Assignment 3. Ontologies / ER Diagram Design Exercise

IS 531: Foundations of Data Curation

University of Illinois at Urbana-Champaign

Andrew Nedilko

1. ER Diagram for the Pre-Owned Dealer Database

The relations in file "*Exercise3_FaileA.xls*" contain a lot of inconsistencies due to data input errors and need the following data cleaning:

- The BUY, TRADE, and SALE attributes of the TRANSACTIONS relation, when not applicable to a transaction, have the values of "NULL", "no", "-", or "--" which are variations of NULL handling. In one case an associate even used "y" (transaction 10123465) to denote this fact. For convenience and uniformity, if I had to correct these attributes I would use, for example, Boolean values everywhere (YES if it applies to the transaction, as suggested in the LEGEND in the xlsx file, and NO otherwise).
- The BUY_PRICE, TRADE_VALUE, STICKER, and SALE_PRICE attributes of the TRANSACTIONS relation, when not applicable to a transaction, have the values of "N/A", "NULL", or an empty field (also variations of NULL handling). I would use just NULL in this case.
- The BUY_VIN and SALE_VIN attributes of the TRANSACTIONS relation, when not applicable to a transaction, can have the values of ""NULL", "--", or an empty field (the same type of variations as above). I would use just NULL in this case.
- Sequential transaction numbers do not follow the sequence of dates (e.g. transaction 10123464 occurred later than 10123465). This is quite a mystery to me, and I would enquire the Sales Department about why they do this. Maybe they need to change their procedure and be more consistent.
- Customers' addresses do not have cities which are more convenient to look up (e.g. online) than zip codes. Need to complete addresses.
- Customer Gandalf does not have a last name. That's not right. Consult books by Tolkien for resolution.

Also, it is arguable if ASSOCIATE_NAME should be an entity or attribute. I considered it an attribute since it is just one column associated with a transaction + there is no such an attribute in the Assignment 1 database. So, even if I make it an entity now, it may have a lot of NULL values (a sign of attribute according to "Database Design Using Entity-Relationship Diagrams"), and I will still have to convert it to an attribute in the combined schema.

The ER diagram for the pre-owned dealer database looks as follows.

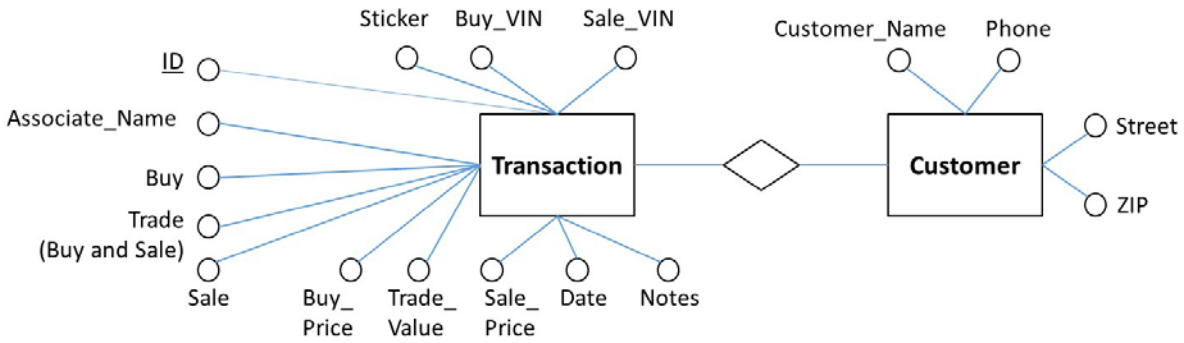

Figure 1. ER Diagram for Pre-Owned Dealer Database

There are two entities: Transaction (has one candidate key "ID" - a strong entity) and Customer (no candidate keys - a weak entity). All attributes are single-valued, atomic; depending on a transaction, some of them may be NULL.

