	100	UNIQUE NOT NULL	Country Name.

Database Design Deliverable #5 – Physical Model Schema Design Diagram

This ***sub-section*** describes the **FIFTH DATABASE DESIGN DELIVERABLE** is the **Physical Model Data Dictionary** which is the ***third Component*** of our ***Waterfall Methodology*** **DESIGN PHASE PHYSICAL MODEL DESIGN!** This is a very important deliverable whose purpose and creation are as follows:

a) **This diagram is created by combining the Normalized Logical Model Diagram with the Physical Model Data Dictionary to create a NEW DIAGRAM called the Physical Model Schema Design Diagram**. ***(If interested in learning more in the future, see CST3504 Database Design Course Lectures to learn the steps for this process***.

b) The **Physical Model Schema Design Diagram is the DIAGRAM USED TO IMPLEMENT THE DATABASE in the DBMS APPLICATION**. **THIS DIAGRAM**



Database Implementation Deliverable #6 – Development & Implementation

This ***sub-section*** describes the **SIXTH DATABASE DELIVERABLE** of the database design and implementation process, which is the **Database Implementation Phase or implementing the Database**.

This ***sub-section*** is to ***IMPLEMENT/CREATE*** the ***DATABASE APPLICATION*** using **Database Design Deliverable #4 – Physical Model Schema Design Diagram** in addition to **ORACLE DBMS CREATE TABLE STATEMENTS** to ***CREATE*** every **TABLE & RELATIONSHIP** listed in the **Physical Model Schema Design Diagram**.

- The **Development & Implementation** is to develop/implement the design that is finalized with the **Physical Schema Design Diagram**. In this section is to ***IMPLEMENT*** that **Database Management System (DBMS) Application** using **Oracle Server DBMS**.

Below is the code implemented in Oracle Server ARMDB(Auto Rental Management Database):

```
CREATE TABLE DriverLincense
(
        DriverLicenseNumber  VARCHAR(25)   PRIMARY KEY,
        DriverLicenseExpDate DATE          NOT NULL,
        DriverLicenseState   CHAR(2)       NOT NULL
);

CREATE TABLE CustomerUserAccount
(
        CustomerUserAccountID       RAW(16)       DEFAULT SYS_GUID()   PRIMARY KEY,
        Username                    VARCHAR2(50)  UNIQUE       NOT NULL,
        Password                    VARCHAR2(50)  NOT NULL,
        Email                       VARCHAR(100)  UNIQUE       NOT NULL
);

CREATE TABLE Customer
(
        CustomerID              NUMBER(10) GENERATED ALWAYS AS IDENTITY     NOT NULL,
        FirstName               VARCHAR2(50)                                NOT NULL,
        LastName                VARCHAR2(50)                                NOT NULL,
        BirthDate               DATE                                        NOT NULL,
        AddressLine1            VARCHAR2(50)                                NOT NULL,
        AddressLine2            VARCHAR2(50)                                NULL,
        City                    VARCHAR2(30)                                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)                               UNIQUE NOT NULL,
        DriverLicenseNumber     VARCHAR2(25)                                UNIQUE NOT NULL,
        CustomerUserAccountID   RAW(16)                                     NULL,
        CustomerType            CHAR(1)                                     NOT NULL,

        CONSTRAINT pk_CustomerID PRIMARY KEY (CustomerID),
```

```
        CONSTRAINT fk_DriverLicenseNumber
        FOREIGN KEY(DriverLicenseNumber)
        REFERENCES DriverLincense(DriverLicenseNumber)
        ON DELETE CASCADE,

        CONSTRAINT fk_CustomerUserAccountID
        FOREIGN KEY(CustomerUserAccountID)
        REFERENCES CustomerUserAccount(CustomerUserAccountID)
        ON DELETE CASCADE

);

CREATE TABLE CreditCardMerchant
(
        MerchantCode        NUMBER(2)     CHECK(MerchantCode>=1 AND MerchantCode<=20)    NOT NULL,
        MerchantName        VARCHAR2(50)                                                 UNIQUE        NOT NULL,

        CONSTRAINT pk_MerchantCode PRIMARY KEY (MerchantCode)
);

CREATE TABLE CreditCard
(
        CreditCardNumber            VARCHAR2(16)        PRIMARY KEY,
        CreditCardOwnerName         VARCHAR2(50)        NOT NULL,
        CreditCardIssuingCompany    VARCHAR2(50)        NOT NULL,
        MerchantCode        NUMBER(2)     CHECK(MerchantCode>=1 AND MerchantCode<=20)    NOT NULL,
        ExpDate                     DATE                NOT NULL,
        AddressLine1                VARCHAR2(50)        NOT NULL,
        AddressLine2                VARCHAR2(50)        NULL,
        City                        VARCHAR2(30)        NOT NULL,
        StateCode                   CHAR(2)             NOT NULL,
        Zipcode                     VARCHAR2(10)        NOT NULL,
        Country                     VARCHAR2(100)               NOT NULL,
        CreditCardLimit             NUMBER(8, 2)        NOT NULL,
        CreditCardBalance           NUMBER(8, 2)        NOT NULL,
        ActivationStatus            NUMBER(1)     CHECK(ActivationStatus IN ('0', '1'))    NOT NULL,

        CONSTRAINT fk_MerchantCode
        FOREIGN KEY(MerchantCode)
        REFERENCES CreditCardMerchant(MerchantCode)
        ON DELETE CASCADE
);

CREATE TABLE Customer_CreditCard
(
        CreditCardNumber            VARCHAR2(16)        NOT NULL,
        CustomerID                  NUMBER(10)          NOT NULL,

        CONSTRAINT pk_CreditCardNumber_CustomerID PRIMARY KEY (CreditCardNumber, CustomerID),
```

```
        CONSTRAINT fk_CreditCardNumber
        FOREIGN KEY(CreditCardNumber)
        REFERENCES CreditCard(CreditCardNumber)
        ON DELETE CASCADE,

        CONSTRAINT fk_CustomerID
        FOREIGN KEY(CustomerID)
        REFERENCES Customer(CustomerID)
        ON DELETE CASCADE
);

CREATE TABLE Discount
(
        DiscountID          NUMBER GENERATED ALWAYS AS IDENTITY   NOT NULL,
        DiscountCode        CHAR(10)                              UNIQUE       NOT NULL,
        DiscountCodeDesc    VARCHAR2(200)                         NOT NULL,

        CONSTRAINT pk_DiscountID PRIMARY KEY (DiscountID)
);

CREATE TABLE EZPlus
(
        EZPlusID                    NUMBER(10) GENERATED ALWAYS AS IDENTITY          NOT NULL,
        EZPlusRewardsCode           CHAR(13)                                UNIQUE   NOT NULL,
        EZPlusRewardsEarnedPoints   NUMBER(6)                                        NULL,

        CONSTRAINT pk_EZPlusID PRIMARY KEY (EZPlusID)
);

CREATE TABLE RetailCustomer
(
        CustomerID      NUMBER(10)      PRIMARY KEY,
        DiscountID      NUMBER          NULL,
        EZPlusID        NUMBER(10)      NULL,

        CONSTRAINT fk_CustomerID_RetailCustomer
        FOREIGN KEY(CustomerID)
        REFERENCES Customer(CustomerID)
        ON DELETE CASCADE,

        CONSTRAINT fk_DiscountID
        FOREIGN KEY(DiscountID)
        REFERENCES Discount(DiscountID)
        ON DELETE CASCADE,

        CONSTRAINT fk_EZPlusID
        FOREIGN KEY(EZPlusID)
        REFERENCES EZPlus(EZPlusID)
```

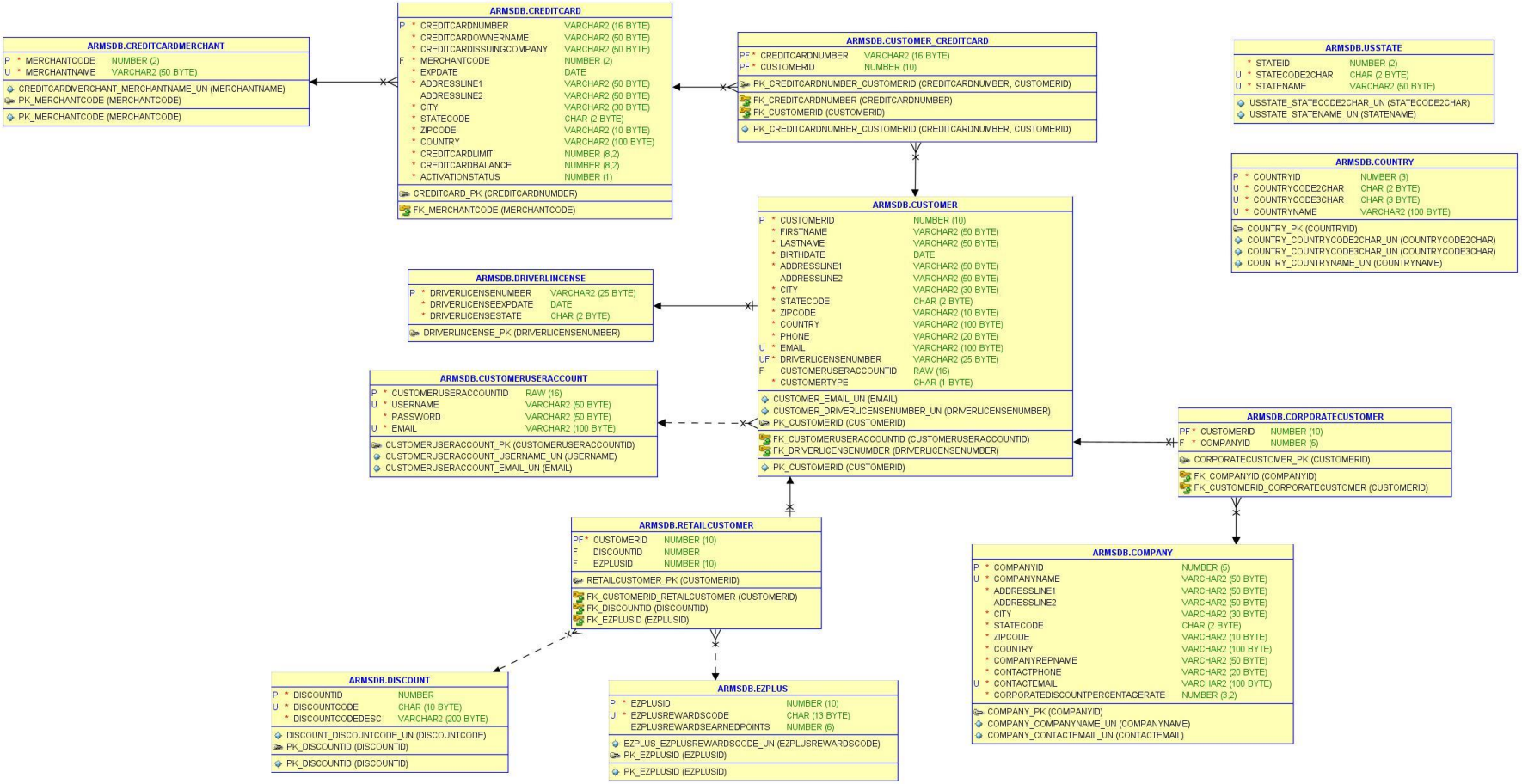
```
        ON DELETE CASCADE
);

CREATE TABLE Company
(
        CompanyID               NUMBER(5)       CHECK(CompanyID between 1 and 20000)            PRIMARY KEY,
        CompanyName             VARCHAR2(50)                    UNIQUE          NOT NULL,
        AddressLine1            VARCHAR2(50)                    NOT NULL,
        AddressLine2            VARCHAR2(50)                    NULL,
        City                    VARCHAR2(30)                    NOT NULL,
        StateCode               CHAR(2)                         NOT NULL,
        ZipCode                 VARCHAR2(10)                    NOT NULL,
        Country                 VARCHAR2(100)                   NOT NULL,
        CompanyRepName          VARCHAR2(50)                    NOT NULL,
        ContactPhone            VARCHAR2(20)                    NOT NULL,
        ContactEmail            VARCHAR2(100)                   UNIQUE          NOT NULL,
        CorporateDiscountPercentageRate     Number(3,2)     NOT NULL
);

CREATE TABLE CorporateCustomer
(
        CustomerID      NUMBER(10)      PRIMARY KEY,
        CompanyID       NUMBER(5)       CHECK(CompanyID between 1 and 20000)        NOT NULL,

        CONSTRAINT fk_CustomerID_CorporateCustomer
        FOREIGN KEY(CustomerID)
        REFERENCES Customer(CustomerID)
        ON DELETE CASCADE,

        CONSTRAINT fk_CompanyID
        FOREIGN KEY(CompanyID)
        REFERENCES Company(CompanyID)
        ON DELETE CASCADE
);

CREATE TABLE USState
(
        StateID         NUMBER(2)       CHECK(StateID between 1 and 75)             NOT NULL,
        StateCode2Char  CHAR(2)                 UNIQUE          NOT NULL,
        StateName       VARCHAR2(50) UNIQUE             NOT NULL
);

CREATE TABLE Country
(
        CountryID               NUMBER(3)       CHECK(CountryID between 1 and 250) PRIMARY KEY,
        CountryCode2Char        CHAR(2)                 UNIQUE          NOT NULL,
        CountryCode3Char        CHAR(3)                 UNIQUE          NOT NULL,
        CountryName             VARCHAR2(100)           UNIQUE          NOT NULL
);
```

Database Implementation Deliverable #7 – Implemented Physical Schema Diagram

This ***sub-section*** describes the **SEVENTH DATABASE DELIVERABLE** of the database design and implementation process, which to ***GENERATE/EXPORT*** **the Implemented PHYSICAL SCHEMA DIAGRAM from the Database**. This **Implemented Physical Schema Diagram** is an ***EXPORTED*** **DIAGRAM of every TABLE & RELATIONSHIP that was *CREATE* using the ORACLE DBMS CREATE TABLE STATEMENTS**.

The **Implemented Physical Schema Diagram** is to ***Comparing & Validating*** the **Physical Schema Design Diagram** with the ***GENERATED*** **Physical Model Schema Diagram**.

Below is the **Physical Model Schema Diagram** generated by the ARMDB database using Oracle Server.



Database Implementation Deliverable #8 – Database Validation Testing

The section Database Validation Unit Testing is the Database Developer Executing SQL Data Manipulation Language (DML) statements such as SELECT, INSERT, UPDATE & DELETE to test the Physical Schema IMPLEMENTED.

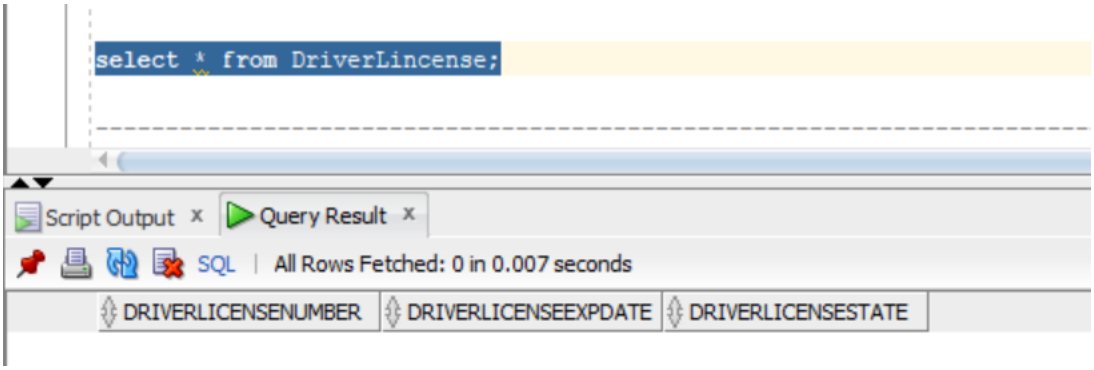
The following is populating the 13 tables:

1. **Populating The DriverLincense Table**
   This section shows the content of DriverLincense Table before and after populating it with records using an SQL INSERT STATEMENTS.

- **Table Before Execution of SQL INSERT STATEMENTS:**
  The following SQL SELECT Statement was used to list the content of the DriverLincense table:

```
SELECT * FROM DriverLincense;
```

The current state of the DriverLincense Table is:



- **List of SQL INSERT STATEMENTS used to populate the table:**
  The following SQL INSERT Statements were used to INSERT records to the DriverLincense table:

```
INSERT INTO DriverLincense(DriverLicenseNumber, DriverLicenseExpDate, DriverLicenseState)
VALUES ('151878811', '03-DEC-2025', 'NY');
INSERT INTO DriverLincense(DriverLicenseNumber, DriverLicenseExpDate, DriverLicenseState)
VALUES ('851878812', '05-DEC-2025', 'NY');
INSERT INTO DriverLincense(DriverLicenseNumber, DriverLicenseExpDate, DriverLicenseState)
VALUES ('L151878813l2577', '12-OCT-2027', 'NJ');
INSERT INTO DriverLincense(DriverLicenseNumber, DriverLicenseExpDate, DriverLicenseState)
VALUES ('751878814', '17-MAY-2028', 'NY');
INSERT INTO DriverLincense(DriverLicenseNumber, DriverLicenseExpDate, DriverLicenseState)
VALUES ('L1518788131299', '26-JAN-2027', 'NJ');

INSERT INTO DriverLincense(DriverLicenseNumber, DriverLicenseExpDate, DriverLicenseState)
VALUES ('A15648915', '26-JAN-2028', 'VA');
INSERT INTO DriverLincense(DriverLicenseNumber, DriverLicenseExpDate, DriverLicenseState)
VALUES ('14456288', '30-MAY-2028', 'PA');
INSERT INTO DriverLincense(DriverLicenseNumber, DriverLicenseExpDate, DriverLicenseState)
VALUES ('12345668', '25-MAY-2027', 'PA');
INSERT INTO DriverLincense(DriverLicenseNumber, DriverLicenseExpDate, DriverLicenseState)
```

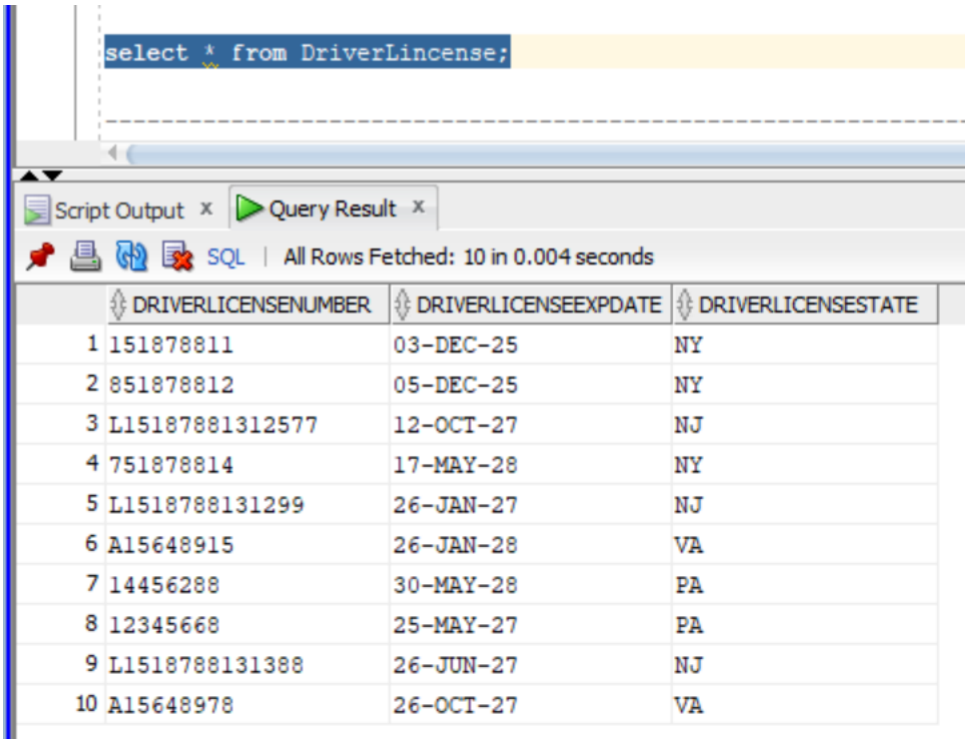
```
VALUES ('L1518788131388', '26-JUN-2027', 'NJ');
INSERT INTO DriverLincense(DriverLicenseNumber, DriverLicenseExpDate, DriverLicenseState)
VALUES ('A15648978', '26-OCT-2027', 'VA');
```

❖ **Table After Execution of SQL INSERT STATEMENTS:**

The following SQL SELECT Statement was used to list the content of the DriverLincense table:

```
SELECT * FROM DriverLincense;
```

The final state of the DriverLincense Table is:



```
select * from DriverLincense;
```

Script Output | Query Result | SQL | All Rows Fetched: 10 in 0.004 seconds

	DRIVERLICENSENUMBER	DRIVERLICENSEEXPDATE	DRIVERLICENSESTATE
1	151878811	03-DEC-25	NY
2	851878812	05-DEC-25	NY
3	L15187881312577	12-OCT-27	NJ
4	751878814	17-MAY-28	NY
5	L1518788131299	26-JAN-27	NJ
6	A15648915	26-JAN-28	VA
7	14456288	30-MAY-28	PA
8	12345668	25-MAY-27	PA
9	L1518788131388	26-JUN-27	NJ
10	A15648978	26-OCT-27	VA

2. Populating The CustomerUserAccount Table

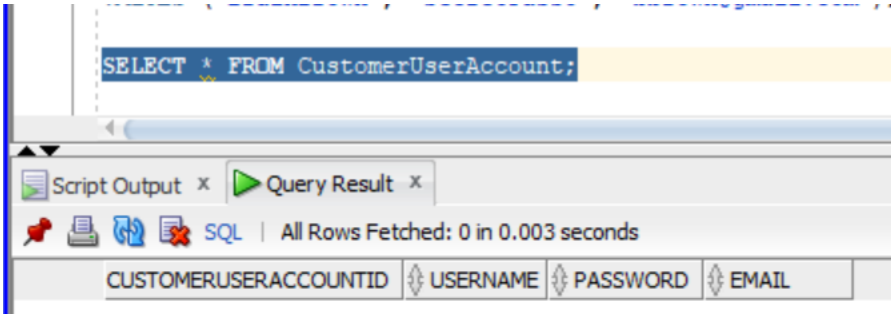
This section shows the content of CustomerUserAccount Table before and after populating it with records using an SQL INSERT STATEMENTS.

❖ **Table Before Execution of SQL INSERT STATEMENTS:**

The following SQL SELECT Statement was used to list the content of the CustomerUserAccount table:

```
SELECT * FROM CustomerUserAccount;
```

The current state of the CustomerUserAccount Table is:



```
SELECT * FROM CustomerUserAccount;
```

Script Output | Query Result | SQL | All Rows Fetched: 0 in 0.003 seconds

CUSTOMERUSERACCOUNTID	USERNAME	PASSWORD	EMAIL

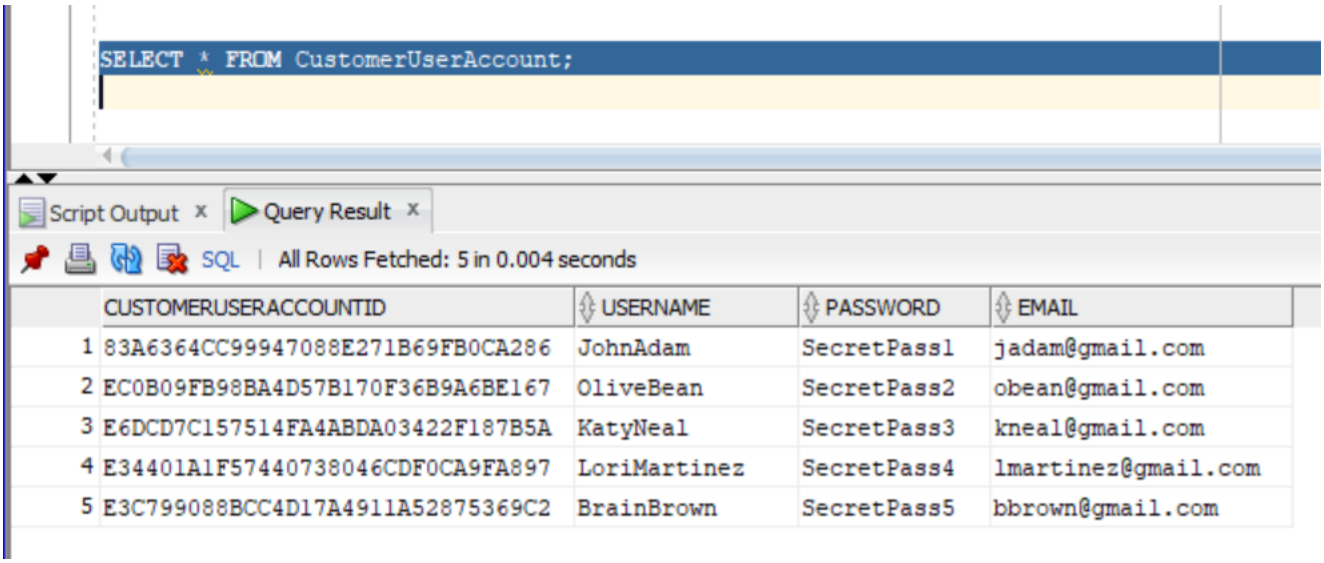
❖ **List of SQL INSERT STATEMENTS used to populate the table:**
The following SQL INSERT Statements were used to INSERT records to the CustomerUserAccount table:

```
INSERT INTO CustomerUserAccount(Username, Password, Email)
VALUES ('JohnAdam', 'SecretPass1', 'jadam@gmail.com');
INSERT INTO CustomerUserAccount(Username, Password, Email)
VALUES ('OliveBean', 'SecretPass2', 'obean@gmail.com');
INSERT INTO CustomerUserAccount(Username, Password, Email)
VALUES ('KatyNeal', 'SecretPass3', 'kneal@gmail.com');
INSERT INTO CustomerUserAccount(Username, Password, Email)
VALUES ('LoriMartinez', 'SecretPass4', 'lmartinez@gmail.com');
INSERT INTO CustomerUserAccount(Username, Password, Email)
VALUES ('BrainBrown', 'SecretPass5', 'bbrown@gmail.com');
```

❖ **Table After Execution of SQL INSERT STATEMENTS:**
The following SQL SELECT Statement was used to list the content of the CustomerUserAccount table:

```
SELECT * FROM CustomerUserAccount;
```

The final state of the CustomerUserAccount Table is:



	CUSTOMERUSERACCOUNTID	USERNAME	PASSWORD	EMAIL
1	83A6364CC99947088E271B69FB0CA286	JohnAdam	SecretPass1	jadam@gmail.com
2	EC0B09FB98BA4D57B170F36B9A6BE167	OliveBean	SecretPass2	obean@gmail.com
3	E6DCD7C157514FA4ABDA03422F187B5A	KatyNeal	SecretPass3	kneal@gmail.com
4	E34401A1F57440738046CDF0CA9FA897	LoriMartinez	SecretPass4	lmartinez@gmail.com
5	E3C799088BCC4D17A4911A52875369C2	BrainBrown	SecretPass5	bbrown@gmail.com

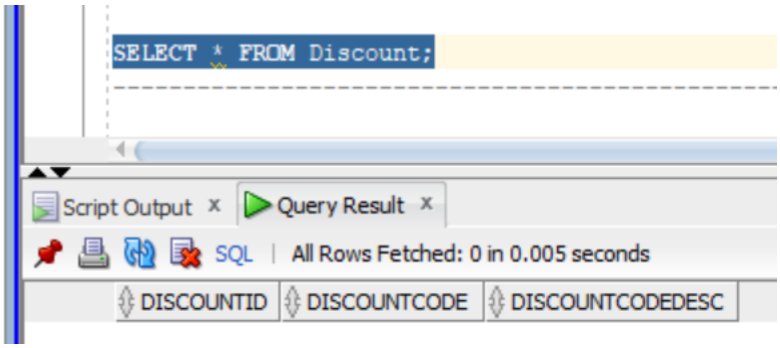
3. Populating The Discount Table

This section shows the content of Discount Table before and after populating it with records using an SQL INSERT STATEMENTS.

