Management Information Systems, 15TH ED.
MANAGING THE DIGITAL FIRM
Kenneth C. Laudon • Jane P. Laudon

Chapter 6: Foundations of Business Intelligence: *Databases and Information Management*

Learning Track 1: Database Design, Normalization, and Entity-Relationship Diagramming

This Hands-On Guide will show you how to design a relational database system for a small business using normalization and entity-relationship diagrams. The system we will be developing is for a small but growing barber shop/hair salon business. The shop, named HisNHers, is owned by Clarence and Clarissa. They have big plans to expand soon and believe their system of maintaining customer information on index cards can no longer support their business needs.

Information Gathering

Your first step is to gather information about how the new system will be used, what information the user needs, how a new system can speed up and simplify operations, as well as how the system could help the business to grow. A database is a model not only of reality but also of the future. If there is a need to know information which is not yet stored anywhere or does not currently exist, room for this data should be included in the system design.

Both Clarence and Clarissa want a system that can maintain information about their business with over 300 customers. Also, they want to be able to begin tailoring advertising to particular customers based on their profiles and to send birthday cards containing special offers.

Designing the Database: A Conceptual Schema

To begin developing a conceptual schema of the system, Clarence and Clarissa (and perhaps their employees) need to describe their business. You will need to look at any paperwork generated by the business, starting with those index cards, which contain a lot of information about each customer. But you'll also look at the phone log, the appointment book, any financial reports that are generated, and perhaps talk to their accountant and/or see if they are using a small accounting package such as QuickBooks, and study what they do with it. There is often informal written information that is also good to know about—do the staff at HisNHers keep their own notes about anything?

continued

The HisNHers Salon now receives payment immediately after the service is performed. But with the proper design of a database, a billing system could be integrated. Targeted marketing could be performed whenever a new product (a new purple hair rinse for all the gray-haired customers) becomes available.

HisNHers is a small business where a personal computer running Microsoft Access would be an appropriate platform for its new system. Since you are most likely using Access in your course work, we will illustrate the design of the database as it would be implemented in Access.

After completing your information requirements analysis, your next task it to develop a conceptual schema of the database. One problem that can arise in this step is that the discussions with users may have shown multiple views of the system. These multiple views will need to be integrated into a single conceptual schema.

For example, at HisNHers, both Clarissa and Clarence often referred to "products." Only after Clarence said that they might like to keep a "product" inventory and Clarissa mentioned that a customer made an appointment for a "product" did you realize that Clarence meant items such as a package of hair-coloring and Clarissa meant services, (e.g., hair cuts). This sort of double meaning for a single term is called a homonym and needs to be resolved before an integrated view of the system can be developed. In a similar fashion, when Clarence referred to the "customers" and Clarissa referred to the "heads," it was clear that "heads" was a synonym for "customer." Unfortunately not all synonyms are so obvious.

In a large organization, personnel with different jobs often see the data in light of their own job function. For example to the colorist applying the hair-coloring, vendor is only an attribute of the product. To the person doing the ordering, vendor is an entity with attributes of its own such as address and phone number.

Entities, Relationships, and Normalization

Once a cohesive view of the system is established, the entity classes (tables) need to be defined along with their attributes (fields). Review the discussion of entities, attributes, and relational databases in Chapter 7. Your requirements analysis showed that two main documents are central to HisNHers' operation—Clarissa's index cards and the salon's appointment book.

Clarissa used the index cards to store information about the customers of the salon. They contain: the customer's name, sometimes the address, usually the home phone number and/or a cell phone number, a small number scribbled in the corner, a few notes about the customer's social life, plans, or children, and occasional comments about a particular appointment—(e.g., used Clairol #12 on 10/15).

The appointment book contains the date and time of the appointment, the customer's name, phone number, service requested (e.g., hair cut, beard trim, frosting) and staff member who will do the job.

continued

These documents indicate the need for both a CUSTOMER Table and an APPOINTMENT Table. But, what other tables are needed? What information belongs in each table? How should the tables be linked together? It is important to start slowly and carefully, building on what is obvious. Sometimes it helps to visualize one instance of the data (e.g., one customer and his appointments). Begin sketching out a simplified entity-relationship (ER) diagram to help you visualize the system's tables and relationships. The verbs used for describing the relationships in the ER diagram are often helpful, (e.g., a customer *makes* an appointment, a service *uses* a product).

