INTRO TO DATABASE PART TWO

Information Structuring in Databases

Let's briefly look at some of the processes required to codify data and turn it into meaningful information. Interesting questions about information include, "What is information?" "How does it exist in the world?" and "How are databases used to help organize, structure, record, and provide information by identifying new relationships that were not seen previously?"

There are many different definitions and philosophical concepts about information. Our short discussion will barely scratch the surface of defining information. However, we can note that information is more than just data or facts. For our purposes, let us define information as a set of data facts that (1) can be labeled, (2) are identified within a larger context, and (3) have identified relationships with other facts. For example, if we have a piece of data such as "2875039275," that does not provide much information. However, if I label it as a phone number, then it begins to contain meaningful information. If I add that this phone number belongs to John Appleby, then I have added both a context and a relationship, and even more information is available to me.

Additional information is added if I identify that it is a cell phone number; now I know I can send a text message using that number. We also gain additional information on phone numbers by adding more contextual and relationship information. This might include whether a particular phone number is a business telephone or a residential telephone, and whether it is land line, with a physical location, or a cellular, with a current cell tower location.

As you can see, meaningful information is more than just a set of facts. Databases are uniquely qualified to store a large set of data facts, attach labels to them, maintain data about the context, and establish important relationships.

Entities and Attributes

Every database will contain information about many different entities. A database schema will describe both the details about each entity and the relationships between entities. Entities usually come from the physical world items such as Customers, Vehicles, or Purchases. Detailed information about the properties of each entity are called attributes. For example, attributes of Customers will be things like Name, Address, and Phone Number. The database schema will identify all the entities to be included in the database and the attributes for each entity. The schema describes the structure of the data in the database.

Figure 1-2 is an example of a visual data model of some of the information that is maintained in a database as part of the schema. This diagram is a partial database model for a shoe retail store. This shoe store maintains detailed information about its customers. Information about each sale is kept in the database along with the individual pairs of shoes that are sold (SaleItems) as part of the sale. Other information includes product information (which are shoes) and the manufacturers of the shoes. As you can see in the figure, there are six different entities as identified by the boxes.

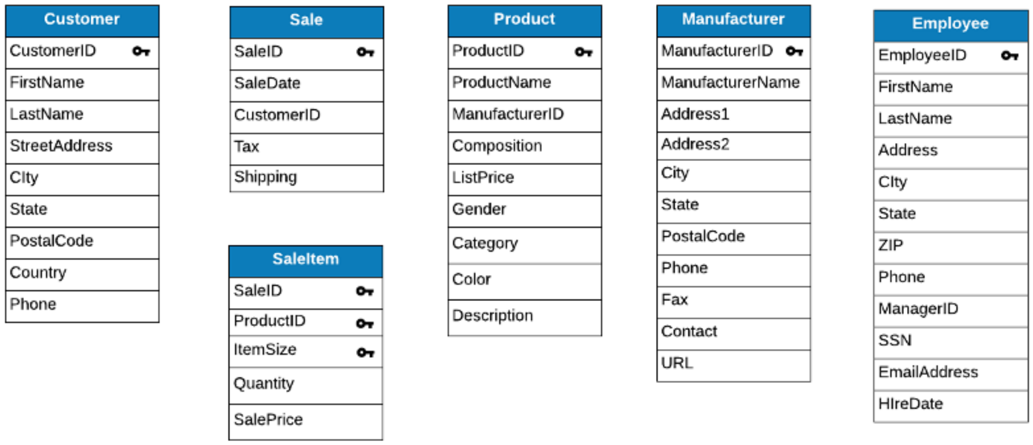

Figure 1.2: Data Entities for a shoe retail store

Within each box is a list of the attributes for each entity. The attributes define the detailed information that is kept about each entity. The entities and their list of attributes are part of the database schema. In Figure 1-2, we see that attributes for Employee are EmployeeID, FirstName, LastName, Address, and so forth.

