

# Databases

Alan M. Durham

Computer Science Department  
University of São Paulo  
alan@ime.usp.br

# Questions

- What is a database management system (DBMS)?
- What is a database?
- Why use a DBMS?

# Databases vs. Flatfiles - Flatfiles

- in flat files you have independent files with the information you need
- each line of the file will contain the items you need for each entry
- ex: thesis bibliography:
  - \_ Author's name
  - \_ Paper title/Book title
  - \_ Volume+Issue
  - \_ Year
  - \_ Publisher

# Databases vs. Flatfiles - Flatfiles

- searching:
  - \_ just browse lines and check if information is there
- formatting
  - \_ how to separate fields? (can use special character)
  - \_ how do we handle multiple fields? (multiple authors)
  - \_ can be solved, fancier programming
- redundancy problems
  - \_ what if we misspelled “Anguraj Sadanadam”, I mean, “Anguraj Sadanandam”?
- extending
  - \_ we would like to be able to search all articles on “building pipelines”

# How do DBs solve this?

- Databases are collections of *tables* that describe our information and the relationships
- With tables to describe relationships we are able to avoid redundancy, and therefore inconsistency

# Example: tables for data

- Table 1: Author's table
  - \_ name, email address, institution
- Table 2: Publisher's table
  - \_ publisher's name
  - \_ publisher's address
- Table 3: Journal table
  - \_ Journal Name
  - \_ Publisher
  - \_ ?

# Example: tables for data AND relationships

- Table 4: Article Table
  - \_ title
  - \_ journal
  - \_ year
  - \_ volume
  - \_ Location
  - \_ PDF?
- Article author field? NO!
- Table 5: Authorship
  - \_ author vs. article
  - \_ one entry per relation
  - \_ do not use names, but indexes on tables 1 and 3

# Relationship Tables

- handle multiplicity well
  - \_ many entries, one for each author/article pair
- can eliminate redundancy
  - \_ only indexes in tables where original information is: author name only typed once

# Relational Model Example

SSN	Researcher_Name	Home Institution
12345679	Alan	Univ.Sao Paulo
98765432	Dinesh	ICGEB
24680123	Chuong	NIH

Id	Title
1	Going Home
2	Having Lunch
3	Staying Awake

Auth.SSN	Article_Id
98765432	1
24680123	2
12345679	1
98765432	3

Who has published more?

# Database terms

(adapted from [www.geekgirls.com](http://www.geekgirls.com))

- **database:**
  - \_ A collection of related information stored in a structured format. Database is often used interchangeably with the term table, but databases generally consist of many tables
- **DBMS:**
  - \_ A program which lets you manage information in databases: Oracle, Access, MySQL
- **data entry:**
  - \_ The process of getting information into a database,

# DataBase terms

- **field:**
  - \_ Fields describe a single aspect of each member of a table.
- **flat file:**
  - \_ A database that consists of a single table.
- **index:**
  - \_ A summary table which lets you quickly locate a particular record or group of records in a table.

# Relational Model Example

SSN	Researcher_Name	Home Institution
12345679	Alan	Univ.Sao Paulo
98765432	Dinesh	ICGEB
24680123	Chuong	NIH

Id	Title
1	Going Home
2	Having Lunch
3	Staying Awake

Auth.SSN	Article_Id
98765432	1
24680123	2
12345679	1
98765432	3

Who has published more?

# Database terms

- **key field** :
  - \_ You can sort and quickly retrieve information from a database by choosing one or more fields to act as keys. This is basically an implementation issue for DataBases, indicating to the DBMS witch are the fields you are generally going to use in a search (e.g. Author Name, as opposed to email)
- **primary key**:
  - \_ A field that uniquely identifies a record in a table. Also implementation issue. This is the field used in relationship tables.

# Relational Model Example

SSN	Researcher_Name	Home Institution
12345679	Alan	Univ.Sao Paulo
98765432	Dinesh	ICGEB
24680123	Chuong	NIH

Id	Title
1	Going Home
2	Having Lunch
3	Staying Awake

Auth.SSN	Article_Id
98765432	1
24680123	2
12345679	1
98765432	3

Who has published more?

# Database terms

- **record:**
  - \_ table entry
- **relational database:**
  - \_ A database consisting of more than one table. We describe these databases using an *entity-relationship schema*
- **table:**
  - \_ A single store of related information. A table consists of records, and each record is made up of a number of fields. Database management system.

# Relational Model Example

SSN	Researcher_Name	Email address
12345679	Alan	alan@usp.br
98765432	Dinesh	dinesh@icgeb.in
24680123	Chuong	chuong@nih.gov

Id	Title
1	Going Home
2	Having Lunch
3	Staying Awake

Auth.SSN	Article_Id
98765432	1
24680123	2
12345679	1
98765432	3

Who has published more?

# What does a DBMS gives me?

- protection (I don't want everyone to access to my data)
- consistency: transactions (ATM machines)
- queries: we can ask questions

# Extending the Database (won't fit one slide)

- Article table:
  - \_ id
  - \_ journal
  - \_ year
  - \_ rating (very important, relevant, accessory)
  - \_ subject

# What types of questions DBs answer?

- which are the papers I have that were written by J. Kissinger, and published after 2001 (add year)
- any articles authored both by Kissinger and Gupta that appeared in Nature after 2000? (add journal)
- What are the email addresses of all researchers in papers about promoter detection? (add subject)
- Do I have a paper on perl scripting?”
- If so, on which folders are they located?” ,
- Show me the journals where the papers rated ‘very important’ to my thesis are.””

# Queries depend on DB structure

- The types of queries depend on how you build your tables
- efficient searches need the field to be a key
- to relate information we need it to be in the same record or to have a relationship table
- you need good design!!!!!!

# How do I query a database

- pre-prepared queries
  - \_ someone writes a program that does specific queries for you
- query language: SQL (need to know structure of database)
- example query: “I want to know authors and years of Nature articles after 2000
- Query steps
  - 1.get entries of tables with particular values of keys
    - \_ get articles from Nature published after 2000
  - 2.select fields of records
    - \_ author index and year from article
  - 3.join tables or “virtual tables” (result of queries)
    - \_ join articles with specific author index with author table

# When do you use a DBMS?

- Need to share data
- big quantity of data
- interconnected data of different kinds
  - \_ images, short text, long text
- data mining
  - \_ by hand
  - \_ automatic

# Advantages of Using a DBMS

- Controlling redundancy
- Restricted Unauthorized Access
- Providing persistent storage for program data structure
- Providing multiple user interfaces
- Representing Complex Relationships Among Data
- Enforcing Integrity Constraints
- Providing backup and recovery

# When Not to Use a DBMS

- Due to unnecessary overhead cost
- Not able to model
  - \_ Do not understand yet the nature of data
  - \_ Do not understand yet nature of problems
  - \_ No modeling expertise

# Why use Models

- Databases are models for the reality
- Models are useful to represent the real world; easier to manage than the real world
- The cost of working with a model is considerably less than experimenting with a real world system.

# Where next?

- try this internet tutorial (introduces db and shows how to build a flatfile db):
  - \_ [http://www.geekgirls.com/menu\\_geekgirlguides.htm](http://www.geekgirls.com/menu_geekgirlguides.htm)
- db course
  - \_ designing databases is HARD
  - \_ should not be done by novices
- many books for novices
  - \_ will hardly make you an expert
  - \_ try “SQL for Dummies”, Allen G. Taylor
- databases to use ? (examples, not recommendations)
  - \_ large scale: Oracle, Sybase
  - \_ medium scale: mySql, Postgre
  - \_ small scale: Access, etc.