CS 450
Database Concepts

Lecture One - Introduction

Instructor: Dr. Jessica Lin

Basics

Instructor: Dr. Jessica Lin

Contact Info:

Email: jessica@cs.gmu.edu
Homepage: http://www.cs.gmu.edu/~jessica
Office: Engineering Building Room 4419
Phone: (703)993-4693
Office Hours: Wednesday/Thursday 2-3pm

Class Meeting: **Wednesday 4:30-7:10pm Enterprise Hall 276**

Pre-requisites: C or better in CS 310 (Data Structures) and CS 330 (Formal Models and Methods)

TA: Yuan Li (ylif@gmu.edu)

Outline

- Course syllabus
- Introduction to DB & DBMS
- Introduction to ER (Entity-Relationship) model

Administration Trivia

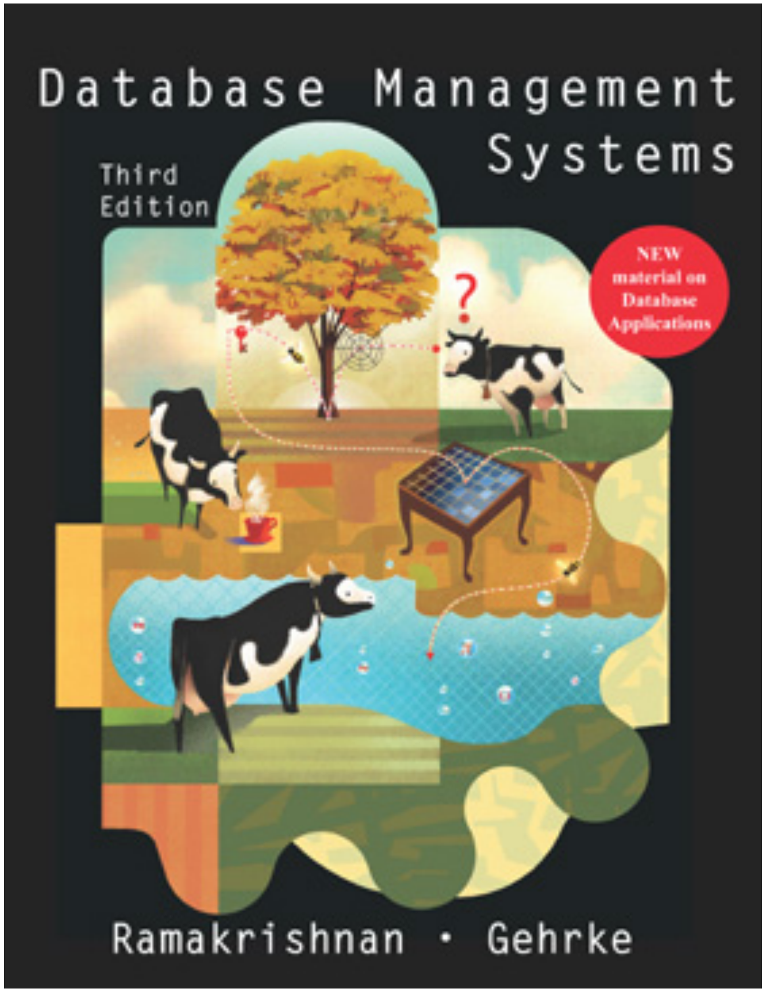
- Class webpage: http://www.cs.gmu.edu/~jessica/cs450_s12.html
- In most cases (except today), I will put the slides online the night before the lecture.
- I recommend that you print out the slides before attending lecture.
- You are 100% responsible for any announcements and updates on the class webpage, so visit the page frequently.
- You may bring your laptops to take notes, but if you use it for anything else, I may ask you to leave the class.

Textbook

- Required Book:
  - Database Management Systems, 3rd ed. by Raghu Ramakrishnan & Johannes Gehrke, McGraw-Hill.
- Recommended Book:
  - Oracle 10g Programming: a Primer, by Rajshkhar Sunderraman, Addison Wesley, ISBN 0321463048

www.cs.wisc.edu/~dbbook

3rd Edition


Database Management Systems
Third Edition
NEW material on Database Applications
Ramakrishnan • Gehrke

Grading

- Midterm Exam: 25%
- Final Exam: 35%
- Homework Assignments: 20%
- Project(s): 20%
- Quizzes (extra credit): up to 3%

- You may earn up to 3% extra credit from the pop quizzes (given at the beginning of class). You cannot make up missed quizzes.

- Homework is due on my desk in the first 5 minutes of the class on the date in question, unless specified otherwise. After 5 minutes, the homework is considered late, and I will take 10% off of the grade**. NO SUBMISSION WILL BE ACCEPTED ONCE I LEAVE THE CLASSROOM AT THE END OF THE CLASS.**

- If you cannot come to class the day an assignment is due, it's your responsibility to make sure that the assignment is delivered on-time (e.g. emailing it to the TA, or by making arrangement with me).

