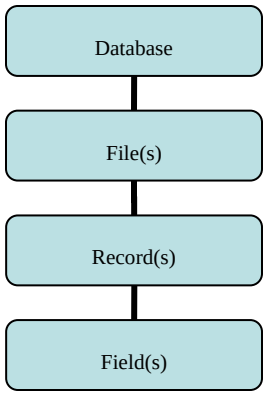
Database – A Brief Introduction

I. Basic Definitions

A **database** is a collection of data that is stored in a computer system. The software that manages the data is known as a database management system or **DBMS**. Suppose I wished to collect information about students in my class. First I would have to decide on which items I wished to store. I might choose such items as *firstname, lastname, age, major* and so on. These elementary items are known as **fields**. The complete set of information for each student is called a **record**, while the complete set of records for the entire class is known as a **file**. Each record represents the information on a single student. While each record consists of the same fields, obviously the information contained in those fields will be different. A field which is left empty is known as **null field** (e.g., *middleintial*, for students who do not have a middle name.) The choice of fields requires careful study of the purpose for which we are collecting information and is both an art and a science.

Of course, I am not the only one in the college who maintains information on the students in my class; the Registrar has their own files consisting of student academic records; the Bursar maintains a file of financial information; and the Nurse maintains yet another file. Each file contains different records on each student, yet taken together the three files represent all the information that is maintained on the students in the college. The collection of files is known as a database.



Thus a database is a collection of files, which in turn is a collection of records; each record is a collection of fields.

Each record in the database is identified by a **key.** Keys may be distinguished on the basis of whether they are **unique** (e.g., a student id) or **non-unique** (e.g., zip codes). Keys are often fields in a record, or they may be created arbitrarily (such as a magazine subscription code which consists of one or more characters of your name, address and expiration date.)

Keys may also be distinguished by whether they are designated as **primary keys** or **secondary keys**. A primary key is used to represent the **physical ordering** of the records in the file. For example, the Library of Congress Number is used to arrange the books in a library. Often however, we wish to access our records in a different order than their physical arrangement. It is convenient to imagine that the books are ordered by author, or perhaps by subject. We use an **index** to provide us with a method of accessing the books in other than their physical organization. The key used to determine the **virtual order** of the books is known as a secondary key. In our example, the library of congress number is the primary key, while author and subject would be the secondary keys.

Physically rearranging the records is known as **sorting** and, for large files, is quite time consuming. Creating the indexes by which we wish to **search** our database is known as

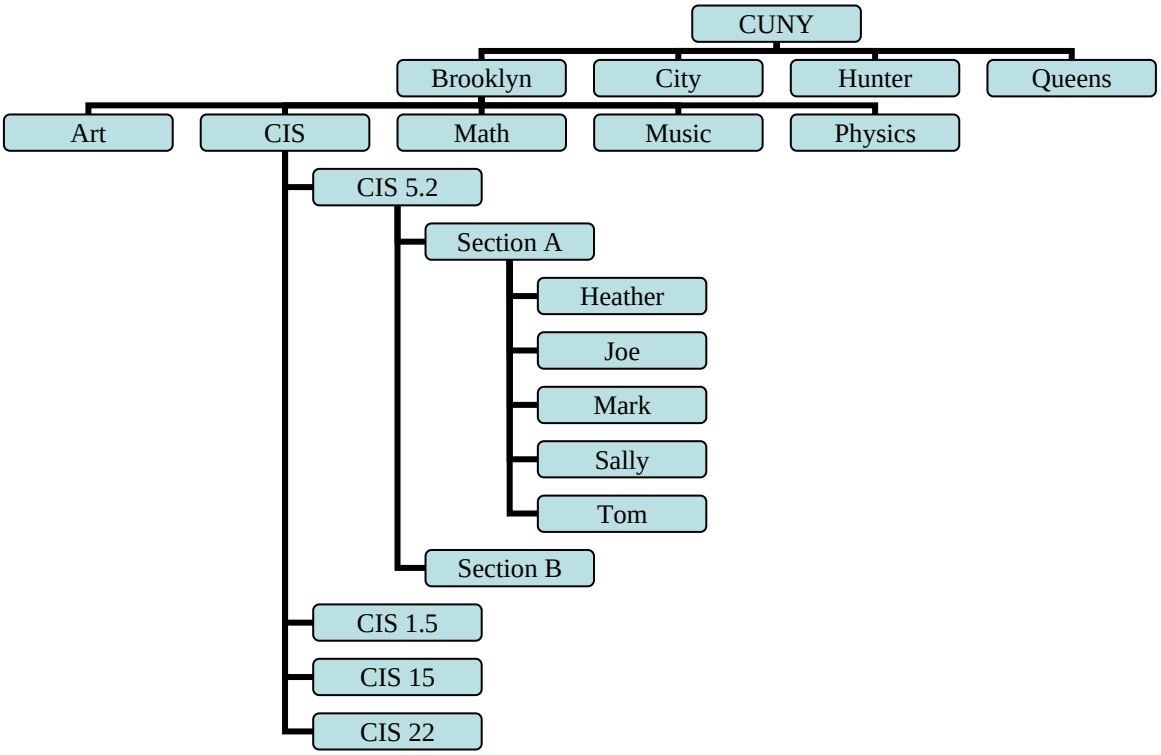
indexing. One of the important functions of a database management system is the creation of one or more indexes; providing us with the flexibility of searching our database by multiple criteria.

