3.1 Create an ER diagram for Pre-owned dealer database, as described in the attached file.

3.1.1 A schema is an overall description of a database, and it is usually represented by the entity relationship diagram (ERD). Below is a list of items to consider during the design process of a database.

3.1.1.1 Entity: Real-world object or thing with an independent existence and which is distinguishable from other objects. E.g. car, customer, product, book etc.

3.1.1.2 Weak Entity: Weak entity is depends on strong entity to ensure the existence of weak entity. Like strong entity weak entity does not have any primary key, it has partial discriminator key. E.g. Professor is a strong entity, and the primary key is Professor ID. However, another entity is Professor_Dependents, which is our Weak Entity.

3.1.1.3 Attributes: An Entity is represented by a set of attributes (descriptive properties), e.g. name, age, price etc. Attribute values eventually become data stored in a database.

3.1.1.4 With each attribute a domain (data type) is associated, i.e., a set of permitted values for an attribute. Possible domains are integer, string, date, etc.

3.1.1.5 Entity Type: Collection of entities all having the same attributes, e.g. persons, cars etc.

3.1.1.6 Entity Set: Collection of entities of a particular entity type at any point in time.

3.1.1.7 Relationship: If entities are related, relationship exists between them.

3.1.2 For Pre-owned dealer database,

Please refer some of the observations below.

3.1.2.1 There are many unclear attributes, attribute notes occur twice.

3.1.2.2 Also, consider transaction 10123460, 10123457 both BUY and TRADE are set to 'Y". Ideally, buy/trade each represent unique transaction, but we cannot distinguish between buy/trade/sold in current design.

Hence, provided data has data quality issues. If you have bad data quality, you will not have good information quality. With bad information quality you will lack actionable knowledge in business operations. Hence, current data model needs to be corrected to provide more quality data which can provide precise details.

Below design precisely captures data provided in more effective way. After reprocessing the data with data curation activities like duplicate removal, normalization, cleansing, we can say that data is better prepared for final integration with assignment 1 design.

3.1.3 Detailed Description:

3.1.3.1 Customer Table: As customer physically exists, customer table is an entity with attributes. Customer name, phone number acts as unique key for the same. One customer can participate in multiple sales resulting one-to-many relationship to sales table.

3.1.3.1 Associate table: As associate physically exists, associate table is an entity with attributes. One associate can have multiple sales records.

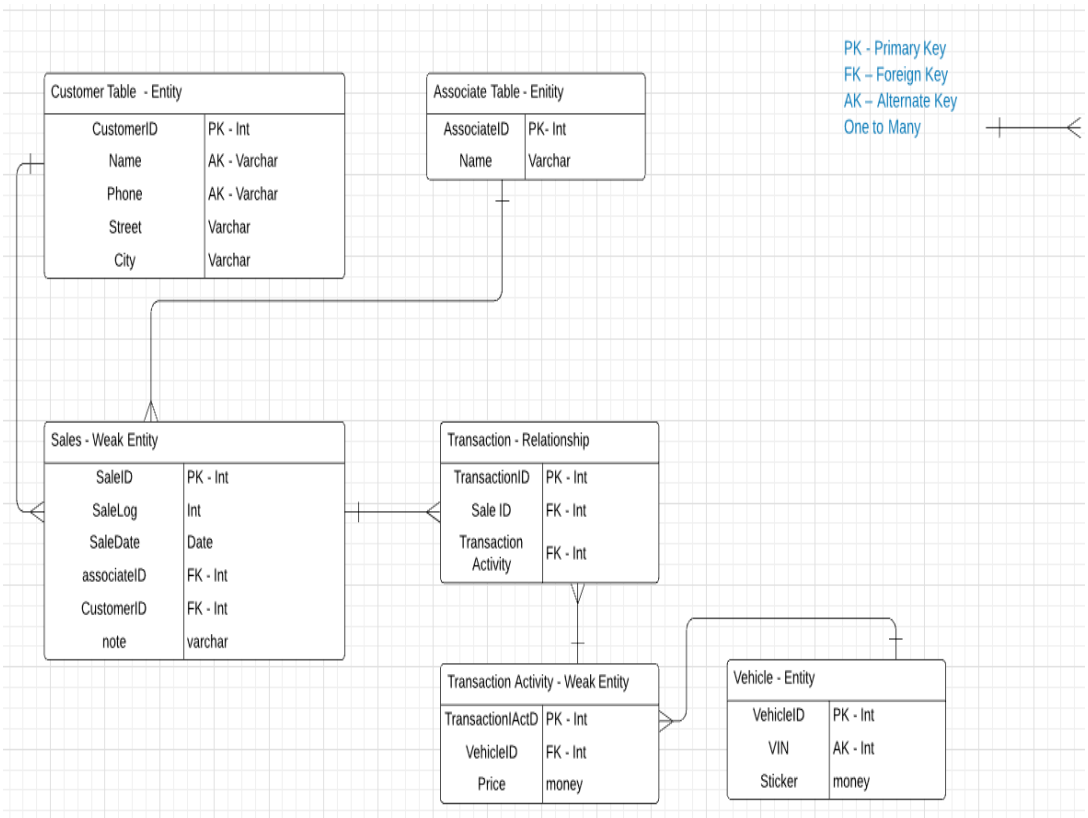
3.1.3.2 Vehicle table: As vehicle physically exists, vehicle table is an entity with attributes. Vehicle can be sold and bought multiple times. Hence, it can have multiple transactions records.

