Final Project Milestone 2

Nathan Brooks

Danyell Reno

**Part 1. Project Overview**

System Unified Parking Retrieval Accounting System (SUPRAS) is a relational database for a valet service called ValetPlus. ValetPlus is a company that allows customers to check in their car. When checking in, a customer can valet one and only one vehicle. A unique ValetID is assigned to keep track of details associated with the valet. Upon checking in the car, customers can choose from different Car Care Packages as well as what amenities they would like for their car in terms of parking.

Each Car Care Package is set to come with certain amenities. These amenities may be changed at a later date if management cares to do so. The price of each package is calculated by adding up all the amenities in a package. Packages are not customizable, that is, customers cannot add or omit services, it’s all or nothing.

Also, customers get to choose a spot for their car based on certain amenities. Each car is given a spot, within a lot. The spot can have different amenities within the lot. The lots are found in different locations. Each spot has different characteristics, such as covered or uncovered; these are the amenities of the spot.

Management wants to keep track of customer experience, so customers may write reviews about their experience in general. Their reviews will not be associated with a specific time they valeted their car. A customer may write no or many reviews. Management also gives discounts for various reasons, such as military discounts. Discounts are stackable.

Part 2. Data Specifications Overview

Part A:  List specific BUSINESS RULES Satisfied by the ERD and normal DB Constraints (PK, FK, DK, CK).

Checking In

When a customer checks in their car, they are assigned a unique ValetID. The ValetID keeps track of details about each valet transaction, that is, it keeps track of the Car Care package the customer chose, the Lot the car was put in, when the car was checked in and out, the CustomerID, any discounts that may be associated with the ValetID, the make and model of the vehicle, and whether or not it was a motorcycle, the valet status, which may be A for active, C for complete, or X for cancelled, and the spotID of where the car was put.

Customer Information

The CustomerID stored in the Valet table corresponds to as set of information about the customer. The customer information is recorded in a separate table called Customer. Each customer is assigned a unique CustomerID, and their first name, last name, full address, phone and email are kept track of.

Writing Reviews

Customers may write zero to many Reviews. The review is about the experience in general, not a specific time they valeted their car, that is a review is related to a CustomerID, not to a ValetID. Reviews are uniquely identified by a generated ReviewID and contain text, a 1 to 5 rating, and a Flag. The flag exists in case a customer gives a bad review so management can look into any problems the customer had with their service and try to resolve it. The flag is set to X by default for not applicable, in case of a good review. The flag set to V means that management took care of the bad review, that they addressed the customer concerns and fixed any problems. A flag set to L means that the customer gave a bad review and a management needs to look into how to fix it.

Car Care

Each ValetID is associated with a Car Care Package. The packages are set up to come with certain amenities. These amenities may be changed at a later date if management cares to do so. Each Car Care Package is assigned a unique Car Care Package ID. This ID is associated with Car Care Amenity ID numbers that have a lookup table associated with it to describe the amenity and give the price.