❖ **Table Before Execution of SQL INSERT STATEMENTS:**
The following SQL SELECT Statement was used to list the content of the Discount table:

```
SELECT * FROM Discount;
```

The current state of the Discount Table is:



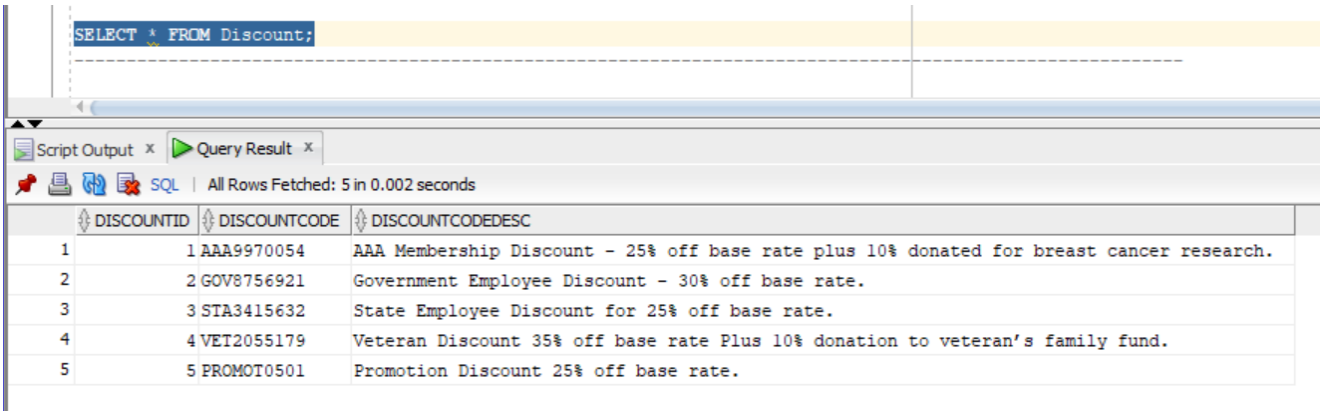
❖ **List of SQL INSERT STATEMENTS used to populate the table:**
The following SQL INSERT Statements were used to INSERT records to the Discount table:

```
INSERT INTO Discount(DiscountCode, DiscountCodeDesc)
VALUES ('AAA9970054', 'AAA Membership Discount - 25% off base rate plus 10% donated for breast cancer research.');
INSERT INTO Discount(DiscountCode, DiscountCodeDesc)
VALUES ('GOV8756921', 'Government Employee Discount - 30% off base rate.');
INSERT INTO Discount(DiscountCode, DiscountCodeDesc)
VALUES ('STA3415632', 'State Employee Discount for 25% off base rate.');
INSERT INTO Discount(DiscountCode, DiscountCodeDesc)
VALUES ('VET2055179', 'Veteran Discount 35% off base rate Plus 10% donation to veteran's family fund.');
INSERT INTO Discount(DiscountCode, DiscountCodeDesc)
VALUES ('PROMOT0501', 'Promotion Discount 25% off base rate.');
```

❖ **Table After Execution of SQL INSERT STATEMENTS:**
The following SQL SELECT Statement was used to list the content of the Discount table:

```
SELECT * FROM Discount;
```

The final state of the Discount Table is:



	DISCOUNTID	DISCOUNTCODE	DISCOUNTCODEDESC
1	1	AAA9970054	AAA Membership Discount - 25% off base rate plus 10% donated for breast cancer research.
2	2	GOV8756921	Government Employee Discount - 30% off base rate.
3	3	STA3415632	State Employee Discount for 25% off base rate.
4	4	VET2055179	Veteran Discount 35% off base rate Plus 10% donation to veteran's family fund.
5	5	PROMOT0501	Promotion Discount 25% off base rate.

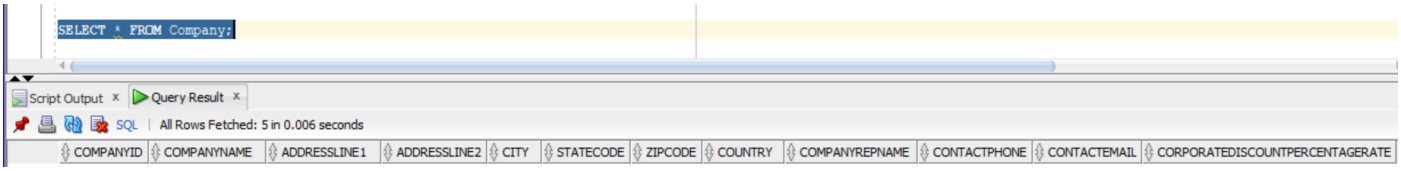
4. Populating The Company Table

This section shows the content of Company Table before and after populating it with records using an SQL INSERT STATEMENTS.

❖ **Table Before Execution of SQL INSERT STATEMENTS:**
The following SQL SELECT Statement was used to list the content of the Company table:

```
SELECT * FROM Company;
```

The current state of the Company Table is:



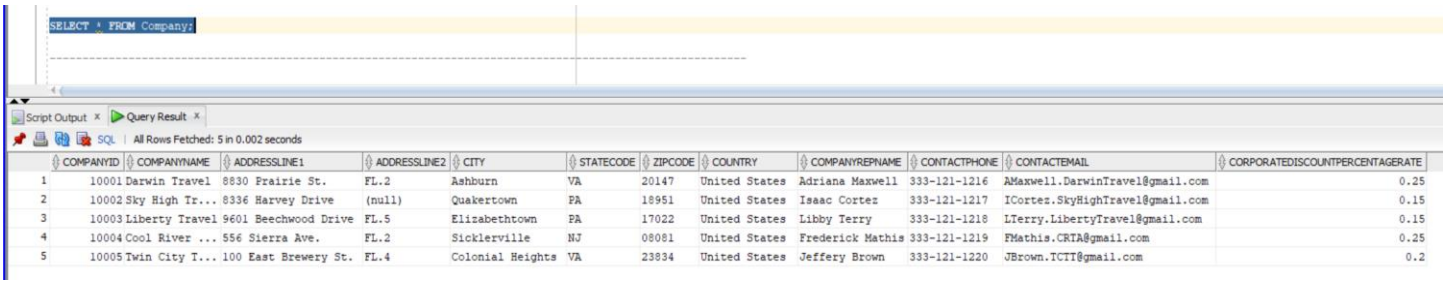
- **List of SQL INSERT STATEMENTS used to populate the table:**
  The following SQL INSERT Statements were used to INSERT records to the Company table:

```
INSERT INTO Company(CompanyID, CompanyName, AddressLine1, AddressLine2, City, StateCode, ZipCode, Country, CompanyRepName, ContactPhone, ContactEmail, CorporateDiscountPercentageRate) VALUES(10001, 'Darwin Travel', '8830 Prairie St.', 'FL.2','Ashburn', 'VA', '20147', 'United States', 'Adriana Maxwell', '333-121-1216', 'AMaxwell.DarwinTravel@gmail.com', 0.25);

INSERT INTO Company(CompanyID, CompanyName, AddressLine1, City, StateCode, ZipCode, Country, CompanyRepName, ContactPhone, ContactEmail, CorporateDiscountPercentageRate) VALUES(10002, 'Sky High Travel', '8336 Harvey Drive', 'Quakertown', 'PA', '18951', 'United States', 'Isaac Cortez', '333-121-1217', 'ICortez.SkyHighTravel@gmail.com', 0.15);

INSERT INTO Company(CompanyID, CompanyName, AddressLine1, AddressLine2, City, StateCode, ZipCode, Country, CompanyRepName, ContactPhone, ContactEmail, CorporateDiscountPercentageRate) VALUES(10003, 'Liberty Travel', '9601 Beechwood Drive', 'FL.5', 'Elizabethtown', 'PA', '17022', 'United States', 'Libby Terry', '333-121-1218', 'LTerry.LibertyTravel@gmail.com', 0.15);

INSERT INTO Company(CompanyID, CompanyName, AddressLine1, AddressLine2, City, StateCode, ZipCode, Country, CompanyRepName, ContactPhone, ContactEmail, CorporateDiscountPercentageRate) VALUES(10004, 'Cool River Travel Agency', '556 Sierra Ave.', 'FL.2', 'Sicklerville', 'NJ', '08081', 'United States', 'Frederick Mathis', '333-121-1219', 'FMathis.CRTA@gmail.com', 0.25);

INSERT INTO Company(CompanyID, CompanyName, AddressLine1, AddressLine2, City, StateCode, ZipCode, Country, CompanyRepName, ContactPhone, ContactEmail, CorporateDiscountPercentageRate) VALUES(10005, 'Twin City Travel Tourism', '100 East Brewery St.', 'FL.4', 'Colonial Heights', 'VA', '23834', 'United States', 'Jeffery Brown', '333-121-1220', 'JBrown.TCTT@gmail.com', 0.20);
```

- **Table After Execution of SQL INSERT STATEMENTS:**
  The following SQL SELECT Statement was used to list the content of the Company table:

```
SELECT * FROM Company;
```

The final state of the Company Table is:



	COMPANYID	COMPANYNAME	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY	COMPANYREPNAME	CONTACTPHONE	CONTACTEMAIL	CORPORATEDISCOUNTPERCENTAGERATE
1	10001	Darwin Travel	8830 Prairie St.	FL.2	Ashburn	VA	20147	United States	Adriana Maxwell	333-121-1216	AMaxwell.DarwinTravel@gmail.com	0.25
2	10002	Sky High Tr...	8336 Harvey Drive	(null)	Quakertown	PA	18951	United States	Isaac Cortez	333-121-1217	ICortez.SkyHighTravel@gmail.com	0.15
3	10003	Liberty Travel	9601 Beechwood Drive	FL.5	Elizabethtown	PA	17022	United States	Libby Terry	333-121-1218	LTerry.LibertyTravel@gmail.com	0.15
4	10004	Cool River ...	556 Sierra Ave.	FL.2	Sicklerville	NJ	08081	United States	Frederick Mathis	333-121-1219	FMathis.CRTA@gmail.com	0.25
5	10005	Twin City T...	100 East Brewery St.	FL.4	Colonial Heights	VA	23834	United States	Jeffery Brown	333-121-1220	JBrown.TCTT@gmail.com	0.2

5. Populating The CreditCardMerchant Table

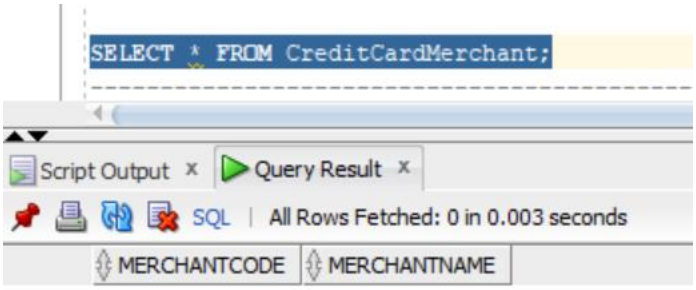
This section shows the content of CreditCardMerchant Table before and after populating it with records using an SQL INSERT STATEMENTS.

❖ **Table Before Execution of SQL INSERT STATEMENTS:**

The following SQL SELECT Statement was used to list the content of the CreditCardMerchant table:

```
SELECT * FROM CreditCardMerchant;
```

The current state of the CreditCardMerchant Table is:



❖ **List of SQL INSERT STATEMENTS used to populate the table:**

The following SQL INSERT Statements were used to INSERT records to the CreditCardMerchant table:

```
INSERT INTO CreditCardMerchant(MerchantCode, MerchantName)
VALUES (1, 'Stax by Fattmerchant');
INSERT INTO CreditCardMerchant(MerchantCode, MerchantName)
VALUES (2, 'Helcim');
INSERT INTO CreditCardMerchant(MerchantCode, MerchantName)
VALUES (3, 'Dharma Merchant Services');
INSERT INTO CreditCardMerchant(MerchantCode, MerchantName)
VALUES (4, 'Payment Depot');
INSERT INTO CreditCardMerchant(MerchantCode, MerchantName)
VALUES (5, 'National Processing');
INSERT INTO CreditCardMerchant(MerchantCode, MerchantName)
VALUES (6, 'Block');
INSERT INTO CreditCardMerchant(MerchantCode, MerchantName)
VALUES (7, 'Intuit Quickbooks');
INSERT INTO CreditCardMerchant(MerchantCode, MerchantName)
VALUES (8, 'PayPal');
INSERT INTO CreditCardMerchant(MerchantCode, MerchantName)
VALUES (9, 'Stripe');
```

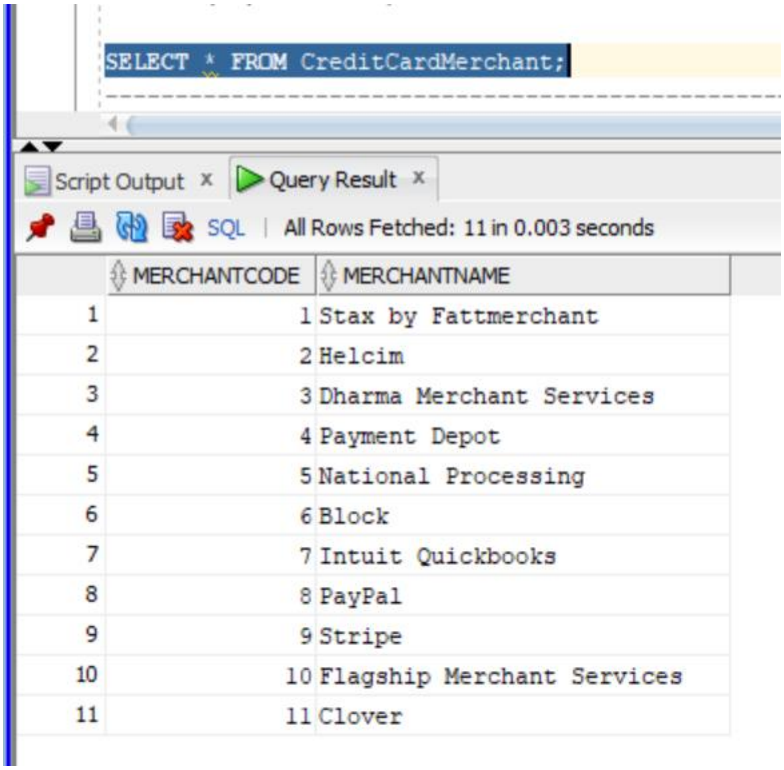
```
INSERT INTO CreditCardMerchant(MerchantCode, MerchantName)
VALUES (10, 'Flagship Merchant Services');
INSERT INTO CreditCardMerchant(MerchantCode, MerchantName)
VALUES (11, 'Clover');
```

❖ **Table After Execution of SQL INSERT STATEMENTS:**

The following SQL SELECT Statement was used to list the content of the CreditCardMerchant table:

```
SELECT * FROM CreditCardMerchant;
```

The final state of the CreditCardMerchant Table is:



```
SELECT * FROM CreditCardMerchant;
```

Script Output | Query Result

SQL | All Rows Fetched: 11 in 0.003 seconds

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

6. Populating The Customer Table

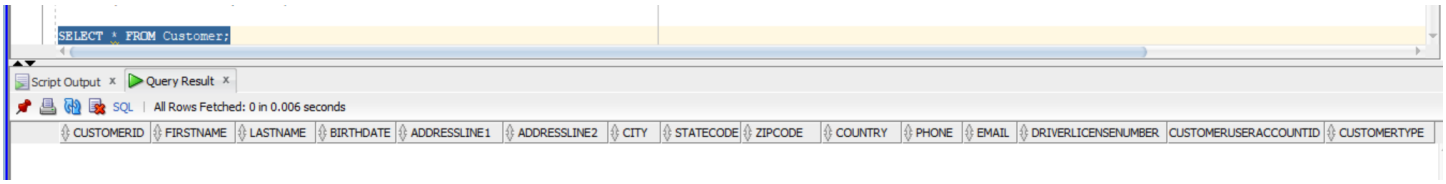
This section shows the content of Customer Table before and after populating it with records using an SQL INSERT STATEMENTS.

❖ **Table Before Execution of SQL INSERT STATEMENTS:**

The following SQL SELECT Statement was used to list the content of the Customer table:

```
SELECT * FROM Customer;
```

The current state of the Customer Table is:



```
SELECT * FROM Customer;
```

Script Output | Query Result

SQL | All Rows Fetched: 0 in 0.006 seconds

CUSTOMERID	FIRSTNAME	LASTNAME	BIRTHDATE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY	PHONE	EMAIL	DRIVERLICENSENUMBER	CUSTOMERUSERACCOUNTID	CUSTOMERTYPE

❖ **List of SQL INSERT STATEMENTS used to populate the table:**
The following SQL INSERT Statements were used to INSERT records to the Customer table:

```
INSERT INTO Customer(FirstName, LastName, BirthDate, AddressLine1, AddressLine2, City, StateCode,
ZipCode, Country, Phone, Email, DriverLicenseNumber, CustomerUserAccountID, CustomerType)
VALUES ('John', 'Adam', '01-JAN-1979', '10 Fulton Ave.', 'Apt 2B', 'Brooklyn', 'NY', '11203', 'United States',
'333-121-1211', 'jadam@gmail.com', '151878811', '83A6364CC99947088E271B69FB0CA286', 'R');

INSERT INTO Customer(FirstName, LastName, BirthDate, AddressLine1, AddressLine2, City, StateCode,
ZipCode, Country, Phone, Email, DriverLicenseNumber, CustomerUserAccountID, CustomerType)
VALUES ('Olive', 'Bean', '02-JAN-1979', '32 State Ave.', 'Apt 3B', 'Brooklyn', 'NY', '11204', 'United States', '333-
121-1212', 'obean@gmail.com', '851878812', 'EC0B09FB98BA4D57B170F36B9A6BE167', 'R');

INSERT INTO Customer(FirstName, LastName, BirthDate, AddressLine1, AddressLine2, City, StateCode,
ZipCode, Country, Phone, Email, DriverLicenseNumber, CustomerUserAccountID, CustomerType)
VALUES ('Katy', 'Neal', '19-MAY-1975', '2823 Spring Haven Trail', 'Apt 2B', 'Lake Mohawk', 'NJ', '07871',
'United States', '333-121-1213', 'kneal@gmail.com', 'L15187881312577',
'E6DCD7C157514FA4ABDA03422F187B5A', 'R');

INSERT INTO Customer(FirstName, LastName, BirthDate, AddressLine1, AddressLine2, City, StateCode,
ZipCode, Country, Phone, Email, DriverLicenseNumber, CustomerUserAccountID, CustomerType)
VALUES ('Lori', 'Martinez', '01-FEB-1988', '6 Oakwood Ave.', 'Apt 4C', 'Bronx', 'NY', '10472', 'United States',
'333-121-1214', 'lmartinez@gmail.com', '751878814', 'E34401A1F57440738046CDF0CA9FA897', 'R');

INSERT INTO Customer(FirstName, LastName, BirthDate, AddressLine1, City, StateCode, ZipCode,
Country, Phone, Email, DriverLicenseNumber, CustomerUserAccountID, CustomerType)
VALUES ('Brain', 'Brown', '01-JAN-1996', '1872 Granville Lane', 'Newark', 'NJ', '07662', 'United States', '333-
121-1215', 'bbrown@gmail.com', 'L1518788131299', 'E3C799088BCC4D17A4911A52875369C2', 'R');

-----
INSERT INTO Customer(FirstName, LastName, BirthDate, AddressLine1, AddressLine2, City, StateCode,
ZipCode, Country, Phone, Email, DriverLicenseNumber, CustomerType)
VALUES ('Adriana', 'Maxwell', '01-JAN-1996', '8830 Prairie St.', 'FL.2', 'Ashburn', 'VA', '20147', 'United States',
'333-121-1216', 'AMaxwell.DarwinTravel@gmail.com', 'A15648915', 'C');

INSERT INTO Customer(FirstName, LastName, BirthDate, AddressLine1, City, StateCode, ZipCode,
Country, Phone, Email, DriverLicenseNumber, CustomerType)
VALUES ('Isaac', 'Cortez', '18-MAY-1997', '8336 Harvey Drive', 'Quakertown', 'PA', '18951', 'United States',
'333-121-1217', 'ICortez.SkyHighTravel@gmail.com', '14456288', 'C');

INSERT INTO Customer(FirstName, LastName, BirthDate, AddressLine1, AddressLine2, City, StateCode,
ZipCode, Country, Phone, Email, DriverLicenseNumber, CustomerType)
VALUES ('Libby', 'Terry', '28-JUN-1996', '9601 Beechwood Drive', 'FL.5', 'Elizabethtown', 'PA', '17022', 'United
States', '333-121-1218', 'LTerry.LibertyTravel@gmail.com', '12345668', 'C');

INSERT INTO Customer(FirstName, LastName, BirthDate, AddressLine1, AddressLine2, City, StateCode,
ZipCode, Country, Phone, Email, DriverLicenseNumber, CustomerType)
VALUES ('Frederick', 'Mathis', '07-AUG-1976', '556 Sierra Ave.', 'FL.2', 'Sicklerville', 'NJ', '08081', 'United
States', '333-121-1219', 'fmathis@gmail.com', 'L1518788131388', 'C');
```

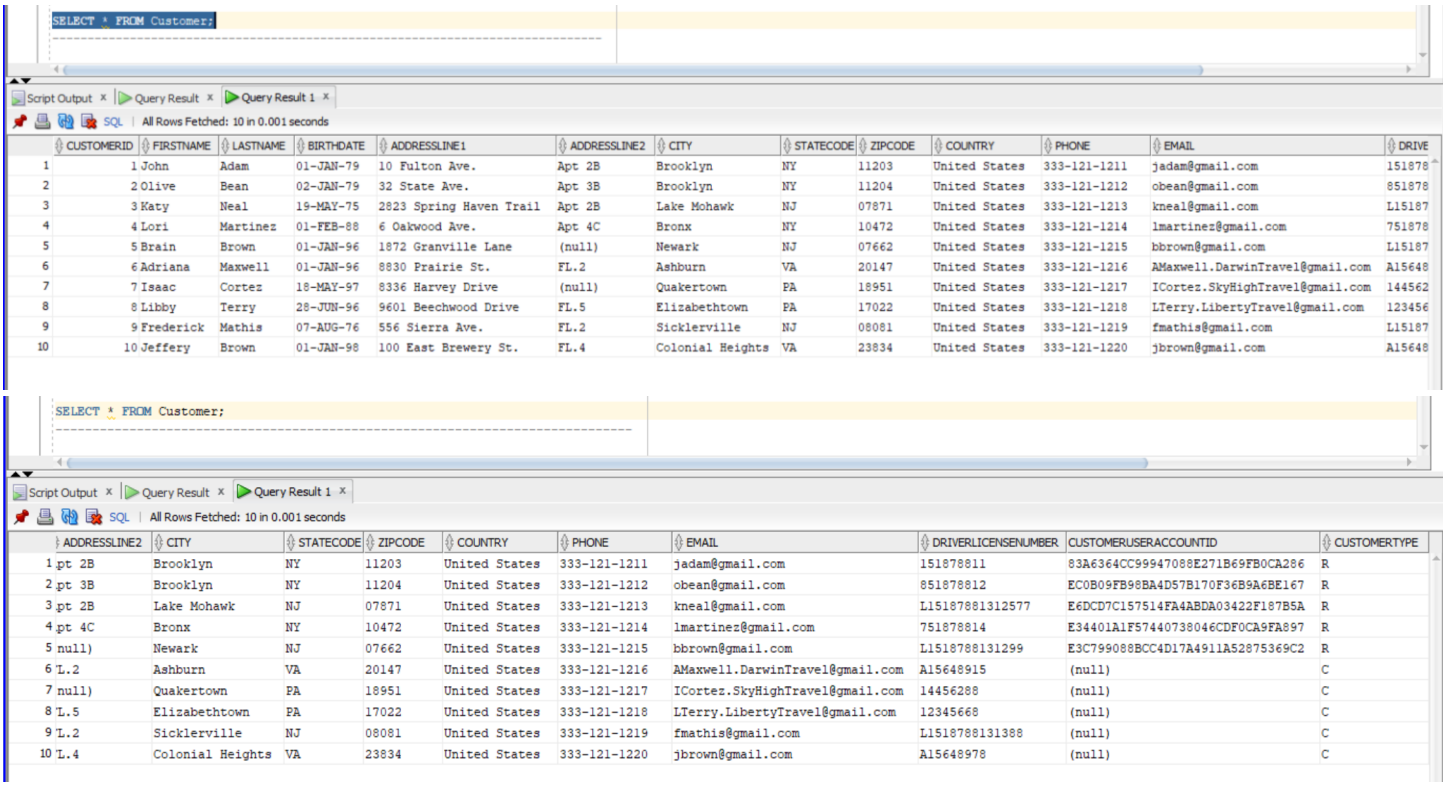
```
INSERT INTO Customer(FirstName, LastName, BirthDate, AddressLine1, AddressLine2, City, StateCode,
ZipCode, Country, Phone, Email, DriverLicenseNumber, CustomerType)
VALUES ('Jeffery', 'Brown', '01-JAN-1998', '100 East Brewery St.', 'FL.4', 'Colonial Heights', 'VA', '23834',
'United States', '333-121-1220', 'jbrown@gmail.com', 'A15648978', 'C');
```

