**Student Projects Chapter 6 - Normalizing the Relational Model for the Student Project and Creating a Normalized Oracle Database**

Read the sample project steps for this chapter and apply the same techniques to the student project that you are developing.

Step 6.1 - Begin with the list of the tables that the entities and relationships from the E-R diagram mapped to naturally, from the sample project section at the end of chapter 4. For each table on the list, identify functional dependencies and normalize the relation to BCNF. Then decide whether the resulting tables should be implemented in that form. If not, explain why.

The following tables resulted from the mapping:

**Ad**(placedIn, initialDate, totalCost, frequency, contactPerson, areaCode, phoneNumber)

**Customer**(custId, firstName, lastName,street,city,state,zip, areaCode, phoneNumber, driversLicNo,  referredBy,adSeen)

**NewCar**(VIN, make, model, listPrice, dateManufactured, placeManufactured, cylinders, doors, weight, capacity, color, delDate, delMiles)

**OptionsMenu**(carOption, price)

**NewCar-Options**(VIN, carOption)

**CustomizationMenu**(customItem, price)

**Salesperson**(empId, firstName, lastName,street,city,state,zip, homeAreaCode, homePhoneNumber, office AreaCode, officePhoneNumber, cellAreaCode, cellPhoneNumber, dateHired)

**Registration**(registrationNo, plateNo, fee)

**Insurance**(InsPolicyNo, companyName,street,city,state,zip, AreaCode, PhoneNumber, startDate, endDate)

**Financing**(FinPolicyNo, companyName,street,city,state,zip, AreaCode, PhoneNumber, startDate, amountFinanced, rate, numberMonths)

**Warrantee**(warType, cost, period)

**UsedCar**(VIN, make, model, cylinders, doors, weight, capacity, color, modelYear, mileage, bookValue)

**UsedCarFeatures**(VIN, feature)

**Sale**(*invoiceNo*, saleDate, salePrice,tax, registrationFee, tradeInAmount, financedAmount, amountPaid, amountDue,commission, saleMiles, *custId, empId, newCarVIN, usedCarVIN insPolNo, insCoName, finPolNo,finCoName,* tradeInVIN*, registrationNo, warType)*

**Sale\_CustomItem**(*invoiceNo, customItem*)

**Survey**(surveyNumber, dealershipRating, carRating, salespersonRating, *invoiceNo*)

**Ad**

Ad(placedIn, initialDate, totalCost, frequency, contactPerson, areaCode, phoneNumber)

placedIn + initialDate → all attributes

It is better to use a numeric field for a key value than a character string, which is subject to differences in spelling, capitalization and punctuation in entering data. Data entry errors or variations in entering string data can cause errors when we try to compare values; in particular, when the fields are used as foreign keys. Therefore, we will create a unique numeric identifier for each ad, which we will call adNumber, and make that the primary key of the first table. All the attributes of the first table will be functionally dependent on it.

We therefore have as our final ad table:

**(1)Ad**(adNumber,placedIn,initialDate,totalCost,frequency,contactPerson,areaCode,phoneNumber)

**Customer**

Customer(custId, firstName, lastName,street,city,state,zip, areaCode, phoneNumber, driversLicNo,  referredBy,adSeen)

For Customer we have the following FDs:

custId → all attributes

firstName + lastName + areaCode + phoneNumber→ all attributes

areaCode + telephoneNumber → street, city, state, zip

zip → city, state

driversLicNo → all attributes

Recall that BCNF permits determinants that are candidate keys to remain in the table, so we do not have a problem with leaving driversLicNo in the table.

For strict BCNF we would break up Customer as follows:

Customer(custId, firstName, lastName,street,city,state,zip, areaCode, phoneNumber, driversLicNo,  referredBy,adSeen)

Phones(areaCode, phoneNumber, street, *zip*)

Zips(zip, city, state)

We choose to create a table that keeps the street and zip code with the customer data, and to create the Zips table to determine city and state. We also use custId as primary key.

However, this design would require that we use the telephone number in order to get the street and zip code of a customer, and that we do two joins whenever we want to get a customer’s complete address. For the sake of efficiency, we will compromise and put the street and zip back in the Customer table. We choose to leave the Zips table as it is, noting that complete zip code tables are available for purchase in electronic form.

For strict BCNF we would break up Customer and Zip as follows:

**(2)**Customer(custId, firstName, lastName,street,*zip*, areaCode, phoneNumber, driversLicNo,  referredBy,*adSeen*)

**(3)**Zips(zip, city, state)

**NewCar**

NewCar(VIN, make, model, listPrice, dateManufactured, placeManufactured, cylinders, doors, weight, capacity, color, delDate, delMiles)

For NewCar, we have the following FDs:

VIN → all attributes

We were using the VIN of the car as a foreign key here. Although there is some logical connection between the dates and attributes of the car, there is no functional dependency between them.

We will create a unique identifier for each newCar as newCarId.

Recall that BCNF permits determinants that are candidate keys to remain in the table, so we do not have a problem with leaving VIN in the table.

**(4)NewCar**(newCarId, VIN, make, model, listPrice, dateManufactured, placeManufactured, cylinders, doors, weight, capacity, color, delDate, delMiles)

**OptionsMenu**

For the OptionsMenu table there are no non-trivial functional dependencies, so we could keep the table in its current form.

**(5)OptionsMenu**(carOption, price)

**NewCar\_Options**

For the NewCar\_Options table there are no non-trivial functional dependencies, so we could keep the table in its current form. However, we wish to use newcarId to identify the newCar.

**(6)NewCar\_Options**(newCarId, carOption)

**CustomizationMenu**

For the CustomizationMenu table there are no non-trivial functional dependencies, so we could keep the table in its current form. We will just add a numeric attribute to serve as primary key: customId.

