3/9/2022

EZRental Auto Rental POS Management System Database Design and Implementation

Edras Hernandez
CST3604
PROFESSOR RODRIGUEZ

Table of Contents *Page#*

EZRental Auto Rental POS Management System Database Design and Implementation

EXECUTIVE SUMMARY:

The purpose of EZ AutoRental has hired NYC-Tech Solutions is because their rental system is outdated. Therefore, NYC-TECH solutions has hired a CONSULTANT APPLICATION & DATABASE DEVELOPER to implement and code the application. The plan is to use the two most popular Project Management Methodologies used in industry: The Waterfall Methodology and Agile Methodology. We plan to deliver the whole project in four phases where EZ AUTORENTAL will have a report of the status of project. Our goal is to meet with every single requirement EZ AutoRental provided.

Problem Statement & Objectives

EZ Rental Inc. *hired* ***NYC-Tech Solutions Inc****., to design & implement suite of* ***Auto Rental Point-of-Sales Management System Applications*** *that included* ***EZ Rental POS*** *intended for* ***Customer Service Representative*** *and other* ***employees*** *in their* ***rental agencies****, in addition to an* ***INTRANET Corporate Website*** *named* ***EZRentalCorp.com*** *intended for business employees in the corporate offices, and finally, an e-commerce* ***INTERNET Website*** *name* ***EZRental.com*** *intended for customers via the internet to make and manage reservations.*

- ***NYC-Tech Solutions Inc****. hired a* ***CONSULTANT APPLICATION & DATABASE DEVELOPER*** *to* ***implement*** *and code the application.*

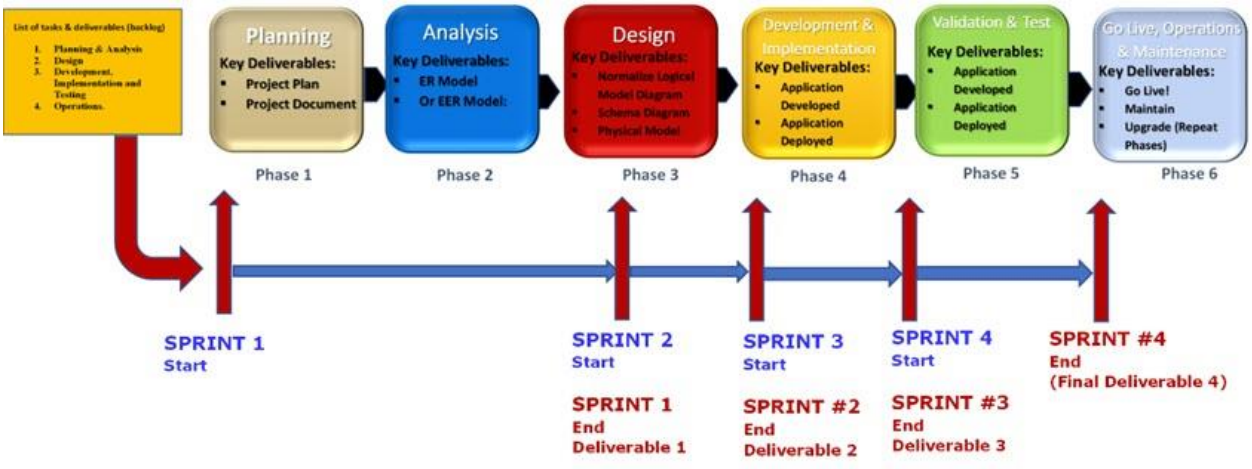
Basic objectives and architecture were targeted:

- *The* ***EZ Rental POS System*** *was designed to allow customers, both retail and corporate, to reserve vehicles for renting like existing in-person or online car rental systems such as* ***Avis, Hertz, Budget,*** *etc.*
- *The application needed to provide the required functionalities for our Customer Service representatives and other front-line workers in their rental agencies, business users in the corporate offices and customers via the internet.*
- *The application had 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 were looking to expand further globally into other markets in Asia, Africa, and the Mediterranean.*

Project Management Methodology

In this project, ***NYC Tech solutions Inc.*** used a project management methodology to design and implement the entire application, client side and database server component of our application. For this **Auto Rental Management System project**, a high-level overview of the following two most popular Project Management Methodologies used in industry:

- **Waterfall Project Management Methodology** – The popular and historical standard.
- **Agile Project Management Methodology** – The new quick methodology becoming the new standard.



As shown is the image above, the system design for Auto Rental Management was split into four deliverables using the waterfall method in which each deliverable had their own process.

Application Business requirements:

Business Requirements

About Us:

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

Current Challenges:

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

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

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

Our Agencies:

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

Our Customers:

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

All Customers

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

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

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

Application Business requirements:

Customers (Cont.):

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

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

Application Business requirements:

Corporate Customers

Corporate Customers are customers who are renting vehicle during business travel and their company have a contract with EZRental Inc. These companies get special corporate rate for their employee's rental services. Therefore, for their corporate customers only, they must store the following attributes/properties: unique ***company ID*** (they have a unique ID number for each company doing business with us), ***company name, company address*** which contains the elements: ***address line1, address line 2*** *(*which is optional and used for apartment number, suite or any additional address information required), ***city, state code, zip code*** (which is the two-character code for a state in the US) & ***country,*** in addition, ***company contact*** which is composed of ***company representative name, contact phone number & contact email*** (unique as all email addresses). And finally, they needed to store the ***company discount percentage rate*** which is the discounted percentage applied to a corporate customers rental. The company Discount percentage rate was stored in the database as a decimal percentage value, for example 20% is stored as 0.02, 30% as 0.03. 50% as 0.05 etc. This discount percentage (0.0x) is applied to the Vehicle Rental Categories which determines the price of each category to determine the total discount. Therefore, when a corporate customer rents a vehicle from a vehicle category (such as economic, compact, standard etc.), this discount percentage is applied to each of the categories during the rental/reservation process. Note that every company has a different percentage rating depending on their contract with EZ-Rentals Inc. For example, some companies have 20% discount towards their rentals, which would be stored as 0.20 in the database, some have 30% (0.03) etc. Vehicle Rental Categories are discussed in more details later in these requirements.

Retail Customers

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

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

Application Business requirements:

Vehicles:

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

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

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

- Other attributes include the ***vehicle name*** composed of ***make, model*** & **year**. Additional attributes are ***color***, also the ***license plate*** composed of the following components: ***license plate number***, ***license plate state***.

- More attributes are ***mileage***, ***transmission type*** of the vehicle. The Transmission Type attribute has business value thus used in reports and in the business processes. The values used for transmission type is a combination of a transmission code and a transmission code description as follows: from transmission type they use a **Semi-automatic** & **dual-clutch)** and

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

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

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

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

Application Business Requirements

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.

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

1. ***A reservation is NOT made for a specific vehicle, but to a vehicle rental category. Rental category examples are economy, intermediate, full size, luxury***.
2. ***Thus, a customer makes a reservation of a vehicle rental category at a rental agency. Therefore, the reservation process involves a customer a vehicle rental category and the rental agency*** where the vehicle will be picked up.

A **Vehicle Rental Category** contains a list of vehicles depending on the vehicle type: Car (economy, intermediate, full size, luxury), SUV (standard, full size etc.), or Cargo Van etc. Each of these categories have a different price range. Therefore, for a vehicle rental category they needed 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. They used a specific code for their 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:

Application Business requirements:

Reservation Process (Cont.):

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

Vehicle Rental Category ID	*Vehicle Rental Category Name*	*Category Daily Rental Rate*
1	Car-Economic	$113.99
2	Car-Compact	$115.99
3	Car-Intermediate	$116.67
4	Car-Standard	$119.99
5	Car-Full Size	$121.99
6	Car-Premium	$127.79
7	Car-Luxury	$139.99
8	SUV-Intermediate	$127.99
9	SUV-Standard	$128.99
10	SUV-Standard Elite	$135.99
11	SUV-Full Size	$148.99
12	SUV-Premium	$157.99
13	Minivan-Standard	$152.99
14	Van-Cargo Van	$19.95
15	Pick Up-Mid Size	$69.95
16	Pick Up-Full Size	$105.99
17	Motorcycle-Touring	$19.95
18	Motorcycle-Cruiser	$199.99
19	Motorcycle-Scooter	$79.95
20	Passenger Van (12 passengers)	$161.00
21	Passenger Shuttle (16 passengers)	$180.00
22	Passenger Shuttle (20 passengers)	$220.00
23	Passenger Mini-Bus (30 passengers)	$250.00
24	Passenger Mini-Bus (40 passengers)	$280.00
25	Passenger Large-Bus (80 passengers)	$300.00

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

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

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

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

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

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

Business Requirements (Cont.)

Reservation Process (Cont.):

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

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

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

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

The Rental Process:

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

For the rental process, the following business rules apply:

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

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

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

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

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

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

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

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

Application Business requirements:

Business Requirements (Cont.)

The Rental Process (Cont.):

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

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

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

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

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

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

Application Business requirements:

Business Requirements (Cont.)

The Rental Process (Cont.):

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

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

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

Our Employees:

EZ-Car Rental 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***. In addition, we need to capture the employee ***social security number***. Below are some business rules and usage for the ***EmployeeID*** and the ***social security number***.

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

Security & Application Access:

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

- Employee user accounts
- Customer user accounts

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

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

Application Business requirements:

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

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

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

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

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

Conclusion:

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

Application Development & Technical Requirements:

The Technical Requirements are the FOUNDATION for the Application Design and used by the Full-Stack Application Architect to engage in the Application Design and OBJECT-ORIENTED DESIGN. In data modeling, from the business requirements we derived the following Diagrams:

- *Application Physical Architecture Design*
- *Application Software Design & Programming.*
- *Application Feature List of AGILE BACKLOG.*
- *User-Interface Wireframe Design & Programming.* *Other Application Related Implementations.*

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 their 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. They wanted to invest in modernizing thier 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.

Application Development & Technical Requirements:

Rental Agency Application Architecture Requirements:

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

1. Client application processing, transaction and response must be fast to minimize service time for a customer.

2. All transaction processing should be done in the user's computer or desktop for fast processing and response.

3. Application Architecture must be reusable and scalable to support future updates and new feature enhancements, without a long development lifecycle.

4. Depending on the architecture NYC-Tech Solutions Inc., decides for the application in the rental agencies (Desktop client or Web client), the primary Application Development Platform we use is ***C# & .NET technologies***. For any Web related development, we support JavaScript, React, NodeJs and other standard Web Technologies. We have aligned ***C#.NET*** & **ASP.NET Web developers** that have been assigned to assist, support and update the application once NYCTech consultants complete the project and development of this system.

5. Rental Agency Desktop Application Security Authentication System – Proper security and authentication must be implemented to make sure only authorized customer service representative and other rental office employees can access the Point-0f-Sales with appropriate conditional access.

Application Development & Technical Requirements:

Application Development & Technical Requirements (Cont.)

Rental Agency Application Features and Functionalities Requirements:

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

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

Rental Agency Application Graphical User Interface Requirements:

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

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 their 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 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 they 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:

Application Development & Technical Requirements (cont'd)

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

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

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

Rental Agency:

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

Application Physical Technical Architecture

After a design meeting with the architects and full-stack developers a decision was made on the application architecture for the EZ Rental POS application.

After a thorough review of both the business requirements and technical requirements. by the project team, resultant decisions on architecture (s) were based on the following:

Rental Agency Employees:

o The system in their agencies used by the customer service representatives or front-line workers, must be able to quickly respond and execute the necessary requests such as

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

o customer reservations, rentals, returns, customer management etc., therefore fast response and performance is required to quickly service a customer and minimize the wait. This is more important in Airports and other high-traffic locations.

o They also wanted to provide to their customer service agents with a rich user-interface experience.

o The system in the agencies is also used by other back-end personnel such as vehicle inventory managers and

administrators, service personnel, vehicle transport drivers, etc. Therefore, the system needed to also perform well.

▪ **Corporate Offices:**

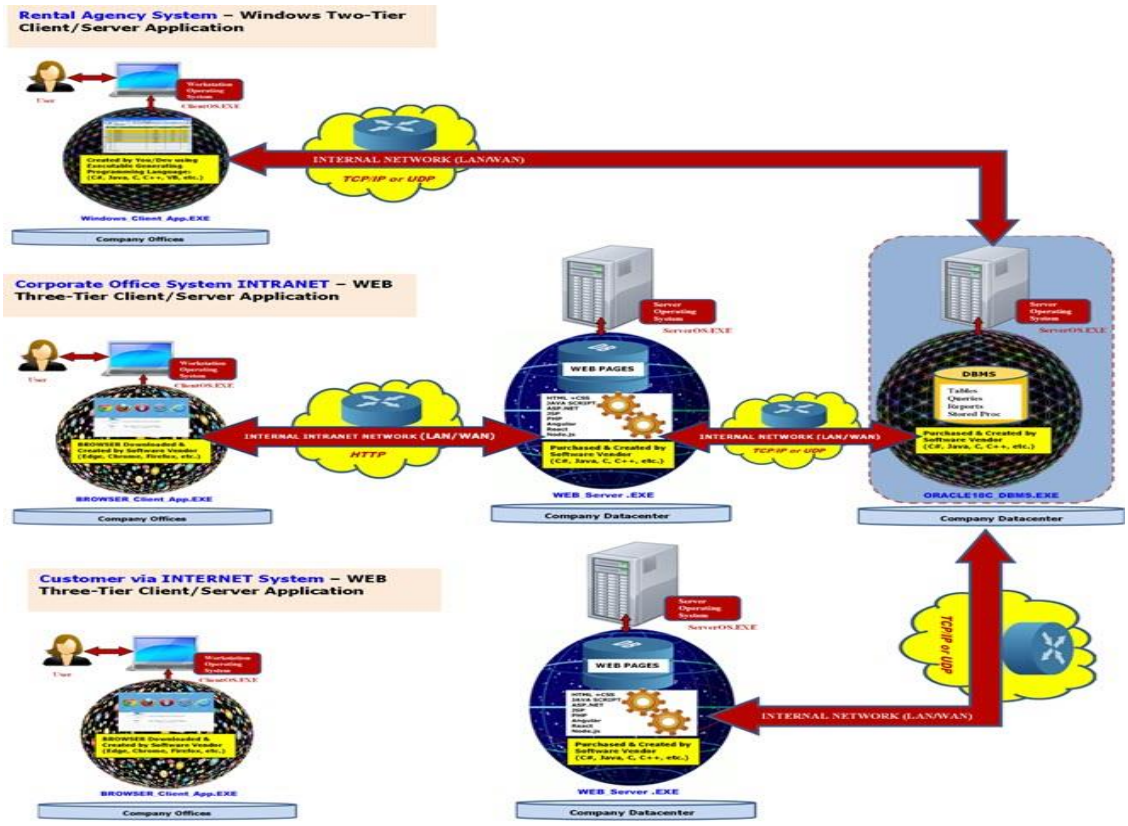
The corporate offices are where their business operations are managed by their business employees & employees at the rental agencies via the INTRANET Web Portal.

