Database Normalization Analysis

Your Name  
Institution Name

# Functional Dependency

In simple terms, a functional dependency means that one field's value determines another's. For instance, when we look at the table, each car's VIN number (CarVin#) is unique. That means if we know the CarVin#, we can determine details like the year of the car, the manufacturer, the sales price, and even who bought it. So, CarVin# determines Year, Manufacturer, SalesPrice, and CustID.  
  
Similarly, if we know a customer’s CustID, we can find their last name, first name, address, city, and date of birth. So, CustID determines all those customer-related fields.  
  
These are the functional dependencies:  
- CarVin# → Year, Manufacturer, SalesPrice, CustID  
- CustID → lname, fname, StAddr, City, DOB

# Primary Key and Alternate Keys

The most logical choice for the primary key here is CarVin#, because each car has a unique VIN. No two rows have the same CarVin#, and each car sale is uniquely identified by this number.  
  
An alternate key could be CustID. While customers can appear multiple times (because someone can buy more than one car), CustID alone can still determine who the customer is, just not the entire sale record. So it's useful when isolating customer information, but not enough to uniquely define each row in the sales table.

# Normalization Analysis

At first glance, it may seem like the table is in good shape, but it’s actually not in 3rd Normal Form. Let me explain. For a table to be in 3NF, we need to eliminate transitive dependencies — meaning, we don't want one non-key field depending on another.  
  
For example, if we take CustID and see that it gives us StAddr, and then StAddr gives us City, that's a transitive dependency. Because of that, the table doesn't qualify for 3NF. Instead, this table is in 1st Normal Form (1NF), where all values are atomic and properly formatted, but further normalization is still needed.

# 3NF Tables Using SQL

To bring this table into 3NF, we’ll need to separate the data into three tables: Customers, Cars, and Sales. Each table will hold data that only depends on its primary key, avoiding redundancy and dependency chains.

Here’s how we’d write the SQL commands for SQLite or MS Access:

CREATE TABLE Customers (  
 CustID INTEGER PRIMARY KEY,  
 lname TEXT NOT NULL,  
 fname TEXT NOT NULL,  
 StAddr TEXT NOT NULL,  
 City TEXT NOT NULL,  
 DOB TEXT NOT NULL  
);  
  
CREATE TABLE Cars (  
 CarVin TEXT PRIMARY KEY,  
 Year INTEGER NOT NULL,  
 Manufacturer TEXT NOT NULL  
);  
  
CREATE TABLE Sales (  
 SaleID INTEGER PRIMARY KEY AUTOINCREMENT,  
 CarVin TEXT NOT NULL,  
 CustID INTEGER NOT NULL,  
 SalesPrice INTEGER NOT NULL,  
 FOREIGN KEY (CarVin) REFERENCES Cars(CarVin),  
 FOREIGN KEY (CustID) REFERENCES Customers(CustID)  
);

# References

American Psychological Association. (2020). \*Publication manual of the American Psychological Association\* (7th ed.). https://doi.org/10.1037/0000165-000