At HisNHers, Clarissa's index cards contain the beginnings of the Customer entity. The entity known as the CUSTOMER Table will contain demographic data about the person as well as a notes where special "conversational" or unusual items about the customer can be stored. An initial breakdown of the attributes of this entity (or putting it more simply, the fields for the CUSTOMER Table) along with their type (as represented in Access) and size are:

FIELD NAME	TYPE	SIZE
CUSTID	Long Integer	4
FNAME	Text	20
MIDNAME	Text	15
LNAME	Text	20
ADDR1	Text	30
ADR2	Text	30
CITY	Text	20
STATE	Text	2
ZIP	Text	5
HOMEPHONE	Text	10
WORKPHONE	Text	10
CELLPHONE	Text	10
EMAIL	Text	50
DOB	Date/Time	8
SEX	Text	1
PERSONAL_INFO	Memo	–
OLDCUSTNUMBER	Integer	3

The customer name field should be broken into first name, middle, and last name. This will insure the ability to sort reports based on last name and allow for personalized reports which use just the first name. Breaking up data into its smallest usable elements is known as *atomization*.

While HisNHers doesn't currently have all of this customer demographic information, providing for these fields will allow the system to grow and meet Clarence's desire for targeted marketing and a possible billing system. The last field in the CUSTOMER Table is the "old customer number."

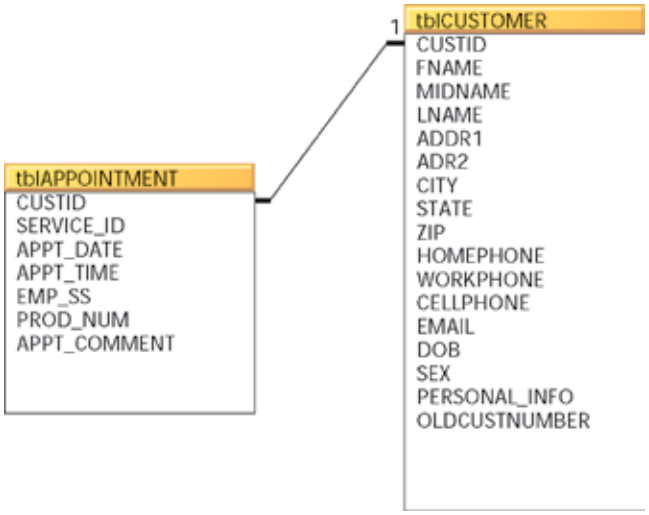
continued

It is taken from Clarissa's index cards and will allow her to use both the old and new customer numbers until she is comfortable with the new system. This is often a good practice when converting manual or legacy systems (those done in an older computer program).

Discussions with the staff and studying the appointment book indicate that another entity would be the APPOINTMENT Table. It would include the following attributes (fields).

FIELD NAME	TYPE	SIZE
CUSTID	Long Integer	4
SERVICE_ID	Long Integer	4
APPT_DATE	Date/Time	8
APPT_TIME	Date/Time	8
EMP_SS	Text	9
PROD_NUM	Long Integer	4
APPT_COMMENT	Memo	—