2. Assignment 1 Schema

For your convenience, Fig. 2 shows my Assignment 1 schema

Legend:	
	Primary Key
	Foreign Key

Car Inventory	Car ID	Make	Model	Year	VIN	Drive	Color	Body	Engine	MSRP	Sold
Data type:	Varchar (10)	Varchar (30)	Varchar (30)	Smallint	Varchar (20)	Selection from a list of varchars	Selection from a list of varchars	Selection from a list of varchars	Selection from a list of varchars	Money	Boolean (or Bit)
Constraints:	No Null				No Null					No Null	No Null
Key:	PK				Alternate key						

Sales	Transaction ID	Customer ID	Car ID	Sale Date	Discount	Trade-In Value	Purchase Price
Data type:	Varchar (15)	Varchar (15)	Varchar (10)	Smalldatetime	Varchar (30)	Money	Money
Constraints:	No Null	No Null	No Null	No Null			No Null
Key:	PK	FK	FK				

Customers	Customer ID	Last Name	First Name & MI	Address	City	State	Country	Zip Code	Occupation	Needs Financing
Data type:	Varchar (15)	Varchar (30)	Varchar (30)	Varchar (50)	Varchar (50)	Varchar (15)	Varchar (30)	Varchar (10)	Varchar (30)	Varchar (30)
Constraints:	No Null	No Null	No Null	No Null	No Null	No Null	No Null	No Null		
Key:	PK									

Figure 2. My Assignment 1 Schema

An ER diagram for this schema is presented in Fig. 3.

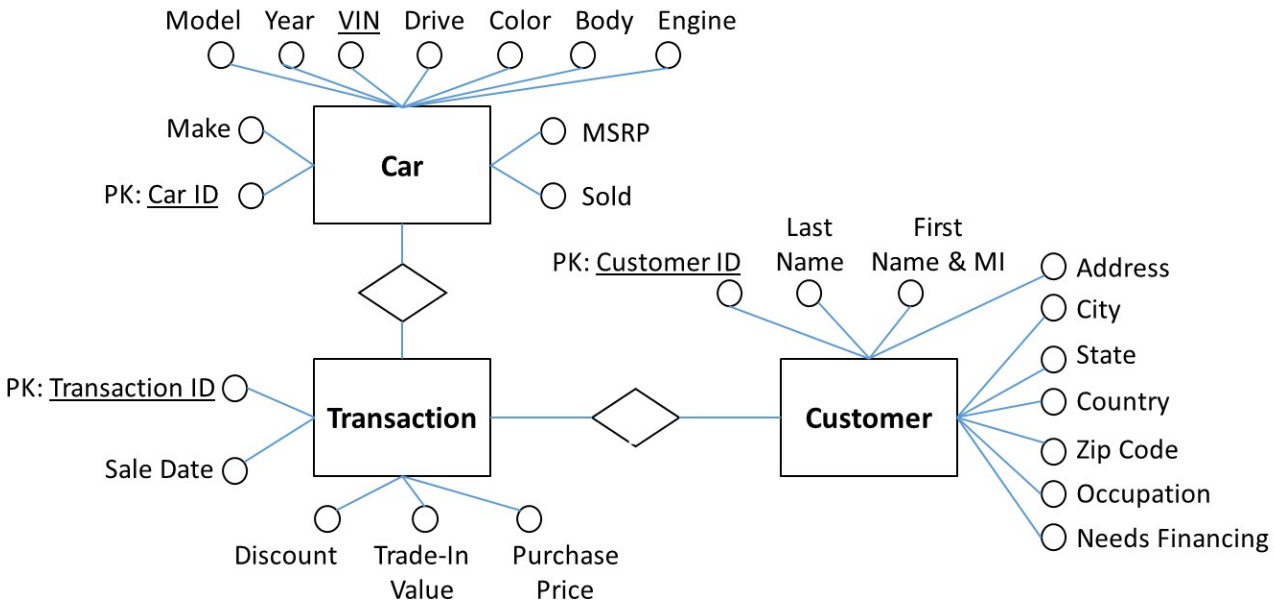

Figure 3. ER Diagram for My Assignment 1 Schema

There are strong entities with single-valued, atomic attributes. One of them (Car) has even two candidate keys with Car ID being the primary key. Primary keys are shown as PK.