Lots and Spots

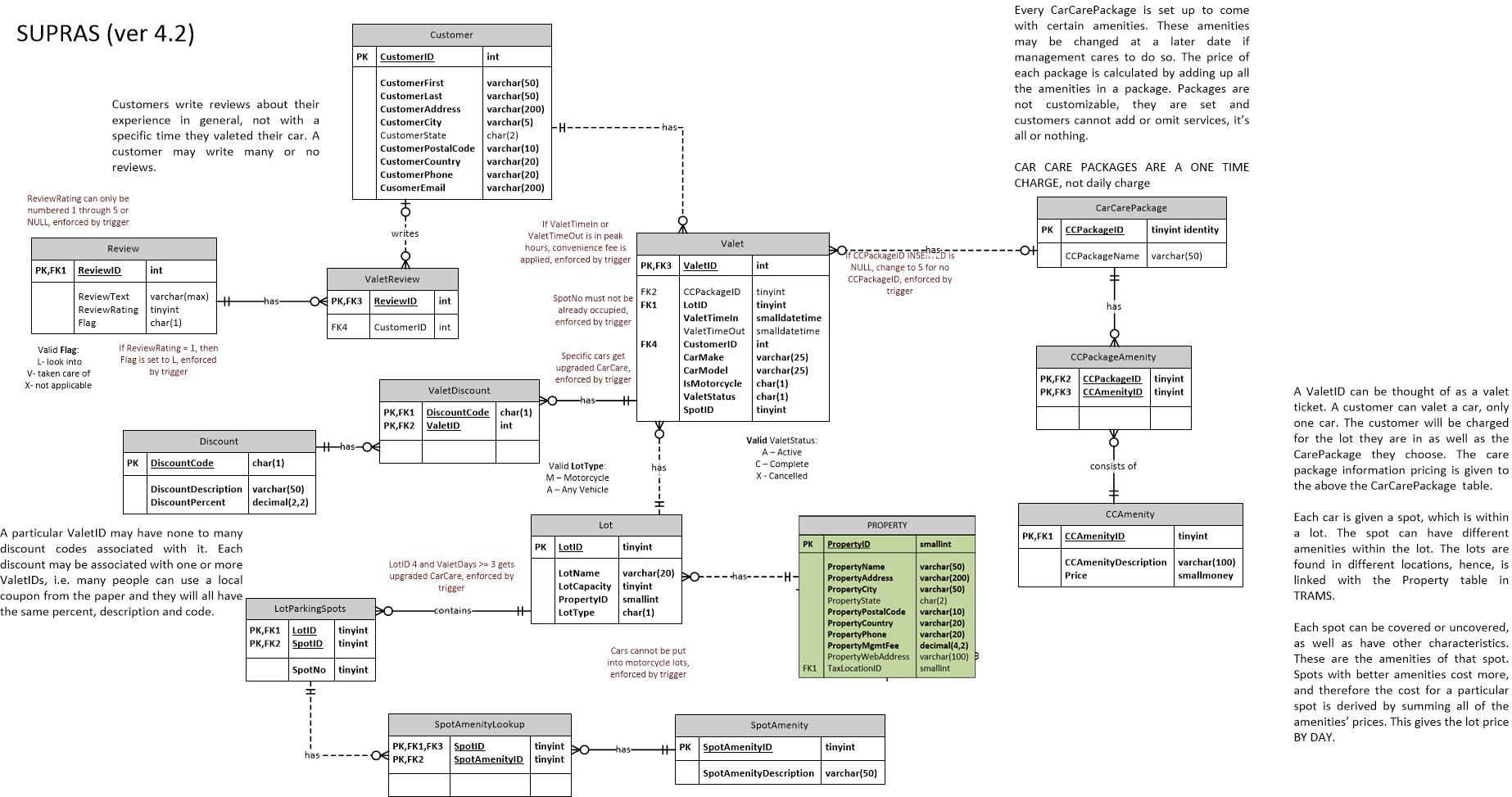
Cars are put into lots. Each LotID is associated with one and only one lot name, like “Waterfront Lot”, a lot capacity, like 50, a PropertyID, wich loosely links with the TRAMS database, and a lot type. A lot type can be either M for a motorcycle lot, or A for any vehicle. Each lot has specific SpotID’s associated with it. The spots are also numbered within a lot, like 1 to 100. Each spot has a set of amenities associated with it, and each amenity has one and only one associated description.

Discounts

A particular ValetID may have none to many discount codes associated with it. Each discount may be associated with one or may ValetIDs, i.e. many people can use a local coupon from the paper and they will all have the same percent, description, and code.

Charging Customers

Any cost or refund associated with SUPRAS will correspond to a new transaction type called “Valet Plus Charge.”