- **Table After Execution of SQL INSERT STATEMENTS:**

  The following SQL SELECT Statement was used to list the content of the Customer table:

```
SELECT * FROM Customer;
```

The final state of the Customer Table is:



```
SELECT * FROM Customer;
```

Script Output | Query Result | Query Result 1

All Rows Fetched: 10 in 0.001 seconds

	CUSTOMERID	FIRSTNAME	LASTNAME	BIRTHDATE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY	PHONE	EMAIL	DRIVE
1	1	John	Adam	01-JAN-79	10 Fulton Ave.	Apt 2B	Brooklyn	NY	11203	United States	333-121-1211	jadam@gmail.com	151878
2	2	Olive	Bean	02-JAN-79	32 State Ave.	Apt 3B	Brooklyn	NY	11204	United States	333-121-1212	obean@gmail.com	851878
3	3	Katy	Neal	19-MAY-75	2823 Spring Haven Trail	Apt 2B	Lake Mohawk	NJ	07871	United States	333-121-1213	kneal@gmail.com	L15187
4	4	Lori	Martinez	01-FEB-88	6 Oakwood Ave.	Apt 4C	Bronx	NY	10472	United States	333-121-1214	lmartinez@gmail.com	751878
5	5	Brain	Brown	01-JAN-96	1872 Granville Lane	(null)	Newark	NJ	07662	United States	333-121-1215	bbrown@gmail.com	L15187
6	6	Adriana	Maxwell	01-JAN-96	8830 Prairie St.	FL.2	Ashburn	VA	20147	United States	333-121-1216	AMaxwell.DarwinTravel@gmail.com	A15648
7	7	Isaac	Cortez	18-MAY-97	8336 Harvey Drive	(null)	Quakertown	PA	18951	United States	333-121-1217	ICortez.SkyHighTravel@gmail.com	144562
8	8	Libby	Terry	28-JUN-96	9601 Beechwood Drive	FL.5	Elizabethtown	PA	17022	United States	333-121-1218	LTerry.LibertyTravel@gmail.com	123456
9	9	Frederick	Mathis	07-AUG-76	556 Sierra Ave.	FL.2	Sicklerville	NJ	08081	United States	333-121-1219	fmathis@gmail.com	L15187
10	10	Jeffery	Brown	01-JAN-98	100 East Brewery St.	FL.4	Colonial Heights	VA	23834	United States	333-121-1220	jbrown@gmail.com	A15648

```
SELECT * FROM Customer;
```

Script Output | Query Result | Query Result 1

All Rows Fetched: 10 in 0.001 seconds

	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY	PHONE	EMAIL	DRIVERLICENSENUMBER	CUSTOMERUSERACCOUNTID	CUSTOMERTYPE
1	pt 2B	Brooklyn	NY	11203	United States	333-121-1211	jadam@gmail.com	151878811	83A6364CC99947088E271B69FB0CA286	R
2	pt 3B	Brooklyn	NY	11204	United States	333-121-1212	obean@gmail.com	851878812	EC0B09FB98BA4D57B170F36B9A6BE167	R
3	pt 2B	Lake Mohawk	NJ	07871	United States	333-121-1213	kneal@gmail.com	L15187881312577	E6DCD7C157514FA4ABDA03422F187B5A	R
4	pt 4C	Bronx	NY	10472	United States	333-121-1214	lmartinez@gmail.com	751878814	E34401A1F57440738046CDF0CA9FA897	R
5	null)	Newark	NJ	07662	United States	333-121-1215	bbrown@gmail.com	L1518788131299	E3C799088BCC4D17A4911A52875369C2	R
6	L.2	Ashburn	VA	20147	United States	333-121-1216	AMaxwell.DarwinTravel@gmail.com	A15648915	(null)	C
7	null)	Quakertown	PA	18951	United States	333-121-1217	ICortez.SkyHighTravel@gmail.com	14456288	(null)	C
8	L.5	Elizabethtown	PA	17022	United States	333-121-1218	LTerry.LibertyTravel@gmail.com	12345668	(null)	C
9	L.2	Sicklerville	NJ	08081	United States	333-121-1219	fmathis@gmail.com	L1518788131388	(null)	C
10	L.4	Colonial Heights	VA	23834	United States	333-121-1220	jbrown@gmail.com	A15648978	(null)	C

7. Populating The EZPlus Table

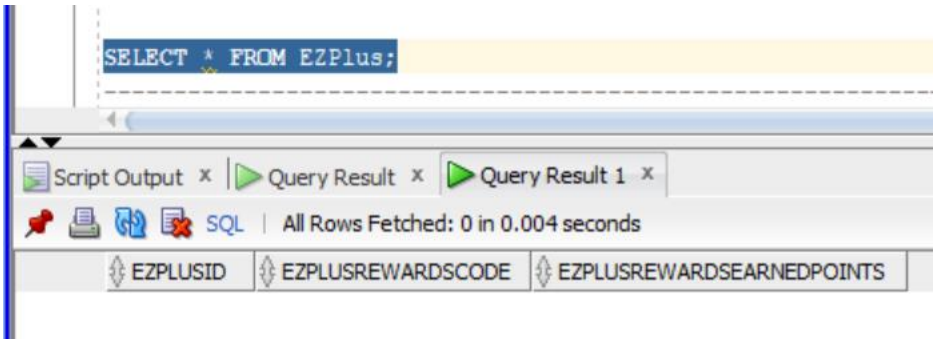
This section shows the content of EZPlus Table before and after populating it with records using an SQL INSERT STATEMENTS.

- **Table Before Execution of SQL INSERT STATEMENTS:**

  The following SQL SELECT Statement was used to list the content of the EZPlus table:

```
SELECT * FROM EZPlus;
```

The current state of the EZPlus Table is:



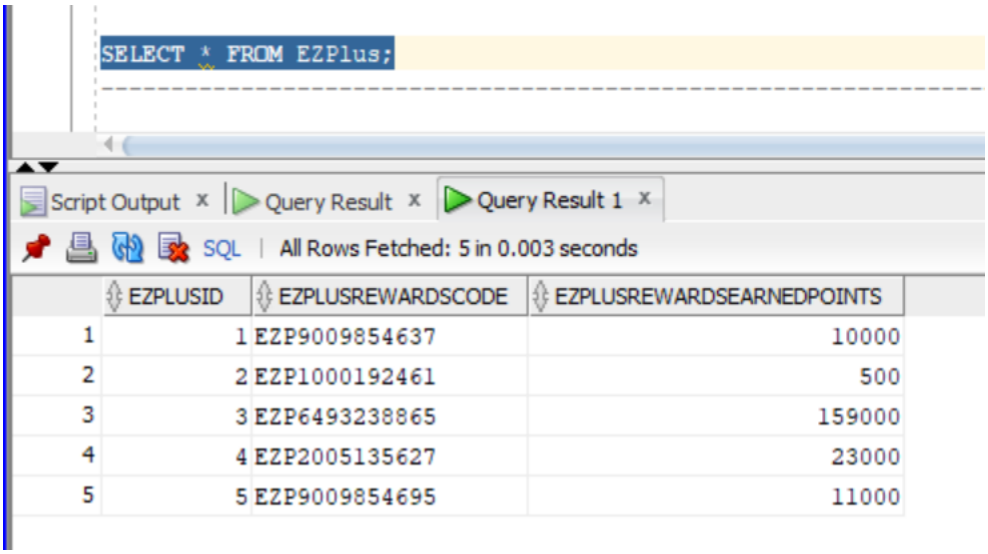
- **List of SQL INSERT STATEMENTS used to populate the table:**
  The following SQL INSERT Statements were used to INSERT records to the EZPlus table:

```
INSERT INTO EZPlus(EZPlusRewardsCode, EZPlusRewardsEarnedPoints)
VALUES ('EZP9009854637', 10000);

INSERT INTO EZPlus(EZPlusRewardsCode, EZPlusRewardsEarnedPoints)
VALUES ('EZP1000192461', 500);

INSERT INTO EZPlus(EZPlusRewardsCode, EZPlusRewardsEarnedPoints)
VALUES ('EZP6493238865', 159000);

INSERT INTO EZPlus(EZPlusRewardsCode, EZPlusRewardsEarnedPoints)
VALUES ('EZP2005135627', 23000);

INSERT INTO EZPlus(EZPlusRewardsCode, EZPlusRewardsEarnedPoints)
VALUES ('EZP9009854695', 11000);
```

- **Table After Execution of SQL INSERT STATEMENTS:**
  The following SQL SELECT Statement was used to list the content of the EZPlus table:

```
SELECT * FROM EZPlus;
```

The final state of the EZPlus Table is:



	EZPLUSID	EZPLUSREWARDSCODE	EZPLUSREWARDSEARNEDPOINTS
1	1	EZP9009854637	10000
2	2	EZP1000192461	500
3	3	EZP6493238865	159000
4	4	EZP2005135627	23000
5	5	EZP9009854695	11000

8. Populating The RetailCustomer Table

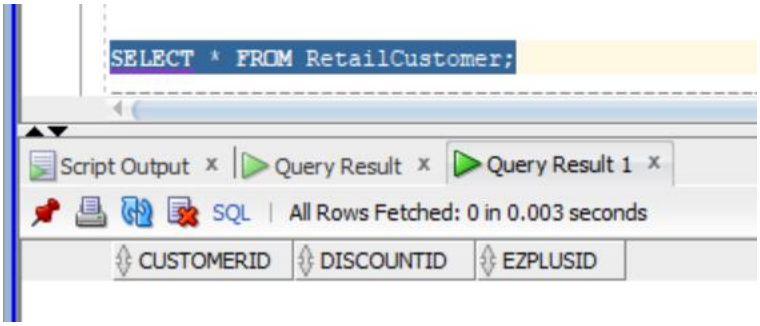
This section shows the content of RetailCustomer Table before and after populating it with records using an SQL INSERT STATEMENTS.

❖ **Table Before Execution of SQL INSERT STATEMENTS:**

The following SQL SELECT Statement was used to list the content of the RetailCustomer table:

```
SELECT * FROM RetailCustomer;
```

The current state of the RetailCustomer Table is:



❖ **List of SQL INSERT STATEMENTS used to populate the table:**

The following SQL INSERT Statements were used to INSERT records to the RetailCustomer table:

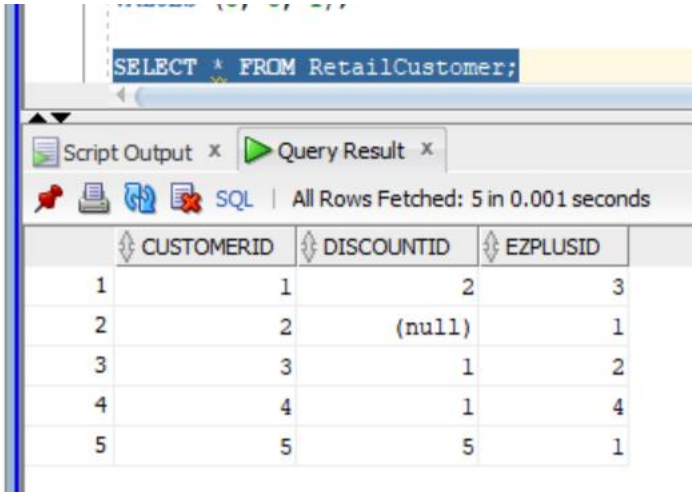
```
INSERT INTO RetailCustomer(CustomerID, DiscountID, EZPlusID)
VALUES (1, 2, 3);

INSERT INTO RetailCustomer(CustomerID, EZPlusID)
VALUES (2, 1);

INSERT INTO RetailCustomer(CustomerID, DiscountID, EZPlusID)
VALUES (3, 1, 2);

INSERT INTO RetailCustomer(CustomerID, DiscountID, EZPlusID)
VALUES (4, 1, 4);

INSERT INTO RetailCustomer(CustomerID, DiscountID, EZPlusID)
VALUES (5, 5, 1);
```

❖ **Table After Execution of SQL INSERT STATEMENTS:**

The following SQL SELECT Statement was used to list the content of the RetailCustomer table:

```
SELECT * FROM RetailCustomer;
```

The final state of the DriverLincense Table is:



9. Populating The CorporateCustomer Table

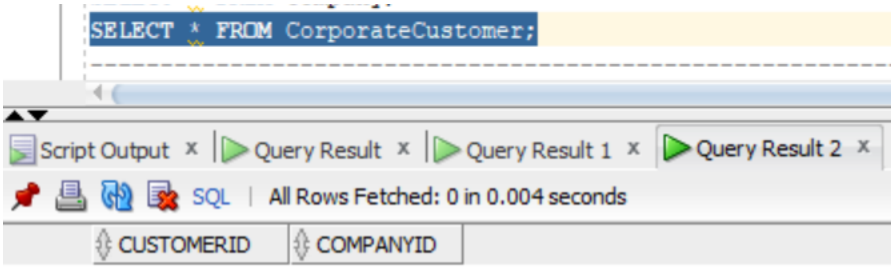
This section shows the content of CorporateCustomer Table before and after populating it with records using an SQL INSERT STATEMENTS.

❖ **Table Before Execution of SQL INSERT STATEMENTS:**

The following SQL SELECT Statement was used to list the content of the CorporateCustomer table:

```
SELECT * FROM CorporateCustomer;
```

The current state of the CorporateCustomer Table is:



❖ **List of SQL INSERT STATEMENTS used to populate the table:**

The following SQL INSERT Statements were used to INSERT records to the CorporateCustomer table:

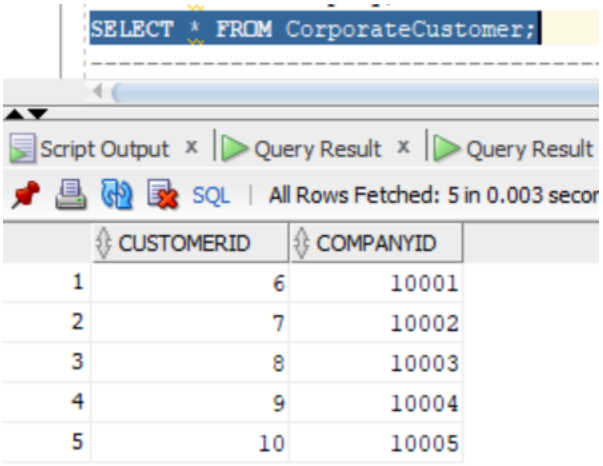
```
INSERT INTO CorporateCustomer(CustomerID, CompanyID)
VALUES (6, 10001);
INSERT INTO CorporateCustomer(CustomerID, CompanyID)
VALUES (7, 10002);
INSERT INTO CorporateCustomer(CustomerID, CompanyID)
VALUES (8, 10003);
INSERT INTO CorporateCustomer(CustomerID, CompanyID)
VALUES (9, 10004);
INSERT INTO CorporateCustomer(CustomerID, CompanyID)
VALUES (10, 10005);
```

❖ **Table After Execution of SQL INSERT STATEMENTS:**

The following SQL SELECT Statement was used to list the content of the CorporateCustomer:

```
SELECT * FROM CorporateCustomer;
```

The final state of the CorporateCustomer Table is:



	CUSTOMERID	COMPANYID
1	6	10001
2	7	10002
3	8	10003
4	9	10004
5	10	10005

10. Populating The CreditCard Table

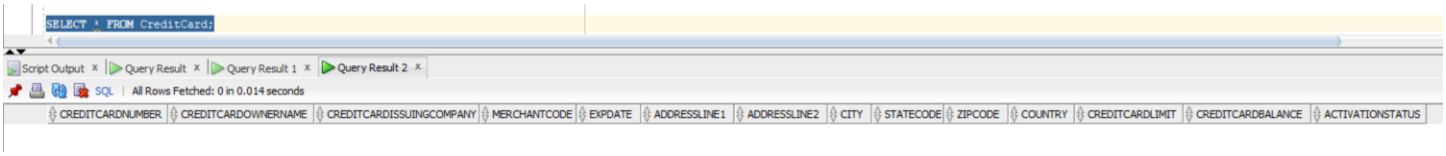
This section shows the content of CreditCard Table before and after populating it with records using an SQL INSERT STATEMENTS.

- **Table Before Execution of SQL INSERT STATEMENTS:**

The following SQL SELECT Statement was used to list the content of the CreditCard:

```
SELECT * FROM CreditCard;
```

The current state of the CreditCard Table is:



- **List of SQL INSERT STATEMENTS used to populate the table:**

The following SQL INSERT Statements were used to INSERT records to the CreditCard table:

```
-- A Customer OWNS only ONE Credit Card
INSERT INTO CreditCard(CreditCardNumber, CreditCardOwnerName, CreditCardIssuingCompany,
MerchantCode, ExpDate, AddressLine1, AddressLine2, City, StateCode, Zipcode, Country,
CreditCardLimit, CreditCardBalance, ActivationStatus)
VALUES ('3333333333333331', 'John Adam', 'American Express', 1, '08-DEC-2023', '10 Fulton Ave.', 'Apt 2B',
'Brooklyn', 'NY', '11203', 'United States', 20000.00, 5486.12, '1');

-- A Credit Card is OWNED by two Customers (Olive Bean & Katy Neal)
INSERT INTO CreditCard(CreditCardNumber, CreditCardOwnerName, CreditCardIssuingCompany,
MerchantCode, ExpDate, AddressLine1, AddressLine2, City, StateCode, Zipcode, Country,
CreditCardLimit, CreditCardBalance, ActivationStatus)
VALUES ('3333333333333332', 'Olive Bean', 'American Express', 1, '25-JUN-2024', '32 State Ave.', 'Apt 3B',
'Brooklyn', 'NY', '11204', 'United States', 22000.00, 6486.32, '1');
```

```
-- A Customer OWNS two Credit Cards
INSERT INTO CreditCard(CreditCardNumber, CreditCardOwnerName, CreditCardIssuingCompany,
MerchantCode, ExpDate, AddressLine1, AddressLine2, City, StateCode, Zipcode, Country,
CreditCardLimit, CreditCardBalance, ActivationStatus)
VALUES ('3333333333333333', 'Lori Martinez', 'Chase', 8, '18-DEC-2027', '6 Oakwood Ave.', 'Apt 4C', 'Bronx',
'NY', '10472', 'United States', 15000.00, 6286.12, '1');
INSERT INTO CreditCard(CreditCardNumber, CreditCardOwnerName, CreditCardIssuingCompany,
MerchantCode, ExpDate, AddressLine1, AddressLine2, City, StateCode, Zipcode, Country,
CreditCardLimit, CreditCardBalance, ActivationStatus)
VALUES ('3333333333333334', 'Lori Martinez', 'Capital One', 8, '25-MAY-2025', '6 Oakwood Ave.', 'Apt 4C',
'Bronx', 'NY', '10472', 'United States', 18000.00, 286.12, '1');

-- A Customer OWNS only ONE Credit Card
INSERT INTO CreditCard(CreditCardNumber, CreditCardOwnerName, CreditCardIssuingCompany,
MerchantCode, ExpDate, AddressLine1, City, StateCode, Zipcode, Country, CreditCardLimit,
CreditCardBalance, ActivationStatus)
VALUES ('3333333333333335', 'Brain Brown', 'Chase', 8, '11-DEC-2027', '1872 Granville Lane', 'Newark', 'NJ',
'07662', 'United States', 20000.00, 5486.32, '1');

-- A Customer OWNS only ONE Credit Card
INSERT INTO CreditCard(CreditCardNumber, CreditCardOwnerName, CreditCardIssuingCompany,
MerchantCode, ExpDate, AddressLine1, AddressLine2, City, StateCode, Zipcode, Country,
CreditCardLimit, CreditCardBalance, ActivationStatus)
VALUES ('3333333333333336', 'Adriana Maxwell', 'Capital One', 1, '11-DEC-2027', '8830 Prairie St.', 'FL.2',
'Ashburn', 'VA', '20147', 'United States', 22000.00, 1486.32, '1');

-- A Customer OWNS only ONE Credit Card
INSERT INTO CreditCard(CreditCardNumber, CreditCardOwnerName, CreditCardIssuingCompany,
MerchantCode, ExpDate, AddressLine1, City, StateCode, Zipcode, Country, CreditCardLimit,
CreditCardBalance, ActivationStatus)
VALUES ('3333333333333337', 'Isaac Cortez', 'Capital One', 3, '11-DEC-2027', '8336 Harvey Drive',
'Quakertown', 'PA', '18951', 'United States', 20000.00, 5486.32, '1');

-- A Customer OWNS only ONE Credit Card
INSERT INTO CreditCard(CreditCardNumber, CreditCardOwnerName, CreditCardIssuingCompany,
MerchantCode, ExpDate, AddressLine1, AddressLine2, City, StateCode, Zipcode, Country,
CreditCardLimit, CreditCardBalance, ActivationStatus)
VALUES ('3333333333333338', 'Libby Terry', 'Capital One', 3, '11-DEC-2027', '9601 Beechwood Drive', 'FL.5',
'Elizabethtown', 'PA', '17022', 'United States', 20000.00, 5486.32, '1');

-- A Customer OWNS only ONE Credit Card
INSERT INTO CreditCard(CreditCardNumber, CreditCardOwnerName, CreditCardIssuingCompany,
MerchantCode, ExpDate, AddressLine1, AddressLine2, City, StateCode, Zipcode, Country,
CreditCardLimit, CreditCardBalance, ActivationStatus)
VALUES ('3333333333333339', 'Frederick Mathis', 'Chase', 8, '11-DEC-2027', '556 Sierra Ave.', 'FL.2',
'Sicklerville', 'NJ', '08081', 'United States', 20000.00, 5486.32, '1');

--A Credit Card is OWNED by two Customers.(Jeffery Brown&Brain Brown)
```

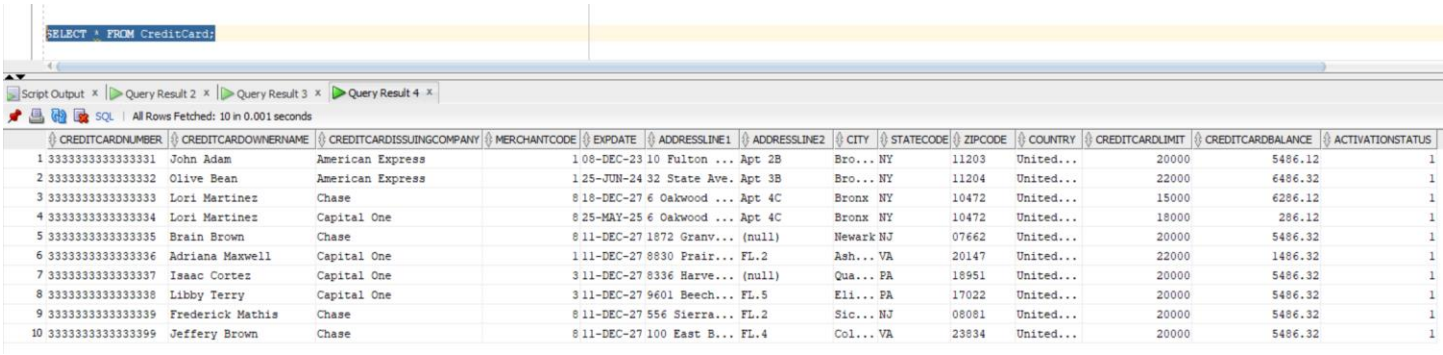
INSERT INTO CreditCard(CreditCardNumber, CreditCardOwnerName, CreditCardIssuingCompany, MerchantCode, ExpDate, AddressLine1, AddressLine2, City, StateCode, Zipcode, Country, CreditCardLimit, CreditCardBalance, ActivationStatus)
VALUES ('3333333333333399', 'Jeffery Brown', 'Chase', 8, '11-DEC-2027', '100 East Brewery St.', 'FL.4', 'Colonial Heights', 'VA', '23834', 'United States', 20000.00, 5486.32, '1');

❖ **Table After Execution of SQL INSERT STATEMENTS:**

The following SQL SELECT Statement was used to list the content of the CreditCard:

```
SELECT * FROM CreditCard;
```

The final state of the CreditCard Table is:



SELECT * FROM CreditCard;

Script Output | Query Result 2 | Query Result 3 | Query Result 4

SQL | All Rows Fetched: 10 in 0.001 seconds

	CREDITCARDNUMBER	CREDITCARDOWNERNAME	CREDITCARDISSUINGCOMPANY	MERCHANTCODE	EXPDATE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY	CREDITCARDLIMIT	CREDITCARDBALANCE	ACTIVATIONSTATUS
1	3333333333333331	John Adam	American Express	1	08-DEC-23	10 Fulton ...	Apt 2B	Bro...	NY	11203	United...	20000	5486.12	1
2	3333333333333332	Olive Bean	American Express	1	25-JUN-24	32 State Ave.	Apt 3B	Bro...	NY	11204	United...	22000	6486.32	1
3	3333333333333333	Lori Martinez	Chase	8	18-DEC-27	6 Oakwood ...	Apt 4C	Bronx	NY	10472	United...	15000	6286.12	1
4	3333333333333334	Lori Martinez	Capital One	8	25-MAY-25	6 Oakwood ...	Apt 4C	Bronx	NY	10472	United...	18000	286.12	1
5	3333333333333335	Brain Brown	Chase	8	11-DEC-27	1872 Granv...	(null)	Newark	NJ	07662	United...	20000	5486.32	1
6	3333333333333336	Adriana Maxwell	Capital One	1	11-DEC-27	8830 Prair...	FL.2	Ash...	VA	20147	United...	22000	1486.32	1
7	3333333333333337	Isaac Cortez	Capital One	3	11-DEC-27	8336 Harve...	(null)	Qua...	PA	18951	United...	20000	5486.32	1
8	3333333333333338	Libby Terry	Capital One	3	11-DEC-27	9601 Beech...	FL.5	Eli...	PA	17022	United...	20000	5486.32	1
9	3333333333333339	Frederick Mathis	Chase	8	11-DEC-27	556 Sierra...	FL.2	Sic...	NJ	08081	United...	20000	5486.32	1
10	3333333333333399	Jeffery Brown	Chase	8	11-DEC-27	100 East B...	FL.4	Col...	VA	23834	United...	20000	5486.32	1

11. Populating The Customer_CreditCard Table

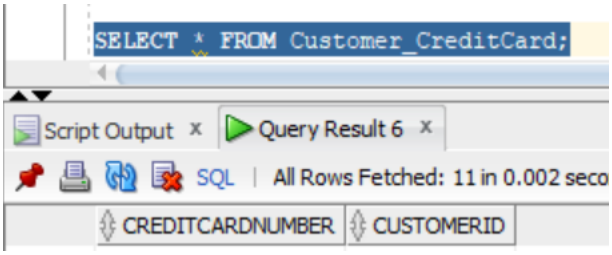
This section shows the content of Customer_CreditCard Table before and after populating it with records using an SQL INSERT STATEMENTS.