These features include

Application Physical Technical Architecture (cont'd)

- **Intranet Web Enterprise Resource Planning Systems (ERP) Portal Feature** such as providing access to **Enterprise Resource Planning Systems (ERP) Applications** such as: **Customer Credit Card Management System**, **Vehicle Inventory Management System**, **Customer Relationship Management (CRM)**, **Human Resource Management System**, & **Finance & Operations System, Marketing System, Customer & Field Service System etc.**

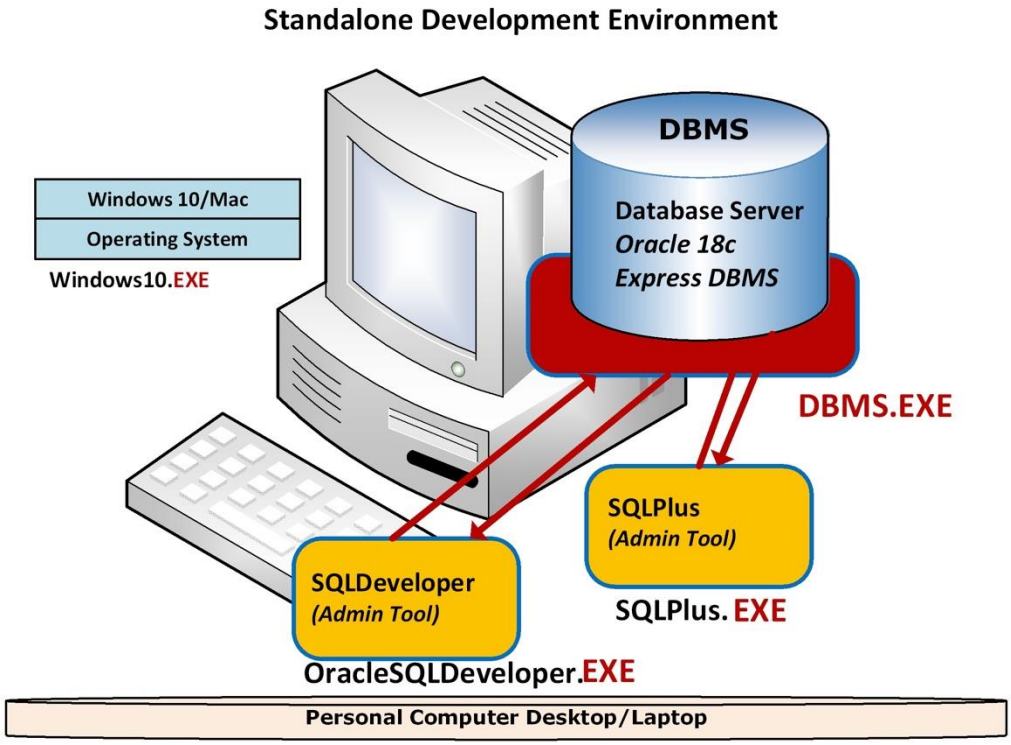
- ***Web EZRental Point-of-Sales Corporate Management Feature*** *which allows employees to manage & execute* ***Point-of-Sales (POS)*** *transaction via the* ***Intranet Web Poral*** *such as:* ***Search Customer Profile Information***, ***Customer Account Management***, ***Customer Registration***, ***Customer Update***, ***Customer Delete***, & ***Customer Listing functionalities, Manage & Make Reservations of a Vehicle***, ***Manage an existing Rental***, ***etc.***



Above is a picture diagram of this multi-component client/server architecture. Note that both the Windows Client Application, the Corporate Office Browser Web Client Applications, and the Customer Internet Browser Web Client Applications are both sharing the same MS SQL Server DBMS Server Application

Database Management System Development Environment & Physical Architecture:

- Database Tier – the Database Management System (DBMS) in scope ***is Oracle Server 18c Express Edition (EX)*** since this is the standard DBMS used at ***EZ Rental Inc***.
- The objectives are to install **Oracle Server 18c Express Edition (EX)** which includes **SQL PLUS command-**



In the image above, we can see the development environment for Oracle Server 18c being installed on the Database administrator machine. The administrator will have access to SQL DEVELOPER AND SQL PLUS.

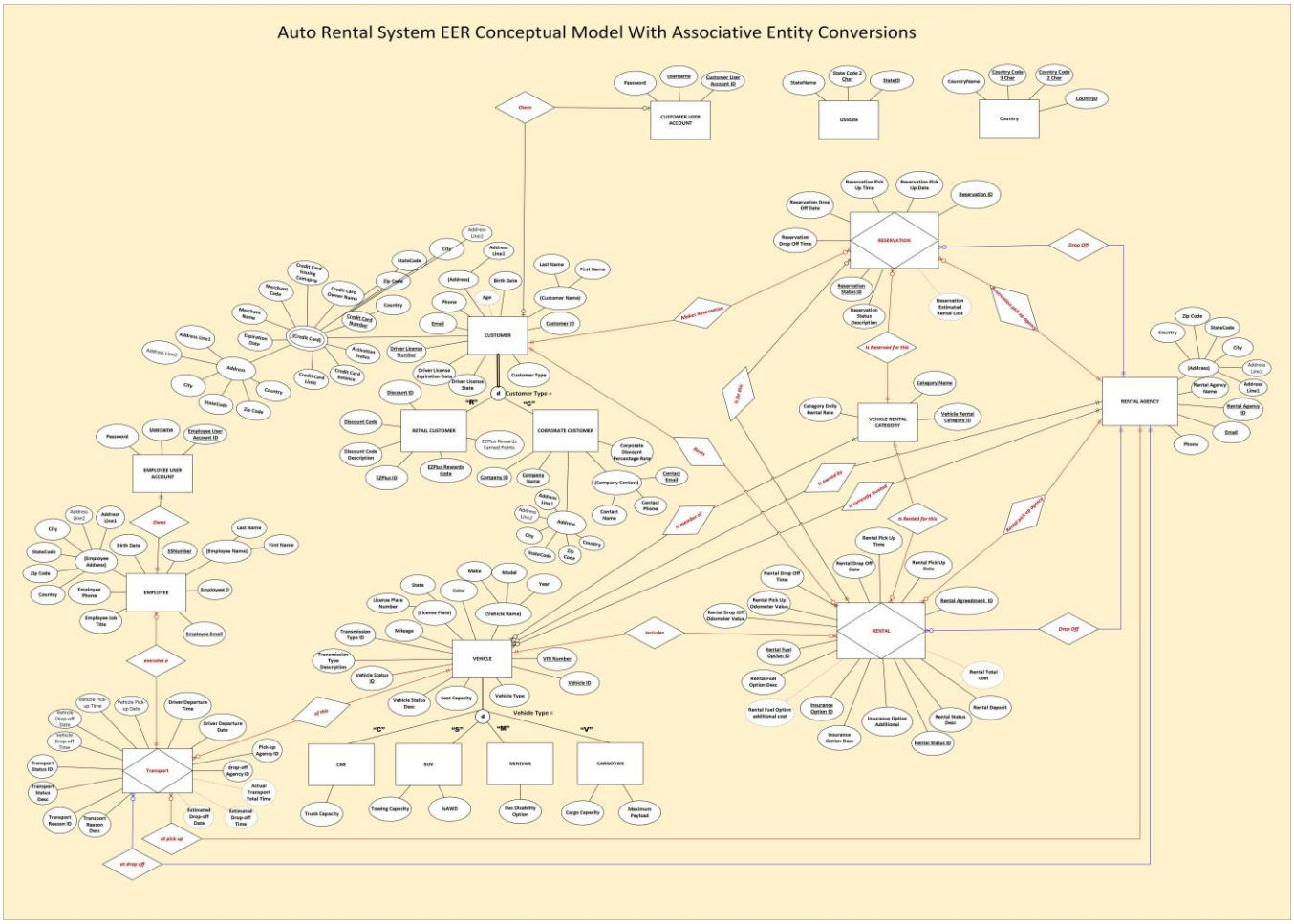
Project Roles & Responsibilities:

Summary of the DBMS Server Application Development Roles and Responsibilities

Our company will assemble the required database development team, and the table below describes each of the roles and the individual (s) that will execute the roles:

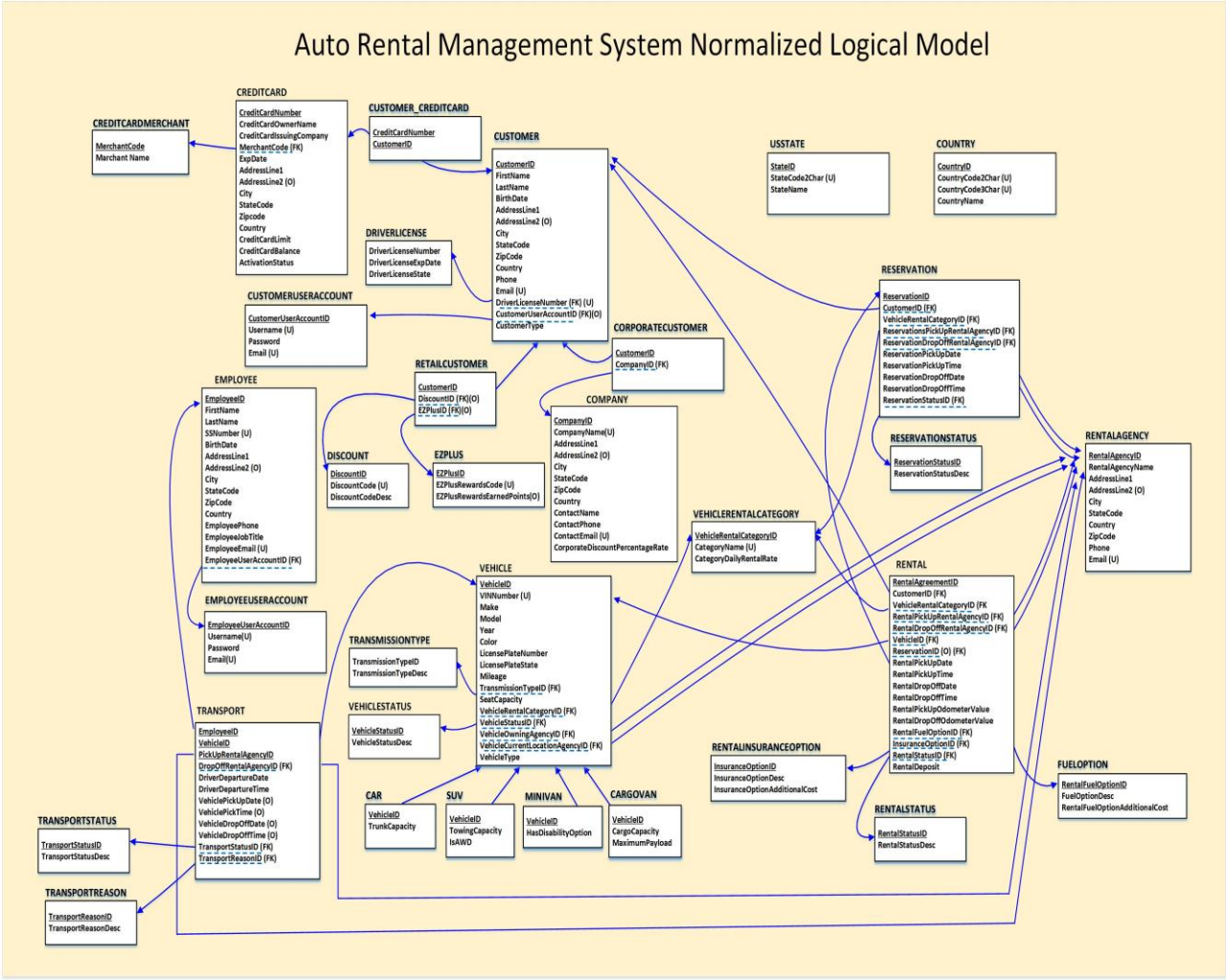
Person	Role	Description
Mr. Rodriguez	**Program Manager & Project Manager**	▪ Owner of the project and liaison to Manage the EZRental Inc., the customer. ▪ Activities include but not limited to. 1. Owner of project responsible for the success of the project. 2. Project Management 3. Scrum Master that 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 **(CST 3604 Edras Hernandez)**	**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 (CST3604 STUDENT – Edras Hernandez)	**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**.

Database Design Deliverable#1–ER/EER Conceptual Model Diagram:



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

Database Design Deliverable #2 – Normalized Logical Model Diagram:



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

Database Design Deliverable #3 – Physical Model Data Dictionary

1.

CREDITCARD							
Column Num.	***Attribute/Column Name***	***Generic Data Type Name***	***Oracle Data Type Name***	***Is it Required?***	***Length/Size /Format***	***Constraints***	***Description/ purpose***
1.	***CreditCardNumber***	*STRING*	***VARCHAR2(16)***	*YES*	*16*	*Primary key*	*Primary key that identifies specific credit card number*
2.	***CreditCardOwnerName***	*STRING*	*VARCHAR2(50)*	*YES*	*50*	*NOT NULL*	*Name of card owner*
3.	***CreditCardIssuingCompany***	*STRING*	*VARCHAR2(20)*	*YES*	*20*	*NOT NULL*	*Company name*
4.	***MerchantCode***	*NUMBER*	*NUMBER(2)*	*YES*	*2*	*Foreign key NOT NULL*	*FK from creditcardMerchant*
5.	***ExpDate***	*DATE*	*DATE*	*YES*	*DD/MON/YY*		*Date when card expires*
6.	***AddressLine1***	*STRING*	*VARCHAR2(30)*	*YES*	*30*	*NOT NULL*	*Address of user*
7.	***AddressLine2***	*STRING*	*VARCHAR2(5)*	*NO*	*5*	*NULL*	*Optional address line 2*
8.	***City***	*STRING*	*VARCHAR2(25)*	*YES*	*50*	*NOT NULL*	*Name of city*
9.	***StateCode***	*CHARACTER*	*CHAR(2)*	*YES*	*2*	*NOT NULL*	*Name of State*
10.	***Zipcode***	*STRING*	*VARCHAR2(10)*	*YES*	*10*	*NOT NULL*	*Zip code area*
11.	***Country***	*STRING*	*VARCHAR2(100)*	*YES*	*100*	*NOT NULL*	*Name of country*
12.	***CreditCardLimit***	*NUMBER*	*NUMBER(5,2)*	*YES*	*X=5 Y=2*	*NOT NULL*	*Number of credit limit*
13.	***CreditCardBalance***	*NUMBER*	*NUMBER(5,2)*	*YES*	*X=5 Y=2*	*NOT NULL*	*Current balance on card*
14.	***ActivationStatus***	*CHARACTER*	*CHAR(1)*	*YES*	*1*	*NOT NULL*	*Status of card activation*

Database Design Deliverable #3 – Physical Model Data Dictionary

2.