**(7)CustomizationMenu**(customId, item, price)

**Salesperson**

For Salesperson we have:

empId → all attributes

firstName, lastName -> all attributes

zip → city, state

homeAreaCode + homePhoneNumber → street, city, state, zip

cellAreaCode + cellPhoneNumber → street, city, state, zip

officeAreaCode + officePhoneNumber → street, city, state, zip

Is it possible for two salesperson records with the same telephone number to have two different addresses? If the telephone is a landline, the addresses should be the same (the address of the location of the telephone). If it is a cellphone, may be two salespersons share it to keep records together. We will keep this as a functional dependency. We might then want to ask areaCode + telephoneNumber →? all attributes

We decide this is not the case, since we consider the possibility that two salespersons may share the same home or studio and the same telephone number there, but still have different names, and so on.

The same with office telephone, where two salespersons may share the same office and they don’t live in the same city as they work.

Removing the transitive dependency and making use of the existing Zips table, we have

**(8)Salesperson**(empId, firstName, lastName,street,*zip*, homeAreaCode, homePhoneNumber,office AreaCode, officePhoneNumber, cellAreaCode, cellPhoneNumber, dateHired)

**Registration**

For the Registration table there are no non-trivial functional dependencies, so we could keep the table in its current form.

**(9)Registration**(registrationNo, plateNo, fee)

**Insurance**

For Insurance table we have:

insPolicyNo + companyName → all attributes

areaCode + phoneNumber → street, city, state, zip

zip → city, state

As well as the FDs involving telephone numbers and zips codes, as we saw earlier. As we did for other tables, we will create a numeric primary key, insId. Using the same pattern as for those tables, we design a new Insurance table and make use of the Zips table designed previously.

**(10)Insurance**(insId,insPolicyNo, companyName,street,city,state,zip, areaCode, phoneNumber, startDate, endDate)

**Financing**

For Financing table we have:

finPolicyNo + companyName → all attributes

areaCode + phoneNumber → street, city, state, zip

zip → city, state

As well as the FDs involving telephone numbers and zips codes, as we saw earlier. As we did for other tables, we will create a numeric primary key, finId. Using the same pattern as for those tables, we design a new Financing table and make use of the Zips table designed previously.

**(11)Financing**(finId,finPolicyNo, companyName,street,city,state,zip, AreaCode, PhoneNumber, startDate, amountFinanced, rate, numberMonths)

**Warrantee**

For Warrantee table we have:

warType → all attributes

As we did for other tables, we will create a numeric primary key, warId.

**(12)Warrantee**(warId, warType, cost, period)

**UsedCar**

For UsedCar, we have the following FDs:

VIN → all attributes

We were using the VIN of the car as a foreign key here. We will create a unique identifier for each usedCar as usedCarId.

Recall that BCNF permits determinants that are candidate keys to remain in the table, so we do not have a problem with leaving VIN in the table.

**(13)UsedCar**(usedCarId, VIN, make, model, cylinders, doors, weight, capacity, color, modelYear, mileage, bookValue)

**UsedCarFeatures**

For the UsedCarFeatures table there are no non-trivial functional dependencies, so we could keep the table in its current form. However, we wish to use usedCarId to identify the UsedCar.

**(14)UsedCarFeatures**(usedCarId, feature)

**Sale**

For the Sale table, we have the FD:

invoiceNo → all attributes

Since each car is sold at most once, we also have

newCarId → all attributes

usedCarId → all attributes

Since it is permissible to keep candidate key in the relation, these do not present a problem.

We will substitute the newCarId and usedCarId for the newCarVIN and usedCarVIN, and the insId, finId and warId for the insPolicyNo, insCompany, finPolicyNo, finCompany, and warType, as foreign keys.

If the commission is a constant percentage of the sale price, then we have

salePrice → commission

We do not assume that sale price determines tax, since some customers, such as non-profit organizations, may be tax-exempt. However, the sale total is just the sum of sale price and tax, so we have

salePrice + tax → saleTotal

We could also have the table Commissions(salePrice, salespersonCommission), but we do would not need to store this, since the commission is easily calculated. Similarly, because of the arithmetic relationship among the attributes, we do not need to store the table Totals(salePrice, saleTax, SaleTotal). Also refereeing to amountPaid, tradeInAmount and amountDue that have an arithmetic relationship.

We do not need registration fee, since it has a relation to registration number

registrationNo → registrationFee

Similarly, we do not need financedAmount.

finId → financedAmount

Removing the FDs identified here, we form the new table:

**(15)Sale**(*invoiceNo*, saleDate, salePrice,tax, amountPaid,tradeInAmount,amountDue,saleMiles, *custId, empId, newCarId, usedCarId, insId, finId,* tradeInVIN*, registrationNo, warId)*

**Sale\_CustomItem**

For the Sale\_CustomItem table there are no non-trivial functional dependencies, so we could keep the table in its current form. However, we wish to use customId to identify the Customization item.

**(16)Sale\_CustomItem**(*invoiceNo, customId*)

**Survey**

For the Survey table there are no non-trivial functional dependencies, so we could keep the table in its current form.

**(17)Survey**(surveyNumber, dealershipRating, carRating, salespersonRating, *invoiceNo*)

The tables in boldface, numbered 1-10, will be used as the final set of tables for the relational design for this database.

Step 6.2 - Update the data dictionary and list of assumptions as needed.

