CS 450: Database Concepts

Fall 2015

Introduction

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 2-3pm
Thursday 1:30-2:30pm

Class Meeting: Tuesday/Thursday 3:00-4:15pm
Innovation Hall 206

Pre-requisites: C or better in CS 310 and CS 330

TA: Yuyang Gao (ygao13@gmu.edu)
TA Office Hours: TBA

Outline

- Course syllabus
- Introduction to DB & DBMS

Administration Trivia

- Class webpage: http://www.cs.gmu.edu/~jessica/cs450_f15.html
- In most cases, 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.

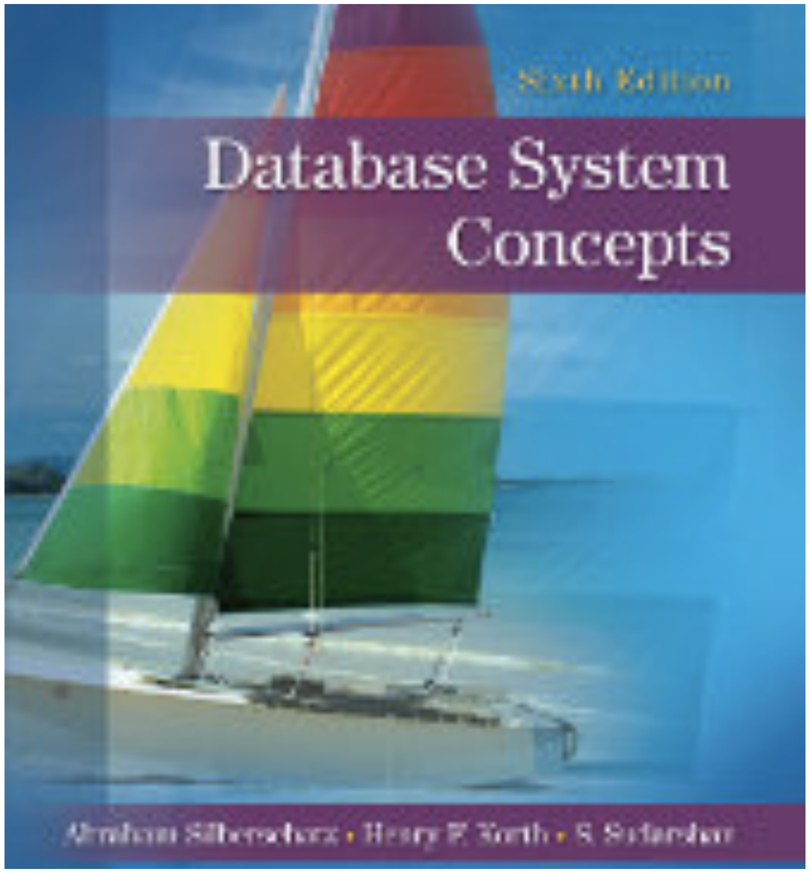
Textbook

- Required Book:
  - Database System Concepts by Silberschatz, Korth and Sudarshan, 6th edition

- Recommended Book:
  - Oracle 10g Programming: a Primer, by Rajshkhar Sunderraman, Addison Wesley, ISBN 0321463048

http://www.db-book.com/

6th Edition



Grading

- Midterm Exams: 30%
- Final Exam: 35%
- Project: 20%
- Assignments: 15%

- There will be two midterms and one final exam.
- Final exam is comprehensive
- All exams are closed-book, closed-notes.

Honor Code System

- GMU honor Code http://www.gmu.edu/academics/catalog/9798/honorcod.html
- In addition, the CS Department has specific honor code policies for programming projects, etc.: http://cs.gmu.edu/wiki/pmwiki.php/HonorCode/CSHonorCodePolicies
- For this class
  - You may work in a team of 2 for the project.
  - Homework: individual effort
  - Exams: individual effort, closed books/notes

Tools we will use for the class

- Oracle
  - More on this later
- 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 do not give out answers for assignments, obviously.
- Blackboard

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 and project
  - Checking grades
  - Getting course materials such as homework solutions
- Please bring a hardcopy to class.