Homework/Project Submission

- https://mymasonportal.gmu.edu/webapps/portal/frameset.jsp
- Login with your GMU student account
- Click on the Courses tab on the upper right hand corner
- Choose CS450
- Use Blackboard for:
  - Electronic submission of assignments/projects
  - Checking grades
  - Getting course materials such as homework solutions

Honor Code System

- GMU honor Code http://academicintegrity.gmu.edu/honorcode/
- In addition, the CS Department has specific honor code policies for programming projects, etc.: http://cs.gmu.edu/wiki/pmwiki.php/HonorCode/HomePage
- For this class
  - Assignments are individual, unless specified otherwise. Group discussions are encouraged but final solution and write up must be individual.
  - Exams: individual effort, closed books

Contact Policy

- Contact the Oracle support group (oracle@ite.gmu.edu) if you
  - Encounter technical difficulties with Oracle
- Contact the TA if you
  - Have questions about the assignments/projects (including grading)
- Contact me if you
  - Have questions about the course materials
  - Have questions about the exams
  - Have general questions/concerns about the course

Email Policy

- Always use email as the **first** means to communicate with me! I strongly prefer that you only email me from your official GMU email. If you must email me from another account, you must state your full name and your official GMU email address.
- We will use Piazza, a free online class Q&A platform. You should have received an invitation to sign up.
- Think before posting the question (e.g. is the answer in the book or the lecture slides?). You are encouraged to answer each other's questions, but make sure you don't just give away the answers to the assignments/projects. This will be considered a form of **academic dishonesty**. If your questions reveal your solutions to the assignments, send them to me as private posts.

Re-grading Policy

- When you request that a certain problem be re-graded, unless it’s an obvious error (e.g. due to computation error), we will re-grade your entire homework/exam, not just the problem.
- The logic is that if we messed up on one problem, we could have messed up on more problems.
- So be very certain that you deserve the point(s) before you ask.

Class Schedule

- See class website for the tentative class schedule: http://www.cs.gmu.edu/~jessica/cs450_s12.html

Useful links for your computing needs

- http://labs.ite.gmu.edu/ for
  - Mason account information (FAQ -> "I need an account or password…")
  - ITE computing lab and Oracle DBMS information (Go to FAQ->Oracle).
- http://www.oracle.com/technology/software/index.html for personal edition of Oracle 10g.

Any Questions?

Introduction to Databases

What are Database and DBMS?

- Database:
  - A very large, integrated collection of data.
  - Models real-world *enterprise.*
    - Entities (e.g., students, courses)
    - Relationships (e.g., Frodo taking CS 450)
- A *Database Management System (DBMS)* is a software package designed to store, provide access and manage databases.

Examples of DBMS Usage

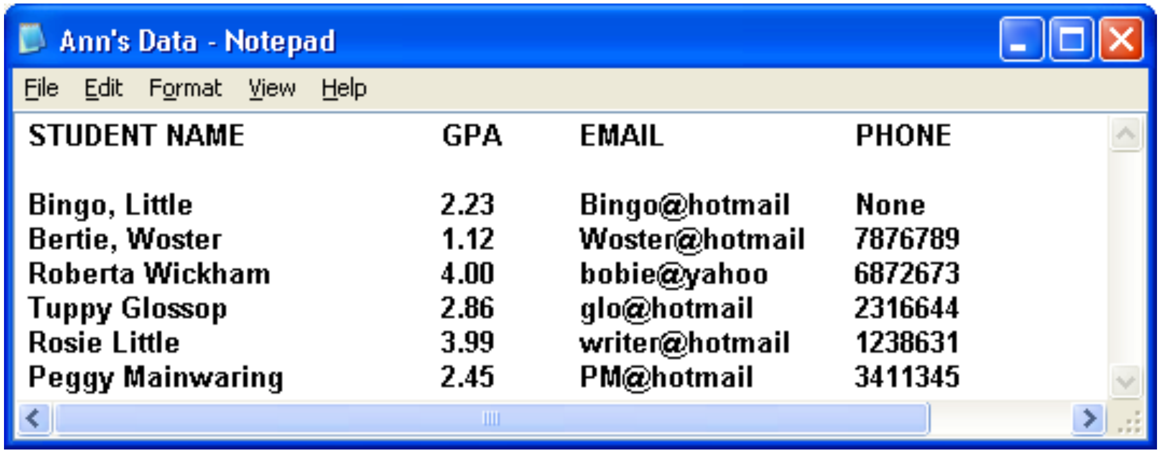
- Airlines: reservations and schedules (expedia.com)
- Universities: student info, grades
- Banking: customer info and accounts (bankofamerica.com)
- Credit Cards: customer info, transactions
- Sales: customer info, inventory (Amazon.com)
- Government: taxes, census

Why do we need database management systems?

