EZRENTAL POS

Tyler.Walker@mail.citytech.cuny.edu
NYC-TECH SOLUTIONS INC

PROJECT 1 – EZRENTAL POS PROJECT OBJECTIVES, DEFINITION & DATABASE DESIGN, AND IMPLEMENTATION

Executive Summary:

EZRental Inc. has only focused on the physical aspect of interacting with customers, neglecting the need for business information systems to help improve their businesses efficiency. They needed a way for their business to accept payments from customers and keep track of sales, vehicles, and other aspects of their busy. They have decided to search for a new management software solution to help them achieve their goal of becoming a successful car rental service.

Problem Statement & Objectives:

- EZRental Inc., has hired NYC-Tech Solutions Inc., to design and implement a Client/Server Application Auto Rental Point-of-Sales Management System named EZRental POS, which includes an e-commerce site name EZRental.com.
- Basic objectives and architectures being targeted:
    - The EZRental POS System is designed to allow customers, both retail and corporate, to reserve vehicles for renting like existing in-person or online car rental systems such as Avis, Hertz, Budget, etc.
    - The application is to be designed to support dozens of major cities around the world. In addition, provide a great user experience both in the physical rental agencies as well as online system with the best competitive pricing available in the market.
    - The company currently has rental agency branches in US, Canada, Mexico, United Kingdom, Japan & Australia and looking to expand further globally into other markets in Asia, Africa, and the Mediterranean.

Project Roles & Responsibilities:

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

Person	Role	Description
Mr. Rodriguez	**Program Manager & Project Manager**	▪ 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 that ensures the project stays on time and moving in the right direction. Clear any obstacles impeding the team's progress etc.
Consultant #1: **Mr. Rodriguez**	**Business & Database Analyst**	▪ 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 **Tyler Walker**	**Database Developers**	▪ Use the Normalized Logical Model created by consultant #2 to create the Data Dictionary, Physical Schema Diagram, and Implement the Database Application for the Auto Rental System. ▪ 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 each logical table targeting Oracle 18c Data Types** & **2) Create Physical Schema Diagram**. 2. From these two deliverables, **1) implement the Database Application using Oracle 18c** for the Auto Rental System.
Consultant #6 **Tyler Walker**	**Database Administrator**	▪ As the DB Admin, install the DBMS, maintain, and operate the DBMS throughout its lifetime. ▪ Activities include but not limited to: 1. As DB Admin, you are to **1) Setup & install Oracle 18c DBMS. 2) Oracle SQL Developer Administrative tool**. 2. Also, as DB Admin, you are to **3) Operate & Maintain the DBMS**.

Application Business requirements:

Auto Rental Management System Business Requirements

Business Requirements Interview, Data Gathering Results

- A **Database Business Analyst** was hired by Mr. Rodriguez to ***interview*** **EZRental Inc.,** project business stakeholders and ***gather*** the ***business data requirements*** for the application to derive the database design model.

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

Business Requirements

About Us:

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

Current Challenges:

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

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

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

Our Agencies:

A rental agency is identified by a unique number *rental agency ID*, *agency name*, *address* that is composed of the following elements: *address line1, address line 2*, *city*, *state code*, *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 and get special corporate rates for their employees. On the other hand, retail customers are consumers not associated with a company.

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

- A *Customer ID* number which uniquely identifies the customer, *customer name* which is composed of: ***first name***, ***last name***.
- ***Birth date***, ***Age***, ***Address*** which includes the elements: ***address line1***, ***address line 2***, ***city***, ***state code***, ***zip code*** & ***country***.
- Customer ***phone number*** & ***email*** (unique like all emails), which is required to rent.
- In addition, a driver license is required to rent, therefore we need to capture the unique *driver* ***license number***, ***driver license expiration Date*** and ***driver license state***.
- A very important attribute we need to capture for every customer is the ***credit card***. **You cannot rent one of our vehicles without a credit card**. A credit card includes the following components: ***credit card number*** that uniquely identifies the credit card, credit card ***owner name***, ***merchant name***, ***expiration date***, ***billing address*** composed of ***address line1***, ***address line 2***, ***city***, ***state code***, ***zip code*** & ***country***. Other attributes of credit card are ***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.
- Business rules related to a credit card are:

1. ***A customer can have many credit cards they can use to pay for rental transactions***.

2. ***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 one the same credit card and a credit card can be owned by many customers***.

Business Requirements

Our Customers (Cont.):

For our **corporate customers only**, we must store the following properties: unique *company ID* (we have an ID number for each company), ***company name***, company address which includes the elements: ***address line1***, ***address line 2***, ***city***, ***state code***, ***zip code*** & ***country***, in addition, *company contact* which is composed of ***account representative name***, ***contact phone number*** & ***contact email*** (unique as all email addresses). Finally, we need to store the ***company discount percentage rate*** (in %) which is the discounted percentage applied to a specific corporate customers rental. This discount percentage is applied to the **Vehicle Rental Categories** rates discussed later in these requirements. 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% etc.

Retail Customers can (but don't have to) leverage promotional discounts or coupons obtain from other businesses, internet, magazine and organizations to save money on their rentals. Therefore, data unique to a retail customer that we need to capture for the promotional discount are unique random number ***discount ID*** to uniquely identify a discount, a unique ***discount code*** or coupon code that is an alphanumeric code 10-characters long. This code is generated by the marketing team and published to magazines, newspapers, internet e-commerce sites, etc. Finally, the last attribute is ***discount code description***, 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 can redeem these points for future rentals. Note that the EZPlus 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 a random code generated by the client application that starts with the 3-characters EZP and a 10-digit number e.g., **EZP9999999999**, and finally 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:

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

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

Our Vehicles:

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

- Each vehicle is identified by the random number ***vehicle ID***. In addition, each vehicle is also identified by the alpha-numeric vehicle ***VIN number***. Other attributes include the ***vehicle name*** composed of ***make***, ***model*** & **year**.
- Additional attributes are ***color***, also the ***license plate*** composed of the following components: ***license plate number***, ***license plate state***. More attributes are ***mileage***, ***transmission type*** (values currently used are **Manual**, **Automatic**, **Continuously Variable Transmission (e.g. CVT)**, **Semi-automatic** & **dual-clutch**) and ***seat capacity***.
- All vehicles also have a special identifier we use to track the vehicle status named ***vehicle status ID***. This is a unique number that identifies the status of a vehicle, which works in conjunction with ***vehicle status description*** which describes the status, such as **reserved**, **rented**, **available**, **maintenance**, **not available**, **transferred**, etc. Below Is the list of vehicle status IDs we are currently using and their descriptions:

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

In addition to these attributes shared by all vehicles, the unique characteristics for each of the 4 vehicle types available in all agencies are as follows:

- A **Car** is a vehicle whose ***trunk capacity*** measured in cubic feet volume is advertised to our customers. Customers can decide which vehicles better fits their needs based on the number of luggage they are carrying etc. For example, a luxury Mercedes E class car has a trunk capacity of 18.5 cubic ft.
- An **SUV** has a ***towing capacity*** in pounds. Towing capacity is number in pounds or could also be a decimal in pounds. For example, some of our SUV have a maximum towing capacity of 3,000 pounds. Another attribute of SUV is classification if the SUV is ***All-Wheel-Drive***, which values are a YES/NO or TRUE/FALSE.
- A **Minivan** has the option of ***having a disability package***, which is also a YES/NO, TRUE/FALSE value.
- 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 vans have a typical max load 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. In addition, a reservation or rental can only be for one of these four categories of vehicles not a combination. You can only rent either a car, SUV minivans, cargo van or other for a reservation or rental, not a combination such as a car & SUV at the same time. Each reservation is unique to one vehicle.

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

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***.

Business Requirements (Cont.)

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:

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***.

A rental category contains a list of vehicles depending on the vehicle type: Car (economy, intermediate, full size, luxury), SUV (standard, full size etc.), or Cargo Van etc. Each of these categories have a different price range. Therefore, for a vehicle rental category we need to capture the unique ***vehicle rental category ID*** that identifies the category of the vehicle being reserved or rented, ***category name*** and finally ***category daily rental rate*** for the category. We used a specific code for our vehicle rental category ID, category name & daily rental rate. The table below shows the ID, category names and 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-Passenger Van (12 passengers)	$161.00
15	Van-Cargo Van	$19.95
16	Pick Up-Mid Size	$69.95
17	Pick Up-Full Size	$105.99
18	Motorcycle-Touring	$19.95
19	Motorcycle-Cruiser	$199.99
20	Motorcycle-Scooter	$79.95

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

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 category for a specific rental agency, we wish to capture the following:

- A unique ***reservation ID*** to track the reservation, 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 & Closed.**
4	**Fulfilled & Closed.**
Etc..	**Etc..**

For a reservation w

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 at a rental agency. This means the rental process requires the customer, vehicle, and & rental agency for a rental to be complete***.
2. ***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***, ***rental drop off odometer value*** & ***rental total cost*** which can be calculated based on selected fuel option, insurance option, vehicle rental category price and other factors.

Business Requirements (Cont.)

The Rental Process (Cont.):

- In addition, 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 or return the call with full tank of gas. Therefore, we need to capture the unique ***rental fuel option ID*** or option to choose, ***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.):

- Finally, 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** for any damage or other charges. This deposit is refunded to the customer's credit card when the vehicle is returned in the condition in which it was rented.

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

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 employees consist of customer service agents who interact with our customer to reserve and rent vehicles. In addition, we have auto specialists who work in our services centers servicing our vehicles. In addition, drivers to transport our vehicles from one agency to another and maintenance personnel who maintain our agencies and finally our business team that handles the day-to-day business activities in our agencies and other roles. For now, we are only interested in storing the following data for all these types of employees:

- An ***Employee ID*** which uniquely identifies the employee, ***employee name*** which is composed of: ***first name***, ***last name***, also ***employee address*** which includes the components: ***address line1***, ***address line 2***, ***city***, ***state code***, ***zip code*** & ***country***. Also, ***employee phone***, ***employee job title*** and ***employee email***.

Security & Access:

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

- Employee user accounts
- Customer user accounts

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

For our authorized employees & customer service employees to access the agencies Point-0f-Sales & Back-End Management systems they need to log in by entering a username & password for access to the application. This means every employee owns an employee user account.

An employee user account should store the user ***employee user account ID*** a unique identifier alpha-numeric string that identifies the employee user account, ***employee username*** another unique alpha-numeric that identifies each individual user, and finally the ***employee password*** alpha-numeric that is known only to the user, An employee can own one employee user account only, and an employee user account can only be owned by one employee only since the user account represents the identify of that one employee.

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

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

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

Business Requirements (Cont.)

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 tran 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.

Conclusion:

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

Application Development & Technical Requirements:

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. NYC-Tech Solutions wants 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.

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.

- 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.

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

- **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:
  - 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.

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

- **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:
  - 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 TRANSPORTATIONOPTIONS 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 TRANSPORTATIONSTATUS 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.):

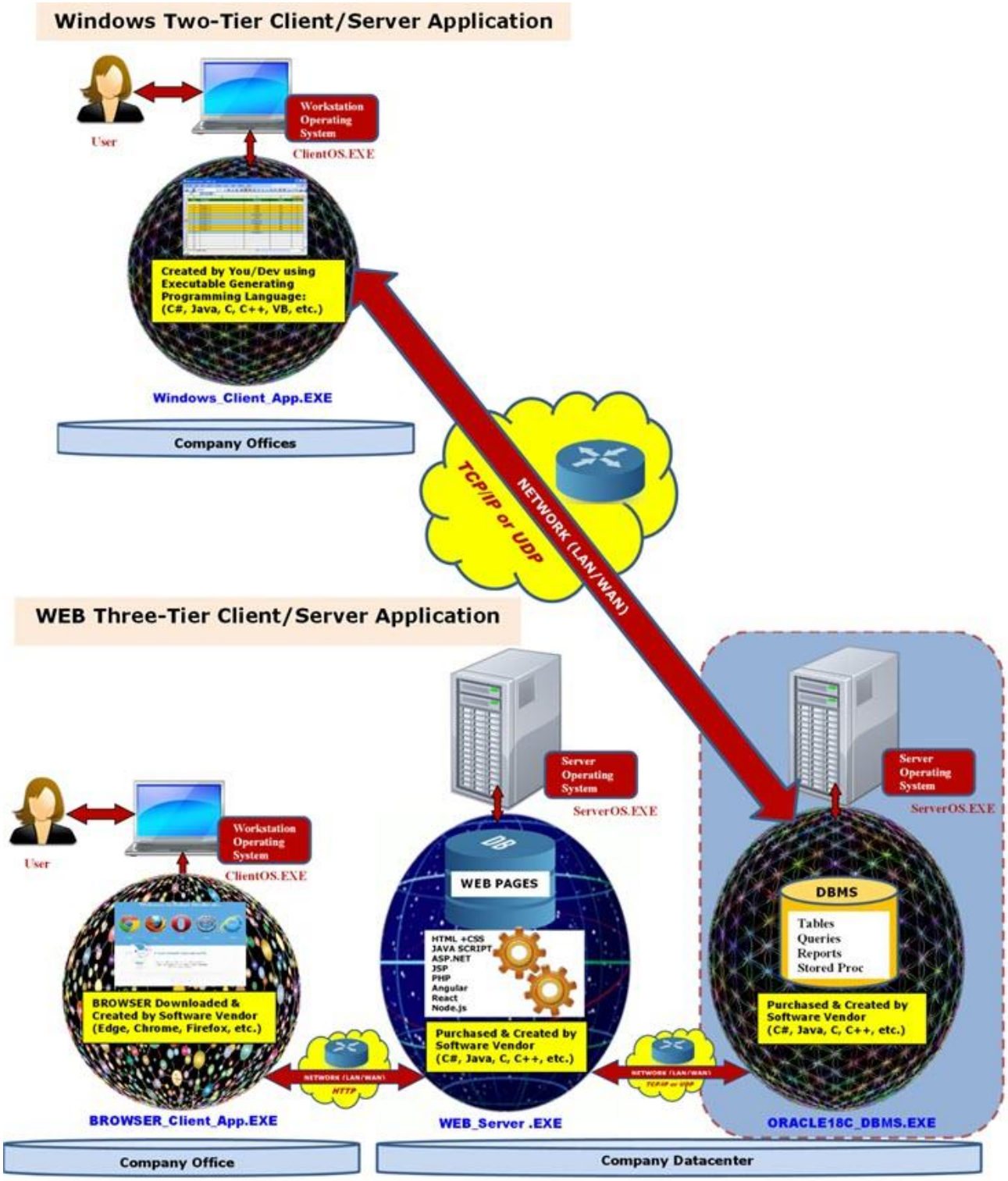
- **EZPlus Rewards Codes:**
  - The EZPlus Reward web page screens with customer's EZPLUS REWARDS CODE fields should be prepopulated with the EZPlus Rewards code for the customer is being applied to. The idea is the user should be able to select the EZPLUS REWARD CODE to apply to a customer entry from 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.

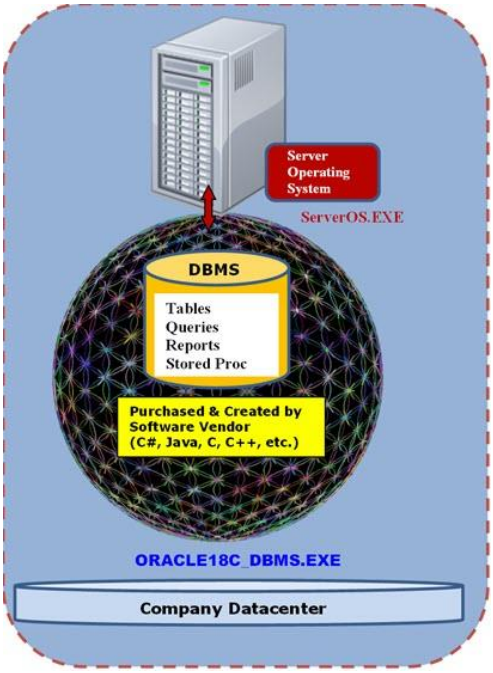
- **Vehicle Rental Category:**
  - Web pages that require the use of the VEHICLE RENTAL CATEGORY fields, must be prepopulated with the list of vehicle rental categories. Based on the business requirements, the current list of vehicle rental categories is as follows:

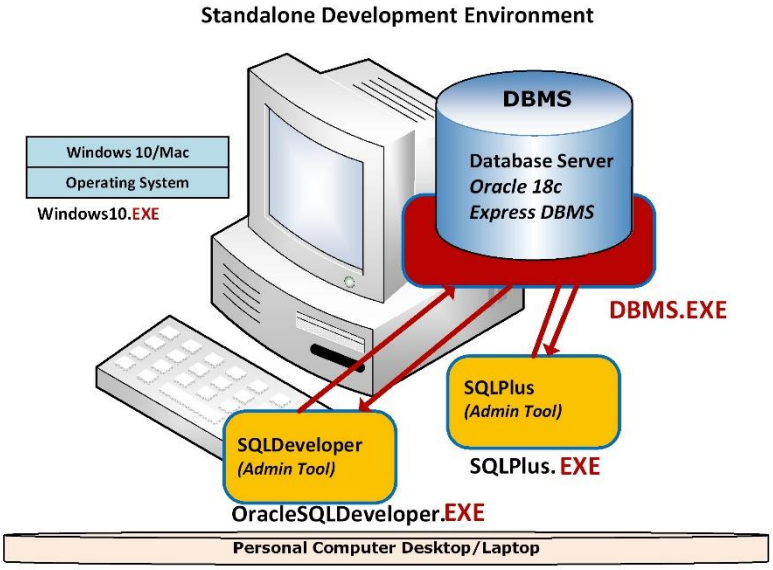
Vehicle Rental Category ID	*Vehicle Rental Category Name*	*Category Daily Rental Rate*
1	Car-Economic	$113.99
2	Car-Compact	$115.99
3	Car-Intermediate	$116.67
4	Car-Standard	$119.99
5	Car-Full Size	$121.99
6	Car-Premium	$127.79
7	Car-Luxury	$139.99
8	SUV-Intermediate	$127.99
9	SUV-Standard	$128.99
10	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

1. Application Physical Technical Architecture:

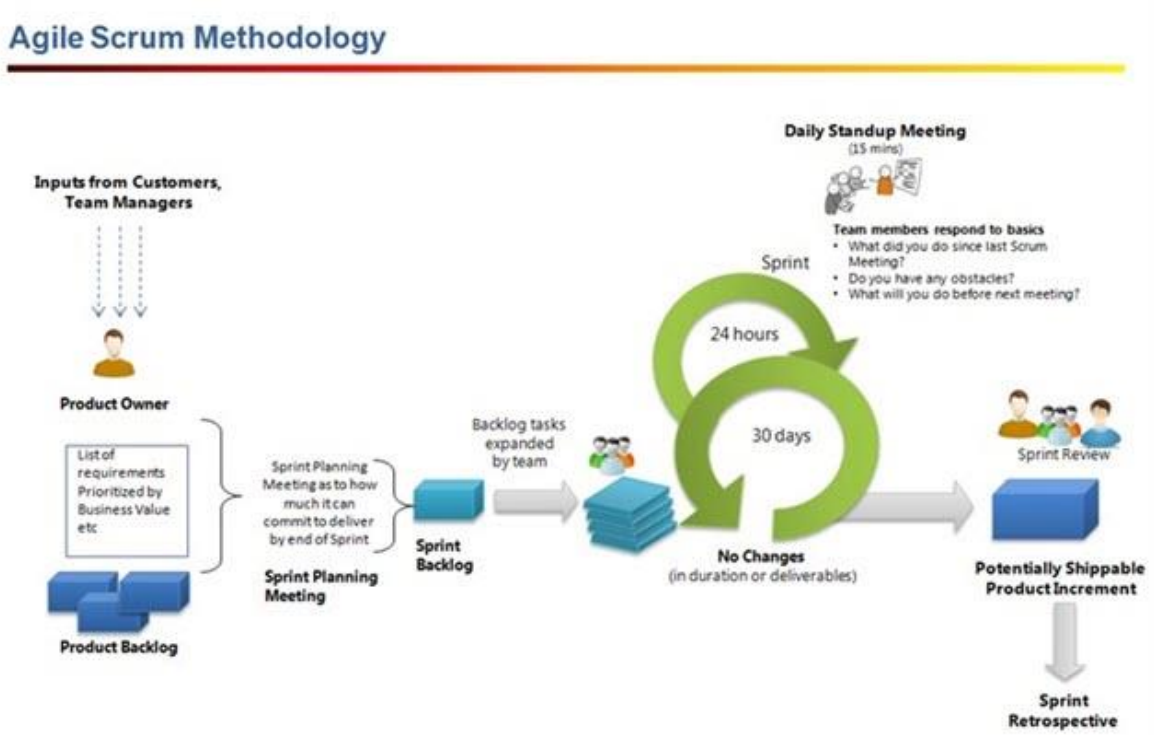


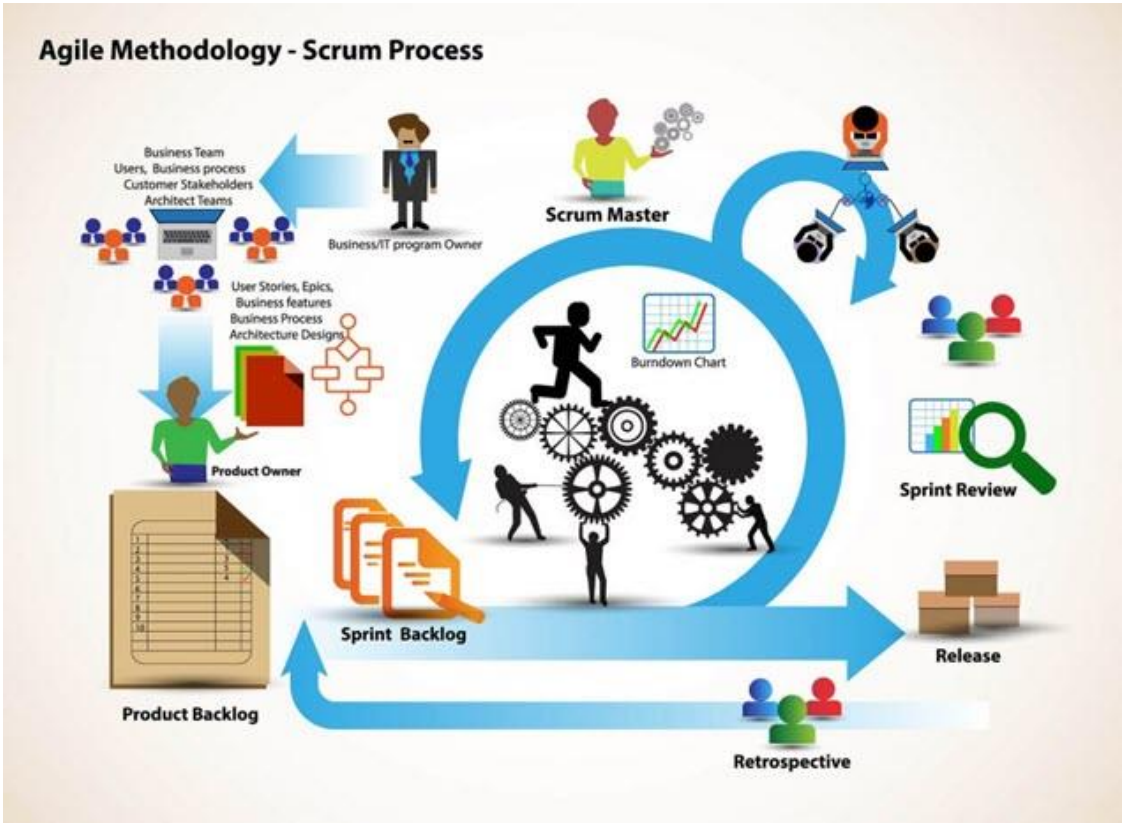
2. Database Management Physical Architecture:



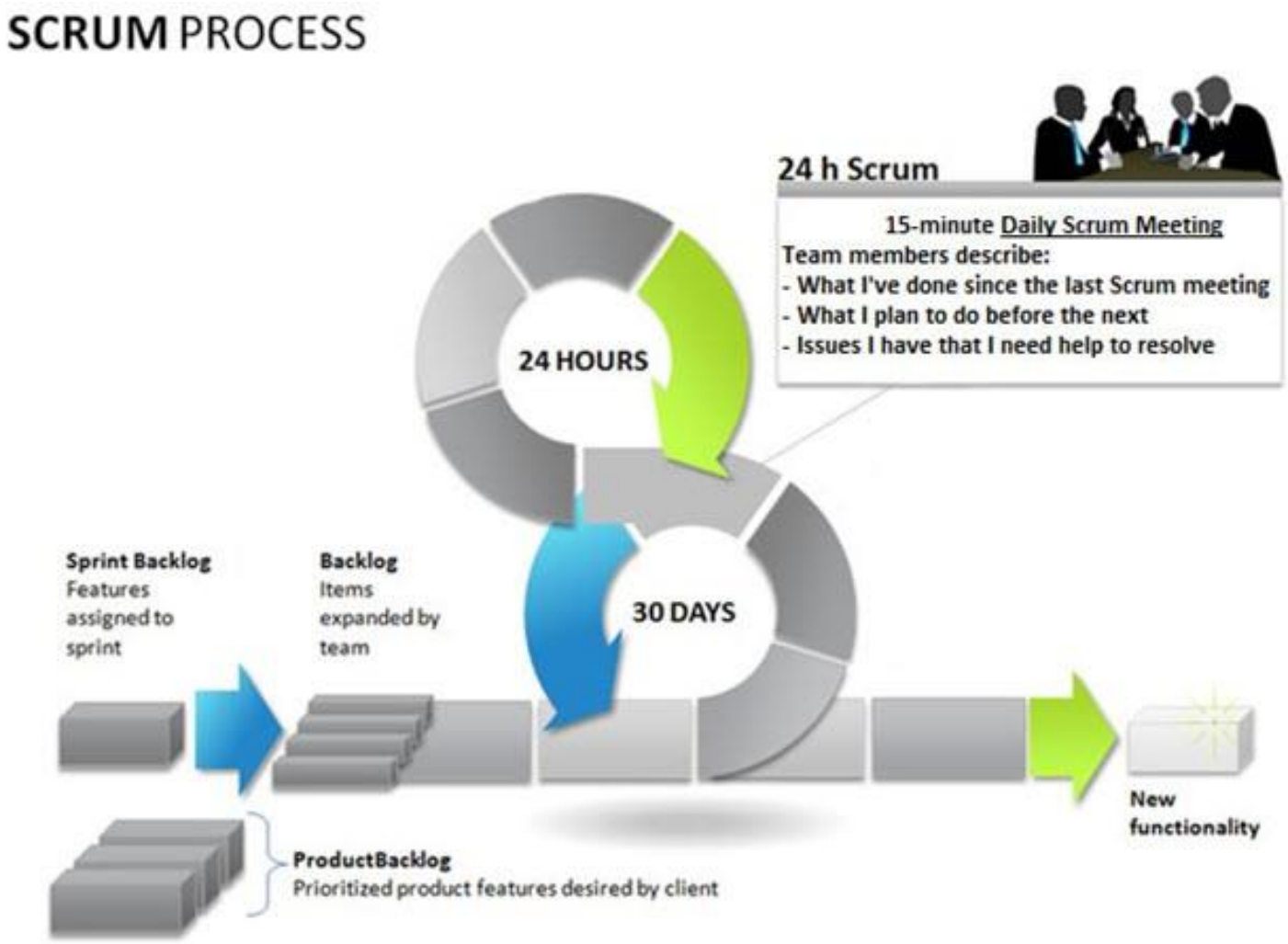


Project Management Methodology:

