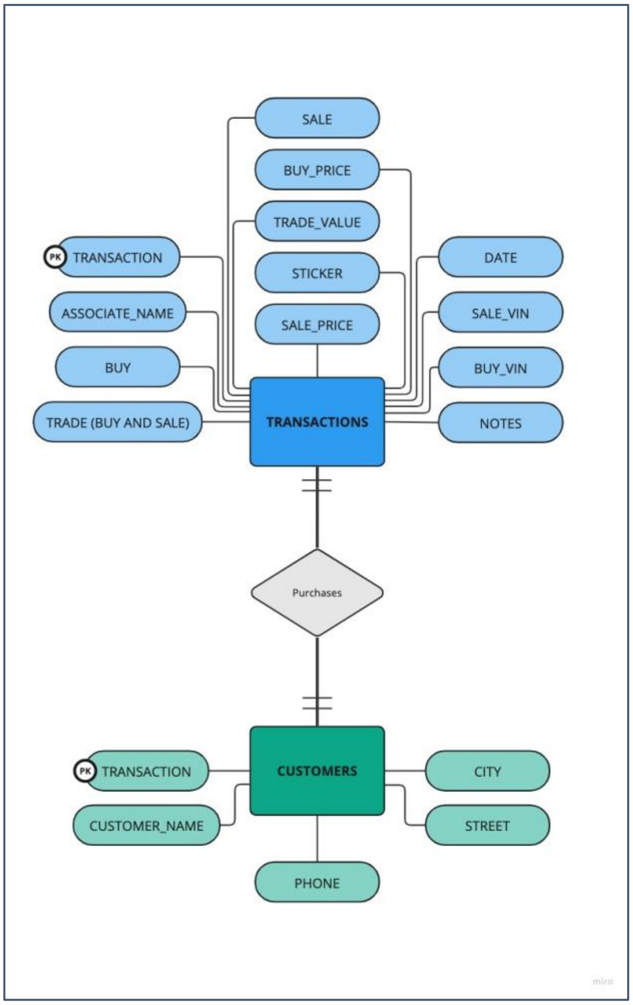
Assignment 1 Part 2: Data Curation using the Relational Model

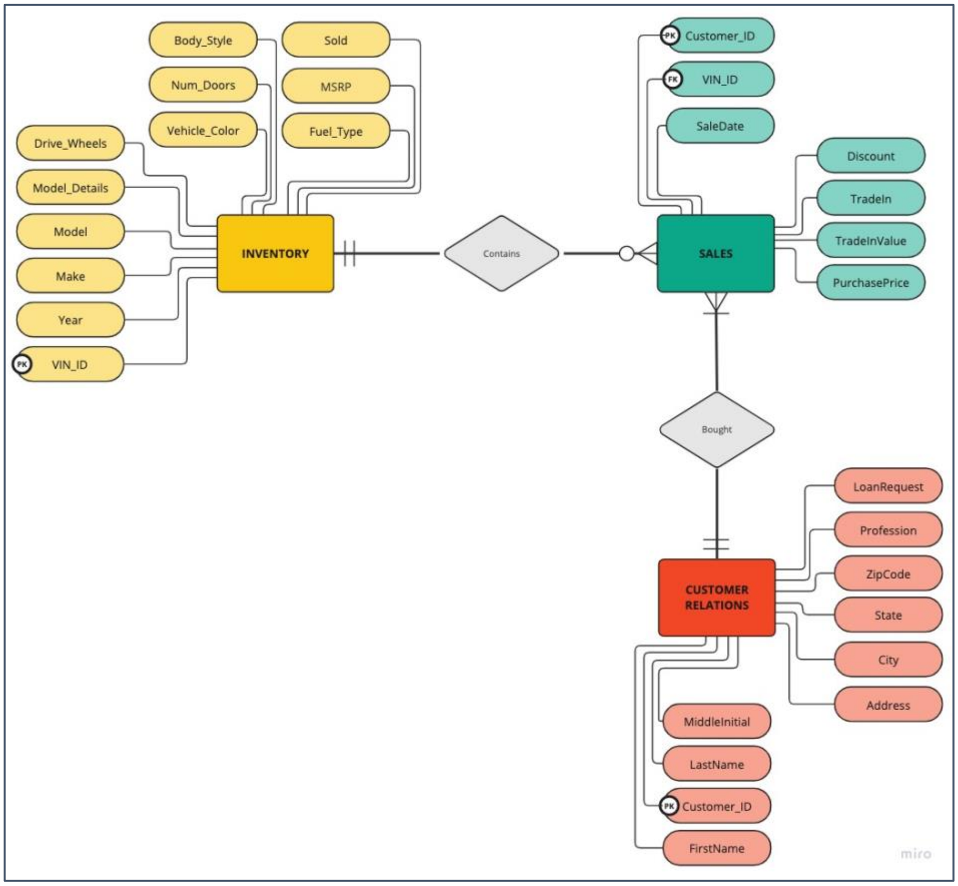
1. Entity Relationship Diagrams for Each Database