- A **Database Management System** (DBMS) is a tool that allows us to store, modify, and query data.

However, I can store, modify and query data in a text file!

What can a DBMS do that I can't do with my text file solution?



STUDENT NAME	GPA	EMAIL	PHONE
Bingo, Little	2.23	Bingo@hotmail	None
Bertie, Woster	1.12	Woster@hotmail	7876789
Roberta Wickham	4.00	bobie@yahoo	6872673
Tuppy Glossop	2.86	glo@hotmail	2316644
Rosie Little	3.99	writer@hotmail	1238631
Peggy Mainwaring	2.45	PM@hotmail	3411345

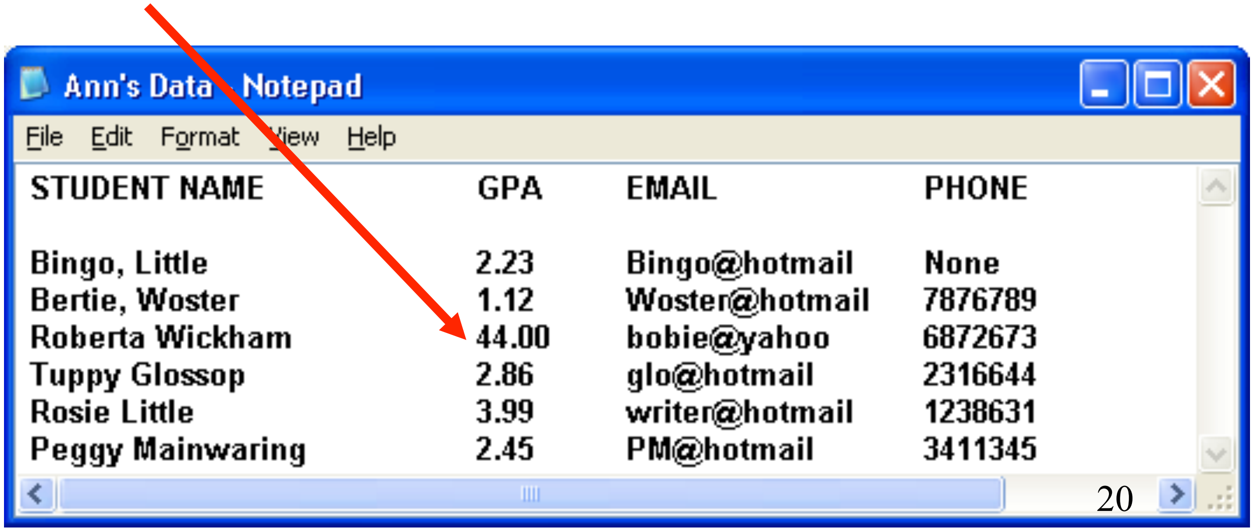
A simple solution to manage data:- stick them all in a text file!

Enforcing Constraints

• With the text file solution, there is no way to enforce integrity constraints on the data. In other words people can put bad data into the text file.

• In contrast, a DBMS allows us to enforce all kinds of constraints. This really helps (but does not guarantee) that our data is correct.