II. Database models[1]

A database model defines how the items in the database are related to each other. Several different methods for the organization of the information in the database have been defined. Each of these models has both advantages and disadvantages which are beyond the scope of this class. We will briefly review three of these methods.

a. Hierarchical model

In a hierarchical data model, data is organized into a tree-like structure. The structure allows repeating information using parent/child relationships: each parent can have many children but each child only has one parent.

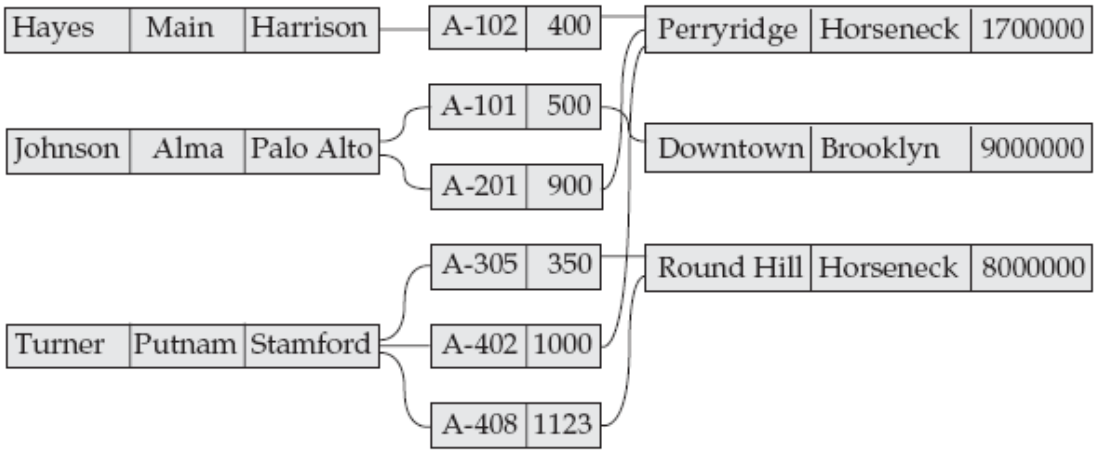


[1] Wikipedia

b. Network Model

The network model is a database model conceived as a flexible way of representing objects and their relationships. The network model allows each record to have multiple parent and child records, forming a lattice structure.

In the following example the diagram represents the relationship of account holders (name, street, and city), bank accounts (account number, balance) and bank branch (branch name, branch city, assets). Note that an account holder may own one or more accounts at one or more branches.



c. Relational Model

The basic data structure of the relational model can be thought of as a table, where information about a particular entity (say, a student) is represented in columns and rows. Each column represents a field and each row represents a record. Multiple tables (files) are related to each other by keys which they share in common. Any value occurring in two different records belonging to different tables implies a relationship among those two records.

The relational model may also be viewed as a mathematical construct that defines the form of the data within it and which must conform to some basic rules. These rules enforce **database integrity** (a method of ensuring that changes made to the database by authorized users do not result in a loss of data consistency) and define the operations that may be performed on it. A technique called **database normalization** may be used to optimally organize the tables in the database. These techniques lie outside the scope of this course and are covered in detail in CIS 45.