Pre-owned Dealership Entity Relationship Diagram



Above we can see the schema as an ER Diagram. The relationship between the Customers & the Transactions are that the customer will purchase a vehicle. It is a one to one relationship, as for each transaction there can only be one customer, whether they are selling a vehicle, buying a vehicle or trading in, each one is distinctly identified by a singular transaction. I created this with the Transaction as the Primary Key as that is what links these two tables.

Part 1 Schema Entity Relationship Diagram



Above we can see the ER diagram for the schema I created in part 1.

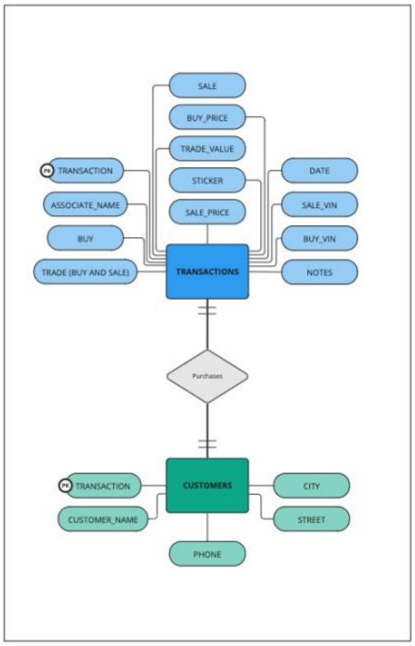
Customer Relations and Sales are linked by the decision of buying a vehicle. It is a one-to-many relationship, specifically a one and only one on the customer side and a one or many on the Sales side. This is because no matter how many vehicles are purchased, there will only ever be one customer entry in the Customer Relations table for that customer. But, a customer can purchase as many vehicles as they want, thus the many on the Sales end. This is contingent on the fact that they do indeed buy a vehicle, as someone would not be in this dealerships database if they have never purchased a vehicle from them, thus the minimum one on the Sales side.

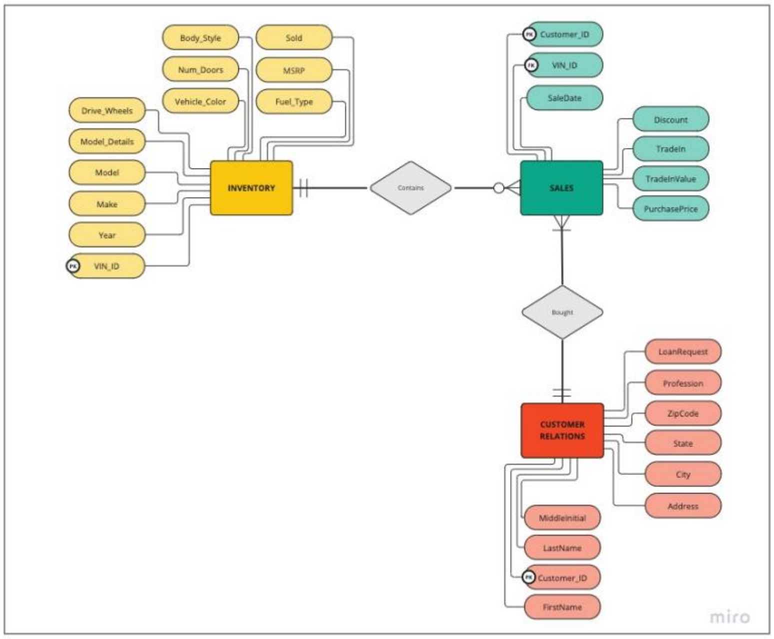
The Sales and Inventory tables are linked by a contains decision, as the inventory will always contain a sold vehicle. There is a one and only one marking from the inventory side, as if a vehicle is sold it can only be sold once. On the Sales side, it is a zero or many mark as a vehicle in the inventory could have been sold but it is not required, and there can be many sold vehicles that are in the inventory.