CREDITCARDMERCHANT							
Column Num.	***Attribute/Column Name***	***Generic Data Type Name***	***Oracle Data Type Name***	***Is it Required?***	***Length/Size /Format***	***Constraints***	***Description/ purpose***
1.	***MerchantCode***	*NUMBER*	*NUMBER(2)*	*YES*	*2*	*PRIMARY KEY*	*Cod of merchant*
2.	***MerchantName***	*STRING*	*VARCHAR2(50)*	*yes*	*50*	*NOT NULL*	*Name of card merchant*

Database Design Deliverable #3 – Physical Model Data Dictionary

3.

CUSTOMER							
Column Num.	***Attribute/Column Name***	***Generic Data Type Name***	***Oracle Data Type Name***	***Is it Required?***	***Length/Size /Format***	***Constraints***	***Description/ purpose***
•	***CustomerID***	*Number*	*Number GENERATED ALWAYS AS Identity*	*Yes*	*DEFAULT SIZE OF DATA TYPE*	*Identity Primary Key*	*Identifies each customer.* ***Key has no business meaning***
•	***FirstName***	*STRING*	*VARCHAR2(50)*	*YES*	*50*	*NOT NULL*	*Name of customer*
•	***LastName***	*STRING*	*VARCHAR2(50)*	*YES*	*50*	*NOT NULL*	*Last name of cusotmer*
•	***BirthDate***	*DATE*	*DATE*	*YES*	*DD/MON/YY*	*NOT NULL*	*Customer birth date*
•	***AddressLine1***	*STRING*	*VARCHAR2(30)*	*YES*	*30*	*NOT NULL*	*Customer address*
•	***AddressLine2***	*STRING*	*VARCHAR2(5)*	*NO*	*5*	*NULL*	*Optional address line*
•	***City***	*STRING*	*VARCHAR2(25)*	*YES*	*25*	*NOT NULL*	*City name*
•	***StateCode***	*CHARACTER*	*CHAR(2)*	*YES*	*2*	*NOT NULL*	*Name of state*
•	***ZipCode***	*STRING*	*VARCHAR2(10)*	*YES*	*10*	*NOT NULL*	*Zipcode number*
•	***Country***	*STRING*	*VARCHAR2(100)*	*YES*	*100*	*NOT NULL*	*Name of country*
•	***Phone***	*STRING*	*VARCHAR2(20)*	*YES*	*20*	*NOT NULL*	*Phone number*
•	***Email***	*STRING*	*VARCHAR2(50)*	*YES*	*50*	*UNIQUE NOT NULL*	*Email address*
•	***DriverLicenseNumber***	*STRING*	*VARCHAR2(25)*	*YES*	*25*	*UNIQUE NOT NULL*	*Driver license number*
•	***CustomerUserAccountID***	*BINARY*	*RAW(16)*	*NO*	*16*	*FOREIGN KEY NULL*	*Customer's account*
•	***CustomerType***	*CHARACTER*	*CHAR(1)*	*YES*	*1*	*NOT NULL*	*Define the type of customer*

Database Design Deliverable #3 – Physical Model Data Dictionary

4.

CUSTOMER_CREDITCARD							
Column Num.	***Attribute/Column Name***	***Generic Data Type Name***	***Oracle Data Type Name***	***Is it Required?***	***Length/Size /Format***	***Constraints***	***Description/ purpose***
1.	***CreditCardNumber***	*STRING*	*VARCHAR2(16)*	*YES*	*16*	*Composite Primary KEY Foreign Key*	*Composite key that defines customer credit card*
2.	***CustomerID***	*NUMBER*	*NUMBER*	*YES*	*DEFAULT SIZE OF DATA TYPE*	*Composite Primary Key Foreign key*	*Composite key that defines customer credit card*

5.

RETAILCUSTOMER							
Column Num	***Attribute/Column Name***	***Generic Data Type Name***	***Oracle Data Type Name***	***Is it Required?***	***Length/Size /Format***	***Constraints***	***Description/ purpose***
1.	***CustomerID***	*NUMBER*	*NUMBER*	*YES*	*DEFAULT SIZE OF DATA TYPE*	*Primary key*	*Customer unique number*
2.	***DiscountID***	*NUMBER*	*NUMBER*	*NO*	*NUMBER*	*NULL*	*Unique discount ID*
3.	***EZPlusID***	*NUMBER*	*NUMBER(10)*	*NO*	*10*	*NULL*	*Number for EZ plus ID*

Database Design Deliverable #3 – Physical Model Data Dictionary

6.

DISCOUNT							
Column Num	***Attribute/Column Name***	***Generic Data Type Name***	***Oracle Data Type Name***	***Is it Required?***	***Length/Size /Format***	***Constraints***	***Description/ purpose***
1.	***DiscountID***	*Number*	*Number Generated always as IDENTITY*	*yes*	*Default Size of Data Type*	*Identity Primary key*	*Unique Identifier for each Code. But key has not business meaning*
2.	***DiscountCode***	*STRING*	*VARCHAR2(10)*	*YES*	*10*	*UNIQUE NOT NULL*	*Code for any discounts*
3.	***DiscountCodeDesc***	*STRING*	*VARCHAR2(150)*	*YES*	*150*	*NOT NULL*	*Description of code applied*

7.

EZPLUS							
Column Num	***Attribute/Column Name***	***Generic Data Type Name***	***Oracle Data Type Name***	***Is it Required?***	***Length/Size /Format***	***Constraints***	***Description/ purpose***
1.	***EZPlusID***	*Number*	*Number(10)Generated always as IDENTITY*	*yes*	*Default size of Data type*	*Identity primary Key*	*Primary Key with no business meaning. Use to identify each EZ plus*
2.	***EZPlusRewardsCode***	*CHARACTER*	*CHAR(8)*	*YES*	*8*	*UNIQUE Not null*	*Unique code for reward*
3.	***EZPlusRewardsEarnedPoints***	*NUMBER*	*NUMBER(6)*	*NO*	*6*	*NULL*	*Earned points*

Database Design Deliverable #3 – Physical Model Data Dictionary

8.

COPRORATECUSTOMER							
Column Num	***Attribute/Column Name***	***Generic Data Type Name***	***Oracle Data Type Name***	***Is it Required?***	***Length/Size /Format***	***Constraints***	***Description/ purpose***
1.	***CustomerID***	*NUMBER*	*NUMBER*	*YES*	*DEFAULT SIZE OF DATA TYPE*	*PRIMARY KEY*	*Unique identifier for each customer*
2.	***CompanyID***	*NUMBER*	*NUMBER(5)*	*YES*	*5*	*CHECK(CompanyID BETWEEN 1 AND 20000) NOT NULL*	*Unique Identifier for each company*

Database Design Deliverable #3 – Physical Model Data Dictionary

9.

COMPANY							
Column Num.	***Attribute/Column Name***	***Generic Data Type Name***	***Oracle Data Type Name***	***Is it Required?***	***Length/Size /Format***	***Constraints***	***Description/ purpose***
1.	***CompanyID***	*Number*	*Number (5)*	*YES*	*5*	*Primary Key CHECK(CompanyID BETWEEN 1 AND 20000)*	*Primary key with business meaning*
2.	***CompanyName***	*STRING*	*VARCHAR2(50)*	*YES*	*50*	*UNIQUE NOT NULL*	*Name of company*
3.	***AddressLine1***	*STRING*	*VARCHAR2(30)*	*YES*	*30*	*NOT NULL*	*Address name for company*
4.	***AddressLine2***	*STRING*	*VARCHAR2(5)*	*NO*	*5*	*NULL*	*Optional address line*
5.	***City***	*STRING*	*VARCHAR2(25)*	*YES*	*25*	*NOT NULL*	*Name of city*
6.	***StateCode***	*CHARACTER*	*CHAR(2)*	*YES*	*2*	*NOT NULL*	*Name of state*
7.	***ZipCode***	*STRING*	*VARCHAR2(10)*	*YES*	*10*	*NOT NULL*	*Zipcode number*
8.	***Country***	*STRING*	*VARCHAR2(100)*	*YES*	*100*	*NOT NULL*	*Name of country*
9.	***CompanyRepName***	*STRING*	*VARCHAR2(50)*	*YES*	*50*	*NOT NULL*	*Name of represenratiuve*
10.	***ContactPhone***	*STRING*	*VARCHAR2(20)*	*YES*	*20*	*NOT NULL*	*Contact phone number*
11.	***ContactEmail***	*STRING*	*VARCHAR2(50)*	*YES*	*50*	*UNIQUE NOT NULL*	*Email address*
12.	***CorporateDiscountPercentageRate***	*NUMBER*	*NUMBER(1,2)*	*YES*	*X=1 Y=2*	*NOT NULL*	*Rate for corporate discount*

Database Design Deliverable #3 – Physical Model Data Dictionary

10.

DRIVERLICENSE							
Column Num.	***Attribute/Column Name***	***Generic Data Type Name***	***Oracle Data Type Name***	***Is it Required?***	***Length/Size /Format***	***Constraints***	***Description/ purpose***
1.	***DriverLicenseNumber***	*STRING*	*VARCHAR2(25)*	*yes*	*25*	*Primary key*	*Unique ID with business meaning TO identify driver*
2.	***DriverLicenseExpDate***	*DATE*	*DATE*	*YES*	*MM/MON/YY*	*NOT NULL*	*Expiration date*
3.	***DriverLicenseState***	*CHARACTER*	*CHAR(2)*	*YES*	*2*	*NOT NULL*	*State name*

11.

CUSTOMERUSERACCOUNT							
Column Num.	***Attribute/Column Name***	***Generic Data Type Name***	***Oracle Data Type Name***	***Is it Required?***	***Length/Size /Format***	***Constraints***	***Description/ purpose***
1.	***UserAccountID***	*BINARY*	*RAW(16)*	*YES*	*16*	*PRIMARY KEY DEFAULT SYS_GUID()*	*Unique identifier for each customer account*
2.	***Username***	*STRING*	*VARCHAR2(50)*	*YES*	*50*	*UNIQUE NOT NULL*	*Username for customer*
3.	***Password***	*STRING*	*VARCHAR2(50)*	*YES*	*50*	*NOT NULL*	*Customer password*
4.	***Email***	*STRING*	*VARCHAR2(50)*	*YES*	*50*	*UNIQUE NOT NULL*	*UNIQUE customer email*

Database Design Deliverable #3 – Physical Model Data Dictionary

12.

VEHICLE							
Column Num	**Attribute/Column Name**	**Generic Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
1.	**VehicleID**	Number	Number Generated always as IDENTITY	Yes	Default size of Data type	Identity Primary key	Unique Identifier of each vehicle. Key has no business meaning
2.	**VINNumber**	STRING	VARCHAR2(25)	YES	25	UNIQUE NOT NULL	VINNumber for each car
3.	**Make**	STRING	VARCHAR2(15)	YES	15	NOT NULL	Make number
4.	**Model**	STRING	VARCHAR2(25)	YES	25	NOT NULL	Model name
5.	**Year**	NUMBER	NUMBER(4)	YES	4	NOT NULL	Year number
6.	**Color**	STRING	VARCHAR2(15)	YES	15	NOT NULL	Color of each car
7.	**LicensePlateNumber**	STRING	VARCHAR(20)	YES	20	NOT NULL	License number for each vehicle
8.	**LicensePlateState**	CHARACTER	CHAR(2)	YES	2	NOT NULL	Name of state
9.	**Mileage**	NUMBER	NUMBER(4,3)	YES	X=4 Y=3	NOT NULL	Mileage number
10.	**TransmissionTypeID**	NUMBER	NUMEBR(2)	YES	2	Foreign Key NOT NULL	Transmision type of car
11.	**SeatCapacity**	NUMBER	NUMBER(2)	YES	2	NOT NULL	Seat capacity of card
12.	**VehicleRentalCategoryID**	NUMBER	NUMBER(2)	YES	2	FOREIGN KEY NOT NULL	Vehicle category
13.	**VehicleStatusID**	NUMBER	NUMBER(1)	YES	NOT NULL	FOREIGN KEY NOT NULL	Status of each vehicle
14.	**VehicleOwningAgencyID**	NUMBER	NUMBER(5)	YES	FOREIGN KEY NOT NULL		Unique Identifier for each agency
15.	**VehicleCurrentLocationAgencyID**						
16.	**VehicleType**	CHARACTER	CHAR(1)	YES	1	NOT NULL	Type of vehicle

Database Design Deliverable #3 – Physical Model Data Dictionary

13.

TRANSMISSIONTYPE							
Column Num	***Attribute/Column Name***	***Generic Data Type Name***	***Oracle Data Type Name***	***Is it Required?***	***Length/Size /Format***	***Constraints***	***Description/ purpose***
1.	***TransmissionTypeID***	*Number*	*Number (2)*	*yes*	*2*	*Primary key*	*PK to indentify the transmission of the vehicle*
2.	***TransmissionTypeDesc***	*STRING*	*VARCHAR2(50)*	*YES*	*50*	*NOT NULL*	*Description of each transmission*

14.

CAR							
Column Num	***Attribute/Column Name***	***Generic Data Type Name***	***Oracle Data Type Name***	***Is it Required?***	***Length/Size /Format***	***Constraints***	***Description/ purpose***
1.	***VehicleID***	*NUMBER*	*NUMBER*	*Y*	*DEFAULT SIZE OF DATA TYPE*	*PRIMARY KEY*	*Unique identifier for each vehicle. KEY HAS NO BUSINNES MEANING*
2.	***TrunkCapacity***	*NUMBER*	*NUMBER(2,2)*	*YES*	*X=2 Y=2*	*NOT NULL*	*Capacity of trunk*

Database Design Deliverable #3 – Physical Model Data Dictionary

15.