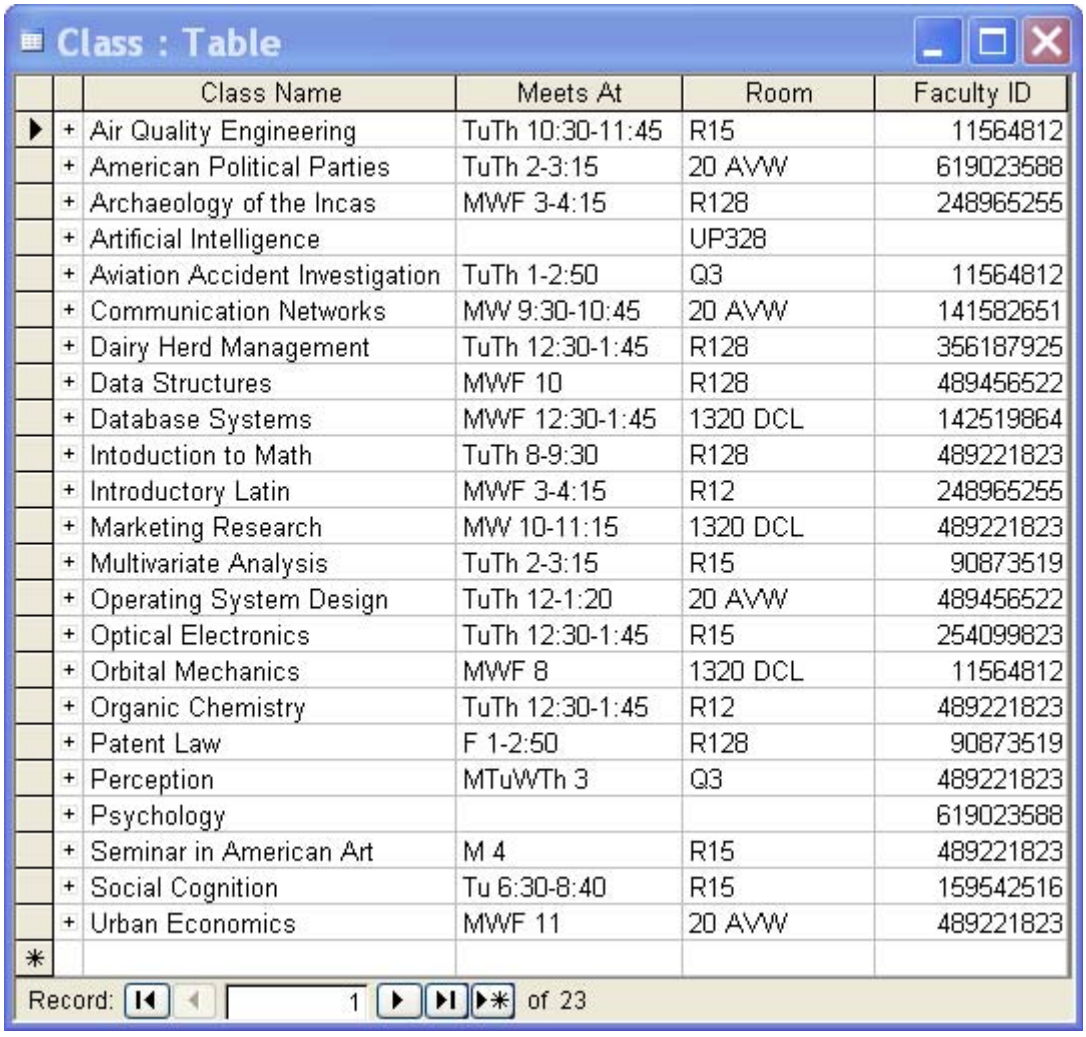
In the following example I will illustrate the relational model for a University Scheduling database that has been implemented using Microsoft Access.

III. Microsoft Access

Microsoft Access is a *Relational Database Management System* or **RDBMS** that provides a user interface and interactive design capabilities (including wizards) to help you track and report information with ease. It includes many prebuilt applications that you can modify or adapt to changing business needs. You enter the information through forms and create and edit detailed reports that display sorted, filtered, and grouped information in a way that helps you make sense of the data for informed decision-making.