2. Schema Integration

2.1 Starting Point

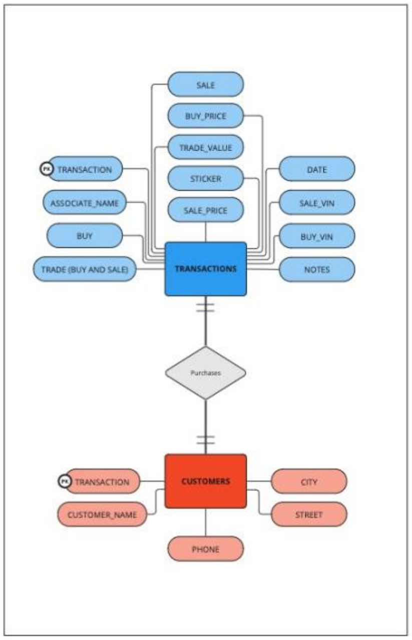
We start with our two diagrams as seen below:

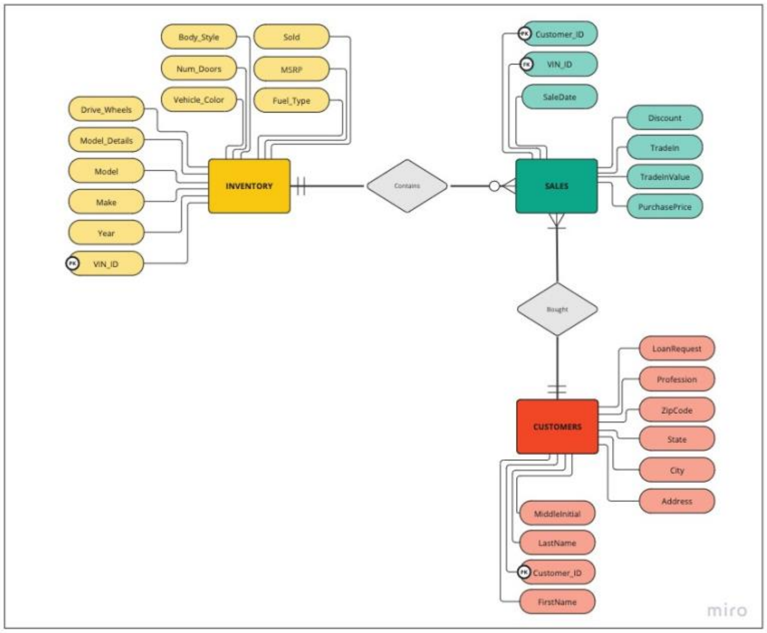




2.2 Semantic Change – Pre-Owned Customers & Customer Relations → Customers

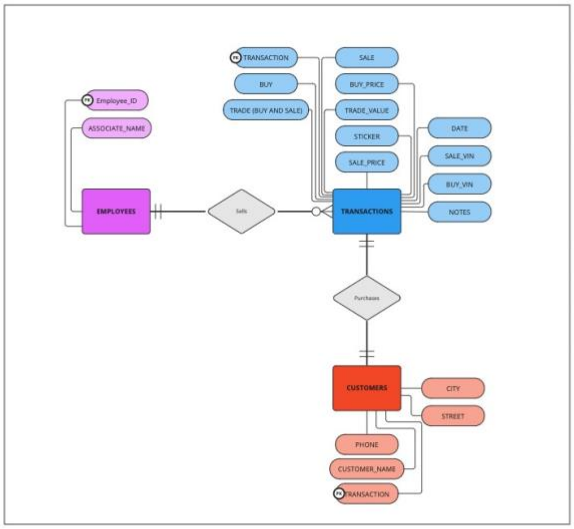
The first change I made was a semantic change of the names of the tables with customer data. For simplicities sake, I decided to go with "Customers" for both tables with Customer Data.