We need to add to the data dictionary the new identifiers we created, as follows:

***adNumber*** *A unique numeric identifier created for ad.*

***newCarId*** *A unique numeric identifier created for each new car.*

***customId*** *A unique numeric identifier created for each car customization*.

***insId*** *A unique numeric identifier created for each insurance.*

***finId*** *A unique numeric identifier created for each financing.*

***warId*** *A unique numeric identifier created for each warrantee*.

***usedCarId*** *A unique numeric identifier created for each used car.*

We make a note of the calculated items that are not to be stored

**saleTotal** The total dollar amount of a sale, including price and tax, for a car*; calculated from salePrice and tax. Calculated item*

**salespersonCommission** The dollar amount of commission for a salesperson for the sale of a car. *Calculated item*

**amountDue** The dollar amount due to pay discounting amoundPaid and tradeInAmount. *Calculated item*

There are no changes to the list of assumptions.

Step 6.3 - For each table, write the table name and write out the names, data types, and sizes of all the data items, Identify any constraints, using the conventions of the DBMS you will use for implementation.

**TABLE customer**

custId int, PRIMARY KEY AUTO\_INCREMENT,

firstName varchar 45 NOT NULL,

lastName varchar 45 NOT NULL,

street varchar 45,

zip int, FOREIGN KEY

phoneArea int,

phoneNumber int,

driveLicenseNum varchar 11 NOT NULL UNIQUE,

referredBy varchar 45,

adSeen varchar 45 NOT NULL FOREIGN KEY

**TABLE zips**

zip int PRIMARY KEY,

city varchar 45,

state char 2,

**TABLE salesperson**

salespersonId int PRIMARY KEY AUTO\_INCREMENT,

firstName varchar 45 NOT NULL UNIQUE (firstName, lastName),

lastName varchar 45 NOT NULL UNIQUE (firstName, lastName),

street varchar 45,

zip int NOT NULL FOREIGN KEY,

homePhoneArea int,

homePhoneNumber int,

officePhoneArea int,

officePhoneNumber int,

cellPhoneArea int,

cellPhoneNumber int,

dateHired date,

**TABLE newcar**

newCarId int PRIMARY KEY AUTO\_INCREMENT,

VIN varchar 30 NOT NULL UNIQUE,

make varchar 45,

model varchar 45,

listPrice decimal 10,2,

dateOfManufacture date,

placeOfManufacture varchar 45,

cylinders int,

doors int,

weight int,

capacity int,

color varchar 45,

deliveryDate date,

deliveryMiles int,

**TABLE newcar\_options**

newCarId int PRIMARY KEY (newCarId, carOption) FOREIGN KEY,

carOption int PRIMARY KEY (newCarId, carOption) FOREIGN KEY

**TABLE options\_menu**

carOption int PRIMARY KEY (carOption),

price decimal 10,2,

**TABLE usedcar**

usedCarId int PRIMARY KEY AUTO\_INCREMENT,

VIN varchar 30 NOT NULL UNIQUE,

make varchar 45,

model varchar 45,

cylinders int,

doors int,

weight int,

capacity int,

color varchar 45,

modelYear int,

mileage int,

bookValue decimal 10,2,

**TABLE usedcar\_feature**

usedCarId int PRIMARY KEY (usedCarId, feature),

feature varchar 30 PRIMARY KEY (usedCarId, feature)

**TABLE insurance**

insId int PRIMARY KEY AUTO\_INCREMENT

policy varchar 45 NOT NULL UNIQUE (policy, company),

company varchar 45 NOT NULL UNIQUE (policy, company),

street varchar 45,

zip int, NOT NULL FOREIGN KEY

phoneArea int,

phoneNumber int,

issueDate date,

expirationDate date,

**TABLE registration**

regNumber int PRIMARY KEY AUTO\_INCREMENT,

licensePlate varchar 45,

licenseFee decimal 10,2,

**TABLE sale**

invoiceNumber int PRIMARY KEY AUTO\_INCREMENT,

saleDate date,

salePrice decimal 10,2,

tax decimal 10,2,

tradeInAmount decimal 10,2 DEFAULT '0.00',

amountPaid decimal 10,2,

amountDue decimal 10,2,

saleMiles int,

custId int NOT NULL FOREIGN KEY,

salespersonId int NOT NULL FOREIGN KEY,

newCarId int DEFAULT NULL FOREIGN KEY,

usedCarId int DEFAULT NULL FOREIGN KEY,

insuranceId int NOT NULL FOREIGN KEY,

financingId int DEFAULT NULL FOREIGN KEY,

tradeInVIN varchar 30 DEFAULT '0',

registrationNumber int NOT NULL FOREIGN KEY,

warranteeId int NOT NULL FOREIGN KEY,

**TABLE customization\_menu**

customId intPRIMARY KEY AUTO\_INCREMENT,

item varchar 45 NOT NULL UNIQUE,

price decimal 10,2,

**TABLE sale\_custom\_item**

invoiceNumber int PRIMARY KEY(invoiceNumber, customItem) FOREIGN KEY

customItem int PRIMARY KEY(invoiceNumber, customItem) FOREIGN KEY

**TABLE warrantee**

warId int PRIMARY KEY AUTO\_INCREMENT,

warType varchar 45 NOT NULL UNIQUE,

cost decimal 10,2,

period varchar 30,

**TABLE financing**

insId int PRIMARY KEY AUTO\_INCREMENT,

policy varchar 45 NOT NULL UNIQUE (policy, company),

company varchar 45 NOT NULL UNIQUE (policy, company),

street varchar 45,

zip int, NOT NULL FOREIGN KEY

phoneArea int,

phoneNumber int,

startDate date,

amountFinanced decimal 10,2,

rate varchar 10 DEFAULT '4% p.m',

numberMonths int,

**TABLE survey**

surveyNumber int PRIMARY KEY AUTO\_INCREMENT,

dealershipRating int,

salespersonRating int,

carRating int,

invoiceNumber int NOT NULL FOREIGN KEY,

**TABLE ad**

adNumber int PRIMARY KEY AUTO\_INCREMENT,

placedIn varchar 45 NOT NULL UNIQUE (placedIn, initialDate),

initialDate date NOT NULL UNIQUE (placedIn, initialDate),

totalCost decimal 10,2,

frequency varchar 30,

contactPerson varchar 45,

phoneArea int,

phoneNumber int,

Step 6.4 - Write and execute SQL statements to create all the tables needed to implement the design.

