Introduction to Databases

In this section

What is a database

A database is a collection of data in a useful format. Databases allow you to store data and retrieve it or modify it when you need to, easily and efficiently, regardless of the amount of data being manipulated.

Traditionally, databases ran on large, powerful mainframes for business applications. However, with the advent of small, powerful personal computers, databases have become more readily usable by the average computer user. Microsoft Access is a popular PC-based database engine.

It is estimated that 80% of all applications are connected to some kind of database.

Why are databases so important?

Databases have quickly become integral to the design and development of web sites and to the services they offer.

Consider a site like Amazon.com that must be able to allow users to quickly jump through a vast virtual warehouse of books and compact disks.

How could Amazon.com create web pages for every single item in their inventory and how could they keep all those pages up to date? Well the answer is that their web pages are created on the "fly" by a program that "queries" a database of inventory items and produces an HTML page based on the results of that query.

Relational Databases

At the core of the relational model is the concept of a ***table*** (also called ***entity***) in which all data is stored. Each table is made up of ***records*** (horizontal rows) and ***fields*** (vertical columns also known as ***attributes***). A relational database contains more than one table that are related to each other to prevent duplication of data or fields.

Table 1

tblStudent				
Student ID	Name	Address	Phone	Email
123456	Tom Smith	45 Walker Street Sydney 2000	147212365	tom@email.com
789654	Susan	123 Pacific HWY Hornsby 2077	78964521	susan@email.com

The **table** or **entity** above represents the "tblStudent" table.

Student ID, Name, Address, Phone and Email are called **fields** or **attributes**. They help describe the student.

The row "123456 Tom Smith 45 Walker Street Sydney 2000 147212365 tom@email.com" represents a student and is called a **row** or a **record**.

The same information may appear in my address book. If I want to look up Tom Smith's address I would most likely look under "S" in my address book and scan the "S" page until I found his name. Once I found his name, I would take note of his address.

This is fine if you don't have too many friends. Imagine TAFE's student records book if it had to keep track of all the students' details on paper. Imagine backing up paper!

Database Design

Analysis and design is the most important step in creating a new application. Mistakes made in the analysis and design phase are very time consuming later on in the development phase and are therefore costly.

The more time spent in the analysis and design phase directly increases the effectiveness of the resulting application.

Data Modelling

Most people follow some kind of methodology when they are involved in designing a new application. A methodology is a set of processes through which a developer analyses the client's requirements and develops the application. In other words it is the procedure they follow when creating new systems.

One technique commonly used is Data Modelling. The purpose of data modelling is to create an accurate representation (in the form of diagrams) of the application.

Parts that make up a database

Entities (Tables)

The first step in creating the model is to identify the entities (tables). These are usually subjects of interest e.g. Student, Teacher, Subject, Timetable

Tables are represented by a rectangle:

tblStudent

Attributes (Fields)

An attribute describes the entity (table). A student has a name; name is an attribute of the entity student. An attribute is any detail that serves to identify or describe the entity.

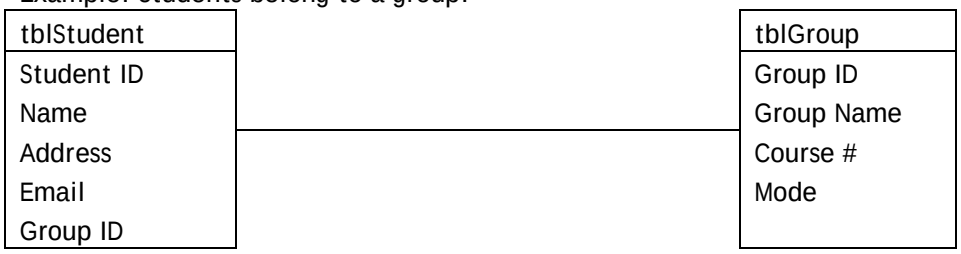
Attributes are represented as follows:

tblStudent
Student ID
Name
Address
Email

Relationships

A relationship is any association, linkage, or connection between entities.

Example: Students belong to a group.



Types of relationships

One to one – student and student card (students have one student card, student cards belong to one student)

One to many – student and groups (many students belong to the web group)

Many to many – student and teacher (students are taught by many teachers, teachers teach many students)

Primary Keys

A ***primary key*** is a field or group of fields, which ***uniquely*** identifies a record.

In our "student" example, it is the Student ID. The student name could have been used but there is no guarantee that you will never have 2 or more students with exactly the same name.

Rule: Every table must have a primary key.