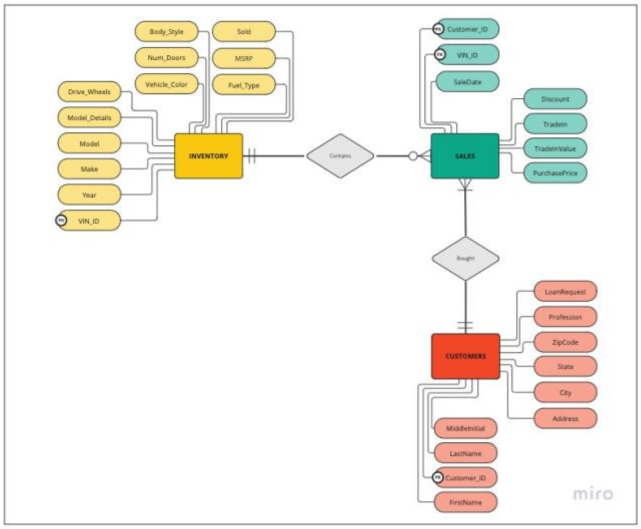




2.3 Conceptual Overlap – Employee Attributes as an Entity

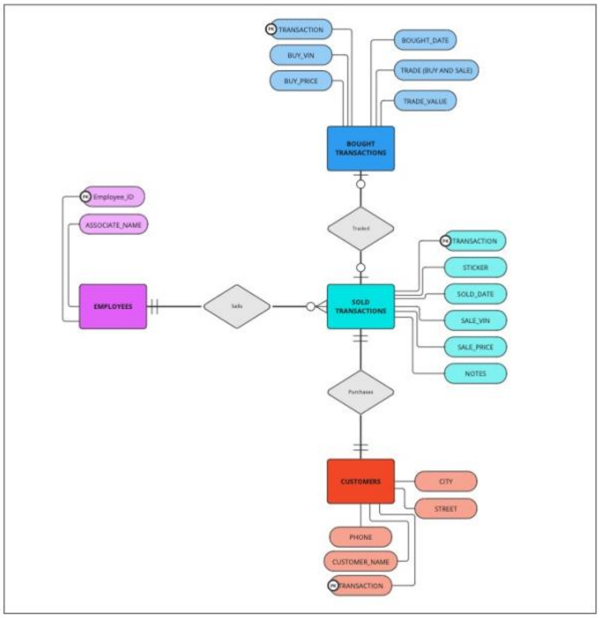
I made a conceptual change on the preowned table to separate the employee attributes as its own entity. I created a new key called Employee ID which will serve as the primary key for this table. This table made sense for me to make as we may want to store other employee information, so it is an entity that could be added to in the future.