Each entity has one or more attributes that is defined as a key attribute. In the figure above, we have identified those attributes with a small key icon in the attribute rectangle. For example, for a Customer entity, the key attribute is CustomerID. A key is a defined as an attribute whose value is unique or distinct for each record in the entity. Notice that a SaleItem requires three fields as the key - SaleID, ProductID, and ItemSize. It takes all three attributes to uniquely identify each SaleItem. Examples of key values are given below in the explanation of data.

Relationships

The database also needs to capture and describe information about the relationships between the entities. In this example we want to know which sales were done by which customers. Sale Items are associated with each sale. In other words, a sale may include several pairs of shoes. Each sale item, which is a pair of shoes, is described in detail by product information. Finally each product is manufactured by a particular manufacturer. The visual data model in Figure 1-2 is not sufficient to provide this information. Figure 1-3 expands the model by defining the important relationships. These relationships are identified by lines connecting the boxes of related entities.

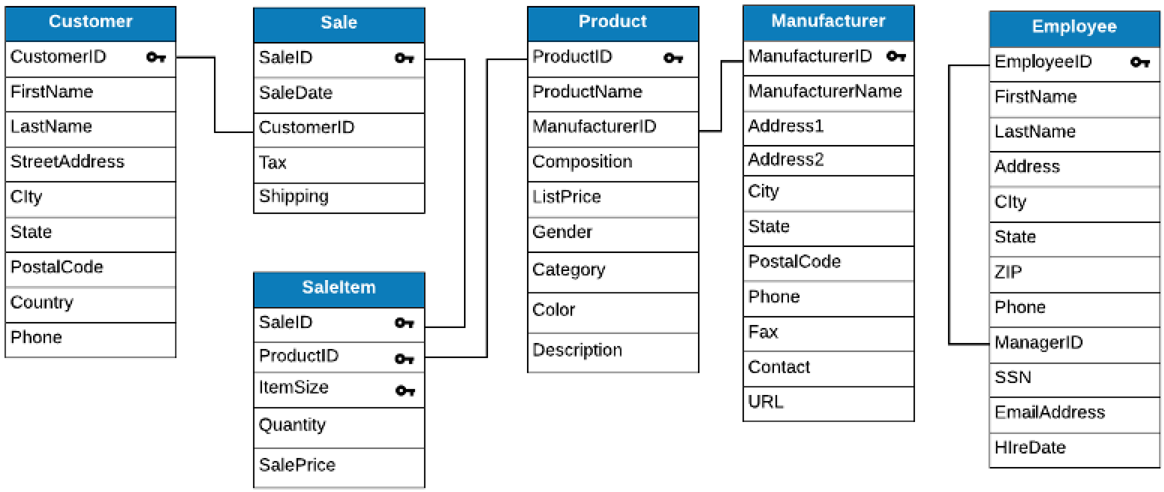

Figure 1.3: Data entities with relationships for a shoe retail store

One good way to capture the relationship information is by using an entity's key. Because the key uniquely identifies a record, if we put that key value in the data of a related record, then the two records are related to each other. In Figure 1-3 the Sale entity includes the CustomerID attribute. Thus we know which customer purchased items on a particular sale. The Product entity includes the ManufacturerID as a field. Thus we know who the manufacturer is for any given product. When a key attribute for one entity is placed in a different entity, it is called a foreign key in that different entity. In other words, it is a key attribute, but it belongs to a different or foreign entity. Notice that a foreign key may also be a key attribute. In SaleItem, the SaleID is a foreign key because it is the key for a Sale. It is also part of the key for a SaleItem

There is also a relationship that connects an entity with itself. This is called a recursive relationship. For an employee, the manager is also an employee, so the ManagerID value must refer to an EmployeeID.

This visual data model is an easy, and powerful, technique that is used to understand the schema for a given database. Obviously, the DBMS must contain this information within the database schema so that it can organize and maintain the correct data as shown in Figure 1-3.

Data

As indicated above, the database schema describes the structure of the database. The schema is not the same as the actual data. Each one of the entities, along with its attributes, can be thought of as a database table. A database table is like a spreadsheet with the name of the spreadsheet the same as the entity name, the attributes are the columns of the spreadsheet, and the rows of the spreadsheet are the data. In database terminology, the rows called rows or records. Figure 1-4 illustrates this concept for the Employee entity.