The University Database defines four tables:

a. **Class** – which contains scheduling information for the classes offered at the university

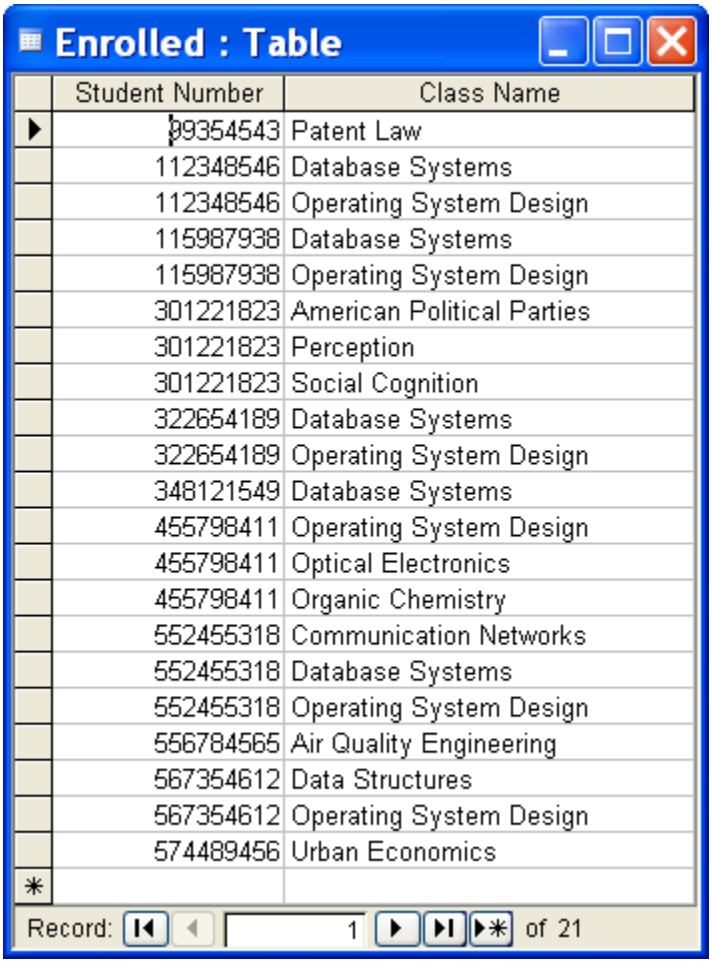


Class : Table

Class Name	Meets At	Room	Faculty ID
Air Quality Engineering	TuTh 10:30-11:45	R15	11564812
American Political Parties	TuTh 2-3:15	20 AVW	619023588
Archaeology of the Incas	MWF 3-4:15	R128	248965255
Artificial Intelligence		UP328	
Aviation Accident Investigation	TuTh 1-2:50	Q3	11564812
Communication Networks	MW 9:30-10:45	20 AVW	141582651
Dairy Herd Management	TuTh 12:30-1:45	R128	356187925
Data Structures	MWF 10	R128	489456522
Database Systems	MWF 12:30-1:45	1320 DCL	142519864
Intoduction to Math	TuTh 8-9:30	R128	489221823
Introductory Latin	MWF 3-4:15	R12	248965255
Marketing Research	MW 10-11:15	1320 DCL	489221823
Multivariate Analysis	TuTh 2-3:15	R15	90873519
Operating System Design	TuTh 12-1:20	20 AVW	489456522
Optical Electronics	TuTh 12:30-1:45	R15	254099823
Orbital Mechanics	MWF 8	1320 DCL	11564812
Organic Chemistry	TuTh 12:30-1:45	R12	489221823
Patent Law	F 1-2:50	R128	90873519
Perception	MTuWTh 3	Q3	489221823
Psychology			619023588
Seminar in American Art	M 4	R15	489221823
Social Cognition	Tu 6:30-8:40	R15	159542516
Urban Economics	MWF 11	20 AVW	489221823