❖ **Table Before Execution of SQL INSERT STATEMENTS:**

The following SQL SELECT Statement was used to list the content of the Customer_CreditCard:

```
SELECT * FROM Customer_CreditCard;
```

The current state of the Customer_CreditCard Table is:



SELECT * FROM Customer_CreditCard;

Script Output | Query Result 6

SQL | All Rows Fetched: 11 in 0.002 seco

CREDITCARDNUMBER	CUSTOMERID

❖ **List of SQL INSERT STATEMENTS used to populate the table:**

The following SQL INSERT Statements were used to INSERT records to the Customer_CreditCard table:

INSERT INTO Customer_CreditCard(CreditCardNumber, CustomerID)
VALUES ('3333333333333331', 1);
INSERT INTO Customer_CreditCard(CreditCardNumber, CustomerID)
VALUES ('3333333333333332', 2);
INSERT INTO Customer_CreditCard(CreditCardNumber, CustomerID)
VALUES ('3333333333333332', 3);
INSERT INTO Customer_CreditCard(CreditCardNumber, CustomerID)

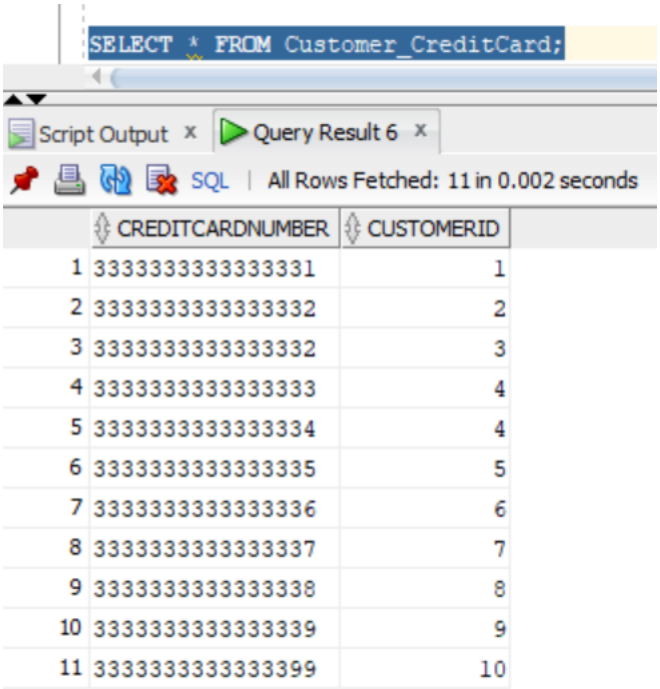
```
VALUES ('3333333333333333', 4);
INSERT INTO Customer_CreditCard(CreditCardNumber, CustomerID)
VALUES ('3333333333333334', 4);
INSERT INTO Customer_CreditCard(CreditCardNumber, CustomerID)
VALUES ('3333333333333335', 5);
INSERT INTO Customer_CreditCard(CreditCardNumber, CustomerID)
VALUES ('3333333333333336', 6);
INSERT INTO Customer_CreditCard(CreditCardNumber, CustomerID)
VALUES ('3333333333333337', 7);
INSERT INTO Customer_CreditCard(CreditCardNumber, CustomerID)
VALUES ('3333333333333338', 8);
INSERT INTO Customer_CreditCard(CreditCardNumber, CustomerID)
VALUES ('3333333333333339', 9);
INSERT INTO Customer_CreditCard(CreditCardNumber, CustomerID)
VALUES ('3333333333333399', 10);
```

- **Table After Execution of SQL INSERT STATEMENTS:**

The following SQL SELECT Statement was used to list the content of the Customer_CreditCard:

```
SELECT * FROM Customer_CreditCard;
```

The final state of the Customer_CreditCard Table is:



```
SELECT * FROM Customer_CreditCard;
```

Script Output | Query Result 6 | SQL | All Rows Fetched: 11 in 0.002 seconds

	CREDITCARDNUMBER	CUSTOMERID
1	3333333333333331	1
2	3333333333333332	2
3	3333333333333332	3
4	3333333333333333	4
5	3333333333333334	4
6	3333333333333335	5
7	3333333333333336	6
8	3333333333333337	7
9	3333333333333338	8
10	3333333333333339	9
11	3333333333333399	10

12. Populating The USState Table

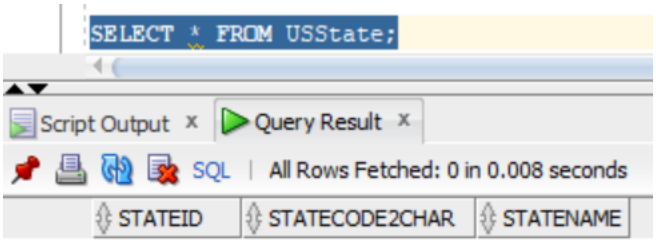
This section shows the content of USState Table before and after populating it with records using an SQL INSERT STATEMENTS.

- **Table Before IMPORT** USState**.csv file:**

The following SQL SELECT Statement was used to list the content of the USState table:

```
SELECT * FROM USState;
```

The current state of the USState Table is:

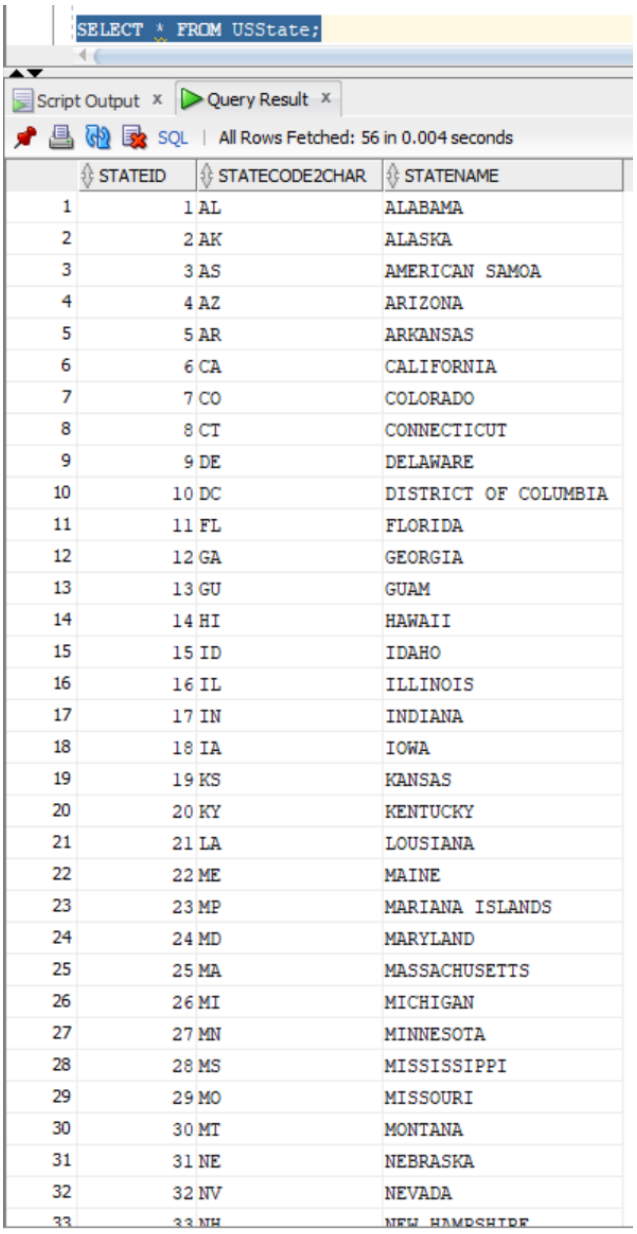


- **Table After IMPORT USState.csv file:**

The following SQL SELECT Statement was used to list the content of the USState:

```
SELECT * FROM USState;
```

The final state of the USState Table is:



	STATEID	STATECODE2CHAR	STATENAME
1	1	AL	ALABAMA
2	2	AK	ALASKA
3	3	AS	AMERICAN SAMOA
4	4	AZ	ARIZONA
5	5	AR	ARKANSAS
6	6	CA	CALIFORNIA
7	7	CO	COLORADO
8	8	CT	CONNECTICUT
9	9	DE	DELAWARE
10	10	DC	DISTRICT OF COLUMBIA
11	11	FL	FLORIDA
12	12	GA	GEORGIA
13	13	GU	GUAM
14	14	HI	HAWAII
15	15	ID	IDAHO
16	16	IL	ILLINOIS
17	17	IN	INDIANA
18	18	IA	IOWA
19	19	KS	KANSAS
20	20	KY	KENTUCKY
21	21	LA	LOUSIANA
22	22	ME	MAINE
23	23	MP	MARIANA ISLANDS
24	24	MD	MARYLAND
25	25	MA	MASSACHUSETTS
26	26	MI	MICHIGAN
27	27	MN	MINNESOTA
28	28	MS	MISSISSIPPI
29	29	MO	MISSOURI
30	30	MT	MONTANA
31	31	NE	NEBRASKA
32	32	NV	NEVADA
33	33	NH	NEW HAMPSHIRE

```
SELECT * FROM USState;
```

Script Output | Query Result | SQL | All Rows Fetched: 56 in 0.004 seconds

	STATEID	STATECODE2CHAR	STATENAME
32	32	NV	NEVADA
33	33	NH	NEW HAMPSHIRE
34	34	NJ	NEW JERSEY
35	35	NM	NEW MEXICO
36	36	NY	NEW YORK
37	37	NC	NORTH CAROLINA
38	38	ND	NORTH DAKOTA
39	39	OH	OHIO
40	40	OK	OKLAHOMA
41	41	OR	OREGON
42	42	PA	PENNSYLVANIA
43	43	PR	PUERTO RICO
44	44	RI	RHODE ISLAND
45	45	SC	SOUTH CAROLINA
46	46	SD	SOUTH DAKOTA
47	47	TN	TENNESSEE
48	48	TX	TEXAS
49	49	UT	UTAH
50	50	VT	VERMONT
51	51	VI	VIRGIN ISLANDS
52	52	VA	VIRGINIA
53	53	WA	WASHINGTON
54	54	WV	WEST VIRGINIA
55	55	WI	WISCONSIN
56	56	WY	WYOMING

13. Populating The Country Table

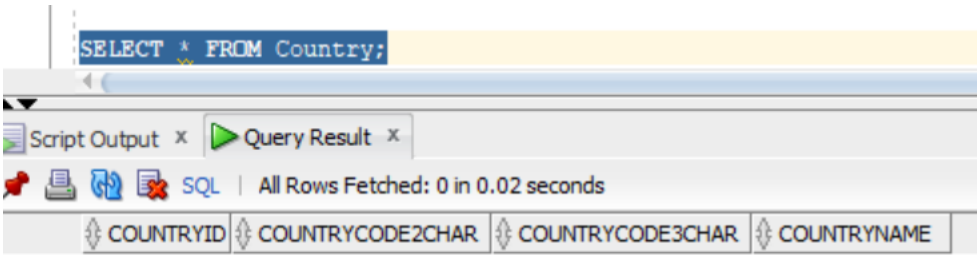
This section shows the content of Country Table before and after populating it with records using an SQL INSERT STATEMENTS.

- **Table Before Execution of SQL INSERT STATEMENTS:**

The following SQL SELECT Statement was used to list the content of the Country table:

```
SELECT * FROM Country;
```

The current state of the Country Table is:

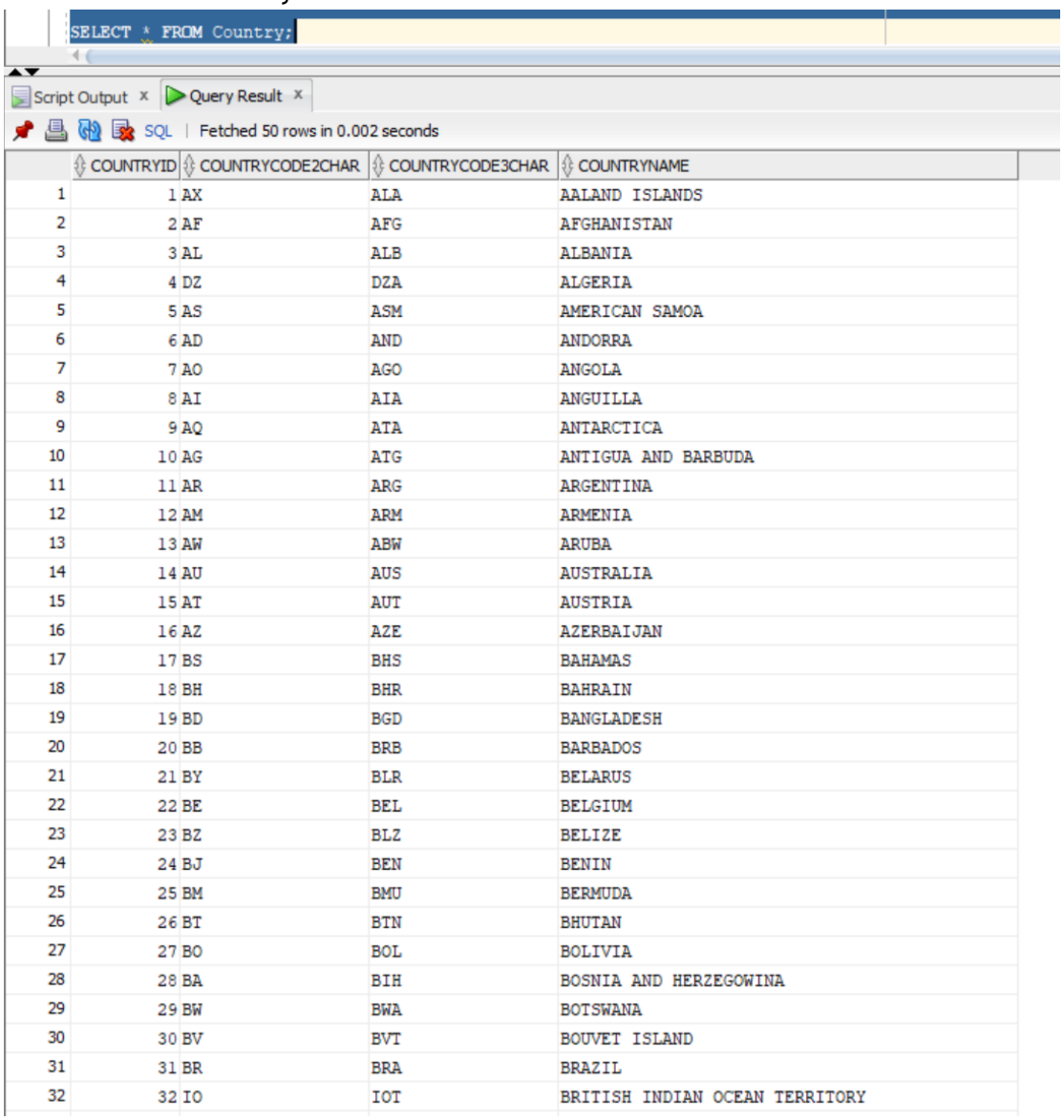


❖ **Table After Execution of SQL INSERT STATEMENTS:**

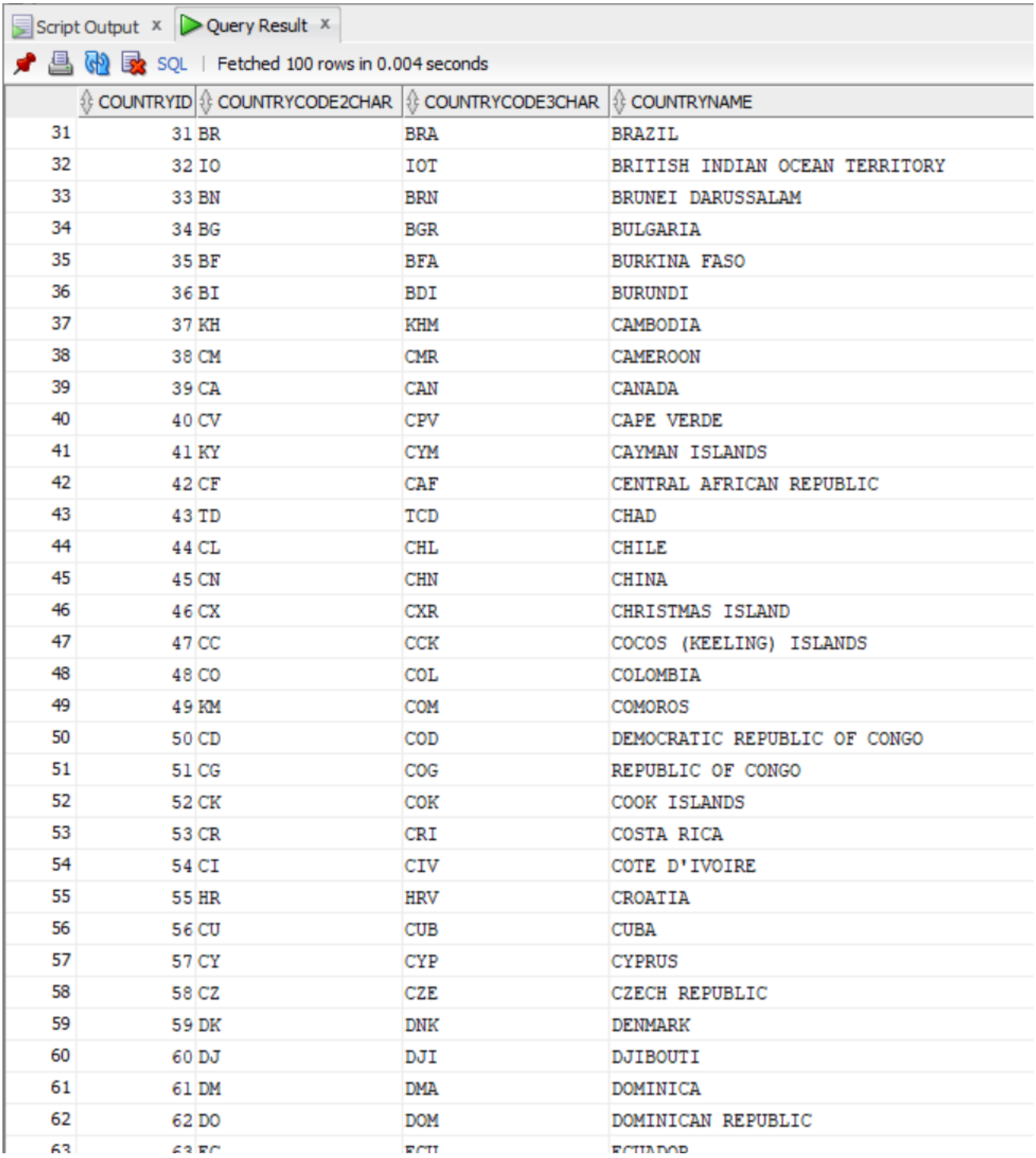
The following SQL SELECT Statement was used to list the content of the Country:

```
SELECT * FROM Country;
```

The final state of the Country Table is:



	COUNTRYID	COUNTRYCODE2CHAR	COUNTRYCODE3CHAR	COUNTRYNAME
1	1	AX	ALA	AALAND ISLANDS
2	2	AF	AFG	AFGHANISTAN
3	3	AL	ALB	ALBANIA
4	4	DZ	DZA	ALGERIA
5	5	AS	ASM	AMERICAN SAMOA
6	6	AD	AND	ANDORRA
7	7	AO	AGO	ANGOLA
8	8	AI	AIA	ANGUILLA
9	9	AQ	ATA	ANTARCTICA
10	10	AG	ATG	ANTIGUA AND BARBUDA
11	11	AR	ARG	ARGENTINA
12	12	AM	ARM	ARMENIA
13	13	AW	ABW	ARUBA
14	14	AU	AUS	AUSTRALIA
15	15	AT	AUT	AUSTRIA
16	16	AZ	AZE	AZERBAIJAN
17	17	BS	BHS	BAHAMAS
18	18	BH	BHR	BAHRAIN
19	19	BD	BGD	BANGLADESH
20	20	BB	BRB	BARBADOS
21	21	BY	BLR	BELARUS
22	22	BE	BEL	BELGIUM
23	23	BZ	BLZ	BELIZE
24	24	BJ	BEN	BENIN
25	25	BM	BMU	BERMUDA
26	26	BT	BTN	BHUTAN
27	27	BO	BOL	BOLIVIA
28	28	BA	BIH	BOSNIA AND HERZEGOWINA
29	29	BW	BWA	BOTSWANA
30	30	BV	BVT	BOUVET ISLAND
31	31	BR	BRA	BRAZIL
32	32	IO	IOT	BRITISH INDIAN OCEAN TERRITORY



Script Output | Query Result

SQL | Fetched 100 rows in 0.004 seconds

	COUNTRYID	COUNTRYCODE2CHAR	COUNTRYCODE3CHAR	COUNTRYNAME
31	31	BR	BRA	BRAZIL
32	32	IO	IOT	BRITISH INDIAN OCEAN TERRITORY
33	33	BN	BRN	BRUNEI DARUSSALAM
34	34	BG	BGR	BULGARIA
35	35	BF	BFA	BURKINA FASO
36	36	BI	BDI	BURUNDI
37	37	KH	KHM	CAMBODIA
38	38	CM	CMR	CAMEROON
39	39	CA	CAN	CANADA
40	40	CV	CPV	CAPE VERDE
41	41	KY	CYM	CAYMAN ISLANDS
42	42	CF	CAF	CENTRAL AFRICAN REPUBLIC
43	43	TD	TCD	CHAD
44	44	CL	CHL	CHILE
45	45	CN	CHN	CHINA
46	46	CX	CXR	CHRISTMAS ISLAND
47	47	CC	CCK	COCOS (KEELING) ISLANDS
48	48	CO	COL	COLOMBIA
49	49	KM	COM	COMOROS
50	50	CD	COD	DEMOCRATIC REPUBLIC OF CONGO
51	51	CG	COG	REPUBLIC OF CONGO
52	52	CK	COK	COOK ISLANDS
53	53	CR	CRI	COSTA RICA
54	54	CI	CIV	COTE D'IVOIRE
55	55	HR	HRV	CROATIA
56	56	CU	CUB	CUBA
57	57	CY	CYP	CYPRUS
58	58	CZ	CZE	CZECH REPUBLIC
59	59	DK	DNK	DENMARK
60	60	DJ	DJI	DJIBOUTI
61	61	DM	DMA	DOMINICA
62	62	DO	DOM	DOMINICAN REPUBLIC
63	63	EC	ECU	ECUADOR

Script Output | Query Result

SQL | Fetched 100 rows in 0.004 seconds

	COUNTRYID	COUNTRYCODE2CHAR	COUNTRYCODE3CHAR	COUNTRYNAME
61	61	DM	DMA	DOMINICA
62	62	DO	DOM	DOMINICAN REPUBLIC
63	63	EC	ECU	ECUADOR
64	64	EG	EGY	EGYPT
65	65	SV	SLV	EL SALVADOR
66	66	GQ	GNQ	EQUATORIAL GUINEA
67	67	ER	ERI	ERITREA
68	68	EE	EST	ESTONIA
69	69	ET	ETH	ETHIOPIA
70	70	FK	FLK	FALKLAND ISLANDS (MALVINAS)
71	71	FO	FRO	FAROE ISLANDS
72	72	FJ	FJI	FIJI
73	73	FI	FIN	FINLAND
74	74	FR	FRA	FRANCE
75	75	GF	GUF	FRENCH GUIANA
76	76	PF	PYF	FRENCH POLYNESIA
77	77	TF	ATF	FRENCH SOUTHERN TERRITORIES
78	78	GA	GAB	GABON
79	79	GM	GMB	GAMBIA
80	80	GE	GEO	GEORGIA
81	81	DE	DEU	GERMANY
82	82	GH	GHA	GHANA
83	83	GI	GIB	GIBRALTAR
84	84	GR	GRC	GREECE
85	85	GL	GRL	GREENLAND
86	86	GD	GRD	GRENADA
87	87	GP	GLP	GUADELOUPE
88	88	GU	GUM	GUAM
89	89	GT	GTM	GUATEMALA
90	90	GN	GIN	GUINEA
91	91	GW	GNB	GUINEA-BISSAU
92	92	GY	GUY	GUYANA
93	93	HT	HTI	HAITI