A typo gives Roberta Wickham a GPA of 44.00

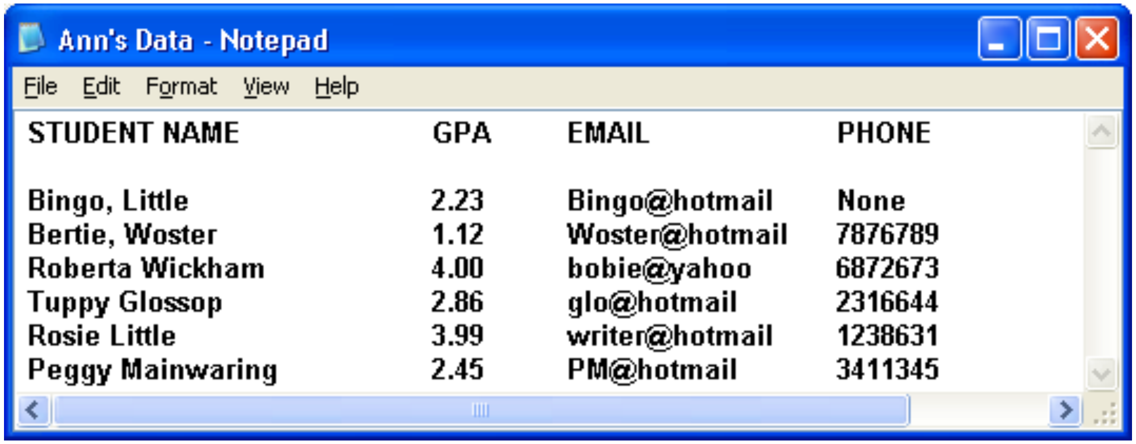


STUDENT NAME	GPA	EMAIL	PHONE
Bingo, Little	2.23	Bingo@hotmail	None
Bertie, Woster	1.12	Woster@hotmail	7876789
Roberta Wickham	44.00	bobie@yahoo	6872673
Tuppy Glossop	2.86	glo@hotmail	2316644
Rosie Little	3.99	writer@hotmail	1238631
Peggy Mainwaring	2.45	PM@hotmail	3411345

Scalability

- The text file solution might work for small datasets. What happens when we have big datasets?

- Most real world datasets are so large that we can only have a small fraction of them in main memory at any time, the rest has to stay on disk.



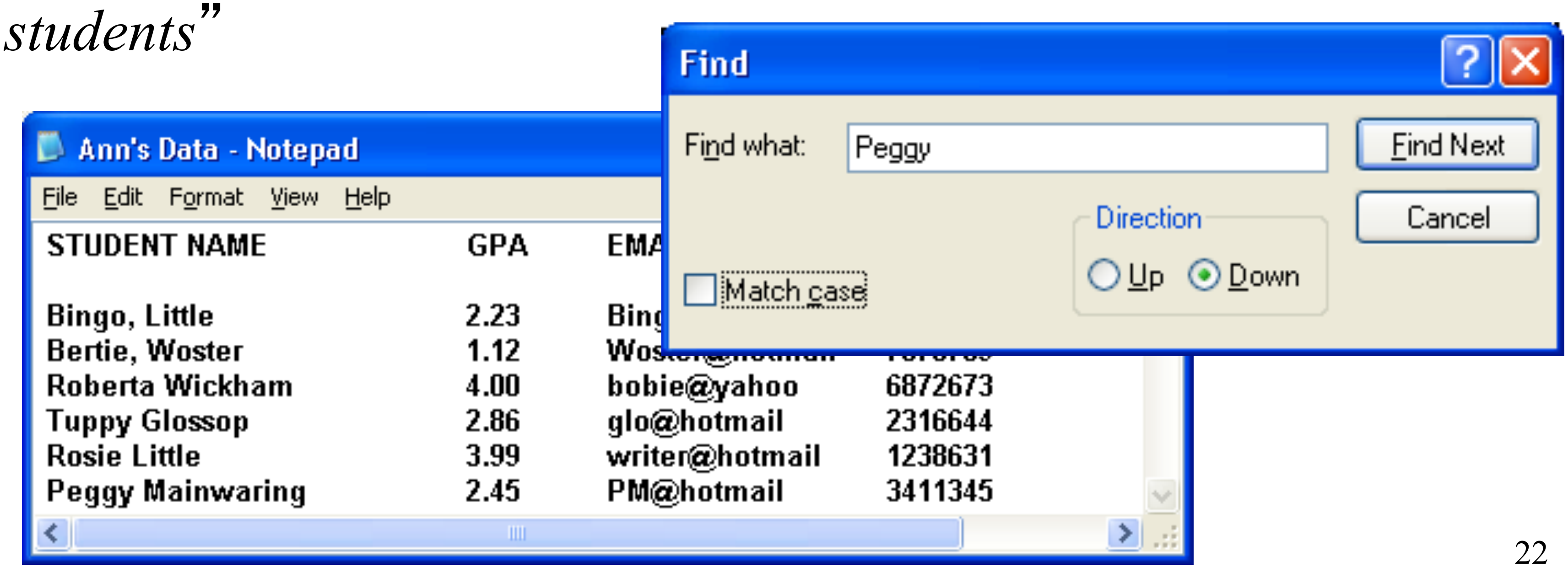
```
STUDENT NAME              GPA     EMAIL             PHONE

Bingo, Little             2.23    Bingo@hotmail     None
Bertie, Woster            1.12    Woster@hotmail    7876789
Roberta Wickham           4.00    bobie@yahoo       6872673
Tuppy Glossop             2.86    glo@hotmail       2316644
Rosie Little              3.99    writer@hotmail    1238631
Peggy Mainwaring          2.45    PM@hotmail        3411345
```

Query Expressiveness

- The text file solution would allow me to search for keywords or certain numbers (slowly).

