**Chapter 4 – Sample Project: Initial Mapping of the E-R Model to Tables for The Art Gallery**

**Step 4.1- Map the E-R model developed at the end of Chapter 3 to relational model, using the guidelines presented in Section 4.7 and Figure 4.7 of the textbook.**

The E-R diagram shown in Figure S.3.1 showed the strong entities Artist, PotentialCustomer, Collector, Show, Sale, Buyer, and Salesperson, and a weak entity, Artwork.

The strong 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.

**Artist**(firstName, lastName, street, city, state, zip, interviewDate, interviewerName, areaCode, telephoneNumber, salesLastYear, salesYearToDate, socialSecurityNumber, usualMedium, usualStyle, usualType)

**PotentialCustomer**(firstName, lastName, areaCode, telephoneNumber, street, city, state, zip, dateFilledIn, preferredMedium, preferredStyle, preferredType

**Collector**(SocialSecurityNumber, firstName, lastName, street, city, state, zip, interviewDate, interviewerName, areaCode, telephonenumber, salesLastYear, salesYearToDate, collectionArtistFirstName, collectionArtistLastName, collectionMedium, collectionStyle, collectionType)

**Show**(showTitle, showClosingDate, showTheme, showOpeningDate)

**Sale**(invoiceNumber, amountRemittedToOwner, saleDate, salePrice, saleSalesPersonCommission, saleTax, SaleTotal)

**Buyer**(firstName, lastName, street, city, state, zip, areaCode, telephoneNumber, purchasesLastYear, purchasesYearToDate)

**Salesperson**(socialSecurityNumber, firstName, lastName, street, city, state, zip)

Since the weak entity Artwork depends on Artist, we add the key of Artist to the Artwork table and combined it with the weak entity’s partial key to form a primary key:

**Artwork**(artistLastName, artistFirstName, workTitle, askingPrice, dateListed, dateReturned, dateShown, status, workMedium, workSize, workStyle, workType, workYearCompleted)

The relationship sets are PreferredBy, CollectedBy, Creates, FeaturedIn, Owns, ShownIn, SoldIn, SoldTo, and SoldBy.

The one-to-many PreferredBy will be represented by a foreign key. Therefore we need to put artistLastName and artistFirstName in the PotentialCustomer table, calling them preferredArtistLastName and preferredArtistFirstName, which together will form a foreign key, indicated by italics in the schema shown below in boldface.

The CollectedBy relationship is also one-to-many, and can be represented by placing the key of Artist in Collector. We place collectionArtistLastName, collectionArtistFirstName in the Collector table.

The Creates relationship has already been represented by placing the primary key of Artist in Artwork. We show by using italics that these attributes, although part of the primary key, also form a foreign key in Artwork.

FeaturedIn is a one-to-many relationship that can be represented using a foreign key, adding the attributes showFeaturedArtistLastName, showFeaturedArtistFirstName.

The one-to-many Owns relationship can be represented by placing the primary key of Collector in the Artwork table, so we add collectorSocialSecurityNumber to that table, using italics to show that it is a foreign key.

The ShownIn relationship is many-to-many, so we must construct a table with the primary keys of Artwork and Show to represent it. Since there are no descriptive attributes on the E-R diagram, this is an “all key” table, with no non-key attributes. We also note that its attributes are foreign keys, referring to the tables where they are primary keys.

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

The SoldTo relationship is many-to-one. We represent it by placing the key of the “one” side, Buyer, in the table for the “many” side, Sale, where it is the foreign key buyerLastName, buyerFirstName, areaCode, telephoneNumber.

Similarly, to represent the one-to-many SoldBy relationship, we place the primary key of Salesperson in the Sale table, where it becomes the foreign key salespersonSocialSecurityNumber.

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

**Artist**(firstName, lastName, street, city, state, zip, interviewDate, interviewerName, areaCode, telephoneNumber, salesLastYear, salesYearToDate, socialSecurityNumber, usualMedium, usualStyle, usualType)

**PotentialCustomer**(firstName, lastName, street, city, state, zip, areaCode, telephoneNumber, dateFilledIn, *preferredArtistLastName, preferredArtistFirstName*, preferredMedium, preferredStyle, preferredType)

**Artwork**(*artistLastName, artistFirstName*, workTitle, askingPrice, dateListed, dateReturned, dateShown, status, workMedium, workSize, workStyle, workType, workYearCompleted, *collectorSocialSecurityNumber*)

**ShownIn**(*artistLastName, artistFirstName, workTitle, showTitle*)

**Collector**(SocialSecurityNumber, firstName, lastName, street, city, state, zip, interviewDate, interviewerName, areaCode, telephonenumber, salesLastYear, salesYearToDate, *collectionArtistFirstName, collectionArtistLastName*, collectionMedium, collectionStyle, collectionType, SalesLastYear, SalesYearToDate)

**Show**(showTitle, *showFeaturedArtistLastName, showFeaturedArtistFirstName*, showClosingDate, showTheme, showOpeningDate)

**Sale**(InvoiceNumber, *artistLastName, artistFirstName*, *workTitle,* amountRemittedToOwner, saleDate, salePrice, saleSalesPersonCommission, saleTax, SaleTotal, *buyerLastName, buyerFirstName, buyerAreaCode, buyerTelephoneNumber salespersonSocialSecurityNumber*)

**Buyer**(firstName, lastName, areaCode, telephoneNumber, street, city, state, zip, purchasesLastYear, purchasesYearToDate)

**Salesperson**(socialSecurityNumber, firstName, lastName, street, city, state, zip)

We note that the primary keys of Artist, Buyer, and PotentialCustomer consist of two or more 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.