Record: 1 of 23

b. **Enrolled** – which contains the students who are enrolled in each class

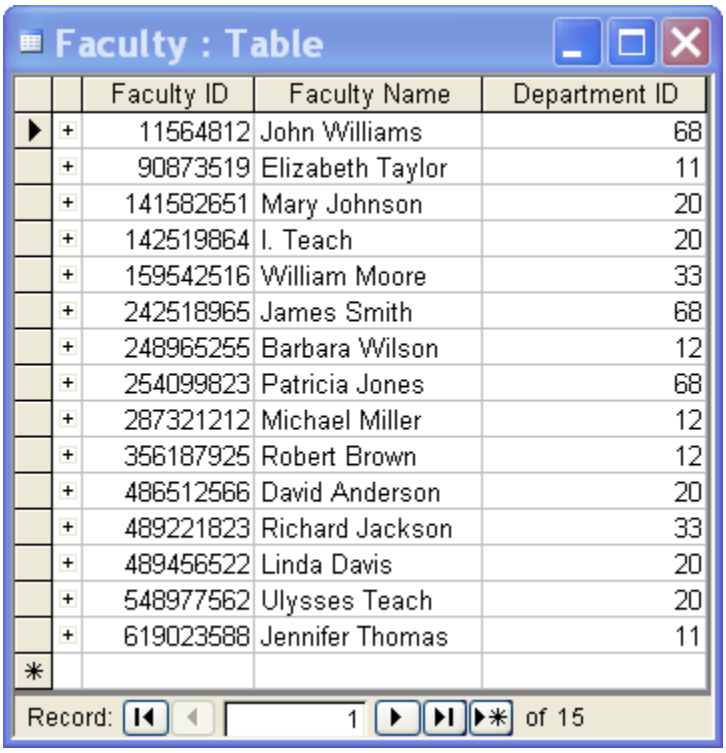


Enrolled : Table

Student Number	Class Name
99354543	Patent Law
112348546	Database Systems
112348546	Operating System Design
115987938	Database Systems
115987938	Operating System Design
301221823	American Political Parties
301221823	Perception
301221823	Social Cognition
322654189	Database Systems
322654189	Operating System Design
348121549	Database Systems
455798411	Operating System Design
455798411	Optical Electronics
455798411	Organic Chemistry
552455318	Communication Networks
552455318	Database Systems
552455318	Operating System Design
556784565	Air Quality Engineering
567354612	Data Structures
567354612	Operating System Design
574489456	Urban Economics

Record: 1 of 21

c. **Faculty** – which contains information regarding the faculty of the university



Faculty : Table

Faculty ID	Faculty Name	Department ID
11564812	John Williams	68
90873519	Elizabeth Taylor	11
141582651	Mary Johnson	20
142519864	I. Teach	20
159542516	William Moore	33
242518965	James Smith	68
248965255	Barbara Wilson	12
254099823	Patricia Jones	68
287321212	Michael Miller	12
356187925	Robert Brown	12
486512566	David Anderson	20
489221823	Richard Jackson	33
489456522	Linda Davis	20
548977562	Ulysses Teach	20
619023588	Jennifer Thomas	11

Record: 1 of 15

d. **Student** – which contains information regarding the students enrolled in the university