- With a DBMS I can search with much more expressive queries. For example I can ask.. "*Find all students whose GPA is greater than 2.5, and who don't own a phone*" or "*what is the average GPA of the students*"



STUDENT NAME	GPA	EMA	
Bingo, Little	2.23	Bing	
Bertie, Woster	1.12	Wos	
Roberta Wickham	4.00	bobie@yahoo	6872673
Tuppy Glossop	2.86	glo@hotmail	2316644
Rosie Little	3.99	writer@hotmail	1238631
Peggy Mainwaring	2.45	PM@hotmail	3411345

Find what: Peggy

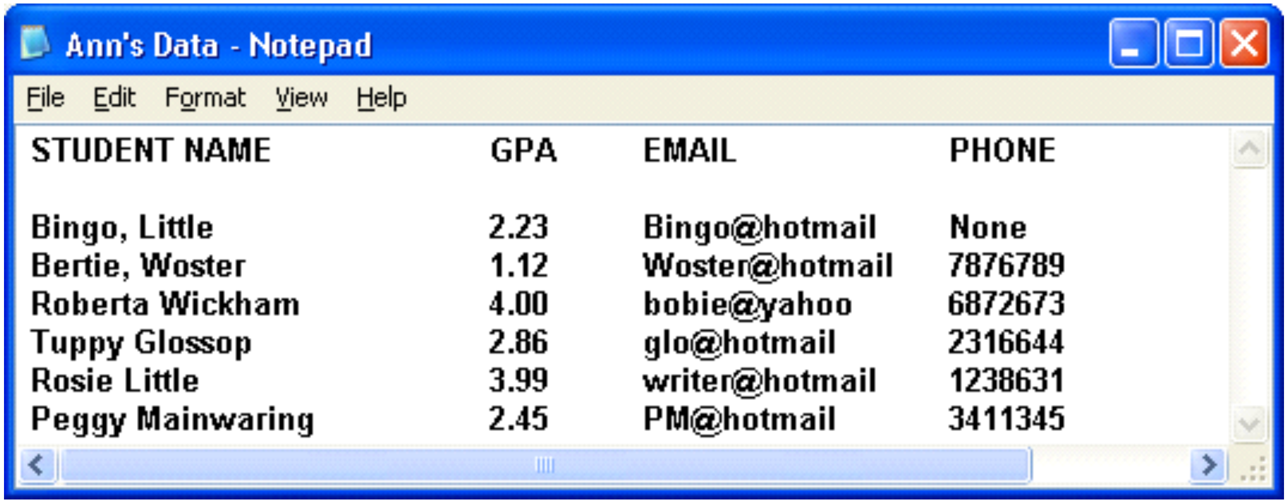
☐ Match case

Direction: ○ Up ● Down

Query Expressiveness II

- I could write some program that might allow more expressive queries on my text file, but it would be tied into the structure of my data and the operating system etc..

- With a DBMS we are completely isolated from the physical structure of our data. If we change the structure of our data (by adding a field, for example) or moving from a PC to a Mac, nothing changes at the front end!

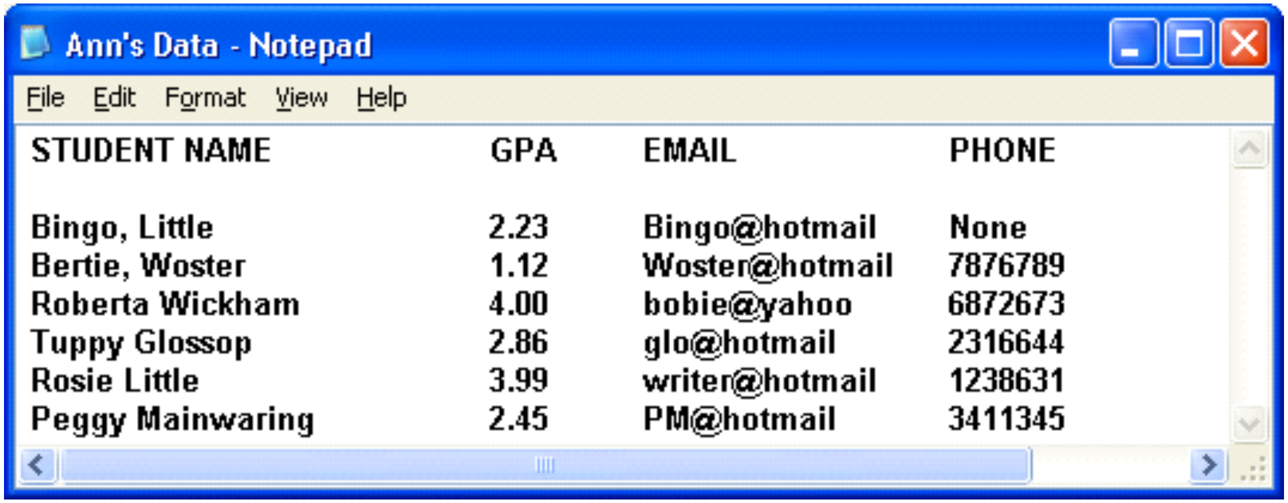


STUDENT NAME	GPA	EMAIL	PHONE
Bingo, Little	2.23	Bingo@hotmail	None
Bertie, Woster	1.12	Woster@hotmail	7876789
Roberta Wickham	4.00	bobie@yahoo	6872673
Tuppy Glossop	2.86	glo@hotmail	2316644
Rosie Little	3.99	writer@hotmail	1238631
Peggy Mainwaring	2.45	PM@hotmail	3411345