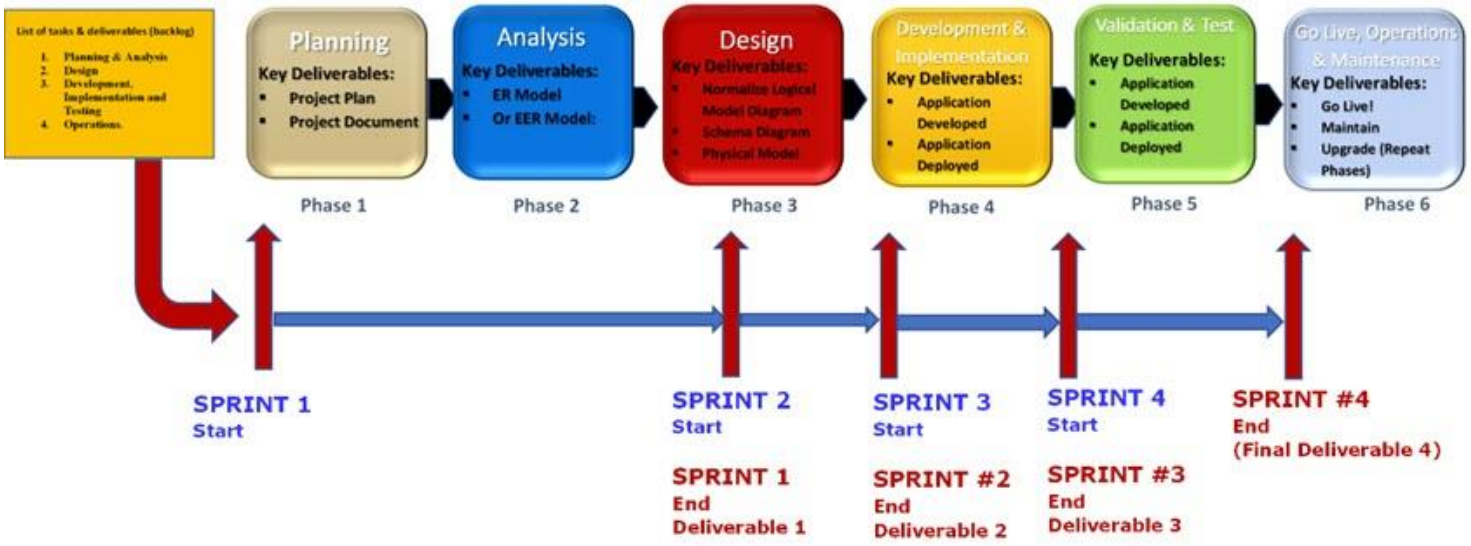




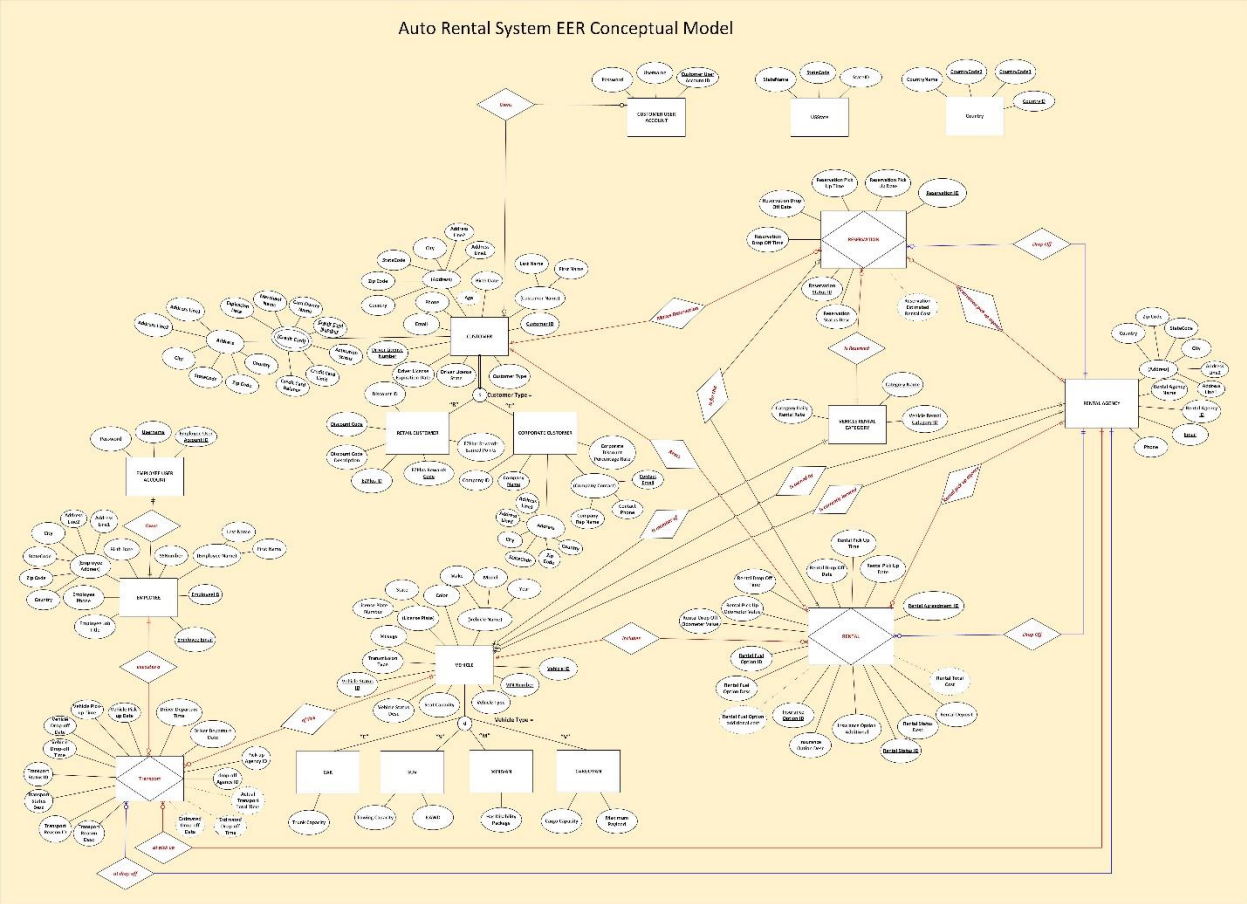
NYC-Tech solutions will employ a scrum process methodology to manage this project's product development. We will host 15-minute daily scrum meetings to discuss what we've done during the previous scrum meeting, what we will do today, and what hindrances we'll have in the future.



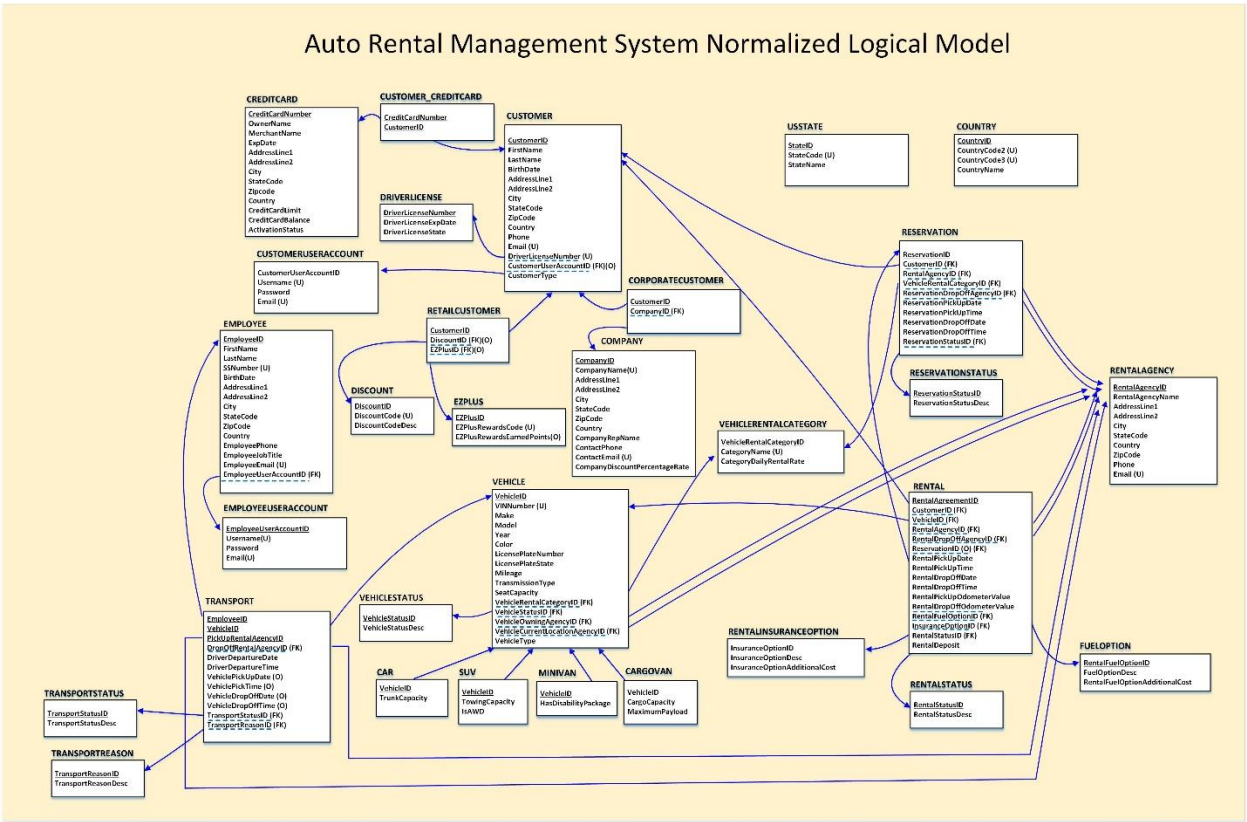
EZRENTAL POS



ER/EER Conceptual Model



Normalized Logical Model



Physical Model Data Dictionary

CUSTOMER						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
CustomerID	NUMBER	NUMBER	YES	Automatically generated – (9) max	PRIMARY KEY	Customer identifier
FirstName	CHARACTER	VARCHAR2	YES	30	-	Customer First Name
LastName	CHARACTER	VARCHAR2	YES	30	-	Customer Last Name
BirthDate	DATE	DATE	YES	DD-MON-YY	-	Date of Birth
AddressLine1	CHARACTER	VARCHAR2	YES	75	-	1st Address Line
AddresLine2	CHARACTER	VARCHAR2	YES	75	-	2nd Address Line
City	CHARACTER	VARCHAR2	YES	13	-	City Name
StateCode	CHARACTER	CHAR	YES	2	-	State code number
Zipcode	CHARACTER	CHAR	YES	10	-	Zip code number
Country	CHARACTER	VARCHAR2	YES	50	-	Country Name
Phone	CHARACTER	CHAR	YES	20	-	Customer Phone Number
Email	CHARACTER	VARCHAR2	YES	75	UNIQUE	Customer Email
DriverLicenseNumber	CHARACTER	VARCHAR2	YES	20	FOREIGN KEY | UNIQUE	Customer Driver License Number
CustomerUserAccountID	RAW	RAW	NO	16	-	Customer user account identifier
CustomerType	CHARACTER	CHAR	YES	1	-	Type of Customer

RETAILCUSTOMER						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
CustomerID	NUMBER	NUMBER	YES	9	PRIMARY KEY	Customer identifier
DiscountID	NUMBER	NUMBER	NO	9	FOREIGN KEY	Discount identifier
EZPlusID	NUMBER	NUMBER	NO	10	FOREIGN KEY	EZPlus identifier

CORPORATECUSTOMER						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
CustomerID	NUMBER	NUMBER	YES	9	PRIAMRY KEY	Customer identifier
CompanyID	NUMBER	NUMBER	YES	9	FOREIGN KEY	Company identifier

DRIVERLICENSE						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
DriverLicenseNumber	CHARACTER	VARCHAR2	YES	20	PRIMARY KEY	Drivers License #
DriverLicenseExpDate	DATE	DATE	YES	-	-	Expiration date

DriverLicenseState	CHARACTER	CHAR	YES	2	-	State of License

DISCOUNT

Attribute/Column Name	General Data Type Name	Oracle Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/ purpose
DiscountID	NUMBER	NUMBER	YES	Automatically generated	PRIMARY KEY	Discount identifier
DiscountCode	CHARACTER	CHAR	YES	10	UNIQUE	Code for Discount
DiscountCodeDesc	VARCHAR2	VARCHAR2	NO	150	-	Description for Discount Code

CUSTOMERCREDITCARD

Attribute/Column Name	General Data Type Name	Oracle Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/ purpose
CardNumber	CHARACTER	VARCHAR2	YES	16	PRIMARY KEY	Customer Credit Card Number
CustomerID	NUMBER	NUMBER	YES	9	PRIMARY KEY	Customer identifier

CUSTOMERUSERACCOUNT

Attribute/Column Name	General Data Type Name	Oracle Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/ purpose
UserAccountID	RAW	RAW	YES	16	PRIMARY KEY | Default SYS_GUID()	UserAccount identifier
Username	CHARACTER	VARCHAR2	YES	50	UNIQUE	Customer username
Password	CHARACTER	VARCHAR2	YES	20	-	Customer password
Email	CHARACTER	VARCHAR2	YES	75	UNIQUE	Customer email

EZPLUS

Attribute/Column Name	General Data Type Name	Oracle Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/ purpose
EZPlusID	NUMBER	NUMBER	YES	Automatically generated | (10)	PRIMARY KEY	EZPlus identifier
EZPlusRewardsCode	CHARACTER	CHAR	YES	13	UNIQUE	Reward Code
EZPlusRewardsEarnedPoints	NUMBER	NUMBER	NO	6	-	Points earned

COMPANY						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
CompanyID	NUMBER	NUMBER	YES	Automatically generated	PRIMARY KEY	Company identifier
CompanyName	CHARACTER	VARCHAR2	YES	50	UNIQUE	Name of company
AddressLine1	CHARACTER	VARCHAR2	YES	75	-	1st address of company
AddressLine2	CHARACTER	VARCHAR2	NO	75	OPTIONAL	2nd address of company
City	CHARACTER	VARCHAR2	YES	50	-	City name
StateCode	CHARACTER	CHAR	YES	2	-	State code
ZipCode	CHARACTER	CHAR	YES	10	-	Zip code
Country	CHARACTER	VARCHAR2	YES	90	-	County Name
CompanyRepName	CHARACTER	VARCHAR2	YES	30	-	Company representative name
ContactPhone	CHARACTER	VARCHAR2	YES	20	UNIQUE	Company phone number
ContactEmail	CHARACTER	VARCHAR2	YES	75	UNIQUE	Company email
CompanyDiscountPercentageRate	NUMBER	NUMBER	YES	3	-	Company discount percentage rate

CREDITCARD						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
CreditCardNumber	CHARACTER	VARCHAR2	YES	16	PRIMARY KEY	Credit card number
OwnerName	CHARACTER	VARCHAR2	YES	50	-	Owner's name
MerchantName	CHARACTER	VARCHAR2	YES	50	-	Merchant's name
ExpDate	DATETIME	DATE	YES	YYYY-MM-DD	-	Credit card expiration date
AddressLine1	CHARACTER	VARCHAR2	YES	75	-	1st address line
AddressLine2	CHARACTER	VARCHAR2	YES	75	OPTIONAL	2nd address line
City	CHARACTER	VARCHAR2	YES	50	-	City name
StateCode	CHARACTER	CHAR	YES	2	-	State code
ZipCode	CHARACTER	CHAR	YES	10	-	Zip code
Country	CHARACTER	VARCHAR2	YES	75	-	Country name
CreditCardLimit	NUMBER	NUMBER	YES	(7,2)	-	Credit card limit
CreditCardBalance	NUMBER	NUMBER	YES	(7,2)	-	Credit card balance
ActivationStatus	NUMBER	NUMBER	YES	1	-	Credit card activation status

VEHICLE						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
VehicleID	NUMBER	NUMBER	YES	Automatically generated	PRIMARY KEY	Vehicle identifier
VINNumber	CHARACTER	VARCHAR2	YES	17	UNIQUE	Vehicle VIN
Make	CHARACTER	VARCHAR2	YES	20	-	Vehicle make
Model	CHARACTER	VARCHAR2	YES	20	-	Vehicle model
Year	NUMBER	NUMBER	YES	4	-	Vehicle year
Color	CHARACTER	VARCHAR2	YES	20	-	Vehicle color
LicensePlateNumber	CHARACTER	VARCHAR2	YES	7	-	Vehicle license plate number
LicensePlateState	CHARACTER	CHAR	YES	2	-	State for license plate
Mileage	NUMBER	NUMBER	YES	6	-	Vehicle mileage
TransmissionType	CHARACTER	VARCHAR2	YES	50	-	Transmission Type
SeatCapacity	NUMBER	NUMBER	YES	1	-	Seat capacity of vehicle
VehicleRentalCategoryID	CHARACTER	VARCHAR2	YES	2	FOREIGN KEY	Rental vehicle category identifier
VehicleStatusID	NUMBER	NUMBER	YES	1	FOREIGN KEY	Vehicle status identifier
VehicleOwningStatusID	NUMBER	NUMBER	YES	1	FOREIGN KEY	Vehicle owning status identifier
VehicleCurrentLocationAgencyID	NUMBER	NUMBER	YES	9	FOREIGN KEY	Vehicle's current location Agency identifier
VehicleType	CHARACTER	CHAR	YES	1	CHECK VEHICLETYPE IN ('C', 'S','M',V')	Type of vehicle

VEHICLERENTALCATEGORY						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
VehicleRentalCategoryID	NUMBER	NUMBER	YES	2	PRIMARY KEY	Vehicle rental category identifier
CategoryName	CHARACTER	VARCHAR2	YES	50	-	Category name
CategoryDailyRentalRate	NUMBER	NUMBER	YES	3	-	Category rental rate

VEHICLESTATUS						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
VehicleStatusID	NUMBER	NUMBER	YES	1	PRIMARY KEY	Identifier for vehicle status
VehicleStatusDesc	CHARACTER	VARCHAR2	YES	30	-	Description for vehicle status

CAR						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
VehicleID	NUMBER	NUMBER	YES	9	PRIMARY KEY	Vehicle identifier
TrunkCapacity	NUMBER	NUMBER	YES	(4,2)		Car's trunk capacity

MINIVAN						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
VehicleID	NUMBER	NUMBER	YES	9	PRIMARY KEY	Vehicle identifier
HasDisabillityPackage	CHARACTER	CHAR	YES	1	CHECK (HASDISABILLITYPACKAGE IN ('YES', 'NO'))	If the minivan has a disability package

SUV						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
VehicleID	NUMBER	NUMBER	YES	9	PRIMARY KEY	Vehicle identifier
TowingCapacity	NUMBER	NUMBER	YES	(6,2)	-	SUV towing capacity in pounds
isAWD	CHARACTER	CHAR	YES	1	-	If the SUV is all wheel drive

CARGOVAN						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
VehicleID	NUMBER	NUMBER	YES	9	PRIMARY KEY	Vehicle identifier
CargoCapacity	NUMBER	NUMBER	YES	3	-	Cargo capacity in cubic ft
MaximumPayload	NUMBER	NUMBER	YES	4	-	Maximum payload of cargo van in lbs.

USSTATE						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
StateID	NUMBER	NUMBER	YES	Automatically generated	PRIMARY KEY	State's number identifier
StateCode	CHARACTER	CHAR	YES	2	UNIQUE	State's code
StateName	CHARACTER	VARCHAR2	YES	40	-	Name of State

COUNTRY						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
CountryID	NUMBER	NUMBER	YES	3	PRIMARY KEY	Country identifier
CountryCode2	CHARACTER	CHAR	YES	2	-	Countries, Country code
CountryCode3	CHARACTER	CHAR	YES	3	-	Countries ISO code
CountryName	CHARACTER	VARCHAR2	YES	75	-	Full name of Country

RESERVATION						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
ReservationID	NUMBER	NUMBER	YES	-	PRIMARY KEY	Reservation identifier
CustomerID	NUMBER	NUMBER	YES	9	FOREIGN KEY	Customer identifier
RentalAgencyID	NUMBER	NUMBER	YES	9	FOREIGN KEY	Rental agency identifier
VehicleRentalCategoryID	NUMBER	NUMBER	YES	2	FOREIGN KEY	Rental category identifier
ReservationDropOffAgencyID	NUMBER	NUMBER	YES	2	FOREIGN KEY	Drop off agency identifier
ReservationPickupDate	DATETIME	DATE	YES	YYYY-MM-DD	-	Pick up date
ReservationPickUpTime	DATETIME	TIMESTAMP	YES	YYYY-MM-DD HH24:MI:SS	-	Pick up time
ReservatuibDropOffDate	DATETIME	DATE	YES	YYYY-MM-DD	-	Drop off date
ReservationDropOffTime	DATETIME	TIMESTAMP	YES	YYYY-MM-DD HH24:MI:SS	-	Drop off time
ReservationStatusID	NUMBER	NUMBER	YES	1	FOREIGN KEY	Reservation status identifier

RESERVATIONSTATUS						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
ReservationStatusID	NUMBER	NUMBER	YES	Automatically generated	PRIMARY KEY	Reservation status identifier
ReservationStatusDesc	CHARACTER	VARCHAR2	YES	100	-	Reservation Status description

RENTALAGENCY						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
RentalAgencyID	NUMBER	NUMBER	YES	5	PRIMARY KEY	Rental Agency Identifier
RentalAgencyName	CHARACTER	VARCHAR2	YES	50	-	Name of rental agency
AddressLine1	CHARACTER	VARCHAR2	YES	75	-	
AddressLine2	CHARACTER	VARCHAR2	YES	75	OPTIONAL	
City	CHARACTER	VARCHAR2	YES	50	-	
StateCode	CHARACTER	CHAR	YES	2	-	
Country	CHARACTER	VARCHAR2	YES	50	-	
ZipCode	CHARACTER	VARCHAR2	YES	10	-	
Phone	CHARACTER	VARCHAR2	YES	20	UNIQUE	Rental Agency number
Email	CHARACTER	VARCHAR2	YES	75	UNIQUE	Rental Agency email

RENTAL						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
RentalAgreementID	NUMBER	NUMBER	YES	Automatically generated	PRIMARY KEY	Rental agreement identifier
CustomerID	NUMBER	NUMBER	YES	9	FOREIGN KEY	Customer identifier
VehicleID	NUMBER	NUMBER	YES	9	FOREIGN KEY	Vehicle identifier
RentalAgencyID	NUMBER	NUMBER	YES	5	FOREIGN KEY	Rental agency identifier
RentalDropOffAgencyID	CHARACTER	VARCHAR2	YES	5	FOREIGN KEY	Rental drop agency identifier
ReservationID	NUMBER	NUMBER	NO	-	FOREIGN KEY	Reservation identifier
RentalPickUpDate	DATETIME	DATE	YES	YYYY-MM-DD	-	Pick-up date
RentalPickUpTime	DATETIME	TIMESTAMP	YES	YYYY-MM-DD HH24:MI:SS	-	Pick-up time
RentalDropOffDate	DATETIME	DATE	YES	YYYY-MM-DD	-	Drop-off date
RentalDropOffTime	DATETIME	TIMESTAMP	YES	YYYY-MM-DD HH24:MI:SS	-	Drop-off time
RentalPickUpOdometerValue	NUMBER	NUMBER	YES	3	-	Odometer value during pick-up
RentalDropOffOdometerValue	NUMBER	NUMBER	YES	3	-	Odometer value during drop-off
RentalFuelOptionID	NUMBER	NUMBER	YES	-	FOREIGN KEY	Fuel option identifier
InsuranceOptionID	NUMBER	NUMBER	YES	-	FOREIGN KEY	Insurance option identifier
RentalStatusID	NUMBER	NUMBER	YES	-	FOREIGN KEY	Rental status identifier
RentalDeposit	NUMBER	NUMBER	YES	(8,2)	-	Rental deposit amount

RENTALINSURANCEOPTION						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
InsuranceOptionID	NUMBER	NUMBER	YES	1	PRIMARY KEY	Insurance option identifier
InsuranceOptionDesc	CHARACTER	VARCHAR2	YES	200	-	Insurance option Description
InsuranceOptionAdditionalCost	NUMBER	NUMBER	YES	(4,2)	-	Additional cost per day for rental insurance

RENTALSTATUS						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
RentalStatusID	NUMBER	NUMBER	YES	1	PRIMARY KEY	Rental status identifier
RentalStatusDesc	CHARACTER	VARCHAR2	YES	100	-	Rental status description