Contact Policy

- Contact the Oracle support group (oracle@vse.gmu.edu) if you
  - Encounter technical difficulties with Oracle
- Contact the TA (or post on Piazza) if you
  - Have questions about the course materials
  - Have questions about homework or project
- Contact me if you
  - Have questions about the exams
  - Have general questions/concerns about the course

Email Policy

- Please 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.
- Please put [CS450] in the beginning of the subject line

Class Schedule

- See class website for the tentative class schedule (will be available tonight): http://www.cs.gmu.edu/~jessica/cs450_f15.html

Useful links for your computing needs

- http://labs.vse.gmu.edu/index.php/ for
  - Mason account information (Student FAQs -> "I need an account or password…")
  - VSE computing lab and Oracle DBMS information (Go to Student FAQs->Oracle).

Any Questions?

Introduction to Databases

Chapter 1

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 CS450)
- 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

Top 10 Largest Databases in the World (as of 2010)

10. Library of Congress

130 million items; 20TB of data; 530 miles of shelves

9. CIA

Comprehensive statistics on > 250 countries and entities

8. Amazon

59 million customers; > 42 TB of data

7. YouTube

45 TB of videos as of 2006

http://www.comparebusinessproducts.com/fyi/10-largest-databases-in-the-world

Top 10 Largest Databases in the World

6. ChoicePoint

   information on 250 million people; 250TB

5. Sprint

   2.85 trillion database rows; 365M call records processed/day

4. Google

   91M searches/day -> 33 trillion DB entries after 1 year

3. AT&T

   320 TB of info; 1.9 trillion phone call records

Top 10 Largest Databases in the World

2. National Energy Research Scientific Computing Center

   2.8 PB of data

1. World Data Centre for Climate

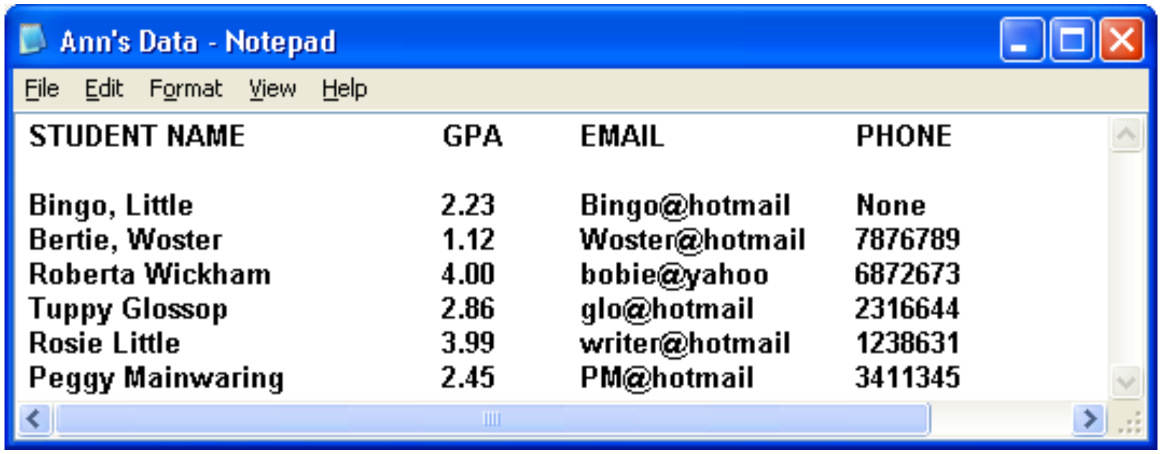
   220 TB of web data; 6 PB of additional data