Select * from Employee

EmployeeID	FirstName	LastName	Address	City	State	ZIP	Phone	ManagerID	SSN	EmailAddress	HireDate
5	Louis	McDougald	26 Ventura Drive	Scotts Valley	CA	95066	831-438-7850	5	557-77-0715	LouisHMcDougald@gustr.com	2001-10-25
17	Audrey	McLeod	2083 Jim Rosa Lane	San Francisco	CA	94107	415-762-4047	5	626-94-4419	AudreyMMcLeod@fleckens.hu	2002-02-28
35	Kyle	Sisk	2441 Cimmaron Road	Garden Grove	CA	92643	714-636-6046	5	624-27-0067	KyleTSisk@jourrapide.com	2000-06-20
45	Zachary	Cortez	289 Brown Bear Drive	Los Angeles	CA	90017	951-603-6990	5	569-79-8855	ZacharyECortez@fleckens.hu	2006-07-10
59	Thomas	Vaughan	3758 Armbrester Drive	Irvine	CA	92614	310-526-5232	5	620-15-4370	ThomasDVaughan@superrito.com	2006-11-05
61	Jerry	Scranton	1767 Reynolds Alley	Paramount	CA	90723	562-346-7291	17	622-18-9025	JerryEScranton@superrito.com	2011-06-26
68	Helen	Pauley	4332 Park Street	Livermore	CA	94550	925-424-4804	35	610-26-2945	HelenAPauley@dayrep.com	2009-10-17
74	Joseph	Butler	2817 Clarence Court	Los Angeles	CA	90017	909-998-7106	35	618-39-2983	JosephFButler@fleckens.hu	2010-03-13
82	Jessica	Botts	1706 Red Maple Drive	Los Angeles	CA	90031	323-441-6786	35	564-78-5083	JessicaABotts@dayrep.com	2005-04-19
99	Virginia	Hart	201 Diane Street	Oxnard	CA	93032	805-385-4096	35	572-20-9302	VirginiaDHart@gustr.com	2006-08-07
102	Kristen	Browner	1490 Pike Street	San Diego	CA	92126	858-566-5833	45	611-32-7709	KristenBBrowner@einrot.com	2002-08-16
115	Jerry	Myers	4094 Doctors Drive	Santa Monica	CA	90401	310-434-2891	45	616-43-7170	JerrySMyers@teleworm.us	2008-05-23
135	Ronald	Lynch	875 Pike Street	San Diego	CA	92123	858-587-5074	45	551-69-3204	RonaldJLynch@einrot.com	2012-01-09
155	Rickey	Dietz	4405 Beech Street	Antioch	CA	94509	925-756-5184	59	562-88-2667	RickeyCDietz@armyspy.com	2010-06-24
158	Dorothy	May	504 Black Oak Hollow Road	Sunnyvale	CA	94089	408-756-3631	59	606-38-3557	DorothySMay@armyspy.com	2009-05-29
160	Sergio	Silver	4622 Davis Avenue	San Francisco	CA	94107	707-759-6359	59	618-30-7224	SergioCSilver@armyspy.com	2008-02-13

Figure 1.4: Employee data

It is important not to confuse the database schema with the actual data. The schema defines the structure and the data is the actual information about each individual Employee or Establishment or Inspection.

Earlier we introduced the idea of a key attribute. We can see from the data how the key attribute serves as the mechanism to uniquely identify each row or each record in the table. We could have two employees which have the same name or that live in the same house. The key attribute solves this problem by guaranteeing to have a unique value for each record. We are familiar with this concept. Your bank account has a unique number. Your medical records are identified with a key, which may be your Social Security Number. In the example, we have used a somewhat standard notation by naming the key with "ID". At times key fields are also identified by a "_no" or "_id" as part of the name. The DBMS enforces this requirement that no two records in the same table have the same key value.