Plopping data into tables haphazardly is a guaranteed way to create consistency problems and limit what can be extracted from the data base. Good database design is a complex subject, but we can learn the basics.

A database is a set of tables. The tables are sets of records known as tuples composed of fields each having values from some primitive data type.

A table named Doctors composed of two tuples or records composed of seven fields, the first of which is an integer and the rest of which are text fields, some with specific formats.

Specifying A Table

Fields have names, which should be meaningful and there are facilities available for specifying the field types, format, etc.

Terminology

The structure of a data base is called a database schema. The schema specifies:

- The list of table names forming the database
- For each table, the fields of its records
- For each field, its attributes or properties, i.e. data type, key or not key, default value, etc.

A database as the word is normally used, i.e. tables with specific contents, is known as a database instance (of a data base schema).

There can be many instances of a single database schema.

Designing A Database Schema

Suppose a college wants a database of their students, faculty, courses taught, student transcripts, and so forth, what things should go into a design and how should it be organized?

- Students: first name, last name, home address, transcript …
- Faculty: first name, last name, SS#, home address, rank …
- Courses: class name, number, students attending, grades …

Deciding on the schema is called “database design” and it takes a little study to do right … but it’s easy to see the principles in action.

Avoid redundancy!
More Of The S&C DB Schema

The Design windows give the remaining structural information for the data base schema ... notice how Classes, Students etc have unique IDs.

The Tables For The University

Avoid redundancy by keeping like information together in a table and linking to it when it is needed elsewhere.

Bad design ... in the Classes table, do not include instructor's name, email, etc ... rather link to the instructor information by ID.

The User's View

A database system gives users a view of the DB that is meaningful to them, but may be synthesized from tables actually forming the database.

Consider A Student's List Of Classes

Though the admin information at the top comes from the Classes table, the class list at the bottom is not stored explicitly in any table. It is synthesized.

Synthesizing The Class List

❖ One table -- Students And Classes -- contains records that associate students with classes
❖ By listing all records with FIT100's ClassID, a table is created of the students in FIT100 by StudentID
❖ By looking up each student using StudentID, the other fields of the class list can be located.

Why Use This Schema?

❖ Associating a student with a class is the logical idea behind registering for a class, so Students & Classes corresponds to a real phenomenon -- a plus
❖ Having classes listed in the student record violates the goal of a fixed length record, and makes it cumbersome to create a class list -- minus
❖ Having students listed in the class record violates the fixed length record goal, and makes it cumbersome to create a registration list for each student -- minus
❖ “Registering students” -- what STAR does -- can be done without touching either Students or Classes tables -- a plus
Not All Views Are Synthesized

- Many tables will be of interest on their own, too