SUV							
Column Num	***Attribute/Column Name***	***Generic Data Type Name***	***Oracle Data Type Name***	***Is it Required?***	***Length/Size /Format***	***Constraints***	***Description/ purpose***
1.	***VehicleID***	*NUMBER*	*NUMBER*	*YES*	*DEFAULT size of DATA TYPE*	*PRIMARY KEY*	*Unique identifier for each vehicle. KEY HAS NO BUSINNES MEANING*
2.	***TowingCapacity***	*NUMBER*	*NUMBER(5,2)*	*YES*	*X=5 Y=2*	*NOT NULL*	*Capacity number of towing*
3.	***IsAWD***	*NUMBER*	*NUMBER(1)*	*YES*	*1*	*CHECK(IsAWD BETWEEN 0 AND 1) NOT NULL*	*Yes or no response*

16.

MINIVAN							
Column Num	***Attribute/Column Name***	***Generic Data Type Name***	***Oracle Data Type Name***	***Is it Required?***	***Length/Size /Format***	***Constraints***	***Description/ purpose***
1.	***VehicleID***	*NUMBER*	*NUMBER*	*YES*	*DEFAULT SIZE OF THE DATA TYPE*	*PRIMARY KEY*	*Unique identifier for each vehicle. KEY HAS NO BUSINNES MEANING*
2.	***HasDisabilityPackage***	*NUMBER*	*NUMBER(1)*	*YES*	*1*	*NOT NULL CHECK(HasDisabilityPackage BETWEEN 0 AND 1)*	*Yes or no response*

Database Design Deliverable #3 – Physical Model Data Dictionary

17.

CARGOVAN							
Column Num	***Attribute/Column Name***	***Generic Data Type Name***	***Oracle Data Type Name***	***Is it Required?***	***Length/Size /Format***	***Constraints***	***Description/ purpose***
1.	***VehicleID***	*NUMBER*	*NUMBER*	*YES*	*DEFAULT SIZE OF DATA TYPE*	*PRIMARY KEY*	*Unique identifier for each vehicle. KEY HAS NO BUSINNES MEANING*
2.	***CargoCapacity***	*NUMBER*	*NUMBER(3,2)*	*YES*	*X=3 Y=2*	*NOT NULL*	*Capacity of cargo*
3.	***MaximumPayload***	*NUMBER*	*NUMBER(4,3)*	*YES*	*X=4 Y=3*	*NOT NULL*	*Maximun capacity of payload*

18

VEHICLESTATUS							
Column Num	***Attribute/Column Name***	***Generic Data Type Name***	***Oracle Data Type Name***	***Is it Required?***	***Length/Size /Format***	***Constraints***	***Description/ purpose***
1.	***VehicleStatusID***	*NUMBER*	*Number(1)*	*yes*	*1*	*Primary key*	***Business meaning*** *Unique identifier to identify the status of vehicle.*
2.	***VehicleStatusDesc***	*STRING*	*VARCHAR2(50)*	*YES*	*50*	*NOT NULL*	*Status description of each vehicle*

Database Design Deliverable #3 – Physical Model Data Dictionary

RESERVATION							
Column Num	**Attribute/Column Name**	**Generic Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
1.	**ReservationID**	Number	NUMBER(4)	YES	Default size OF DATA TYPE	PRIMARY KEY	UNIQUE identifier to track reservations
2.	**CustomerID**	NUMBER	NUMBER	YES	DEFAULT SIZE OF DATA TYPE	FOREIGN KEY NOT NULL	UNIQUE IDENTIFIER FOR EACH CUSTOMER
3.	**VehicleRentalCategoryID**	NUMBER	NUMBER(2)	YES	2	FOREIGN KEY NOT NULL	Unique Identifier for each rental cateogry
4.	**ReserevationPickUpRentalAgencyID**	NUMBER	NUMBER(5)	YES	5	FOREIGN KEY NOT NULL	Unique identifier for each rental agency
5.	**ReservationDropOffRentalAgencyID**	NUMBER	NUMBER(5)	YES	5	FOREIGN KEY NOT NULL	Unique identifier for each drop off agency
6.	**ReservationPickUplDate**	DATE	DATE	YES	DD/MON/YY	NOT NULL	
7.	**ReservationPickUpTime**	NUMBER	NUMBER(4)	YES	4	CHECK(ReservationPickUpTime BETWEEN 0 AND 2400) NOT NULL	Time for each reservation pick up
8.	**ReservationDropOffDate**	DATE	DATE	YES	DD/MON/YY	NOT NULL	Date for each drop off
9.	**ReservationDropOffTime**	NUMBER	NUMBER(4)	YES	4	NOT NULL CHECK(ReservationDropOffTime BETWEEN 0 AND 2400)	Time for each drop off
10.	**ReservationStatusID**	NUMBER	NUMBER(1)	YES	1	FOREIGN KEY NOT NULL	Status for each reservation

Database Design Deliverable #3 – Physical Model Data Dictionary

20.

VEHICLERENTALCATEGORY							
Column Num	**Attribute/Column Name**	**Generic Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
1.	**VehicleRentalCategoryID**	NUMBER	NUMBER(2)	YES	2	Primary key	Primary key that identifies the category of the vehicle being rented or reserved
2.	**CategoryName**	STRING	VARCHAR2(40)	YES	40	UNIQUE NOT NULL	Name of vehicle category
3.	**CategoryDailyRentalRate**	NUMBER	NUMBER(3,2)	YES	X=3 Y=2	NOT NULL	Rate for each rental category

21.

RESERVATIONSTATUS							
Column Num	**Attribute/Column Name**	**Generic Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
1.	**ReservationStatusID**	NUMBER	NUMBER(1)	YES	1	Primary key	**PK with business meaning** that identifies the status of a reservation
2.	**ReservationStatusDesc**	STRING	VARCHAR2(30)	YES	30	NOT NULL	Description for each reservation

Database Design Deliverable #3 – Physical Model Data Dictionary

22.

RENTALAGENCY							
Column Num	**Attribute/Column Name**	**Generic Data Type Name**	**Oracle Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
1.	**RentalAgencyID**	Number	Number(5)	yes	5	Primary key CHECK(RentalAgencyID BETWEEN 1 AND10000)	Unique identifier to complete rental
2.	**RentalAgencyName**	STRING	VARCHAR2(50)	YES	50	UNIQUE NOT NULL	Name of each rental agency
3.	**AddressLine1**	STRING	VARCHAR2(30)	YES	30	NOT NULL	Address of each rental agency
4.	**AddressLine2**	STRING	VARCHAR2(5)	NO	5	NULL	Optional address line
5.	**City**	STRING	VARCHAR2(25)	YES	25	NOT NULL	Name of city
6.	**StateCode**	CHARACTER	CHAR(2)	YES	2	NOT NULL	Name of state
7.	**Country**	STRING	VARCHAR2(100)	YES	100	NOT NULL	Name of country
8.	**ZipCode**	STRING	VARCHAR2(10)	YES	10	NOT NULL	Zip code number
9.	**Phone**	STRING	VARCHAR2(20)	YES	20	NOT NULL	Phone number
10.	**Email**	STRING	VARCHAR2(50)	YES	50	UNIQUE NOT NULL	Unique email address

Database Design Deliverable #3 – Physical Model Data Dictionary

RENTAL							
Column Num	***Attribute/Column Name***	***Generic Data Type Name***	***Oracle Data Type Name***	***Is it Required?***	***Length/Size /Format***	***Constraints***	***Description/ purpose***
1.	***RentalAgreementID***	*NUMBER*	*NUMBER*	*YES*	*Default size of datatype*	*Primary KEY*	*Unique identifier that identifies rental transactions*
2.	***CustomerID***	*NUMBER*	*NUMBER*	*YES*	*DEFAULT SIZE OF DATA TYPE*	*FOREIGN KEY NOT NULL*	*Customer ID*
3.	***VehicleRentalCategoryID***	*NUMBER*	*NUMBER(2)*	*YES*	*2*	*FOREIGN KEY NOT NULL*	*Rental category ID*
4.	***RentalPickUpRentalAgencyID***	*NUMBER*	*NUMBER(5)*	*YES*	*5*	*FOREIGN KEY NOT NULL*	*UNIQUE rental agency ID*
5.	***RentalDropOffREntalAgencyID***	*NUMBER*	*NUMBER(5)*	*YES*	*5*	*FOREGIN KEY NOT NULL*	*Unique rental drop off ID*
6.	***VehicleID***	*NUMBER*	*NUMBER*	*YES*	*DEFAULT SIZE OF DATA TYPE*	*FOREIGN KEY NOT NULL*	*Vehicle ID*
7.	***ReservationID***	*NUMBER*	*NUMBER*	*YES*	*DEFAULT SIZE OF DATA TYPE*	*FOREIGN KEY NULL*	*Reservation ID*
8.	***RentalPickUpDate***	*DATE*	*DATE*	*YES*	*DD/MON/YY*	*NOT NULL*	*DATE for rental pick up*
9.	***RentalPickUpTime***	*NUMBER*	*NUMBER(4)*	*YES*	*4*	*NOT NULL CHECK(RentalPickUpTime BETWEEN 0 AND 2400)*	*Time for rental pick up*
10.	***RentalDropOffDate***	*DATE*	*DATE*	*YES*	*MM/MON/YY*	*NOT NULL*	*Date for drop off*
11.	***RentalDropOffTime***	*NUMBER*	*NUMBER(4)*	*YES*	*4*	*NOT NULL CHECK(**RentalDropOffTime** BETWEEN 0 AND 2400)*	*NOT NULL*
12.	***RentalPickUpOdometerValue***	*NUMBER*	*NUMBER(5,2)*	*YES*	*X=5 Y=2*	*NOT NULL*	*Value for pick up odometer*
13.	***RentalDropOffOdometerValue***	*NUMBER*	*NUMBER(5,2)*	*YES*	*X=5 Y=2*	*NOT NULL*	*Value for drop off odometer*
14.	***RentalFuelOptionID***	*NUMBER*	*NUMBER(1)*	*YES*	*1*	*FOREIGN KEY NOT NULL*	*Unique fuel option ID*

15.	***InsuranceOptionID***	*NUMBER*	*NUMBER(1)*	*YES*	*1*	*FOREIGN KEY NOT NULL*	*UNIQUE identifier for insurance*
16.	***RentalStatusID***	*NUMBER*	*NUMBER(1)*	*YES*	*1*	*FOREIGN KEY N NOT NULL*	*Freign key for status of rental*

Database Design Deliverable #3 – Physical Model Data Dictionary

24.

RENTALSTATUS							
Column Num	***Attribute/Column Name***	***Generic Data Type Name***	***Oracle Data Type Name***	***Is it Required?***	***Length/Size /Format***	***Constraints***	***Description/ purpose***
1.	***RentalStatusID***	*NUMBER*	*NUMBER(1)*	*YES*	*1*	*Primary key*	*Pk that identifies status of a rental.*
2.	***RentalStatusDesc***	*String*	*VARCHAR2(50)*	*YES*	*50*	*NOT NULL*	*DESCRIPTION OF RENTAL STATUS*

25.

FUELOPTION							
Column Num	***Attribute/Column Name***	***Generic Data Type Name***	***Oracle Data Type Name***	***Is it Required?***	***Length/Size /Format***	***Constraints***	***Description/ purpose***
1.	***FuelOptionID***	*NUMBER*	*NUMBER(1)*	*YES*	*1*	*PRIMARY KEY*	*PK that identifies the fuel option by the customer*
2.	***FuelOptionDesc***	*STRING*	*VARCHAR2(50)*	*YES*	*50*	*NOT NULL*	*Description of fuel option*

Database Design Deliverable #3 – Physical Model Data Dictionary

26.

RENTALINSURANCEOPTION							
Column Num	***Attribute/Column Name***	***Generic Data Type Name***	***Oracle Data Type Name***	***Is it Required?***	***Length/Size /Format***	***Constraints***	***Description/ purpose***
1.	***InsuranceOptionID***	*NUMBER*	*NUMBER(1)*	*yes*	*1*	*Primary Key*	*Pk that provides insurance options*
2.	***InsuranceOptionDesc***	*STRING*	*VARCHAR2(150)*	*YES*	*150*	*NOT NULL*	*Description of insurance selected*
3.	***InsuranceOptionAdditionalCost***	*NUMBER*	*NUMBER(2,2)*	*YES*	*X=2* *Y=2*	*NOT NULL*	*Additional cost*

Database Design Deliverable #3 – Physical Model Data Dictionary

27.

EMPLOYEE							
Column Num	***Attribute/Column Name***	***Generic Data Type Name***	***Oracle Data Type Name***	***Is it Required?***	***Length/Size /Format***	***Constraints***	***Description/ purpose***
1.	***EmployeeID***	*NUMBER*	*NUMBER(5)*	*YES*	*5*	*PRIMARY KEY*	*Primary Key FOR EACH EMPLOYEE*
2.	***FirstName***	*STRING*	*VARCHAR2(50)*	*YES*	*50*	*NOT NULL*	*First name of employee*
3.	***LastName***	*STRING*	*VARCHAR2(50)*	*YES*	*50*	*NOT NULL*	*Last name of employee*
4.	***SSNumber***	*STRING*	*VARCHAR2(11)*	*YES*	*11*	*UNIQUE NOT NULL*	*Unique ss number*
5.	***BirthDate***	*DATE*	*Date*	*YES*	*DD/MON/YY*	*NOT NULL*	*Employee birthdate*
6.	***AddressLine1***	*STRING*	*VARCHAR2(30)*	*YES*	*30*	*NOT NULL*	*Employee address*
7.	***AddressLine2***	*STRING*	*VARCHAR2(5)*	*NO*	*5*	*NULL*	*Optional address line*
8.	***City***	*STRING*	*VARCHAR2(25)*	*YES*	*25*	*NOT NULL*	*Name of city*
9.	***StateCode***	*CHARACTER*	*CHAR(2)*	*YES*	*2*	*NOT NULL*	*Name of state*
10.	***ZipCode***	*STRING*	*VARCHAR2(10)*	*YES*	*10*	*NOT NULL*	*Zipcode number*
11.	***Country***	*STRING*	*VARCHAR2(100)*	*YES*	*100*	*NOT NULL*	*Name of country*
12.	***EmployeePhone***	*STRING*	*VARCHAR2(20)*	*YES*	*20*	*NOT NUTLL*	*Employee number*
13.	***EmployeeJobTitle***	*STRING*	*VARCHAR2(30)*	*YES*	*30*	*NOT NULL*	*Job title of employee*
14.	***EmployeeEmail***	*STRING*	*VARCHAR2(50)*	*YES*	*50*	*UNIQUE NOT NULL*	*Unique email of employee*
15.	***EmployeeUserAccountID***	*BINARY*	*RAW(16)*	*YES*	*16*	*FOREIGN KEY NOT NULL*	*Account ID of employee*

Database Design Deliverable #3 – Physical Model Data Dictionary

28.