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

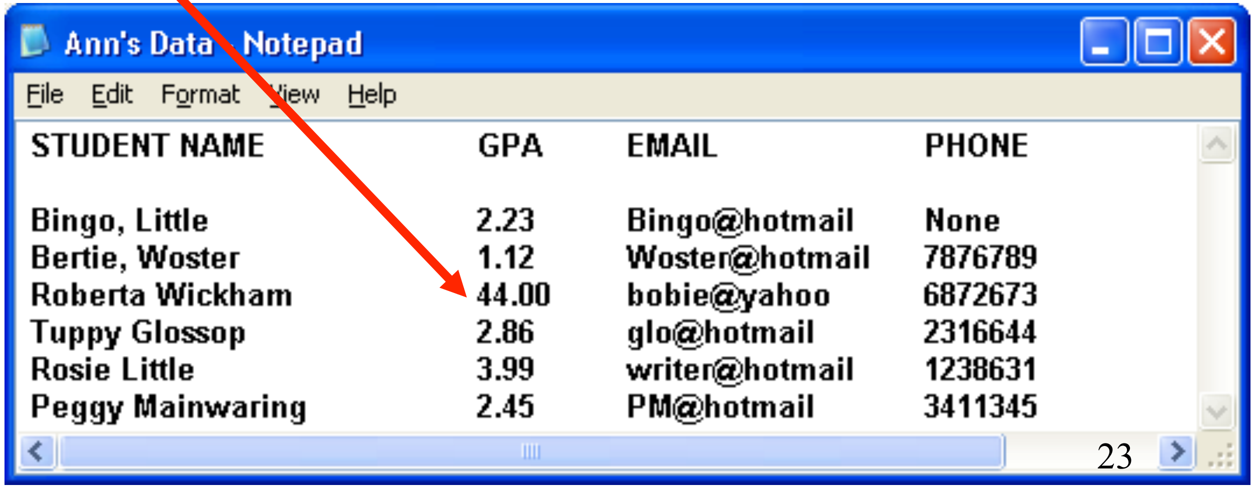
A simple solution to manage data:- stick them all in a text file!

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

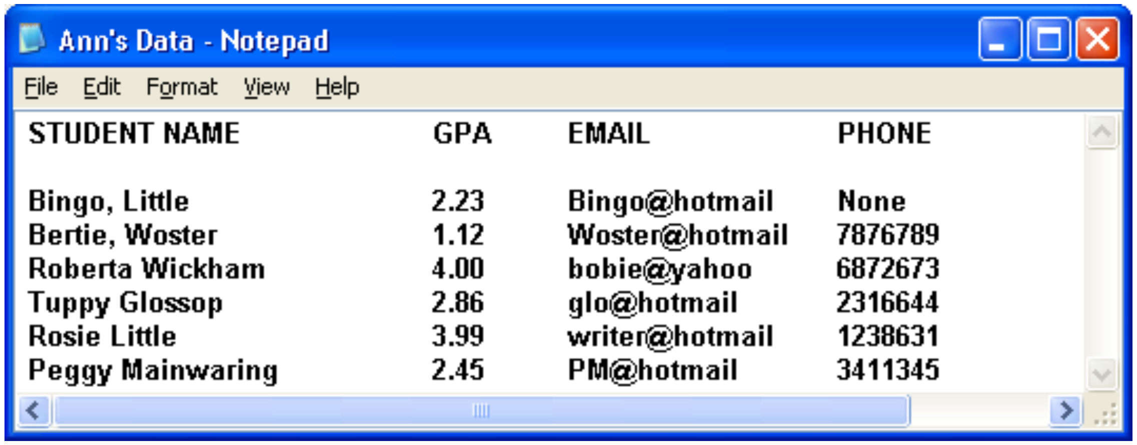


STUDENT NAME	GPA	EMAIL	PHONE
Bingo, Little	2.23	Bingo@hotmail	None
Bertie, Woster	1.12	Woster@hotmail	7876789
Roberta Wickham	44.00	bobie@yahoo	6872673
Tuppy Glossop	2.86	glo@hotmail	2316644
Rosie Little	3.99	writer@hotmail	1238631
Peggy Mainwaring	2.45	PM@hotmail	3411345

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.



