**DOCUMENTATION FOR EAS PROJECT**

**EAS: - E-Contracting Application System**

* It is not like that legacy (Paper Based Contracting) system.
* It is best made between parties who live in different places of the world and have to enter into an agreement or contract in easy way.

**Client: - Safeguard**

**OEM: - Original Equipment Manufacturer** (example:Audi,Yamaha,Hyundai ,Volkswagn etc.)

* OEM is called as program in our side.
* When customer purchase contract, the dealer is the one who purchase the contracts from the OEM via the respective portals for the customer.
* In order to make contracts, the OEM have tie up with Safeguard International which is our client.

Providing and Managing contracts: -

1. 1.EAS Team - E-Contracting Application System
2. 2.DMS Team -Dealer Management System

EAS Side:

* EAS side concentrates on SQL server and it is done by MayuraTech in USA.

DMS Side:

* DMS concentrates on Postgres SQL and it is done by iboss team in Delhi.

**Tables involved in EAS Side: -**

**1.Tables for Dealers:**

* On the dealer side we have many tables like Dealer, Dealer\_program\_association, Dealer\_address etc. We can get the DMS Dealer Number from the Dealer Table.
* **Dealer Program Association** table provides direct mapping between dealer and program (OEM).
* **Dealer\_Address** table contains columns like address, id, address\_type\_ID. For example, address\_type\_ID 1111 is which we can get from a table named **ref\_lookup\_value**.
* **Ref\_state\_province** table which contains the state province code, which tells the state to which the dealer belongs. Each state will be having a unique code to identify the state name (Like TN for Tamil Nādu).
* If we want to retrieve dealer’s informations like address, contact etc. it will take time since we have to fetch these details from different tables. So, for that purpose we have a **view** called **V\_dealer** which contains various columns from each of these dealer tables.

**Tables for VIN:**

* VIN stands for **Vehicle Identification Number which** is 17-digit number.
* VIN\_details table contain the VIN pattern, year, make (OEM), model, trim (extension of model), body\_type, vehicle\_type, fuel\_type, engine\_name, basic\_warranty\_terms(in months) etc.
* When the dealer enters the VIN in the portal, it will be passed as decoded. It will decode the VIN and get the required information for further processing. The 18-digits VIN will be decoded into 10 digits.

**Decoding Steps: -**

1. In 18-digit number, Remove the last 6 digits.
2. Leave next 2 characters as it is.
3. Remove the next.

Then we will get VIN Pattern like this (VIN Pattern: 1G1FB1RXM0)-it should be in 10 digits

* **VCI\_Vehicle\_retails\_sales** -it is another table in the VIN side. This table contain the columns like VIN (18 digits), sale date etc.
* Another important term is **ISD** which stands for **In Service Date**. ISD is the number of days the vehicle has been used on the road after purchase. It plays a major role in buying contracts. Like if we are buying a vehicle and not using it for the first month, and the service period is for 6 months means, then 6 months after that first month will be calculated for guarantee period, not for the first month (because we didn’t use that vehicle for first one month).

We have 2 types of teams in our system:

1. Rating Team

2. DMS (Dealer Management System) Team

* The Rate team will receive the requirements from OEM (business) the same, they will provide to EAS team. Also, they prepare the templates and the excel data which is called as the Rate file.
* This excel data is loaded to the original database by EAS team which is after verified by the DMS team.
* BA (Business Analyst) is the one who analyse the requirements which have been received from the business (OEM) and assign the task to team members.

**Tables for Product:**

* Product is something that is provided to the dealer based on their input given to the respective portal.
* **Ref-Product\_Type** from this table we can categorise the product. It is the main table to use by all OEM (program)to categorise the product. Every product type will have a unique code.
* Another table is **Ref\_Product\_Type\_Category** Based on the product type, we have different product category. In this table, for each product\_type\_id there will be corresponding product category.
* The table which contain the list of products under each OEM is **Ref\_Product\_code**.