3. Modification of ER Diagram for Pre-Owned Dealer Database

I made the following modifications of the initial ER diagram for the pre-owned dealer database:

- Each transaction may be associated with 2 VINs which is confusing. I created a separate entity in the ER diagram called CAR associated with a specific transition. This way, each Car entity will have only one VIN attribute. Each Transaction can be associated with only one Car, and in case of a sale with trade-in one Customer will make two separate Transactions (one for sale and one for trade-in). This should make things more straight forward. Of course, the structure of this relationship can be changed depending the needs of the database users.
- In order to simplify data representation (without loss of information), based on the BUY, TRADE, and SALE attributes from the TRANSACTIONS relation I created just one attribute TYPE in the ER diagram with a list of controlled values: Sale / Purchase / Trade In. This can be a less confusing way to represent transactions instead of using three different attributes, the majority of which will have the value of NO, anyway, or, when SALE and TRADE attributes both have a value, the transaction is associated with two different cars which can be misleading. In my schema, each transaction is associated with only one car (see bullet above). Cars, on the other hand, may be associated with several transaction because a car that is traded in may be later sold to a different customer.
- Consequently, the Transaction entity has only one Price attribute corresponding to attribute BUY_PRICE if the car is purchased, or SALE_PRICE if it is sold. Attribute Trade In Value of the new Car entity corresponds to attribute TRADE_VALUE of the initial TRANSACTIONS relation from the pre-owned database and may be NULL if no trade-in occurred.

- The Sticker attribute of CAR has a numeric value if there is a sticker price; otherwise, it is NULL.
- Note: The above inconsistencies and transformations can be used to clean and improve the relations in the relational representation of data shown in Exercise3_FaileA.xls.

Fig. 4 shows the results of these modifications.

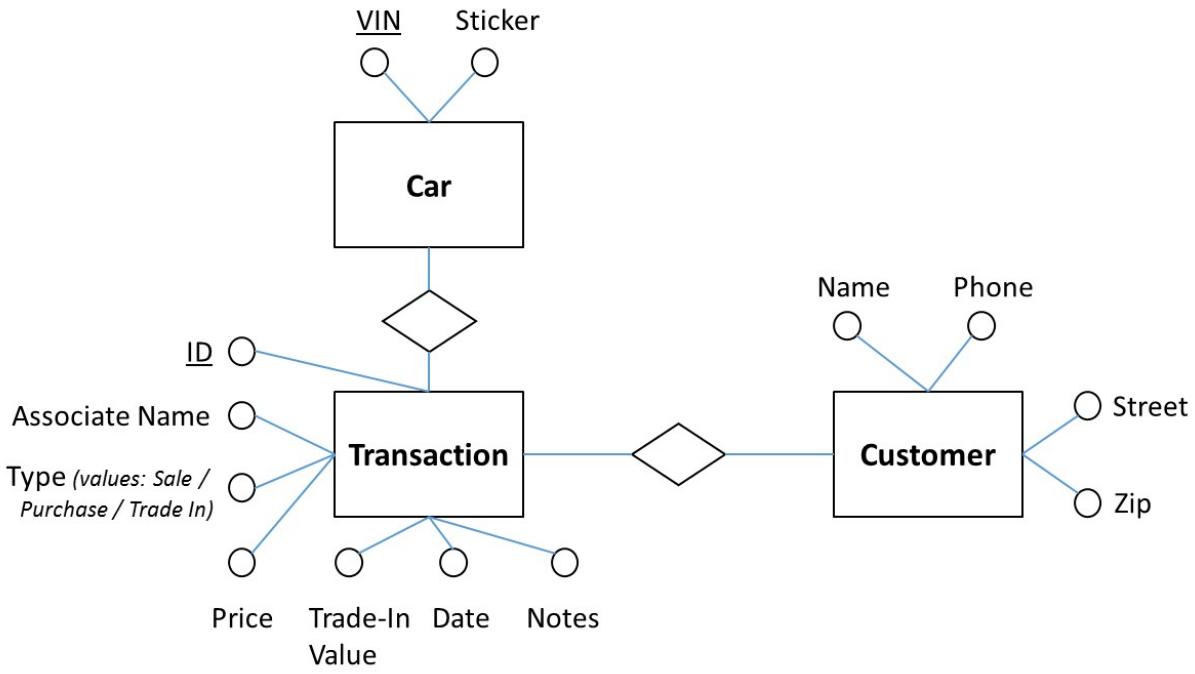

Figure 4. Step 1: Modified ER Diagram for Pre-Owned Dealer Database

There are two strong entities (Car and Transaction) with one candidate/primary key and one weak entity. All attributes are single-valued, atomic (including Type which has a list of controlled values, but can have only one value in a single transaction – so it is single-valued).

4. Preliminary Combined Schema

You can see now that the two schemas have become similar in the structure of entities, and the most significant difference is now the number and composition of attributes for the entities.