FUELOPTION						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
FuelOptionID	NUMBER	NUMBER	YES	1	PRIMARY KEY	Fuel option identifier
FuelOptionDesc	CHARACTER	VARCHAR2	YES	75	-	Fuel option description

EMPLOYEE						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
EmployeeID	NUMBER	NUMBER	YES	9	PRIMARY KEY	Employee identifier
FirstName	CHARACTER	VARCHAR2	YES	20	-	Employee first name
LastName	CHARACTER	VARCHAR2	YES	20	-	Employee last name
SSNumber	CHARACTER	CHAR	YES	9	UNIQUE	Employee social security number
BirthDate	DATETIME	DATE	YES	YYYY-MM-DD	-	Employee date of birth
AddressLine1	CHARACTER	VARCHAR2	YES	75	-	1st Address line
AddressLine2	CHARACTER	VARCHAR2	YES	75	OPTIONAL	2nd Address line
City	CHARACTER	VARCHAR2	YES	50	-	City name
StateCode	CHARACTER	CHAR	YES	2	-	State code
ZipCode	CHARACTER	VARCHAR	YES	10	-	Zip code
Country	CHARACTER	VARCHAR2	YES	75	-	Country
EmployeePhone	CHARACTER	VARCHAR2	YES	20	-	Employee phone number
EmployeeJobTitle	CHARACTER	VARCHAR2	YES	30	-	Employee job title
EmployeeEmail	CHARACTER	VARCHAR2	YES	75	UNIQUE	Employee email address
EmployeeUserAccountID	RAW	RAW	YES	16	FOREIGN KEY	Employee user account identifier

TRANSPORT						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
EmployeeID	NUMBER	NUMBER	YES	9	PRIMARY KEY	Employee identifier
VehicleID	NUMBER	NUMBER	YES	9	PRIMARY KEY	Vehicle identifier
PickUpRentalAgencyID	NUMBER	NUMBER	YES	9	PRIMARY KEY	Pick up rental agency identifier
DropOFfRentalAgencyID	NUMBER	NUMBER	YES	9	-	Drop off rental agency identifier
DriverDepartureDate	DATETIME	DATE	YES	YYYY-MM-DD	-	Driver's departure date
DriverDepartureTime	NUMBER	NUMBER	YES	4	-	Driver's departure time
VehiclePickUpDate	DATETIME	DATE	YES	YYYY-MM-DD	OPTIONAL	Vehicle's pick-up date
VehiclePickUpTime	NUMBER	NUMBER	YES	4	OPTIONAL	Vehicle's pick-up time
VehicleDropOffDate	DATETIME	DATE	YES	YYYY-MM-DD	OPTIONAL	Vehicle drop off date
VehicleDropOffTime	NUMBER	NUMBER	YES	4	OPTIONAL	Vehicle drop off time
TransportStatusID	NUMBER	NUMBER	YES	1	FOREIGN KEY	Transport status identifier
TransportReasonID	NUMBER	NUMBER	YES	1	FOREIGN KEY	Transport reason identifier

TRANSPORTSTATUS						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
TransportStatusID	NUMBER	NUMBER	YES	1	PRIMARY KEY	Transport Status identifier
TransportStatusDesc	CHARACTER	VARCHAR2	YES	50	-	Transport status description

TRANSPORTREASON						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
TransportReasonID	NUMBER	NUMBER	YES	1	PRIMARY KEY	Transport reason identifier
TransportReasonDesc	CHARACTER	VARCHAR2	YES	100	-	Transport reason description

EMPLOYEEUSERACCOUNT						
Attribute/Column Name	**General Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
UserAccountID	RAW	RAW	YES	16	PRIMARY KEY | Default SYS_GUID()	Employee user account identifier
Username	CHARACTER	VARCHAR2	YES	50	UNIQUE	Employee username
Password	CHARACTER	VARCHAR2	YES	20	-	Employee password
Email	CHARACTER	VARCHAR2	NO	75	UNIQUE	Employee Email

Development & Implementation

```
CREATE TABLE CUSTOMERUSERACCOUNT(
   USERACCOUNTID RAW(16) DEFAULT SYS_GUID() PRIMARY KEY,
   USERNAME VARCHAR2(50) NOT NULL UNIQUE,
   PASSWORD VARCHAR2(20) NOT NULL,
   EMAIL VARCHAR2(75) NOT NULL UNIQUE
);
CREATE TABLE DRIVERLICENSE(
   DRIVERLICENSENUMBER VARCHAR2(20) PRIMARY KEY,
   DRIVERLICENSEEXPDATE DATE NOT NULL,
   DRIVERLICENSESTATE CHAR(2) NOT NULL
);

CREATE TABLE DISCOUNT(
   DISCOUNTID NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
   DISCOUNTCODE CHAR(10) NOT NULL,
   DISCOUNTCODEDESC VARCHAR2(150) NOT NULL
);

CREATE TABLE EZPLUS(
   EZPLUSID NUMBER(10) PRIMARY KEY,
   EZPLUSREWARDSCODE CHAR(13) NOT NULL UNIQUE,
   EZPLUSREWARDSEARNEDPOINTS NUMBER(6) NOT NULL
);

CREATE TABLE CUSTOMER(
   CUSTOMERID NUMBER(9) GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
```

```
    FIRSTNAME VARCHAR2(30) NOT NULL,
    LASTNAME VARCHAR2(30) NOT NULL,
    BIRTHDATE DATE NOT NULL,
    ADDRESSLINE1 VARCHAR2(75) NOT NULL,
    ADDRESSLINE2 VARCHAR2(75) NULL,
    CITY VARCHAR2(15) NOT NULL,
    STATECODE CHAR(2) NOT NULL,
    ZIPCODE VARCHAR2(10) NOT NULL,
    COUNTRY VARCHAR2(50) NOT NULL,
    PHONE VARCHAR2(11) NOT NULL,
    EMAIL VARCHAR2(75) NOT NULL UNIQUE,
    DRIVERLICENSENUMBER VARCHAR2(20) NOT NULL UNIQUE,
    CUSTOMERUSERACCOUNTID RAW(16) NOT NULL,
    CUSTOMERTYPE CHAR(1) NOT NULL,
    FOREIGN KEY (CUSTOMERUSERACCOUNTID) REFERENCES CUSTOMERUSERACCOUNT(USERACCOUNTID),
    FOREIGN KEY (DRIVERLICENSENUMBER) REFERENCES DRIVERLICENSE(DRIVERLICENSENUMBER),
    CONSTRAINT CK_CUST_TYPE CHECK (UPPER(CUSTOMERTYPE) IN ('R', 'C'))
);

CREATE TABLE RETAILCUSTOMER(
    CUSTOMERID NUMBER(9) NOT NULL PRIMARY KEY,
    DISCOUNTID NUMBER(9) NULL,
    EZPLUSID NUMBER(10) NULL,
    FOREIGN KEY (CUSTOMERID) REFERENCES CUSTOMER(CUSTOMERID),
    FOREIGN KEY (DISCOUNTID) REFERENCES DISCOUNT(DISCOUNTID),
    FOREIGN KEY (EZPLUSID) REFERENCES EZPLUS(EZPLUSID)
);
```

```
CREATE TABLE COMPANY(
   COMPANYID NUMBER(5) GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
   COMPANYNAME VARCHAR2(50) NOT NULL UNIQUE,
   ADDRESSLINE1 VARCHAR2(75) NOT NULL,
   ADDRESSLINE2 VARCHAR2(75) NULL,
   CITY VARCHAR2(50) NOT NULL,
   STATECODE CHAR(2) NOT NULL,
   ZIPCODE VARCHAR2(10) NOT NULL,
   COUNTRY VARCHAR2(90) NOT NULL,
   COMPANYREPNAME VARCHAR2(30) NOT NULL,
   CONTACTPHONE VARCHAR2(11) NOT NULL UNIQUE,
   CONTACTEMAIL VARCHAR2(75) NOT NULL UNIQUE,
   COMPANYDISCOUNTPERCENTAGE NUMBER(3,1) NOT NULL,
   CONSTRAINT CK_COMPANYLIMIT CHECK (COMPANYID BETWEEN 1 AND 20000)
);

CREATE TABLE CORPORATECUSTOMER(
   CUSTOMERID NUMBER(9) PRIMARY KEY,
   COMPANYID NUMBER(9) NOT NULL,
   FOREIGN KEY (CUSTOMERID) REFERENCES CUSTOMER(CUSTOMERID),
   FOREIGN KEY (COMPANYID) REFERENCES COMPANY(COMPANYID)
);

CREATE TABLE CREDITCARD(
   CREDITCARDNUMBER VARCHAR2(16) PRIMARY KEY,
   OWNERNAME VARCHAR2(50) NOT NULL,
```

```
    MERCHANTNAME VARCHAR2(50) NOT NULL,
    EXPDATE DATE NOT NULL,
    ADDRESSLINE1 VARCHAR2(75) NOT NULL,
    ADDRESSLINE2 VARCHAR2(75) NULL,
    CITY VARCHAR2(50) NOT NULL,
    STATECODE CHAR(2) NOT NULL,
    ZIPCODE VARCHAR2(10) NOT NULL,
    COUNTRY VARCHAR2(75) NOT NULL,
    CREDITCARDLIMIT NUMBER(7,2) NOT NULL,
    CREDITCARDBALANCE NUMBER(7,2) NOT NULL,
    ACTIVATIONSTATUS CHAR(1) NOT NULL,
    CONSTRAINT ck_ISACTIVATED CHECK (ACTIVATIONSTATUS IN ('0','1')),
    CONSTRAINT ck_CREDITCARDMIN CHECK (NOT LENGTH(CREDITCARDNUMBER) <= 12)
);

CREATE TABLE CUSTOMERCREDITCARD(
    CARDNUMBER VARCHAR2(16) NOT NULL,
    CUSTOMERID NUMBER(9) NOT NULL,
    CONSTRAINT PK_CUSTOMERCREDITCARD PRIMARY KEY (CARDNUMBER, CUSTOMERID),
    FOREIGN KEY (CARDNUMBER) REFERENCES CREDITCARD(CREDITCARDNUMBER),
    FOREIGN KEY (CUSTOMERID) REFERENCES CUSTOMER(CUSTOMERID)
);

CREATE TABLE VEHICLERENTALCATEGORY(
    VEHICLERENTALCATEGORYID NUMBER(2) PRIMARY KEY,
    CATEGORYNAME VARCHAR2(50) NOT NULL,
    CATEGORYDAILYRENTALRATE NUMBER(5,2) NOT NULL
```

```
);

CREATE TABLE VEHICLESTATUS(
   VEHICLESTATUSID NUMBER(1) PRIMARY KEY,
   VEHICLESTATUSDESC VARCHAR2(30) NOT NULL
);

CREATE TABLE RENTALAGENCY(
   RENTALAGENCYID NUMBER(5) PRIMARY KEY,
   RENTALAGENCYNAME VARCHAR2(50) NOT NULL,
   ADDRESSLINE1 VARCHAR2(75) NOT NULL,
   ADDRESSLINE2 VARCHAR2(75) NULL,
   CITY VARCHAR(50) NOT NULL,
   STATECODE CHAR(2) NOT NULL,
   COUNTRY VARCHAR2(50) NOT NULL,
   ZIPCODE VARCHAR2(10) NOT NULL,
   PHONE VARCHAR2(20) NOT NULL,
   EMAIL VARCHAR2(75) NOT NULL UNIQUE
);

CREATE TABLE VEHICLE(
   VEHICLEID NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
   VINNUMBER VARCHAR2(17) UNIQUE NOT NULL,
   MAKE VARCHAR2(30) NOT NULL,
   MODEL VARCHAR2(30) NOT NULL,
   YEAR NUMBER(4) NOT NULL,
   COLOR VARCHAR2(20) NOT NULL,
```

```
   LICENSEPLATENUMBER VARCHAR2(10) NOT NULL,
   MILEAGE NUMBER(6) NOT NULL,
   TRANSMISSIONTYPE VARCHAR2(50) NOT NULL,
   SEATCAPACITY NUMBER(1) NOT NULL,
   VEHICLERENTALCATEGORYID NUMBER(2) NOT NULL,
   VEHICLESTATUSID NUMBER(1) NOT NULL ,
   VEHICLEOWNINGAGENCYID NUMBER(9) NOT NULL,
   VEHICLECURRENTLOCATIONAGENCYID NUMBER(9) NOT NULL,
   VEHICLETYPE CHAR(1) CHECK (VEHICLETYPE IN ('C', 'S','M','V')) NOT NULL,
   FOREIGN KEY (VEHICLERENTALCATEGORYID) REFERENCES VEHICLERENTALCATEGORY
(VEHICLERENTALCATEGORYID),
   FOREIGN KEY (VEHICLESTATUSID) REFERENCES VEHICLESTATUS(VEHICLESTATUSID),
   FOREIGN KEY (VEHICLEOWNINGAGENCYID) REFERENCES RENTALAGENCY(RENTALAGENCYID),
   FOREIGN KEY (VEHICLECURRENTLOCATIONAGENCYID) REFERENCES RENTALAGENCY(RENTALAGENCYID),
   CONSTRAINT CK_TRANSMISSIONTYPE CHECK (UPPER(TRANSMISSIONTYPE) IN ('MANUAL', 'AUTOMATIC',
'CONTINUOUSLY VARIABLE TRANSMISSION', 'SEMI-AUTOMATIC', 'DUAL-CLUTCH'))
);

CREATE TABLE CAR(
   VEHICLEID NUMBER(9) PRIMARY KEY,
   TRUNKCAPACITY NUMBER(4,2) NOT NULL,
   FOREIGN KEY (VEHICLEID) REFERENCES VEHICLE(VEHICLEID)
);

CREATE TABLE MINIVAN(
   VEHICLEID NUMBER(9) PRIMARY KEY,
   HASDISABILLITYPACKAGE CHAR(1) NOT NULL,
   FOREIGN KEY (VEHICLEID) REFERENCES VEHICLE(VEHICLEID),
```

```
    CONSTRAINT ck_HASDISABILLITYPACKAGE CHECK (HASDISABILLITYPACKAGE IN ('0','1'))
);

CREATE TABLE SUV(
    VEHICLEID NUMBER(9) PRIMARY KEY,
    TOWINGCAPACITY NUMBER(6,2) NOT NULL,
    ISAWD CHAR(1) NOT NULL,
    CONSTRAINT ck_ISAWD CHECK (ISAWD IN ('0','1')),
    FOREIGN KEY (VEHICLEID) REFERENCES VEHICLE(VEHICLEID)
);

CREATE TABLE CARGOVAN(
    VEHICLEID NUMBER(9) PRIMARY KEY,
    CARGOCAPACITY NUMBER(3) NOT NULL,
    MAXIMUMPAYLOAD NUMBER(4) NOT NULL,
    FOREIGN KEY (VEHICLEID) REFERENCES VEHICLE(VEHICLEID)
);

CREATE TABLE USSTATE(
    STATEID NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
    STATECODE CHAR(2) UNIQUE NOT NULL,
    STATENAME VARCHAR2(40) NOT NULL
);

CREATE TABLE COUNTRY(
    COUNTRYID NUMBER(3) PRIMARY KEY,
    COUNTRYCODE2 CHAR(2) NOT NULL,
```

```
   COUNTRYCODE3 CHAR(3) NOT NULL,
   COUNTRYNAME VARCHAR2(75) NOT NULL
);



CREATE TABLE RENTALINSURANCEOPTION(
   INSURANCEOPTIONID NUMBER(1) PRIMARY KEY,
   INSURANCEOPTIONDESC VARCHAR2(200) NOT NULL,
   INSURANCEOPTIONADDITIONALCOST NUMBER(4,2) NOT NULL
);
CREATE TABLE FUELOPTION(
   FUELOPTIONID NUMBER(1) PRIMARY KEY,
   FUELOPTIONDESC VARCHAR2(100) NOT NULL
);

CREATE TABLE RENTALSTATUS(
   RENTALSTATUSID NUMBER(1) PRIMARY KEY,
   RENTALSTATUSDESC VARCHAR2(100) NOT NULL
);

CREATE TABLE RESERVATIONSTATUS (
   RESERVATIONSTATUSID NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
   RESERVATIONSTATUSDESC VARCHAR2(100) NOT NULL
);

CREATE TABLE RESERVATION(
   RESERVATIONID NUMBER(9) PRIMARY KEY,
```

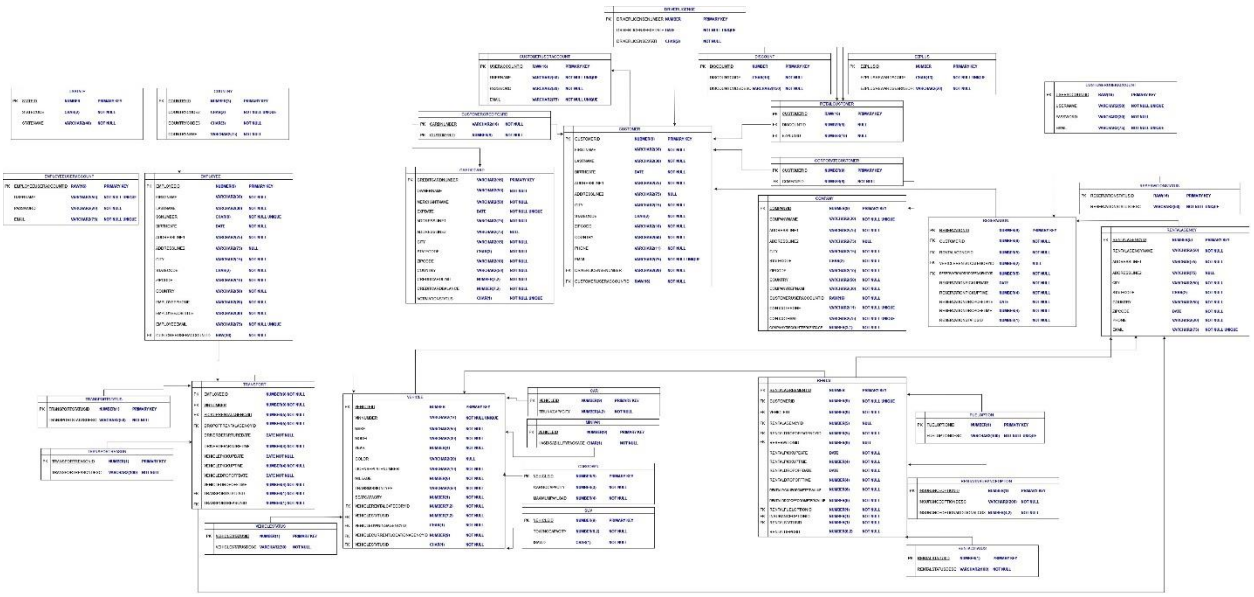
```
    CUSTOMERID NUMBER(9) NOT NULL,
    RENTALAGENCYID NUMBER(5) NOT NULL,
    VEHICLERENTALCATEGORYID NUMBER(2) NOT NULL,
    RESERVATIONDROPOFFAGENCYID NUMBER(5) NOT NULL,
    RESERVATIONPICKUPDATE DATE NOT NULL,
    RESERVATIONPICKUPTIME NUMBER(4) NOT NULL,
    RESERVATIONDROPOFFDATE DATE NOT NULL,
    RESERVATIONDROPOFFTIME NUMBER(4) NOT NULL,
    RESERVATIONSTATUSID NUMBER(1) NOT NULL,
    FOREIGN KEY (CUSTOMERID) REFERENCES CUSTOMER(CUSTOMERID),
    FOREIGN KEY (RENTALAGENCYID) REFERENCES RENTALAGENCY(RENTALAGENCYID),
    FOREIGN KEY (VEHICLERENTALCATEGORYID) REFERENCES
VEHICLERENTALCATEGORY(VEHICLERENTALCATEGORYID),
    FOREIGN KEY (RESERVATIONDROPOFFAGENCYID) REFERENCES RENTALAGENCY(RENTALAGENCYID),
    FOREIGN KEY (RESERVATIONSTATUSID) REFERENCES RESERVATIONSTATUS(RESERVATIONSTATUSID)
);

CREATE TABLE RENTAL(
    RENTALAGREEMENTID NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
    CUSTOMERID NUMBER(9) NOT NULL,
    VEHICLEID NUMBER(9) NOT NULL,
    RENTALAGENCYID NUMBER(5) NOT NULL,
    RENTALDROPOFFAGENCYID NUMBER(5) NOT NULL,
    RESERVATIONID NUMBER(9) NULL,
    RENTALPICKUPDATE DATE NOT NULL,
    RENTALPICKUPTIME NUMBER(4) NOT NULL,
    RENTALDROPOFFDATE DATE NOT NULL,
```

```
    RENTALDROPOFFTIME NUMBER(4) NOT NULL,
    RENTALPICKUPODOMETERVALUE NUMBER(6) NOT NULL,
    RENTALDROPOFFODOMETERVALUE NUMBER(6) NOT NULL,
    RENTALFUELOPTIONID NUMBER(1) NOT NULL,
    INSURANCEOPTIONID NUMBER(1) NOT NULL,
    RENTALSTATUSID NUMBER(1) NOT NULL,
    RENTALDEPOSIT NUMBER(8,2) NOT NULL,
    FOREIGN KEY (CUSTOMERID) REFERENCES CUSTOMER(CUSTOMERID),
    FOREIGN KEY (VEHICLEID) REFERENCES VEHICLE(VEHICLEID),
    FOREIGN KEY (RENTALAGENCYID) REFERENCES RENTALAGENCY(RENTALAGENCYID),
    FOREIGN KEY (RENTALDROPOFFAGENCYID) REFERENCES RENTALAGENCY(RENTALAGENCYID),
    FOREIGN KEY (RESERVATIONID) REFERENCES  RESERVATION(RESERVATIONID),
    FOREIGN KEY (RENTALFUELOPTIONID) REFERENCES FUELOPTION(FUELOPTIONID),
    FOREIGN KEY (INSURANCEOPTIONID) REFERENCES RENTALINSURANCEOPTION(INSURANCEOPTIONID),
    FOREIGN KEY (RENTALSTATUSID) REFERENCES RENTALSTATUS(RENTALSTATUSID)
);

CREATE TABLE EMPLOYEEUSERACCOUNT(
    USERACCOUNTID RAW(16) DEFAULT SYS_GUID() PRIMARY KEY,
    USERNAME VARCHAR2(50) NOT NULL,
    PASSWORD VARCHAR2(20) NOT NULL,
    EMAIL VARCHAR2(75) NOT NULL UNIQUE
);

CREATE TABLE EMPLOYEE(
    EMPLOYEEID NUMBER PRIMARY KEY,
    FIRSTNAME VARCHAR2(20) NOT NULL,
```

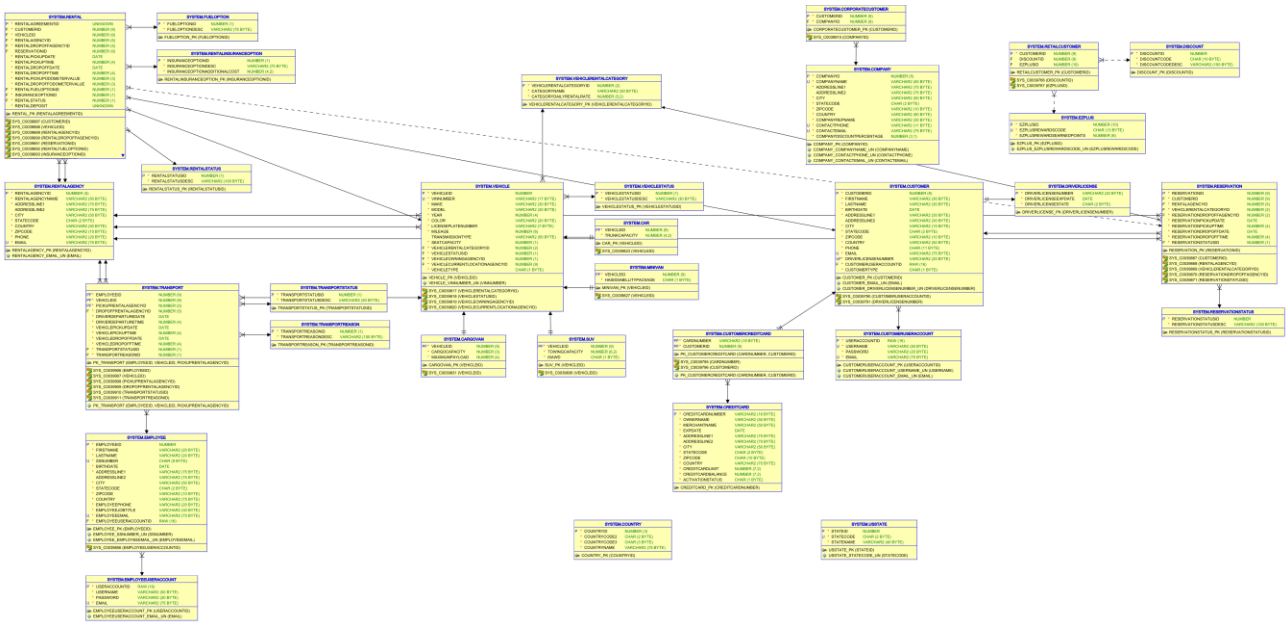
```
    LASTNAME VARCHAR2(20) NOT NULL,
    SSNUMBER CHAR(9) NOT NULL UNIQUE,
    BIRTHDATE DATE NOT NULL,
    ADDRESSLINE1 VARCHAR2(75) NOT NULL,
    ADDRESSLINE2 VARCHAR2(75) NULL,
    CITY VARCHAR2(50) NOT NULL,
    STATECODE CHAR(2) NOT NULL,
    ZIPCODE VARCHAR2(10) NOT NULL,
    COUNTRY VARCHAR2(75) NOT NULL,
    EMPLOYEEPHONE VARCHAR2(20) NOT NULL,
    EMPLOYEEJOBTITLE VARCHAR2(30) NOT NULL,
    EMPLOYEEEMAIL VARCHAR2(75) NOT NULL UNIQUE,
    EMPLOYEEUSERACCOUNTID RAW(16) NOT NULL,
    FOREIGN KEY (EMPLOYEEUSERACCOUNTID) REFERENCES EMPLOYEEUSERACCOUNT(USERACCOUNTID)
);

CREATE TABLE TRANSPORTSTATUS(
    TRANSPORTSTATUSID NUMBER(1) PRIMARY KEY,
    TRANSPORTSTATUSDESC VARCHAR2(50) NOT NULL
);

CREATE TABLE TRANSPORTREASON(
    TRANSPORTREASONID NUMBER(1) PRIMARY KEY,
    TRANSPORTREASONDESC VARCHAR2(100) NOT NULL
);
CREATE TABLE TRANSPORT(
    EMPLOYEEID NUMBER(9) NOT NULL,
```

```
    VEHICLEID NUMBER(9) NOT NULL,
    PICKUPRENTALAGENCYID NUMBER(5) NOT NULL,
    DROPOFFRENTALAGENCYID NUMBER(5) NOT NULL,
    DRIVERDEPARTUREDATE DATE NOT NULL,
    DRIVERDEPARTURETIME NUMBER(4) NOT NULL,
    VEHICLEPICKUPDATE DATE NOT NULL,
    VEHICLEPICKUPTIME NUMBER(4) NOT NULL,
    VEHICLEDROPOFFDATE DATE NOT NULL,
    VEHICLEDROPOFFTIME NUMBER(4) NOT NULL,
    TRANSPORTSTATUSID NUMBER(1) NOT NULL,
    TRANSPORTREASONID NUMBER(1) NOT NULL,
    FOREIGN KEY (EMPLOYEEID) REFERENCES EMPLOYEE(EMPLOYEEID),
    FOREIGN KEY (VEHICLEID) REFERENCES VEHICLE(VEHICLEID),
    FOREIGN KEY (PICKUPRENTALAGENCYID) REFERENCES RENTALAGENCY(RENTALAGENCYID),
    FOREIGN KEY (DROPOFFRENTALAGENCYID) REFERENCES RENTALAGENCY(RENTALAGENCYID),
    FOREIGN KEY (TRANSPORTSTATUSID) REFERENCES TRANSPORTSTATUS(TRANSPORTSTATUSID),
    FOREIGN KEY (TRANSPORTREASONID) REFERENCES TRANSPORTREASON(TRANSPORTREASONID)
);
ALTER TABLE TRANSPORT ADD CONSTRAINT PK_TRANSPORT PRIMARY KEY(EMPLOYEEID, VEHICLEID,
PICKUPRENTALAGENCYID);
```