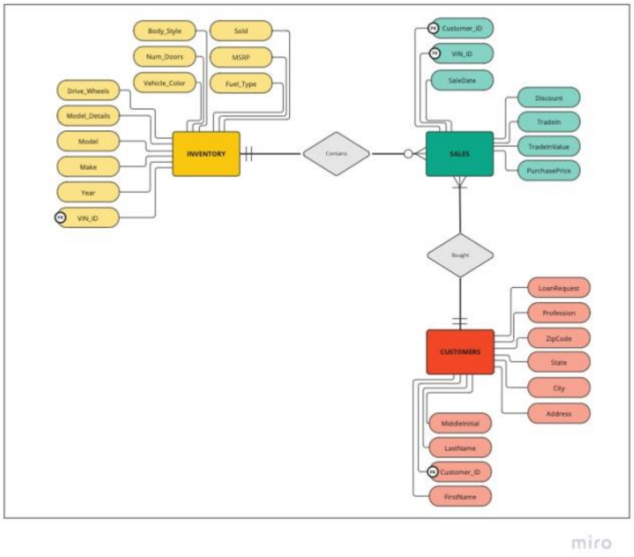




2.4 Conceptual Overlap – Pre-Owned Transactions → Sold & Bought Transactions

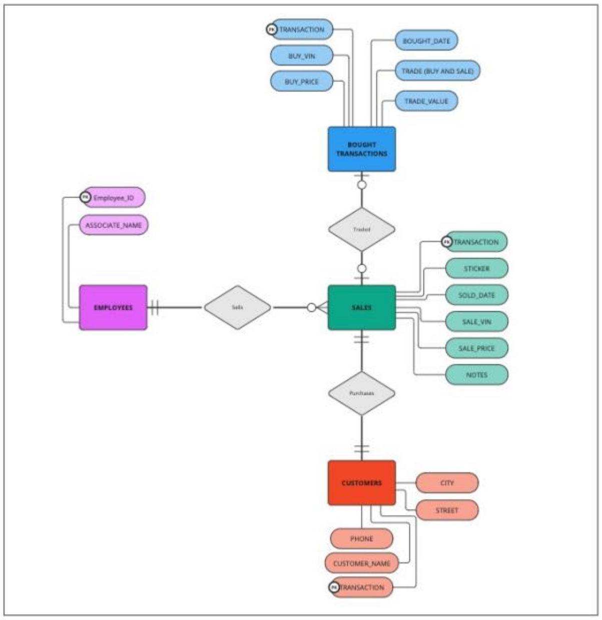
I decided to split the Transactions table into a Sold & Bought Transactions, so that I can properly combine these two schemas later on. This splitting of the entities let me separate the sold information and the bought information. I was able to drop the buy and sale flags from the old Transactions entity, but kept the trade flag in the bought transactions. "Date" became "Bought_Date" and "Sold_Date" to make clear what data was being held.

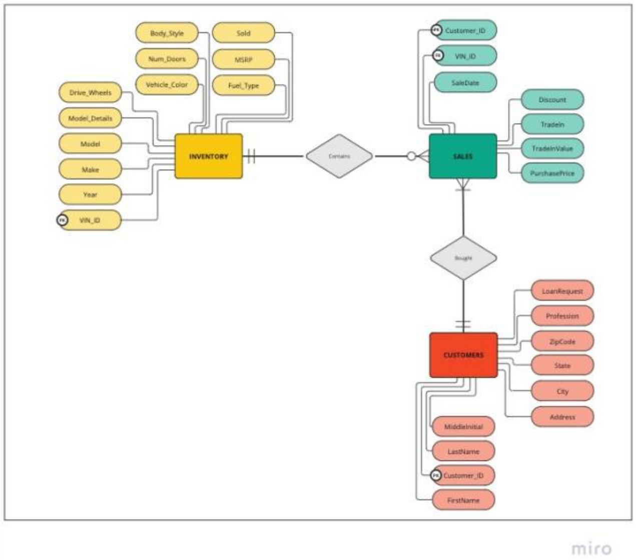




2.5 Semantic Change – Sold Transactions → Sales

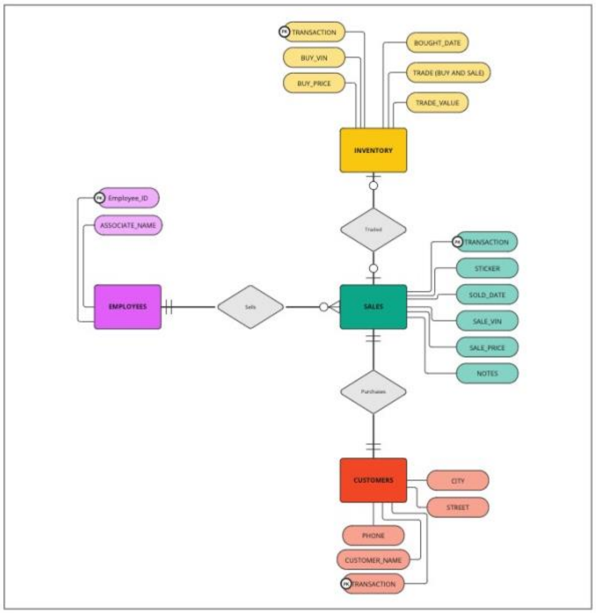
Because we split the entities in the previous step, it allows us to recognize that Sold Transactions is synonymous with Sales. Thus I changed this table to "Sales".