The Sticker and MSRP attributes of the Car entity appear to be synonyms. So are Transaction ID and ID, Sale Date and Date, Trade-In Value attributes of the Transaction entity, and Address and Street, Zip and Zip Code attributes of the Customer entity. In all of these cases I selected the shortest OR the most reasonable name for an attribute in the combined ER diagram.

Since in the Assignment 1 database the cars obtained by the dealer through the trade-in process were not reflected in the database, the Price, Sale Price (as worded in the pre-owned dealer database), and Purchase Price (as worded in Assignment 1) attributes are synonyms in the case of a sale and can be combined. The cars from the Assignment 1 database will simply have no Buy Price, and the Type of the transaction will be Sale.

The Sold attribute of the Assignment 1 Car entity is merged with the Type attribute of the pre-owned Transaction entity.

The Name attribute of the pre-owned database has to be divided into the Last Name and First Name & MI as it was done in the Assignment 1 database. The keys for the Car, Transaction, and Customer entities are Car ID, Transaction ID, and Customer ID, respectively.

Fig. 5 below shows the preliminary combined ER diagram for the two databases based on derivation without any loss of information. There are no homonyms or conceptual overlaps in the two databases.

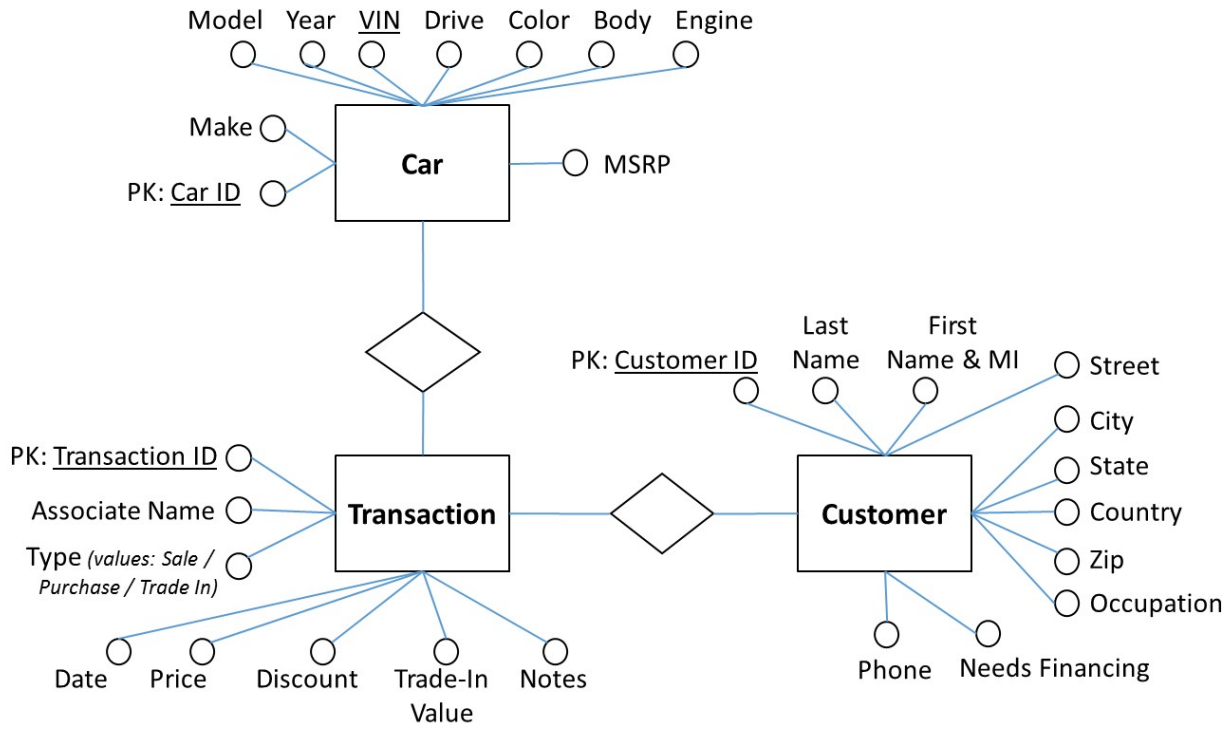

Figure 5. Preliminary Combined ER Diagram

Now we finally have three strong entities with single-valued, atomic attributes generated through the derivation process. Primary keys are shown as PK.

5. Final Combined Schema

I propose to introduce additional features in my ER diagram in order to make it more complete and logical.