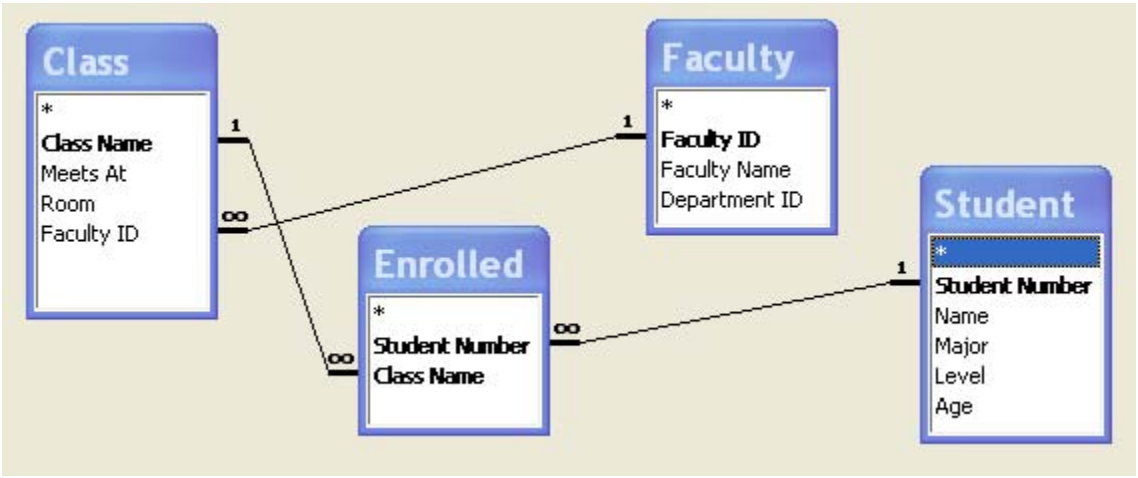
Student : Table

Student Nur	Name	Major	Level	Age
51135593	Maria White	English	SR	21
60839453	Charles Harris	Architecture	SR	22
99354543	Susan Martin	Law	JR	20
112348546	Joseph Thompson	Computer Science	SO	19
115987938	Christopher Garcia	Computer Science	JR	20
132977562	Angela Martinez	History	SR	20
269734834	Thomas Robinson	Psychology	SO	18
280158572	Margaret Clark	Animal Science	FR	18
301221823	Juan Rodriguez	Psychology	JR	20
318548912	Dorthy Lewis	Finance	FR	18
320874981	Daniel Lee	Electrical Engineering	FR	17
322654189	Lisa Walker	Computer Science	SO	17
348121549	Paul Hall	Computer Science	JR	18
351565322	Nancy Allen	Accounting	JR	19
451519864	Mark Young	Finance	FR	18
455798411	Luis Hernandez	Electrical Engineering	FR	17
462156489	Donald King	Mechanical Engineering	SO	19
550156548	George Wright	Education	SR	21
552455318	Ana Lopez	Computer Engineering	SR	19
556784565	Kenneth Hill	Civil Engineering	SR	21
567354612	Karen Scott	Computer Engineering	FR	18
573284895	Steven Green	Kinesiology	SO	19
574489456	Betty Adams	Economics	JR	20
578875478	Edward Baker	Veterinary Medicine	SR	21

Record: 1 of 24

Note that the columns in each table represent the fields and the rows represent the records.

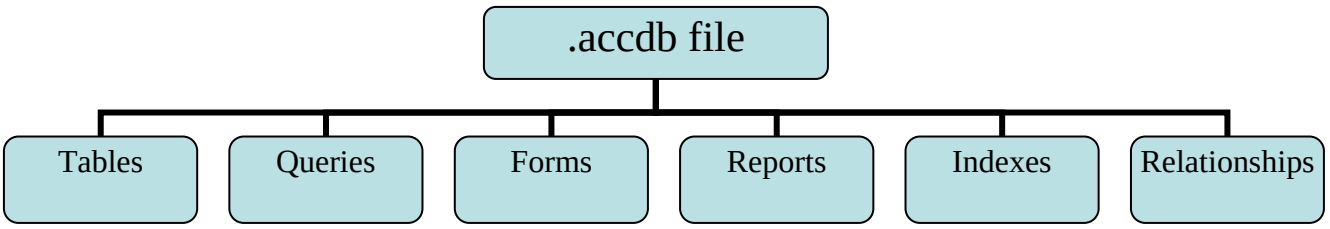
The tables are related to each other using the key fields as shown below. These relationships may be **one-to-one**, **one-to-many**, **many-to-one**, or **many to many**. In the diagram below, the '**1**—— ∞' represents a one-to-many relationship, while the '∞ —— **1**' represents a many-to-one relationship. Make sure you understand these relationships.



Using the techniques described in our textbook you will learn how to define tables and the relationships that they describe. Once the tables have been created, you will then learn how to enter data, and produce simple reports. Advanced techniques that we will cover include the creation of **forms** for the input of data, **queries** which will enable us to extract data according to specific criteria and, the design of elegant **reports** (including mailing labels and envelopes) by which these results may be presented.

At this point, it is useful to distinguish between the **physical organization** of the database and its **virtual organization** or view by which the user experiences its organization. The designer of the database is aware of its physical organization (i.e. the tables and the relationships between them). The user of that database however, does not need to understand the database's underlying organization, Rather, the user needs to be retrieve the information he or she requires as simply as possible. Different users have different needs. Security and privacy concerns may require that some of the data be hidden from unauthorized users. Using a saved query it is possible to extract information from one or more tables and to present the results in one or more a virtual tables. Each query then provides the user with a different 'view' of the database that is appropriate for its user.

All the information in the database is stored in a single Microsoft Access file having the .accdb extension. This file gathers up in one location the components of the database as shown below:


.accdb file
Tables
Queries
Forms
Reports
Indexes
Relationships