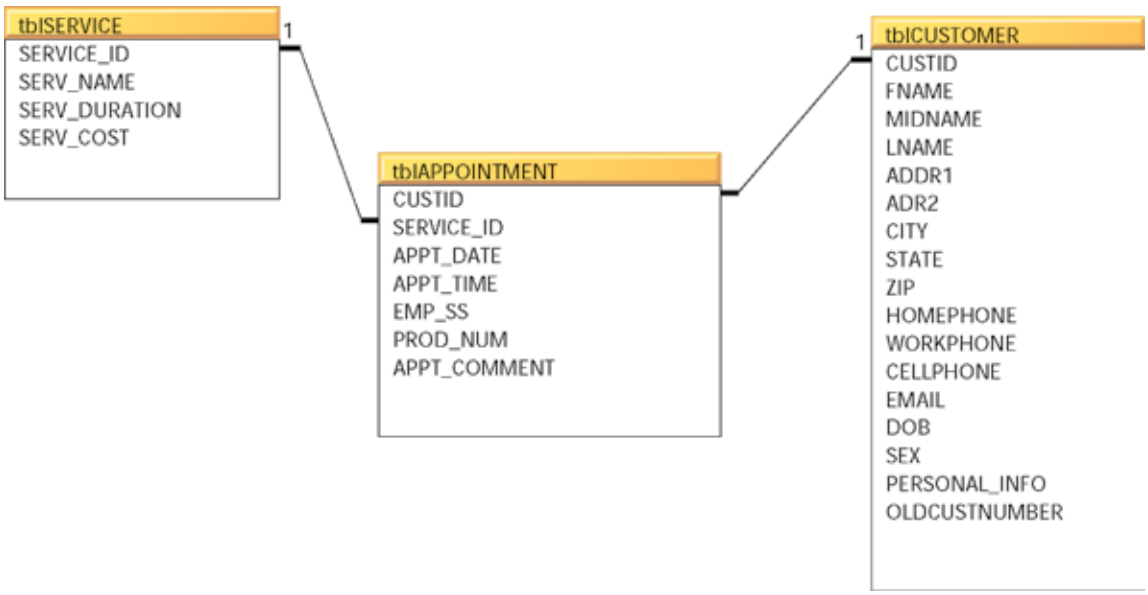
The Appointment Book shows that customers make appointments. In this sentence lies the beginning of the table relationships in the HisNHers database. The ER diagram which shows the relationship between these initial two tables is:



As the system develops, we will add entities and information to this ER diagram until the entire conceptual schema for the database is illustrated.

There are a number of different formats for entity-relationship diagrams. Since we are using Microsoft Access to build the database, we will use ER diagrams as they are shown in Microsoft Access's table relationship view. In these ER diagrams, tables are shown with all of their fields. Related tables are shown via lines which link their key fields together and show whether the entities have a one-to-one or one-to-many relationship. A small number 1 next to the key means that a table is on the one side of a relationship. A small infinity symbol indicates that the table is on the many side of a relationship. For example, one customer makes many appointments.

continued



Because the CUSTOMER Table will be linked to the APPOINTMENT Table, it will not be necessary to include either the customer name or phone number in the APPOINTMENT Table. The CUSTOMER Table and APPOINTMENT Table will be linked based on a key (which is not the customer's name). Thus, only the key field from the CUSTOMER Table needs to be included in the APPOINTMENT Table in order for all of the information about a customer to be available.

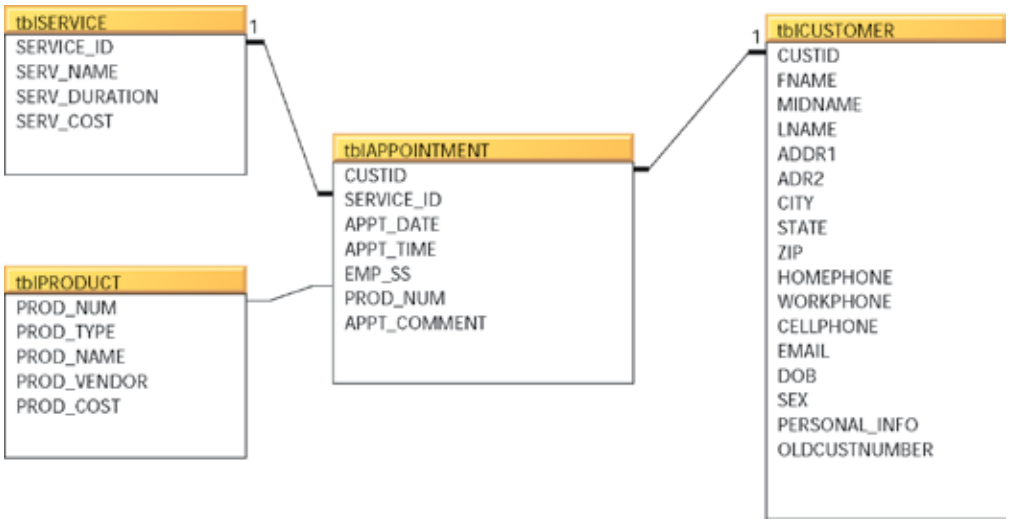
If a field from the CUSTOMER Table, like the phone number, were to be included in the APPOINTMENT Table, it would have to be entered each time an appointment was made. Also, if the phone number were to change, each record for that customer in the APPOINTMENT Table would have to modified. *No entity ever needs to include attributes from an entity to which it is linked.* Doing so would cause repeating information, which would violate the first rule of normalization, which states that there should be no repeating groups or many-to-many relationships among entity classes.