Good primary keys

Numbers (must be unique - for example, you could not use street number as a primary key because 2 or more people will have the same street number)

Small strings, for example the room number G109B, can be used to uniquely identify a classroom. A car registration plate can be used to identify the car. Primary keys should be small as the database engine used to find the record you are looking for uses them. The bigger they are, the slower the database lookups.

Bad primary keys

Long strings – slows down the database

Money – e.g. the price of an item because 2 items can have the same price

Date – e.g. order date because 2 orders can have the same order date

Names – again because of duplication

Foreign keys

A foreign key is a primary key from one table embedded in another table. For example, the group name should appear in the student table so that we can tell in which group each student belongs.

Databases

There are many SQL databases around.

Oracle (www.oracle.com)

Informix (www.informix.com)

Sybase (www.sybase.com)

MySQL (www.mysql.com)

Postgres (www.ca.postgresql.org/index.html)

Access

The only way to become good at database design is to practise. It is a skill that you need to learn by doing not simply reading. The first step in any database or application design is to understand the problem.

What makes a good model?

- Each table must have a primary key
- Fields in a table must describe that table. If they don't, then they don't belong in that table. E.g. order number does not belong in the customer table it belongs in the order table.
- Relationships should be one to many or many to one.
  E.g. (one to one relationship) If we have a table called orders and a table called order dates. The relationship between these 2 tables is one to one (one order has one order date). The order date should actually belong to the order table as a field and not as a separate table, because order date helps describe the orders table. The only time we should have a one to one relationship is when additional information is stored and usually has a different security level (ie very private information).
  Many to many relationships are not good because they cannot be easily converted to database tables and usually represent bad database design.

Quick steps to design a database

Step 1

Understand the problem. This is the most important step when designing any application. If the problem is not given to you in writing, write it down.

Step 2

Pick out the nouns (subjects). These nouns will become either the tables or fields.

Step3

Decide which nouns should be tables and which should be fields. A field belongs to a table if it helps describe the table.

Step 4

Draw the tables, listing the fields.

Step 5

Do the fields help describe the tables they are in? If not maybe, they belong in another table.

Step 6

Decide on the relationships between the tables and draw them in.

Step 7

Look at one to one relationships and try to eliminate them by combining them in the one table.

Step 8

Look at many to many relationships and try to eliminate them by creating new tables.

Step 9

Identify the primary keys for each table.

Step 10

Identify the foreign keys.

Step 11

Put in some test data and check that your database works.

The only way to master this is to do many and varied exercises. Remember, more than one answer can be correct.

Activity 1. (design and create an ER diagram)

1. Keep a record of your friend's addresses and phone numbers. You need to record their first name, last name, contact numbers (some friends may only have 1 contact number but others can have several), home address, work address, and email address.

2. Design a database that keeps track of artists and their CD's. You need to keep a record of the artist's name, whether they are a group, solo female, solo male. For every CD released by that artist you need to keep track of the CD title, duration in minutes, number of tracks, price, date released, music style.
   You do not need to worry about duets. Only one artist or group appears on a particular CD.

3. A zoo wants to record the details of the animals and their diet on their computer system. You have been asked to design a database using the following information:

   Animal name

   Keeper name

   Pager number

   Animal cage number

   Fruit quantity

   Meat quantity

   Cereal quantity

   Assume one diet can be used for more than one animal. An animal has one diet only. An animal is looked after by one keeper only. However one keeper can look after several animals.

4. Below is an example of a form that is completed by all staff that work for Advance Australia Company. This form is used whenever an employee has attended a computer course.

   ID Number:_____
   Surname:_____
   First Name:_____
   Department:_____

   Comp. Course No:_____
   Computer Course:_____
   Date Attended:_____
   Cost of Course:_____

   You have been assigned to design a database that will accommodate the information collected from the above form.

5. You have been asked to design a database for a supplier of statinery products called Paper Pusher Merchants, or PPM for short.

   A general outline of the business is as follows. PPM has a list of customers. PPM also keeps a list of stock items with their description, price and tax rate. The customers place orders, and a single order will normally consist of several items.

6. A suburban dental practice wants to computerise its record. At present each customer card holds the following information:

Customer Name: Samuels Jennifer
Address: 73 Parade St. Wahroonga
Phone: 9983 3333
Health Fund: MBF
Health Fund Number: 934123

Date	Procedure	Cost	Dentist
14/5/08	X-Ray - bottom	$95	Kadinsky Samuel
21/5/08	Extraction - wisdom	$200	Millard Robert
28/5/08	Fluoride	$50	Jones Anne

The dental practice would like the electronic database to hold more information such as:

personal details of each dentist (address, phone, etc)

more details of various medical funds (address, phone).