-- Dropping the existing tables for the car\_dealership

**DROP TABLE sale\_custom\_item;**

**DROP TABLE survey;**

**DROP TABLE sale;**

**DROP TABLE newcar\_options;**

**DROP TABLE usedcar\_feature;**

**DROP TABLE customization\_menu;**

**DROP TABLE financing;**

**DROP TABLE customer;**

**DROP TABLE salesperson;**

**DROP TABLE newcar;**

**DROP TABLE options\_menu;**

**DROP TABLE usedcar;**

**DROP TABLE insurance;**

**DROP TABLE registration;**

**DROP TABLE warrantee;**

**DROP TABLE ad;**

**A screenshot of a cell phone

Description automatically generated**

-- Creating new tables for the car\_dealership

**CREATE TABLE zips (**

**zip int NOT NULL AUTO\_INCREMENT,**

**city varchar(45),**

**state char(2),**

**CONSTRAINT zip\_pk PRIMARY KEY (zip));**

**CREATE TABLE ad (**

**adNumber int NOT NULL AUTO\_INCREMENT,**

**placedIn varchar(45) NOT NULL,**

**initialDate date NOT NULL,**

**totalCost decimal(10,2),**

**frequency varchar(30),**

**contactPerson varchar(45),**

**phoneArea int,**

**phoneNumber int,**

**CONSTRAINT ad\_pk PRIMARY KEY (adNumber));**

**CREATE TABLE customer (**

**custId int NOT NULL AUTO\_INCREMENT,**

**firstName varchar(45) NOT NULL,**

**lastName varchar(45) NOT NULL,**

**street varchar(45),**

**zip int,**

**phoneArea int,**

**phoneNumber int,**

**driveLicenseNum varchar(11),**

**referredBy varchar(45),**

**adSeen int NOT NULL,**

**CONSTRAINT custId\_pk PRIMARY KEY (custId),**

**CONSTRAINT zip\_fk FOREIGN KEY (zip) REFERENCES zips(zip),**

**CONSTRAINT ad\_fk FOREIGN KEY (adSeen) REFERENCES ad(adNumber));**

**CREATE TABLE salesperson (**

**salespersonId int NOT NULL AUTO\_INCREMENT,**

**firstName varchar(45) NOT NULL,**

**lastName varchar(45) NOT NULL,**

**street varchar(45),**

**zip int,**

**homePhoneArea int,**

**homePhoneNumber int,**

**officePhoneArea int,**

**officePhoneNumber int,**

**cellPhoneArea int,**

**cellPhoneNumber int,**

**dateHired date,**

**CONSTRAINT salespersonId\_pk PRIMARY KEY (salespersonId),**

**CONSTRAINT zip1\_fk FOREIGN KEY (zip) REFERENCES zips(zip));**

**CREATE TABLE newcar (**

**newCarId int NOT NULL AUTO\_INCREMENT,**

**VIN varchar(30) NOT NULL,**

**make varchar(45),**

**model varchar(45),**

**listPrice decimal(10,2),**

**dateOfManufacture date,**

**placeOfManufacture varchar(45),**

**cylinders int,**

**doors int,**

**weight int,**

**capacity int,**

**color varchar(45),**

**deliveryDate date,**

**deliveryMiles int,**

**CONSTRAINT newCarId\_pk PRIMARY KEY (newCarId));**

**CREATE TABLE options\_menu (**

**carOption int NOT NULL AUTO\_INCREMENT,**

**price decimal(10,2),**

**CONSTRAINT caroption\_pk PRIMARY KEY (carOption));**

**CREATE TABLE newcar\_options (**

**newCarId int NOT NULL,**

**carOption int NOT NULL,**

**CONSTRAINT caroption\_pk PRIMARY KEY (newCarId, carOption),**

**CONSTRAINT newcarId\_fk FOREIGN KEY (newCarId) REFERENCES newcar (newCarId),**

**CONSTRAINT caroption\_fk FOREIGN KEY (carOption) REFERENCES options\_menu (carOption));**

**CREATE TABLE usedcar (**

**usedCarId int NOT NULL AUTO\_INCREMENT,**

**VIN varchar(30) NOT NULL,**

**make varchar(45),**

**model varchar(45),**

**cylinders int,**

**doors int,**

**weight int,**

**capacity int,**

**color varchar(45),**

**modelYear int,**

**mileage int,**

**bookValue decimal(10,2),**

**CONSTRAINT usedCarId\_pk PRIMARY KEY (usedCarId));**

**CREATE TABLE usedcar\_feature (**

**usedCarId int NOT NULL,**

**feature varchar(30) NOT NULL,**

**CONSTRAINT usedcarfeature\_pk PRIMARY KEY (usedCarId, feature));**

**CREATE TABLE insurance (**

**insId int NOT NULL AUTO\_INCREMENT,**

**policy