HisNHers appointment book includes the service provided. This could be an attribute of the CUSTOMER Entity since many customers have many services performed. However, this would create a many-to-many relationship. A many-to-many relationship requires a join or linking table (sometimes called an intersection relation) to prevent repeating data. The APPOINTMENT entity is, in fact such a join table. The APPOINTMENT entity joins the customer with the service provided on a particular date. Other details about a particular appointment such as the employee who performs the service and the product used are also included. These details or attributes relate only to this particular appointment, not to the customer or service.

For any one CUSTOMER record there may be many related APPOINTMENT records. This is indicated on the ER diagram by the small 1 next the CUSTID field in the CUSTOMER Table and the infinity symbol next to the CUSTID field in the APPOINTMENT Table. Similarly, the small

continued

1 next to the SERVICE_ID field in the SERVICE Table is linked to the SERVICE_ID field in the APPOINTMENT Table with an infinity symbol indicating that for any one type of service there may be many related records in the APPOINTMENT Table. Simply put, Justin Jumpup can make many appointments in the APPOINTMENT Table but there is only one Justin Jumpup in the CUSTOMER Table. (If there were two customers named Justin Jumpup each would have a separate CUSTID.) Hair Cuts can be given in many appointments but there is only one type of service called Hair Cut.



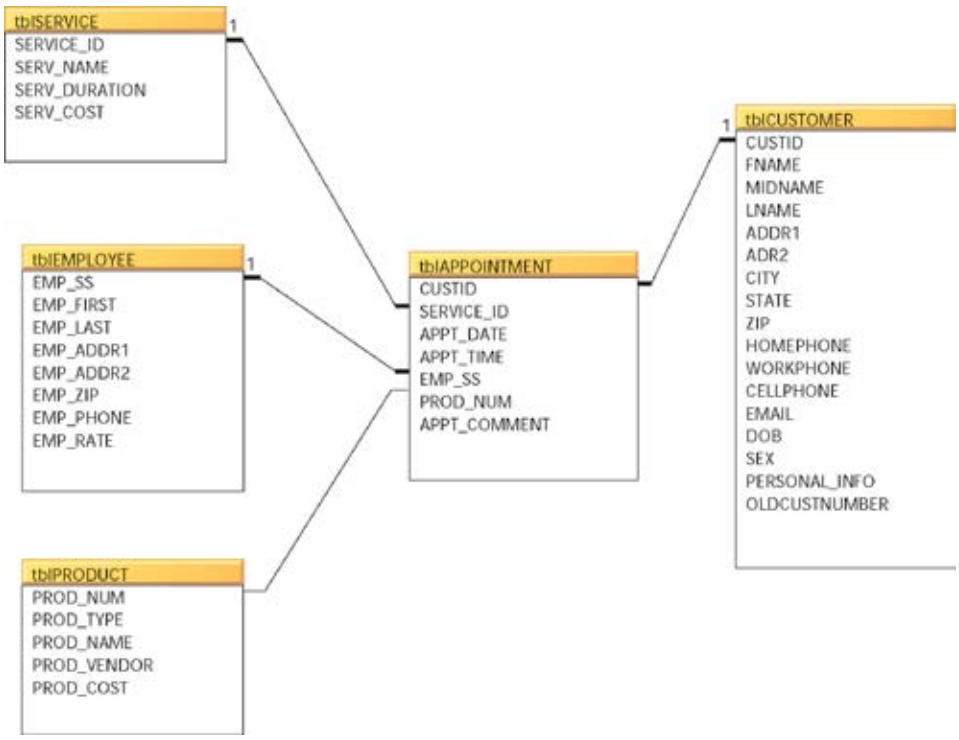
The service (Hair Cut, Perm, etc.) could have been an attribute (a field) of the APPOINTMENT Table but the SERVICE itself has an attribute of Duration (e.g., Hair Cut 1¼2 hour, Coloring 2 hours). *Whenever an attribute itself has attributes, it must become an entity unto itself.* If it does not, it will violate the second rule of normalization, which says that each non-key field must relate to the entire primary key field, not to just one part of it. The key field of the SERVICE entity will be included in the APPOINTMENT entity as a foreign key in order to link the name of the service and its duration.