**Tables for Product Plan:**

* **Product\_plan** table contains the different product plans or coverages for different products. Each product can have multiple number of product plans.
* Product\_plan\_code -Code for particular product plan.
* Product\_code\_id:It will be unique id.
* Name and Description: Name and description of particular product plan.
* Effective Date: It is the date from which we allow the dealer to purchase the product plan.
* Expiration Date: It is the date till which the plan is eligible. If we want to stop a particular plan, then we have to provide the expiration date in this table.
* PPM: Pre-Paid Maintenance. This is only used for PPM products.
* Split\_contract: This is for multi coverage plans.
* Is bundled: Whenever Split\_contract flag is Y, we should maintain is bundled also Y (Yes)
* No\_of\_intervals: This will be given in months.
* Is\_out\_of\_warranty: This means after warranty, you cannot purchase that plan.

**Vehicle condition:**

1. 1.New: New-vehicle.
2. 2.Used: Someone will have used that vehicle but not for a long time (Second hand, hand to hand).
3. 3.CPO (Certified Pre-Owner): In this, someone will verify the vehicle and will certify that the vehicle is fine to sell. (Second hand hand-agent (middle person)- hand)

* **Product\_plan\_association** table is the one which contains all the multi coverage plans.
* Parent\_product\_plan\_Id is one of the fields which gets the value from Product\_plan table.
* In EAS side when we purchase the contract which includes multi coverages (multiple plans), it will return only one contract number. But at the time of rating the contract in DMS side, we will get the multiple number of contract number based on child plans.

**SKU (stock Keeping Unit):**

It is Managed by OEM.

**Product\_plan\_sku** is one of the tables in SKU side.

5 parameters for SKU:

1. Term from: This is always taken in months. Indicates the date from which the plan is available.
2. Term to: The end date to which the plan is limited.
3. Deductible disappearing
4. Milage
5. Deductible amount (like discount)

If any one of the parameters is changed, then that will be considered as a new SKU. Combination of these 5 parameters define a SKU.

**Tables for SKU Eligibility:**

As we know, there are five parameters for SKU to define. As well as there are more parameters which are used to check the SKU eligibility which are stored in table named **Sku\_vsc\_eligibility** table.

* Product\_plan\_sku\_Id: This is a unique id which we get from Product\_plan table.
* Sku\_vsc\_eligibility\_id: get from sku\_vsc\_eligibility table.
* Odometer\_from: In the portal when we try to purchase a contract, you will be asked to enter the details on how many kilo meter you used this vehicle.
* Odometer\_to: This both together is the range.
* Vehicle\_age\_to and from: This is the date from the vehicle purchase date till the contract purchase date.
* Warranty\_fields

For some products like gap, cpo we don’t consider in warranty and out warranty.

**Finance\_fields:**

* Finance\_type: There are four types; Lease, balloon, finance and cash.
* Time\_months\_from and To: It is similar to term month from and to.
* Time\_days\_from and to: Based on the vehicle sale date and contract purchase date.
* Vehicle\_condition\_id: the value will be taken from Ref\_lookup table to identify the type of vehicle.
* Vehicle\_type: Like automobiles, motorvehicle. But most of the OEM who had tie up with SG is mostly for automobiles.
* Is\_eligible\_after\_sale: whether the product will be available even after the time of vehicle purchase.

**Surcharges (additional amount for the product):**

Surcharges and classing are not applicable for GAP products

Surcharges also decided by OEM.

**Tables for Classing:**

**[Classing – Categorizing]**

* Under each OEM there are different classification. For example, take AUDI OEM. Under AUDI, it has many models like A1, A2, A3…but for each of these models, the pricing will be different. Pricing differs for each make, model, trim and other parameter combinations. In EAS side we have different tables to store these values.
* **Ref\_make** table is the main table which contain all the makes like AUDI, ACURA, FORD etc. when the dealer enters the VIN in the respective portal, it will return information like make, model, trim etc of that vehicle. Then we can go to Ref\_make table and can search for that particular make. Similarly, we want make id, model id and trim id to get the different classing for pricing for that particular vehicle. We can get model id from **Ref\_model** table and trim id from **Ref\_trim\_level** table. We can have different models under same make id in Ref\_model table. These three are the master tables from which we can get the list of makes, models and list of trims for make, model combination.
* To find what classifications are available to the given make, model and trim, we use one classification method or function. The function takes **VIN, make, model and trim** as input and it give back all the classification available for that particular product.
* **Ref\_classing\_method** table contain classing\_method\_id, name and description. For example, if the classification name is **‘make’**, then for a single make, whatever model and trim it is, it will be showing the same amount meaning it will come under same classification. If the classification name is **‘yearmakemodel’**, it means classification done by these parameters year of the vehicle, make and model.
* **Ref\_classing\_method** is a reference table which can be referred for any program or product. From this table, we will be able to understand for a particular program and product, what will be the classing method. This table contains the mapping between Ref\_program, Ref\_product and Ref\_classing\_method tables. Same product can have multiple number of classing method under same OEM.
* **Program\_vehicle\_class** is another table that defines the program and class code.
* Next table **Program\_vehicle\_class mapping.** For example, if you got the classing method as make and you know the product and program, to understand what make is, we need this table. This is the main table for classing which has all the information.
* We have a view named **V\_ Program\_vehicle\_class** If in the view, the class code is ‘Excluded’ that means that product is not eligible for that make.