varchar(45) NOT NULL,**

**company varchar(45) NOT NULL,**

**street varchar(45),**

**zip int,**

**phoneArea int,**

**phoneNumber int,**

**issueDate date,**

**expirationDate date,**

**CONSTRAINT insurance\_pk PRIMARY KEY (insId),**

**CONSTRAINT zip2\_fk FOREIGN KEY (zip) REFERENCES zips(zip));**

**CREATE TABLE registration (**

**regNumber int NOT NULL AUTO\_INCREMENT,**

**licensePlate varchar(45),**

**licenseFee decimal(10,2),**

**CONSTRAINT registration\_pk PRIMARY KEY (regNumber));**

**CREATE TABLE customization\_menu (**

**customId int NOT NULL AUTO\_INCREMENT,**

**customItem varchar(45) NOT NULL,**

**price decimal(10,2) DEFAULT NULL,**

**CONSTRAINT custommenu\_pk PRIMARY KEY (customId));**

**CREATE TABLE warrantee (**

**warId int NOT NULL AUTO\_INCREMENT,**

**warType varchar(45) NOT NULL,**

**cost decimal(10,2),**

**period varchar(30),**

**CONSTRAINT warrantee\_pk PRIMARY KEY (warId));**

**CREATE TABLE financing (**

**finId int NOT NULL AUTO\_INCREMENT,**

**policy varchar(45) NOT NULL,**

**company varchar(45) NOT NULL,**

**street varchar(45),**

**zip int,**

**phoneArea int,**

**phoneNumber int,**

**startDate date,**

**amountFinanced decimal(10,2),**

**rate varchar(10) DEFAULT '4% p.m',**

**numberMonths int,**

**CONSTRAINT financing\_pk PRIMARY KEY (finId),**

**CONSTRAINT zip3\_fk FOREIGN KEY (zip) REFERENCES zips(zip));**

**CREATE TABLE sale (**

**invoiceNumber int NOT NULL AUTO\_INCREMENT,**

**saleDate date,**

**salePrice decimal(10,2),**

**tax decimal(10,2),**

**tradeInAmount decimal(10,2) DEFAULT '0.00',**

**amountPaid decimal(10,2),**

**amountDue decimal(10,2),**

**saleMiles int,**

**customerId int NOT NULL,**

**salespersonId int NOT NULL,**

**newCarId int DEFAULT NULL,**

**usedCarId int DEFAULT NULL,**

**insuranceId int NOT NULL,**

**financingId int DEFAULT NULL,**

**tradeInVIN varchar(30) DEFAULT '0',**

**registrationNumber int NOT NULL,**

**warranteeId int NOT NULL,**

**CONSTRAINT sale\_pk PRIMARY KEY (invoiceNumber),**

**CONSTRAINT custId\_fk FOREIGN KEY (customerId) REFERENCES customer (custId),**

**CONSTRAINT salespersonId\_fk FOREIGN KEY (salespersonId) REFERENCES salesperson (salespersonId),**

**CONSTRAINT newcar\_fk FOREIGN KEY (newCarId) REFERENCES newcar (newCarId),**

**CONSTRAINT usedcar\_fk FOREIGN KEY (usedCarId) REFERENCES usedcar (usedCarId),**

**CONSTRAINT insurance\_fk FOREIGN KEY (insuranceId) REFERENCES insurance (insId),**

**CONSTRAINT financing\_fk FOREIGN KEY (financingId) REFERENCES financing (finId),**

**CONSTRAINT registrationnumber\_fk FOREIGN KEY (registrationNumber) REFERENCES registration (regNumber),**

**CONSTRAINT warrantee\_fk FOREIGN KEY (warranteeId) REFERENCES warrantee (warId));**

**CREATE TABLE sale\_custom\_item (**

**invoiceNumber int NOT NULL,**

**customId int NOT NULL,**

**CONSTRAINT salecustom\_pk PRIMARY KEY (invoiceNumber,customId),**

**CONSTRAINT invoicenumber\_fk FOREIGN KEY (invoiceNumber) REFERENCES sale (invoiceNumber),**

**CONSTRAINT customitem\_fk FOREIGN KEY (customId) REFERENCES customization\_menu (customId));**

**CREATE TABLE survey (**

**surveyNumber int NOT NULL AUTO\_INCREMENT,**

**dealershipRating int,**

**salespersonRating int,**

**carRating int,**

**invoiceNumber int NOT NULL,**

**CONSTRAINT survey\_pk PRIMARY KEY (surveyNumber),**

**CONSTRAINT invoice\_fk FOREIGN KEY (invoiceNumber) REFERENCES sale (invoiceNumber));**

A screenshot of a cell phone

Description automatically generated

Step 6.5 - Write and execute SQL statements to create indexes for foreign keys and any other columns that will be used most often for queries.