3.1.3.3 Transaction Activity: As salesperson will sell and buy vehicle, it will be recorded as a transaction with price attribute. It cannot exist on its own. Hence, this is a weak entity. As vehicle can have multiple transactions, this table may contain duplicates. But these are unique transactions and showcase vehicle history.

3.1.3.4 Sale Table: As it cannot exist without information from other tables. It is a weak entity.

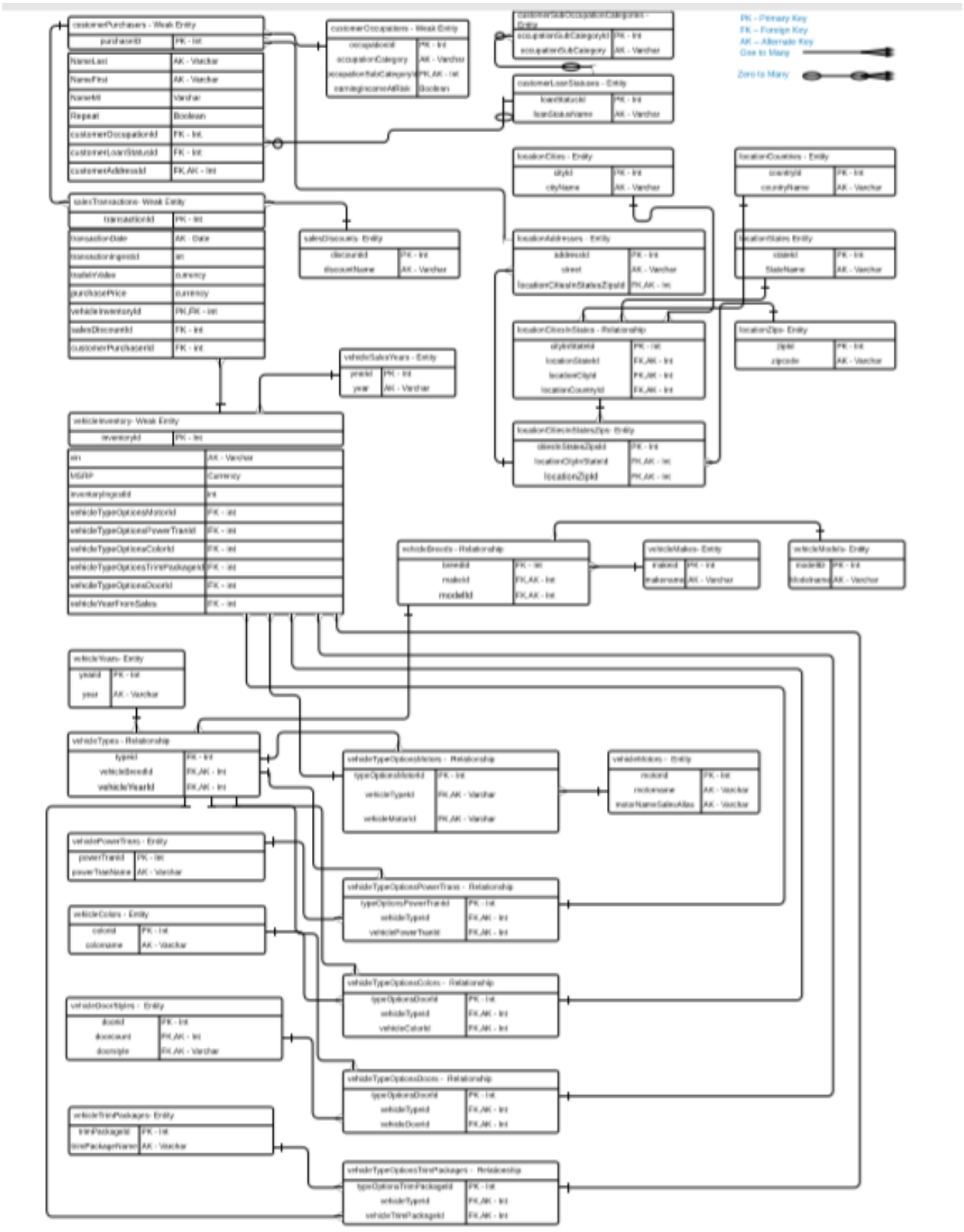
3.1.3.5 Sale Transactions table: This relationship table connects transactions with Sales. Instead of this table, one can consider foreign key constrain on transaction table, but it may not clearly distinguish between entities and relationship. Hence, I have considered it as a separate table.