Physical Model Schema Diagram



Development & Implementation Physical Schema Diagram



Database Development & Implementation Unit Testing

STORED PROCEDURES CREATED:

```
CREATE OR REPLACE PROCEDURE NEW_LICENSE (DL_NUMBER VARCHAR2, DL_EXPDATE DATE, DL_STATE CHAR)

IS

BEGIN

INSERT INTO DRIVERLICENSE(DRIVERLICENSENUMBER, DRIVERLICENSEEXPDATE, DRIVERLICENSESTATE)

VALUES (DL_NUMBER, DL_EXPDATE, DL_STATE);

END;


CREATE OR REPLACE PROCEDURE NEW_CUSTOMERUSERACCOUNT (CUA_USERNAME VARCHAR2,
CUA_PASSWORD VARCHAR2, CUA_EMAIL VARCHAR2)

IS

BEGIN

INSERT INTO CUSTOMERUSERACCOUNT (USERNAME, PASSWORD, EMAIL)

VALUES (CUA_USERNAME, CUA_PASSWORD, CUA_EMAIL);

END;


CREATE OR REPLACE PROCEDURE SIGN_UP_CUSTOMER

(C_FIRSTNAME VARCHAR2,C_LASTNAME VARCHAR2,C_BIRTHDATE DATE,C_ADDRESSLINE1 VARCHAR2, C_CITY
VARCHAR2, C_STATECODE CHAR, C_ZIPCODE VARCHAR2, C_COUNTRY VARCHAR2, C_PHONE VARCHAR2, C_EMAIL
VARCHAR2, C_DRIVERLICENSENUMBER VARCHAR2, C_CUSTOMERACCOUNTID RAW, C_CUSTOMERTYPE CHAR)

IS

BEGIN

INSERT INTO CUSTOMER (FIRSTNAME, LASTNAME, BIRTHDATE, ADDRESSLINE1,CITY, STATECODE, ZIPCODE,
COUNTRY, PHONE, EMAIL, DRIVERLICENSENUMBER, CUSTOMERUSERACCOUNTID, CUSTOMERTYPE)

VALUES (C_FIRSTNAME, C_LASTNAME, C_BIRTHDATE, C_ADDRESSLINE1, C_CITY, C_STATECODE, C_ZIPCODE,
C_COUNTRY, C_PHONE, C_EMAIL, C_DRIVERLICENSENUMBER, C_CUSTOMERACCOUNTID, C_CUSTOMERTYPE);

END;


CREATE OR REPLACE PROCEDURE NEW_DISCOUNT (ND_CODE CHAR, ND_DESCRIPTION VARCHAR2)

IS

BEGIN

INSERT INTO DISCOUNT(DISCOUNTCODE, DISCOUNTCODEDESC) VALUES
```

```
(ND_CODE, ND_DESCRIPTION);

END;


CREATE OR REPLACE PROCEDURE NEW_RETAILCUST(R_CUSTOMERID NUMBER, R_DISCOUNTID NUMBER,
R_EZPLUSID NUMBER)

IS

BEGIN

INSERT INTO RETAILCUSTOMER (CUSTOMERID, DISCOUNTID, EZPLUSID)

VALUES (R_CUSTOMERID, R_DISCOUNTID, R_EZPLUSID);

END;


CREATE OR REPLACE PROCEDURE REGISTER_COMPANY(P_COMPANYNAME VARCHAR2, P_ADDRESSLINE1
VARCHAR2, P_CITY VARCHAR2, P_STATECODE CHAR, P_ZIPCODE VARCHAR2, P_COUNTRY VARCHAR2,

P_COMPANYREPNAME VARCHAR2, P_CONTACTPHONE VARCHAR2, P_CONTACTEMAIL VARCHAR2,
P_COMPANYDISCOUNTPERCENTAGE NUMBER)

IS

BEGIN

INSERT INTO COMPANY (COMPANYNAME, ADDRESSLINE1, CITY, STATECODE, ZIPCODE, COUNTRY,
COMPANYREPNAME, CONTACTPHONE, CONTACTEMAIL, COMPANYDISCOUNTPERCENTAGE)

VALUES (P_COMPANYNAME, P_ADDRESSLINE1, P_CITY, P_STATECODE, P_ZIPCODE, P_COUNTRY,

P_COMPANYREPNAME , P_CONTACTPHONE , P_CONTACTEMAIL , P_COMPANYDISCOUNTPERCENTAGE);

END;


CREATE OR REPLACE PROCEDURE NEW_CORPCUST (C_CUSTOMERID NUMBER, C_COMPANYID NUMBER)

IS

BEGIN

INSERT INTO CORPORATECUSTOMER (CUSTOMERID, COMPANYID)

VALUES (C_CUSTOMERID, C_COMPANYID);

END;


CREATE OR REPLACE PROCEDURE NEW_CREDITCARD (N_CREDITCARDNUMBER VARCHAR2, N_OWNERNAME
VARCHAR2, N_MERCHANTNAME VARCHAR2, N_EXPDATE DATE, N_ADDRESSLINE1 VARCHAR2,

N_CITY VARCHAR2, N_STATECODE CHAR, N_ZIPCODE VARCHAR2, N_COUNTRY VARCHAR2, N_CREDITCARDLIMIT
NUMBER, N_CREDITCARDBALANCE NUMBER, N_ACTIVATIONSTATUS CHAR)
```

```
IS

BEGIN

INSERT INTO CREDITCARD (CREDITCARDNUMBER, OWNERNAME, MERCHANTNAME, EXPDATE, ADDRESSLINE1,
CITY, STATECODE, ZIPCODE, COUNTRY, CREDITCARDLIMIT, CREDITCARDBALANCE, ACTIVATIONSTATUS)

VALUES (N_CREDITCARDNUMBER , N_OWNERNAME , N_MERCHANTNAME , N_EXPDATE ,
N_ADDRESSLINE1 ,N_CITY , N_STATECODE, N_ZIPCODE, N_COUNTRY , N_CREDITCARDLIMIT ,
N_CREDITCARDBALANCE , N_ACTIVATIONSTATUS);

END;


CREATE OR REPLACE PROCEDURE REGISTER_CUSTOMERCARD (R_CARDNUMBER VARCHAR2, R_CUSTOMERID
NUMBER)

IS

BEGIN

INSERT INTO CUSTOMERCREDITCARD(CARDNUMBER, CUSTOMERID) VALUES (R_CARDNUMBER, R_CUSTOMERID);

END;


CREATE OR REPLACE PROCEDURE NEW_RENTALAGENCY(C_RENTALAGENCYID NUMBER, C_RENTALAGENCYNAME
VARCHAR2, C_ADDRESSLINE1 VARCHAR2, C_CITY VARCHAR2, C_STATECODE CHAR, C_COUNTRY VARCHAR2,
C_ZIPCODE VARCHAR2, C_PHONE VARCHAR2, C_EMAIL VARCHAR2)

IS

BEGIN

INSERT INTO RENTALAGENCY (RENTALAGENCYID, RENTALAGENCYNAME, ADDRESSLINE1, CITY, STATECODE,
COUNTRY, ZIPCODE, PHONE, EMAIL)

VALUES (C_RENTALAGENCYID, C_RENTALAGENCYNAME, C_ADDRESSLINE1, C_CITY, C_STATECODE, C_COUNTRY,
C_ZIPCODE, C_PHONE, C_EMAIL);

END;


CREATE OR REPLACE PROCEDURE NEW_VEHICLE (N_VINNUMBER VARCHAR2, N_MAKE VARCHAR2, N_MODEL
VARCHAR2, N_YEAR NUMBER, N_COLOR VARCHAR2, N_LICENSEPLATENUMBER VARCHAR2, N_MILEAGE NUMBER,
N_TRANSMISSIONTYPE VARCHAR2,

N_SEATCAPACITY NUMBER, N_VEHICLERENTALCATEGORYID NUMBER, N_VEHICLESTATUSID NUMBER,
N_VEHICLEOWNINGAGENCYID NUMBER, N_VEHCURRENTLOCAGENCYID NUMBER, N_VEHICLETYPE CHAR)

IS

BEGIN

INSERT INTO VEHICLE (VINNUMBER, MAKE, MODEL, YEAR, COLOR, LICENSEPLATENUMBER , MILEAGE,
TRANSMISSIONTYPE ,
```

```
SEATCAPACITY ,
VEHICLERENTALCATEGORYID,VEHICLESTATUSID ,VEHICLEOWNINGAGENCYID ,VEHICLECURRENTLOCATIONAGENCY
ID , VEHICLETYPE)

VALUES (N_VINNUMBER , N_MAKE , N_MODEL , N_YEAR , N_COLOR , N_LICENSEPLATENUMBER , N_MILEAGE ,
N_TRANSMISSIONTYPE ,

N_SEATCAPACITY , N_VEHICLERENTALCATEGORYID , N_VEHICLESTATUSID , N_VEHICLEOWNINGAGENCYID ,
N_VEHCURRENTLOCAGENCYID , N_VEHICLETYPE);

END;


CREATE OR REPLACE PROCEDURE NEW_CAR (NC_VEHICLE NUMBER, NC_TRUNKCAPACITY NUMBER)

IS

BEGIN

INSERT INTO CAR(VEHICLEID, TRUNKCAPACITY) VALUES (NC_VEHICLE, NC_TRUNKCAPACITY);

END;


CREATE OR REPLACE PROCEDURE NEW_MINIVAN (NM_VEHICLEID NUMBER, NM_HASDISABILLITYPACKAGE CHAR)

IS

BEGIN

INSERT INTO MINIVAN (VEHICLEID, HASDISABILLITYPACKAGE) VALUES (NM_VEHICLEID,
NM_HASDISABILLITYPACKAGE);

END;


CREATE OR REPLACE PROCEDURE NEW_SUV (NS_VEHICLEID NUMBER, NS_TOWINGCAPACITY NUMBER, NS_ISAWD
CHAR)

IS

BEGIN

INSERT INTO SUV (VEHICLEID, TOWINGCAPACITY, ISAWD) VALUES (NS_VEHICLEID, NS_TOWINGCAPACITY,
NS_ISAWD);

END;


CREATE OR REPLACE PROCEDURE NEW_CARGOVAN (NC_VEHICLEID NUMBER, NC_CARGOCAPACITY NUMBER,
NC_MAXIMUMPAYLOAD NUMBER)

IS

BEGIN

INSERT INTO CARGOVAN (VEHICLEID, CARGOCAPACITY, MAXIMUMPAYLOAD) VALUES (NC_VEHICLEID,
NC_CARGOCAPACITY, NC_MAXIMUMPAYLOAD);
```

```
END;


CREATE OR REPLACE PROCEDURE MAKE_RESERVATION (M_RESERVATIONID NUMBER,M_CUSTOMERID NUMBER,
M_RENTALAGENCYID NUMBER, M_VEHICLERENTALCATEGORYID NUMBER, M_RESERVATIONDROPOFFAGENCYID
NUMBER, M_RESERVATIONPICKUPDATE DATE,

M_RESERVATIONPICKUPTIME NUMBER, M_RESERVATIONDROPOFFDATE DATE, M_RESERVATIONDROPOFFTIME
NUMBER, M_RESERVATIONSTATUSID NUMBER)

IS

BEGIN

INSERT INTO RESERVATION (RESERVATIONID, CUSTOMERID, RENTALAGENCYID, VEHICLERENTALCATEGORYID,
RESERVATIONDROPOFFAGENCYID, RESERVATIONPICKUPDATE, RESERVATIONPICKUPTIME,
RESERVATIONDROPOFFDATE, RESERVATIONDROPOFFTIME, RESERVATIONSTATUSID)

VALUES (M_RESERVATIONID, M_CUSTOMERID , M_RENTALAGENCYID , M_VEHICLERENTALCATEGORYID ,
M_RESERVATIONDROPOFFAGENCYID , M_RESERVATIONPICKUPDATE, M_RESERVATIONPICKUPTIME,
M_RESERVATIONDROPOFFDATE , M_RESERVATIONDROPOFFTIME , M_RESERVATIONSTATUSID);

END;


CREATE OR REPLACE PROCEDURE NEW_RENTAL (NR_CUSTOMERID NUMBER, NR_VEHICLEID NUMBER,
NR_RENTALAGENCYID NUMBER, NR_RENTALDROPOFFAGENCYID NUMBER,

NR_RESERVATIONID NUMBER, NR_RENTALPICKUPDATE DATE, NR_RENTALPICKUPTIME NUMBER,
NR_RENTALDROPOFFDATE DATE, NR_RENTALDROPOFFTIME NUMBER,

NR_RENTALPICKUPODOMETERVALUE NUMBER, NR_RENTALDROPOFFODOMETERVALUE
NUMBER,NR_RENTALFUELOPTIONID NUMBER, NR_INSURANCEOPTIONID NUMBER,

NR_RENTALSTATUSID NUMBER, NR_RENTALDEPOSIT NUMBER)

IS

BEGIN

INSERT INTO RENTAL (CUSTOMERID, VEHICLEID, RENTALAGENCYID, RENTALDROPOFFAGENCYID, RESERVATIONID,
RENTALPICKUPDATE, RENTALPICKUPTIME, RENTALDROPOFFDATE,

RENTALDROPOFFTIME, RENTALPICKUPODOMETERVALUE, RENTALDROPOFFODOMETERVALUE,
RENTALFUELOPTIONID, INSURANCEOPTIONID, RENTALSTATUSID, RENTALDEPOSIT)

VALUES (NR_CUSTOMERID, NR_VEHICLEID, NR_RENTALAGENCYID,
NR_RENTALDROPOFFAGENCYID,NR_RESERVATIONID, NR_RENTALPICKUPDATE, NR_RENTALPICKUPTIME,
NR_RENTALDROPOFFDATE,

NR_RENTALDROPOFFTIME,NR_RENTALPICKUPODOMETERVALUE,
NR_RENTALDROPOFFODOMETERVALUE,NR_RENTALFUELOPTIONID, NR_INSURANCEOPTIONID,
NR_RENTALSTATUSID, NR_RENTALDEPOSIT);

END;
```

```
CREATE OR REPLACE PROCEDURE NEW_EMPLOYEEUSERACC (NE_USERNAME VARCHAR2, NE_PASSWORD
VARCHAR2, NE_EMAIL VARCHAR2)

IS

BEGIN

INSERT INTO EMPLOYEEUSERACCOUNT (USERNAME, PASSWORD, EMAIL)

VALUES (NE_USERNAME, NE_PASSWORD, NE_EMAIL);

END;


CREATE OR REPLACE PROCEDURE NEW_EMPLOYEE (N_EMPLOYEEID NUMBER, N_FIRSTNAME VARCHAR2,
N_LASTNAME VARCHAR2, N_SSNUMBER CHAR, N_BIRTHDATE DATE, N_ADDRESSLINE1 VARCHAR2, N_CITY
VARCHAR2, N_STATECODE CHAR,

N_ZIPCODE VARCHAR2, N_COUNTRY VARCHAR2, N_EMPLOYEEPHONE VARCHAR2, N_EMPLOYEEJOBTITLE
VARCHAR2, N_EMPLOYEEEMAIL VARCHAR2, N_EMPLOYEEUSERACCOUNTID RAW)

IS

BEGIN

INSERT INTO EMPLOYEE (EMPLOYEEID, FIRSTNAME, LASTNAME ,SSNUMBER, BIRTHDATE, ADDRESSLINE1 , CITY,
STATECODE,ZIPCODE, COUNTRY, EMPLOYEEPHONE, EMPLOYEEJOBTITLE, EMPLOYEEEMAIL,

EMPLOYEEUSERACCOUNTID)

VALUES (N_EMPLOYEEID, N_FIRSTNAME , N_LASTNAME , N_SSNUMBER , N_BIRTHDATE , N_ADDRESSLINE1 ,
N_CITY , N_STATECODE ,

N_ZIPCODE , N_COUNTRY , N_EMPLOYEEPHONE , N_EMPLOYEEJOBTITLE , N_EMPLOYEEEMAIL,
N_EMPLOYEEUSERACCOUNTID );

END;


CREATE OR REPLACE PROCEDURE NEW_TRANSPORT (NT_EMPLOYEEID NUMBER, NT_VEHICLEID NUMBER,
NT_PICKUPRENTALAGENCYID NUMBER, NT_DROPOFFRENTALAGENCYID NUMBER,

NT_DRIVERDEPARTUREDATE DATE, NT_DRIVERDEPARTURETIME NUMBER, NT_VEHICLEPICKUPDATE DATE,
NT_VEHICLEPICKUPTIME NUMBER,

NT_VEHICLEDROPOFFDATE DATE, NT_VEHICLEDROPOFFTIME NUMBER, NT_TRANSPORTSTATUSID NUMBER,
NT_TRANSPORTREASONID NUMBER)

IS

BEGIN

INSERT INTO TRANSPORT (EMPLOYEEID, VEHICLEID, PICKUPRENTALAGENCYID, DROPOFFRENTALAGENCYID,
DRIVERDEPARTUREDATE,DRIVERDEPARTURETIME,

VEHICLEPICKUPDATE, VEHICLEPICKUPTIME,VEHICLEDROPOFFDATE, VEHICLEDROPOFFTIME,TRANSPORTSTATUSID ,
TRANSPORTREASONID )
```

```
VALUES (NT_EMPLOYEEID, NT_VEHICLEID , NT_PICKUPRENTALAGENCYID ,
NT_DROPOFFRENTALAGENCYID,NT_DRIVERDEPARTUREDATE , NT_DRIVERDEPARTURETIME,

NT_VEHICLEPICKUPDATE , NT_VEHICLEPICKUPTIME,NT_VEHICLEDROPOFFDATE , NT_VEHICLEDROPOFFTIME ,
NT_TRANSPORTSTATUSID , NT_TRANSPORTREASONID);

END;
```

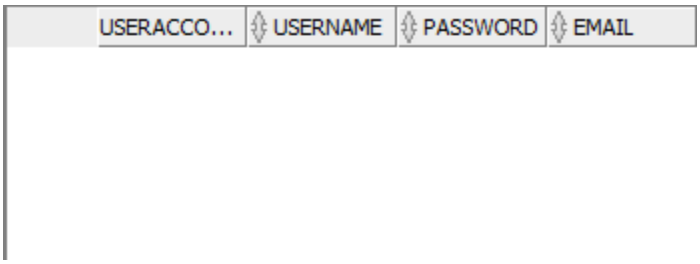
Insert Statements: Prior to populating tables

Using the Stored Procedure SIGN_UP_CUSTOMER that I've created, I was able to efficiently populate the CUSTOMER table

SELECT * FROM CUSTOMER;

Using the Stored Procedure NEW_CUSTOMERUSERACCOUNT that I've created, I was able to efficiently populate the CUSTOMERUSERACCOUNT table

SELECT * FROM CUSTOMERUSERACCOUNT;

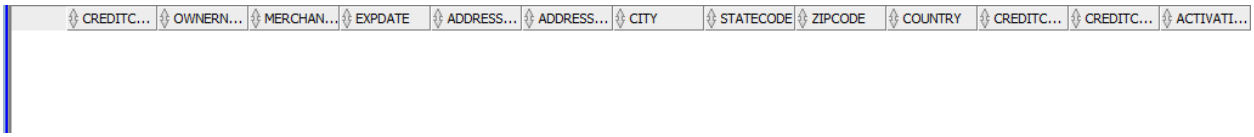


Using the Stored Procedure NEW_LICENSE that I've created, I was able to efficiently populate the DRIVERLICENSE table

SELECT * FROM DRIVERLICENSE;

Using the Stored Procedure NEW_CREDITCARD that I've created, I was able to efficiently populate the CREDITCARD table

SELECT * FROM CREDITCARD;

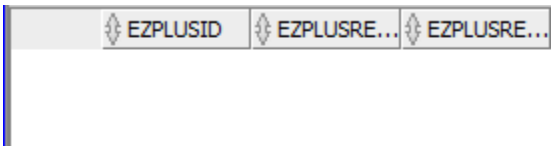


Using the Stored Procedure REGISTER_CUSTOMERCARD that I've created, I was able to efficiently populate the CUSTOMERCREDITCARD table

SELECT * FROM CUSTOMERCREDITCARD;

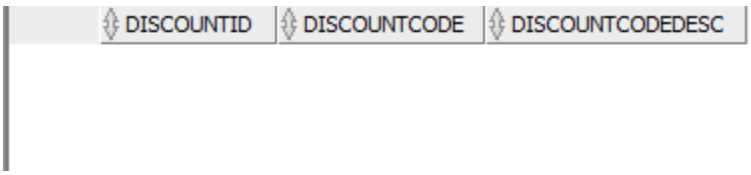
I was able to populate the EZPLUS table using an insert statement

SELECT * FROM EZPLUS;



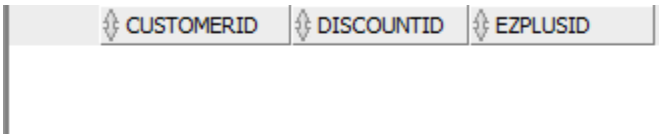
Using the Stored Procedure NEW_DISCOUNT that I've created, I was able to efficiently populate the DISCOUNT table

SELECT * FROM DISCOUNT;



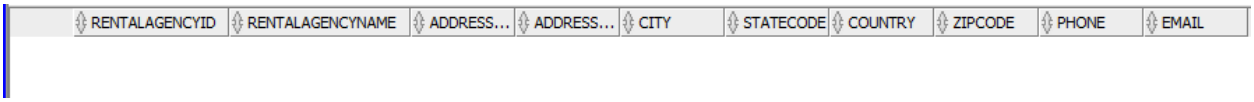
Using the Stored Procedure NEW_RETAILCUST that I've created, I was able to efficiently populate the RETAILCUSTOMER table

SELECT * FROM RETAILCUSTOMER;



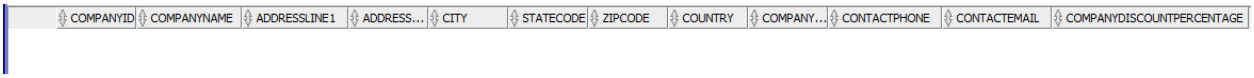
Using the Stored Procedure CREATE_RENTALAGENCY that I've created, I was able to efficiently populate the RENTALAGENCY table

SELECT * FROM RENTALAGENCY;



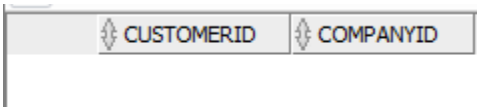
Using the Stored Procedure REGISTER_COMPANY that I've created, I was able to efficiently populate the COMPANY table

SELECT * FROM COMPANY;



Using the Stored Procedure NEW_CORPCUST that I've created, I was able to efficiently populate the CORPORATECUSTOMER table

SELECT * FROM CORPORATECUSTOMER;

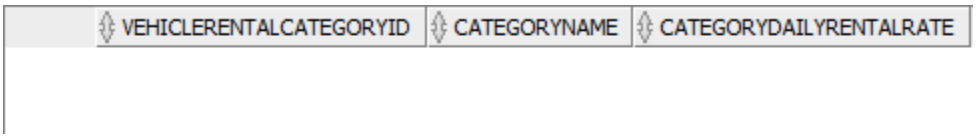


I was able to populate the VEHICLESTATUS table using an insert statement. I felt as if this table did not require the use of stored procedures as the amount of vehicle statuses would be a set amount and not in surplus deeming it unnecessary.