Each SERVICE may or may not include a Product (e.g., haircuts require no product, but a hair frosting requires a colorant). Once again, the attribute of Product itself has attributes (e.g., product type, product number, vendor etc.). Thus, PRODUCT becomes an entity. Its relationship differs slightly from that of APPOINTMENT to SERVICE since not every SERVICE requires a PRODUCT. This information is shown on the ER diagram by the line which links the tables with no small l or infinity symbol present next to the joined fields.

The APPOINTMENT entity could have as an attribute the employee who will provide the SERVICE. However, yet again the employee him/herself also has attributes such as first name and last name. So a separate entity for EMPLOYEE will be created and it will be linked to the APPOINTMENT entity by its key.

The attributes of the APPOINTMENT entity consist almost entirely of foreign keys which link it to information contained in other entities. It is on the many side of these relationships.

continued



The conceptual schema for the HisNHers Salon now includes five separate entities: CUSTOMER, APPOINTMENT, SERVICE, EMPLOYEE, and PRODUCT.

Determining Key Fields

To ensure that each record of an entity class is unduplicated, it must contain a field which is its unique identifier or key. Often referred to as the primary key, this field becomes the link from one entity to another. When the primary key of one entity is used as a field in a table to which it links, it is called a foreign key. For example the field CUSTID in the CUSTOMER Table is its primary key. When the CUSTID field is used in the APPOINTMENT Table to link it to the CUSTOMER Table, it is called a foreign key. Within the physical database, keys are used also used as indexes to speed up record retrieval.

The value of a key field in a record should never change and much care should go into selecting it. Guidelines for selecting a key include simplicity and stability. For the EMPLOYEE Table, Social Security Number is an obvious choice for a key field. This number is known to the employer, is unique, and never changes.

You might be tempted to use the Social Security Number as the key in the CUSTOMER Table as well. However, many people are protective of this number and not inclined to give it out. Another possible choice for the CUSTOMER Table key is the last name but there are can be several customers with the same last name and women frequently change their last name. It is also possible to create a composite key of two or more fields in a table so that first name and last name could be joined into one field. But in this case such a composite key would not ensure a unique record (e.g.,

continued

Maria Jones and her daughter, Maria Jones). The CUSTOMER entity requires the creation of a new attribute to be used as the primary key.

Often a composite key of date and time is a good choice for an entity that is time based, since it precludes duplicates. For the HisNHers database, the APPOINTMENT Table could use such a key if the service provider's ID were added. (Two staff members frequently have appointments at the same time.)

Often the user wants keys that are meaningful to him. For example, in the SERVICE Table, Clarence would like to create a key based on the name of the service, like HCM for male haircut. BT for beard trim, and CF for coloring female. However, in a large database it is easy to run into duplicates with schemes like this, and the keys can quickly become meaningless to all but the person who established them. In a small system, such as HisNHers these problems are minimal, and using familiar abbreviations might make learning to use the new system easier.

If you have the luxury of being able to set up a new primary key, one of the best choices is to allow the system to create an automatically numbering field for each record. The CUSTID field in the CUSTOMER entity is such a field. This will ensure that keys are unique and unduplicated. It is seldom necessary for the user to even be aware of the contents of a primary key field. The job of the primary key is to link tables. Today's database products include powerful look-up features so that you would not need to look up a customer by their ID number but could use their last name, then the first name, then the address, etc., until the requested record is located for use.