3.1.3.6 Please refer below ERD or attached ***PDF 3.1_Ajinkya_Devi_Assignment_3_ERD***.



3.2 Create a separate ER diagram that reflects the schema you designed for Assignment 1. You may update the schema based on feedback you received from instructors after submitting Assignment 1.

3.2.1 Please refer below ERD for assignment one. I could not fit the same in screen print. If it is unclear, please refer attached ***PDF 3.2_Ajinkya_Devi_Assignment_1_ERD***.


PK - Primary Key
FK – Foreign Key
AK – Alternate Key
One to Many
Zero to Many
customerPurchasers - Weak Entity
purchaseID PK - Int
NameLast AK - Varchar
NameFirst AK - Varchar
NameMI Varchar
IRepeat Boolean
customerOccupationId FK - Int
customerLoanStatusId FK - Int
customerAddressId FK,AK - Int
customerOccupations - Weak Entity
occupationId PK - Int
occupationCategory AK - Varchar
occupationSubCategory FK,AK - Int
carriesIgniconsAtRisk Boolean
occupationSubCategories - Entity
occupationSubCategoryId PK - Int
occupationSubCategory AK - Varchar
customerLoanStatuses - Entity
loanStatusId PK - Int
loanStatusName AK - Varchar
locationCities - Entity
cityId PK - Int
cityName AK - Varchar
locationCountries - Entity
countryId PK - Int
countryName AK - Varchar
salesTransactions- Weak Entity
transactionId PK - Int
transactionDate AK - Date
transactionAgentId an
tradeInValue currency
purchasePrice currency
vehicleInventoryId FK,PK - int
salesDiscountId FK - Int
customerPurchaseId FK - Int
salesDiscounts- Entity
discountId PK - Int
discountName AK - Varchar
locationAddresses - Entity
addressId PK - Int
street AK - Varchar
locationCitiesInStatesZipsId FK,AK - Int
locationStates Entity
stateId PK - Int
StateName AK - Varchar
locationCitiesInStates - Relationship
cityInStateId PK - Int
locationStateId FK,AK - Int
locationCityId FK,AK - Int
locationCountryId FK,AK - Int
locationZips- Entity
zipId PK - Int
zipcode AK - Varchar
vehicleSalesYears - Entity
yearId PK - Int
year AK - Varchar
vehicleInventory- Weak Entity
inventoryId PK - Int
vin AK - Varchar
MSRP Currency
inventoryIngestId int
vehicleTypeOptionsMotorId FK - int
vehicleTypeOptionsPowerTrainId FK - int
vehicleTypeOptionsColorId FK - int
vehicleTypeOptionsTrimPackageId FK - int
vehicleTypeOptionsDoorId FK - int
vehicleYearFromSales FK - Int
locationCitiesInStatesZips- Entity
citiesInStatesZipsId PK - Int
locationCityInStateId FK,AK - Int
locationZipId FK,AK - Int
vehicleBrands - Relationship
brandId PK - Int
makeId FK,AK - Int
modelId FK,AK - Int
vehicleMakes- Entity
makeId PK - Int
makename AK - Varchar
vehicleModels- Entity
modelId PK - Int
Modelname AK - Varchar
vehicleYears- Entity
yearId PK - Int
year AK - Varchar
vehicleTypes - Relationship
typeId PK - Int
vehicleBrandId FK,AK - Int
vehicleYearId FK,AK - Int
vehicleTypeOptionsMotors - Relationship
typeOptionsMotorId PK - Int
vehicleTypeId FK,AK - Varchar
vehicleMotorId FK,AK - Varchar
vehicleMotors - Entity
motorId PK - Int
motorname AK - Varchar
motorNameSalesAlias AK - Varchar
vehiclePowerTrains - Entity
powerTrainId PK - Int
powerTrainName AK - Varchar
vehicleTypeOptionsPowerTrains - Relationship
typeOptionsPowerTrainId PK - Int
vehicleTypeId FK,AK - Int
vehiclePowerTrainId FK,AK - Int
vehicleColors - Entity
colorId PK - Int
colorname AK - Varchar
vehicleTypeOptionsColors - Relationship
typeOptionsDoorId PK - Int
vehicleTypeId FK,AK - Int
vehicleColorId FK,AK - Int
vehicleDoorOptions - Entity
doorId PK - Int
doorcount FK,AK - Int
doorstyle FK,AK - Varchar
vehicleTypeOptionsDoors - Relationship
typeOptionsDoorId PK - Int
vehicleTypeId FK,AK - Int
vehicleDoorId FK,AK - Int
vehicleTrimPackages- Entity
trimPackageId PK - Int
trimPackageName AK - Varchar
vehicleTypeOptionsTrimPackages - Relationship
typeOptionsTrimPackageId PK - Int
vehicleTypeId FK,AK - Int
vehicleTrimPackageId FK,AK - Int