ERD  


**Part B:  List specific BUSINESS RULES Satisfied by TRIGGERS and outlines for each trigger you plan to program**

**The first 6 (numbered) are the ones we are most likely to do. The 7th is for enforcing referential integrity between databases. The others are backups in case of unforeseen circumstances arise.**

**1.**

**- Name: tr\_BadReview**

**- Type: INSERT (into REVIEW)**

**- Description: If a customer gives a bad review, i.e. gives a review rating of 1, then this trigger will force the Flag in the Review table to be set to L for Look Into. This will help management so they can keep their customers happy and correct problems.**

**2.**

**- Name: tr\_JdmUpgrade**

**- Type: INSERT (into VALET)**

**- Description: If a customer checks in any of the following: Nissan GTR, Toyota Supra, Subaru WRX STI, Mazda RX-7, or Honda NSX, they get an automatic upgrade to the highest Car Care package. No money is returned.**

**3.**

**- Name: tr\_ConvenienceFee**

**- Type: INSERT, UPDATE (into/on VALET)**

**- Description: If a customer checks in or out their car between certain hours, there is a convenience fee added. If it is between 7pm and 10pm, a $3 convenience fee is added, between 10pm and 2am, a $5 convenience fee is added, between 2 am and 6am, a $2 convenience fee is added. This is stackable. That is, a customer can be charged for checking in their car during these hours as well as checking it out during these hours.**

**4.**

**- Name: tr\_NewValet**

**- Type: INSERT, UPDATE (into/on VALET)**

**- Description: When a customer checks in their car, their car is parked in a unique spot that has a SpotID. This SpotID must not already have a vehicle in it, and if it does, this will prompt the user with an error that the desired spot is already taken.**

**5.**

**- Name: tr\_BigSpenderRebate**

**- Type: INSERT, UPDATE (into/on VALET)**

**- Description: If a customer spends over $100 in their parking services only, that is, if the price of the lot they were in times the total number of days their car was checked in is more than $100, the customer is given a rebate of 10% of this price. This is independent of the Car Care Package price. It only concerns the amount of money to actually park it.**

**6.**

**- Name: tr\_BadLot**

**- Type: INSERT (into VALET)**

**- Description: If a car is assigned to a Motorcycle lot, then an error is generated, and prompts the user to retry with one of the lots that is available to park cars and the transaction attempting to put the car in the motorcycle lot is rolled back.**

**7.**

**- Name: tr\_LinkWithTrams**

**- Type: INSERT (into LOT)**

**- Description: Each lot is linked with a PropertyID. PropertyID from LOT from SUPRA must correspond with PropertyID in PROPERTY from TRAMS.**

**(Below are backup triggers, as explained above.)**

**8.**

**- Name: tr\_RatingIsValid**

**- Type: INSERT (into REVIEW)**

**- Description: This was intentionally not enforced by check constraint to allow for this trigger to exist. ReviewRating must be a number 1 through 5. If the rating is not a number 1 through 5, an error message will be generated asking the user to insert a valid number and the transaction will be rolled back.**

**9.**

**- Name: tr\_NullCarCare**

**- Type: INSERT (into VALET)**

**- Description: This was intentionally allowed to be NULL in the script to allow for this trigger to exist. If the CarCareID column is left empty (NULL) on an insert, then the CarCareID should be changed to 5, which corresponds to no additional services.**

**10.**

**- Name: tr\_CheckOutCar**

**- Type: UPDATE (ValetTimeOut in VALET)**

**- Description: Will call both the convenience fee trigger and the stored procedures GenerateGratuity, GenerateDiscount, and GenerateTotalBill (IN THAT ORDER). It will charge the customer gratuity, give them rebates on the discounts they may have had, and then display all of the items that they will be charged for.**

**Part C/D Spocs and UDFs**

**1.**

**- Name: sp\_CalculateTax**

**- Description: Finds the tax for an inputted item. Similar to Calculate Tax in TRAMS. This will actually link through the Lot table, which each car has a part of, will call traverse to TRAMS, get the tax for the tax area the lot is in, then return this tax rate, then calculate tax for a charge. This helps management not get locked up by the IRS.**