SELECT * FROM VEHICLESTATUS;

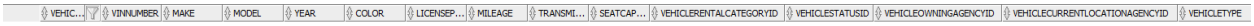
I was able to populate the VEHICLERENTALCATEGORY table using an insert statement. I felt as if this table did not require the use of stored procedures as the amount of vehicle categories would be a set amount and not needed surplus deeming it unnecessary.

SELECT * FROM VEHICLERENTALCATEGORY;



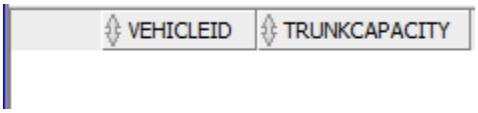
Using the Stored Procedure NEW_VEHICLE that I've created, I was able to efficiently populate the VEHICLE table

SELECT * FROM VEHICLE;



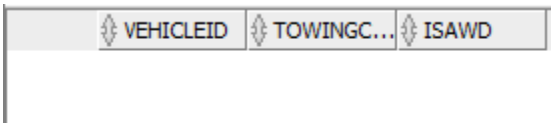
Using the Stored Procedure NEW_CAR that I've created, I was able to efficiently populate the CAR table

SELECT * FROM CAR;



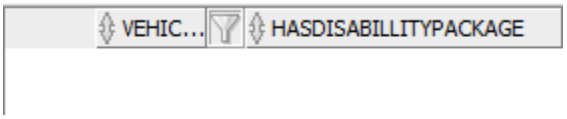
Using the Stored Procedure NEW_SUV that I've created, I was able to efficiently populate the SUV table

SELECT * FROM SUV;



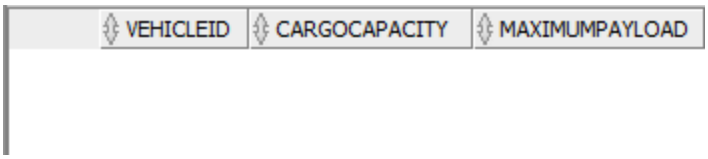
Using the Stored Procedure NEW_MINIVAN that I've created, I was able to efficiently populate the MINIVAN table

SELECT * FROM MINIVAN;



Using the Stored Procedure NEW_CARGOVAN that I've created, I was able to efficiently populate the CARGOVAN table

SELECT * FROM CARGOVAN;



I was able to populate the RESERVATIONSTATUS table using an insert statement. I felt as if this table did not require the use of stored procedures as the amount of RESERVATION STATUSES would be a set amount and not in surplus deeming it unnecessary.

SELECT * FROM RESERVATIONSTATUS;

I was able to populate the RENTALSTATUS table using an insert statement. I felt as if this table did not require the use of stored procedures as the amount of RENTAL STATUSES would be a set amount and not in surplus deeming it unnecessary.

SELECT * FROM RENTALSTATUS

I was able to populate the RENTALINSURANCEOPTION table using an insert statement. I felt as if this table did not require the use of stored procedures as the amount of RENTAL INSURANCE OPTIONS would be a set amount and not in surplus deeming it unnecessary.

SELECT * FROM RENTALINSURANCEOPTION;

I was able to populate the FUELOPTION table using an insert statement. I felt as if this table did not require the use of stored procedures as the amount of FUEL OPTIONS would be a set amount of 2 so I deemed it unnecessary.

SELECT * FROM FUELOPTION;

Using the Stored Procedure MAKE_RESERVATION that I've created, I was able to efficiently populate the RESERVATION table

SELECT * FROM RESERVATION;

Using the Stored Procedure NEW_RENTAL that I've created, I was able to efficiently populate the RENTAL table

SELECT * FROM RENTAL;

Using the Stored Procedure NEW_EMPLOYEEUSERACC that I've created, I was able to efficiently populate the EMPLOYEEUSERACCOUNT table

SELECT * FROM EMPLOYEEUSERACCOUNT;

Using the Stored Procedure NEW_EMPLOYEE that I've created, I was able to efficiently populate the EMPLOYEE table

SELECT * FROM EMPLOYEE;

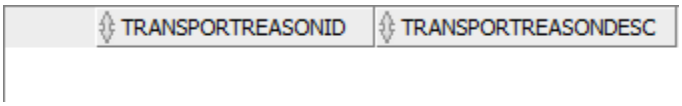
I was able to populate the TRANSPORTSTATUS table using an insert statement. I felt as if this table did not require the use of stored procedures as the amount of TRANSPORT STATUSES would be a set amount, so I deemed it unnecessary.

SELECT * FROM TRANSPORTSTATUS;

TRANSPORTSTATUSID	TRANSPORTSTATUSDESC

I was able to populate the TRANSPORTREASON table using an insert statement. I felt as if this table did not require the use of stored procedures as the amount of TRANSPORT REASONS would be a set amount, so I deemed it unnecessary.

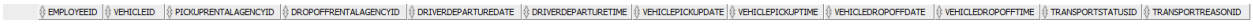
SELECT * FROM TRANSPORTREASON;



TRANSPORTREASONID	TRANSPORTREASONDESC

Using the Stored Procedure NEW_TRANSPORT that I've created, I was able to efficiently populate the TRANSPORT table

SELECT * FROM TRANSPORT;



EMPLOYEEID	VEHICLEID	PICKUPRENTALAGENCYID	DROPOFFRENTALAGENCYID	DRIVERDEPARTUREDATE	DRIVERDEPARTURETIME	VEHICLEPICKUPDATE	VEHICLEPICKUPTIME	VEHICLEDROPOFFDATE	VEHICLEDROPOFFTIME	TRANSPORTSTATUSID	TRANSPORTREASONID

Insert Statements: After populating tables

CUSTOMER

INSERT INTO CUSTOMER (FIRSTNAME, LASTNAME, BIRTHDATE, ADDRESSLINE1, CITY, STATECODE, ZIPCODE, COUNTRY, PHONE, EMAIL, DRIVERLICENSENUMBER, CUSTOMERUSERACCOUNTID, CUSTOMERTYPE)

VALUES ('Tyler', 'Walker', '25-SEP-2000', 'Example Address1234', 'Brooklyn', 'NY', '11236', 'United States', '9292002258', 'walktyler25@gmail.com', '12345678910987', '4216CD1D7E494698B9BFA9A755E12BB1', 'R');

INSERT INTO CUSTOMER (FIRSTNAME, LASTNAME, BIRTHDATE, ADDRESSLINE1, CITY, STATECODE, ZIPCODE, COUNTRY, PHONE, EMAIL, DRIVERLICENSENUMBER, CUSTOMERUSERACCOUNTID, CUSTOMERTYPE)

VALUES ('Jacob', 'Howards', '09-JAN-2006', 'Another Example Address1234', 'Brooklyn', 'NY', '11208', 'United States', '6782002258', 'jacobhow@gmail.com', '51345678910987', 'E2C43E6BFBCD40439EF8AD3E4A34EE66', 'c');

INSERT INTO CUSTOMER (FIRSTNAME, LASTNAME, BIRTHDATE, ADDRESSLINE1, CITY, STATECODE, ZIPCODE, COUNTRY, PHONE, EMAIL, DRIVERLICENSENUMBER, CUSTOMERUSERACCOUNTID, CUSTOMERTYPE)

VALUES ('Alex', 'Rodington', '01-OCT-1996', 'Another Example Address2468', 'Queens', 'NY', '11289', 'United States', '7182002258', 'alexrod5@gmail.com', '55555678910987', '1117A59A3D894B77A920FC12991ACE43', 'r');

CALL SIGN_UP_CUSTOMER('JOHNNY', 'CHAMBELL', '01-JAN-02', 'SOME RANDOM ADDRESS 123', 'ALBANY', 'NY', '12084', 'United States', '8888887777', 'JohnBell02@yahoo.com', '43215678910987', '12FAF150EC5644BA8F34F0453E85A0D3', 'C');

CALL SIGN_UP_CUSTOMER('John', 'Shall', '01-MAR-1996', 'SOMEADDRESS SOMEWHERE', 'BRONX', 'NY', '10451', 'UNITED STATES', '1112223333', 'JOHNSHALL55@gmail.com', '888517921', '4C5418C967D04304AA0AD021BEE3EC0B', 'R');

CALL SIGN_UP_CUSTOMER('JAKE', 'NULL', '07-MAR-1999', 'SOMEADDRESS SOMEWHERE', 'BROOKYLN', 'NY', '11204', 'UNITED STATES', '1113334444', 'NULLJAKE@mail.com', '888557622', '023B999EB44940709200AFC1A29A0698', 'R');

CALL SIGN_UP_CUSTOMER('MACK', 'QUEEN', '23-NOV-2002', 'SOMEADDRESS SOMEWHERE', 'MANHATTAN', 'NY', '10013', 'UNITED STATES', '1114445555', 'MAQUEENSTEIN@outlook.com', '888547523', '680A4357DAE5461D95334A5E5B8D589B', 'C');

CALL SIGN_UP_CUSTOMER('MALOTTO', 'ARASHIO', '25-JAN-1998', 'SOMEADDRESS SOMEWHERE', 'MANHATTAN', 'NY', '10012', 'UNITED STATES', '1115556666', 'MALOTTO112@gmail.com', '888527424', 'B3EC0D8D9E774C268CEA60B9101ADD72', 'C');

CALL SIGN_UP_CUSTOMER('KEKE', 'WILLKINS', '05-SEP-1997', 'SOMEADDRESS SOMEWHERE', 'BRONX', 'NY', '10452', 'UNITED STATES', '1112223333', 'KEEEK921@gmail.com', '888517125', 'B64D04DA20DB4275BBF34FBA54549745', 'C');

SELECT * FROM CUSTOMER;

CUSTOMERID	FIRSTNAME	LASTNAME	BIRTHDATE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY	PHONE	EMAIL	DRIVERLICENSENUMBER	CUSTOMERUSERACCOUNTID	CUSTOMERTYPE
1	TYLER	Walker	25-SEP-00	Example Address1234	(null)	Brooklyn	NY	11236	United States	9292002258	walktyler25@gmail.com	12345678910987	4216CD1D7E494698B9BFA9A755E12BB1	R
2	Jacob	Howards	09-JAN-06	Another Example Address1234	(null)	Brooklyn	NY	11208	United States	6782002258	jacobhow@gmail.com	51345678910987	E2C43E6BFBCD40439EF8AD3E4A34EE66	C
3	Alex	Rodington	01-OCT-96	Another Example Address2468	(null)	Queens	NY	11289	United States	7182002258	alexrod5@gmail.com	55555678910987	1117A59A3D894B77A920FC12991ACE43	R
4	JOHNNY	CHAMBELL	01-JAN-02	SOME RANDOM ADDRESS 123	(null)	ALBANY	NY	12084	United States	8888887777	JohnBell02@yahoo.com	43215678910987	12FAF150EC5644BA8F34F0453E85A0D3	C
5	KEKE	WILLKINS	05-SEP-97	SOMEADDRESS SOMEWHERE	(null)	BRONX	NY	10452	UNITED STATES	1112223333	KEEEK921@gmail.com	888517125	B64D04DA20DB4275BBF34FBA54549745	C
6	MALOTTO	ARASHIO	25-JAN-98	SOMEADDRESS SOMEWHERE	(null)	MANHATTAN	NY	10012	UNITED STATES	1115556666	MALOTTO112@gmail.com	888527424	B3EC0D8D9E774C268CEA60B9101ADD72	C
7	John	Shall	01-MAR-96	SOMEADDRESS SOMEWHERE	(null)	BRONX	NY	10451	UNITED STATES	1112223333	JOHNSHALL55@gmail.com	888517921	7B954DA80DC541A39151FDA7FFAE2BAD	R
8	JAKE	NULL	07-MAR-99	SOMEADDRESS SOMEWHERE	(null)	BROOKYLN	NY	11204	UNITED STATES	1113334444	NULLJAKE@mail.com	888557622	023B999EB44940709200AFC1A29A0698	R
9	MACK	QUEEN	23-NOV-02	SOMEADDRESS SOMEWHERE	(null)	MANHATTAN	NY	10013	UNITED STATES	1114445555	MAQUEENSTEIN@outlook.com	888547523	680A4357DAE5461D95334A5E5B8D589B	C

CUSTOMERUSERACCOUNT

INSERT ALL

INTO CUSTOMERUSERACCOUNT (USERNAME, PASSWORD,EMAIL) VALUES ('Walktyler', 'password1', 'walktyler25@gmail.com')

INTO CUSTOMERUSERACCOUNT (USERNAME, PASSWORD,EMAIL) VALUES ('JacobHOW', 'password2', 'jacobhow@gmail.com')

INTO CUSTOMERUSERACCOUNT (USERNAME, PASSWORD,EMAIL) VALUES ('ALEXROD', 'password3', 'alexrod5@gmail.com') SELECT 1 FROM DUAL;

CALL NEW_CUSTOMERUSERACCOUNT ('JOHNSHALL', 'randompassword142', 'JOHNSHALL55@gmail.com');

CALL NEW_CUSTOMERUSERACCOUNT ('JAKENULL00', 'randompassword132', 'NULLJAKE@mail.com');

CALL NEW_CUSTOMERUSERACCOUNT ('MAQUEENSTEIN', 'randompass112', 'MAQUEENSTEIN@outlook.com');

CALL NEW_CUSTOMERUSERACCOUNT ('MALOTTO112', 'randompassword102', 'MALOTTO112@gmail.com');

CALL NEW_CUSTOMERUSERACCOUNT ('KEEEKEEEK', 'randompassword199', 'KEEEK921@gmail.com');

SELECT * FROM CUSTOMERUSERACCOUNT;

USERACCOUNTID	USERNAME	PASSWORD	EMAIL
12FAF150EC5644BA8F34F0453E85A0D3	JohnBall	password02	JohnBall02@yahoo.com
4216CD1D7E494698B9BFA9A755E12BB1	Walktyler	password1	walktyler25@gmail.com
E2C43E6BFBCD40439EF8AD3E4A34EE66	JacobHOW	password2	jacobhow@gmail.com
1117A59A3D894B77A920FC12991ACE43	ALEXROD	password3	alexrod5@gmail.com
DF1B4B3F365D4B168B4B0682558929A6	DallardGary	password04	GaryDallard@aol.com
7B954DA80DC541A39151FDA7FFAE2BAD	JOHNSHALL	randompassword142	JOHNSHALL55@gmail.com
023B999EB44940709200AFC1A29A0698	JAKENULL00	randompassword132	NULLJAKE@mail.com
680A4357DAE5461D95334A5E5B8D589B	MAQUEENSTEIN	randompass112	MAQUEENSTEIN@outlook.com
B3EC0D8D9E774C268CEA60B9101ADD72	MALOTTO112	randompassword102	MALOTTO112@gmail.com
B64D04DA20DB4275BBF34FBA54549745	KEEEKEEEK	randompassword199	KEEEK921@gmail.com

DRIVERLICENSE

INSERT INTO DRIVERLICENSE VALUES ('12345678910987', '09-SEP-2030', 'NY');

INSERT INTO DRIVERLICENSE VALUES ('51345678910987', '05-FEB-2031', 'NY');

INSERT INTO DRIVERLICENSE VALUES ('55555678910987', '09-APR-2026', 'FL');

CALL NEW_LICENSE('888517921', '01-MAR-2026' ,'NY');

CALL NEW_LICENSE('888557622', '18-JUL-2029' ,'NY');

CALL NEW_LICENSE('888547523', '22-MAR-2030' ,'NY');

CALL NEW_LICENSE('888527424', '10-JAN-2022' ,'NY');

CALL NEW_LICENSE('888517125', '25-DEC-2021' ,'NY');

SELECT * FROM DRIVERLICENSE;

DRIVERLICENSENUMBER	DRIVERLICENSEEXPDATE	DRIVERLICENSESTATE
12345678910987	09-SEP-30	NY
51345678910987	05-FEB-31	NY
55555678910987	09-APR-26	FL
43215678910987	02-APR-32	NY
888517921	01-MAR-26	NY
888557622	18-JUL-29	NY
888547523	22-MAR-30	NY
888527424	10-JAN-22	NY
888517125	25-DEC-21	NY

CREDITCARD

CALL NEW_CREDITCARD('4024007135713930', 'MACK QUEEN', 'VISA', '07-FEB-2027', 'SOMEADDRESS SOMEWHERE', 'MANHATTAN', 'NY', '10013','UNITED STATES', 10000, 2000, '1');

CALL NEW_CREDITCARD('6011368248848918', 'MALOTTO ARASHIO', 'DISCOVER', '13-DEC-2031', 'SOMEADDRESS SOMEWHERE', 'MANHATTAN', 'NY', '10012','UNITED STATES', 30000, 18600, '1');

CALL NEW_CREDITCARD('373102971940153', 'KEKE WILLKINS', 'AMERICAN EXPRESS', '16-NOV-2022', 'SOMEADDRESS SOMEWHERE', 'MANHATTAN', 'NY', '10013','UNITED STATES', 20000, 11000, '1');

CALL NEW_CREDITCARD('345953192054975', 'JAKE NULL', 'AMERICAN EXPRESS', '01-JAN-2025', 'SOMEADDRESS SOMEWHERE', 'BROOKLYN', 'NY', '11204','UNITED STATES', 13000, 4500, '1');

CALL NEW_CREDITCARD('2221003690226132', 'JOHN SHALL', 'MASTERCARD', '27-MAR-2023', 'SOMEADDRESS SOMEWHERE', 'BRONX', 'NY', '10451','UNITED STATES', 8000, 5000, '1');

SELECT * FROM CREDITCARD;

CREDITCARDNUMBER	OWNERNAME	MERCHANTNAME	EXPDATE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY	CREDITCARDLIMIT	CREDITCARDBALANCE	ACTIVATIONSTATUS
4024007135713930	MACK QUEEN	VISA	07-FEB-27	SOMEADDRESS	SOMEWHERE	(null)	MANHATTAN	NY	10013	UNITED STATES	10000	2000	1
6011368248848918	MALOTTO ARASHIO	DISCOVER	13-DEC-31	SOMEADDRESS	SOMEWHERE	(null)	MANHATTAN	NY	10012	UNITED STATES	30000	18600	1
373102971940153	KEKE WILLKINS	AMERICAN EXPRESS	16-NOV-22	SOMEADDRESS	SOMEWHERE	(null)	MANHATTAN	NY	10013	UNITED STATES	20000	11000	1
345953192054975	JAKE NULL	AMERICAN EXPRESS	01-JAN-25	SOMEADDRESS	SOMEWHERE	(null)	BROOKLYN	NY	11204	UNITED STATES	13000	4500	1
2221003690226132	JOHN SHALL	MASTERCARD	27-MAR-23	SOMEADDRESS	SOMEWHERE	(null)	BRONX	NY	10451	UNITED STATES	8000	5000	1

CUSTOMERCREDITCARD

CALL REGISTER_CUSTOMERCARD('4024007135713930', 9);

CALL REGISTER_CUSTOMERCARD('6011368248848918', 6);

CALL REGISTER_CUSTOMERCARD('373102971940153', 5);

CALL REGISTER_CUSTOMERCARD('345953192054975', 8);

CALL REGISTER_CUSTOMERCARD('2221003690226132', 7);

SELECT * FROM CUSTOMERCREDITCARD;

EZRENTAL POS

CARDNUMBER	CUSTOMERID
2221003690226132	7
345953192054975	8
373102971940153	5
4024007135713930	9
6011368248848918	6

EZPLUS

INSERT ALL

INTO EZPLUS VALUES(1,9009854637,10000)

INTO EZPLUS VALUES(2,1000192461,500)

INTO EZPLUS VALUES(3,6493238865,159000)

INTO EZPLUS VALUES(4,2005135627,23000)

SELECT 1 FROM DUAL;

SELECT * FROM EZPLUS;

EZPLUSID	EZPLUSREWARDSCODE	EZPLUSREWARDSEARNEDPOINTS
3	EZP1244557898	12000
4	EZP1224667898	20000
1	EZP1234567899	10000
2	EZP1234567898	10000

DISCOUNT

CALL NEW_DISCOUNT('AAA9970054', 'AAA Membership Discount - 25% off base rate plus 10% donated for breast cancer research.');

CALL NEW_DISCOUNT('GOV8756921', 'Government Employee Discount - 30% off base rate');

CALL NEW_DISCOUNT('STA3415632', 'State Employee Discount for 25% off base rate');

CALL NEW_DISCOUNT('VET2055179', 'Veteran Discount 35% off base rate Plus 10% donation to veteran’s family fund.');

SELECT * FROM DISCOUNT;

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

RETAILCUSTOMER

CALL NEW_RETAILCUST('1', 4, 1);

CALL NEW_RETAILCUST('3', 2, 2);

CALL NEW_RETAILCUST('7', 3, 3);

CALL NEW_RETAILCUST('8', 1, 4);

SELECT * FROM RETAILCUSTOMER;

CUSTOMERID	DISCOUNTID	EZPLUSID
1	4	1
3	2	2
7	3	3
8	1	4

RENTALAGENCY

CALL CREATE_RENTALAGENCY(1,'GARSHAWN CAR RENTALS', '123 DOVER AVE', 'BROOKLYN', 'NY', 'UNITED STATES',

'11236', '7182221356', 'GARSHAWNRENTALS@GMAIL.COM');

CALL CREATE_RENTALAGENCY(2,'UNIVERSAL RENTALS', '39 Harrison St', 'MANHATTAN', 'NY', 'UNITED STATES', '10013', '2126803009', 'UNVIERSALRENTALS@GMAIL.COM');

CALL CREATE_RENTALAGENCY(3,'FAST TRACK', '246 Mott St', 'MANHATTAN', 'NY', 'UNITED STATES', '10012', '2124651278', 'FASTTRACK@FASTTRACK.COM');

CALL CREATE_RENTALAGENCY(4,'ALPHA CAR COMPANY', '27110 Grand Central Pky', 'QUEENS', 'NY', 'UNITED STATES', '11005', '6466802109', 'ALPHARENTALS@AOL.COM');

CALL CREATE_RENTALAGENCY(5,'ZAWA CAR RENTALS', '2540 Shore Blvd', 'QUEENS', 'NY', 'UNITED STATES', '11102', '6466814135', 'ZAWACARS@GMAIL.COM');

SELECT * FROM RENTALAGENCY;

RENTALAGE...	RENTALAGENCYNAME	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	COUNTRY	ZIPCODE	PHONE	EMAIL
1	GARSHAWN CAR RENTALS	123 DOVER AVE	(null)	BROOKLYN	NY	UNITED STATES	11236	7182221356	GARSHAWNRENTALS@GMAIL.COM
2	UNIVERSAL RENTALS	39 Harrison St	(null)	MANHATTAN	NY	UNITED STATES	10013	2126803009	UNVIERSALRENTALS@GMAIL.COM
3	FAST TRACK	246 Mott St	(null)	MANHATTAN	NY	UNITED STATES	10012	2124651278	FASTTRACK@FASTTRACK.COM
4	ALPHA CAR COMPANY	27110 Grand Central Pky	(null)	QUEENS	NY	UNITED STATES	11005	6466802109	ALPHARENTALS@AOL.COM
5	ZAWA CAR RENTALS	2540 Shore Blvd	(null)	QUEENS	NY	UNITED STATES	11102	6466814135	ZAWACARS@GMAIL.COM

COMPANY

CALL REGISTER_COMPANY('The Helping Hands', 'RANDOM COMPANY ADDRESS', 'BRONX', 'NY', '10451', 'UNITED STATES', 'JOEL STANLEY', '8189002121', 'HELPINGHANDS@GMAIL.COM', .10);

CALL REGISTER_COMPANY('Alternate ', 'RANDOM COMPANY ADDRESS2', 'BROOKYLN', 'NY', '11206', 'UNITED STATES', 'MASON RUJIERD', '8189152011', 'THEALTERNATE@GMAIL.COM', .20);

CALL REGISTER_COMPANY('Sanguine Services', 'RANDOM COMPANY ADDRESS3', 'BROOKLYN', 'NY', '11208', 'UNITED STATES', 'RUDY SANGUINE', '8185911421', 'SANGUINESERVES@GMAIL.COM', .13);

CALL REGISTER_COMPANY('Snared Services', 'RANDOM COMPANY ADDRESS4', 'QUEENS', 'NY', '11005', 'UNITED STATES', 'ERIS SHERLEY', '8189648721', 'SNAREDSERVICES@GMAIL.COM', .30);

EZRENTAL POS

CALL REGISTER_COMPANY('Random Repairs', 'RANDOM COMPANY ADDRESS5', 'MANHATTAN', 'NY', '10013', 'UNITED STATES', 'ROXY MIGURDIA', '8189120021', 'THERANDOMREPAIRS@GMAIL.COM', .25);

SELECT * FROM COMPANY;

COMP...	COMPANYNAME	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY	COMPANYREPNAME	CONTACTPHONE	CONTACTEMAIL	COMPANYDISCOUNTPERCENTAGE
1	The Helping Hands	RANDOM COMPANY ADDRESS	(null)	BRONX	NY	10451	UNITED STATES	JOEL STANLEY	8189002121	HELPINGHANDS@GMAIL.COM	0.1
2	Alternate	RANDOM COMPANY ADDRESS2	(null)	BROOKYLN	NY	11206	UNITED STATES	MASON RUJIERD	8189152011	THEALTERNATE@GMAIL.COM	0.2
3	THE DEADEND	RANDOM COMPANY ADDRESS3	(null)	BROOKLYN	NY	11208	UNITED STATES	RUDY GREYRAT	8185911421	DEADEND@GMAIL.COM	0.1
4	Snared Services	RANDOM COMPANY ADDRESS4	(null)	QUEENS	NY	11005	UNITED STATES	ERIS SHERLEY	8189648721	SNAREDSERVICES@GMAIL.COM	0.3
5	Random Repairs	RANDOM COMPANY ADDRESS5	(null)	MANHATTAN	NY	10013	UNITED STATES	ROXY MIGURDIA	8189120021	THERANDOMREPAIRS@GMAIL.COM	0.3

CORPORATECUSTOMER

CALL NEW_CORPCUST (2, 1);

CALL NEW_CORPCUST (4, 2);

CALL NEW_CORPCUST (5, 4);

CALL NEW_CORPCUST (6, 3);

CALL NEW_CORPCUST (9, 5);

SELECT * FROM COMPANY;

CUSTO...	COMPANYID
2	1
4	2
5	4
6	3
9	5

VEHICLESTATUS

INSERT ALL

INTO VEHICLESTATUS VALUES(1,'Reserved.')