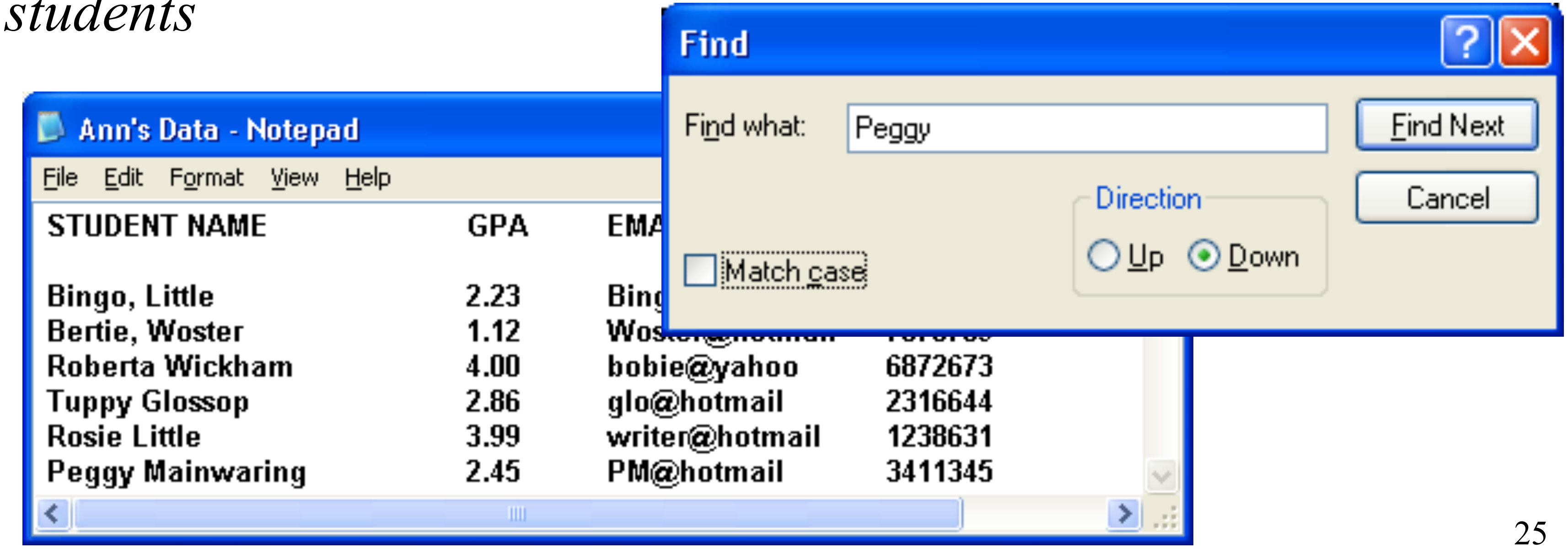
```
STUDENT NAME          GPA    EMAIL             PHONE

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

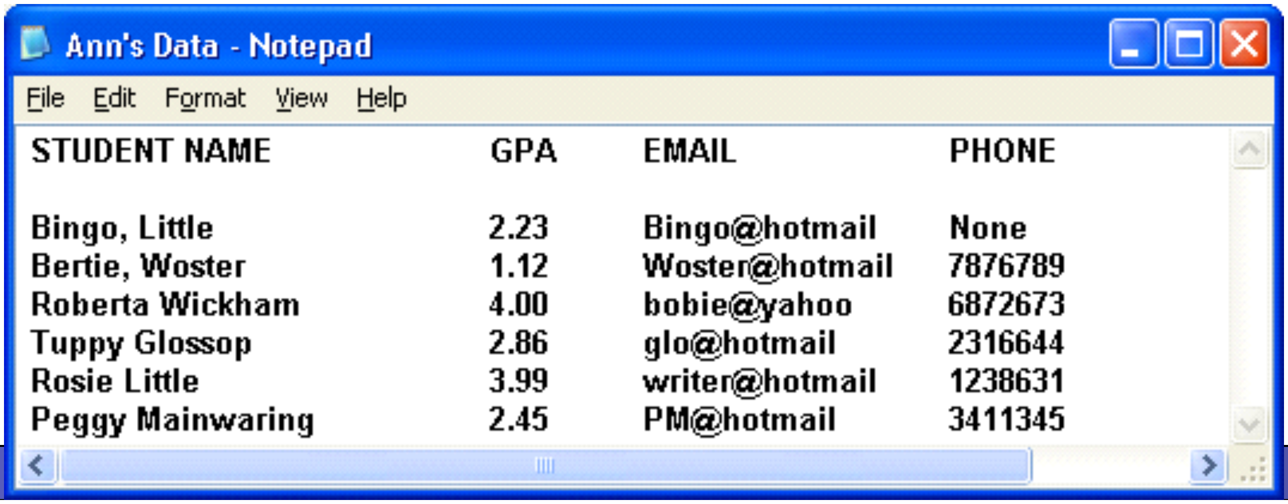
Direction: Up [✓ Down]