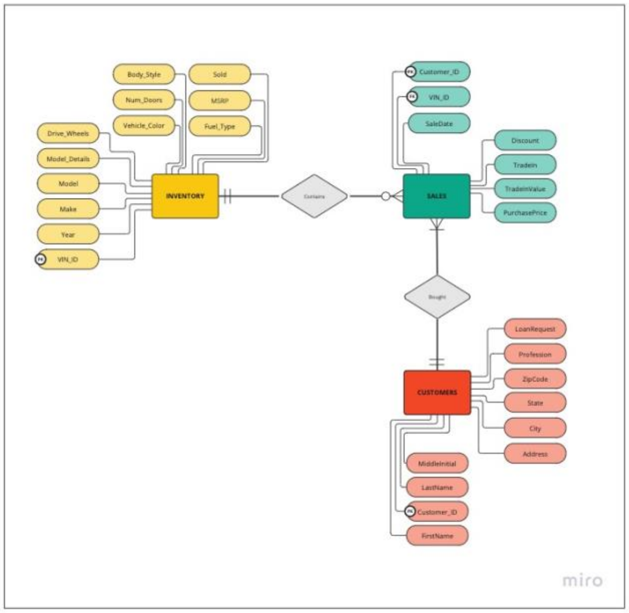




2.6 Semantic Change – Bought Transactions → Inventory

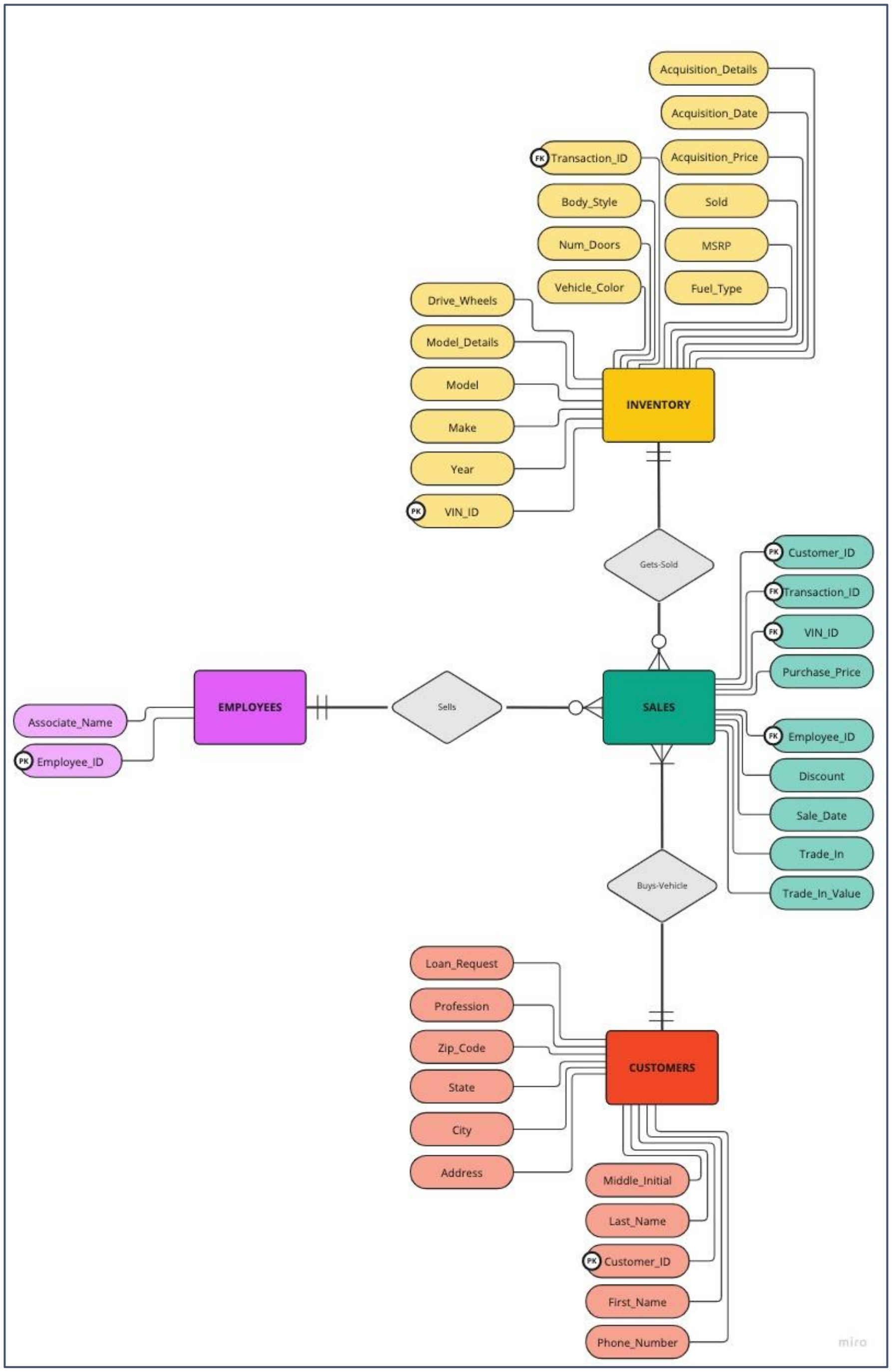
Again because of that entity change, we can see the semantic change from Bought Transactions as Inventory. If I dealership buys a car, it is to ultimately sell it. Thus, it would make sense that the vehicles that they have bought is their inventory.