3.3 Create 1 to 3 intermediate ER diagrams that showcase your integration process. These diagrams should be accompanied by narrative prose (either in a separate document or as annotations directly to the diagram) that describe each of the integration steps taken on the diagram. See the integration process described in the data integration slides for examples of what this might look like, and follow the example shown in the "Schema integration: an example" lecture. There is no one right way to do this, but your decisions should be justifiable, and should minimize the potential for information loss.

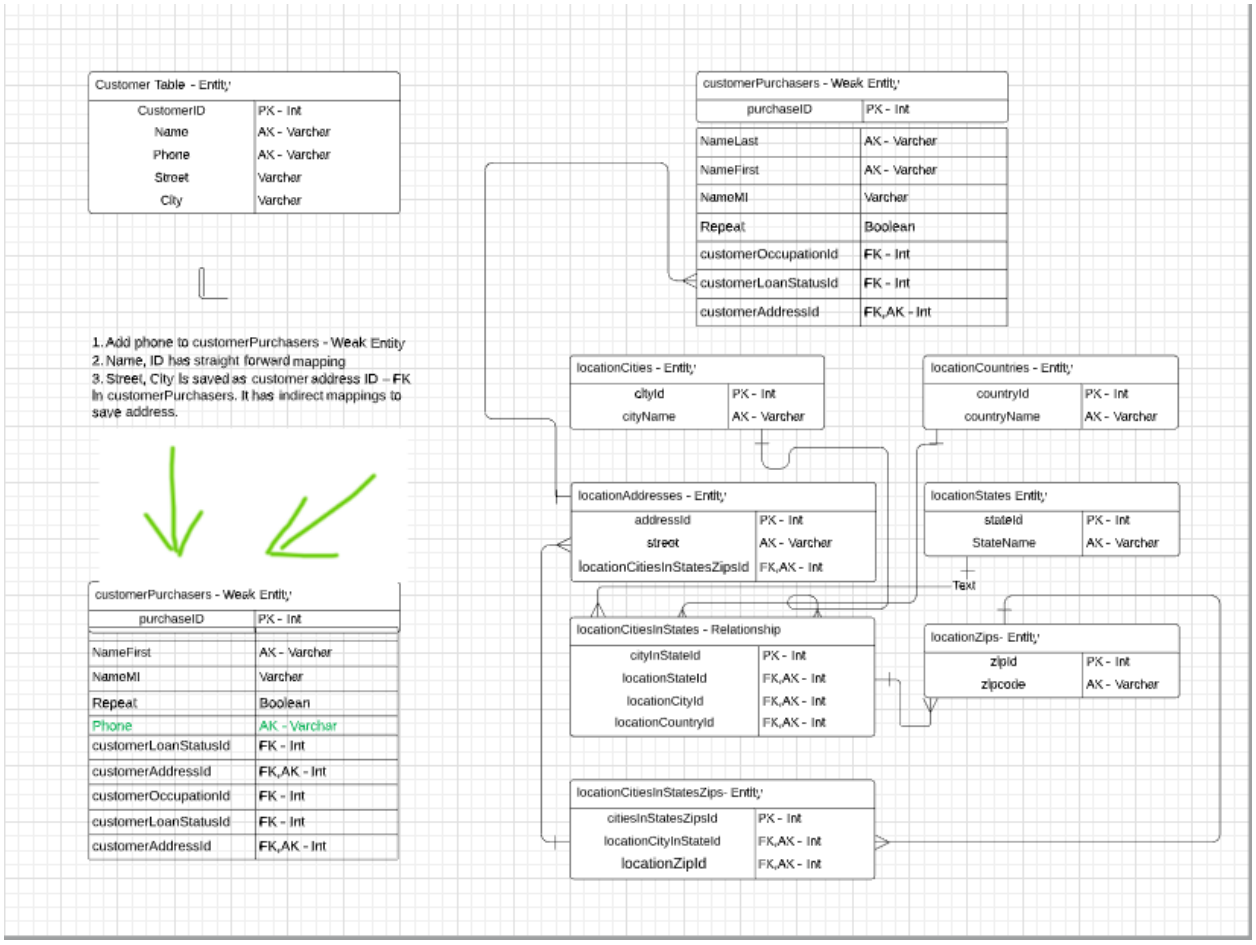
3.3.2 Integration (customer table -> customerPurchasers): Core part of sales DB and process is customer. Hence, 1st step could be integrating customer table into customerPurchasers table from assignment 1.