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), nothing changes at the front end!

• Also lead to reduced application development time.



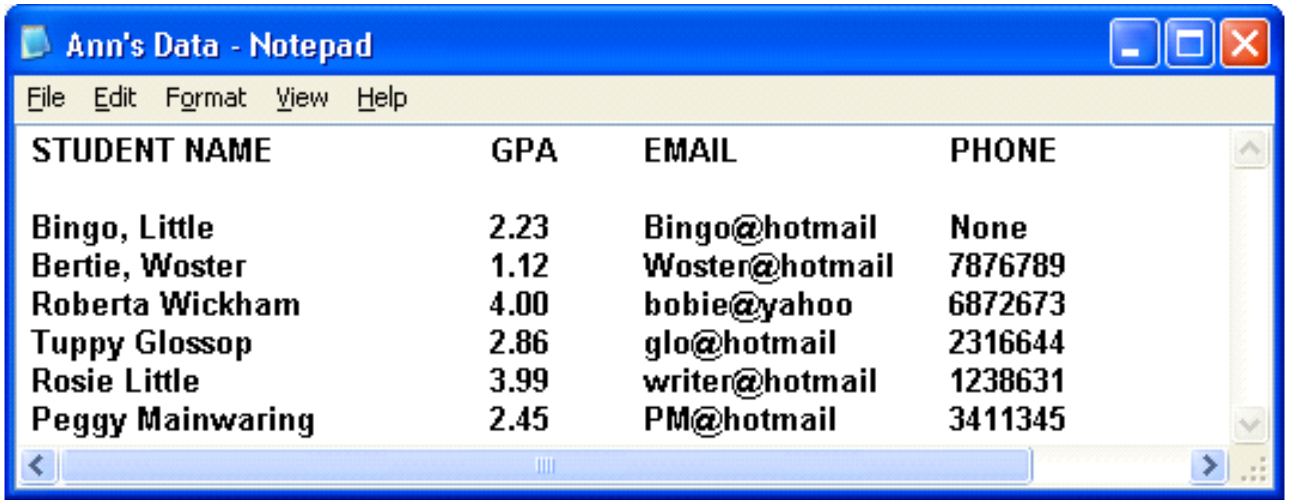
```
STUDENT NAME          GPA    EMAIL            PHONE