2.7 Combine Schemas for Final Integrated Result

We are now able to combine our schemas and are left with Inventory, Sales, Customers, and Employee entities. There were no unnecessary relationships so we do not need to consolidate any further. Please see the schema details on why the fields were chosen/kept underneath the table. Semantic changes were made to have a consistent and clear naming convention for the fields within each entity.



★ = Primary Key ☆ = Foreign Key

INVENTORY	
★ VIN_ID	Kept from original schema, Primary Key.
Make	Kept from original schema
Model	Kept from original schema
Year	Kept from original schema
Model_Details	Kept from original schema
Drive_Wheels	Kept from original schema
☆ Transaction_ID	Added to Inventory table to track buying and selling of vehicles.
Body_Style	Kept from original schema
Num_Doors	Kept from original schema
Vehicle_Color	Kept from original schema
Acquisition_Details	Name of Seller & Contact Info. Added this field to track who the dealership has bought the vehicle from. Although this could be considered "Customer" Data, it was kept separate because if we add the Transaction ID to the customer table, we would get duplicate lines from customers with multiple transactions with this dealership.
Acquisition_Date	Date of Acquisition added to track info on buy transaction.
Acquisition_Price	Price of Acquisition added to track info on buy transaction.
Sold	Kept from original schema
MSRP	Kept from original schema
Fuel_Type	Kept from original schema

SALES	
★ Customer_ID	Kept from original schema, Primary Key
☆ Transaction_ID	Added to Sales table to track buying and selling of vehicles between Inventory and Sales tables.
☆ VIN_ID	Kept from original schema
Purchase_Price	Kept from original schema
☆ Empolyee_ID	Added to Sales table to track Employee sales data and to allow joining between Employee & Sales tables.
Discount	Kept from original schema – "Notes" from Pre-Owned Transactions table information would be absorbed here.
Sale_Date	Added from Pre-Owned to track when the vehicle was sold
Trade_In	Kept from original schema
Trade_In_Value	Kept from original schema

EMPLOYEES	
★ Employee_ID	Primary Key to ID Employee
Associate_Name	Name of Associate

Customers	
★ Customer_ID	Kept from original schema, Primary Key
First_Name	Kept from original schema
Last_Name	Kept from original schema
Middle_Initial	Kept from original schema
Phone_Number	Added from Pre-Owned Customer Table, good column to have in order to reach out to Customers
Address	Kept from original schema
City	Kept from original schema
State	Kept from original schema
Zip_Code	Kept from original schema
Profession	Kept from original schema
Loan_Request	Kept from original schema

One thing I wish I did differently is having some way to track the Acquisition Customer Information. As of now that information is stored as columns in the inventory itself, but I would consider changing that in the future as its not the best way to keep that information. While the data loss is kept minimal and it does make sense, it could be a little more cleanly structured.