3.3.3.1 Both tables have customer first name and last name. In addition to name, customer table has phone number which should be added to customerPurchasers. The phone from the pre-owned car DB assignment 3 was included into the customerPurchasers table which will be null for the assignment 1, but it will avoid information loss.

3.3.3.2 Street, city attributes (which are part of address information in customer table) are saved as location address detail (separate independent entity) table in 1st assignment. Hence, we need to move street, city to location address table and reference it as foreign key in customerPurchasers.

3.3.3.3 With above process, we can say integration of customer table into customerPurchasers table is complete. Of course, we need to communicate and confirm these changes with preowned sales business team. Now we have single customerPurchasers table which can store details for both the assignments 1/3 efficiently and without any information loss.

3.3.3.4 Please refer below ERD or attached PDF
3.3.1_Ajinkya_Devi_Assignment_3_Integration_1_ERD



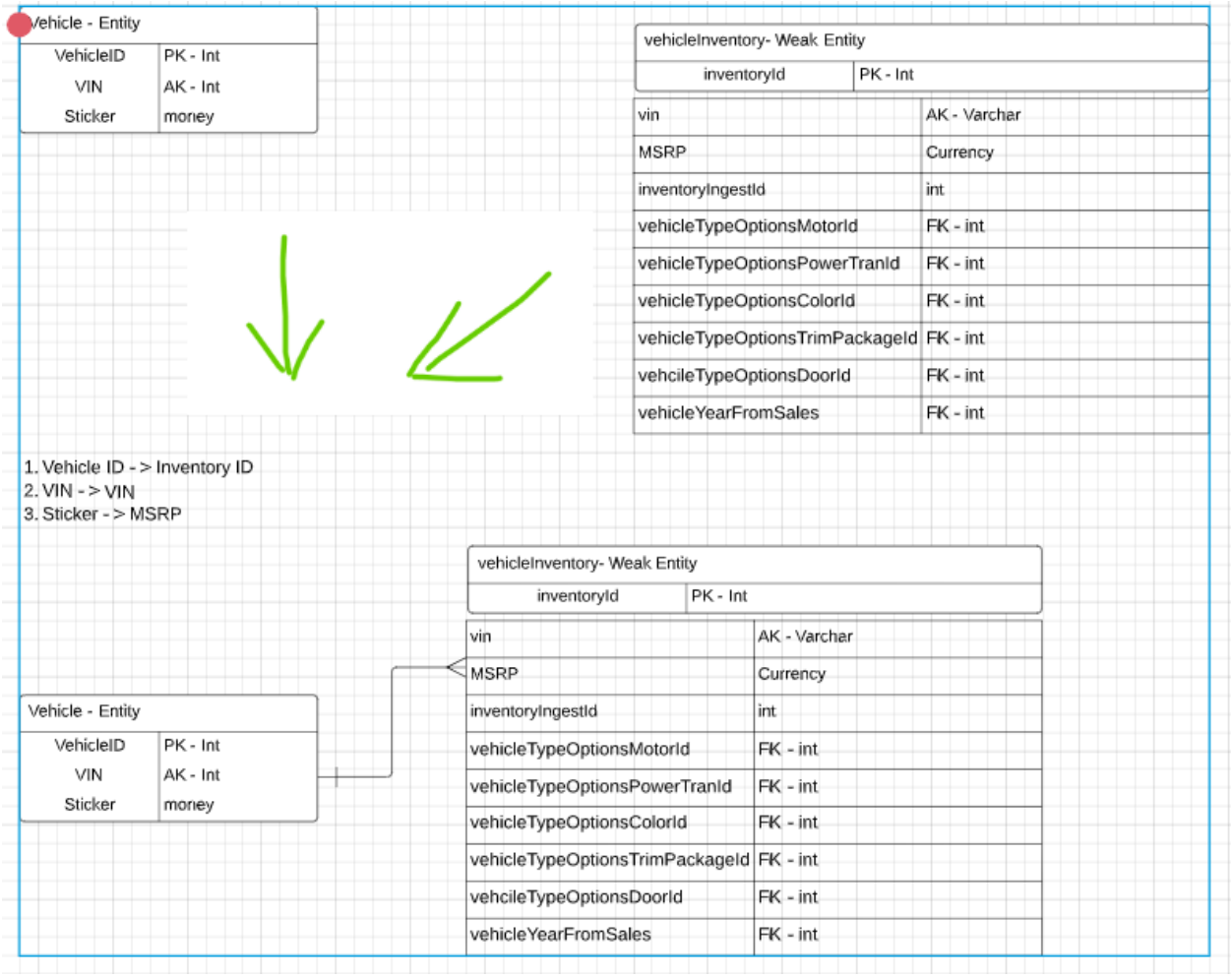
3.3.3 Integration (vehicle table -> vehicle Inventory): This is straight forward integration.

3.3.3.1 We can see clear mapping for vehicle ID and VIN from assignment 3 to inventory ID, VIN of assignment 1. Inventory Id seems more natural as compare to vehicle Id. Hence, I have kept inventory Id in integrated schema of assignment 1.

3.3.3.2 Same is case with Sticker and MSRP. MSRP is more natural match and is part final integrated schema.

3.3.3.3 Of course, we need to communicate and confirm these changes with preowned sales business team. Thus, we are able merge both tables efficiently and without information loss.

3.3.3.4 Please refer below ERD or attached PDF
3.3.2_Ajinkya_Devi_Assignment_3_Integration_2_ERD



3.3.4 Integration (Sales Tables): The last step is to merge sales information from both schemas. If we compare both schemas, we can find few common attributes but solution in assignment 1 was more specific to selling new cars. However, after the analysis of both the schemas, I was able to conclude that we need to design new common schema for both due to representational variations.