**CREATE INDEX newCarId\_idx ON newcar\_options (newCarId);**

**CREATE INDEX caroption\_idx ON newcar\_options (carOption);**

**CREATE INDEX custId\_idx ON sale (customerId);**

**CREATE INDEX salespersonId\_idx ON sale (salespersonId);**

**CREATE INDEX newcar\_idx ON sale (newCarId);**

**CREATE INDEX usedcar\_idx ON sale (usedCarId);**

**CREATE INDEX insId\_idx ON sale (insuranceId);**

**CREATE INDEX finId\_idx ON sale (financingId);**

**CREATE INDEX registnumber\_idx ON sale (registrationNumber);**

**CREATE INDEX warrantee\_idx ON sale (warranteeId);**

**CREATE INDEX customId\_idx ON sale\_custom\_item (customId);**

**CREATE INDEX invoicenumber\_idx ON sale\_custom\_item (invoiceNumber);**

**CREATE INDEX invoicenum\_idx ON survey (invoiceNumber);**

**CREATE INDEX adSeen\_idx ON customer (adSeen);**

**CREATE UNIQUE INDEX salespersonname\_idx ON salesperson (firstName, lastName);**

**CREATE UNIQUE INDEX customerdrivelicense\_idx ON customer (driveLicenseNum);**

**CREATE UNIQUE INDEX newcarvin\_idx ON newcar (VIN);**

**CREATE UNIQUE INDEX usedcarvin\_idx ON usedcar (VIN);**

**CREATE UNIQUE INDEX inspolicycomp\_idx ON insurance (policy, company);**

**CREATE UNIQUE INDEX finpolicycomp\_idx ON financing (policy, company);**

**CREATE UNIQUE INDEX custommenu\_idx ON customization\_menu (customItem);**

**CREATE UNIQUE INDEX warrantee\_idx ON warrantee (warType);**

**CREATE UNIQUE INDEX ad\_idx ON ad (placedIn, initialDate);**

A screenshot of a cell phone

Description automatically generated

Step 6.6 - Write and execute SQL statements to insert at least five records in each table, preserving all constraints. Put in enough data to demonstrate how the database will function.

**INSERT INTO zips (zip, city, state)**

**VALUES**

**(13905,'Lynn','MA'),**

**(13217,'Syracuse','NY'),**

**(14624,'Rochester','NY'),**

**(10155,'New York City','NY'),**

**(17405,'York','PA');**

**INSERT INTO ad (adNumber, placedIn, initialDate, totalCost, frequency, contactPerson, phoneArea, PhoneNumber)**

**VALUES**

**(DEFAULT,'Internet','2019-02-09',143.00,'3','Sheff Nern',462,4183703),**

**(DEFAULT,'Magazine','2019-10-12',196.00,'3','Rodney Luke',950,5379358),**

**(DEFAULT,'Newspaper','2019-03-18',275.00,'2','Gunther Sole',914,6011763),**

**(DEFAULT,'Radio','2019-02-15',264.00,'5','Anselma O\'Connolly',224,4736611),**

**(DEFAULT,'TV','2019-06-21',208.00,'1','Dannel Matyushonok',495,6948514);**

**INSERT INTO customer (custId, firstName, lastName, street, zip, phoneArea, phoneNumber, driveLicenseNum, referredBy, adSeen)**

**VALUES**

**(DEFAULT,'Ari','Feavearyear','29468 Trailsway Hill',13905,565,1519997,'592-10-6337','Greg Tomas',5),**

**(DEFAULT,'Merrick','Ogger','1 Pond Alley',13217,684,4276587,'798-62-0623','Bill Miler',4),**

**(DEFAULT,'Pauletta','Stallwood','774 Larry Circle',14624,729,8809712,'686-23-0548','Carol Han',4),**

**(DEFAULT,'Marge','Bonhill','33041 Chinook Point',10155,983,2858761,'722-04-0644','Chun Kai',2),**

**(DEFAULT,'Rachael','Nourse','59254 Sage Center',17405,834,2586399,'886-52-1866','Garret Cart',2);**

**INSERT INTO customization\_menu (customId, customItem, price)**

**VALUES**

**(DEFAULT,'gps system',156.00),**

**(DEFAULT,'alarm system',243.00),**

**(DEFAULT,'sun roof',465.00),**

**(DEFAULT,'leather seats',122.00),**

**(DEFAULT,'new color',321.00);**

**INSERT INTO financing (finId, policy, company, street, zip, phoneArea, phoneNumber, startDate, amountFinanced, rate, numberMonths)**

**VALUES**

**(DEFAULT,'FIN001','Citibank','655 Straubel Way',14624,914,4028607,'2019-01-05',15000.00,NULL,36),**

**(DEFAULT,'FIN002','HSBC','77119 Prairieview Way',14624,203,7563868,'2019-11-11',12000.00,NULL,36),**

**(DEFAULT,'FIN003','US Bank','9 Springview Drive',13217,974,9438954,'2019-07-10',10000.00,NULL,24),**

**(DEFAULT,'FIN004','Citibank','12 Summer Ridge Terrace', 17405,172,8915964,'2018-10-30',15000.00,NULL,48),**

**(DEFAULT,'FIN005','HSBC','261 Bobwhite Park',10155,914,1240079,'2019-06-15',13000.00,NULL,36);**

**INSERT INTO insurance (insId, policy, company, street, zip, phoneArea, phoneNumber, issueDate, expirationDate)**

**VALUES**

**(DEFAULT,'INS001','Geico','2243 Central Park Ave', 14624,203,2077847,'2019-01-04','2019-07-04'),**

**(DEFAULT,'INS002','Liberty','2243 Main Ave', 14624,203,2077847,'2019-10-09','2020-04-09'),**

**(DEFAULT,'INS003','Geico','2243 Central Park Ave',10155,914,2077847,'2019-08-24','2020-02-24'),**

**(DEFAULT,'INS004','Geico','2243 Central Park Ave',10155,914,2077847,'2019-08-27','2020-02-27'),**

**(DEFAULT,'INS005','Liberty','2651 Strang Blvd Floor One',17405,172,7850180,'2019-11-20','2020-05-20');**

**INSERT INTO newcar (newCarId, VIN, make, model, listPrice, dateOfManufacture, placeOfManufacture, cylinders, doors, weight, capacity, color, deliveryDate, deliveryMiles)**

**VALUES**

**(DEFAULT,'1C3CCBCG8DN043829','Land Rover','Discovery',47337.00,'2019-05-22','New York',6,5,3693,5,'Black','2019-06-01',102),**

**(DEFAULT,'1D7RB1CT7BS881175','Volkswagen','Golf',17264.00,'2019-08-10','New York',4,5,5868,5,'White','2019-12-02',129),**