Different Views

- The text file solution only allows one view of the data.

- With a DBMS I can arrange for different people to have different views of the data. For example, I can see everything, a student can see only his/her data, the TA can see data for students in his/her section, etc.

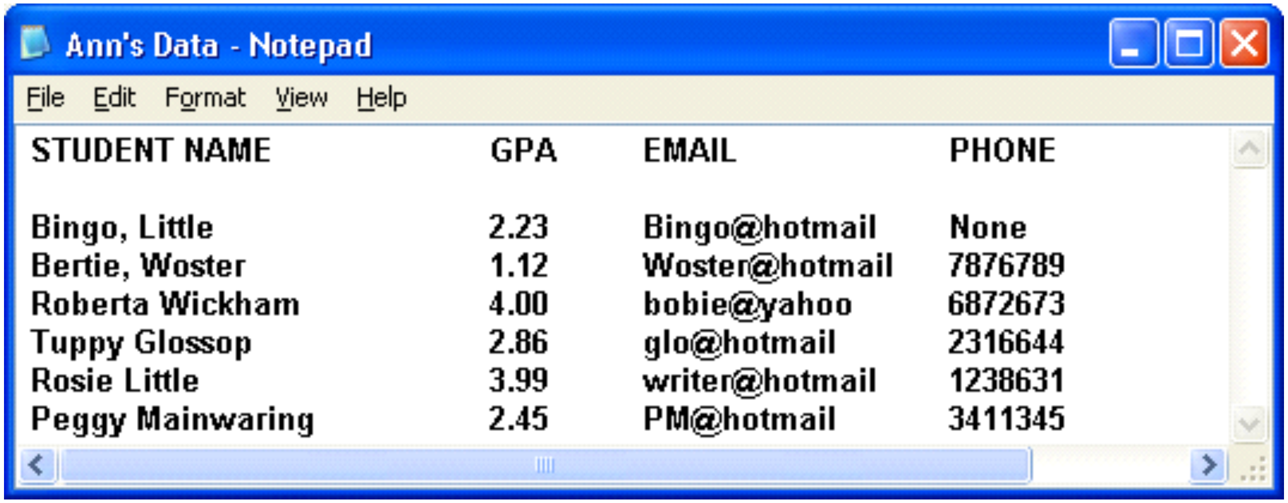


STUDENT NAME	GPA	EMAIL	PHONE
Bingo, Little	2.23	Bingo@hotmail	None
Bertie, Woster	1.12	Woster@hotmail	7876789
Roberta Wickham	4.00	bobie@yahoo	6872673
Tuppy Glossop	2.86	glo@hotmail	2316644
Rosie Little	3.99	writer@hotmail	1238631
Peggy Mainwaring	2.45	PM@hotmail	3411345

Concurrency

- Suppose I leave my text file on UNIX account, and I log in and begin to modify it at the same time my TA is modifying it!

- A DBMS will automatically make sure that this kind of thing cannot happen.