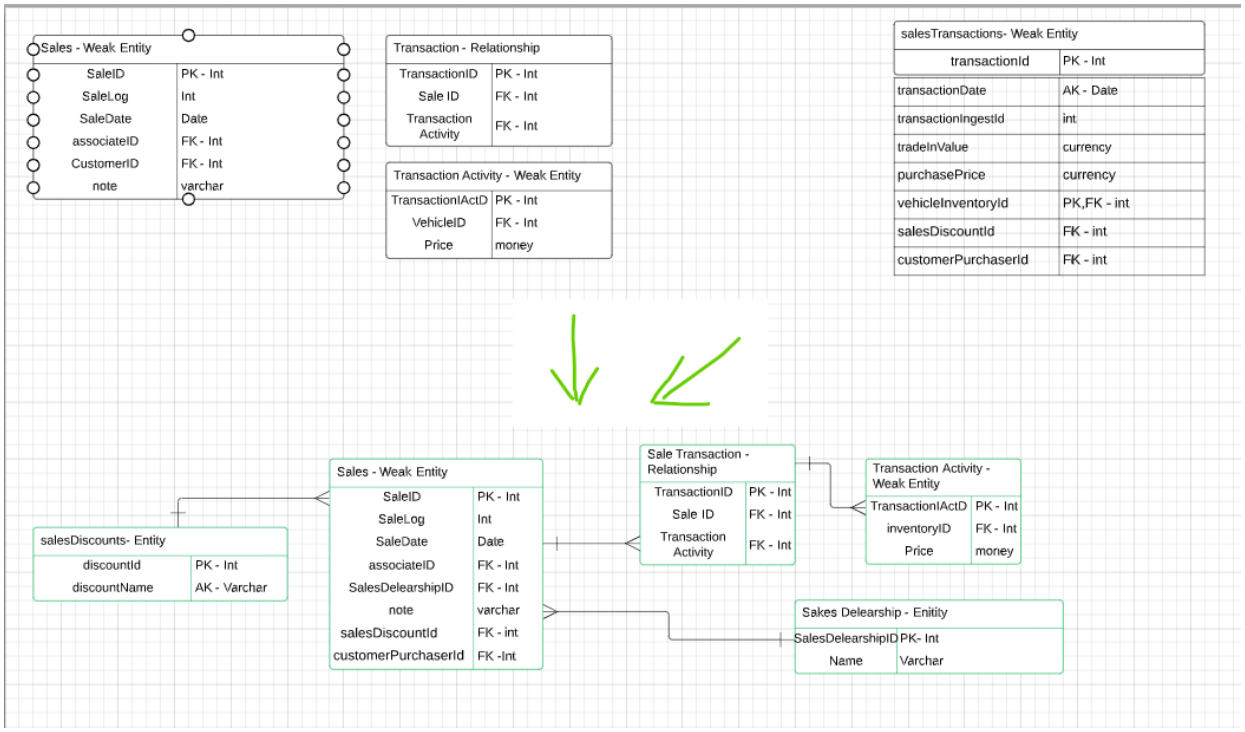
3.3.4.1 After identifying common attributes, I have added customer Purchaser Id to new sales table which connects with customer purchase table of assignment 1 without information loss.

3.3.4.2 Assignment 1 has unique requirement related to sales discount. Hence, it was added to new sales table and it connects with sales discount table. So even with new changes we can see that sales discount can still be applied.

3.3.4.3 Sales transaction relationship and weak entity sales transaction activity can be implemented exactly similar to assignment 3 without any information loss. This was specific requirement for assignment 3.