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.



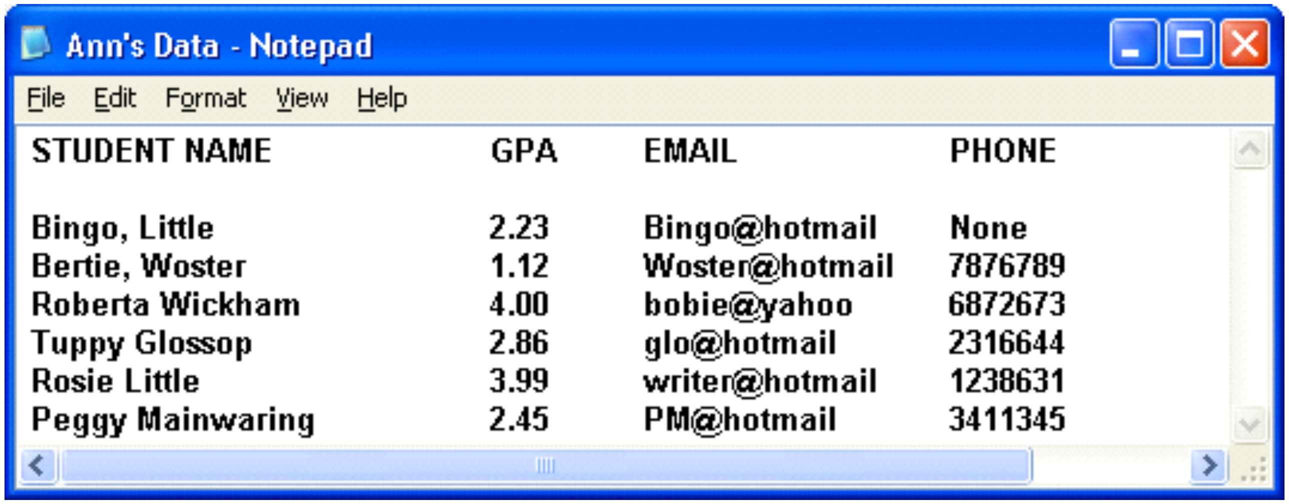
```
STUDENT NAME          GPA    EMAIL            PHONE

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 handle concurrent requests.

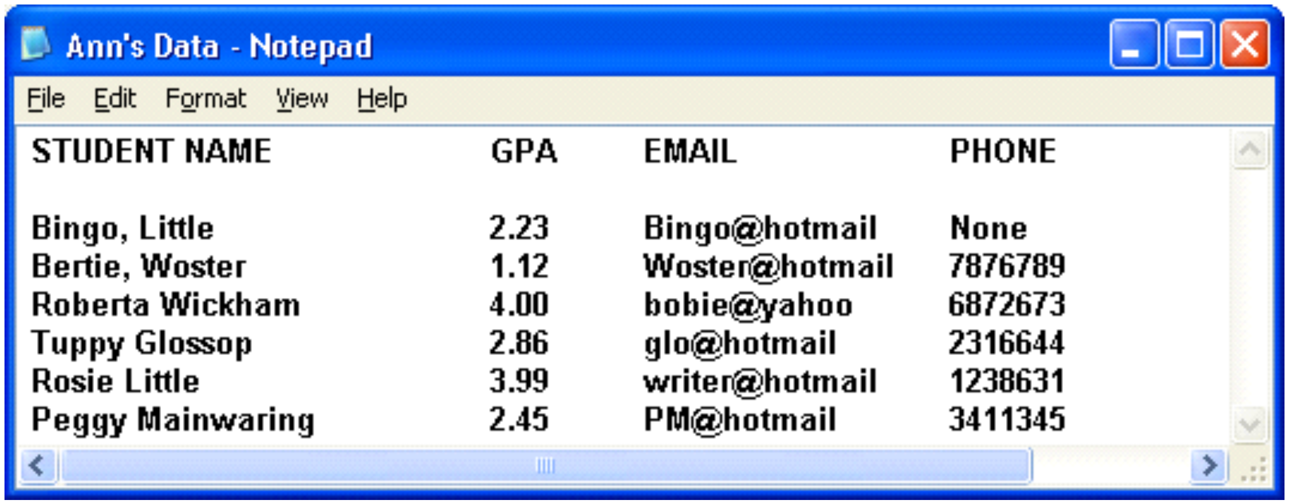


STUDENT NAME	GPA	EMAIL	PHONE
Bingo, Little	2.23	Bingo@hotmail	None
Bertie, Woster	1.12	Woster@hotmail	7876789
Roberta Wickham	4.00	bobie@yahoo	6872673
Tuppy Glossop	2.86	glo@hotmail	2316644
Rosie Little	3.99	writer@hotmail	1238631
Peggy Mainwaring	2.45	PM@hotmail	3411345

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.

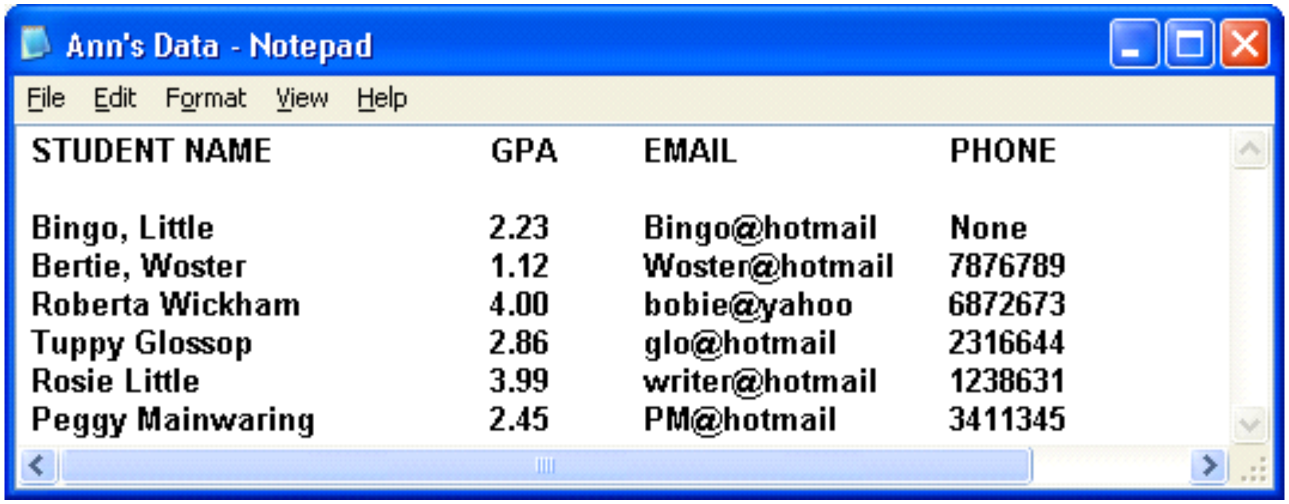


STUDENT NAME	GPA	EMAIL	PHONE
Bingo, Little	2.23	Bingo@hotmail	None