Script Output x | Query Result x

SQL | Fetched 150 rows in 0.007 seconds

	COUNTRYID	COUNTRYCODE2CHAR	COUNTRYCODE3CHAR	COUNTRYNAME
91	91	GW	GNB	GUINEA-BISSAU
92	92	GY	GUY	GUYANA
93	93	HT	HTI	HAITI
94	94	HM	HMD	HEARD AND MC DONALD ISLANDS
95	95	HN	HND	HONDURAS
96	96	HK	HKG	HONG KONG
97	97	HU	HUN	HUNGARY
98	98	IS	ISL	ICELAND
99	99	IN	IND	INDIA
100	100	ID	IDN	INDONESIA
101	101	IR	IRN	IRAN
102	102	IQ	IRQ	IRAQ
103	103	IE	IRL	IRELAND
104	104	IL	ISR	ISRAEL
105	105	IT	ITA	ITALY
106	106	JM	JAM	JAMAICA
107	107	JP	JPN	JAPAN
108	108	JO	JOR	JORDAN
109	109	KZ	KAZ	KAZAKHSTAN
110	110	KE	KEN	KENYA
111	111	KI	KIR	KIRIBATI
112	112	KP	PRK	DEMOCRATIC PEOPLE'S REPUBLIC OF KOREA
113	113	KR	KOR	REPUBLIC OF KOREA
114	114	KW	KWT	KUWAIT
115	115	KG	KGZ	KYRGYZSTAN
116	116	LA	LAO	LAO PEOPLE'S DEMOCRATIC REPUBLIC
117	117	LV	LVA	LATVIA
118	118	LB	LBN	LEBANON
119	119	LS	LSO	LESOTHO
120	120	LR	LBR	LIBERIA
121	121	LY	LBY	LIBYAN ARAB JAMAHIRIYA
122	122	LI	LIE	LIECHTENSTEIN
123	123	LT	LTU	LITHUANIA

Script Output x | Query Result x

SQL | Fetched 200 rows in 0.009 seconds

	COUNTRYID	COUNTRYCODE2CHAR	COUNTRYCODE3CHAR	COUNTRYNAME
121	121	LY	LBY	LIBYAN ARAB JAMAHIRIYA
122	122	LI	LIE	LIECHTENSTEIN
123	123	LT	LTU	LITHUANIA
124	124	LU	LUX	LUXEMBOURG
125	125	MO	MAC	MACAU
126	126	MK	MKD	THE FORMER YUGOSLAV REPUBLIC OF MACEDONIA
127	127	MG	MDG	MADAGASCAR
128	128	MW	MWI	MALAWI
129	129	MY	MYS	MALAYSIA
130	130	MV	MDV	MALDIVES
131	131	ML	MLI	MALI
132	132	MT	MLT	MALTA
133	133	MH	MHL	MARSHALL ISLANDS
134	134	MQ	MTQ	MARTINIQUE
135	135	MR	MRT	MAURITANIA
136	136	MU	MUS	MAURITIUS
137	137	YT	MYT	MAYOTTE
138	138	MX	MEX	MEXICO
139	139	FM	FSM	FEDERATED STATES OF MICRONESIA
140	140	MD	MDA	REPUBLIC OF MOLDOVA
141	141	MC	MCO	MONACO
142	142	MN	MNG	MONGOLIA
143	143	MS	MSR	MONTSERRAT
144	144	MA	MAR	MOROCCO
145	145	MZ	MOZ	MOZAMBIQUE
146	146	MM	MMR	MYANMAR
147	147	NA	NAM	NAMIBIA
148	148	NR	NRU	NAURU
149	149	NP	NPL	NEPAL
150	150	NL	NLD	NETHERLANDS
151	151	AN	ANT	NETHERLANDS ANTILLES
152	152	NC	NCL	NEW CALEDONIA
153	153	NZ	NZL	NEW ZEALAND

Script Output x | Query Result x

SQL | Fetched 200 rows in 0.009 seconds

	COUNTRYID	COUNTRYCODE2CHAR	COUNTRYCODE3CHAR	COUNTRYNAME
151	151	AN	ANT	NETHERLANDS ANTILLES
152	152	NC	NCL	NEW CALEDONIA
153	153	NZ	NZL	NEW ZEALAND
154	154	NI	NIC	NICARAGUA
155	155	NE	NER	NIGER
156	156	NG	NGA	NIGERIA
157	157	NU	NIU	NIUE
158	158	NF	NFK	NORFOLK ISLAND
159	159	MP	MNP	NORTHERN MARIANA ISLANDS
160	160	NO	NOR	NORWAY
161	161	OM	OMN	OMAN
162	162	PK	PAK	PAKISTAN
163	163	PW	PLW	PALAU
164	164	PS	PSE	PALESTINIAN TERRITORY
165	165	PA	PAN	PANAMA
166	166	PG	PNG	PAPUA NEW GUINEA
167	167	PY	PRY	PARAGUAY
168	168	PE	PER	PERU
169	169	PH	PHL	PHILIPPINES
170	170	PN	PCN	PITCAIRN
171	171	PL	POL	POLAND
172	172	PT	PRT	PORTUGAL
173	173	PR	PRI	PUERTO RICO
174	174	QA	QAT	QATAR
175	175	RE	REU	REUNION
176	176	RO	ROU	ROMANIA
177	177	RU	RUS	RUSSIAN FEDERATION
178	178	RW	RWA	RWANDA
179	179	SH	SHN	SAINT HELENA
180	180	KN	KNA	SAINT KITTS AND NEVIS
181	181	LC	LCA	SAINT LUCIA
182	182	PM	SPM	SAINT PIERRE AND MIQUELON

Script Output | Query Result

SQL | All Rows Fetched: 240 in 0.009 seconds

	COUNTRYID	COUNTRYCODE2CHAR	COUNTRYCODE3CHAR	COUNTRYNAME
181	181	LC	LCA	SAINT LUCIA
182	182	PM	SPM	SAINT PIERRE AND MIQUELON
183	183	VC	VCT	SAINT VINCENT AND THE GRENADINES
184	184	WS	WSM	SAMOA
185	185	SM	SMR	SAN MARINO
186	186	ST	STP	SAO TOME AND PRINCIPE
187	187	SA	SAU	SAUDI ARABIA
188	188	SN	SEN	SENEGAL
189	189	CS	SCG	SERBIA AND MONTENEGRO
190	190	SC	SYC	SEYCHELLES
191	191	SL	SLE	SIERRA LEONE
192	192	SG	SGP	SINGAPORE
193	193	SK	SVK	SLOVAKIA
194	194	SI	SVN	SLOVENIA
195	195	SB	SLB	SOLOMON ISLANDS
196	196	SO	SOM	SOMALIA
197	197	ZA	ZAF	SOUTH AFRICA
198	198	GS	SGS	SOUTH GEORGIA AND THE SOUTH SANDWICH ISLANDS
199	199	ES	ESP	SPAIN
200	200	LK	LKA	SRI LANKA
201	201	SD	SDN	SUDAN
202	202	SR	SUR	SURINAME
203	203	SJ	SJM	SVALBARD AND JAN MAYEN ISLANDS
204	204	SZ	SWZ	SWAZILAND
205	205	SE	SWE	SWEDEN
206	206	CH	CHE	SWITZERLAND
207	207	SY	SYR	SYRIAN ARAB REPUBLIC
208	208	TW	TWN	TAIWAN
209	209	TJ	TJK	TAJIKISTAN
210	210	TZ	TZA	UNITED REPUBLIC OF TANZANIA
211	211	TH	THA	THAILAND
212	212	TL	TLS	TIMOR-LESTE
213	213	TG	TGO	TOGO

Script Output x | Query Result x

SQL | All Rows Fetched: 240 in 0.009 seconds

	COUNTRYID	COUNTRYCODE2CHAR	COUNTRYCODE3CHAR	COUNTRYNAME
208	208	TW	TWN	TAIWAN
209	209	TJ	TJK	TAJIKISTAN
210	210	TZ	TZA	UNITED REPUBLIC OF TANZANIA
211	211	TH	THA	THAILAND
212	212	TL	TLS	TIMOR-LESTE
213	213	TG	TGO	TOGO
214	214	TK	TKL	TOKELAU
215	215	TO	TON	TONGA
216	216	TT	TTO	TRINIDAD AND TOBAGO
217	217	TN	TUN	TUNISIA
218	218	TR	TUR	TURKEY
219	219	TM	TKM	TURKMENISTAN
220	220	TC	TCA	TURKS AND CAICOS ISLANDS
221	221	TV	TUV	TUVALU
222	222	UG	UGA	UGANDA
223	223	UA	UKR	UKRAINE
224	224	AE	ARE	UNITED ARAB EMIRATES
225	225	GB	GBR	UNITED KINGDOM
226	226	US	USA	UNITED STATES
227	227	UM	UMI	UNITED STATES MINOR OUTLYING ISLANDS
228	228	UY	URY	URUGUAY
229	229	UZ	UZB	UZBEKISTAN
230	230	VU	VUT	VANUATU
231	231	VA	VAT	VATICAN CITY STATE (HOLY SEE)
232	232	VE	VEN	VENEZUELA
233	233	VN	VNM	VIET NAM
234	234	VG	VGB	VIRGIN ISLANDS (BRITISH)
235	235	VI	VIR	VIRGIN ISLANDS (U.S.)
236	236	WF	WLF	WALLIS AND FUTUNA ISLANDS
237	237	EH	ESH	WESTERN SAHARA
238	238	YE	YEM	YEMEN
239	239	ZM	ZMB	ZAMBIA
240	240	ZW	ZWE	ZIMBABWE

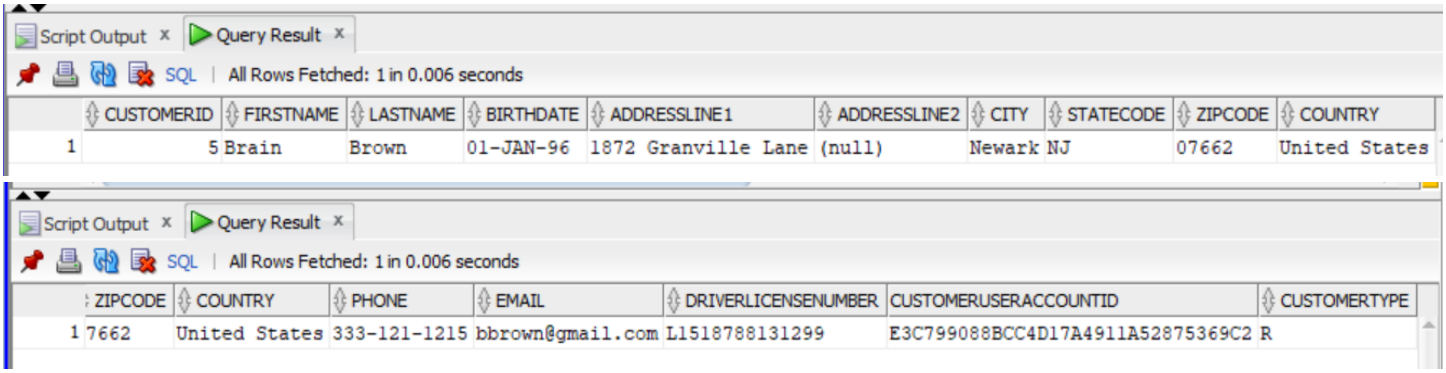
TESTING QUERY: SELECT STATEMENT#1

Write a query of select statement using WHERE clause to return a customer record of Customer_ID is 5.

Query:

```
SELECT * FROM CUSTOMER WHERE CustomerID = 5;
```

Screenshot:



Script Output | Query Result

SQL | All Rows Fetched: 1 in 0.006 seconds

	CUSTOMERID	FIRSTNAME	LASTNAME	BIRTHDATE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY
1	5	Brain	Brown	01-JAN-96	1872 Granville Lane	(null)	Newark	NJ	07662	United States

Script Output | Query Result

SQL | All Rows Fetched: 1 in 0.006 seconds

	ZIPCODE	COUNTRY	PHONE	EMAIL	DRIVERLICENSENUMBER	CUSTOMERUSERACCOUNTID	CUSTOMERTYPE
1	7662	United States	333-121-1215	bbrown@gmail.com	L1518788131299	E3C799088BCC4D17A4911A52875369C2	R

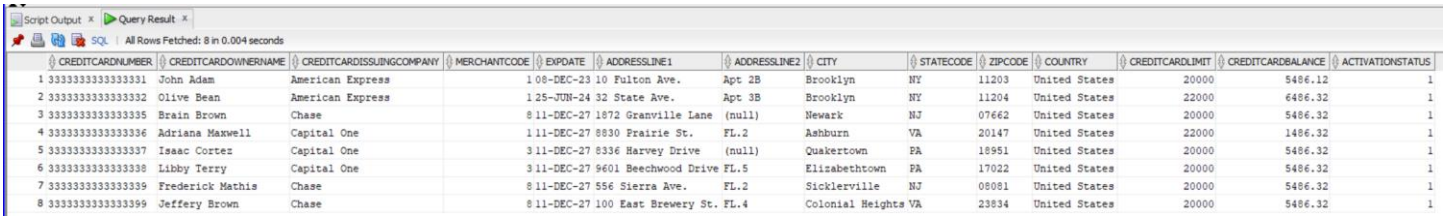
TESTING QUERY: SELECT STATEMENT#2

Write a query to retrieve the record(s) that is equal or greater than 20,000.

Query:

```
SELECT * FROM CreditCard
WHERE CreditCardLimit >= 20000;
```

Screenshot:



Script Output | Query Result

SQL | All Rows Fetched: 8 in 0.004 seconds

	CREDITCARDNUMBER	CREDITCARDOWNERNAME	CREDITCARDISSUINGCOMPANY	MERCHANTCODE	EXPDATE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY	CREDITCARDLIMIT	CREDITCARDBALANCE	ACTIVATIONSTATUS
1	3333333333333331	John Adam	American Express	1	08-DEC-23	10 Fulton Ave.	Apt 2B	Brooklyn	NY	11203	United States	20000	5486.12	1
2	3333333333333332	Olive Bean	American Express	1	25-JUN-24	32 State Ave.	Apt 3B	Brooklyn	NY	11204	United States	22000	6486.32	1
3	3333333333333335	Brain Brown	Chase	8	11-DEC-27	1872 Granville Lane	(null)	Newark	NJ	07662	United States	20000	5486.32	1
4	3333333333333336	Adriana Maxwell	Capital One	1	11-DEC-27	8830 Prairie St.	FL.2	Ashburn	VA	20147	United States	22000	1486.32	1
5	3333333333333337	Isaac Cortez	Capital One	3	11-DEC-27	8336 Harvey Drive	(null)	Quakertown	PA	18951	United States	20000	5486.32	1
6	3333333333333338	Libby Terry	Capital One	3	11-DEC-27	9601 Beechwood Drive	FL.5	Elizabethtown	PA	17022	United States	20000	5486.32	1
7	3333333333333339	Frederick Mathis	Chase	8	11-DEC-27	556 Sierra Ave.	FL.2	Sicklerville	NJ	08081	United States	20000	5486.32	1
8	3333333333333399	Jeffery Brown	Chase	8	11-DEC-27	100 East Brewery St.	FL.4	Colonial Heights	VA	23834	United States	20000	5486.32	1

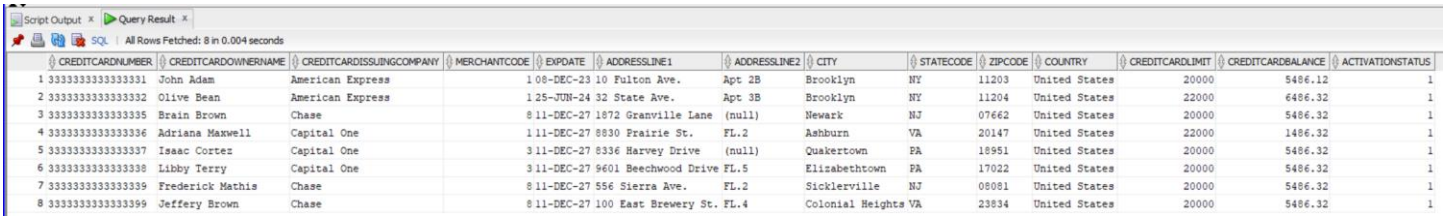
TESTING QUERY: SELECT STATEMENT#3

Write a query to retrieve the credit card number, credit card issuing company of a customer, and correspond to the customer ID, first name, and last name.

Query:

```
SELECT CreditCard.CreditCardNumber, CreditCardIssuingCompany, Customer.CustomerID, FirstName, LastName
FROM CreditCard, Customer_CreditCard, Customer
WHERE CreditCard.CreditCardNumber = Customer_CreditCard. CreditCardNumber
AND Customer_CreditCard.CustomerID = Customer. CustomerID;
```

Screenshot:



Script Output | Query Result | All Rows Fetched: 8 in 0.004 seconds

	CREDITCARDNUMBER	CREDITCARDOWNERNAME	CREDITCARDISSUINGCOMPANY	MERCHANTCODE	EXPDATE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY	CREDITCARDLIMIT	CREDITCARDBALANCE	ACTIVATIONSTATUS
1	3333333333333331	John Adam	American Express	1	08-DEC-23	10 Fulton Ave.	Apt 2B	Brooklyn	NY	11203	United States	20000	5486.12	1
2	3333333333333332	Olive Bean	American Express	1	25-JUN-24	32 State Ave.	Apt 3B	Brooklyn	NY	11204	United States	22000	6486.32	1
3	3333333333333335	Brain Brown	Chase	8	11-DEC-27	1872 Granville Lane	(null)	Newark	NJ	07662	United States	20000	5486.32	1
4	3333333333333336	Adriana Maxwell	Capital One	1	11-DEC-27	8830 Prairie St.	FL.2	Ashburn	VA	20147	United States	22000	1486.32	1
5	3333333333333337	Isaac Cortez	Capital One	3	11-DEC-27	8336 Harvey Drive	(null)	Quakertown	PA	18951	United States	20000	5486.32	1
6	3333333333333338	Libby Terry	Capital One	3	11-DEC-27	9601 Beechwood Drive	FL.5	Elizabethtown	PA	17022	United States	20000	5486.32	1
7	3333333333333339	Frederick Mathis	Chase	8	11-DEC-27	556 Sierra Ave.	FL.2	Sicklerville	NJ	08081	United States	20000	5486.32	1
8	3333333333333399	Jeffery Brown	Chase	8	11-DEC-27	100 East Brewery St.	FL.4	Colonial Heights	VA	23834	United States	20000	5486.32	1

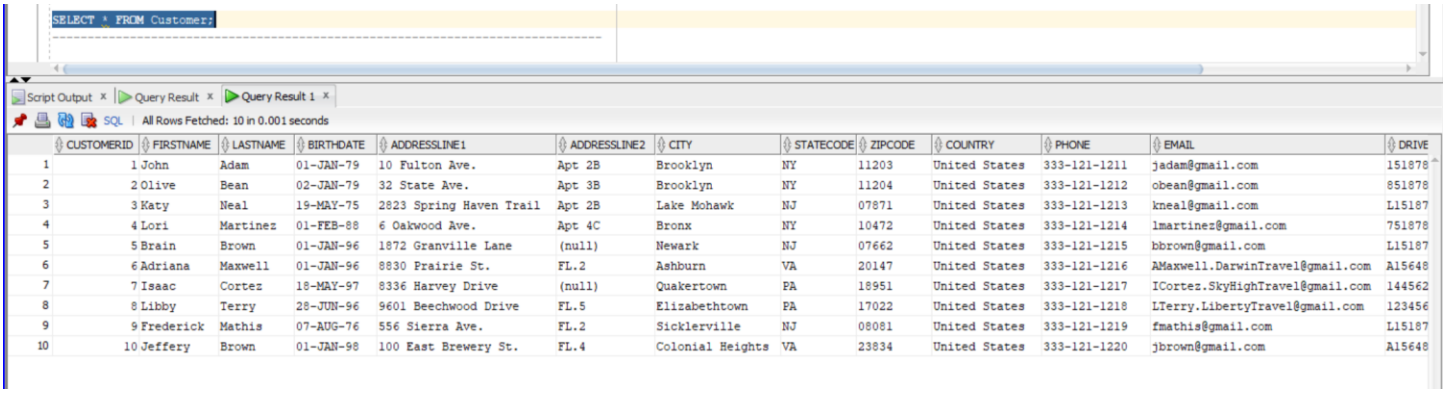
Script Output | Query Result | SQL | All Rows Fetched: 11 in 0.007 seconds

	CREDITCARDNUMBER	CREDITCARDISSUINGCOMPANY	CUSTOMERID	FIRSTNAME	LASTNAME
1	3333333333333331	American Express	1	John	Adam
2	3333333333333332	American Express	2	Olive	Bean
3	3333333333333332	American Express	3	Katy	Neal
4	3333333333333333	Chase	4	Lori	Martinez
5	3333333333333334	Capital One	4	Lori	Martinez
6	3333333333333335	Chase	5	Brain	Brown
7	3333333333333336	Capital One	6	Adriana	Maxwell
8	3333333333333337	Capital One	7	Isaac	Cortez
9	3333333333333338	Capital One	8	Libby	Terry
10	3333333333333339	Chase	9	Frederick	Mathis
11	3333333333333399	Chase	10	Jeffery	Brown

TESTING QUERY: UPDATE STATEMENT#1

Write a query to update all columns of one record on CustomerID = 3.

Before executing the query of Customer table:

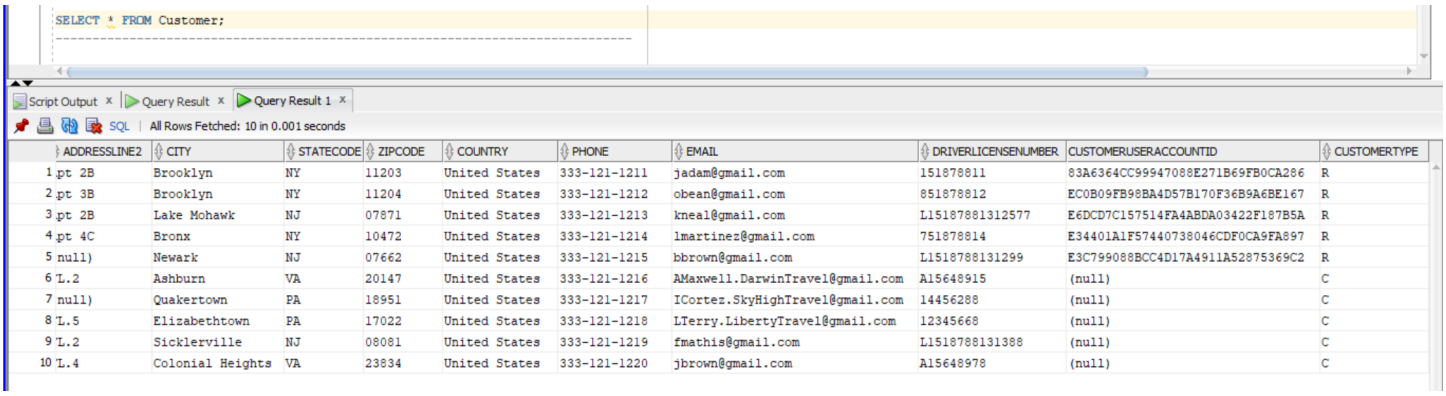


```
SELECT * FROM Customer;
```

Script Output | Query Result | Query Result 1

All Rows Fetched: 10 in 0.001 seconds

	CUSTOMERID	FIRSTNAME	LASTNAME	BIRTHDATE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY	PHONE	EMAIL	DRIVE
1	1	John	Adam	01-JAN-79	10 Fulton Ave.	Apt 2B	Brooklyn	NY	11203	United States	333-121-1211	jadam@gmail.com	151878
2	2	Olive	Bean	02-JAN-79	32 State Ave.	Apt 3B	Brooklyn	NY	11204	United States	333-121-1212	obean@gmail.com	851878
3	3	Katy	Neal	19-MAY-75	2823 Spring Haven Trail	Apt 2B	Lake Mohawk	NJ	07871	United States	333-121-1213	kneal@gmail.com	L15187
4	4	Lori	Martinez	01-FEB-88	6 Oakwood Ave.	Apt 4C	Bronx	NY	10472	United States	333-121-1214	lmartinez@gmail.com	751878
5	5	Brain	Brown	01-JAN-96	1872 Granville Lane	(null)	Newark	NJ	07662	United States	333-121-1215	bbrown@gmail.com	L15187
6	6	Adriana	Maxwell	01-JAN-96	8830 Prairie St.	FL.2	Ashburn	VA	20147	United States	333-121-1216	AMaxwell.DarwinTravel@gmail.com	A15648
7	7	Isaac	Cortez	18-MAY-97	8336 Harvey Drive	(null)	Quakertown	PA	18951	United States	333-121-1217	ICortez.SkyHighTravel@gmail.com	144562
8	8	Libby	Terry	28-JUN-96	9601 Beechwood Drive	FL.5	Elizabethtown	PA	17022	United States	333-121-1218	LTerry.LibertyTravel@gmail.com	123456
9	9	Frederick	Mathis	07-AUG-76	556 Sierra Ave.	FL.2	Sicklerville	NJ	08081	United States	333-121-1219	fmathis@gmail.com	L15187
10	10	Jeffery	Brown	01-JAN-98	100 East Brewery St.	FL.4	Colonial Heights	VA	23834	United States	333-121-1220	jbrown@gmail.com	A15648



```
SELECT * FROM Customer;
```

Script Output | Query Result | Query Result 1

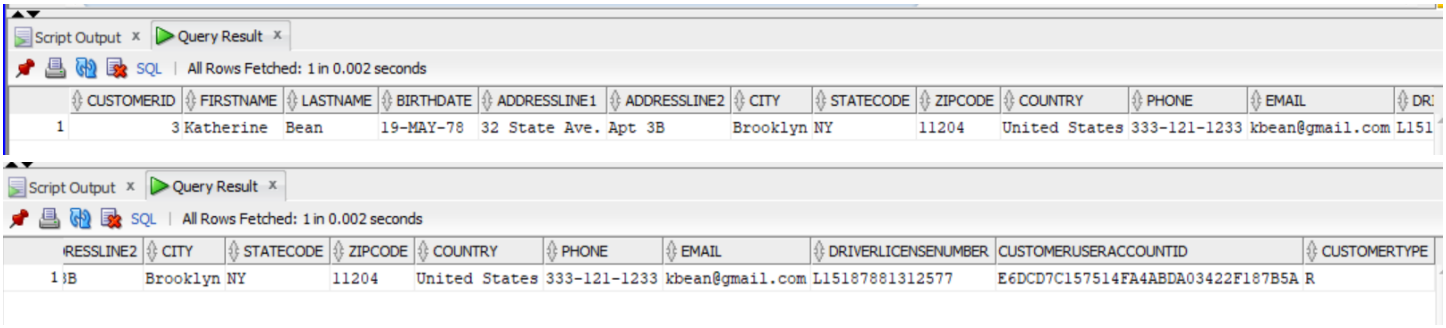
All Rows Fetched: 10 in 0.001 seconds

	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY	PHONE	EMAIL	DRIVERLICENSENUMBER	CUSTOMERUSERACCOUNTID	CUSTOMERTYPE
1	pt 2B	Brooklyn	NY	11203	United States	333-121-1211	jadam@gmail.com	151878811	83A6364CC99947088E271B69FB0CA286	R
2	pt 3B	Brooklyn	NY	11204	United States	333-121-1212	obean@gmail.com	851878812	EC0B09FB98BA4D57B170F36B9A6BE167	R
3	pt 2B	Lake Mohawk	NJ	07871	United States	333-121-1213	kneal@gmail.com	L15187881312577	E6DCD7C157514FA4ABDA03422F187B5A	R
4	pt 4C	Bronx	NY	10472	United States	333-121-1214	lmartinez@gmail.com	751878814	E34401A1F57440738046CDF0CA9FA897	R
5	null)	Newark	NJ	07662	United States	333-121-1215	bbrown@gmail.com	L1518788131299	E3C799088BCC4D17A4911A52875369C2	R
6	L.2	Ashburn	VA	20147	United States	333-121-1216	AMaxwell.DarwinTravel@gmail.com	A15648915	(null)	C
7	null)	Quakertown	PA	18951	United States	333-121-1217	ICortez.SkyHighTravel@gmail.com	14456288	(null)	C
8	L.5	Elizabethtown	PA	17022	United States	333-121-1218	LTerry.LibertyTravel@gmail.com	12345668	(null)	C
9	L.2	Sicklerville	NJ	08081	United States	333-121-1219	fmathis@gmail.com	L1518788131388	(null)	C
10	L.4	Colonial Heights	VA	23834	United States	333-121-1220	jbrown@gmail.com	A15648978	(null)	C

Query:

```
UPDATE Customer
SET FirstName = 'Katherine',
        LastName = 'Bean',
        BirthDate = '19-MAY-78',
        AddressLine1 = '32 State Ave.',
        AddressLine2 = 'Apt 3B',
        City = 'Brooklyn',
        StateCode= 'NY',
        Zipcode='11204',
        Country = 'United States',
        Phone = '333-121-1233',
        Email = 'kbean@gmail.com',
        DriverLicenseNumber = 'L15187881312577',
        CustomerUserAccountID = 'E6DCD7C157514FA4ABDA03422F187B5A',
        CustomerType='R'
WHERE CustomerID = 3;

SELECT * FROM CUSTOMER WHERE CustomerID = 3;
```

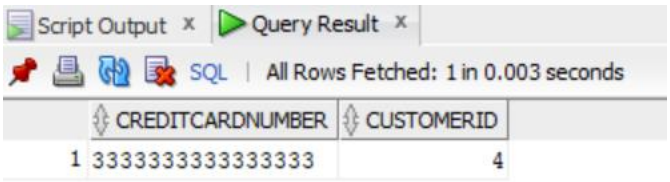
After executing the query to update the information of customerID = 3:



TESTING QUERY: UPDATE STATEMENT#2

Write a query to change creditcard number from 3333333333333333 to 3333333333333377 for CustomerID is 4.