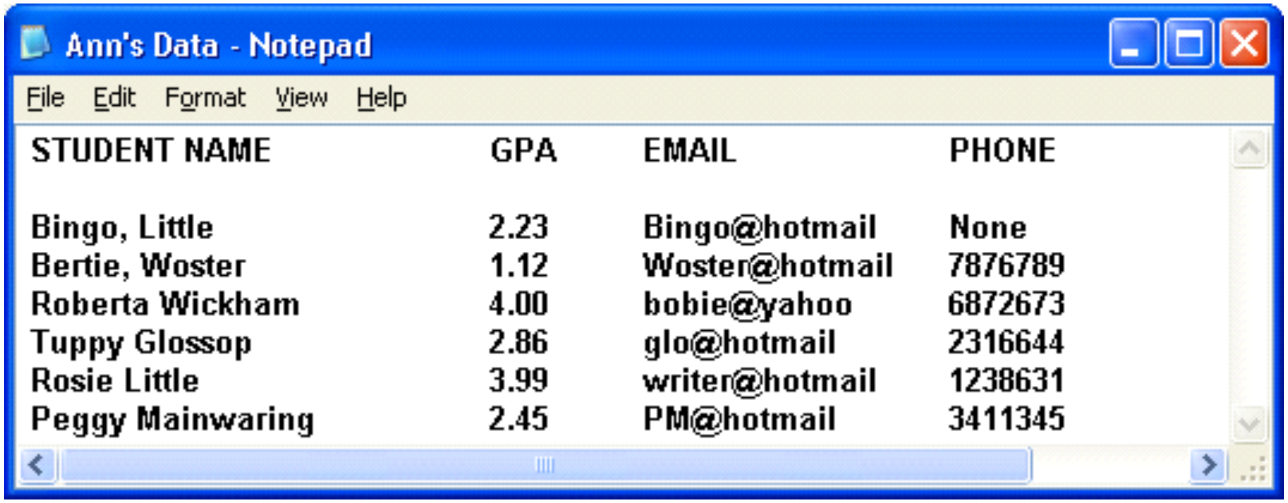


STUDENT NAME	GPA	EMAIL	PHONE
Bingo, Little	2.23	Bingo@hotmail	None
Bertie, Woster	1.12	Woster@hotmail	7876789
Roberta Wickham	4.00	bobie@yahoo	6872673
Tuppy Glossop	2.86	glo@hotmail	2316644
Rosie Little	3.99	writer@hotmail	1238631
Peggy Mainwaring	2.45	PM@hotmail	3411345

Security

- Suppose I leave my text file on UNIX account, and a student hacks in and changes their grades…

- A DBMS will allow multiple levels of security.

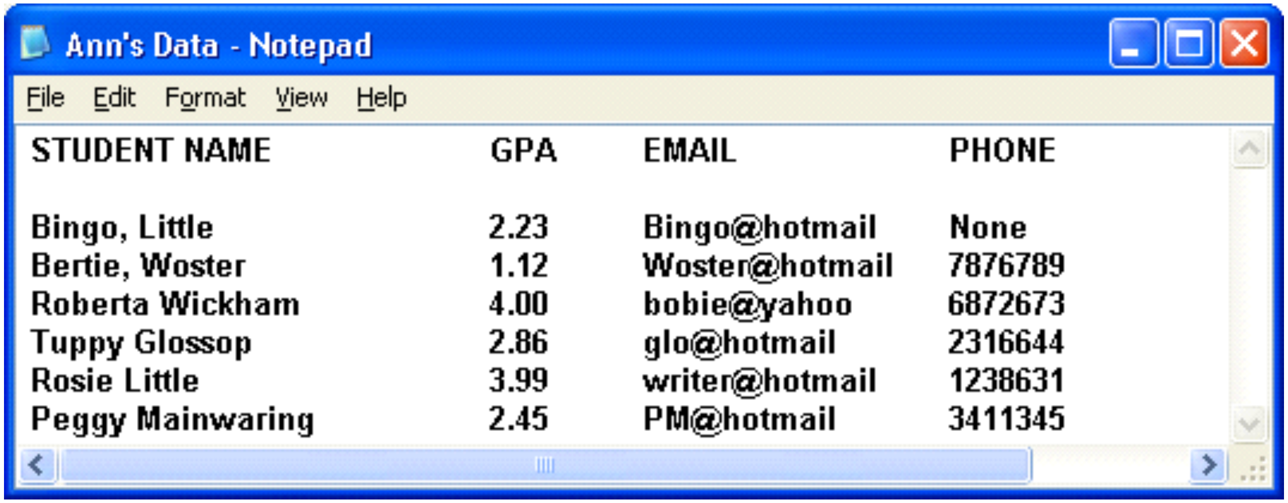


STUDENT NAME	GPA	EMAIL	PHONE
Bingo, Little	2.23	Bingo@hotmail	None
Bertie, Woster	1.12	Woster@hotmail	7876789
Roberta Wickham	4.00	bobie@yahoo	6872673
Tuppy Glossop	2.86	glo@hotmail	2316644
Rosie Little	3.99	writer@hotmail	1238631
Peggy Mainwaring	2.45	PM@hotmail	3411345

Crash Recovery

- Suppose I am editing my text file and the system crashes!

- A DBMS is able to guarantee 100% recovery from system crashes (to a consistent state).



STUDENT NAME	GPA	EMAIL	PHONE
Bingo, Little	2.23	Bingo@hotmail	None
Bertie, Woster	1.12	Woster@hotmail	7876789
Roberta Wickham	4.00	bobie@yahoo	6872673
Tuppy Glossop	2.86	glo@hotmail	2316644
Rosie Little	3.99	writer@hotmail	1238631
Peggy Mainwaring	2.45	PM@hotmail	3411345

Data Independence

- Applications are insulated from how data is structured and stored.
- *Logical data independence*: Protection from changes in *logical* structure of data.
- *Physical data independence*: Protection from changes in *physical* structure of data.