Bertie, Woster	1.12	Woster@hotmail	7876789
Roberta Wickham	4.00	bobie@yahoo	6872673
Tuppy Glossop	2.86	glo@hotmail	2316644
Rosie Little	3.99	writer@hotmail	1238631
Peggy Mainwaring	2.45	PM@hotmail	3411345

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 / naïve 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!

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

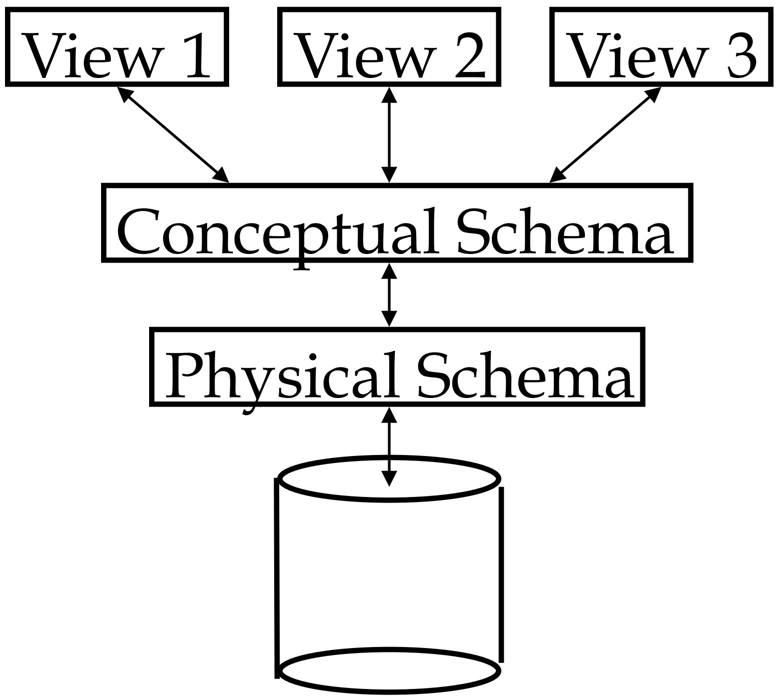
Students(*sid*:string, *name*:string, *login*:string, *age*:integer, *gpa*:real)
Faculty(*fid*:string, *fname*:string, *sal*:real)
Courses(*cid*:string, *cname*:string, *credits*:integer)
Enrolled(*sid*:string, *cid*:string, *grade*:string)
Teaches(*fid*:string, *cid*:string)

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