**Tables for Rating:**

* Depending on the state, the rate will be different for different products and different program.
* **Rate\_System** Rating will be based on dealer’s state. Rate System is the master table which contain the different rate systems for different products. This table contains information like for which program and for which product, what are the rate systems available. Some cases, for one program and one product, there will be multiple rate systems available.
* Another column in this table is the **regulated\_rate**. It can have values like **Non-Reg, Fixed, Reg and Reg-Max**. If the flag is non-Reg, it means it is used for MRP and the dealer will not be able to change the amount shown on the portal. Fixed and Non-Reg are same. Reg flag means it is used for some bargains for discount. If flag is Reg, then the dealer is allowed to change the cost in the portal. If flag is Max-Reg, then we can modify the amount but upto a limit specified.

* **Rate\_system\_state\_association** table contains the mapping between rates and systems. It contains for which all states what are the rate systems available. This table contains the rate\_system\_id and the corresponding **state\_province** to which the rate system belongs to.
* Next table is **Rate\_system\_application** which is the mapping between dealer and the rate system. It contains fields like Rate\_system\_id, Program\_id, Dealer\_id etc.
* For example, from this table we can understand that for Dealer\_id 5, there are 3 rate systems available. That 3 can belong to different products. There are fields like Sales\_effective and Sales\_expiration dates. If it has an expiration date specified and if the dealer is searching for that product in portal after the expiration date, then it will not be shown in the portal.

* Last table in the rating side is the **Dealer\_product\_plan\_exception**  It contains the details on the restriction on product plan for dealers, it will not allow to purchase all the product plans in certain cases. For a particular dealer\_id, what all Product\_plan\_id is there, that product plans will not be available to that dealer. (They will not be able to sell some of the plans under products) That exception is controlled based on the effective and expiration dates.

**Template load: -**

**Feed the data to excel sheets.**

Types:

* EAS Rate (In EAS there are 5 excel sheets to load the data into actual database one or more than one table)
* CFC
* Dealer
* PPM(Pre Paid Maintenance)
* Product Exclusion
* Program Water Vehicle Class

[Sheets will be differ based on the templates type]

1)**Product plan sku** –it is first excel sheet of EAS Rate template

* Product code table, Product plan table, Product plan sku tables are get data from this sheet.
* (Single excel sheet data will be passed to more than than one table in database)

2)**Client pricing sheet** This sheet contains huge data rather than other sheets.

* Through this sheet rates will be managed.
* (Here we need to combine Product plan sku and Classing Sheets)
* Rating sheets are populated after the client pricing sheet.

3)**Classing** (Example: for Yamaha Vehicle -->Yamaha Protection Plan-Make model Vehicle type, YMOT – Yamaha Motorcycle (Classing based on make and Vehicle type etc.)

4)**Rate System State** (Price will be differ based on the states)

5)**Rate System Mapping** (1. Dealer and product Mapping, 2. Dealer and Rate Mapping)

**Golden copy**

[Template should not be differed from the output]

* [Based on the template type it should take these data from database]
* Golden copy Parameters:
* exec getSKUByProgram 20336,'APGI','07/15/2021' -- Product Plan SKU
* exec getProgramVehicleClassing 20336,'APGI','07/15/2021' -- Classing
* exec getEASRateByProgramOrDealer 20336,'APGI','07/15/2021' -- Client Pricing
* exec getRateSystemMapping 20336,'APGI','07/15/2021' -- Rate System Mapping
* exec getRateSystemStateAssociation 20336,'APGI','07/15/2021' -- Rate System State