In many businesses some sort of key already exists for an entity. For example, Clarence and Clarissa had been using a small number that had been written in the corner of each customer's index card, which could become the key. When you are using existing numbers, gaps in a numbering scheme do not matter but care must be taken that the numbers to be used in the primary key field are both unique and not duplicated, which is often not the case with a key that is used in a manual system.

If a key has been in use for a while on an older system but is found to no longer be appropriate as a primary key, you may need to carry this information into the new system, to help the user identify the records. For example if Clarissa's index card numbers were inappropriate to use as a key field, but she had typed up notes about her personal customers using these numbers instead of names, you would carry the old key as a field (but not a key field) in the new database. When reports are printed from the new system they will include this old number so she can match the new reports to her old notes.

Reviewing the Design

After the "final" conceptual schema has been developed and the ER diagram includes all of the entities with their keys, attributes, and relationships, it is time for a design review. All of the principal stakeholders in the system should study it to be sure that the necessary information is

continued

included and available. Also, provisions for possible expansion should be included. If the system expands to include inventory and ordering, a sixth entity, VENDOR, would have to be added. Allowance has been made for this growth in the PROD_VENDOR field of the PRODUCT entity. While it isn't impossible to redesign a system after data is entered, it is time consuming, error prone and requires serious skill in handling data via SQL commands.

Developing the User Interface: Forms and Reports

The user interface includes menus, data entry forms, and reports. In short, despite your beautiful ER diagram, the interface is what the user will think of when he or she thinks of the database. The tools to create these objects vary significantly among database packages. In Microsoft Access, the development of both entry forms and reports can be quickly accomplished via wizards that create basic objects and then allow them to be customized.

HisNHers Shop APPOINTMENTS: November 05, 2004

TIME	CUSTOMER	PROVIDER	SERVICE	PRODUCT NEEDED
8:30 AM	Malone, Patrick	Clarissa	Hair Cut	
8:30 AM	MacGregor, Mark	Clarence	Hair Cut	
9:00 AM	Smith, Mary Allen	Clarence	Coloring	Henna
9:30 AM	Greene, Jeanne	Clarissa	Blow Dry	Spray2Set
10:00 AM	Montague, Juliet	Clarence	Set	Spray2Set
10:30 AM	Brown, Thomas	Clarence	Consultation	
11:30 AM	Jumpup, Justin	Clarissa	Set	
12:00 PM	Smith, Mary Allen	Clarence	Coloring	EasyGlo Gold
1:00 PM	Nother, Zoe	Clarissa	Permanent Wave	
2:30 PM	Smith, Mary Ann	Clarence	Blow Dry	Spray2Set
3:00 PM	Smith, Mary Ellen	Clarissa	Hair Cut	
3:30 PM	Brown, Georgia	Clarence	Hair Cut	
4:30 PM	Jones, John Paul	Clarence	Hair Cut	

Development of reports for the system can teach you much about the design of the system, and it is critical to have sample reports examined by the user. If the owners of HisNHers wanted to see the customer's hair color on a report, you would soon realize that there is no field for this information. While it's possible to change the database to include a field for hair color, this type of oversight should have been caught at an earlier stage of the design process.

The owners of HisNHers wish to have a daily appointment list which shows not just the customer and appointment time but also the provider, the service requested, and any special product that will be needed. Above is a sample from this report. Creation of this report requires data from

continued

all five of the tables in the database. If tables are improperly related to each other, reports will be difficult if not impossible to create.

Since each table is related to another by a key field, records will be linked together based on matching data in the key fields. If a record in one table has no matching record in another table, it will not be included on the report. For example, only records from the tblAPPOINTMENT in which the appointment date = November 5, are selected for the 11/5 appointment list. If the group of records selected from the APPOINTMENT Table where appointment date = 11/5/04 contain no record with the customer ID for Jim Brown, Jim Brown will not show on the appointment list. Since Jim didn't have an appointment on 11/5, he shouldn't show!