INTO VEHICLESTATUS VALUES(2,'Rented.')

INTO VEHICLESTATUS VALUES(3,'Available.')

INTO VEHICLESTATUS VALUES(4,'Not Available')

INTO VEHICLESTATUS VALUES(5,'Maintenance.')

INTO VEHICLESTATUS VALUES(6,'Transferred to another agency')

SELECT 1 FROM DUAL;

SELECT * FROM VEHICLESTATUS;

VEHICLESTA...	VEHICLESTATUSDESC
1	Reserved.
2	Rented.
3	Available.
4	Not Available
5	Maintenance.
6	Transferred to another agency

VEHICLERENTALCATEGORY

```
INSERT ALL
INTO VEHICLERENTALCATEGORY VALUES(1,'Car-Economic', 113.99)
INTO VEHICLERENTALCATEGORY VALUES(2,'Car-Compact', 115.99)
INTO VEHICLERENTALCATEGORY VALUES(3,'Car-Intermediate', 116.67)
INTO VEHICLERENTALCATEGORY VALUES(4,'Car-Standard', 119.99)
INTO VEHICLERENTALCATEGORY VALUES(5,'Car-Full Size', 121.99)
INTO VEHICLERENTALCATEGORY VALUES(6,'Car-Premium', 127.79)
INTO VEHICLERENTALCATEGORY VALUES(7,'Car-Luxury', 139.99)
INTO VEHICLERENTALCATEGORY VALUES(8,'SUV-Intermediate', 127.99)
INTO VEHICLERENTALCATEGORY VALUES(9,'SUV-Standard', 128.99)
INTO VEHICLERENTALCATEGORY VALUES(10,'SUV-Standard Elite', 135.99)
INTO VEHICLERENTALCATEGORY VALUES(11,'SUV-Full Size', 148.99)
INTO VEHICLERENTALCATEGORY VALUES(12,'SUV-Premium', 157.99)
INTO VEHICLERENTALCATEGORY VALUES(13,'Minivan-Standard', 152.99)
INTO VEHICLERENTALCATEGORY VALUES(14,'Van-Passenger Van (12 passengers)', 161.00)
INTO VEHICLERENTALCATEGORY VALUES(15,'Van-Cargo Van', 19.95)
INTO VEHICLERENTALCATEGORY VALUES(16,'Pick Up-Mid Size', 69.95)
INTO VEHICLERENTALCATEGORY VALUES(17,'Pick Up-Full Size', 105.99)
INTO VEHICLERENTALCATEGORY VALUES(18,'Motorcycle-Touring', 19.95)
INTO VEHICLERENTALCATEGORY VALUES(19,'Motorcycle-Cruiser', 199.99)
INTO VEHICLERENTALCATEGORY VALUES(20,'Motorcycle-Scooter', 79.95)
SELECT 1 FROM DUAL;
SELECT * FROM VEHICLERENTALCATEGORY;
```

VEHICLERENTALCATEGORYID	CATEGORYNAME	CATEGORYDAILYRENTALRATE
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
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

VEHICLE

CALL NEW_VEHICLE ('1Y1SK5362RZ011666','Honda', 'Civic', 2019, 'Red', 'GA11828', 5219, 'Automatic', 4, 4, 1, 1, 1, 'C');

CALL NEW_VEHICLE ('KMHEC4A46DA063184','Volkswagen', 'Arteon', 2020, 'Black', '7999JZ', 2331, 'Automatic', 4, 4 ,3, 1, 1, 'C');

CALL NEW_VEHICLE ('2C3CCARG6DH634610','Dodge', 'Challenger', 2020, 'Black', 'BMJ1145', 200, 'Semi-automatic', 2, 4 ,4, 1, 1, 'C');

CALL NEW_VEHICLE ('1HGFA16577L069052','Nissan', 'Skyline', 2019, 'Burgundy', 'KGN8999', 2940, 'Continuously Variable Transmission', 4, 4, 1, 2, 2, 'C');

CALL NEW_VEHICLE ('KL1TD56628B083710','Nissan', 'Frontier', 2021, 'Blue', '5LOG380', 3114, 'Manual', 4, 4, 1, 2, 2, 'C');

CALL NEW_VEHICLE ('1FT7X2B6XCEB45481','Nissan', 'Skyline', 2020, 'White', 'SSS025', 4815, 'Manual', 4, 4, 3, 2, 5, 'C');

CALL NEW_VEHICLE ('1GTN2TEA7CZ340054','Chevrolet', 'Monza', 2020 , 'White', '7FJG526', 7349, 'Continuously Variable Transmission', 4, 4, 1, 3, 2, 'C');

CALL NEW_VEHICLE ('5VPWB36N7D3078709','Chrysler', '300', 2021, 'White', '8EYH330', 10010, 'Automatic', 4, 4, 3, 3, 1, 'C');

CALL NEW_VEHICLE ('1FMEU15H8PLA37915','Nissan', 'Maxima', 2018, 'White', '765YRT', 31095, 'dual-clutch', 4, 4, 1, 3, 1, 'C');

CALL NEW_VEHICLE ('YV1LS56D0Y2607105','Honda', 'Accord', 2020, 'Red', '855QKE', 2331, 'dual-clutch', 4, 4, 3, 4, 4, 'C');

CALL NEW_VEHICLE ('JF2SHADC0CH473478','Dodge', 'Charger', 2021, 'Blue', '359YUY', 904, 'Automatic', 2, 4, 3, 4, 4, 'C');

CALL NEW_VEHICLE ('3FAHP0JG0AR435957','Dodge', 'Durango', 2022, 'Black', '6HQA861', 2980, 'Automatic', 4, 4, 1, 4, 3, 'S');

CALL NEW_VEHICLE ('2CNDL63F496276211','GMC', 'Acadia', 2020, 'White', '8AJC490', 2211, 'Continuously Variable Transmission', 4, 4, 1, 5, 5, 'C');

CALL NEW_VEHICLE ('1FTYR14V71TA45330','Nissan', 'Altima', 2022, 'Grey', '6MOS144', 8799, 'Manual', 4, 4, 6, 5, 5, 'C');

CALL NEW_VEHICLE ('2T3BFREV8EW281337','Volkswagen', 'Arteon', 2020, 'Blue', '8FES335', 3117, 'dual-clutch', 4, 4, 1, 5, 2, 'C');

CALL NEW_VEHICLE ('2GCFC29K9L1206508','Honda', 'Civic', 2022, 'Grey', '8CYM960', 1200, 'Manual', 4, 4, 1, 1, 5, 'C');

CALL NEW_VEHICLE ('JM1BK323441184063','Nissan', 'Altima', 2020, 'White', 'GUX547', 21410, 'Automatic', 4, 4, 1, 2, 3, 'C');

CALL NEW_VEHICLE ('2T3ZFREV6DW035111','Jeep', 'Cherokee', 2021, 'Blue', 'DP361XR', 32130, 'Continuously Variable Transmission', 4, 4, 3, 3, 3, 'S');

CALL NEW_VEHICLE ('YV1RH59H432287819','Chrysler', '300', 2021, 'White', 'VVL8101', 33155, 'Manual', 4, 4, 2, 4, 3, 'C');

CALL NEW_VEHICLE ('1HGCP2F75AA036501','Jeep', 'Cherokee', 2021, 'Green', 'KHR2104', 13630, 'Continuously Variable Transmission', 4, 4, 6, 5, 1, 'S');

SELECT * FROM VEHICLE;

VEHI...	VINNUMBER	MAKE	MODEL	YEAR	COLOR	LICENSEPLATENUMBER	MILEAGE	TRANSMISSIONTYPE	SEATCAPACITY	VEHICLERENTALCATEGORYID	VEHICLESTATUSID	VEHICLEOWNINGAGENCYID	VEHICLECURRENTLOCATIONAGENCYID	VEHICLETYPE
1	1Y1SK5362RZ011666	Honda	Civic	2019	Red	GA11828	5219	Automatic	4	4	1	1	1	C
2	KMHEC4A46DA063184	Volkswagen	Arteon	2020	Black	7999JZ	2331	Automatic	4	4	3	1	1	C
3	2C3CCARG6DH634610	Dodge	Challenger	2020	Black	BMJ1145	200	Semi-automatic	2	4	4	1	1	C
4	1HGFA16577L069052	Nissan	Skyline	2019	Burgundy	KGN8999	2940	Continuously Variable Transmission	4	4	1	2	2	C
5	KL1TD56628B083710	Nissan	Frontier	2021	Blue	5LOG380	3114	Manual	4	4	1	2	2	C
6	1FT7X2B6XCEB45481	Nissan	Skyline	2020	White	SSS025	4815	Manual	4	4	3	2	5	C
7	1GTN2TEA7CZ340054	Chevrolet	Monza	2020	White	7FJG526	7349	Continuously Variable Transmission	4	4	1	3	2	C
8	5VFWB36N7D3078709	Chrysler	300	2021	White	8EYH330	10010	Automatic	4	4	3	3	1	C
9	1FMEU15H8PLA37915	Nissan	Maxima	2018	White	765YRT	31095	dual-clutch	4	4	1	3	1	C
10	YV1LS56D0Y2607105	Honda	Accord	2020	Red	855QKE	2331	dual-clutch	4	4	3	4	4	C
11	JF2SHADC0CH473478	Dodge	Charger	2021	Blue	359YUY	904	Automatic	2	4	3	4	4	C
12	3FAHP0JG0AR435957	Dodge	Durango	2022	Black	6HQA861	2980	Automatic	4	4	1	4	3	S
13	2CNDL63F496276211	GMC	Acadia	2020	White	8AJC490	2211	Continuously Variable Transmission	4	4	1	5	5	C
14	1FTYR14V71TA45330	Nissan	Altima	2022	Grey	6MOS144	8799	Manual	4	4	6	5	5	C
15	2T3BFREV8EW281337	Volkswagen	Arteon	2020	Blue	8FES335	3117	dual-clutch	4	4	1	5	2	C
16	2GCFC29K9L1206508	Honda	Civic	2022	Grey	8CYM960	1200	Manual	4	4	1	1	5	C
17	JM1BK323441184063	Nissan	Altima	2020	White	GUX547	21410	Automatic	4	4	1	2	3	C
18	2T3ZFREV6DW035111	Jeep	Cherokee	2021	Blue	DP361XR	32130	Continuously Variable Transmission	4	4	3	3	3	S
19	YV1RH59H432287819	Chrysler	300	2021	White	VVL8101	33155	Manual	4	4	2	4	3	C
20	1HGCP2F75AA036501	Jeep	Cherokee	2021	Green	KHR2104	13630	Continuously Variable Transmission	4	4	6	5	1	S

CAR

CALL NEW_CAR(1,11.00);

CALL NEW_CAR(2,20.1);

CALL NEW_CAR(3,17);

CALL NEW_CAR(4,17);

CALL NEW_CAR(5,17);

CALL NEW_CAR(6,17);

CALL NEW_CAR(7,17);

EZRENTAL POS

```
CALL NEW_CAR(8,17);
CALL NEW_CAR(9,17);
CALL NEW_CAR(10,17);
CALL NEW_CAR(11,17);
CALL NEW_CAR(13,16);
CALL NEW_CAR(14,16);
CALL NEW_CAR(15,19);
CALL NEW_CAR(16,21);
CALL NEW_CAR(17,18);
CALL NEW_CAR(19,24);
SELECT * FROM CAR;
```

VEHICLEID	TRUNKCAPACITY
1	11
2	20.1
3	17
4	17
5	17
6	17
7	17
8	17
9	17
10	17
11	17
13	16
14	16
15	19
16	21
17	18
19	24

SUV

```
CALL NEW_SUV(12, 8700,'1');
CALL NEW_SUV(18, 6200,'1');
CALL NEW_SUV(20, 6200,'1');
SELECT * FROM SUV;
```

VEHICLEID	TOWINGCAPACITY	ISAWD
12	8700	1
18	6200	1
20	6200	1

RESERVATIONSTATUS

```
INSERT ALL
INTO RESERVATIONSTATUS VALUES(1,'Confirmed.')
INTO RESERVATIONSTATUS VALUES(2,'Modified & Reconfirmed.')
INTO RESERVATIONSTATUS VALUES(3,'Cancelled & Closed.')
INTO RESERVATIONSTATUS VALUES(4,'Fulfilled & Closed.')
SELECT 1 FROM DUAL;
SELECT * FROM RESERVATIONSTATUS;
```

EZRENTAL POS

RESERVATIONSTATUSID	RESERVATIONSTATUSDESC
1	Confirmed.
2	Modified & Reconfirmed.
3	Cancelled & Closed.
4	Fulfilled & Closed.

RENTALSTATUS

```
INSERT ALL

INTO RENTALSTATUS VALUES(1,'Picked up as scheduled.')

INTO RENTALSTATUS VALUES(2,'Dropped off as scheduled.')

INTO RENTALSTATUS VALUES(3,'Returned late.')

INTO RENTALSTATUS VALUES(4,'In progress.')

INTO RENTALSTATUS VALUES(5,'Roadside assistance in progress.')

INTO RENTALSTATUS VALUES(6,'Unknown.')

SELECT 1 FROM DUAL;
```

SELECT * FROM RENTALSTATUS;

RENTALSTATUSID	RENTALSTATUSDESC
1	Picked up as scheduled.
2	Dropped off as scheduled.
3	Returned late.
4	In progress.
5	Roadside assistance in progress.
6	Unknown.

RENTALINSURANCEOPTION

INSERT INTO RENTALINSURANCEOPTION VALUES (1, 'No insurance. Opt-out.', 0.00);

INSERT INTO RENTALINSURANCEOPTION VALUES (2, 'Collision Damage Waiver Max - Agency will pay for damage, lost or stolen vehicle.', 49.99);

INSERT INTO RENTALINSURANCEOPTION VALUES (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);

INSERT INTO RENTALINSURANCEOPTION VALUES (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);

INSERT INTO RENTALINSURANCEOPTION VALUES (5, 'Roadside Assistance Plus – 24/7 roadside assistance, replacement for lost keys, flat tire service, fuel delivery, etc.', 15.99);

SELECT * FROM RENTALINSURANCEOPTION;

EZRENTAL POS

INSURANCEOPTIONID	INSURANCEOPTIONDESC	INSURANCEOPTIONADDITIONALCOST
1	No insurance. Opt-out.	0
2	Collision Damage Waiver Max - Agency will pay for damage, lost or stolen vehicle.	49.99
3	Collision Damage Waiver 3000 - Agency will pay for first $3,000 of loss or damage, renter pays all loss & damage after $3,000.	39.99
4	Liability Extended Protection - Agency provides renter with third party liability protection up to $1 Million per accident for bodily injury or death or property damage to others.	89.99
5	Roadside Assistance Plus - 24/7 roadside assistance, replacement for lost keys, flat tire service, fuel delivery, etc.	15.99

FUELOPTION

INSERT INTO FUELOPTION VALUES (1, 'Return with a full tank or on return, pay for gas that is missing');

INSERT INTO FUELOPTION VALUES (2, 'Pay for full tank in advanced at time of rental, return car empty. No refund for unused gas.');

SELECT * FROM FUELOPTION;

FUELOPTIONID	FUELOPTIONDESC
1	Return with a full tank or on return, pay for gas that is missing
2	Pay for full tank in advanced at time of rental, return car empty. No refund for unused gas.

RESERVATION

CALL MAKE_RESERVATION(1, 1, 3, 4, 1, '01-JAN-2022', 1230, '08-JAN-2022', 1400, 1);

CALL MAKE_RESERVATION(2, 2, 1, 4, 2, '03-JAN-2022', 1200, '13-JAN-2022', 1200, 1);

CALL MAKE_RESERVATION(3, 3, 2, 4, 3, '10-JAN-2022', 0830, '21-JAN-2022', 1530, 1);

CALL MAKE_RESERVATION(4, 4, 5, 4, 4, '09-JAN-2022', 0900, '21-JAN-2022', 1600, 1);

CALL MAKE_RESERVATION(5, 5, 4, 4, 4, '06-JAN-2022', 1030, '24-JAN-2022', 1500, 1);

SELECT * FROM RESERVATION;

RESERVATIONID	CUSTOMERID	RENTALAGENCYID	VEHICLERENTALCATEGORYID	RESERVATIONDROPOFFAGENCYID	RESERVATIONPICKUPDATE	RESERVATIONPICKUPTIME	RESERVATIONDROPOFFDATE	RESERVATIONDROPOFFTIME	RESERVATIONSTATUSID
1	1	3	4	1	01-JAN-22	1230	08-JAN-22	1400	1
2	2	1	4	2	03-JAN-22	1200	13-JAN-22	1200	1
3	3	2	4	3	10-JAN-22	830	21-JAN-22	1530	1
4	4	5	4	4	09-JAN-22	900	21-JAN-22	1600	1
5	5	4	4	4	06-JAN-22	1030	24-JAN-22	1500	1

RENTAL

CALL NEW_RENTAL(1, 3, 1, 1, 1, '01-JAN-2022', 0930, '11-JAN-2022', 1400, 210, 350, 1, 5, 1, 300);

CALL NEW_RENTAL(2, 7, 3, 4, 2, '03-JAN-2022', 1200, '13-JAN-2022', 1200, 7349, 7601, 1, 5, 1, 150);

CALL NEW_RENTAL(3, 11, 4, 4, 3, '10-JAN-2022', 0830, '21-JAN-2022', 1530, 904, 1410, 1, 5, 1, 300);

CALL NEW_RENTAL(4, 15, 5, 3, 4, '09-JAN-2022', 0900, '21-JAN-2022', 1600, 2980, 3312, 1, 5, 1, 150);

CALL NEW_RENTAL(5, 12, 4, 2, 5, '06-JAN-2022', 1030, '24-JAN-2022', 1400, 3117, 1943, 1, 5, 1, 400);

SELECT * FROM RENTAL;

RENTALAGREEMENTID	CUSTOMERID	VEHICLEID	RENTALAGENCYID	RENTALDROPOFFAGENCYID	RESERVATIONID	RENTALPICKUPDATE	RENTALPICKUPTIME	RENTALDROPOFFDATE	RENTALDROPOFFTIME	RENTALPICKUPODOMETERVALUE	RENTALDROPOFFODOMETERVALUE	RENTALFUELOPTIONID	INSURANCEOPTIONID	RENTALSTATUSID	RENTALDEPOSIT
1	1	3	1	1	1	01-JAN-22	930	11-JAN-22	1400	210	350	1	5	1	300
2	2	7	3	4	2	03-JAN-22	1200	13-JAN-22	1200	7349	7601	1	5	1	150
3	3	11	4	4	3	10-JAN-22	830	21-JAN-22	1530	904	1410	1	5	1	300
4	4	15	5	3	4	09-JAN-22	900	21-JAN-22	1600	2980	3312	1	5	1	150
5	5	12	4	2	5	06-JAN-22	1030	24-JAN-22	1400	3117	1943	1	5	1	400

EMPLOYEEUSERACCOUNT

CALL NEW_EMPLOYEEUSERACC('PAULGREY','greysonpaul123!','paulgrey@gmail.com');

CALL NEW_EMPLOYEEUSERACC('EVELYNHOWS','PASSEVEL','howEvelyn@gmail.com');

CALL NEW_EMPLOYEEUSERACC('ZENITH','PASSZENTH','ZENITHGOODMEN@gmail.com');

CALL NEW_EMPLOYEEUSERACC('JAMESXER','JAMEYXER','XERATHJAMES@gmail.com');

CALL NEW_EMPLOYEEUSERACC('GABRIELL','PASSGAB222','GABJULIO@gmail.com');

SELECT * FROM EMPLOYEEUSERACCOUNT;

USERACCOUNTID	USERNAME	PASSWORD	EMAIL
3FDFCB600DEE4869BAA3B667D2E45E78	PAULGREY	greysonpaul123!	paulgrey@gmail.com
D03E210B938F4E4B9C96F1596DC3FA2A	EVELYNHOWS	PASSEVEL	howEvelyn@gmail.com
A8E802C0623B49728691911AFFC8BDAD	ZENITH	PASSZENTH	ZENITHGOODMEN@gmail.com
942305511F124CB1BACEC089F98A19A1	JAMESXER	JAMEYXER	XERATHJAMES@gmail.com
9CEC498CFCA941B99F57D2483361A69F	GABRIELL	PASSGAB222	GABJULIO@gmail.com

EMPLOYEE

CALL NEW_EMPLOYEE('Paul', 'Greyson', '425691234', '01-JAN-1986', 'SOME RANDOM ADDRESS 0000', 'Brooklyn', 'NY', '11236', 'United States', '8568647644','Rental Delivery Agent', 'paulgrey@gmail.com', '3FDFCB600DEE4869BAA3B667D2E45E78');

CALL NEW_EMPLOYEE('Evelyn', 'Howards', '613467237', '11-MAR-1976', 'SOME RANDOM ADDRESS 98322', 'Manhattan', 'NY', '10003', 'United States', '8325793485','Rental Delivery Agent', 'howEvelyn@gmail.com', 'D03E210B938F4E4B9C96F1596DC3FA2A');

CALL NEW_EMPLOYEE('Zenith', 'Goodman', '236719712', '25-DEC-1995', 'SOME RANDOM ADDRESS 1166', 'Bronx', 'NY', '10465 ', 'United States', '6017624532','Rental Delivery Agent', 'ZENITHGOODMEN@gmail.com', 'A8E802C0623B49728691911AFFC8BDAD');

CALL NEW_EMPLOYEE('James', 'Xerath', '721671534', '07-APR-2000', 'SOME RANDOM ADDRESS 1234', 'Manhattan', 'NY', '10024 ', 'United States', '4193285639','Rental Delivery Agent', 'XERATHJAMES@gmail.com', '942305511F124CB1BACEC089F98A19A1');

CALL NEW_EMPLOYEE('Gabriel', 'Julioso', '425461858', '22-DEC-1992', 'SOME RANDOM ADDRESS 7238', 'Brooklyn', 'NY', '11214', 'United States', '3478953308','Rental Delivery Agent', 'GABJULIO@gmail.com', '9CEC498CFCA941B99F57D2483361A69F');

SELECT * FROM EMPLOYEE;

EMPLOYEEID	FIRSTNAME	LASTNAME	SSNUMBER	BIRTHDATE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY	EMPLOYEEPHONE	EMPLOYEEJOBTITLE	EMPLOYEEEMAIL	EMPLOYEEUSERACCOUNTID
5	Gabriel	Julioso	425461858	22-DEC-92	SOME RANDOM ADDRESS 7238	2ND ADDRESS	Brooklyn	NY	11214	United States	3478953308	Rental Delivery Agent	GABJULIO@gmail.com	9CEC498CFCA941B99F57D2483361A69F
1	Paul	Greyson	425691234	01-JAN-86	SOME RANDOM ADDRESS 0000	(null)	Brooklyn	NY	11236	United States	8568647644	Rental Delivery Agent	paulgrey@gmail.com	3FDFCB600DEE4869BAA3B667D2E45E78
2	Evelyn	Howards	613467237	11-MAR-76	SOME RANDOM ADDRESS 98322	(null)	Manhattan	NY	10003	United States	8325793485	Rental Delivery Agent	howEvelyn@gmail.com	D03E210B938F4E4B9C96F1596DC3FA2A
3	Zenith	Goodman	236719712	25-DEC-95	SOME RANDOM ADDRESS 1166	(null)	Bronx	NY	10465	United States	6017624532	Rental Delivery Agent	ZENITHGOODMEN@gmail.com	A8E802C0623B49728691911AFFC8BDAD
4	James	Xerath	721671534	07-APR-00	SOME RANDOM ADDRESS 1234	(null)	Manhattan	NY	10024	United States	4193285639	Rental Delivery Agent	XERATHJAMES@gmail.com	942305511F124CB1BACEC089F98A19A1

TRANSPORTSTATUS

INSERT INTO TRANSPORTSTATUS VALUES(1, 'Transport completed');

INSERT INTO TRANSPORTSTATUS VALUES(2, 'On route to pick up location');

INSERT INTO TRANSPORTSTATUS VALUES(3, 'On route from pick up location');

INSERT INTO TRANSPORTSTATUS VALUES(4, 'At pickup location. In progress (Loading etc.)');

INSERT INTO TRANSPORTSTATUS VALUES(5, 'Pickup location delay');

INSERT INTO TRANSPORTSTATUS VALUES(6, 'Unknown');

SELECT * FROM TRANSPORTSTATUS;

TRANSPORTSTA...	TRANSPORTSTATUSDESC
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
6	Unknown

TRANSPORTREASON

```
INSERT ALL
INTO TRANSPORTREASON VALUES(1,'Rental Drop off at different location')
INTO TRANSPORTREASON VALUES(2,'Vehicle Loaned to another Agency')
INTO TRANSPORTREASON VALUES(3,'Pick up from Distribution Center')
INTO TRANSPORTREASON VALUES(4,'Drop off to Distribution Center')
INTO TRANSPORTREASON VALUES(5,'Vehicle sent for maintenance')
INTO TRANSPORTREASON VALUES(6,'Unknown.')
SELECT 1 FROM DUAL;
```

```
SELECT * FROM TRANSPORTREASON;
```

TRANSPORTREAS...	TRANSPORTREASONDESC
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
6	Unknown.