EMPLOYEEUSERACCOUNT							
Column Num.	***Attribute/Column Name***	***Generic Data Type Name***	***Oracle Data Type Name***	***Is it Required?***	***Length/Size /Format***	***Constraints***	***Description/ purpose***
1.	***UserAccountID***	*BINARY*	*RAW(16)*	*Yes*	*16*	*Primary Key*	*GUID for each customer.Therefore,* **businees meaning**
2.	***Username***	*STRING*	*VARCHAR2(50)*	*YES*	*50*	*UNIQUE NOT NULL*	*Username of employee*
3.	***Password***	*STRING*	*VARCHAR2(50)*	*YES*	*50*	*NOT NULL*	*Employee account password*
4.	***Email***	*STRING*	*VARCHAR2(50)*	*YES*	*50*	*UNIQUE NOT NULL*	*UNIQUE email address*

Database Design Deliverable #3 – Physical Model Data Dictionary

29.

TRANSPORT							
Column Num	***Attribute/Column Name***	***Generic Data Type Name***	***Oracle Data Type Name***	***Is it Required?***	***Length/Size /Format***	***Constraints***	***Description/ purpose***
1.	***EmployeeID***	*NUMBER*	*NUMBER(5)*	*YES*	*5*	*Primary key check(EmployeeID BETWEEN 1AND 20000)*	**Composite key** *that defines transport table*
2.	***VehicleID***	*NUMBER*	*NUMBER*	*YES*	*DEFAULT SIZE OF THE DATA TYPE*	*PRIMARY KEY*	*Composite vehicle key*
3.	***RentalAgencyID***	*NUMBER*	*NUMBER(5)*	*YES*	*5*	*PRIMARY KEY CHECK(RentalAgencyID BETWEEN 1 AND 10000)*	*Composite key, rental agency ID*
4.	***DropOffRentalAgencyID***	*NUMBER*	*NUMBER(5)*	*YES*	*5*	*FOREIGN KEY NOT NULL*	*UNIQUE IDENTIFIER FOR EACH RENTAL AGENCY*
5.	***DriverDepartureDate***	*DATE*	*DATE*	*YES*	*DD/MON/YY*	*NOT NULL*	*DATE FOR DEPARTURE*
6.	***DriverDepartureTime***	*NUMBER*	*NUMBER (4)*	*YES*	*4*	*NOT NULL CHECK(**DriverDepartureTime** BETWEEN 0 AND 2400)*	*TIME FOR EACH DEPATURE*
7.	***VehiclePickUpDate***	*DATE*	*DATE*	*NO*		*NULL*	*DATE FOR PICK UP*
8.	***VehiclePickTime***	*NUMBER*	*NUMBER(4)*	*NO*	*4*	*NULL CHECK(VehiclePickTime FROM 0 AND 2400)*	*TIME for a vehicle pick up*
9.	***VehicleDropOffDate***	*DATE*	*DATE*	*NO*	*DD/MON/YY*	*NULL*	*Date for drop off*
10.	***VehicleDropOffTime***	*NUMBER*	*NUMBER(4)*	*NO*	*4*	*NULL CHECK(VehicleDropOffTime FROM 0 AND 2400)*	*Time for drop off*
11.	***TransportStatusID***	*NUMBER*	*NUMBER(1)*	*YES*	*1*	*FOREIGN KEY NOT NULL*	*Status of transport*
12.	***TransportReasonID***	*NUMBER*	*NUMBER(1)*	*YES*	*1*	*FOREIGN KEY NOT NULL*	

Database Design Deliverable #3 – Physical Model Data Dictionary

30.

TRANSPORTSTATUS							
Column Num	***Attribute/Column Name***	***Generic Data Type Name***	***Oracle Data Type Name***	***Is it Required?***	***Length/Size /Format***	***Constraints***	***Description/ purpose***
1.	***TransportStatusID***	*NUMBER*	*NUMBER(1)*	*YES*	*1*	*Primary key*	***UNIQUE** Identifier for each transport status*
2.	***TransportStatusDesc***	*String*	*VARCHAR2(50)*	*YES*	*50*	*NO NULL*	*Description of transport status*

31.

TRANSPORTREASON							
Column Num	***Attribute/Column Name***	***Generic Data Type Name***	***Oracle Data Type Name***	***Is it Required?***	***Length/Size /Format***	***Constraints***	***Description/ purpose***
1.	***TransportReasonID***	*Number*	*NUMBER(1)*	*Yes*	*1*	*Primary Key*	*Unique key that identifies each transport reason.*
2.	***TransportReasonDesc***	*STRING*	*VARCHAR2(50)*	*YES*	*50*	*NOT NULL*	*Description of a transportation*

Database Design Deliverable #3 – Physical Model Data Dictionary

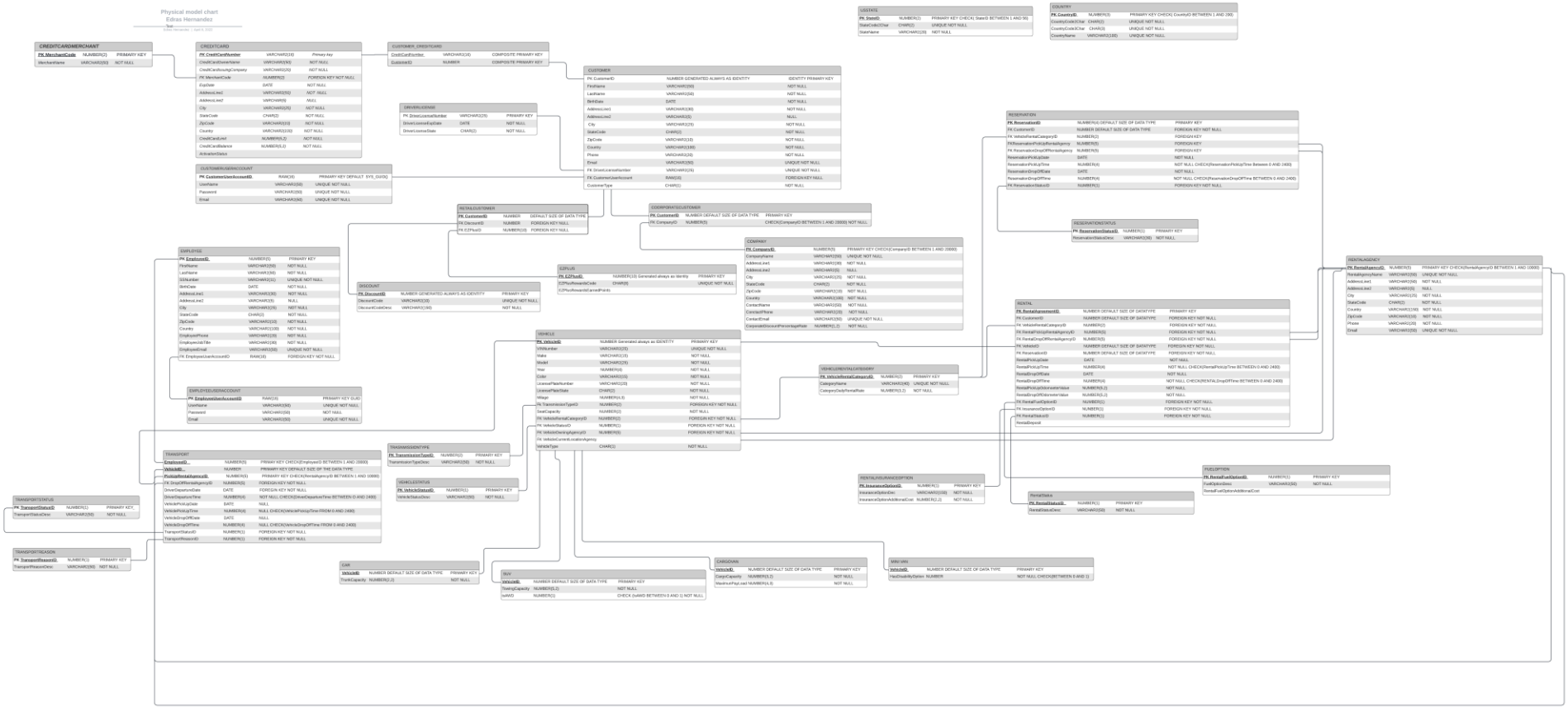
32.

USSTATE							
Column Num.	***Attribute/Column Name***	***Generic Data Type Name***	***Oracle Data Type Name***	***Is it Required?***	***Length/Size /Format***	***Constraints***	***Description/ purpose***
1.	***StateID***	*NUMBER*	*NUMBER(2)*	*yes*	*2*	*PRIMARY KEY Check between 1-56*	*Unique identifier of each state that support UI*
2.	***StateCode2Char***	*CHARACTER*	*CHAR(2)*	*YES*	*2*	*UNIQUE NOT NULL*	*Abbreviation of each state*
3.	***StateName***	*STRING*	*VARCHAR(20)*	*YES*	*20*	*NOT NULL*	*Name of each state*

33

COUNTRY							
Column Num.	***Attribute/Column Name***	***Generic Data Type Name***	***Oracle Data Type Name***	***Is it Required?***	***Length/Size /Format***	***Constraints***	***Description/ purpose***
1.	***CountryID***	*NUMBER*	*NUMBER(3)*	*YES*	*3*	*PRIMRARY KEYCheck (CountryID Between 1-200) NOT NULL*	**Unique** *identifier for each country*
2.	***CountryCode2Char***	*CHARACTER*	*CHAR(2)*	*YES*	*2*	*UNIQUE NOT NULL*	***UNIQUE*** *country code*
3.	***CountryCode3Char***	*CHARACTER*	*CHAR(3)*	*YES*	*3*	*UNIQUE NOT NULL*	*Additional* ***UNIQUE*** *country code*
4.	***CountryName***	*STRING*	*VARCHAR2(100)*	*YES*	*100*	*UNIQUE NOT NULL*	*Name of country*

Database Design Deliverable #4 – Physical Model Schema Design Diagram



Database implementation Deliverable #5 –Development & Implementation

- *Below are the thirty-three SQL statements that were used in ORACLE to create the tables for EZ AUTORENTAL company. Each statement containing its datatype according to the business requirements.*

```
CREATE TABLE CREDITCARDMERCHANT
(
MerchantCode Number(2) PRIMARY KEY,
MerchantName VARCHAR2(50) NOT NULL
);

CREATE TABLE CREDITCARD
(
CreditCardNumber VARCHAR2(16),
CreditCardOwnerName VARCHAR2(50) NOT NULL,
CreditCardIssuingCompany VARCHAR2(50) NOT NULL,
MerchantCode NUMBER(2) NOT NULL,
ExpDate DATE NOT NULL,
AddressLine1 VARCHAR2(50) NOT NULL,
AddressLine2 VARCHAR2(5) NULL,
City VARCHAR2(25) NOT NULL,
StateCode CHAR(2) NOT NULL,
ZipCode VARCHAR2(10) NOT NULL,
Country VARCHAR2(100) NOT NULL,
CreditCardLimit NUMBER(5,2) NOT NULL,
CreditCardBalance NUMBER(5,2),
ActivationStatus CHAR(1) NOT NULL,
CONSTRAINT CreditCardNumber_PK PRIMARY KEY(CreditCardNumber),
CONSTRAINT MerchantCode_FK FOREIGN KEY (MerchantCode) REFERENCES CREDITCARDMERCHANT(MerchantCode)
);

CREATE TABLE DRIVERLICENSE
```

```
(
DriverLicenseNumber VARCHAR2(25) PRIMARY KEY,
DriverLicenseExpDate DATE NOT NULL,
DriverLicenseState CHAR(2) NOT NULL
);

CREATE TABLE CUSTOMERUSERACCOUNT
(
CustomerUserAccountID RAW(16) DEFAULT SYS_GUID() PRIMARY KEY,
UserName VARCHAR2(50) UNIQUE NOT NULL,
Password VARCHAR2(50) UNIQUE NOT NULL,
Email VARCHAR2(50) UNIQUE NOT NULL
);

CREATE TABLE CUSTOMER
(
CustomerID NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
FirstName VARCHAR2(50) NOT NULL,
LastName VARCHAR2(50) NOT NULL,
BirthDate DATE NOT NULL,
AddressLine1 VARCHAR2(30) NOT NULL,
AddressLine2 VARCHAR2(5) NULL,
City VARCHAR2(25) NOT NULL,
StateCode CHAR(2) NOT NULL,
ZipCode VARCHAR2(10) NOT NULL,
Country VARCHAR2(100) NOT NULL,
Phone VARCHAR2(20) NOT NULL,
Email VARCHAR2(50) UNIQUE NOT NULL,
DriverLicenseNumber VARCHAR2(25) NOT NULL,
CustomerUserAccountID RAW(16) NULL,
CustomerType CHAR(1) NOT NULL,
```

```
CONSTRAINT DriverLicenseNumber_FK UNIQUE(DriverLicenseNumber),
CONSTRAINT CustomerUserAccountID_FK FOREIGN KEY(CustomerUserAccountID) REFERENCES
CUSTOMERUSERACCOUNT(CustomerUserAccountID)
);

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

CONSTRAINT COMPOSITE_KEY PRIMARY KEY(CreditCardNumber, CustomerID)
);

CREATE TABLE DISCOUNT
(
DiscountID NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
DicountCode VARCHAR2(10) UNIQUE NOT NULL,
DiscountCodeDesc VARCHAR2(150) NOT NULL
);

CREATE TABLE EZPLUS
(
EZPlusID NUMBER(10)GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
EZPlusRewardsCode CHAR(8) UNIQUE NOT NULL,
EZPlusRewardaEarnedPoints NUMBER(6) NULL
);

CREATE TABLE RETAILCUSTOMER
(
CustomerID NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
```

```
DiscountID NUMBER NOT NULL,
EZPlusID NUMBER(10) NOT NULL,
CONSTRAINT DiscountID_FK FOREIGN KEY(DiscountID) REFERENCES DISCOUNT(DiscountID),
CONSTRAINT EZPlusID_FK FOREIGN KEY(EZPlusID) REFERENCES EZPLUS(EZPlusID)
);

CREATE TABLE COMPANY
(
CompanyID NUMBER(5) CHECK(CompanyID BETWEEN 1 AND 20000) PRIMARY KEY,
CompanyName VARCHAR2(50) UNIQUE NOT NULL,
AddressLine1 VARCHAR2(30) NOT NULL,
Adressline2 VARCHAR2(5) NULL,
City VARCHAR2(25) NOT NULL,
StateCode CHAR(2) NOT NULL,
ZipCode VARCHAR2(10) NOT NULL,
Country VARCHAR2(100) NOT NULL,
ContactName VARCHAR2(50) NOT NULL,
ContactPhone VARCHAR2(20) NOT NULL,
ContactEmail VARCHAR2(50) UNIQUE NOT NULL,
CorporateDiscountPercentageRate NUMBER(1,2) NOT NULL
);

CREATE TABLE COORPORATECUSTOMER
(
CustomerID NUMBER GENERATED ALWAYS AS IDENTITY,
CompanyID NUMBER(5) CHECK(CompanyID BETWEEN 1 AND 20000),

CONSTRAINT CustomerID_PK PRIMARY KEY(CustomerID),
CONSTRAINT CompanyID_FK FOREIGN KEY(CompanyID) REFERENCES COMPANY(CompanyID)
);
```