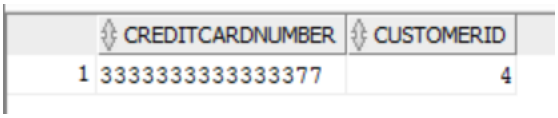
Before executing the query:



Query:

```
INSERT INTO CreditCard(CreditCardNumber, CreditCardOwnerName, CreditCardIssuingCompany, MerchantCode,
ExpDate, AddressLine1, AddressLine2, City, StateCode, Zipcode, Country, CreditCardLimit, CreditCardBalance,
ActivationStatus)
VALUES ('3333333333333377', 'Lori Martinez', 'Capital One', 8, '25-MAY-2025', '6 Oakwood Ave.', 'Apt 4C', 'Bronx',
'NY', '10472', 'United States', 18000.00, 286.12, '1');

UPDATE Customer_Creditcard
SET CreditcardNumber = '3333333333333377'
WHERE CreditcardNumber = '3333333333333333' and CustomerID = 4;

SELECT * FROM Customer_Creditcard
WHERE CreditcardNumber = '3333333333333377' and CustomerID = 4;
```

After executing the query:



TESTING QUERY: DELETE STATEMENT#1

Delete the record of the credit card number is 3333333333333333 in the CreditCard table.

Before executing the query:

Script Output | Query Result

All Rows Fetched: 11 in 0.005 seconds

	CREDITCARDNUMBER	CREDITCARDOWNERNAME	CREDITCARDISSUINGCOMPANY	MERCHANTCODE	EXPDATE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY	CREDITCARDLIMIT	CREDITCARDBALANCE	ACTIVATIONSTATUS
1	3333333333333331	John Adam	American Express	1	08-DEC-23	10 Fulton Ave.	Apt 2B	Brooklyn	NY	11203	United States	20000	5486.12	1
2	3333333333333332	Olive Bean	American Express	1	25-JUN-24	32 State Ave.	Apt 3B	Brooklyn	NY	11204	United States	22000	6486.32	1
3	3333333333333333	Lori Martinez	Chase	8	18-DEC-27	6 Oakwood Ave.	Apt 4C	Bronx	NY	10472	United States	15000	6286.12	1
4	3333333333333334	Lori Martinez	Capital One	8	25-MAY-25	6 Oakwood Ave.	Apt 4C	Bronx	NY	10472	United States	18000	286.12	1
5	3333333333333335	Brain Brown	Chase	8	11-DEC-27	1872 Granville Lane	(null)	Newark	NJ	07662	United States	20000	5486.32	1
6	3333333333333336	Adriana Maxwell	Capital One	1	11-DEC-27	8830 Prairie St.	FL.2	Ashburn	VA	20147	United States	22000	1486.32	1
7	3333333333333337	Isaac Cortez	Capital One	3	11-DEC-27	8336 Harvey Drive	(null)	Quakertown	PA	18951	United States	20000	5486.32	1
8	3333333333333338	Libby Terry	Capital One	3	11-DEC-27	9601 Beechwood Drive	FL.5	Elizabethtown	PA	17022	United States	20000	5486.32	1
9	3333333333333339	Frederick Mathis	Chase	8	11-DEC-27	556 Sierra Ave.	FL.2	Sicklerville	NJ	08081	United States	20000	5486.32	1
10	3333333333333399	Jeffery Brown	Chase	8	11-DEC-27	100 East Brewery St.	FL.4	Colonial Heights	VA	23834	United States	20000	5486.32	1
11	3333333333333377	Lori Martinez	Capital One	8	25-MAY-25	6 Oakwood Ave.	Apt 4C	Bronx	NY	10472	United States	18000	286.12	1

Query:

```
DELETE FROM CreditCard
WHERE CreditcardNumber = '3333333333333333';

SELECT *FROM CreditCard;
```

After executing the query:

Script Output | Query Result

All Rows Fetched: 10 in 0.001 seconds

	CREDITCARDNUMBER	CREDITCARDOWNERNAME	CREDITCARDISSUINGCOMPANY	MERCHANTCODE	EXPDATE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY	CREDITCARDLIMIT	CREDITCARDBALANCE	ACTIVATIONSTATUS
1	3333333333333331	John Adam	American Express	1	08-DEC-23	10 Fulton Ave.	Apt 2B	Brooklyn	NY	11203	United States	20000	5486.12	1
2	3333333333333332	Olive Bean	American Express	1	25-JUN-24	32 State Ave.	Apt 3B	Brooklyn	NY	11204	United States	22000	6486.32	1
3	3333333333333334	Lori Martinez	Capital One	8	25-MAY-25	6 Oakwood Ave.	Apt 4C	Bronx	NY	10472	United States	18000	286.12	1
4	3333333333333335	Brain Brown	Chase	8	11-DEC-27	1872 Granville Lane	(null)	Newark	NJ	07662	United States	20000	5486.32	1
5	3333333333333336	Adriana Maxwell	Capital One	1	11-DEC-27	8830 Prairie St.	FL.2	Ashburn	VA	20147	United States	22000	1486.32	1
6	3333333333333337	Isaac Cortez	Capital One	3	11-DEC-27	8336 Harvey Drive	(null)	Quakertown	PA	18951	United States	20000	5486.32	1
7	3333333333333338	Libby Terry	Capital One	3	11-DEC-27	9601 Beechwood Drive	FL.5	Elizabethtown	PA	17022	United States	20000	5486.32	1
8	3333333333333339	Frederick Mathis	Chase	8	11-DEC-27	556 Sierra Ave.	FL.2	Sicklerville	NJ	08081	United States	20000	5486.32	1
9	3333333333333399	Jeffery Brown	Chase	8	11-DEC-27	100 East Brewery St.	FL.4	Colonial Heights	VA	23834	United States	20000	5486.32	1
10	3333333333333377	Lori Martinez	Capital One	8	25-MAY-25	6 Oakwood Ave.	Apt 4C	Bronx	NY	10472	United States	18000	286.12	1

TESTING QUERY: DELETE STATEMENT#2

Delete the record of the credit card number is 3333333333333377 on the CreditCard table and then delete on the Customer_CreditCard table.

The CreditCard table before executing the query:

Script Output | Query Result

All Rows Fetched: 10 in 0.001 seconds

	CREDITCARDNUMBER	CREDITCARDOWNERNAME	CREDITCARDISSUINGCOMPANY	MERCHANTCODE	EXPDATE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY	CREDITCARDLIMIT	CREDITCARDBALANCE	ACTIVATIONSTATUS
1	3333333333333331	John Adam	American Express	1	08-DEC-23	10 Fulton Ave.	Apt 2B	Brooklyn	NY	11203	United States	20000	5486.12	1
2	3333333333333332	Olive Bean	American Express	1	25-JUN-24	32 State Ave.	Apt 3B	Brooklyn	NY	11204	United States	22000	6486.32	1
3	3333333333333334	Lori Martinez	Capital One	8	25-MAY-25	6 Oakwood Ave.	Apt 4C	Bronx	NY	10472	United States	18000	286.12	1
4	3333333333333335	Brain Brown	Chase	8	11-DEC-27	1872 Granville Lane	(null)	Newark	NJ	07662	United States	20000	5486.32	1
5	3333333333333336	Adriana Maxwell	Capital One	1	11-DEC-27	8830 Prairie St.	FL.2	Ashburn	VA	20147	United States	22000	1486.32	1
6	3333333333333337	Isaac Cortez	Capital One	3	11-DEC-27	8336 Harvey Drive	(null)	Quakertown	PA	18951	United States	20000	5486.32	1
7	3333333333333338	Libby Terry	Capital One	3	11-DEC-27	9601 Beechwood Drive	FL.5	Elizabethtown	PA	17022	United States	20000	5486.32	1
8	3333333333333339	Frederick Mathis	Chase	8	11-DEC-27	556 Sierra Ave.	FL.2	Sicklerville	NJ	08081	United States	20000	5486.32	1
9	3333333333333399	Jeffery Brown	Chase	8	11-DEC-27	100 East Brewery St.	FL.4	Colonial Heights	VA	23834	United States	20000	5486.32	1
10	3333333333333377	Lori Martinez	Capital One	8	25-MAY-25	6 Oakwood Ave.	Apt 4C	Bronx	NY	10472	United States	18000	286.12	1

The Customer_CreditCard table before executing the query:

Script Output | Query Result

SQL | All Rows Fetched: 11 in 0.004 seconds

	CREDITCARDNUMBER	CUSTOMERID
1	3333333333333331	1
2	3333333333333332	2
3	3333333333333332	3
4	3333333333333334	4
5	3333333333333335	5
6	3333333333333336	6
7	3333333333333337	7
8	3333333333333338	8
9	3333333333333339	9
10	3333333333333377	4
11	3333333333333399	10

Query:

```
DELETE FROM CreditCard
WHERE CreditcardNumber = '3333333333333377';

--No need to delete further in the Customer_CreditCard table.
--DELETE FROM Customer_CreditCard
--WHERE CreditcardNumber = '3333333333333377' and CustomerID = 4;

SELECT *FROM CreditCard;
SELECT *FROM Customer_CreditCard;
```

The CreditCard table after executing the query:

Query Result | Query Result 1 | Script Output | Query Result 2 | Query Result 3

SQL | All Rows Fetched: 9 in 0.001 seconds

	CREDITCARDNUMBER	CREDITCARDOWNERNAME	CREDITCARDISSUINGCOMPANY	MERCHANTCODE	EXPDATE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY	CREDITCARDLIMIT	CREDITCARDBALANCE	ACTIVATIONSTATUS
1	3333333333333331	John Adam	American Express	1	08-DEC-23	10 Fulton Ave.	Apt 2B	Brooklyn	NY	11203	United States	20000	5486.12	1
2	3333333333333332	Olive Bean	American Express	1	25-JUN-24	32 State Ave.	Apt 3B	Brooklyn	NY	11204	United States	22000	6486.32	1
3	3333333333333334	Lori Martinez	Capital One	8	25-MAY-25	6 Oakwood Ave.	Apt 4C	Bronx	NY	10472	United States	18000	286.12	1
4	3333333333333335	Brain Brown	Chase	8	11-DEC-27	1872 Granville Lane	(null)	Newark	NJ	07662	United States	20000	5486.32	1
5	3333333333333336	Adriana Maxwell	Capital One	1	11-DEC-27	8830 Prairie St.	FL.2	Ashburn	VA	20147	United States	22000	1486.32	1
6	3333333333333337	Isaac Cortez	Capital One	3	11-DEC-27	8336 Harvey Drive	(null)	Quakertown	PA	18951	United States	20000	5486.32	1
7	3333333333333338	Libby Terry	Capital One	3	11-DEC-27	9601 Beechwood Drive	FL.5	Elizabethtown	PA	17022	United States	20000	5486.32	1
8	3333333333333339	Frederick Mathis	Chase	8	11-DEC-27	556 Sierra Ave.	FL.2	Sicklerville	NJ	08081	United States	20000	5486.32	1
9	3333333333333399	Jeffery Brown	Chase	8	11-DEC-27	100 East Brewery St.	FL.4	Colonial Heights	VA	23834	United States	20000	5486.32	1

The Customer_CreditCard table before executing the query:

Query Result | Query Result 1 | Script

SQL | All Rows Fetched: 10 in 0.0

	CREDITCARDNUMBER	CUSTOMERID
1	3333333333333331	1
2	3333333333333332	2
3	3333333333333332	3
4	3333333333333334	4
5	3333333333333335	5
6	3333333333333336	6
7	3333333333333337	7
8	3333333333333338	8
9	3333333333333339	9
10	3333333333333399	10

Conclusion

The purpose of the project is to develop a Database Management System for the company EZ-Car Rental and ensure functions correctly and meets the requirements of the system it is being used for. The process involves creating a schema, defining tables, columns, and relationships, and populating the database with data.

In this project, a pilot of 13 tables is being implemented and tested. During the testing phase, various scenarios are being examined to ensure that the database functions as intended.

Overall, a DBMS for EZ-Car Rental can help improve the efficiency and accuracy of the company's operations, leading to better customer experiences and increased profitability.

PROJECT 2 – EZRental POS Operational & Strategic Decision Making via Business Reports

Project #2 Objectives

- The Design and Implementation of the EZRental POS, Auto Rental Point-of-Sales Management System of Version 1.0 was a success.

- EZRental Inc. re-hired a consultant from NYC-TECH to UPGRADE the EZRental POS, Auto Rental Point-of Sales Management System to VERSION 2.0 by adding some new features.

- The business came back with NEW REQUIREMENTS to upgrade the Database tier to Auto Rental Point-of-Sales Management Application.

- In this VERSION 2 will repeat our Project Management Application Development Methodology process as needed (Plan, Analysis, Design, Development/Implementation & Operations) to UPGRADE the Application.

- Below is two major objectives of this Project 2:

  In VERSION 2 will follow the Project Management Methodology using the Database Application Development Lifecycle (PLANNING, ANALYSIS, DESIGN, DEVELOPMENT/IMPLEMENTATION & OPERATIONS/MAINTENANCE) to UPGRADE the first version by creating.

  1) Design & Implement BUSINESS REPORT QUERIES to be used by decision makers throughout the business providing them the DATA they need to make important BUSINESS DECISIONS.

  2) Package these BUSINESS REPORT QUERIES inside STORED PROCEDURES to deliver BACK-END PROCESSING thus improving the design and performance

Business Reports Queries & Summary

In this sub-section, will be Design & Implement BUSINESS REPORT QUERIES to be used by decision makers throughout the business providing them the DATA they need to make important BUSINESS DECISIONS.

Business Report #1 – Decision to target Retail Customers by Birthday for Birthday Discount

- Report goals, objectives & decisions:

GUIDELINES	ANSWER
Q1 – What is the Business Scenario & Objectives?	▪ A promotional campaign by the Marketing Team is targeting all retail customers who were born in the month of May, offering them a birthday discount in order to promote customer loyalty and retain the customers.
Q2 – Target Persona/Decision Maker?	▪ Marketing team.
Q3 – What is the Decision to be made?	▪ The Marketing team needs to decide which retail customers to email and mail the discount coupon to. ▪ This decision is based on retail customers who were born in a particular month.
Q4 – What Data is required to make the Decision?	▪ The data required to make decision includes Customer Name, Customer Email Address, Home Address, Customer Type, and which month of the birthday is important to target.
Q5 – WHAT TABLES & COLUMNS Contain the Data Identified in Q4 and what Query Components may be involved in creating the SELECT STATEMENT?	▪ The following Table in the database schema was ANALYZED and identified to contain the required data in its columns: o CUSTOMER – FirstName, LastName, BirthDate, AddressLine1, AddressLine2, City, StateCode, ZipCode, Country, Email, CustomerType, are found in this table. ▪ Using a SELECT statement to get the data needed.

- Derived Query for Business Report #1:

```
SELECT FirstName, LastName, BirthDate, AddressLine1, AddressLine2, City, StateCode, ZipCode, Country,
Email, CustomerType
FROM Customer
WHERE CustomerType = 'R'
AND EXTRACT(MONTH FROM BirthDate) = 5;
```

- Data that currently reside in the Tables being queried by Business Report #1:

```
SELECT FirstName, LastName, BirthDate, AddressLine1, AddressLine2, City, StateCode, ZipCode, Country,
Email, CustomerType
FROM Customer;
```

Query Result | Query Result 1

SQL | All Rows Fetched: 10 in 0.003 seconds

	FIRSTNAME	LASTNAME	BIRTHDATE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY	EMAIL	CUSTOMERTYPE
1	John	Adam	01-JAN-79	10 Fulton Ave.	Apt 2B	Brooklyn	NY	11203	United States	jadam@gmail.com	R
2	Olive	Bean	02-JAN-79	32 State Ave.	Apt 3B	Brooklyn	NY	11204	United States	obean@gmail.com	R
3	Katherine	Bean	19-MAY-78	32 State Ave.	Apt 3B	Brooklyn	NY	11204	United States	kbean@gmail.com	R
4	Lori	Martinez	01-FEB-88	6 Oakwood Ave.	Apt 4C	Bronx	NY	10472	United States	lmartinez@gmail.com	R
5	Brain	Brown	01-JAN-96	1872 Granville Lane	(null)	Newark	NJ	07662	United States	bbrown@gmail.com	R
6	Adriana	Maxwell	01-JAN-96	8830 Prairie St.	FL.2	Ashburn	VA	20147	United States	AMaxwell.DarwinTravel@gmail.com	C
7	Isaac	Cortez	18-MAY-97	8336 Harvey Drive	(null)	Quakertown	PA	18951	United States	ICortez.SkyHighTravel@gmail.com	C
8	Libby	Terry	28-JUN-96	9601 Beechwood Drive	FL.5	Elizabethtown	PA	17022	United States	LTerry.LibertyTravel@gmail.com	C
9	Frederick	Mathis	07-AUG-76	556 Sierra Ave.	FL.2	Sicklerville	NJ	08081	United States	fmathis@gmail.com	C
10	Jeffery	Brown	01-JAN-98	100 East Brewery St.	FL.4	Colonial Heights	VA	23834	United States	jbrown@gmail.com	C

- Results of execution of Business Report #1 Query:

Query Result | Query Result 1

SQL | All Rows Fetched: 1 in 0.002 seconds

	FIRSTNAME	LASTNAME	BIRTHDATE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY	EMAIL	CUSTOMERTYPE
1	Katherine	Bean	19-MAY-78	32 State Ave.	Apt 3B	Brooklyn	NY	11204	United States	kbean@gmail.com	R

Business Report #2 – Decision to target Customers who have a Customer User Account to Reset Password

- Report goals, objectives & decisions:

GUIDELINES	ANSWER
Q1 – What is the Business Scenario & Objectives?	▪ This business scenario aims to target all customers who have a Customer User Account. For security reasons, the IT team will set the passwords to expire every 6 months and inform the user to reset.
Q2 – Target Persona/Decision Maker?	▪ IT team.
Q3 – What is the Decision to be made?	▪ The IT team needs to decide which customers to email the information. ▪ This decision is based on customers who have a Customer User Account.
Q4 – What Data is required to make the Decision?	▪ The data required to make decision includes Customer Name, Customer Email Address, Customer Username, and Password.
Q5 – WHAT TABLES & COLUMNS Contain the Data Identified in Q4 and what Query Components may be involved in creating the SELECT STATEMENT?	▪ The following Tables in the database schema were ANALYZED and identified to contain the required data in their columns: o CUSTOMER – FirstName, and LastName are found in this table. o CUSTOMERUSERACCOUNT –Username, and Password are found in this table. ▪ Using a SELECT statement that JOINS and queries these two tables CUSTOMER & CUSTOMERUSERACCOUNT to get the data needed.

- Derived Query for Business Report #2:

```
SELECT CustomerUserAccount.Username, CustomerUserAccount.Password, CustomerUserAccount.Email,
Customer.FirstName, Customer.LastName
From CustomerUserAccount
INNER JOIN Customer
ON CustomerUserAccount.CustomerUserAccountID = Customer.CustomerUserAccountID;
```

- Data that currently reside in the Tables being queried by Business Report #2:

```
SELECT * FROM CustomerUserAccount;
```

Query Result | Query Result 1

SQL | All Rows Fetched: 5 in 0.005 seconds

	CUSTOMERUSERACCOUNTID	USERNAME	PASSWORD	EMAIL
1	83A6364CC99947088E271B69FB0CA286	JohnAdam	SecretPass1	jadam@gmail.com
2	EC0B09FB98BA4D57B170F36B9A6BE167	OliveBean	SecretPass2	obean@gmail.com
3	E6DCD7C157514FA4ABDA03422F187B5A	KatyNeal	SecretPass3	kneal@gmail.com
4	E34401A1F57440738046CDF0CA9FA897	LoriMartinez	SecretPass4	lmartinez@gmail.com
5	E3C799088BCC4D17A4911A52875369C2	BrainBrown	SecretPass5	bbrown@gmail.com

SELECT * FROM Customer;

Query Result | Query Result 1

SQL | All Rows Fetched: 10 in 0.003 seconds

	CUSTOMERID	FIRSTNAME	LASTNAME	BIRTHDATE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY	PHONE	EMAIL	DRIVERLICENSENUMBER
1	1	John	Adam	01-JAN-79	10 Fulton Ave.	Apt 2B	Brooklyn	NY	11203	United States	333-121-1211	jadam@gmail.com	151878811
2	2	Olive	Bean	02-JAN-79	32 State Ave.	Apt 3B	Brooklyn	NY	11204	United States	333-121-1212	obean@gmail.com	851878812
3	3	Katherine	Bean	19-MAY-78	32 State Ave.	Apt 3B	Brooklyn	NY	11204	United States	333-121-1233	kbean@gmail.com	L15187881312577
4	4	Lori	Martinez	01-FEB-88	6 Oakwood Ave.	Apt 4C	Bronx	NY	10472	United States	333-121-1214	lmartinez@gmail.com	751878814
5	5	Brain	Brown	01-JAN-96	1872 Granville Lane	(null)	Newark	NJ	07662	United States	333-121-1215	bbrown@gmail.com	L1518788131299
6	6	Adriana	Maxwell	01-JAN-96	8830 Prairie St.	FL.2	Ashburn	VA	20147	United States	333-121-1216	AMaxwell.DarwinTravel@gmail.com	A15648915
7	7	Isaac	Cortez	18-MAY-97	8336 Harvey Drive	(null)	Quakertown	PA	18951	United States	333-121-1217	ICortez.SkyHighTravel@gmail.com	14456288
8	8	Libby	Terry	28-JUN-96	9601 Beechwood Drive	FL.5	Elizabethtown	PA	17022	United States	333-121-1218	LTerry.LibertyTravel@gmail.com	12345668
9	9	Frederick	Mathis	07-AUG-76	556 Sierra Ave.	FL.2	Sicklerville	NJ	08081	United States	333-121-1219	fmathis@gmail.com	L1518788131388
10	10	Jeffery	Brown	01-JAN-98	100 East Brewery St.	FL.4	Colonial Heights	VA	23834	United States	333-121-1220	jbrown@gmail.com	A15648978

DRIVERLICENSENUMBER	CUSTOMERUSERACCOUNTID	CUSTOMERTYPE
151878811	83A6364CC99947088E271B69FB0CA286	R
851878812	EC0B09FB98BA4D57B170F36B9A6BE167	R
L15187881312577	E6DCD7C157514FA4ABDA03422F187B5A	R
751878814	E34401A1F57440738046CDF0CA9FA897	R
L1518788131299	E3C799088BCC4D17A4911A52875369C2	R
A15648915	(null)	C
14456288	(null)	C
12345668	(null)	C
L1518788131388	(null)	C
A15648978	(null)	C

- Results of execution of Business Report #2 Query:

Query Result | Query Result 1 | Query Result 2

SQL | All Rows Fetched: 5 in 0.003 seconds

	USERNAME	PASSWORD	EMAIL	FIRSTNAME	LASTNAME
1	JohnAdam	SecretPass1	jadam@gmail.com	John	Adam
2	LoriMartinez	SecretPass4	lmartinez@gmail.com	Lori	Martinez
3	BrainBrown	SecretPass5	bbrown@gmail.com	Brain	Brown
4	KatyNeal	SecretPass3	kneal@gmail.com	Katherine	Bean
5	OliveBean	SecretPass2	obean@gmail.com	Olive	Bean

Business Report #3 – Decision to target a State to Open a NEW Branch by the Amount of Customers

- Report goals, objectives & decisions:

GUIDELINES	ANSWER
Q1 – What is the Business Scenario & Objectives?	▪ As the company continues to attract a growing number of customers, the management team and executives have made the decision to open a new branch. Currently, they are in the process of targeting a specific state as the location for this new branch by the amount of customers in that specific state.
Q2 – Target Persona/Decision Maker?	▪ EZRental Executives and Management team.
Q3 – What is the Decision to be made?	▪ The Executives and Management team needs to decide which state to open the branch. ▪ This decision is based on the number of customers in a specific state.
Q4 – What Data is required to make the Decision?	▪ The data required to make decision includes amount of customers in each state.
Q5 – WHAT TABLES & COLUMNS Contain the Data Identified in Q4 and what Query Components may be involved in creating the SELECT STATEMENT?	▪ The following Table in the database schema was ANALYZED and identified to contain the required data in its columns: o CUSTOMER – CustomerID, and StateCode, are found in this table. ▪ Using a SELECT, COUNT & GROUP BY statement to get the data needed.