**(DEFAULT,'1G4HH5E90AU691486','Volkswagen','Passat',22209.00,'2019-07-18','New York',4,5,3800,5,'Blue','2019-08-02',51),**

**(DEFAULT,'3C63D3HL7CG670654','Subaru','Crosstrek',20461.00,'2019-09-13','New York',4,5,4888,5,'Blue','2019-11-10',567),**

**(DEFAULT,'3VW517AT1EM034793','Ford','Edge',31292.00,'2019-08-05','New Jersey',6,5,4413,5,'Blue','2019-09-15',648);**

**INSERT INTO options\_menu (carOption, price)**

**VALUES**

**(DEFAULT,1643.00),**

**(DEFAULT,945.00),**

**(DEFAULT,695.00),**

**(DEFAULT,539.00),**

**(DEFAULT,224.00);**

**INSERT INTO newcar\_options (newCarId, carOption)**

**VALUES**

**(1,1),**

**(2,2),**

**(3,3),**

**(4,4),**

**(5,5);**

**INSERT INTO registration (regNumber, licensePlate, licenseFee)**

**VALUES**

**(DEFAULT,'Y31FAC',102.00),**

**(DEFAULT,'N16FKN',81.00),**

**(DEFAULT,'F44FAE',81.00),**

**(DEFAULT,'H98FHO',110.00),**

**(DEFAULT,'K99FHB',85.00);**

**INSERT INTO usedcar (usedCarId, VIN, make, model, cylinders, doors, weight, capacity, color, modelYear, mileage, bookValue)**

**VALUES**

**(DEFAULT,'2HNYD28257H276330','Subaru','Impreza',4,5,4320,5,'Black',2015,9923,15000.00),**

**(DEFAULT,'3N1CN7AP4FL777122','Honda','CRV',4,5,3790,5,'Blue',2016,10612,17000.00),**

**(DEFAULT,'WA1C8AFP8DA597506','BMW','328i',4,5,4123,5,'Black',2017,13199,26000.00),**

**(DEFAULT,'WAUCVAFRXCA314714','Nissan','Rogue',4,5,4090,5,'Black',2018,21308,16000.00),**

**(DEFAULT,'WBAKG1C57BE375422','Volkswagen','Golf',4,5,3902,5,'Gray',2015,9407,13000.00),**

**(DEFAULT,'WDDHF2EB6DA278138','Subaru','Crosstrek',4,5,4145,5,'White',2016,3374,17000.00);**

**INSERT INTO usedcar\_feature (usedCarId, feature)**

**VALUES**

**(1,'sunroof'),**

**(2,'gpsSystem'),**

**(3,'sunroof+gpsSystem'),**

**(4,'gpsSystem'),**

**(5,'sunroof');**

**INSERT INTO warrantee (warId, warType, cost, period)**

**VALUES**

**(DEFAULT,'Accessories Warranty',600.00,'2 years'),**

**(DEFAULT,'New Car Warranty',1000.00,'2 years'),**

**(DEFAULT,'New Car Warranty Extension',1700.00,'4 years'),**

**(DEFAULT,'Powertrain Warranty',1300.00,'2 years'),**

**(DEFAULT,'Used Car Warrantee',500.00,'90 days');**

**INSERT INTO salesperson (salespersonId, firstName, lastName, street, zip, homePhoneArea, homePhoneNumber, officePhoneArea, officePhoneNumber, cellPhoneArea, cellPhoneNumber, dateHired)**

**VALUES**

**(DEFAULT,'Pat','Cawtheray','5 Pankratz Park',14624,786,3542296,422,7436544,786,9632453,'2019-02-23'),**

**(DEFAULT,'Ari','Rembrant','3 Texas Plaza',10155,122,4076779,422,7436554,122,6435647,'2019-01-26'),**

**(DEFAULT,'Timotheus','Yeowell','03634 Talmadge Lane',10155,522,5740369,422,7436564,522,8646321,'2018-08-13'),**

**(DEFAULT,'Rosalyn','Kuhnhardt','62084 Dexter Point',10155,282,9200832,422,7436574,282,9654771,'2018-03-20'),**

**(DEFAULT,'Lyman','Bottomer','054 Amoth Avenue',14624,936,3733865,422,7436584,936,7129836,'2018-06-23');**

**INSERT INTO sale (invoiceNumber, saleDate, salePrice, tax, tradeInAmount, amountPaid, amountDue, saleMiles, customerId, salespersonId, newCarId, usedCarId, insuranceId, financingId, tradeInVIN, registrationNumber, warranteeId)**