3.3.4.4 Addition of new sales dealership table ensures model is generic and can be implemented for multiple dealerships.

3.3.4.5 Please refer below ERD or attached PDF
3.3.3_Ajinkya_Devi_Assignment_3_Integration_3_ERD



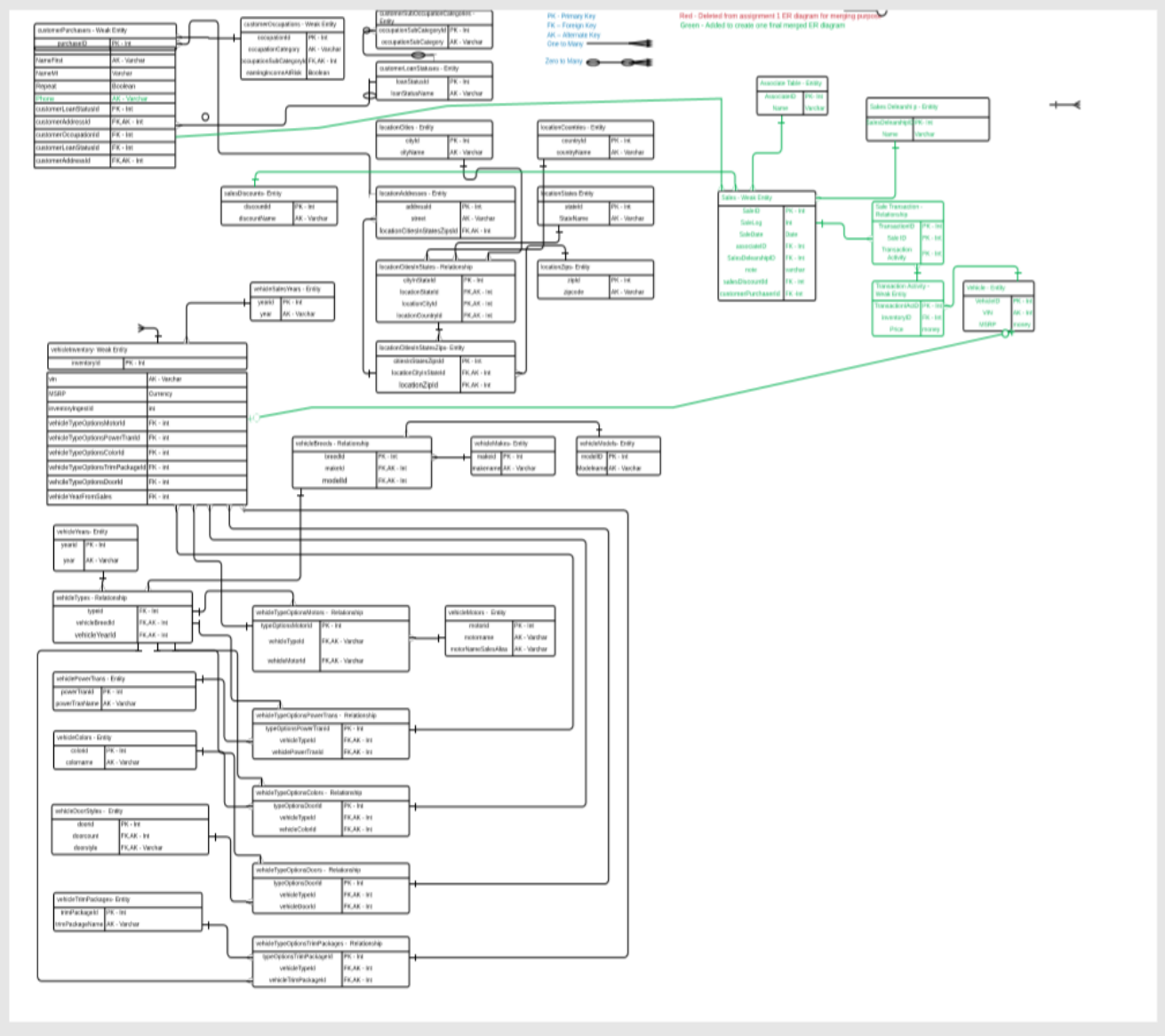
3.4 Finally, create one integrated ER diagram represented the merged schemas of the two dealerships (i.e., the final product of the integration process). Be sure to describe any final integration steps taken at this point (as described in step 3 above).

3.4.2 In additional to steps a, b and c of integration process (3.3) only additional task is to connect

3.4.2.1 Connect customer Purchasers with sales table

3.4.2.2 Vehicle inventory with vehicle table

3.4.3 Please refer below ERD. I could fit the same in screen print. If it is unclear, please refer attached **3.4_Ajinkya_Devi_Assignment_1_3_merged_ERD.**



3.5 Difficult decisions

3.5.2 One of the issues that I have mentioned above with pre-owned car DB is data quality. It was difficult to justify table design based on incomplete / low quality data. I would set up meeting with preowned car dealership to understand complete user requirements so that I can correctly define strong/weak entities and relationships. If you have bad data quality, you will not have good information quality. With bad information quality you will lack actionable knowledge in business operations. Good quality data, non-redundant data is foundation of data curation activities.

3.5.3 Another challenge was merging the 2 schemas. We cannot have 2 columns for same purpose (Sticker/MSRP). Hence, we need to ensure that we understand significance of every column and design solution which is agreeable to both parties (assignment 1 car dealership and pre owned car dealership). I would setup a meeting with both

parties to ensure requirement understanding and use most significant terms in our solution.

3.6 Future Considerations

3.6.2 One of the missing aspects in above implementation of assignment 3/1 would security and compliance. If we consider CRM like Salesforce, we can easily address security issues with features like profiles, permission sets etc. Business rules/state related rules should be modified and applied according new model. This will be part of technology review board meeting of the project as it varies with technology. It is easier to implement security and compliance with CRM tools however it would require substantial amount effort with traditional technologies like Java.

3.6.3 Business process of both assignment 1 dealership and assignment 3 dealership may need certain changes according to new solution.

3.6.4 Also, every business creates different reports, dashboards. With schema design change, we need to change the way reports and dashboards are generated. This should be part of design review meeting.

3.7 References

3.7.2 Tools https://www.lucidchart.com/
3.7.3 CS 598: Foundations of Data Curation
3.7.4 https://www.wikipedia.org
3.7.5 https://www.tutorialspoint.com/dbms/er_diagram_representation.htm