TRANSPORT

CALL NEW_TRANSPORT(1, 3, 3, 1,'01-JAN-22', 0730,'01-JAN-22',0830, '01-JAN-22', 0900, 4, 1);

CALL NEW_TRANSPORT(2, 7, 1, 2,'03-JAN-22', 0930,'03-JAN-22',1030, '03-JAN-22', 1130, 4, 1);

CALL NEW_TRANSPORT(3, 11, 2, 3,'28-DEC-21', 0730,'28-DEC-21',1045, '28-DEC-21', 1530, 2, 4);

CALL NEW_TRANSPORT(4, 11, 2, 3,'02-JAN-22', 0930,'02-JAN-22',1030, '02-JAN-22', 1400, 3, 3);

SELECT * FROM TRANSPORT;

EMPLO...	VEHICLEID	PICKUPRENTALAGENCYID	DROPOFFRENTALAGENCYID	DRIVERDEPARTUREDATE	DRIVERDEPARTURETIME	VEHICLEPICKUPDATE	VEHICLEPICKUPTIME	VEHICLEDROPOFFDATE	VEHICLEDROPOFFTIME	TRANSPORTSTATUSID	TRANSPORTREASONID
1	3	3	1	01-JAN-22	730	01-JAN-22	830	01-JAN-22	900	4	1
2	7	1	2	03-JAN-22	930	03-JAN-22	1030	03-JAN-22	1130	4	1
3	11	2	3	28-DEC-21	730	28-DEC-21	1045	28-DEC-21	1530	2	4
4	11	2	3	02-JAN-22	930	02-JAN-22	1030	02-JAN-22	1400	3	3

SELECT QUERIES:

This query returns the Make, Total amount of cars within that make, and Average trunk capacity of that make. I use a left join to connect the VEHICLE and CAR table on their related column of VEHICLEID. I then group them and utilize the having clause to find the specific make that has a trunk capacity greater than or equal to 20.

```
SELECT  MAKE, COUNT(MAKE) AS "TOTAL CARS", AVG(TRUNKCAPACITY) AS "AVERAGE TRUNK CAPACITY"
FROM CAR

LEFT JOIN VEHICLE ON VEHICLE.VEHICLEID = CAR.VEHICLEID

GROUP BY MAKE

HAVING  AVG(TRUNKCAPACITY) >= 20;
```

MAKE	TOTAL CARS	AVERAGE TRUNK CAPACITY
Chrysler	2	20.5

This query returns the customer's ID, customer's credit card number, their first and last name, the type of customer they are, their credit card limit, and their credit card balance. Using a left join, I was able to connect the 3 tables of CUSTOMER, CREDITCARD AND CUSTOMERCREDITCARD using the related columns between these tables which are CUSTOMERID and CARDNUMBER. The WHERE clause is used to only return those who have a balance under 10000. Lastly, I order the results by the CUSTOMERID in an ascending order.

```
SELECT CC.CUSTOMERID, CC.CARDNUMBER, CUS.FIRSTNAME, CUS.LASTNAME, CUS.CUSTOMERTYPE,
CRED.CREDITCARDLIMIT, CRED.CREDITCARDBALANCE FROM CUSTOMERCREDITCARD CC

LEFT JOIN CUSTOMER CUS ON CUS.CUSTOMERID = CC.CUSTOMERID

LEFT JOIN CREDITCARD CRED ON CRED.CREDITCARDNUMBER = CC.CARDNUMBER

WHERE CRED.CREDITCARDBALANCE <= 10000

ORDER BY CUSTOMERID;
```

CUSTOMERID	CARDNUMBER	FIRSTNAME	LASTNAME	CUSTOMERTYPE	CREDITCARDLIMIT	CREDITCARDBALANCE
7	2221003690226132	John	Shall	R	8000	5000
8	345953192054975	JAKE	NULL	R	13000	4500
9	4024007135713930	MACK	QUEEN	C	10000	2000

EZRENTAL POS

This query returns the customer's CUSTOMERID, their FIRST and LAST NAME, DISCOUNT CODE associated with their ACCOUNT, and the amount of EZPLUS REWARDS POINTS they currently own. Using a left join, I was able to connect the RETAILCUSTOMER, DISCOUNT and EZPLUS table on their related column of CUSTOMERID, DISCOUNTID, AND EZPLUSID. I decide to order the results by the EZPLUSREWARDSEARNEDPOINTS in the descending order as I'd like to see who has the most points.

```
SELECT CUSTOMER.CUSTOMERID, FIRSTNAME, LASTNAME, DISCOUNTCODE, EZPLUSREWARDSEARNEDPOINTS
FROM RETAILCUSTOMER

LEFT JOIN CUSTOMER ON RETAILCUSTOMER.CUSTOMERID = CUSTOMER.CUSTOMERID

LEFT JOIN DISCOUNT ON RETAILCUSTOMER.DISCOUNTID = DISCOUNT.DISCOUNTID

LEFT JOIN EZPLUS ON RETAILCUSTOMER.EZPLUSID = EZPLUS.EZPLUSID

ORDER BY EZPLUSREWARDSEARNEDPOINTS DESC;
```

CUSTOMERID	FIRSTNAME	LASTNAME	DISCOUNTCODE	EZPLUSREWARDSEARNEDPOINTS
8	JAKE	NULL	AAA9970054	20000
7	John	Shall	STA3415632	12000
1	TYLER	Walker	VET2055179	10000
3	Alex	Rodington	GOV8756921	10000

UPDATE QUERIES:

CREDITCARD

The update statement I have executed has changed every column for the user of the credit card that is 4024007135713930.

CREDITCARDNUMBER	OWNERNAME	MERCHANTNAME	EXPDATE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY	CREDITCARDLIMIT	CREDITCARDBALANCE	ACTIVATIONSTATUS
4024007135713930	MACK QUEEN	VISA	07-FEB-27	SOMEADDRESS	SOMEWHERE	(null)	MANHATTAN	NY	10013	UNITED STATES	10000	2000	1
6011368248848918	MALOTTO ARASHIO	DISCOVER	13-DEC-31	SOMEADDRESS	SOMEWHERE	(null)	MANHATTAN	NY	10012	UNITED STATES	30000	18600	1
373102971940153	KEKE WILLKINS	AMERICAN EXPRESS	16-NOV-22	SOMEADDRESS	SOMEWHERE	(null)	MANHATTAN	NY	10013	UNITED STATES	20000	11000	1
345953192054975	JAKE NULL	AMERICAN EXPRESS	01-JAN-25	SOMEADDRESS	SOMEWHERE	(null)	BROOKLYN	NY	11204	UNITED STATES	13000	4500	1
2221003690226132	JOHN SHALL	MASTERCARD	27-MAR-23	SOMEADDRESS	SOMEWHERE	(null)	BRONX	NY	10451	UNITED STATES	8000	5000	1

UPDATE CREDITCARD SET OWNERNAME = 'Mack Queen', MERCHANTNAME = 'Visa', EXPDATE = '07-MAR-2028', ADDRESSLINE1 = 'OFFICIAL ADDRESS OF THIS RESIDENT', CITY = 'Manhattan', STATECODE = 'FL', ZIPCODE = '10013', COUNTRY = 'United States of America', CREDITCARDLIMIT = 15000, CREDITCARDBALANCE = 3500, ACTIVATIONSTATUS = '0' WHERE CREDITCARDNUMBER = '4024007135713930';

	CREDITCARDNUMBER	OWNERNAME	MERCHANTNAME	EXPDATE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY	CREDITCARDLIMIT	CREDITCARDBALANCE	ACTIVATIONSTATUS
1	4024007135713930	Mack Queen	Visa	07-MAR-28	OFFICIAL ADDRESS OF THIS RESIDENT	(null)	Manhattan	FL	10013	United States of America	15000	3500	0
2	6011368248848918	MALOTTO ARASHIO	DISCOVER	13-DEC-31	SOMEADDRESS	SOMEWHERE	(null)	MANHATTAN	NY	10012	UNITED STATES	30000	18600	1
3	373102971940153	KEKE WILLKINS	AMERICAN EXPRESS	16-NOV-22	SOMEADDRESS	SOMEWHERE	(null)	MANHATTAN	NY	10013	UNITED STATES	20000	11000	1
4	345953192054975	JAKE NULL	AMERICAN EXPRESS	01-JAN-25	SOMEADDRESS	SOMEWHERE	(null)	BROOKLYN	NY	11204	UNITED STATES	13000	4500	1
5	2221003690226132	JOHN SHALL	MASTERCARD	27-MAR-23	SOMEADDRESS	SOMEWHERE	(null)	BRONX	NY	10451	UNITED STATES	8000	5000	1

CUSTOMERCREDITCARD

The update statement I have executed exchanges ownership of the credit card 4024007135713930 from customer 9 to customer 4.

CARDNUMBER	CUSTOMERID
2221003690226132	7
345953192054975	8
373102971940153	5
4024007135713930	9
6011368248848918	6

UPDATE CUSTOMERCREDITCARD SET CUSTOMERID = 4 WHERE CARDNUMBER = '4024007135713930' AND CUSTOMERID = 9;

CARDNUMBER	CUSTOMERID
2221003690226132	7
345953192054975	8
373102971940153	5
4024007135713930	4
6011368248848918	6

DELETE QUERIES:

EMPLOYEE

This delete statement, deletes from the EMPLOYEE table where the EMPLOYEEID is equal to 2.

SELECT * FROM EMPLOYEE;

EMPLOYEEID	FIRSTNAME	LASTNAME	SSNUMBER	BIRTHDATE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY	EMPLOYEEPHONE	EMPLOYEEJOBTITLE	EMPLOYEEEMAIL	EMPLOYEEUSERACCOUNTID
5	Gabriel	Julioso	425461858	22-DEC-92	SOME RANDOM ADDRESS	7238 2ND ADDRESS	Brooklyn	NY	11214	United States	3478953308	Rental Delivery Agent	GABJULIO@gmail.com	9CEC498CFCA941B99F57D2483361A69F
1	Paul	Greyson	425691234	01-JAN-86	SOME RANDOM ADDRESS	0000 (null)	Brooklyn	NY	11236	United States	8568647644	Rental Delivery Agent	paulgrey@gmail.com	3FDFCB600DEE4869BAA3B667D2E45E78
2	Evelyn	Howards	613467237	11-MAR-76	SOME RANDOM ADDRESS	98322 (null)	Manhattan	NY	10003	United States	8325793485	Rental Delivery Agent	howEvelyn@gmail.com	D03E210B938F4E4B9C96F1596DC3FA2A
3	Zenith	Goodman	236719712	25-DEC-95	SOME RANDOM ADDRESS	1166 (null)	Bronx	NY	10465	United States	6017624532	Rental Delivery Agent	ZENITHGOODMEN@gmail.com	A8E802C0623B4972869191 1AFFC8BDAD
4	James	Xerath	721671534	07-APR-00	SOME RANDOM ADDRESS	1234 (null)	Manhattan	NY	10024	United States	4193285639	Rental Delivery Agent	XERATHJAMES@gmail.com	942305511F124CB1BACEC089F98A19A1

DELETE FROM EMPLOYEE WHERE EMPLOYEEID = 5;

EMPLOYEEID	FIRSTNAME	LASTNAME	SSNUMBER	BIRTHDATE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY	EMPLOYEEPHONE	EMPLOYEEJOBTITLE	EMPLOYEEEMAIL	EMPLOYEEUSERACCOUNTID
1	Paul	Greyson	425691234	01-JAN-86	SOME RANDOM ADDRESS	0000 (null)	Brooklyn	NY	11236	United States	8568647644	Rental Delivery Agent	paulgrey@gmail.com	3FDFCB600DEE4869BAA3B667D2E45E78
2	Evelyn	Howards	613467237	11-MAR-76	SOME RANDOM ADDRESS	98322 (null)	Manhattan	NY	10003	United States	8325793485	Rental Delivery Agent	howEvelyn@gmail.com	D03E210B938F4E4B9C96F1596DC3FA2A
3	Zenith	Goodman	236719712	25-DEC-95	SOME RANDOM ADDRESS	1166 (null)	Bronx	NY	10465	United States	6017624532	Rental Delivery Agent	ZENITHGOODMEN@gmail.com	A8E802C0623B4972869191 1AFFC8BDAD
4	James	Xerath	721671534	07-APR-00	SOME RANDOM ADDRESS	1234 (null)	Manhattan	NY	10024	United States	4193285639	Rental Delivery Agent	XERATHJAMES@gmail.com	942305511F124CB1BACEC089F98A19A1

CUSTOMERCREDITCARD

This delete statement, deletes from the CUSTOMERCREDITCARD table where it'll delete where the CUSTOMERID is equal to 4.

	CARDNUMBER	CUSTOMERID
1	2221003690226132	7
2	345953192054975	8
3	373102971940153	5
4	4024007135713930	4
5	6011368248848918	6

DELETE FROM CUSTOMERCREDITCARD WHERE CUSTOMERID = 4;

CARDNUMBER	CUSTOMERID
2221003690226132	7
345953192054975	8
373102971940153	5
6011368248848918	6

Conclusion

It can be well concluded that the point-of-sale system implemented by NYC-Tech Solutions Inc. allows for both retail and corporate customers to reserve and rent the vehicles of their choice. Giving EZRental Inc. the capability to track all necessary transactions and information pertaining their multiple establishments.

EZRENTAL POS AUTO RENTAL POINT-OF-SALES MANAGEMENT SYSTEM VERSION 2

Upgrade Project objectives

Business Reports Queries & Summary

Business Reports Stored Procedures & Summary

Conclusion

Upgrade Project objectives

Rehired by NYC-Tech Solutions as a consultant, I was tasked to upgrade the EZRental Point of Sales Management System to its Version 2.0.

I will be responsible for creating business report queries and stored procedures that would increase the productivity and efficiency for the multiple roles within this business.

These roles include, but are not limited to:

- EZRental Executives
- Marketing, Finance, Accounting, HR etc.
- Front-desk Customer Service Team, Online Customer Service Team etc.
  Housekeeping management & staff (those who manage the cleaning of the rooms, buildings, etc.)
- Maintenance (people who manage and execute any repairs such as plumbing, electricians, etc.)
- Vehicle Inventory managers – People who manage the inventory of rental vehicles and manage the fleet.
- Vehicle Maintenance (people who manage and execute any repairs of the vehicles etc.)
- Vehicle Transport personnel (people who pick-up and drop-off vehicles as part of business activities)
- Customers who rent vehicles.

After research and using my previous and newfound knowledge of renting vehicles I was able to come up with multiple relevant business report queries.

Business Reports Queries & Summary

REPORT #1

DESIGN GUIDELINES	ANSWERS
1) Business Scenario & Objectives	To provide vehicle maintenance for each rental agency service with cars who's mileage ranges between 10,000 to 35,000 miles for precautionary tune ups & repairs.
2) Target Persona/Decision Maker	Vehicle Maintenance & Vehicle Inventory managers
3) What is the Decision to be made?	The Vehicle Maintenance Crew needs to decide which vehicles need tune ups and repairs. Vehicle Inventory managers need to know which vehicles need to be set to unavailable due to this maintenance.
4) What Data is required to make the Decision?	The name of the rental agency that owns the car to notify them of the repairs that need to take action, the vehicles vin numbers for identification on receival, the vehicle's make and model for part ordering purposes, the vehicles mileage, the vehicles current status (Directed towards Vehicle inventory managers responsible for changing the status once they've confirmed it was sent and received for repair)

5) Identify Tables that Contains the Data Identified in # 4 & idea of the SELECT Query required	The following Tables in the database schema were analyzed and identified to contain the required data: ❖ VEHICLE – VINNUMBER, MAKE, MODEL, MILEAGE ❖ RENTALAGENCY – RENTALAGENCYNAME ❖ VEHICLESTATUS – VEHICLESTATUSDESC We would need a SELECT statement that JOINS and queries these three tables: **VEHICLE, RENTALAGENCY & VEHICLESTATUS**

Query Report:

```
SELECT RA.RENTALAGENCYNAME, VINNUMBER, MAKE, MODEL, MILEAGE, VEHICLESTATUSDESC FROM
VEHICLE V

LEFT JOIN RENTALAGENCY RA ON RA.RENTALAGENCYID = V.VEHICLEOWNINGAGENCYID

LEFT JOIN VEHICLESTATUS VS ON VS.VEHICLESTATUSID = V.VEHICLESTATUSID

WHERE MILEAGE BETWEEN 10000 AND 35000

ORDER BY RA.RENTALAGENCYNAME;
```

Results of Execution:

RENTALAGENCYNAME	VINNUMBER	MAKE	MODEL	MILEAGE	VEHICLESTATUSDESC
ALPHA CAR COMPANY	YV1RH59H432287819	Chrysler	300	33155	Rented.
FAST TRACK	5VPWB36N7D3078709	Chrysler	300	10010	Available.
FAST TRACK	1FMEU15H8PLA37915	Nissan	Maxima	31095	Reserved.
FAST TRACK	2T3ZFREV6DW035111	Jeep	Cherokee	32130	Available.
UNIVERSAL RENTALS	JM1BK323441184063	Nissan	Altima	21410	Reserved.
ZAWA CAR RENTALS	1HGCP2F75AA036501	Jeep	Cherokee	13630	Transferred to another agency

REPORT #2

DESIGN GUIDELINES	ANSWERS
1) Business Scenario & Objectives	A promotional campaign to target customers who has rented 3 or more vehicles in a 4-month span of their first rental. Giving these customers a physical coupon and a digital discount through email and text
2) Target Persona/Decision Maker	Marketing team
3) What is the Decision to be made?	The Marketing team needs to decide which customers to email the discount to and from which car rental agency do they receive said discount for. This is decided by retail customers who has rented 3 or more vehicles in a 4-month span since their first rental.
4) What Data is required to make the Decision?	Customer Name, Customer's Primary Address, Customer's phone number, Customer's Email. First rental pick up date, Total cars rented Time until discount (4 months after first rental)
5) Identify Tables that Contain the Data Identified in # 4 & idea of the SELECT Query required	The following Tables in the database schema were analyzed and identified to contain the required data: ❖ CUSTOMER – FIRSTNAME,

	LASTNAME, ADDRESSLINE1, CITY, STATECODE, ZIPCODE, COUNTRY, PHONE, EMAIL, ❖ RENTALAGENCY – RENTALAGENCYNAME ❖ RENTAL – RENTALPICKUPDATE We would need a SELECT statement that JOINS and queries these two tables: **CUSTOMER & RENTAL**

Query Report:

```
SELECT  FIRSTNAME ||' '|| LASTNAME "FULL NAME", CUS.ADDRESSLINE1 ||' '||CUS.CITY|| ' ' ||
CUS.STATECODE || ' ' || CUS.ZIPCODE ||' ' || CUS.COUNTRY AS "Mailing Address", CUS.PHONE, CUS.EMAIL,

MIN(RENTAL.RENTALPICKUPDATE) "First Rental Date", count(*) "Total Cars
Rented" ,ADD_MONTHS(MIN(RENTAL.RENTALPICKUPDATE) , 4) "4 Month Span" FROM RENTAL

LEFT JOIN CUSTOMER CUS ON CUS.CUSTOMERID = RENTAL.CUSTOMERID

GROUP BY CUS.CUSTOMERID, FIRSTNAME ||' '|| LASTNAME, CUS.ADDRESSLINE1 ||' '||CUS.CITY|| ' ' ||
CUS.STATECODE || ' ' || CUS.ZIPCODE ||' ' || CUS.COUNTRY, CUS.PHONE, CUS.EMAIL

HAVING COUNT(*) >= 3

ORDER BY CUS.CUSTOMERID;
```

Results of Execution:

FULL NAME	Mailing Address	PHONE	EMAIL	First Rental Date	Total Cars Rented	4 Month Span
TYLER Walker	Example Address1234 Brooklyn NY 11236 United States	9292002258	walktyler25@gmail.com	01-JAN-22	4	01-MAY-22
JAKE NULL	SOMEADDRESS SOMEWHERE BROOKYLN NY 11204 UNITED STATES	1113334444	NULLJAKE@mail.com	10-MAY-22	3	10-SEP-22

REPORT #3

DESIGN GUIDELINES	ANSWERS
1) Business Scenario & Objectives	A promotional campaign to reward retail customers who frequently rents from the same car rental agency twice or more. Giving these

	customers a physical coupon and a digital discount through email and text. The discount will only be given to those that frequent a car rental service.
2) Target Persona/Decision Maker	Marketing team
3) What is the Decision to be made?	The Marketing team needs to decide which retail customers to email the discount to, and from which car rental agency do they receive said discount for. They also need to be able to determine which car rental agency they frequent the most.
4) What Data is required to make the Decision?	Customer Name, Customer's Primary Address, Customer's phone number, Customer's Email. Rental Agency Name
5) Identify Tables that Contain the Data Identified in # 4 & idea of the SELECT Query required	The following Tables in the database schema were analyzed and identified to contain the required data: ❖ CUSTOMER – FIRSTNAME, LASTNAME, ADDRESSLINE1, CITY, STATECODE, ZIPCODE, COUNTRY, PHONE, EMAIL, ❖ RENTALAGENCY – RENTALAGENCYNAME We would need a SELECT statement that JOINS and queries these two tables: **CUSTOMER & RENTALAGENCY & RENTAL**

Query Report:

```
SELECT CUS.CUSTOMERID, FIRSTNAME ||' '|| LASTNAME "FULL NAME", CUS.ADDRESSLINE1 ||'
'||CUS.CITY|| ' ' || CUS.STATECODE || ' ' || CUS.ZIPCODE ||' ' || CUS.COUNTRY AS "Mailing Address",
CUS.PHONE, CUS.EMAIL, RA.RENTALAGENCYNAME, COUNT(RA.RENTALAGENCYNAME) AS "# OF TIMES
RENTED FROM" FROM CUSTOMER CUS

RIGHT JOIN RENTAL ON RENTAL.CUSTOMERID = CUS.CUSTOMERID

LEFT JOIN RENTALAGENCY RA ON RA.RENTALAGENCYID = RENTAL.RENTALAGENCYID

WHERE CUS.CUSTOMERTYPE = 'R'

GROUP BY CUS.CUSTOMERID, RA.RENTALAGENCYID, FIRSTNAME ||' '|| LASTNAME, CUS.ADDRESSLINE1 ||'
'||CUS.CITY|| ' ' || CUS.STATECODE || ' ' || CUS.ZIPCODE ||' ' || CUS.COUNTRY, CUS.PHONE,

CUS.EMAIL, RA.RENTALAGENCYNAME

ORDER BY COUNT(RA.RENTALAGENCYNAME) DESC;
```

Results of Execution:

CUSTOMERID	FULL NAME	Mailing Address	PHONE	EMAIL	RENTALAGENCYNAME	# OF TIMES RENTED FROM
1	TYLER Walker	Example Address1234 Brooklyn NY 11236 United States	9292002258	walktyler25@gmail.com	GARSHAWN CAR RENTALS	2
8	JAKE NULL	SOMEADDRESS SOMEWHERE BROOKYLN NY 11204 UNITED STATES	1113334444	NULLJAKE@mail.com	UNIVERSAL RENTALS	2
1	TYLER Walker	Example Address1234 Brooklyn NY 11236 United States	9292002258	walktyler25@gmail.com	ALPHA CAR COMPANY	1
8	JAKE NULL	SOMEADDRESS SOMEWHERE BROOKYLN NY 11204 UNITED STATES	1113334444	NULLJAKE@mail.com	GARSHAWN CAR RENTALS	1
1	TYLER Walker	Example Address1234 Brooklyn NY 11236 United States	9292002258	walktyler25@gmail.com	ZAWA CAR RENTALS	1
3	Alex Rodington	Another Example Address2468 Queens NY 11289 United States	7182002258	alexrod5@gmail.com	ALPHA CAR COMPANY	1

REPORT #4

DESIGN GUIDELINES	ANSWERS
1) Business Scenario & Objectives	To establish which car rental agency is underperforming and needing to be closed temporarily.
2) Target Persona/Decision Maker	EZRental Executives / Managerial Staff
3) What is the Decision to be made?	The Executives of EZRental needs to find the car rental agency that generates the least profit within its lifetime.
4) What Data is required to make the Decision?	The rental agency's name, total rental agency's generated revenue =

	total deposits, total insurance, category daily rental rate vehicle * days rented. Lastly the time frame from the first rental until the most recent car rental
5) Identify Tables that Contain the Data Identified in # 4 & idea of the SELECT Query required	The following Tables in the database schema were analyzed and identified to contain the required data: ❖ RENTALAGENCY – RENTALAGENCYNAME ❖ VEHICLERENTALCATEGORY - CATEGORYDAILYRENTALRATE ❖ RENTAL – RENTALDEPOSIT ❖ RENTALINSURANCEOPTION – INSURANCEOPTIONADDITIONALCOST We would need a SELECT statement that JOINS and queries these five tables: **VEHICLERENTALCATEGORY, RENTALAGENCY, RENTALINSURANCEOPTION, VEHICLE & RENTAL**

Query Report:

```
SELECT RA.RENTALAGENCYNAME, SUM(RENTAL.RENTALDEPOSIT + RIO.INSURANCEOPTIONADDITIONALCOST
+ (VRC.CATEGORYDAILYRENTALRATE * ABS((RENTAL.RENTALPICKUPDATE - RENTAL.RENTALDROPOFFDATE))))
AS "GENERATED REVENUE", MIN(RENTAL.RENTALPICKUPDATE)||' THROUGH '||
MAX(RENTAL.RENTALPICKUPDATE) AS "TIMESPAN" FROM RENTAL

LEFT JOIN RENTALAGENCY RA ON RA.RENTALAGENCYID = RENTAL.RENTALAGENCYID

LEFT JOIN VEHICLE V ON V.VEHICLEID = RENTAL.VEHICLEID
```

```
LEFT JOIN VEHICLERENTALCATEGORY VRC ON VRC.VEHICLERENTALCATEGORYID =
V.VEHICLERENTALCATEGORYID

LEFT JOIN RENTALINSURANCEOPTION RIO ON RIO.INSURANCEOPTIONID = RENTAL.INSURANCEOPTIONID

GROUP BY RA.RENTALAGENCYNAME

ORDER BY "GENERATED REVENUE" DESC;
```

Results of Execution:

RENTALAGENCYNAME	GENERATED REVENUE	TIMESPAN
UNIVERSAL RENTALS	5749.57	30-MAY-22 THROUGH 10-JUL-22
ALPHA CAR COMPANY	5331.62	06-JAN-22 THROUGH 02-APR-22
GARSHAWN CAR RENTALS	4395.7	01-JAN-22 THROUGH 10-MAY-22
ZAWA CAR RENTALS	2725.81	09-JAN-22 THROUGH 15-MAR-22
FAST TRACK	1365.89	03-JAN-22 THROUGH 03-JAN-22