- Derived Query for Business Report #3:

```
SELECT COUNT(CustomerID), StateCode
FROM Customer
GROUP BY StateCode;
```

- Data that currently reside in the Tables being queried by Business Report #3:

```
SELECT * FROM Customer;
```

Query Result

SQL | All Rows Fetched: 10 in 0.002 seconds

	CUSTOMERID	FIRSTNAME	LASTNAME	BIRTHDATE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY
1	1	John	Adam	01-JAN-79	10 Fulton Ave.	Apt 2B	Brooklyn	NY	11203	United States
2	2	Olive	Bean	02-JAN-79	32 State Ave.	Apt 3B	Brooklyn	NY	11204	United States
3	3	Katherine	Bean	19-MAY-78	32 State Ave.	Apt 3B	Brooklyn	NY	11204	United States
4	4	Lori	Martinez	01-FEB-88	6 Oakwood Ave.	Apt 4C	Bronx	NY	10472	United States
5	5	Brain	Brown	01-JAN-96	1872 Granville Lane	(null)	Newark	NJ	07662	United States
6	6	Adriana	Maxwell	01-JAN-96	8830 Prairie St.	FL.2	Ashburn	VA	20147	United States
7	7	Isaac	Cortez	18-MAY-97	8336 Harvey Drive	(null)	Quakertown	PA	18951	United States
8	8	Libby	Terry	28-JUN-96	9601 Beechwood Drive	FL.5	Elizabethtown	PA	17022	United States
9	9	Frederick	Mathis	07-AUG-76	556 Sierra Ave.	FL.2	Sicklerville	NJ	08081	United States
10	10	Jeffery	Brown	01-JAN-98	100 East Brewery St.	FL.4	Colonial Heights	VA	23834	United States

ZIPCODE	COUNTRY	PHONE	EMAIL	DRIVERLICENSENUMBER	CUSTOMERUSERACCOUNTID	CUSTOMERTYPE
11203	United States	333-121-1211	jadam@gmail.com	151878811	83A6364CC99947088E271B69FB0CA286	R
11204	United States	333-121-1212	obean@gmail.com	851878812	EC0B09FB98BA4D57B170F36B9A6BE167	R
11204	United States	333-121-1233	kbean@gmail.com	L15187881312577	E6DCD7C157514FA4ABDA03422F187B5A	R
10472	United States	333-121-1214	lmartinez@gmail.com	751878814	E34401A1F57440738046CDF0CA9FA897	R
07662	United States	333-121-1215	bbrown@gmail.com	L1518788131299	E3C799088BCC4D17A4911A52875369C2	R
20147	United States	333-121-1216	AMaxwell.DarwinTravel@gmail.com	A15648915	(null)	C
18951	United States	333-121-1217	ICortez.SkyHighTravel@gmail.com	14456288	(null)	C
17022	United States	333-121-1218	LTerry.LibertyTravel@gmail.com	12345668	(null)	C
08081	United States	333-121-1219	fmathis@gmail.com	L1518788131388	(null)	C
23834	United States	333-121-1220	jbrown@gmail.com	A15648978	(null)	C

- Results of execution of Business Report #3 Query:

	COUNT(CUSTOMERID)	STATECODE
1	2	PA
2	2	VA
3	4	NY
4	2	NJ

Business Report #4 – Decision to target Companies by Discount Percentage Rate for Promotional Discount

- Report goals, objectives & decisions:

GUIDELINES	ANSWER
Q1 – What is the Business Scenario & Objectives?	▪ A promotional campaign by the Marketing & Sales team is targeting all corporate customers who have a Corporate Discount Percentage Rate of less than 20%. They are offering these customers a special discount with the aim of promoting customer loyalty and encouraging continued engagement with the company.
Q2 – Target Persona/Decision Maker?	▪ Marketing & Sales team.
Q3 – What is the Decision to be made?	▪ The Marketing & Sales team needs to decide which cooperate customers to email and mail the discount coupon to. ▪ This decision is based on the corporate customers who have a Corporate Discount Percentage Rate of less than 20%.
Q4 – What Data is required to make the Decision?	▪ The data required to make decision Company Name, Company Representative Name, Contact Email, and Corporate Discount Percentage Rate
Q5 – WHAT TABLES & COLUMNS Contain the Data Identified in Q4 and what Query Components may be involved in creating the SELECT STATEMENT?	▪ The following Table in the database schema was ANALYZED and identified to contain the required data in its columns: o COMPANY – CompanyName, AddressLine1, AddressLine2, City, StateCode, ZipCode, Country, CompanyRepName, ContactEmail, CorporateDiscountPercentageRate are found in this table. ▪ Using a SELECT statement to get the data needed.

- Derived Query for Business Report #4:

```
SELECT CompanyName, AddressLine1, AddressLine2, City, StateCode, ZipCode, Country,
CompanyRepName, ContactEmail, CorporateDiscountPercentageRate
FROM Company
WHERE CorporateDiscountPercentageRate < 0.2;
```

- Data that currently reside in the Tables being queried by Business Report #4:

```
SELECT * FROM Company;
```

	COMPANYID	COMPANYNAME	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY	COMPANYREPNAME	CONTACTPHONE	CONTACTEMAIL	CORPORATEDISCOUNTPERCENTAGERATE
1	10001	Darwin Travel	8830 Prairie St.	FL.2	Ashburn	VA	20147	United States	Adriana Maxwell	333-121-1216	AMaxwell.DarwinTravel@gmail.com	0.25
2	10002	Sky High Tr...	8336 Harvey Drive	(null)	Quakertown	PA	18951	United States	Isaac Cortez	333-121-1217	ICortez.SkyHighTravel@gmail.com	0.15
3	10003	Liberty Travel	9601 Beechwood Drive	FL.5	Elizabethtown	PA	17022	United States	Libby Terry	333-121-1218	LTerry.LibertyTravel@gmail.com	0.15
4	10004	Cool River ...	556 Sierra Ave.	FL.2	Sicklerville	NJ	08081	United States	Frederick Mathis	333-121-1219	FMathis.CRTA@gmail.com	0.25
5	10005	Twin City T...	100 East Brewery St.	FL.4	Colonial Heights	VA	23834	United States	Jeffery Brown	333-121-1220	JBrown.TCTT@gmail.com	0.2

- Results of execution of Business Report #4 Query:

	COMPANYNAME	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY	COMPANYREPNAME	CONTACTEMAIL	CORPORATEDISCOUNTPERCENTAGERATE
1	Sky High Travel	8336 Harvey Drive	(null)	Quakertown	PA	18951	United States	Isaac Cortez	ICortez.SkyHighTravel@gmail.com	0.15
2	Liberty Travel	9601 Beechwood Drive	FL.5	Elizabethtown	PA	17022	United States	Libby Terry	LTerry.LibertyTravel@gmail.com	0.15

Business Report #5 – Decision to target Future Credit Card Merchants

- Report goals, objectives & decisions:

GUIDELINES	ANSWER
Q1 – What is the Business Scenario & Objectives?	▪ In this business scenario, the objective is to determine what are popular credit card merchants. Upon identifying the popular merchants, the company intends to continue using their services. This analysis will provide valuable insights into customer preferences, enabling the company to make an informed decision regarding the selection of their credit card merchant for future use.
Q2 – Target Persona/Decision Maker?	▪ Finance Management team.
Q3 – What is the Decision to be made?	▪ The Finance Management team determine which credit card merchant continue to use. ▪ The decision is based on the number of users for each credit card merchant.
Q4 – What Data is required to make the Decision?	▪ The data required to make decision Company Name, Company Representative Name, Contact Email, and Corporate Discount Percentage Rate
Q5 – WHAT TABLES & COLUMNS Contain the Data Identified in Q4 and what Query Components may be involved in creating the SELECT STATEMENT?	▪ The following Tables and columns would be used: o CreditCard – CreditCardOwnerName, MerchantCode, ActivationStatus are found in this table. o CreditCardMerchant – MerchantCode, MerchatName are found in this table. ▪ Using a SELECT, COUNT statement to get the data needed.

- Derived Query for Business Report #5:

```
SELECT CreditCardMerchant.MerchantName,
COUNT(CreditCard.CreditCardOwnerName) as "Number of Customer"
FROM CreditCardMerchant
INNER JOIN CreditCard
ON CreditCardMerchant.MerchantCode = CreditCard.MerchantCode
WHERE ActivationStatus = 1
GROUP BY MerchantName
ORDER BY "Number of Customer" DESC;
```

- Data that currently reside in the Tables being queried by Business Report #5:

```
SELECT * FROM CreditCard;
```

	CREDITCARDNUMBER	CREDITCARDOWNERNAME	CREDITCARDISSUINGCOMPANY	MERCHANTCODE	EXPDATE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	COUNTRY	ZIPCODE
1	3333333333333331	John Adam	American Express	1	08-DEC-23	10 Fulton ...	Apt 2B	Brooklyn	NY	United States	11203
2	3333333333333332	Olive Bean	American Express	1	25-JUN-24	32 State Ave.	Apt 3B	Brooklyn	NY	United States	11204
3	3333333333333334	Lori Martinez	Capital One	8	25-MAY-25	6 Oakwood ...	Apt 4C	Bronx	NY	United States	10472
4	3333333333333335	Brain Brown	Chase	8	11-DEC-27	1872 Granv...	(null)	Newark	NJ	United States	07662
5	3333333333333336	Adriana Maxwell	Capital One	1	11-DEC-27	8830 Prair...	FL.2	Ashburn	VA	United States	20147
6	3333333333333337	Isaac Cortez	Capital One	3	11-DEC-27	8336 Harve...	(null)	Quakertown	PA	United States	18951
7	3333333333333338	Libby Terry	Capital One	3	11-DEC-27	9601 Beech...	FL.5	Elizabethtown	PA	United States	17022
8	3333333333333339	Frederick Mathis	Chase	8	11-DEC-27	556 Sierra...	FL.2	Sicklerville	NJ	United States	08081
9	3333333333333399	Jeffery Brown	Chase	8	11-DEC-27	100 East B...	FL.4	Colonial Heights	VA	United States	23834

ZIPCODE	CREDITCARDLIMIT	CREDITCARDBALANCE	ACTIVATIONSTATUS
11203	20000	5486.12	1
11204	22000	6486.32	1
10472	18000	286.12	1
07662	20000	5486.32	1
20147	22000	1486.32	1
18951	20000	5486.32	1
17022	20000	5486.32	1
08081	20000	5486.32	1
23834	20000	5486.32	1

SELECT * FROM CreditCardMerchant;

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

- Results of execution of Business Report #5 Query:

	MERCHANTNAME	Number of Customer
1	PayPal	4
2	Stax by Fattmerchant	3
3	Dharma Merchant Services	2

Business Reports Stored Procedures & Summary

Design the structure to 5 STORED PROCEDURES HOSTING EACH OF THE OPERATIONAL & STRATEGIC BUSINESS SCENARIO OBJECTIVES & SELECT QUERIES that IMPLEMENTS each BUSINESS REPORT from the last sub-section. Use STORED PROCEDURES to deliver BACK-END PROCESSING thus improving the design and performance.

Stored Procedure for Business Report #1 – Decision to target Retail Customers by Birthday for Birthday Discount

- Stored Procedure Business Report goals & objectives:

GUIDELINES	ANSWER
Q1 – What is the Business Scenario & Objectives?	▪ A promotional campaign by the Marketing Team is targeting all retail customers who were born in the month of May, offering them a birthday discount in order to promote customer loyalty and retain the customers.
Q2 – Target Persona/Decision Maker?	▪ Marketing team.
Q3 – What is the Decision to be made?	▪ The Marketing team needs to decide which retail customers to email and mail the discount coupon to. ▪ This decision is based on retail customers who were born in a particular month.
Q4 – What Data is required to make the Decision?	▪ The data required to make decision includes Customer Name, Customer Email Address, Home Address, Customer Type, and which month of the birthday is important to target.
Q5 – WHAT TABLES & COLUMNS Contain the Data Identified in Q4 and what Query Components may be involved in creating the SELECT STATEMENT?	▪ The following Table in the database schema was ANALYZED and identified to contain the required data in its columns: o CUSTOMER – FirstName, LastName, BirthDate, AddressLine1, AddressLine2, City, StateCode, ZipCode, Country, Email, CustomerType, are found in this table. ▪ We would need a STORED PROCEDURE that includes a SELECT statement that queries the table CUSTOMER to get the data needed. ▪ Multiple records can be returned; therefore, this STORED PROCEDURE will need to use CURSORS.

- Target Query Report from the last sub-section that is the foundation for the Stored Procedure:

```
SELECT FirstName, LastName, BirthDate, AddressLine1, AddressLine2, City, StateCode, ZipCode, Country,
Email, CustomerType
FROM Customer
WHERE CustomerType = 'R'
AND EXTRACT(MONTH FROM BirthDate) = 5;
```

- Stored Procedure implementation code Listing:

```
CREATE OR REPLACE PROCEDURE GetRetailCustomersByBirthday(p_CustomerType IN CHAR, p_BirthDate IN
NUMBER)
```

```
IS
        --declare variable needed for SELECT INTO statement
        v_FirstName VARCHAR2(50);
        v_LastName VARCHAR2(50);
        v_BirthDate DATE;
        v_AddressLine1 VARCHAR2(50);
        v_AddressLine2 VARCHAR2(50);
        v_City VARCHAR2(30);
        v_StateCode CHAR(2);
        v_ZipCode VARCHAR2(10);
        v_Country VARCHAR2(100);
        v_Email VARCHAR2(100);
        v_CustomerType CHAR(1);

        --Declare Cursor
        CURSOR cur_Customer IS

                -- Query cursor will point to results
                SELECT FirstName, LastName, BirthDate, AddressLine1, AddressLine2, City, StateCode,
ZipCode, Country, Email, CustomerType
                FROM Customer
                WHERE CustomerType = p_CustomerType /*CustomerType = 'R' */
                AND EXTRACT(MONTH FROM BirthDate) = p_BirthDate;


--Start Execution section

BEGIN
        --Open Cursor
        OPEN cur_Customer; -- open cursor for use
        --loop to display each record returned by cursor
        --Use PL/SQL language control or loop to display each record pointed by cursor
        LOOP
                -- Fetch cursor data
                FETCH cur_Customer INTO v_FirstName, v_LastName, v_BirthDate, v_AddressLine1,
v_AddressLine2, v_City, v_StateCode, v_ZipCode, v_Country, v_Email, v_CustomerType;
                EXIT WHEN cur_Customer%NOTFOUND;
                --Display each record
                DBMS_OUTPUT.PUT_LINE('Customer info: ' ||v_FirstName|| ' | ' ||v_LastName||
                ' | ' ||v_BirthDate|| ' | ' ||v_AddressLine1||  ' | '  ||v_AddressLine2|| ' | ' ||v_City|| ' | '
||v_StateCode|| ' | ' ||v_ZipCode|| ' | ' ||v_Country|| ' | ' ||v_Email|| ' | ' ||v_CustomerType);
        END LOOP;

        CLOSE cur_Customer; --close cursor

END GetRetailCustomersByBirthday;
```

- Data that currently reside in the Tables being queried by STORED PROCEDURE implementing Business Report #1 from the last sub section:

  SELECT * FROM Customer;

Script Output | Query Result

SQL | All Rows Fetched: 10 in 0.065 seconds

	CUSTOMERID	FIRSTNAME	LASTNAME	BIRTHDATE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY	PHONE
1	1	John	Adam	01-JAN-79	10 Fulton Ave.	Apt 2B	Brooklyn	NY	11203	United States	333-121-1211
2	2	Olive	Bean	02-JAN-79	32 State Ave.	Apt 3B	Brooklyn	NY	11204	United States	333-121-1212
3	3	Katherine	Bean	19-MAY-78	32 State Ave.	Apt 3B	Brooklyn	NY	11204	United States	333-121-1233
4	4	Lori	Martinez	01-FEB-88	6 Oakwood Ave.	Apt 4C	Bronx	NY	10472	United States	333-121-1214
5	5	Brain	Brown	01-JAN-96	1872 Granville Lane	(null)	Newark	NJ	07662	United States	333-121-1215
6	6	Adriana	Maxwell	01-JAN-96	8830 Prairie St.	FL.2	Ashburn	VA	20147	United States	333-121-1216
7	7	Isaac	Cortez	18-MAY-97	8336 Harvey Drive	(null)	Quakertown	PA	18951	United States	333-121-1217
8	8	Libby	Terry	28-JUN-96	9601 Beechwood Drive	FL.5	Elizabethtown	PA	17022	United States	333-121-1218
9	9	Frederick	Mathis	07-AUG-76	556 Sierra Ave.	FL.2	Sicklerville	NJ	08081	United States	333-121-1219
10	10	Jeffery	Brown	01-JAN-98	100 East Brewery St.	FL.4	Colonial Heights	VA	23834	United States	333-121-1220

PHONE	EMAIL	DRIVERLICENSENUMBER	CUSTOMERUSERACCOUNTID	CUSTOMERTYPE
333-121-1211	jadam@gmail.com	151878811	83A6364CC99947088E271B69FB0CA286	R
333-121-1212	obean@gmail.com	851878812	EC0B09FB98BA4D57B170F36B9A6BE167	R
333-121-1233	kbean@gmail.com	L15187881312577	E6DCD7C157514FA4ABDA03422F187B5A	R
333-121-1214	lmartinez@gmail.com	751878814	E34401A1F57440738046CDF0CA9FA897	R
333-121-1215	bbrown@gmail.com	L1518788131299	E3C799088BCC4D17A4911A52875369C2	R
333-121-1216	AMaxwell.DarwinTravel@gmail.com	A15648915	(null)	C
333-121-1217	ICortez.SkyHighTravel@gmail.com	14456288	(null)	C
333-121-1218	LTerry.LibertyTravel@gmail.com	12345668	(null)	C
333-121-1219	fmathis@gmail.com	L1518788131388	(null)	C
333-121-1220	jbrown@gmail.com	A15648978	(null)	C

- Results of execution of STORED PROCEDURE implementing Business Report #1:

  SET SERVEROUTPUT ON;
  EXECUTE GetRetailCustomersByBirthday('R', 5);

Script Output | Query Result

Task completed in 0.074 seconds

```
Customer info: Katherine | Bean | 19-MAY-78 | 32 State Ave. | Apt 3B | Brooklyn | NY | 11204 | United States | kbean@gmail.com | R


PL/SQL procedure successfully completed.
```

- Results of execution of Matching SELECT QUERY BUSINESS REPORT from the last sub-section proving the results are the same:

Query Result | Query Result 1

SQL | All Rows Fetched: 1 in 0.002 seconds

	FIRSTNAME	LASTNAME	BIRTHDATE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY	EMAIL	CUSTOMERTYPE
1	Katherine	Bean	19-MAY-78	32 State Ave.	Apt 3B	Brooklyn	NY	11204	United States	kbean@gmail.com	R

Stored Procedure for Business Report #2 – Decision to target Customers who have a Customer User Account to Reset Password

- Stored Procedure Business Report goals & objectives:

GUIDELINES	ANSWER
Q1 – What is the Business Scenario & Objectives?	▪ This business scenario aims to target all customers who have a Customer User Account. For security reasons, the IT team will set the passwords to expire every 6 months and inform the user to reset.
Q2 – Target Persona/Decision Maker?	▪ IT team.
Q3 – What is the Decision to be made?	▪ The IT team needs to decide which customers to email the information. ▪ This decision is based on customers who have a Customer User Account.
Q4 – What Data is required to make the Decision?	▪ The data required to make decision includes Customer Name, Customer Email Address, Customer Username, and Password.
Q5 – WHAT TABLES & COLUMNS Contain the Data Identified in Q4 and what Query Components may be involved in creating the SELECT STATEMENT?	▪ The following Tables in the database schema were ANALYZED and identified to contain the required data in their columns: o CUSTOMER – FirstName, and LastName are found in this table. o CUSTOMERUSERACCOUNT –Username, and Password are found in this table. ▪ We would need a STORED PROCEDURE that includes a SELECT statement that JOINS and queries these two tables CUSTOMER & CUSTOMERUSERACCOUNT to get the data needed. ▪ Multiple records can be returned; therefore, this STORED PROCEDURE will need to use CURSORS.

- Target Query Report from the last sub-section that is the foundation for the Stored Procedure:

```
SELECT CustomerUserAccount.Username, CustomerUserAccount.Password, CustomerUserAccount.Email,
Customer.FirstName, Customer.LastName
From CustomerUserAccount
INNER JOIN Customer
ON CustomerUserAccount.CustomerUserAccountID = Customer.CustomerUserAccountID;
```

- Stored Procedure implementation code Listing:

```
CREATE OR REPLACE PROCEDURE GetCustomersToResetPass

IS
        --declare variable needed for SELECT INTO statement
        v_Username VARCHAR2(50);
        v_Password VARCHAR2(50);
        v_Email VARCHAR2(100);
        v_FirstName VARCHAR2(50);
        v_LastName VARCHAR2(50);
```

```
        --Declare Cursor
        CURSOR cur_ResetPass IS

                -- Query cursor will point to results
                SELECT CustomerUserAccount.Username, CustomerUserAccount.Password,
CustomerUserAccount.Email, Customer.FirstName, Customer.LastName
                From CustomerUserAccount
                INNER JOIN Customer
                ON CustomerUserAccount.CustomerUserAccountID = Customer.CustomerUserAccountID;

--Start Execution section

BEGIN
        --Open Cursor
        OPEN cur_ResetPass; -- open cursor for use
        --loop to display each record returned by cursor
        --Use PL/SQL language control or loop to display each record pointed by cursor
        LOOP
                -- Fetch cursor data
                FETCH cur_ResetPass INTO v_Username, v_Password, v_Email, v_FirstName, v_LastName;
                EXIT WHEN cur_ResetPass%NOTFOUND;
                --Display each record
                DBMS_OUTPUT.PUT_LINE('Customer info: ' ||v_Username|| ' | ' ||v_Password||
                ' | ' ||v_Email|| ' | ' ||v_FirstName|| ' | ' ||v_LastName);
        END LOOP;

        CLOSE cur_ResetPass; --close cursor

END GetCustomersToResetPass;
```

- Data that currently reside in the Tables being queried by STORED PROCEDURE implementing Business Report #2 from the last sub section:

```
SELECT * FROM CustomerUserAccount;
```

Query Result | Query Result 1

SQL | All Rows Fetched: 5 in 0.005 seconds

	CUSTOMERUSERACCOUNTID	USERNAME	PASSWORD	EMAIL
1	83A6364CC99947088E271B69FB0CA286	JohnAdam	SecretPass1	jadam@gmail.com
2	EC0B09FB98BA4D57B170F36B9A6BE167	OliveBean	SecretPass2	obean@gmail.com
3	E6DCD7C157514FA4ABDA03422F187B5A	KatyNeal	SecretPass3	kneal@gmail.com
4	E34401A1F57440738046CDF0CA9FA897	LoriMartinez	SecretPass4	lmartinez@gmail.com
5	E3C799088BCC4D17A4911A52875369C2	BrainBrown	SecretPass5	bbrown@gmail.com

```
SELECT * FROM Customer;
```

Query Result | Query Result 1

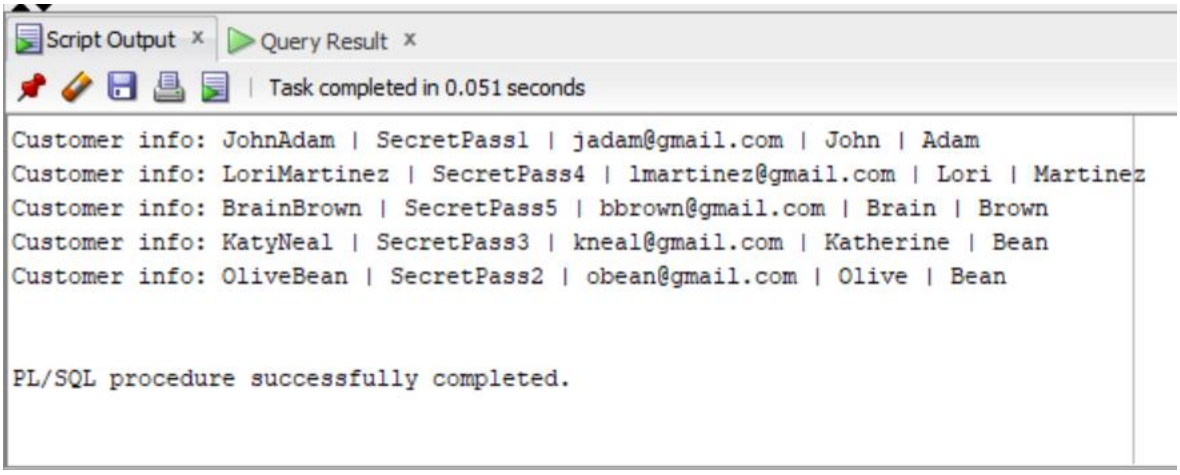
SQL | All Rows Fetched: 10 in 0.003 seconds