Database implementation Deliverable #5 –Development & Implementation (cont'd)

```
CREATE TABLE VEHICLERENTALCATEGORY
(
VehicleRentalCategoryID NUMBER(2) PRIMARY KEY,
CategoryName VARCHAR2(40) UNIQUE NOT NULL,
CategoryDailyRentalRate Number(3,2) NOT NULL
);

CREATE TABLE RENTALAGENCY
(
RentalAgencyID NUMBER(5) CHECK(RentalAgencyID BETWEEN 1 AND 10000) PRIMARY KEY,
RentalAgencyName VARCHAR2(50) UNIQUE NOT NULL,
Adressline1 VARCHAR2(50) NOT NULL,
Adressline2 VARCHAR2(5) NULL,
City VARCHAR2(25) NOT NULL,
StateCode CHAR(2) NOT NULL,
Country VARCHAR2(100) NOT NULL,
ZipCode VARCHAR2(10) NOT NULL,
Phone VARCHAR2(20) NOT NULL,
Email VARCHAR2(50) UNIQUE NOT NULL
);

CREATE TABLE RESERVATIONSTATUS
(
ReservationStatusID NUMBER(1) PRIMARY KEY,
ReservationStatusDesc VARCHAR2(30) NOT NULL
);

CREATE TABLE RESERVATION
(
ReservationID NUMBER(4)NOT NULL,
CustomerID NUMBER NOT NULL,
```

```
VehicleRentalCategoryID NUMBER(2)NOT NULL,
ReservationPickUpRentalAgencyID NUMBER(5) NOT NULL CHECK(ReservationPickUpRentalAgencyID BETWEEN 1 AND 10000),
ReservationDropOffRentalAgencyID NUMBER(5) NOT NULL CHECK(ReservationDropOffRentalAgencyID BETWEEN 1 AND 10000),
ResrvationPickUpDate DATE NOT NULL,
ReservationPickUpTime NUMBER(4)CHECK(ReservationPickUpTime BETWEEN 0 AND 2400) NOT NULL,
ReservationDropOffDate DATE NOT NULL,
ReservationDropOffTime NUMBER(4) CHECK(ReservationDropOffTime BETWEEN 0 AND 2400) NOT NULL,
ReservationStatusID NUMBER(1) NOT NULL,
PRIMARY KEY(ReservationID),
FOREIGN KEY(CustomerID) REFERENCES CUSTOMER(CustomerID),
FOREIGN KEY(VehicleRentalCategoryID) REFERENCES VEHICLERENTALCATEGORY(VehicleRentalCategoryID),
FOREIGN KEY(ReservationPickUpRentalAgencyID) REFERENCES RENTALAGENCY(RentalAgencyID),
FOREIGN KEY(ReservationDropOffRentalAgencyID) REFERENCES RENTALAGENCY(RentalAgencyID),
FOREIGN KEY(ReservationStatusID) REFERENCES RESERVATIONSTATUS(ReservationStatusID)
);

CREATE TABLE TRANSMISSIONTYPE
(
TransmissionTypeID NUMBER(2)PRIMARY KEY,
TransmissionTypeDesc VARCHAR2(50) NOT NULL
);

CREATE TABLE VEHICLESTATUS
(
VehicleStatusID NUMBER(1) PRIMARY KEY,
VehicleStatusDesc VARCHAR2(50) NOT NULL
);

CREATE TABLE VEHICLE
(
VehicleID NUMBER GENERATED ALWAYS AS IDENTITY,
```

```
VINNumber VARCHAR2(25) UNIQUE NOT NULL,
Make VARCHAR2(15) NOT NULL,
VehicleModel VARCHAR2(25) NOT NULL,
VehicleYear NUMBER(4) NOT NULL,
Color VARCHAR2(15) NOT NULL,
LicensePlateNumber VARCHAR2(20) NOT NULL,
LicensePlateState CHAR(2) NOT NULL,
Mileage NUMBER(4,3) NOT NULL,
TransmissionTypeID NUMBER(2) NOT NULL,
SeatCapacity NUMBER(2) NOT NULL,
VehicleRentalCategoryID NUMBER(2) NOT NULL,
VehicleStatusID NUMBER(1) NOT NULL,
VehicleOwningAgencyID NUMBER(5)CHECK(VehicleOwningAgencyID BETWEEN 1 AND 10000),
VehicleCurrentLocationAgencyID NUMBER(5) CHECK(VehicleCurrentLocationAgencyID BETWEEN 1 AND 10000),
VehicleType CHAR(1) NOT NULL,

CONSTRAINT VehicleID_PK PRIMARY KEY(VehicleID),
CONSTRAINT TransmissionTypeID_FK FOREIGN KEY(TransmissionTypeID) REFERENCES TRANSMISSIONTYPE(TransmissionTypeID),
CONSTRAINT VehicleRentalCategoryID_FK FOREIGN KEY(VehicleRentalCategoryID) REFERENCES
VEHICLERENTALCATEGORY(VehicleRentalCategoryID),
CONSTRAINT VehicleStatusID_FK FOREIGN KEY(VehicleStatusID) REFERENCES VEHICLESTATUS(VehicleStatusID),
CONSTRAINT VehicleOwningAgencyID_FK FOREIGN KEY(VehicleOwningAgencyID) REFERENCES RENTALAGENCY(RentalAgencyID),
CONSTRAINT VehicleCurrentLocationAgencyID_FK FOREIGN KEY(VehicleCurrentLocationAgencyID) REFERENCES
RENTALAGENCY(RentalAgencyID)
);

CREATE TABLE RENTALINSURANCEOPTION
(
InsuranceOptionID NUMBER(1)PRIMARY KEY,
InsuranceOptionDesc VARCHAR2(150) NOT NULL,
InsuranceOptionAdditionalCost NUMBER(2,2) NOT NULL
```

```
);

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

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

CREATE TABLE RENTAL
(
RentalAgrementID NUMBER PRIMARY KEY,
CustomerID NUMBER NOT NULL,
VehicleRentalCategoryID NUMBER(2) NOT NULL,
RentalPickUpRentalAgencyID NUMBER(5) NOT NULL CHECK(RentalPickUpRentalAgencyID BETWEEN 1 AND 10000),
RentalDropOffRentalAgencyID NUMBER(5)NOT NULL CHECK(RentalDropOffRentalAgencyID BETWEEN 1 AND 10000),
VehicleID NUMBER NOT NULL,
ReservationID NUMBER(4) NOT NULL,
RentalPickUpDate DATE NOT NULL,
RentalPickUpTime NUMBER(4) NOT NULL CHECK(RentalPickUpTime BETWEEN 0 AND 2400),
RentalDropOffDate DATE NOT NULL,
RentalDropOffTime NUMBER(4) NOT NULL CHECK(RentalDropOffTime BETWEEN 0 AND 2400),
RentalPickUpOdometerValue NUMBER(5,2) NOT NULL,
RentalDropOffOdometerValue NUMBER(5,2) NOT NULL,
FuelOptionID NUMBER(1) NOT NULL,
InsuranceOptionID  NUMBER(1) NOT NULL,
```

```
RentalStatusID NUMBER(1) NOT NULL,

FOREIGN KEY(CustomerID) REFERENCES CUSTOMER(CustomerID),
FOREIGN KEY(VehicleRentalCategoryID) REFERENCES VEHICLERENTALCATEGORY(VehicleRentalCategoryID),
FOREIGN KEY(RentalPickUpRentalAgencyID) REFERENCES RENTALAGENCY(RentalAgencyID),
FOREIGN KEY(RentalDropOffRentalAgencyID) REFERENCES RENTALAGENCY(RentalAgencyID),
FOREIGN KEY(VehicleID) REFERENCES VEHICLE(VehicleID),
FOREIGN KEY(ReservationID) REFERENCES RESERVATION(ReservationID),
FOREIGN KEY(FuelOptionID) REFERENCES FUELOPTION(FuelOptionID),
FOREIGN KEY(InsuranceOptionID) REFERENCES RENTALINSURANCEOPTION(InsuranceOptionID),
FOREIGN KEY(RentalStatusID) REFERENCES RENTALSTATUS(RentalStatusID)
);

CREATE TABLE EMPLOYEEUSERACCOUNT
(
UserAccountID RAW(16) PRIMARY KEY,
UserName VARCHAR2(50) UNIQUE NOT NULL,
Password VARCHAR2(50)NOT NULL,
Email VARCHAR2(50) UNIQUE NOT NULL
);

CREATE TABLE EMPLOYEE
(
EmployeeID NUMBER(5) PRIMARY KEY,
FistName VARCHAR2(50) NOT NULL,
LastName VARCHAR2(50) NOT NULL,
SSNumber VARCHAR2(11) UNIQUE NOT NULL,
BirthDate DATE NOT NULL,
AdressLine1 VARCHAR2(30) NOT NULL,
AddressLine2 VARCHAR2(5) NULL,
City VARCHAR2(25) NOT NULL,
```

```
StateCode CHAR(2) NOT NULL,
ZipCode VARCHAR2(10) NOT NULL,
Country VARCHAR2(100) NOT NULL,
EmployeePhone VARCHAR2(20) NOT NULL,
EmployeeJobTitle VARCHAR2(30) NOT NULL,
EmployeeEmail VARCHAR2(50) UNIQUE NOT NULL,
UserAccountID RAW(16) NOT NULL,

FOREIGN KEY(UserAccountID) 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(50) NOT NULL
);

CREATE TABLE TRANSPORT
(
EmployeeID NUMBER(5) NOT NULL,
VehicleID NUMBER NOT NULL,
RentalAgencyID NUMBER(5) NOT NULL,
DropOffRentalAgencyID NUMBER(5) NOT NULL,
DriverDepartureDate DATE NOT NULL,
DriverDepartureTime NUMBER(4) NOT NULL,
```

```
VehiclePickUpDate DATE NULL,
VechiclePickUpTime NUMBER(4)NOT NULL,
VehicleDroppOffDate DATE NULL,
VehicleDropOffTime NUMBER(4) NOT NULL,
TransportStatusID NUMBER(1) NOT NULL,
TransportReasonID NUMBER(1) NOT NULL,

CONSTRAINT COMPOSITE_KEYS PRIMARY KEY(EmployeeID,VehicleID,RentalAgencyID),
CONSTRAINT DropOffRentalAgencyID_FK FOREIGN KEY(DropOffRentalAgencyID) REFERENCES RENTALAGENCY(RentalAgencyID),
CONSTRAINT TransportStatusID_FK FOREIGN KEY(TransportStatusID) REFERENCES TRANSPORTSTATUS(TransportStatusID),
CONSTRAINT TransportReasonID_FK FOREIGN KEY(TransportReasonID) REFERENCES TRANSPORTREASON(TransportReasonID)
);

CREATE TABLE CAR
(
VehicleID NUMBER PRIMARY KEY,
TrunkCapacity NUMBER(2,2) NOT NULL
);

CREATE TABLE SUV
(
VehicleID NUMBER PRIMARY KEY,
TowingCapacity NUMBER(5,2) NOT NULL,
IsAWD NUMBER (1)NOT NULL CHECK(IsAWD BETWEEN 0 AND 1)
);

CREATE TABLE CARGOVAN
(
VehicleID NUMBER PRIMARY KEY,
CargoCapacity NUMBER(3,2) NOT NULL,
MaximunPayLoad NUMBER(4,3) NOT NULL
```

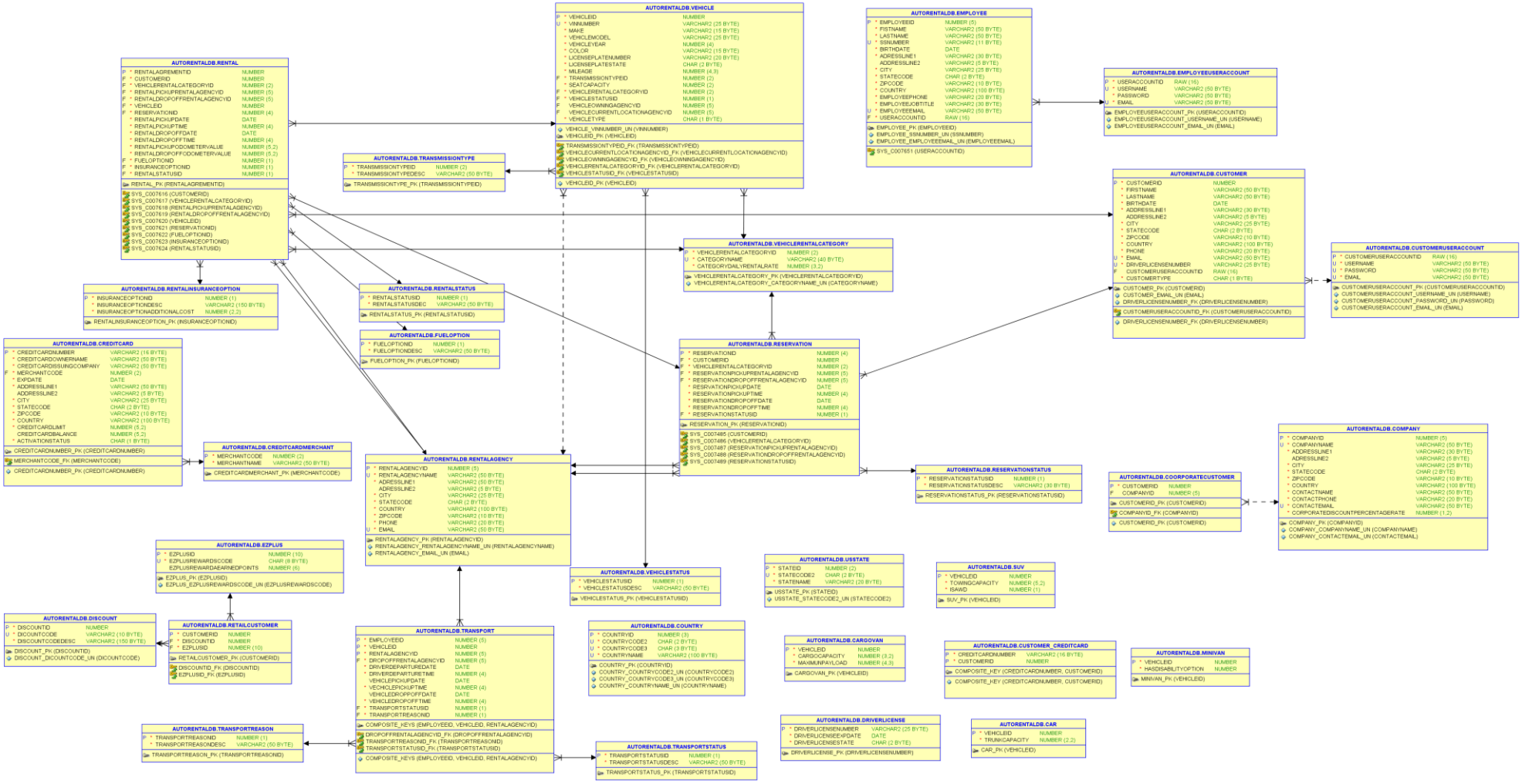
```
);

CREATE TABLE MINIVAN
(
VehicleID NUMBER PRIMARY KEY,
HasDisabilityOption NUMBER NOT NULL,
CHECK(HasDisabilityOption BETWEEN 0 AND 1)
);

CREATE TABLE USSTATE
(
StateID NUMBER(2)PRIMARY KEY,
StateCode2 CHAR(2) UNIQUE NOT NULL,
StateName VARCHAR2(20) NOT NULL,
CHECK(StateID BETWEEN 1 AND 56)
);

CREATE TABLE COUNTRY
(
CountryID NUMBER(3) PRIMARY KEY,
CountryCode2 CHAR(2)UNIQUE NOT NULL,
CountryCode3 CHAR(3) UNIQUE NOT NULL,
CountryName VARCHAR2(100) UNIQUE NOT NULL,
CHECK(CountryID BETWEEN 1 AND 200)
);
```