REPORT #5

DESIGN GUIDELINES	ANSWERS
1) Business Scenario & Objectives	To establish which vehicles are being rented the most, thus increasing the average rental deposit by 10% due to their popularity.
2) Target Persona/Decision Maker	EZRental Executives
3) What is the Decision to be made?	The Executives of EZRental needs to decide which vehicles are of most popularity to decide their rental deposit increase.
4) What Data is required to make the Decision?	The Vehicle's VINNUMBER for identification, MAKE, MODEL, and its average daily rental rate. As well as Rental data which tells us

	which vehicles are popular enough to justify this increase. And rental deposit fees spent on that vehicle.
5) Identify Tables that Contain the Data Identified in # 4 & idea of the SELECT Query required	The following Tables in the database schema were analyzed and identified to contain the required data: ❖ VEHICLE – VINNUMBER, MAKE, MODEL ❖ RENTAL – RENTALDEPOSIT We would need a SELECT statement that JOINS and queries these three tables: **VEHICLERENTALCATEGORY, VEHICLE & RENTAL**

Query Report:

```
SELECT VINNUMBER, MAKE, MODEL, AVG(RENTALDEPOSIT) "AVERAGE RENTAL DEPOSIT", COUNT(*) "# OF TIMES RENTED" FROM VEHICLE V

LEFT JOIN RENTAL ON RENTAL.VEHICLEID = V.VEHICLEID

GROUP BY V.VEHICLEID, MAKE, MODEL, VINNUMBER

HAVING AVG(RENTALDEPOSIT) IS NOT NULL

ORDER BY "# OF TIMES RENTED" DESC, AVG(RENTALDEPOSIT) DESC;
```

Results of Execution:

VINNUMBER	MAKE	MODEL	AVERAGE RENTAL DEPOSIT	# OF TIMES RENTED
KMHEC4A46DA063184	Volkswagen	Arteon	300	2
2C3CCARG6DH634610	Dodge	Challenger	300	2
1FTYR14V71TA45330	Nissan	Altima	400	1
3FAHP0JG0AR435957	Dodge	Durango	400	1
JF2SHADC0CH473478	Dodge	Charger	300	1
1FT7X2B6XCEB45481	Nissan	Skyline	290	1
YV1LS56D0Y2607105	Honda	Accord	280	1
2T3BFREV8EW281337	Volkswagen	Arteon	150	1
1GTN2TEA7CZ340054	Chevrolet	Monza	150	1

Business Reports Stored Procedures & Summary

STORED PROCEDURE FOR REPORT #1

DESIGN GUIDELINES	ANSWERS
1) Business Scenario & Objectives	To provide vehicle maintenance for each rental agency service with cars who's mileage ranges between 10,000 to 35,000 miles for precautionary tune ups & repairs.
2) Target Persona/Decision Maker	Vehicle Maintenance & Vehicle Inventory managers
3) What is the Decision to be made?	The Vehicle Maintenance Crew needs to decide which vehicles need tune ups and repairs. Vehicle Inventory managers need to know which vehicles need to be set to unavailable due to this maintenance.
4) What Data is required to make the Decision?	The name of the rental agency that owns the car to notify them of the repairs that need to take action, the vehicles vin numbers for identification on receival, the vehicle's make and model for part ordering purposes, the vehicles mileage, the vehicles current status (Directed towards Vehicle inventory managers responsible for changing the status once they've confirmed it was sent and received for repair)

Stored Procedure:

```
CREATE OR REPLACE PROCEDURE getUnmaintainedVehicles
IS
   v_RENTALAGENCYNAME VARCHAR(50);
```

```
   v_VINNUMBER VARCHAR2(17);
   v_MAKE VARCHAR2(30);
   v_MODEL VARCHAR2(30);
   MILEAGE NUMBER(6);
   v_VEHICLESTATUSDESC VARCHAR2(30);
   CURSOR cur_maint IS
   SELECT RA.RENTALAGENCYNAME, VINNUMBER, MAKE, MODEL, MILEAGE, VEHICLESTATUSDESC FROM VEHICLE
V
   LEFT JOIN RENTALAGENCY RA ON RA.RENTALAGENCYID = V.VEHICLEOWNINGAGENCYID
   LEFT JOIN VEHICLESTATUS VS ON VS.VEHICLESTATUSID = V.VEHICLESTATUSID
   WHERE MILEAGE BETWEEN 10000 AND 35000
   ORDER BY RA.RENTALAGENCYNAME;
BEGIN
   OPEN cur_maint;
   LOOP
      FETCH cur_maint INTO v_RENTALAGENCYNAME, v_VINNUMBER, v_MAKE, v_MODEL, MILEAGE,
v_VEHICLESTATUSDESC;
      EXIT WHEN cur_maint%NOTFOUND;
      DBMS_OUTPUT.PUT_LINE('VIN: ' || v_VINNUMBER || ' MAKE: ' || v_MAKE || ' MODEL: ' || v_MODEL || '
CURRENT MILEAGE: ' ||
      MILEAGE || ' VEHICLE STATUS: ' || v_VEHICLESTATUSDESC || ' OWNING AGENCY: ' ||
v_RENTALAGENCYNAME);
   END LOOP;
   CLOSE cur_maint;
END getUnmaintainedVehicles;
```

Results of Execution:

```
Procedure GETUNMAINTAINEDVEHICLES compiled

VIN: YV1RH59H432287819 MAKE: Chrysler MODEL: 300 CURRENT MILEAGE: 33155 VEHICLE STATUS: Rented. OWNING AGENCY: ALPHA CAR COMPANY
VIN: 5VPWB36N7D3078709 MAKE: Chrysler MODEL: 300 CURRENT MILEAGE: 10010 VEHICLE STATUS: Available. OWNING AGENCY: FAST TRACK
VIN: 1FMEU15H8PLA37915 MAKE: Nissan MODEL: Maxima CURRENT MILEAGE: 31095 VEHICLE STATUS: Reserved. OWNING AGENCY: FAST TRACK
VIN: 2T3ZFREV6DW035111 MAKE: Jeep MODEL: Cherokee CURRENT MILEAGE: 32130 VEHICLE STATUS: Available. OWNING AGENCY: FAST TRACK
VIN: JM1BK323441184063 MAKE: Nissan MODEL: Altima CURRENT MILEAGE: 21410 VEHICLE STATUS: Reserved. OWNING AGENCY: UNIVERSAL RENTALS
VIN: 1HGCP2F75AA036501 MAKE: Jeep MODEL: Cherokee CURRENT MILEAGE: 13630 VEHICLE STATUS: Transferred to another agency OWNING AGENCY: ZAWA CAR RENTALS


PL/SQL procedure successfully completed.
```

STORED PROCEDURE FOR REPORT #2

DESIGN GUIDELINES	ANSWERS
1) Business Scenario & Objectives	A promotional campaign to target customers who has rented 3 or more vehicles in a 4-month span of their first rental. Giving these customers a physical coupon and a digital discount through email and text
2) Target Persona/Decision Maker	Marketing team
3) What is the Decision to be made?	The Marketing team needs to decide which customers to email the discount to and from which car rental agency do they receive said discount for. This is decided by retail customers who has rented 3 or more vehicles in a 4-month span since their first rental.
4) What Data is required to make the Decision?	Customer Name, Customer's Primary Address, Customer's phone number, Customer's Email. First rental pick up date, Total cars rented Time until discount (4 months after first rental)
5) Identify Tables that Contain the Data Identified in # 4 & idea of the SELECT Query required	The following Tables in the database schema were analyzed and identified to contain the required data: ❖ CUSTOMER – FIRSTNAME, LASTNAME, ADDRESSLINE1, CITY, STATECODE, ZIPCODE,

	COUNTRY, PHONE, EMAIL, ❖ RENTALAGENCY – RENTALAGENCYNAME ❖ RENTAL – RENTALPICKUPDATE We would need a SELECT statement that JOINS and queries these two tables: **CUSTOMER & RENTAL**

Stored Procedure:

```
CREATE OR REPLACE PROCEDURE getRetailCustomerTimespan(p_months NUMBER)
IS
   V_FIRSTNAME VARCHAR2(30);
   V_LASTNAME VARCHAR2(30);
   V_ADDRESSLINE1 VARCHAR2(75);
   V_CITY VARCHAR2(15);
   V_STATECODE CHAR(2);
   V_ZIPCODE VARCHAR2(10);
   V_COUNTRY VARCHAR2(50);
   V_PHONE VARCHAR2(11);
   V_EMAIL VARCHAR2(75);
   V_MINRENTALPICKUPDATE DATE;
   V_TOTALCARSRENTED NUMBER;
   V_MONTH_SPAN DATE;
   CURSOR cur_timespan is
   SELECT  FIRSTNAME, LASTNAME, CUS.ADDRESSLINE1, CUS.CITY, CUS.STATECODE,
CUS.ZIPCODE, CUS.COUNTRY, CUS.PHONE, CUS.EMAIL,
   MIN(RENTAL.RENTALPICKUPDATE), count(*),
ADD_MONTHS(MIN(RENTAL.RENTALPICKUPDATE) ,  p_months) FROM RENTAL
   LEFT JOIN CUSTOMER CUS ON CUS.CUSTOMERID = RENTAL.CUSTOMERID
   GROUP BY CUS.CUSTOMERID, FIRSTNAME, LASTNAME, CUS.ADDRESSLINE1, CUS.CITY,
CUS.STATECODE, CUS.ZIPCODE, CUS.COUNTRY, CUS.PHONE, CUS.EMAIL
   HAVING COUNT(*) >= 3
   ORDER BY CUS.CUSTOMERID;
```

```
BEGIN
    OPEN cur_timespan;
    LOOP
    FETCH cur_timespan INTO V_FIRSTNAME, V_LASTNAME, V_ADDRESSLINE1, V_CITY,
V_STATECODE, V_ZIPCODE, V_COUNTRY, V_PHONE, V_EMAIL, V_MINRENTALPICKUPDATE,
    V_TOTALCARSRENTED, V_MONTH_SPAN;
    EXIT WHEN cur_timespan%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE('Full Name: ' || V_FIRSTNAME || ' ' || V_LASTNAME || ' Mailing
Address: ' || V_ADDRESSLINE1 || ' ' || V_CITY
    || ' ' || V_STATECODE || ' ' || V_ZIPCODE || ' ' || V_COUNTRY || ' Phone #: ' || V_PHONE || ' Email
Address: ' || V_EMAIL || ' First Rental Pick Up Date: ' ||
    ' Total Cars Rented: ' || V_TOTALCARSRENTED || ' ' || V_MONTH_SPAN || ' Month span');
  END LOOP;
  CLOSE cur_timespan;
END getRetailCustomerTimespan;
```

Results of Execution:

```
Procedure GETRETAILCUSTOMERTIMESPAN compiled

Full Name: TYLER Walker Mailing Address: Example Address1234 Brooklyn NY 11236 United States Phone #: 9292002258 Email Address: walktyler25@gmail.com First Rental Pick Up Date:  Total Cars Rented: 4 01-MAY-22 Month span
Full Name: JAKE NULL Mailing Address: SOMEADDRESS SOMEWHERE BROOKYLN NY 11204 UNITED STATES Phone #: 1113334444 Email Address: NULLJAKE@mail.com First Rental Pick Up Date:  Total Cars Rented: 3 10-SEP-22 Month span


PL/SQL procedure successfully completed.
```

STORED PROCEDURE FOR REPORT #3

DESIGN GUIDELINES	ANSWERS
1) Business Scenario & Objectives	A promotional campaign to reward retail customers who frequently rents from the same car rental agency twice or more. Giving these customers a physical coupon and a digital discount through email and text. The discount will only be given to those that frequent a car rental service.

2) Target Persona/Decision Maker	Marketing team
3) What is the Decision to be made?	The Marketing team needs to decide which retail customers to email the discount to, and from which car rental agency do they receive said discount for. They also need to be able to determine which car rental agency they frequent the most.
4) What Data is required to make the Decision?	Customer Name, Customer's Primary Address, Customer's phone number, Customer's Email. Rental Agency Name
5) Identify Tables that Contain the Data Identified in # 4 & idea of the SELECT Query required	The following Tables in the database schema were analyzed and identified to contain the required data: ❖ CUSTOMER – FIRSTNAME, LASTNAME, ADDRESSLINE1, CITY, STATECODE, ZIPCODE, COUNTRY, PHONE, EMAIL, ❖ RENTALAGENCY – RENTALAGENCYNAME We would need a SELECT statement that JOINS and queries these two tables: **CUSTOMER & RENTALAGENCY & RENTAL**

Stored Procedure:

```
CREATE OR REPLACE PROCEDURE getRetailCustFrequents
IS
   V_CUSTOMERID NUMBER;
```

```
   V_FIRSTNAME VARCHAR2(30);
   V_LASTNAME VARCHAR2(30);
   V_ADDRESSLINE1 VARCHAR2(75);
   V_CITY VARCHAR2(15);
   V_STATECODE CHAR(2);
   V_ZIPCODE VARCHAR2(10);
   V_COUNTRY VARCHAR2(50);
   V_PHONE VARCHAR2(11);
   V_EMAIL VARCHAR2(75);
   V_RENTALAGENCYNAME VARCHAR2(50);
   V_TIMESRENTED NUMBER;


   CURSOR cur_frequent is
      SELECT FIRSTNAME, LASTNAME, CUS.ADDRESSLINE1, CUS.CITY, CUS.STATECODE,
CUS.ZIPCODE, CUS.COUNTRY, CUS.PHONE, CUS.EMAIL, RA.RENTALAGENCYNAME,
COUNT(RA.RENTALAGENCYNAME) FROM CUSTOMER CUS
   RIGHT JOIN RENTAL ON RENTAL.CUSTOMERID = CUS.CUSTOMERID
   LEFT JOIN RENTALAGENCY RA ON RA.RENTALAGENCYID = RENTAL.RENTALAGENCYID
   WHERE CUS.CUSTOMERTYPE = 'R'
   GROUP BY CUS.CUSTOMERID, FIRSTNAME, LASTNAME, CUS.ADDRESSLINE1, CUS.CITY,
CUS.STATECODE, CUS.ZIPCODE, CUS.COUNTRY, CUS.PHONE, CUS.EMAIL,
RA.RENTALAGENCYNAME
   ORDER BY COUNT(RA.RENTALAGENCYNAME) DESC;


BEGIN
      OPEN cur_frequent;
      LOOP
      FETCH cur_frequent INTO V_FIRSTNAME, V_LASTNAME, V_ADDRESSLINE1, V_CITY,
V_STATECODE, V_ZIPCODE, V_COUNTRY, V_PHONE, V_EMAIL, V_RENTALAGENCYNAME,
      V_TIMESRENTED;
      EXIT WHEN cur_frequent%NOTFOUND;
      DBMS_OUTPUT.PUT_LINE('Full Name: ' || V_FIRSTNAME || ' ' || V_LASTNAME || ' Mailing
Address: ' || V_ADDRESSLINE1 || ' ' || V_CITY
      || ' ' || V_STATECODE || ' ' || V_ZIPCODE || ' ' || V_COUNTRY || ' Phone #: ' || V_PHONE || ' Email
Address: ' || V_EMAIL || ' Rental Agency Name: ' ||
      V_RentalAgencyName || ' # of Times Rented From: ' || V_TIMESRENTED );
```

```
    END LOOP;
    CLOSE cur_frequent;
END getRetailCustFrequents;
```

Results of Execution:

```
Procedure GETRETAILCUSTFREQUENTS compiled

Full Name: TYLER Walker Mailing Address: Example Address1234 Brooklyn NY 11236 United States Phone #: 9292002258 Email Address: walktyler25@gmail.com Rental Agency Name: GARSHAWN CAR RENTALS # of Times Rented From: 2
Full Name: JAKE NULL Mailing Address: SOMEADDRESS SOMEWHERE BROOKYLN NY 11204 UNITED STATES Phone #: 1113334444 Email Address: NULLJAKE@mail.com Rental Agency Name: UNIVERSAL RENTALS # of Times Rented From: 2
Full Name: TYLER Walker Mailing Address: Example Address1234 Brooklyn NY 11236 United States Phone #: 9292002258 Email Address: walktyler25@gmail.com Rental Agency Name: ALPHA CAR COMPANY # of Times Rented From: 1
Full Name: Alex Rodington Mailing Address: Another Example Address2468 Queens NY 11289 United States Phone #: 7182002258 Email Address: alexrod5@gmail.com Rental Agency Name: ALPHA CAR COMPANY # of Times Rented From: 1
Full Name: TYLER Walker Mailing Address: Example Address1234 Brooklyn NY 11236 United States Phone #: 9292002258 Email Address: walktyler25@gmail.com Rental Agency Name: ZAWA CAR RENTALS # of Times Rented From: 1
Full Name: JAKE NULL Mailing Address: SOMEADDRESS SOMEWHERE BROOKYLN NY 11204 UNITED STATES Phone #: 1113334444 Email Address: NULLJAKE@mail.com Rental Agency Name: GARSHAWN CAR RENTALS # of Times Rented From: 1


PL/SQL procedure successfully completed.
```

STORED PROCEDURE FOR REPORT #4

DESIGN GUIDELINES	ANSWERS
1) Business Scenario & Objectives	To establish which car rental agency is underperforming and needing to be closed temporarily.
2) Target Persona/Decision Maker	EZRental Executives / Managerial Staff
3) What is the Decision to be made?	The Executives of EZRental needs to find the car rental agency that generates the least profit within its lifetime.
4) What Data is required to make the Decision?	The rental agency's name, total rental agency's generated revenue = total deposits, total insurance, category daily rental rate vehicle * days rented. Lastly the time frame from the first rental until the most recent car rental
5) Identify Tables that Contain the Data Identified in # 4 & idea of the SELECT Query required	The following Tables in the database schema were analyzed and identified to contain the required data: ❖ RENTALAGENCY – RENTALAGENCYNAME

	❖ VEHICLERENTALCATEGORY - CATEGORYDAILYRENTALRATE ❖ RENTAL – RENTALDEPOSIT ❖ RENTALINSURANCEOPTION – INSURANCEOPTIONADDITIONALCOST We would need a SELECT statement that JOINS and queries these five tables: **VEHICLERENTALCATEGORY, RENTALAGENCY, RENTALINSURANCEOPTION, VEHICLE & RENTAL**

Stored Procedure:

```
CREATE OR REPLACE PROCEDURE getTotalAgencyRevenue

IS

   V_RENTALAGENCYNAME VARCHAR2(50);

   V_RENTALDEPOSIT NUMBER(8,2);

   V_INSURANCEOPT NUMBER(6,2);

   V_CATEGORYDAILYRENTALRATE NUMBER(5,2);

   V_RENTALPICKUPDATE DATE;

   V_RENTALDROPOFFDATE DATE;

   V_MINRENTALPICKUPDATE DATE;

   V_MAXRENTALPICKUPDATE DATE;

   TotalRev NUMBER(8,2);


   CURSOR cur_exec is

        SELECT RA.RENTALAGENCYNAME, SUM(RENTALDEPOSIT +
RIO.INSURANCEOPTIONADDITIONALCOST + (VRC.CATEGORYDAILYRENTALRATE *
```

```
ABS((RENTAL.RENTALPICKUPDATE - RENTAL.RENTALDROPOFFDATE)))) AS "GENERATED REVENUE" ,
MIN(RENTAL.RENTALPICKUPDATE), MAX(RENTAL.RENTALPICKUPDATE)  FROM RENTAL

   LEFT JOIN RENTALAGENCY RA ON RA.RENTALAGENCYID = RENTAL.RENTALAGENCYID

   LEFT JOIN VEHICLE V ON V.VEHICLEID = RENTAL.VEHICLEID

   LEFT JOIN VEHICLERENTALCATEGORY VRC ON VRC.VEHICLERENTALCATEGORYID =
V.VEHICLERENTALCATEGORYID

   LEFT JOIN RENTALINSURANCEOPTION RIO ON RIO.INSURANCEOPTIONID =
RENTAL.INSURANCEOPTIONID

   GROUP BY RA.RENTALAGENCYNAME

   ORDER BY "GENERATED REVENUE" DESC;


BEGIN

      OPEN cur_exec;

      LOOP

      FETCH cur_exec INTO V_RENTALAGENCYNAME, TotalRev, V_MINRENTALPICKUPDATE,
V_MAXRENTALPICKUPDATE;

      EXIT WHEN cur_exec%NOTFOUND;

      DBMS_OUTPUT.PUT_LINE('Rental Agency Name: ' || V_RENTALAGENCYNAME || ' Total Revenue
Generated: ' || TotalRev || ' From:' || V_MINRENTALPICKUPDATE || ' To:' ||
V_MAXRENTALPICKUPDATE );

   END LOOP;

   CLOSE cur_exec;

END getTotalAgencyRevenue;
```

Results of Execution:

```
Procedure GETTOTALAGENCYREVENUE compiled

Rental Agency Name: UNIVERSAL RENTALS Total Revenue Generated: 5749.57 From:30-MAY-22 To:10-JUL-22
Rental Agency Name: ALPHA CAR COMPANY Total Revenue Generated: 5331.62 From:06-JAN-22 To:02-APR-22
Rental Agency Name: GARSHAWN CAR RENTALS Total Revenue Generated: 4395.7 From:01-JAN-22 To:10-MAY-22
Rental Agency Name: ZAWA CAR RENTALS Total Revenue Generated: 2725.81 From:09-JAN-22 To:15-MAR-22
Rental Agency Name: FAST TRACK Total Revenue Generated: 1365.89 From:03-JAN-22 To:03-JAN-22


PL/SQL procedure successfully completed.
```

STORED PROCEDURE FOR REPORT #5

DESIGN GUIDELINES	ANSWERS
1) Business Scenario & Objectives	To establish which vehicles are being rented the most, thus increasing the average rental deposit by 10% due to their popularity.
2) Target Persona/Decision Maker	EZRental Executives
3) What is the Decision to be made?	The Executives of EZRental needs to decide which vehicles are of most popularity to decide their rental deposit increase.
4) What Data is required to make the Decision?	The Vehicle's VINNUMBER for identification, MAKE, MODEL, and its average daily rental rate. As well as Rental data which tells us which vehicles are popular enough to justify this increase. And rental deposit fees spent on that vehicle.
5) Identify Tables that Contain the Data Identified in # 4 & idea of the SELECT Query required	The following Tables in the database schema were analyzed and identified to contain the required data:

	❖ VEHICLE – VINNUMBER, MAKE, MODEL ❖ RENTAL – RENTALDEPOSIT We would need a SELECT statement that JOINS and queries these three tables: **VEHICLERENTALCATEGORY, VEHICLE & RENTAL**

Stored Procedure:

```
CREATE OR REPLACE PROCEDURE getCarUsage
IS
   V_VINNUMBER VARCHAR2(17);
   V_MAKE VARCHAR2(30);
   V_MODEL VARCHAR2(30);
   V_avgRENTALDEPOSIT NUMBER;
   V_TIMESRENTED NUMBER;


   CURSOR cur_exec is
        SELECT VINNUMBER, MAKE, MODEL, AVG(RENTALDEPOSIT) "AVERAGE RENTAL DEPOSIT", COUNT(*) "# OF TIMES RENTED" FROM VEHICLE V
LEFT JOIN RENTAL ON RENTAL.VEHICLEID = V.VEHICLEID
GROUP BY V.VEHICLEID, MAKE, MODEL, VINNUMBER
HAVING AVG(RENTALDEPOSIT) IS NOT NULL
ORDER BY "# OF TIMES RENTED" DESC, AVG(RENTALDEPOSIT) DESC;


BEGIN
      OPEN cur_exec;
      LOOP
      FETCH cur_exec INTO V_VINNUMBER, V_MAKE, V_MODEL, V_avgRENTALDEPOSIT, V_TIMESRENTED;
      EXIT WHEN cur_exec%NOTFOUND;
```

```
    DBMS_OUTPUT.PUT_LINE('VIN: ' || V_VINNUMBER || ' MAKE: ' || V_MAKE || ' MODEL: ' || V_MODEL || '
Average Rental Deposit:' || V_avgRENTALDEPOSIT || ' # Of Times Rented: ' || V_TIMESRENTED);

  END LOOP;

  CLOSE cur_exec;

END getCarUsage;
```

Results of Execution:

```
Procedure GETCARUSAGE compiled


VIN: KMHEC4A46DA063184 MAKE: Volkswagen MODEL: Arteon Average Rental Deposit:300 # Of Times Rented: 2
VIN: 2C3CCARG6DH634610 MAKE: Dodge MODEL: Challenger Average Rental Deposit:300 # Of Times Rented: 2
VIN: 1FTYR14V71TA45330 MAKE: Nissan MODEL: Altima Average Rental Deposit:400 # Of Times Rented: 1
VIN: 3FAHP0JG0AR435957 MAKE: Dodge MODEL: Durango Average Rental Deposit:400 # Of Times Rented: 1
VIN: JF2SHADC0CH473478 MAKE: Dodge MODEL: Charger Average Rental Deposit:300 # Of Times Rented: 1
VIN: 1FT7X2B6XCEB45481 MAKE: Nissan MODEL: Skyline Average Rental Deposit:290 # Of Times Rented: 1
VIN: YV1LS56D0Y2607105 MAKE: Honda MODEL: Accord Average Rental Deposit:280 # Of Times Rented: 1
VIN: 2T3BFREV8EW281337 MAKE: Volkswagen MODEL: Arteon Average Rental Deposit:150 # Of Times Rented: 1
VIN: 1GTN2TEA7CZ340054 MAKE: Chevrolet MODEL: Monza Average Rental Deposit:150 # Of Times Rented: 1


PL/SQL procedure successfully completed.
```

Conclusion

It can be well concluded that the point-of-sale system implemented by NYC-Tech Solutions Inc. allows for both retail and corporate customers to reserve and rent the vehicles of their choice.

With these added System procedures, It is now assisting EZRental Inc employees giving them added capabilities and functionalities. EZRental Inc. is now capability of displaying all necessary business procedures to help track important information pertaining their multiple establishments, employees, customers, and assets .