**VALUES**

**(DEFAULT,'2019-07-28',47337.00,4733.00,0.00,47337.00,0.00,135,1,2,1,NULL,1,NULL,NULL,1,1),**

**(DEFAULT,'2019-08-24',17264.00,1726.00,0.00,17264.00,0.00,123,2,4,2,NULL,2,NULL,NULL,2,2),**

**(DEFAULT,'2019-12-14',20461.00,2046.00,0.00,20461.00,0.00,123,3,4,4,NULL,3,NULL,'WBAKG1C57BE375422',3,1),**

**(DEFAULT,'2019-04-21',31292.00,3129.00,0.00,21292.00,10000.00,123,4,3,5,NULL,4,3,NULL,4,2),**

**(DEFAULT,'2019-04-29',26000.00,2600.00,0.00,11000.00,15000.00,123,5,5,NULL,3,5,1,NULL,5,3);**

**INSERT INTO sale\_custom\_item (invoiceNumber, customId)**

**VALUES**

**(1,5),**

**(2,1),**

**(3,1),**

**(4,3),**

**(5,2);**

**INSERT INTO survey (surveyNumber, dealershipRating, salespersonRating, carRating, invoiceNumber)**

**VALUES**

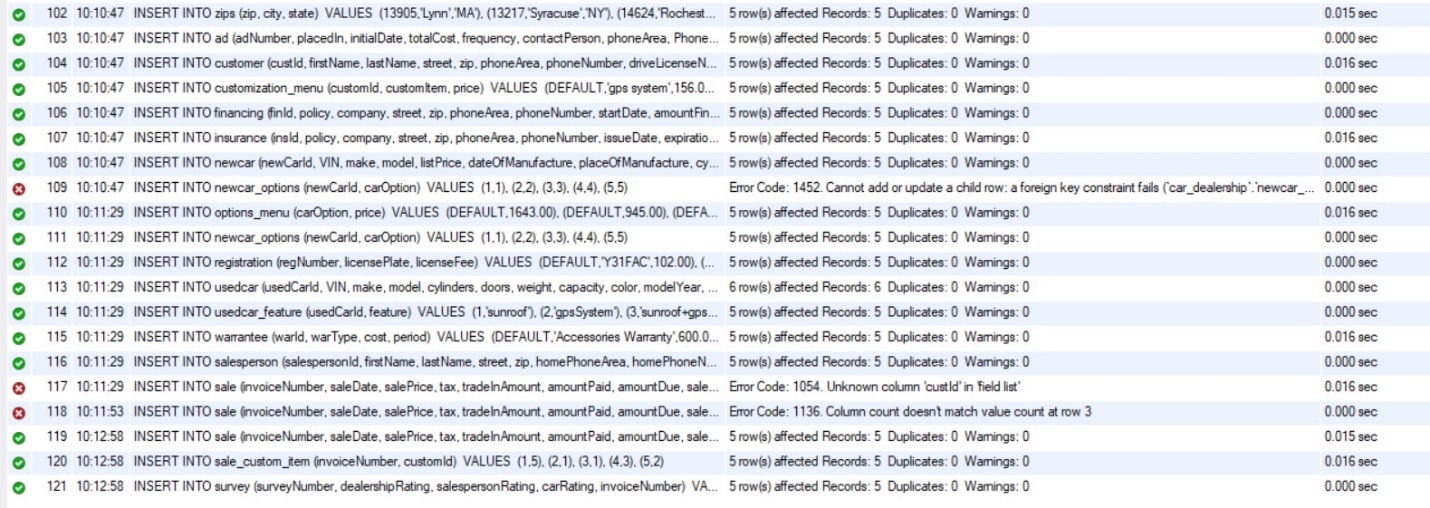
**(DEFAULT,2,2,3,1),**

**(DEFAULT,2,3,3,2),**

**(DEFAULT,2,3,2,3),**

**(DEFAULT,1,2,2,4),**

**(DEFAULT,3,3,2,5);**



Step 6.7 - Write and execute SQL statements that will process five non-routine requests for information from the database just created.

-- Query 01: What is the most expensive new car available?

**SELECT**

**make,**

**model,**

**listPrice**

**FROM newcar**

**WHERE listPrice=(select max(listPrice) from newcar);**

**A screenshot of a cell phone

Description automatically generated**

-- Query 02: Who are the customers who live in New York?

**SELECT**

**c.firstName,**

**c.lastName,**

**c.street,**

**z.city,**

**z.state,**

**z.zip**

**FROM customer c**

**JOIN zips z**

**ON c.zip = z.zip**

**WHERE state='NY';**

**A screenshot of a social media post

Description automatically generated**

-- Query 03: Which sales were made between April 2019 and December 2019?

**SELECT**

**invoiceNumber,**

**saleDate,**

**salePrice,**

**newCarId,**

**usedcarId**

**FROM sale**

**WHERE saleDate BETWEEN '2019-04-01' AND '2019-12-31';**

A screenshot of a cell phone

Description automatically generated

-- Query 04: What model, year, color and price of Subaru used cars available?

**SELECT**

**make,**

**model,**

**modelYear,**

**color,**

**bookValue**

**FROM usedcar**

**WHERE make = 'Subaru';**

A screenshot of a cell phone

Description automatically generated

-- Query 05: Which insurance companies have been contracted by customers?

**SELECT**

**DISTINCT company**

**FROM insurance;**

A screenshot of a cell phone

Description automatically generated

Step 6.8 - Write and execute SQL statements to create at least one trigger.

**CREATE TRIGGER formatting\_values**

**BEFORE INSERT ON customer**

**FOR EACH ROW**

**SET**

**NEW.firstName = CONCAT(UPPER(LEFT(new.firstName, 1)), SUBSTRING(new.firstName, 2)),**

**NEW.lastName = CONCAT(UPPER(LEFT(new.lastName, 1)), SUBSTRING(new.lastName, 2));**

**A screenshot of a social media post

Description automatically generated**

Step 6.9 - Write and execute SQL statements to demonstrate that the trigger is working as expected.

**Checking Trigger formatting\_values:**

1. **BEFORE CREATING TRIGGER:**

**INSERT INTO customer (custId, firstName, lastName, street, zip, phoneArea, phoneNumber, driveLicenseNum, referredBy, adSeen)**

**VALUES**

**(6, 'john', 'smith', '750 Rock Street', 14624, 203, 2876337, '896-22-1112', null, 4);**

A screenshot of a social media post

Description automatically generated

1. **AFTER CREATING TRIGGER:**

**INSERT INTO customer (custId, firstName, lastName, street, zip, phoneArea, phoneNumber, driveLicenseNum, referredBy, adSeen)**

**VALUES**

**(6, 'john', 'smith', '750 Rock Street', 14624, 203, 2876337, '896-22-1112', null, 4);**

A screenshot of a social media post

Description automatically generated