	CUSTOMERID	FIRSTNAME	LASTNAME	BIRTHDATE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY	PHONE	EMAIL	DRIVERLICENSENUMBER
1	1	John	Adam	01-JAN-79	10 Fulton Ave.	Apt 2B	Brooklyn	NY	11203	United States	333-121-1211	jadam@gmail.com	151878811
2	2	Olive	Bean	02-JAN-79	32 State Ave.	Apt 3B	Brooklyn	NY	11204	United States	333-121-1212	obean@gmail.com	851878812
3	3	Katherine	Bean	19-MAY-78	32 State Ave.	Apt 3B	Brooklyn	NY	11204	United States	333-121-1233	kbean@gmail.com	L15187881312577
4	4	Lori	Martinez	01-FEB-88	6 Oakwood Ave.	Apt 4C	Bronx	NY	10472	United States	333-121-1214	lmartinez@gmail.com	751878814
5	5	Brain	Brown	01-JAN-96	1872 Granville Lane	(null)	Newark	NJ	07662	United States	333-121-1215	bbrown@gmail.com	L1518788131299
6	6	Adriana	Maxwell	01-JAN-96	8830 Prairie St.	FL.2	Ashburn	VA	20147	United States	333-121-1216	AMaxwell.DarwinTravel@gmail.com	A15648915
7	7	Isaac	Cortez	18-MAY-97	8336 Harvey Drive	(null)	Quakertown	PA	18951	United States	333-121-1217	ICortez.SkyHighTravel@gmail.com	14456288
8	8	Libby	Terry	28-JUN-96	9601 Beechwood Drive	FL.5	Elizabethtown	PA	17022	United States	333-121-1218	LTerry.LibertyTravel@gmail.com	12345668
9	9	Frederick	Mathis	07-AUG-76	556 Sierra Ave.	FL.2	Sicklerville	NJ	08081	United States	333-121-1219	fmathis@gmail.com	L1518788131388
10	10	Jeffery	Brown	01-JAN-98	100 East Brewery St.	FL.4	Colonial Heights	VA	23834	United States	333-121-1220	jbrown@gmail.com	A15648978

DRIVERLICENSENUMBER	CUSTOMERUSERACCOUNTID	CUSTOMERTYPE
151878811	83A6364CC99947088E271B69FB0CA286	R
851878812	EC0B09FB98BA4D57B170F36B9A6BE167	R
L15187881312577	E6DCD7C157514FA4ABDA03422F187B5A	R
751878814	E34401A1F57440738046CDF0CA9FA897	R
L1518788131299	E3C799088BCC4D17A4911A52875369C2	R
A15648915	(null)	C
14456288	(null)	C
12345668	(null)	C
L1518788131388	(null)	C
A15648978	(null)	C

- Results of execution of STORED PROCEDURE implementing Business Report #2:

```
EXECUTE GetCustomersToResetPass;
```

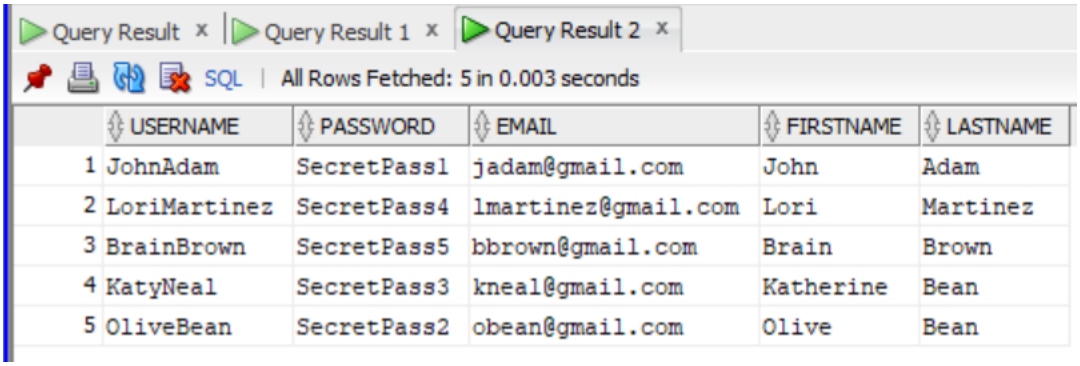


Script Output | Query Result

Task completed in 0.051 seconds

```
Customer info: JohnAdam | SecretPass1 | jadam@gmail.com | John | Adam
Customer info: LoriMartinez | SecretPass4 | lmartinez@gmail.com | Lori | Martinez
Customer info: BrainBrown | SecretPass5 | bbrown@gmail.com | Brain | Brown
Customer info: KatyNeal | SecretPass3 | kneal@gmail.com | Katherine | Bean
Customer info: OliveBean | SecretPass2 | obean@gmail.com | Olive | Bean


PL/SQL procedure successfully completed.
```

- Results of execution of Matching SELECT QUERY BUSINESS REPORT from the last sub-section proving the results are the same:



Query Result | Query Result 1 | Query Result 2

SQL | All Rows Fetched: 5 in 0.003 seconds

	USERNAME	PASSWORD	EMAIL	FIRSTNAME	LASTNAME
1	JohnAdam	SecretPass1	jadam@gmail.com	John	Adam
2	LoriMartinez	SecretPass4	lmartinez@gmail.com	Lori	Martinez
3	BrainBrown	SecretPass5	bbrown@gmail.com	Brain	Brown
4	KatyNeal	SecretPass3	kneal@gmail.com	Katherine	Bean
5	OliveBean	SecretPass2	obean@gmail.com	Olive	Bean

Stored Procedure for Business Report #3 – Decision to target a State to Open a NEW Branch by the Amount of Customers

- Stored Procedure Business Report goals & objectives:

GUIDELINES	ANSWER
Q1 – What is the Business Scenario & Objectives?	▪ As the company continues to attract a growing number of customers, the management team and executives have made the decision to open a new branch. Currently, they are in the process of targeting a specific state as the location for this new branch by the amount of customers in that specific state.
Q2 – Target Persona/Decision Maker?	▪ EZRental Executives and Management team.
Q3 – What is the Decision to be made?	▪ The Executives and Management team needs to decide which state to open the branch. ▪ This decision is based on the number of customers in a specific state.
Q4 – What Data is required to make the Decision?	▪ The data required to make decision includes amount of customers in each state.
Q5 – WHAT TABLES & COLUMNS Contain the Data Identified in Q4 and what Query Components may be involved in creating the SELECT STATEMENT?	▪ The following Table in the database schema was ANALYZED and identified to contain the required data in its columns: o CUSTOMER – CustomerID, and StateCode, are found in this table. ▪ We would need a STORED PROCEDURE that includes a SELECT, COUNT&GROUP BY statement to queries the table CUSTOMER to get the data needed. ▪ Multiple records can be returned; therefore, this STORED PROCEDURE will need to use CURSORS.

- Target Query Report from the last sub-section that is the foundation for the Stored Procedure:

```
SELECT COUNT(CustomerID), StateCode
FROM Customer
GROUP BY StateCode;
```

- Stored Procedure Statement for Business Report #3:

```
CREATE OR REPLACE PROCEDURE GetStateByCustomerAmount

IS
        --declare variable needed for SELECT INTO statement
        v_CustomerID NUMBER;
        v_StateCode CHAR(2);

        --Declare Cursor
        CURSOR cur_State IS

                -- Query cursor will point to results
                SELECT COUNT(CustomerID), StateCode
                FROM Customer
```

```
                GROUP BY StateCode;

--Start Execution section

BEGIN
        --Open Cursor
        OPEN cur_State; -- open cursor for use
        --loop to display each record returned by cursor
        --Use PL/SQL language control or loop to display each record pointed by cursor
        LOOP
                -- Fetch cursor data
                FETCH cur_State INTO v_CustomerID, v_StateCode;
                EXIT WHEN cur_State%NOTFOUND;
                --Display each record
                DBMS_OUTPUT.PUT_LINE('Customer info: ' ||v_CustomerID|| ' | ' ||v_StateCode);
        END LOOP;

   CLOSE cur_State; --close cursor

END GetStateByCustomerAmount;
```

- Data that currently reside in the Tables being queried by STORED PROCEDURE implementing Business Report #3 from the last sub section:

```
SELECT * FROM Customer;
```

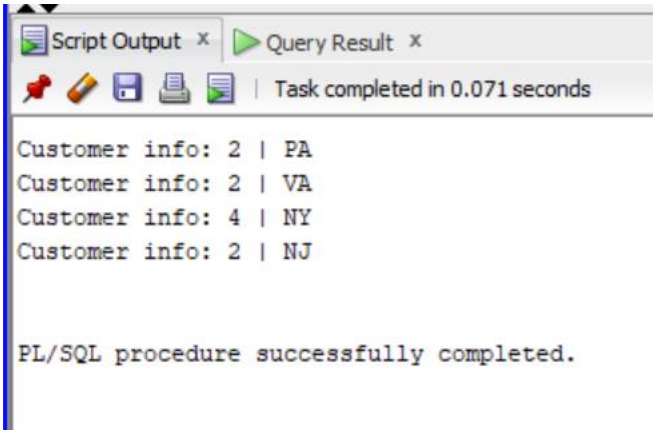
Query Result

SQL | All Rows Fetched: 10 in 0.002 seconds

	CUSTOMERID	FIRSTNAME	LASTNAME	BIRTHDATE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY
1	1	John	Adam	01-JAN-79	10 Fulton Ave.	Apt 2B	Brooklyn	NY	11203	United States
2	2	Olive	Bean	02-JAN-79	32 State Ave.	Apt 3B	Brooklyn	NY	11204	United States
3	3	Katherine	Bean	19-MAY-78	32 State Ave.	Apt 3B	Brooklyn	NY	11204	United States
4	4	Lori	Martinez	01-FEB-88	6 Oakwood Ave.	Apt 4C	Bronx	NY	10472	United States
5	5	Brain	Brown	01-JAN-96	1872 Granville Lane	(null)	Newark	NJ	07662	United States
6	6	Adriana	Maxwell	01-JAN-96	8830 Prairie St.	FL.2	Ashburn	VA	20147	United States
7	7	Isaac	Cortez	18-MAY-97	8336 Harvey Drive	(null)	Quakertown	PA	18951	United States
8	8	Libby	Terry	28-JUN-96	9601 Beechwood Drive	FL.5	Elizabethtown	PA	17022	United States
9	9	Frederick	Mathis	07-AUG-76	556 Sierra Ave.	FL.2	Sicklerville	NJ	08081	United States
10	10	Jeffery	Brown	01-JAN-98	100 East Brewery St.	FL.4	Colonial Heights	VA	23834	United States

ZIPCODE	COUNTRY	PHONE	EMAIL	DRIVERLICENSENUMBER	CUSTOMERUSERACCOUNTID	CUSTOMERTYPE
11203	United States	333-121-1211	jadam@gmail.com	151878811	83A6364CC99947088E271B69FB0CA286	R
11204	United States	333-121-1212	obean@gmail.com	851878812	EC0B09FB98BA4D57B170F36B9A6BE167	R
11204	United States	333-121-1233	kbean@gmail.com	L15187881312577	E6DCD7C157514FA4ABDA03422F187B5A	R
10472	United States	333-121-1214	lmartinez@gmail.com	751878814	E34401A1F57440738046CDF0CA9FA897	R
07662	United States	333-121-1215	bbrown@gmail.com	L1518788131299	E3C799088BCC4D17A4911A52875369C2	R
20147	United States	333-121-1216	AMaxwell.DarwinTravel@gmail.com	A15648915	(null)	C
18951	United States	333-121-1217	ICortez.SkyHighTravel@gmail.com	14456288	(null)	C
17022	United States	333-121-1218	LTerry.LibertyTravel@gmail.com	12345668	(null)	C
08081	United States	333-121-1219	fmathis@gmail.com	L1518788131388	(null)	C
23834	United States	333-121-1220	jbrown@gmail.com	A15648978	(null)	C

- Results of execution of STORED PROCEDURE implementing Business Report #3:
  EXECUTE GetStateByCustomerAmount;



- Results of execution of Matching SELECT QUERY BUSINESS REPORT from the last sub-section proving the results are the same:

	COUNT(CUSTOMERID)	STATECODE
1	2	PA
2	2	VA
3	4	NY
4	2	NJ

Stored Procedure for Business Report #4 – Decision to target Companies by Discount Percentage Rate for Promotional Discount

- Stored Procedure Business Report goals & objectives:

GUIDELINES	ANSWER
Q1 – What is the Business Scenario & Objectives?	▪ A promotional campaign by the Marketing & Sales team is targeting all corporate customers who have a Corporate Discount Percentage Rate of less than 20%. They are offering these customers a special discount with the aim of promoting customer loyalty and encouraging continued engagement with the company.
Q2 – Target Persona/Decision Maker?	▪ Marketing & Sales team.
Q3 – What is the Decision to be made?	▪ The Marketing & Sales team needs to decide which cooperate customers to email and mail the discount coupon to. ▪ This decision is based on the corporate customers who have a Corporate Discount Percentage Rate of less than 20%.
Q4 – What Data is required to make the Decision?	▪ The data required to make decision Company Name, Company Representative Name, Contact Email, and Corporate Discount Percentage Rate
Q5 – WHAT TABLES & COLUMNS Contain the Data Identified in Q4 and what Query Components may be involved in creating the SELECT STATEMENT?	▪ The following Table in the database schema was ANALYZED and identified to contain the required data in its columns: o COMPANY – CompanyName, AddressLine1, AddressLine2, City, StateCode, ZipCode, Country, CompanyRepName, ContactEmail, CorporateDiscountPercentageRate are found in this table. ▪ We would need a STORED PROCEDURE that includes a SELECT statement that queries the table COMPANY to get the data needed. ▪ Multiple records can be returned; therefore, this STORED PROCEDURE will need to use CURSORS.

- Target Query Report from the last sub-section that is the foundation for the Stored Procedure:

```
SELECT CompanyName, AddressLine1, AddressLine2, City, StateCode, ZipCode, Country,
CompanyRepName, ContactEmail, CorporateDiscountPercentageRate
FROM Company
WHERE CorporateDiscountPercentageRate < 0.2;
```

- Stored Procedure Statement for Business Report #4:

```
CREATE OR REPLACE PROCEDURE GetCompaniesByDisRate(p_CorporateDiscountPercentageRate IN
NUMBER)

IS
        --declare variable needed for SELECT INTO statement
        v_CompanyName VARCHAR2(50);
        v_AddressLine1 VARCHAR2(50);
        v_AddressLine2 VARCHAR2(50);
        v_City VARCHAR2(30);
        v_StateCode CHAR(2);
```

```
        v_ZipCode VARCHAR2(10);
        v_Country VARCHAR2(100);
        v_CompanyRepName VARCHAR2(50);
        v_ContactEmail VARCHAR2(100);
        v_CorporateDiscountPercentageRate NUMBER(3,2);

        --Declare Cursor
        CURSOR cur_Companies IS

            -- Query cursor will point to results
            SELECT CompanyName, AddressLine1, AddressLine2, City, StateCode, ZipCode, Country,
CompanyRepName, ContactEmail, CorporateDiscountPercentageRate
            FROM Company
            WHERE CorporateDiscountPercentageRate < p_CorporateDiscountPercentageRate;


--Start Execution section

BEGIN
        --Open Cursor
        OPEN cur_Companies; -- open cursor for use
        --loop to display each record returned by cursor
        --Use PL/SQL language control or loop to display each record pointed by cursor
        LOOP
            -- Fetch cursor data
            FETCH cur_Companies INTO v_CompanyName, v_AddressLine1, v_AddressLine2, v_City,
v_StateCode, v_ZipCode, v_Country, v_CompanyRepName, v_ContactEmail,
v_CorporateDiscountPercentageRate;
            EXIT WHEN cur_Companies%NOTFOUND;
            --Display each record
            DBMS_OUTPUT.PUT_LINE('Customer info: ' ||v_CompanyName||' | ' ||v_AddressLine1|| ' |
' ||v_AddressLine2|| ' | ' ||v_City|| ' | ' ||v_StateCode|| ' | ' ||v_ZipCode|| ' | ' ||v_Country|| ' | '
||v_CompanyRepName|| ' | ' ||v_ContactEmail|| ' | ' ||v_CorporateDiscountPercentageRate);
        END LOOP;

        CLOSE cur_Companies; --close cursor

END GetCompaniesByDisRate;
```

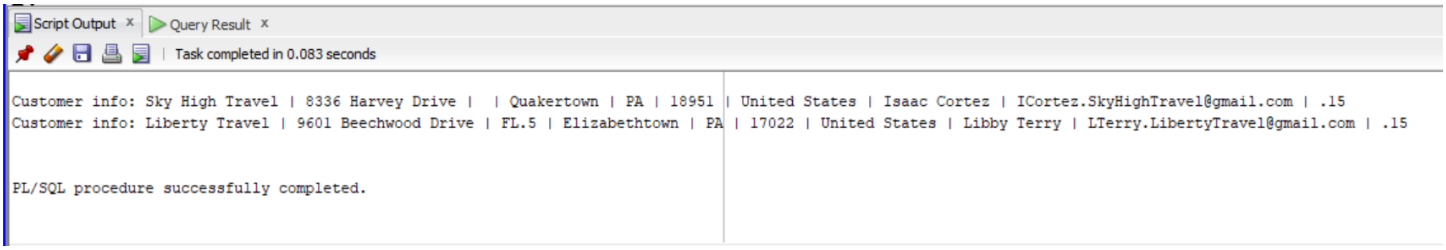
- Data that currently reside in the Tables being queried by STORED PROCEDURE implementing Business Report #4 from the last sub section:

```
SELECT * FROM Company;
```

	COMPANYID	COMPANYNAME	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY	COMPANYREPNAME	CONTACTPHONE	CONTACTEMAIL	CORPORATEDISCOUNTPERCENTAGERATE
1	10001	Darwin Travel	8830 Prairie St.	FL.2	Ashburn	VA	20147	United States	Adriana Maxwell	333-121-1216	AMaxwell.DarwinTravel@gmail.com	0.25
2	10002	Sky High Tr...	8336 Harvey Drive	(null)	Quakertown	PA	18951	United States	Isaac Cortez	333-121-1217	ICortez.SkyHighTravel@gmail.com	0.15
3	10003	Liberty Travel	9601 Beechwood Drive	FL.5	Elizabethtown	PA	17022	United States	Libby Terry	333-121-1218	LTerry.LibertyTravel@gmail.com	0.15
4	10004	Cool River ...	556 Sierra Ave.	FL.2	Sicklerville	NJ	08081	United States	Frederick Mathis	333-121-1219	FMathis.CRTA@gmail.com	0.25
5	10005	Twin City T...	100 East Brewery St.	FL.4	Colonial Heights	VA	23834	United States	Jeffery Brown	333-121-1220	JBrown.TCTT@gmail.com	0.2

- Results of execution of STORED PROCEDURE implementing Business Report #4:

```
EXECUTE GetCompaniesByDisRate(0.2);
```



- Results of execution of Matching SELECT QUERY BUSINESS REPORT from the last sub-section proving the results are the same:

	COMPANYNAME	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY	COMPANYREPNAME	CONTACTEMAIL	CORPORATEDISCOUNTPERCENTAGERATE
1	Sky High Travel	8336 Harvey Drive	(null)	Quakertown	PA	18951	United States	Isaac Cortez	ICortez.SkyHighTravel@gmail.com	0.15
2	Liberty Travel	9601 Beechwood Drive	FL.5	Elizabethtown	PA	17022	United States	Libby Terry	LTerry.LibertyTravel@gmail.com	0.15

Stored Procedure for Business Report #5 – Decision to target Future Credit Card Merchants

- Stored Procedure Business Report goals & objectives:

GUIDELINES	ANSWER
Q1 – What is the Business Scenario & Objectives?	▪ In this business scenario, the objective is to determine what are popular credit card merchants. Upon identifying the popular merchants, the company intends to continue using their services. This analysis will provide valuable insights into customer preferences, enabling the company to make an informed decision regarding the selection of their credit card merchant for future use.
Q2 – Target Persona/Decision Maker?	▪ Finance Management team.
Q3 – What is the Decision to be made?	▪ The Finance Management team determine which credit card merchant continue to use. ▪ The decision is based on the number of users for each credit card merchant.
Q4 – What Data is required to make the Decision?	▪ The data required to make decision Company Name, Company Representative Name, Contact Email, and Corporate Discount Percentage Rate
Q5 – WHAT TABLES & COLUMNS Contain the Data Identified in Q4 and what Query Components may be involved in creating the SELECT STATEMENT?	▪ The following Tables and columns would be used: o CreditCard – CreditCardOwnerName, MerchantCode, ActivationStatus are found in this table. o CreditCardMerchant – MerchantCode, MerchatName are found in this table. ▪ We would need a STORED PROCEDURE that includes a SELECT statement that JOINS and queries these two tables CreditCardMerchant & CreditCard to get the data needed. ▪ Multiple records can be returned; therefore, this STORED PROCEDURE will need to use CURSORS.

- Target Query Report from the last sub-section that is the foundation for the Stored Procedure:

```
SELECT CreditCardMerchant.MerchantName,
COUNT(CreditCard.CreditCardOwnerName) as "Number of Customer"
FROM CreditCardMerchant
INNER JOIN CreditCard
ON CreditCardMerchant.MerchantCode = CreditCard.MerchantCode
WHERE ActivationStatus = 1
GROUP BY MerchantName
ORDER BY "Number of Customer" DESC;
```

- Stored Procedure Statement for Business Report #5:

```
CREATE OR REPLACE PROCEDURE TargetFutureCreditCardMerchants(p_ActivationStatus IN CHAR)

IS
        --declare variable needed for SELECT INTO statement
        v_MerchantName VARCHAR2(50);
        v_CreditCardOwnerName VARCHAR2(50);
```

```
        --Declare Cursor
        CURSOR cur_Merchants IS

                -- Query cursor will point to results
                SELECT CreditCardMerchant.MerchantName,
                COUNT(CreditCard.CreditCardOwnerName) as "Number of Customer"
                FROM CreditCardMerchant
                INNER JOIN CreditCard
                ON CreditCardMerchant.MerchantCode = CreditCard.MerchantCode
                WHERE ActivationStatus = p_ActivationStatus
                GROUP BY MerchantName
                ORDER BY "Number of Customer" DESC;



--Start Execution section

BEGIN
        --Open Cursor
        OPEN cur_Merchants; -- open cursor for use
        --loop to display each record returned by cursor
        --Use PL/SQL language control or loop to display each record pointed by cursor
        LOOP
                -- Fetch cursor data
                FETCH cur_Merchants INTO v_MerchantName, v_CreditCardOwnerName;
                EXIT WHEN cur_Merchants%NOTFOUND;
                --Display each record
                DBMS_OUTPUT.PUT_LINE('Customer info: ' ||v_MerchantName|| ' | '
||v_CreditCardOwnerName);
        END LOOP;

        CLOSE cur_Merchants; --close cursor

END TargetFutureCreditCardMerchants;
```

- Data that currently reside in the Tables being queried by STORED PROCEDURE implementing Business Report #5 from the last sub section:

```
SELECT * FROM CreditCard;
```

	CREDITCARDNUMBER	CREDITCARDOWNERNAME	CREDITCARDISSUINGCOMPANY	MERCHANTCODE	EXPDATE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	COUNTRY	ZIPCODE
1	3333333333333331	John Adam	American Express	1	08-DEC-23	10 Fulton ...	Apt 2B	Brooklyn	NY	United States	11203
2	3333333333333332	Olive Bean	American Express	1	25-JUN-24	32 State Ave.	Apt 3B	Brooklyn	NY	United States	11204
3	3333333333333334	Lori Martinez	Capital One	8	25-MAY-25	6 Oakwood ...	Apt 4C	Bronx	NY	United States	10472
4	3333333333333335	Brain Brown	Chase	8	11-DEC-27	1872 Granv...	(null)	Newark	NJ	United States	07662
5	3333333333333336	Adriana Maxwell	Capital One	1	11-DEC-27	8830 Prair...	FL.2	Ashburn	VA	United States	20147
6	3333333333333337	Isaac Cortez	Capital One	3	11-DEC-27	8336 Harve...	(null)	Quakertown	PA	United States	18951
7	3333333333333338	Libby Terry	Capital One	3	11-DEC-27	9601 Beech...	FL.5	Elizabethtown	PA	United States	17022
8	3333333333333339	Frederick Mathis	Chase	8	11-DEC-27	556 Sierra...	FL.2	Sicklerville	NJ	United States	08081
9	3333333333333399	Jeffery Brown	Chase	8	11-DEC-27	100 East B...	FL.4	Colonial Heights	VA	United States	23834

ZIPCODE	CREDITCARDLIMIT	CREDITCARDBALANCE	ACTIVATIONSTATUS
11203	20000	5486.12	1
11204	22000	6486.32	1
10472	18000	286.12	1
07662	20000	5486.32	1
20147	22000	1486.32	1
18951	20000	5486.32	1
17022	20000	5486.32	1
08081	20000	5486.32	1
23834	20000	5486.32	1

```
SELECT * FROM CreditCardMerchant;
```

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

- Results of execution of STORED PROCEDURE implementing Business Report #5:

```
EXECUTE TargetFutureCreditCardMerchants(1);
```

Script Output | Query Result

Task completed in 0.078 seconds

```
Customer info: PayPal | 4
Customer info: Stax by Fattmerchant | 3
Customer info: Dharma Merchant Services | 2


PL/SQL procedure successfully completed.
```

- Results of execution of Matching SELECT QUERY BUSINESS REPORT from the last sub-section proving the results are the same:

	MERCHANTNAME	Number of Customer
1	PayPal	4
2	Stax by Fattmerchant	3
3	Dharma Merchant Services	2

Conclusion

In this project, the EZRental POS system has been upgraded to the Auto Rental Point-of-Sale Management System VERSION 2.0. The NYC-TECH consultant has designed and implemented BUSINESS REPORT QUERIES that are intended for use by decision-makers throughout the organization. These queries provide them with the necessary data to make important BUSINESS DECISIONS. Furthermore, the BUSINESS REPORT QUERIES have been packaged within STORED PROCEDURES to facilitate back-end processing, resulting in improved system design and performance.