**Student Projects Chapter 4 - Initial Mapping to Tables for Student Projects**

Step 4.1 - Map the E-R model developed at the end of Chapter 3 to a relational model, using the guidelines presented in Section 4.7 and illustrated in the sample project. Write the schema for the database, but do not create the database at this stage.

A diagram of sales

Description automatically generated

The E-R diagram shown above showed the entities Customer, Salesperson, Sale, New Car, Used Car, Insurance, Registration, Customization, Warrantee, Financing, Survey and Ad. The entities map to the following tables.

Note that we have replaced composite attributes by their simple components, and we have underlined primary keys. Although it is not necessary, it is customary to list the primary keys as the first columns in the tables. Note also that we created some ID attributes to use as primary key instead of composite primary key.

**Customer**: custID, firstName, lastName, dateOfBirth, sex, street, city, state, zip, phoneArea, phoneNumber, email, socialSecNo, dateFilledIn, reference, salesperson, preferredCarMake, preferredCarModel, preferredCarColor, preferredCarNewOrUsed

**Salesperson**: salespersonID, firstName, lastName, socialSecNo, dateOfBirth, sex, street, city, state, zip, phoneArea, phoneNumber, email, dateOfAdmission

**Sale**: invoiceNumber, saleDate, salePrice, licenseFee, tax, tradeInPrice, warranty, financingInform, customizationInform, salespersonCommission, priceTotal

**New Car**: newCarVIN, make, model, color, year, price, mileage, capacity, weight, numOfCylinders, numOfDoors, dateOfAcquisition, costOfAcquisition, dateOfDelivery, mileageAtDelivery, dateOfManufacture, placeOfManufacture

**Used Car**: usedCarVIN, make, model, color, year, tradeInPrice, mileage, price

**Insurance**: insuranceCompany, insurancePolicy, insuranceIssueDate, insuranceExpDate

**Registration**: registrationNum, registrationDate, licenseFee, licensePlate

**Customization:** specialTrimPrice, alarmSystemPrice, audioSystemPrice, gpsSystemPrice, sunroofPrice, airconditioningPrice

**Warrantee:** warranteeStd, warranteeExt, warranteeExtPrice

**Financing**: creditApplicationDate, loanAmount, loanPayment, loanRate, loanTotalAmount, employer, position, monthlyIncome, employerStreet, employerCity, employerState, employerZip, yearsThere, bankReference, bankAccountNum, bankManager, bankStreet, bankCity, bankState, bankZip, bankPhoneArea, bankPhoneNum

**Survey:** surveyDate, customerSuggestions, dealershipMeanRate, salespersonMeanRate, vehicleMeanRate

**Ad:** placedIn, initialDate, totalCost, contactPerson, areaCode, phoneNumber

The relationship sets are SoldTo, SoldBy, SoldNew, SoldUsed, TradedIn, Evaluates, Performed, Requires, Provides, MayHave, MayInclude, Sees.

The one-to-many SoldTo relationship will be represented by placing the foreign key of the primary key of the Customer table (custId) in the Sale table. Therefore we need to put the primary key in the Sale table by calling them custId indicated by italics in the schema shown below in boldface. The attribute SocSecNum from the Customer table is unique as well and it is a candidate key.

The Evaluates relationship is one-to-one. If we do not wish to construct a new table, we have the options of placing the primary key of Sale in Survey, or of Survey in Sale. We do not need to do both. We will choose the first alternative.

The SoldNew relationship is also a one-to-one relationship. In this we place the primary key of New Car in the Sale table. The same with the SoldUsed relationship in which we place the primary key of Used Car in the Sale table. Both are default defined as null and when a sale is completed the value of New Car or Used Car is entered to fill that attribute value accordingly to the primary key of the car, or VIN.

The Performed relationship is a one-to-many relationship between Sale and Customization and can be represented by placing the primary key of Sale in Customization. We place the key invoiceNumber from Sale in Customization.

The Requires relationship is a one-to-one relationship between Sale and Registration. In this we place the primary key of Registration in the Sale table.

The Provides relationship is also a one-to-one relationship. In this we place the key of Insurance in the Sale table. Note that the primary key of Insurance is a composite key. In that case we place those two attributes as a foreign key in the Sale table.

The SoldBy relationship is a one-to-many relationship in which we place the primary key of Salesperson (salespersonId) in the Sale table. Also, firstName and lastName are unique attributes when composed.

The MayHave relationship is a one-to-one relationship. In this we can place the primary key of Sale into Financing table or of Financing in Sale table. We choose the first alternative.

The MayInclude relationship is also a one-to-one relationship. In this we can place the primary key of Warantee in the Sale table or the key of Sale in the Warantee table. We choose the second alternative since Warantee doesn’t have a primary key.

The TradedIn relationship is a one-to-one relationship. In this we place the primary key of Used Car in the Sale table.

The Sees is a one-to-many relationship. In this we place the primary key of Customer in the Ad Table since it doesn’t have a primary key.

Note: when we say that some primary key will be placed into another table, we mean it is placed as foreign key.

The resulting tables in the conceptual level relational schema are the following:

**Customer**: custId, firstName, lastName, dateOfBirth, sex, street, city, state, zip, phoneArea, phoneNumber, email, socialSecNo, dateFilledIn, preferredCarMake, preferredCarModel, preferredCarColor, preferredCarNewOrUsed

**Salesperson**: salespersonId, firstName, lastName, socialSecNo, dateOfBirth, sex, street, city, state, zip, phoneArea, phoneNumber, email, dateOfAdmission

**Sale**: invoiceNumber, *custID*, *salespersonID, newCarVIN, usedCarVIN, insuranceCompany, insurancePolicy, registrationNum,* saleDate, salePrice, tax, salespersonCommission

**New Car**: newCarVIN, make, model, color, year, price, mileage, capacity, weight, numOfCylinders, numOfDoors, dateOfAcquisition, costOfAcquisition, dateOfDelivery, mileageAtDelivery, dateOfManufacture, placeOfManufacture

**Used Car**: usedCarVIN, make, model, color, year, tradeInPrice, mileage, price

**Insurance**: insuranceCompany, insurancePolicy, insuranceIssueDate, insuranceExpDate

**Registration**: registrationNum, registrationDate*,* licenseFee, licensePlate

**Customization:** *invoiceNumber,* specialTrimPrice, alarmSystemPrice, audioSystemPrice, gpsSystemPrice, sunroofPrice, airconditioningPrice

**Warrantee:** *invoiceNumber,* warranteeStd, warranteeExt, warranteeExtPrice

**Financing**: *invoiceNumber,* creditApplicationDate, loanAmount, loanPayment, loanRate, loanTotalAmount, employer, position, monthlyIncome, employerStreet, employerCity, employerState, employerZip, yearsThere, bankReference, bankAccountNum, bankManager, bankStreet, bankCity, bankState, bankZip, bankPhoneArea, bankPhoneNum,

**Survey:** *invoiceNum*ber, surveyDate, customerSuggestions, dealershipMeanRate, salespersonMeanRate, vehicleMeanRate

**Ad:** *custId,* placedIn, initialDate, totalCost, contactPerson, areaCode, phoneNumber

We note that the primary key of Insurance consists of two character string attributes. We observe that it becomes burdensome to include these multiple attributes when we use them as foreign keys. We will address this issue in a later chapter.