☛ *One of the most important benefits of using a DBMS!*

Purposes of DBMS

- Provide support for “easy-to-use” data
  - Data model (data)
  - Transaction model (operation)
- Provide efficient storage and access of the data in terms of the data model and transactional model.

To sum up: Why Use a DBMS?

Easier and More Efficient

- Data independence and efficient access.
- Query expressiveness
- Reduced application development time.
- Data integrity and security.
- Concurrent access, recovery from crashes.

- Any reasons to NOT use a DBMS?

Why Learn about DBMS?

- Many decisions about how to use a DBMS for an application depend on the capabilities of the DBMS
- To use it well, it’s necessary to also understand how a DBMS *works*.

Database Users

- End users (or DB application users)
- DB application programmers (more precisely, they are *DBMS* users)
  - E.g. webmasters
- *Database administrator (DBA)*
  - Designs logical /physical schemas
  - Handles security and authorization
  - Data availability, crash recovery
  - Database tuning as needs evolve

Must understand how a DBMS works!

Data Models

- Data model: A class of mathematical structures, with description and operations
- Necessary to be *general* and *intuitive.*
- Conceptual data model: Just structural description
  - Identifies the highest-level relationships between different entities
  - Features include the main entities and the relationships among them

Overview of Database Design

- *Conceptual design*
  - Use *ER Model:* E- *Entities* and R-*Relationships*
  - Decide the *entities* and *relationships* in the enterprise.
  - Decide what information about these entities and relationships should we store in the database.
  - Decide the *integrity constraints* or *business rules*.
- *Implementation (logical design)*
  - Map an ER model into a relational schema.

Example: University Database

Students(*sid*:string, *name*:string, *login*:string, *age*:integer, *gpa*:real)

Faculty(*fid*:string, *fname*:string, *sal*:real)

Courses(*cid*:string, *cname*:string, *credits*:integer)

Enrolled(*sid*:string, *cid*:string, *grade*:string)

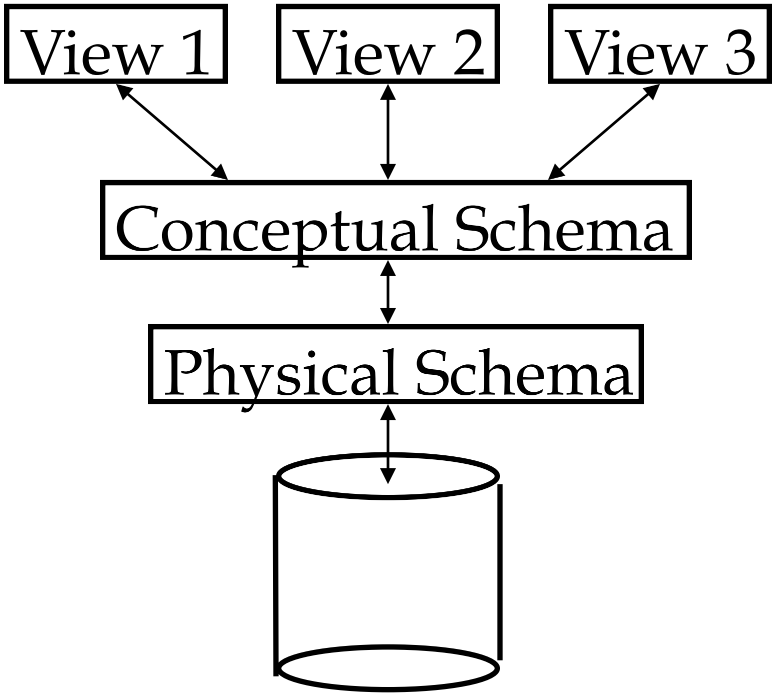
Teaches(*fid*:string, *cid*:string)

Levels of Abstraction

- Many *views*, single *conceptual (logical) schema* and *physical schema*.
  - Views describe how users see the data.

    e.g. CourseInfo(*cid*:string, *fname*:string, *enrollment*:integer)
  - Conceptual schema defines logical structure

    e.g. Courses(*cid*:string, *cname*:string, *credits*:integer)
  - Physical schema describes the files and indexes used.



Summary

- DBMS used to maintain, query large datasets.
- Benefits include recovery from system crashes, concurrent access, quick application development, data integrity and security.
- Levels of abstraction give data independence.
- We will learn how to
  - *Design* and set up a database
    - Design (ER and Relational Models), and refine (Relational Normalization Theory)
  - *Query* the database
    - Relational Algebra and SQL
  - *Implement* database applications