Database implementation Deliverable #6 –Implemented physical schema diagram

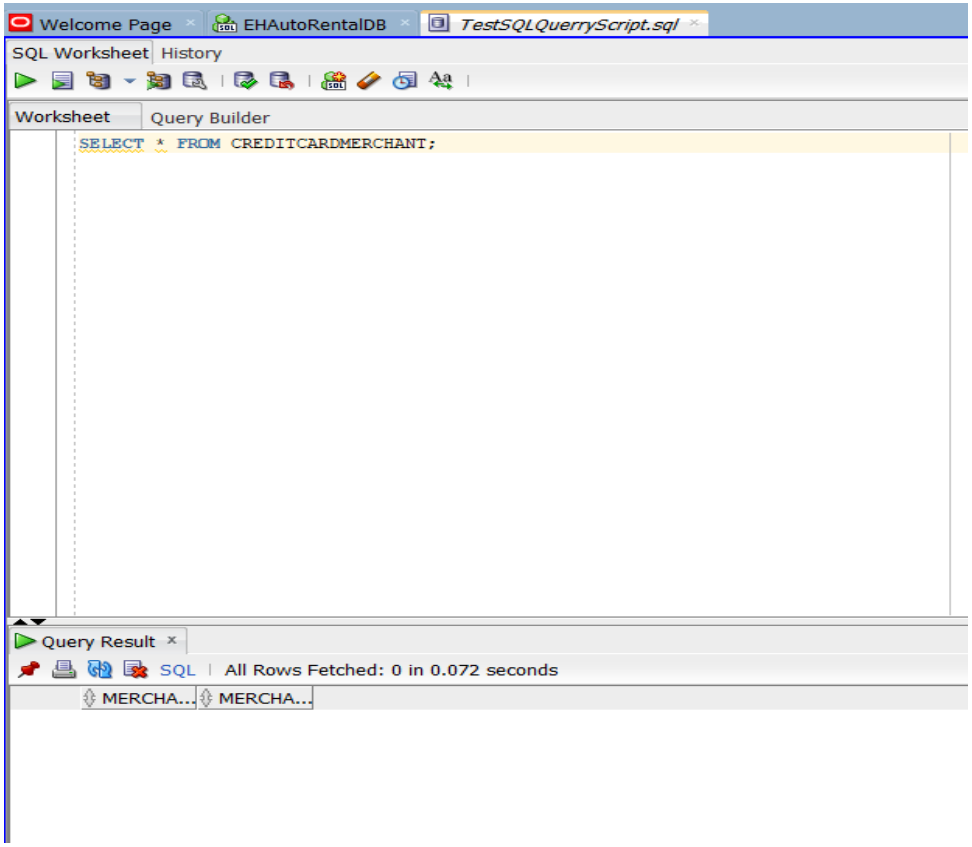


- *Above we can see how the thirty-three tables are fully implemented with their attributes and relationships.*

Database Implementation Deliverable #7 – Database Validation Unit Testing

Insert Statement for CREDITCARDMERCHANT TABLE:

- The following SQL statements were used for inserting data into the CREDITCARDMERCHANT table. The table consisted of two major columns. MerchantCode and MerchantName. Therefore, these two columns were populated with eleven rows as indicated in the business requirements.
  SELECT * FROM CREDITCARDMERCHANT before inserting statements:



Database Implementation Deliverable #7 – Database Validation Unit Testing

Database Implementation Deliverable #7 – Database Validation Unit Testing

Insert Statement for CREDITCARDMERCHANT TABLE:

```
INSERT INTO CREDITCARDMERCHANT (MerchantCode, MerchantName)
VALUES          (1,     'Stax By Fattmerchant');

INSERT INTO CREDITCARDMERCHANT (MerchantCode, MerchantName)
VALUES          (2,     'Helcim');

INSERT INTO CREDITCARDMERCHANT (MerchantCode, MerchantName)
VALUES          (3,     'Dharma Merchant Services');

INSERT INTO CREDITCARDMERCHANT (MerchantCode, MerchantName)
VALUES          (4,     'Payment Depot');

INSERT INTO CREDITCARDMERCHANT (MerchantCode, MerchantName)
VALUES          (5,     'National Processing');

INSERT INTO CREDITCARDMERCHANT (MerchantCode, MerchantName)
VALUES          (6,     'Block');
```

INSERT INTO CREDITCARDMERCHANT (MerchantCode, MerchantName)

```
VALUES          (7,     'Intuit Quickbooks');

INSERT INTO CREDITCARDMERCHANT (MerchantCode, MerchantName)
VALUES          (8,     'PayPal');

INSERT INTO CREDITCARDMERCHANT (MerchantCode, MerchantName)
VALUES          (9,     'Stripe');

INSERT INTO CREDITCARDMERCHANT (MerchantCode, MerchantName)
VALUES          (10,     'Flagship Merchant Services');

INSERT INTO CREDITCARDMERCHANT (MerchantCode, MerchantName)
VALUES          (11,     'Clover');
```

Database Implementation Deliverable #7 – Database Validation Unit Testing

- As we were to see, the insert statements were successfully implemented to the CREDITCARDMERCHANT TABLE. NOW the two columns MerchantName and MerchantCode have eleven rows of data.

```
SELECT * FROM CREDITCARDMERCHANT;
```

Script Output × Query Result ×

SQL | All Rows Fetched: 11 in 0.008 seconds

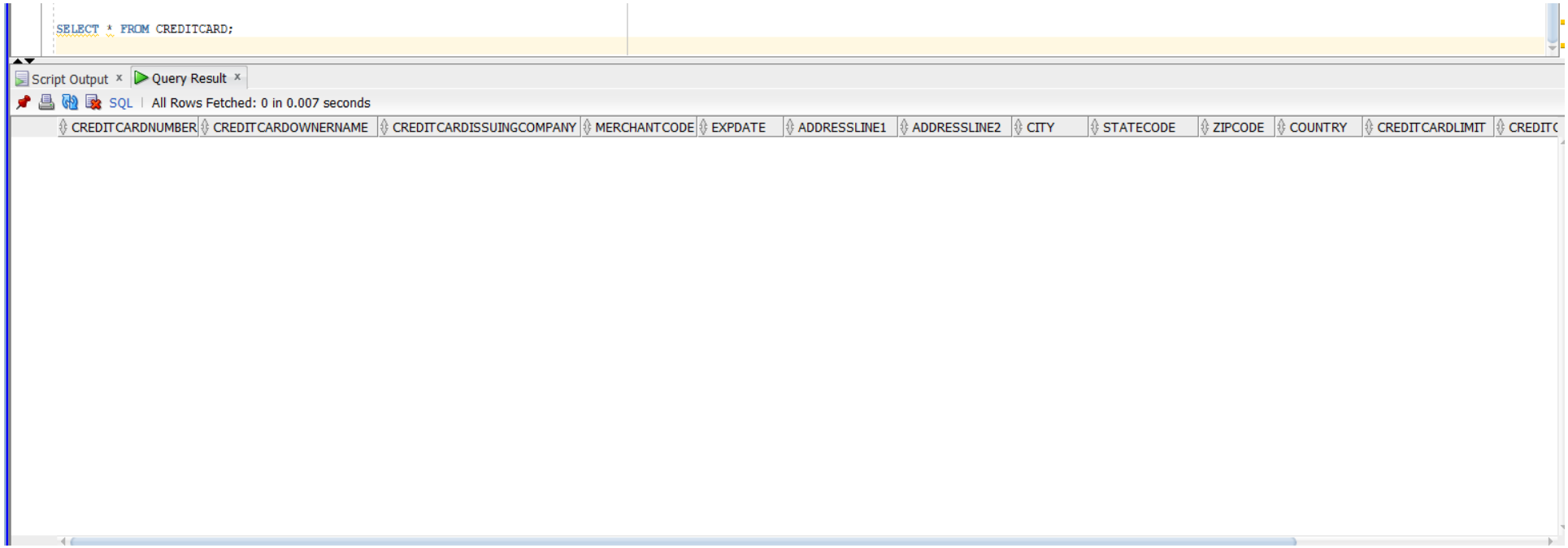
	MERCHANTNAME	MERCHANTCODE
1	Stax By Fattmerchant	1
2	Helcim	2
3	Dharma Merchant Services	3
4	Payment Depot	4
5	National Processing	5
6	Block	6
7	Intuit Quickbooks	7
8	PayPal	8
9	Stripe	9
10	Flagship Merchant Services	10
11	Clover	11

Database Implementation Deliverable #7 – Database Validation Unit Testing

INSERT Statements for CreditCard Table

- The CREDITCARD table had 14 columns without rows.

SELECT * FROM CREDITCARD before inserting statements.



Database Implementation Deliverable #7 – Database Validation Unit Testing

Insert STAMENTS for CREDITCARD Table

INSERT INTO CREDITCARD

Values(4207670249733778, 'Dan Smith', 'Visa', 1, '20-Nov-25', '8919 171 St', '4p', 'Astoria', 'NY', 11103, 'United States', 900.00, 500.00, 'y');

INSERT INTO CREDITCARD

Values(5411467127906009, 'John Doe', 'MasterCard', 2, '12-Mar-26', 'Austin St', '2L', 'Houston', 'TX', 77040, 'United States', 800.00, 700.00, 'y');

INSERT INTO CREDITCARD

Values(5565526038317988, 'Vivian Nelson', 'MasterCard', 3,'11-Jul-23', '3556 Michael St', '5R', 'Jamaica', 'NY', 11432, 'United States', 925.00, 790.00, 'y');

INSERT INTO CREDITCARD

Values(4952190062475420, 'Walter Justice', 'Visa', 4, '26-Dec-27', '3501 Euclid Avenue', '1Fl', 'Brooklyn', 'NY', 11201, 'United States', 999.00, 800.00, 'y');

Database Implementation Deliverable #7 – Database Validation Unit Testing

INSERT INTO CREDITCARD

Values(374275881739854, 'Joe Doyle', 'American Express', 5, '20-Jul-26', '4728 Formula Lane', '8j', 'LewisVille', 'TX', 13460, 'United States', 500.00, 100.00, 'y');

INSERT INTO CREDITCARD

Values(344028159416425, 'Joe Lang', 'American Express', 6, '20-Feb-22', '7656 Facilisis Ave 19', '1b', 'San Diego', 'CA', 55609, 'United States', 600.00, 550.00, 'n');

INSERT INTO CREDITCARD

Values(4246180721894901, 'Eva Wall', 'Visa', 7, '01-Jan-22', '992 Thompson Street', '9L', 'Irvine', 'CA', 92614, 'United States', 975.00, 190.00, 'n');

- The insert statements were successfully implemented into the CREDITCARD table as shown in the image below.

Database Implementation Deliverable #7 – Database Validation Unit Testing

```
SELECT * FROM CREDITCARD;
```

Script Output × | Query Result ×

SQL | All Rows Fetched: 7 in 0.007 seconds

	CREDIT...	CREDIT...	CREDIT...	MERCHA...	EXPDATE	ADDRES...	ADDRESSLINE2	CITY	STATEC...	ZIPCODE	COUNTRY	CREDIT...	CREDIT...	ACTIVA...
1	4207670...	Dan Smith	Visa	1	20-NOV-25	8919 17...	4p	Astoria	NY	11103	United ...	900	500	y
2	5411467...	John Doe	MasterCard	2	12-MAR-26	Austin St	2L	Houston	TX	77040	United ...	800	700	y
3	3742758...	Joe Doyle	America...	5	20-JUL-26	4728 Fo...	8j	LewisVille	TX	13460	United ...	500	100	y
4	3440281...	Joe Lang	America...	6	20-FEB-22	7656 Fa...	1b	San Diego	CA	55609	United ...	600	550	n
5	4246180...	Eva Wall	Visa	7	01-JAN-22	992 Tho...	9L	Irvine	CA	92614	United ...	975	190	n
6	5565526...	Vivian ...	MasterCard	3	11-JUL-23	3556 Mi...	5R	Jamaica	NY	11432	United ...	925	790	y
7	4952190...	Walter ...	Visa	4	26-DEC-27	3501 Eu...	1Fl	Brooklyn	NY	11201	United ...	999	800	y

Database Implementation Deliverable #7 – Database Validation Unit Testing

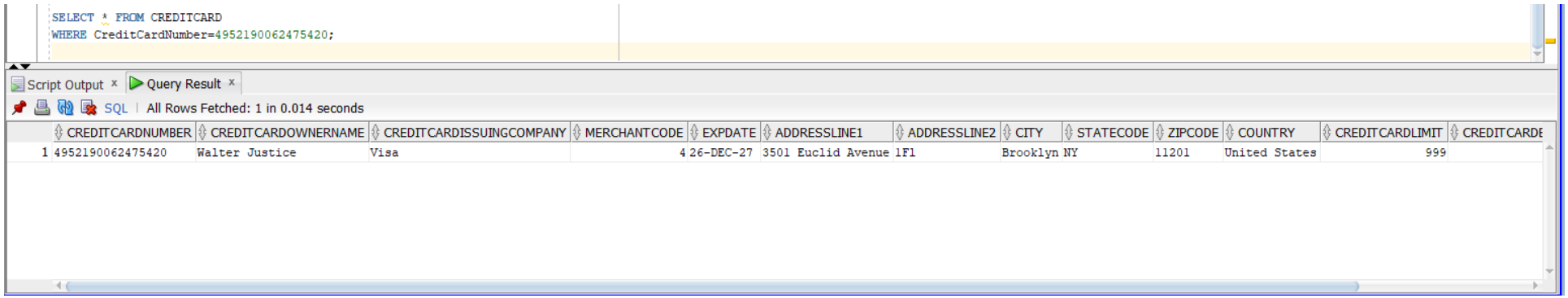
Select Statements that return records

- After both tables were successfully implemented with data using the INSERT statement. Both tables are now ready to retrieve information. This time we selected all records from the table CREDITCARD WHERE CreditCardNumber= 4952190062475420.