**- Input Parameters: PropertyID (smallint), Price (smallmoney)**

**- Returns: Tax (smallmoney)**

**2.**

**- Name: sp\_GenerateTotalBill**

**- Description: Using a cursor, this loops through all the costs associated with a particular customer and displays them. Similar to the GenerateBill in TRAMS. This helps management with billing customers.**

**- Input Parameters: CustomerID (int)**

**- Returns: A displayed report**

**3.**

**- Name: sp\_GenerateLotStats**

**- Description: Finds the percentages, based on the cars currently in the system, ‘A’ for active ValetStatus, of which lot is most to least popular. Displays both the percentage of the lot that is filled up as well as the percentage that accounts for the number of cars in that lot versus the rest of the lots. This helps management decide which lots are the most popular and which lots need better advertising.**

**- Input Parameters: (none – this shows information for all lots in SUPRA)**

**- Returns: A displayed report**

**4.**

**- Name: sp\_GenerateCarCareStats**

**- Description: Finds the TOP THREE packages that are requested the most. This helps management better price their CarCarePackages and advertise what they need to and avoid advertising what they don’t need to.**

**- Input Parameters: (none – this shows information for top three)**

**- Returns: A displayed report**

**5.**

**- Name: sp\_GenerateAverageReviews**

**- Description: Finds the average review for SUPRA. This will not take into account any ReviewRatings with a Flag that is ‘L’ for look into, this will only take into account any incidents management has already taken care of or good reviews. This will artificially inflate the rating of SUPRA and will look good for online advertisement.**

**- Input Parameters: (none – takes into account all of the review ratings that are not Flagged ‘L’)**

**- Returns: Decimal (1,2)**

**6.**

**- Name: sp\_GenerateCustomerReviews**

**- Description: This will use a CURSOR to loop through all reviews, display the customer’s first name, last initial, review rating, and review text.**

**- Input Parameters: (none)**

**- Returns: A displayed report**

**7.**

**- Name: sp\_ListCareAmenities**

**- Description: Will use a CURSOR to loop through all of the amenities associated with the inputted CCPackageID. CCPackageID and CCPackageName will be displayed only once, but their amenities will be listed, CCAmenityID, CCAmenityDescription, and CCAmenityPrice will be listed for each different amenity associated with the Package.**

**- Input Parameters: CarCareID**

**- Returns: A displayed report**

**8.**

**- Name: sp\_GenerateGratuity**

**- Description: Based on the charges accrued by a customer, a progressive gratuity will be added. If the total cost is between 0 and $20, the gratuity will be 5% of the total bill, between $20 and $40, the gratuity will be 10%, between $40 and $60, the gratuity will be 15%, greater than $60, gratuity is 20%.**

**- Input Parameters: CustomerID (int)**

**- Returns: Gratuity (smallmoney)**

**9.**

**- Name: sp\_GenerateDiscount**

**- Description: Finds the total amount in discounts a customer gets, based on the CustomerID. The CustomerID is related to specific charges, the discounts are stackable. The discount is not taken off the original bill, but is put on as a rebate when the car is checked back out.**

**- Input Parameters: CustomerID**

**- Returns: Discount (smallmoney)**

**10.**

**- Name: sp\_ProblemReviews**

**- Description: Generates a report of all customers first name, last name, phone number, review text, and review rating that have a flag of ‘L’ for look into so management can see any bad reviews that have not been looked into thus far. The information displayed will allow management to easily contact disgruntled customers and see what they were upset about. A CURSOR will be used so the customer’s contact information is only generated once, in case one customer left multiple bad reviews.**

**- Input Parameters: (none – finds all reviews with flag ‘L’)**

**- Returns: A displayed report**