- First, the names of the relationships: a Customer **makes** a Transaction, and a Transaction **contains** a Car
- Secondly, let us set the cardinality ratio of the relationships and denote whether the participation is partial or full.
    a) Basically, one Customer must have one or more transactions (a sale with trade in or simply several sales), but each transaction must be associated with only one Customer. Consequently, this is a one to many relationship with a full participation both by the Customer and by Transaction (Customers without Transactions will simply not be entered into this database; if the dealership purchases a used car from another dealership / company, but not from an individual, then that other dealership / company will be the Customer in this case).

b) Similarly, there is a one to many relationship between Transaction and Car with full participation on both sides: each Transaction must have only one Car, but every Car must be associated with [at least] one or [maybe] more Transactions (when a used car is purchased / traded in and then sold).

- Finally, I would like to introduce composite attributes in order to group them because of their great number and make things look more logical. In the Customer entity, the composite attribute Name has now two subattributes Last Name and First Name & MI, and the composite attribute Address has now five subattributes Street, City, State, Zip, Country

Fig. 6 shows the final combined ER diagram.

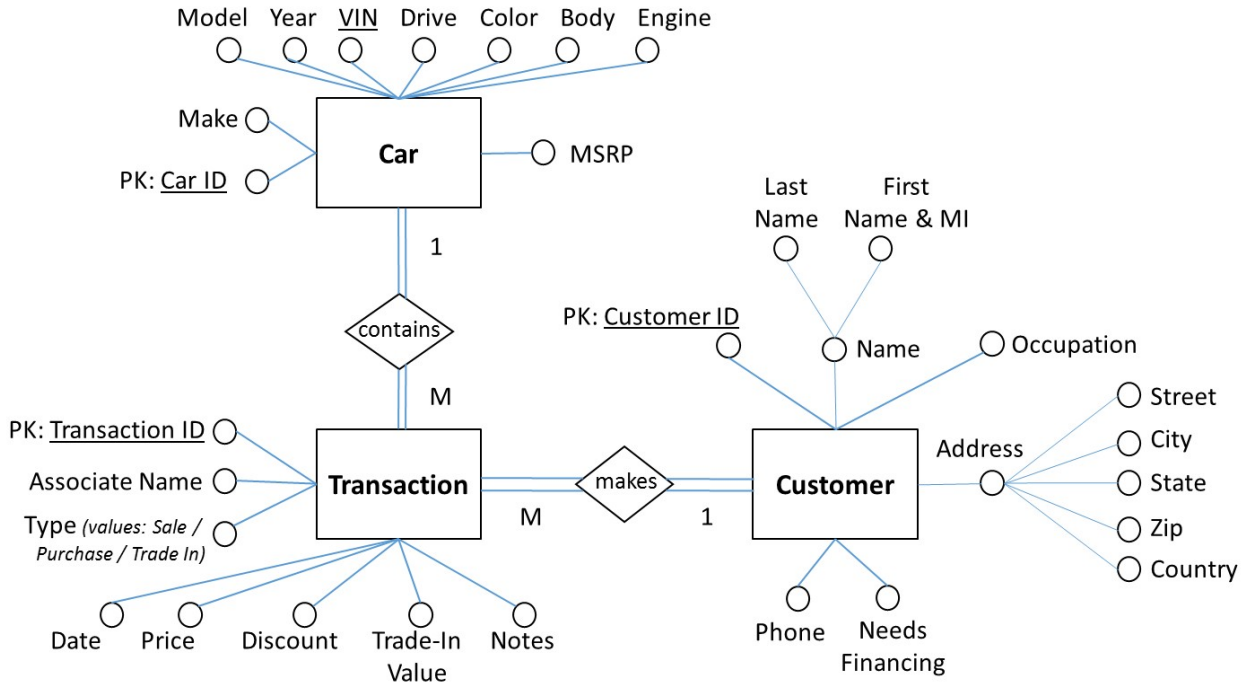

Figure 6. Final Combined ER Diagram

It should be noted that there are other possible cardinality ratios and types of participation for the shown entities, as well as other ways to allocate attributes to the entities. E.g. the Type attribute may be associated with the Car entity, after being renamed to let's say Status, with the same controlled list of three values in which case the Transaction entity can now contain more than one Car and will have slightly different other attributes. I have provided just one possible implementation, and other options will depend on the needs of a particular database owner.