SELECT * FROM CREDITCARD

WHERE CreditCardNumber=4952190062475420;

- This SQL Statement returned only one record because it's based on PRIMARY KEY.

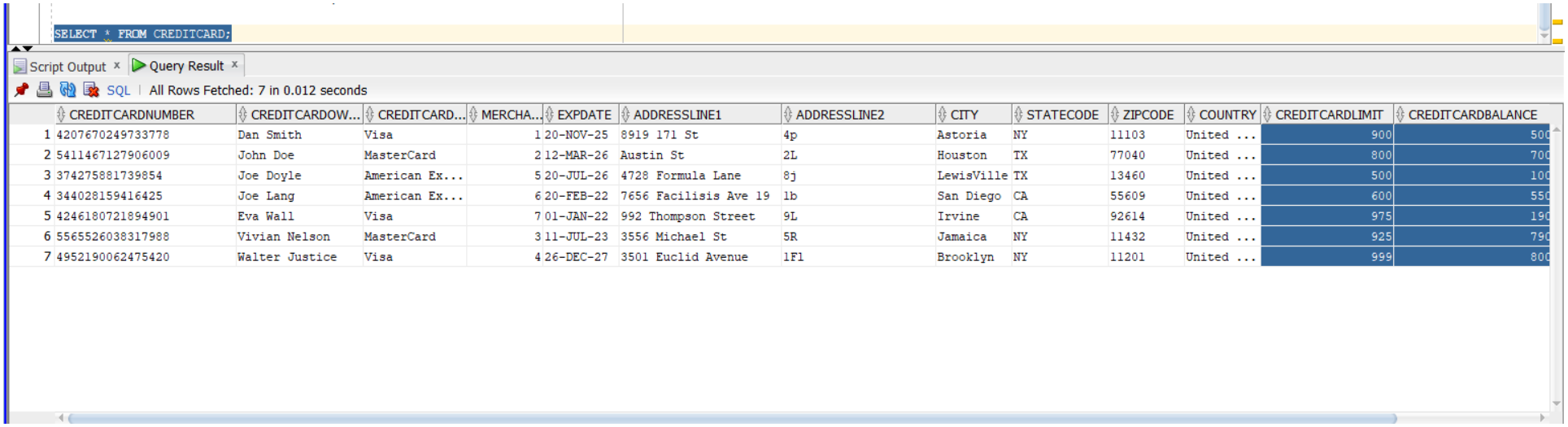


Database Implementation Deliverable #7 – Database Validation Unit Testing

UPDATE STATEMENT that updates records

- During the UPDATE statement one record was updated by modifying all its columns. This is was the CREDITCARD table before the UPDATE of a record.

SELECT * FROM CREDITCARD;



```
SELECT * FROM CREDITCARD;
```

Script Output | Query Result | SQL | All Rows Fetched: 7 in 0.012 seconds

	CREDITCARDNUMBER	CREDITCARDOW...	CREDITCARD...	MERCHA...	EXPDATE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY	CREDITCARDLIMIT	CREDITCARDBALANCE
1	4207670249733778	Dan Smith	Visa	1	20-NOV-25	8919 171 St	4p	Astoria	NY	11103	United ...	900	500
2	5411467127906009	John Doe	MasterCard	2	12-MAR-26	Austin St	2L	Houston	TX	77040	United ...	800	700
3	374275881739854	Joe Doyle	American Ex...	5	20-JUL-26	4728 Formula Lane	8j	LewisVille	TX	13460	United ...	500	100
4	344028159416425	Joe Lang	American Ex...	6	20-FEB-22	7656 Facilisis Ave 19	1b	San Diego	CA	55609	United ...	600	550
5	4246180721894901	Eva Wall	Visa	7	01-JAN-22	992 Thompson Street	9L	Irvine	CA	92614	United ...	975	190
6	5565526038317988	Vivian Nelson	MasterCard	3	11-JUL-23	3556 Michael St	5R	Jamaica	NY	11432	United ...	925	790
7	4952190062475420	Walter Justice	Visa	4	26-DEC-27	3501 Euclid Avenue	1F1	Brooklyn	NY	11201	United ...	999	800

Database Implementation Deliverable #7 – Database Validation Unit Testing

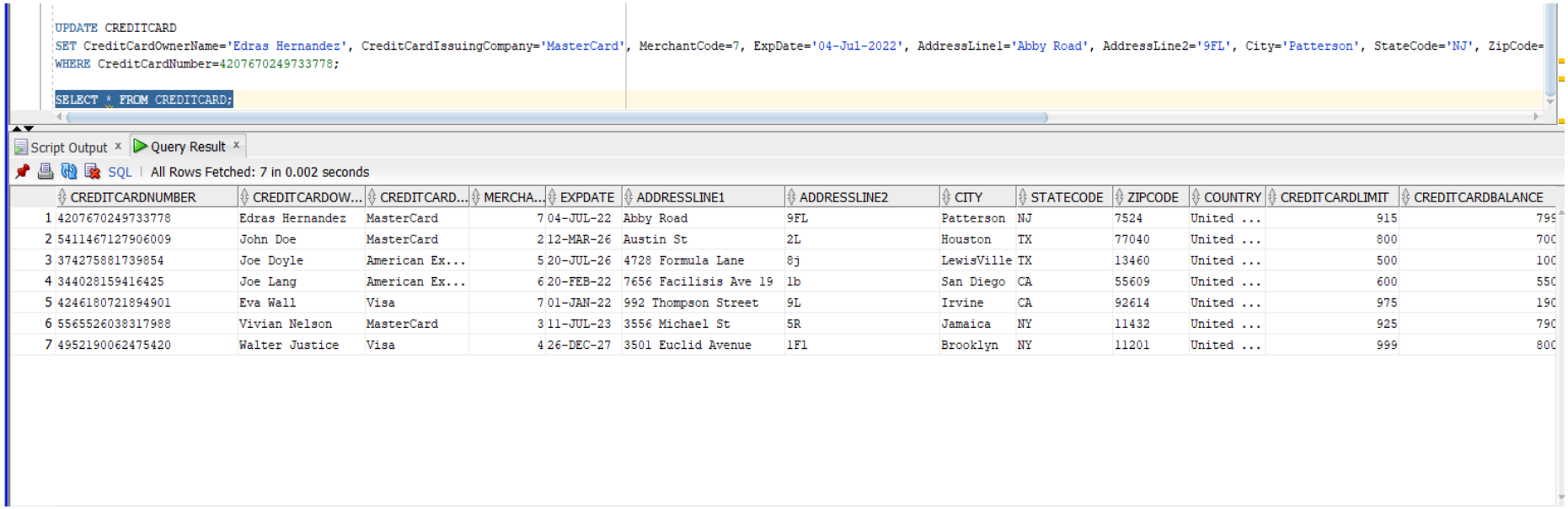
UPDATE STATEMENT that updates records

UPDATE CREDITCARD

SET CreditCardOwnerName='Edras Hernandez', CreditCardIssuingCompany='MasterCard', MerchantCode=7, ExpDate='04-Jul-2022', AddressLine1='Abby Road', AddressLine2='9FL', City='Patterson', StateCode='NJ', ZipCode=07524, Country='United States', CreditCardLimit=915.00, CreditCardBalance=799.00, ActivationStatus='N'

WHERE CreditCardNumber=4207670249733778

- This UPDATE statement updated all columns in the CREDITCARD table where The primary key= 4207670249733778. The UPDATE was successfully implemented as shown in the image below.



```
UPDATE CREDITCARD
SET CreditCardOwnerName='Edras Hernandez', CreditCardIssuingCompany='MasterCard', MerchantCode=7, ExpDate='04-Jul-2022', AddressLine1='Abby Road', AddressLine2='9FL', City='Patterson', StateCode='NJ', ZipCode=
WHERE CreditCardNumber=4207670249733778;

SELECT * FROM CREDITCARD;
```

Script Output | Query Result

SQL | All Rows Fetched: 7 in 0.002 seconds

	CREDITCARDNUMBER	CREDITCARDOW...	CREDITCARD...	MERCHA...	EXPDATE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY	CREDITCARDLIMIT	CREDITCARDBALANCE
1	4207670249733778	Edras Hernandez	MasterCard	7	04-JUL-22	Abby Road	9FL	Patterson	NJ	7524	United ...	915	799
2	5411467127906009	John Doe	MasterCard	2	12-MAR-26	Austin St	2L	Houston	TX	77040	United ...	800	700
3	374275881739854	Joe Doyle	American Ex...	5	20-JUL-26	4728 Formula Lane	8j	LewisVille	TX	13460	United ...	500	100
4	344028159416425	Joe Lang	American Ex...	6	20-FEB-22	7656 Facilisis Ave 19	1b	San Diego	CA	55609	United ...	600	550
5	4246180721894901	Eva Wall	Visa	7	01-JAN-22	992 Thompson Street	9L	Irvine	CA	92614	United ...	975	190
6	5565526038317988	Vivian Nelson	MasterCard	3	11-JUL-23	3556 Michael St	5R	Jamaica	NY	11432	United ...	925	790
7	4952190062475420	Walter Justice	Visa	4	26-DEC-27	3501 Euclid Avenue	1F1	Brooklyn	NY	11201	United ...	999	800

DELETE STATEMENT

- After the CREDITCARD table was successfully updated. We tested the DELETE statement in one of the records. Here's the CREDITCARD table before deleting a record.

```
SELECT * FROM CREDITCARD;
```

Script Output | Query Result

SQL | All Rows Fetched: 7 in 0.002 seconds

	CREDITCARDNUMBER	CREDITCARDOW...	CREDITCARD...	MERCHA...	EXPDATE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY	CREDITCARDLIMIT	CREDITCARDBALANCE
1	4207670249733778	Edras Hernandez	MasterCard	7	04-JUL-22	Abby Road	9FL	Patterson	NJ	7524	United ...	915	799
2	5411467127906009	John Doe	MasterCard	2	12-MAR-26	Austin St	2L	Houston	TX	77040	United ...	800	700
3	374275881739854	Joe Doyle	American Ex...	5	20-JUL-26	4728 Formula Lane	8j	LewisVille	TX	13460	United ...	500	100
4	344028159416425	Joe Lang	American Ex...	6	20-FEB-22	7656 Facilisis Ave 19	1b	San Diego	CA	55609	United ...	600	550
5	4246180721894901	Eva Wall	Visa	7	01-JAN-22	992 Thompson Street	9L	Irvine	CA	92614	United ...	975	190
6	5565526038317988	Vivian Nelson	MasterCard	3	11-JUL-23	3556 Michael St	5R	Jamaica	NY	11432	United ...	925	790
7	4952190062475420	Walter Justice	Visa	4	26-DEC-27	3501 Euclid Avenue	1Fl	Brooklyn	NY	11201	United ...	999	800

DELETE STATEMENT

- Here's the SQL statement that we used to delete a record from the table:

```
DELETE FROM CREDITCARD
WHERE CreditCardNumber=4246180721894901;
```

- The SQL statement was tested successfully and it deleted one record by using the primary key(In this case the (CreditCardNumber).

```
DELETE FROM CREDITCARD
WHERE CreditCardNumber=4246180721894901;

SELECT * FROM CREDITCARD;
```

Script Output | Query Result

SQL | All Rows Fetched: 6 in 0.005 seconds

	CREDITCARDNUMBER	CREDITCARDOW...	CREDITCARD...	MERCHA...	EXPDATE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATECODE	ZIPCODE	COUNTRY	CREDITCARDLIMIT	CREDITCARDBALANCE
1	4207670249733778	Edras Hernandez	MasterCard	7	04-JUL-22	Abby Road	9FL	Patterson	NJ	7524	United ...	915	799
2	5411467127906009	John Doe	MasterCard	2	12-MAR-26	Austin St	2L	Houston	TX	77040	United ...	800	700
3	374275881739854	Joe Doyle	American Ex...	5	20-JUL-26	4728 Formula Lane	8j	LewisVille	TX	13460	United ...	500	100
4	344028159416425	Joe Lang	American Ex...	6	20-FEB-22	7656 Facilisis Ave 19	1b	San Diego	CA	55609	United ...	600	550
5	5565526038317988	Vivian Nelson	MasterCard	3	11-JUL-23	3556 Michael St	5R	Jamaica	NY	11432	United ...	925	790
6	4952190062475420	Walter Justice	Visa	4	26-DEC-27	3501 Euclid Avenue	1F1	Brooklyn	NY	11201	United ...	999	800

- As we can see the column with such CreditCardNumber was deleted from the CREDITCARD table

Conclusion

EZ AUTORENTAL hired NYC-TECH to design and implement of Auto Rental Point-of-Sales Management System Applications that included EZ Rental POS. Their current rental system was outdated. Therefore, NYC-TECH Solutions decided to use two of the major Project Management Methodologies used in industry: The WATERFALL methodology & the AGILE methodology. This project was split and completed into four major phases.

During Phase1 a project plan was discussed and documented that later was sent to EZ AUTORENTAL.

During Phase2 an EER diagram was design that later became relational schema. Then datatype was decided for each table in the relational schema.

During phase 3 ORACLE SQL developer was used to create the tables and populate them according to the data dictionary.

Last but not least, during phase 4 NYC-TECH SOLUTIONS test the database inserting data into the tables. The way we made sure the database was working perfectly was by performing INSERT, SELECT, UPDATE AND DELETE statements.