



EECS 484: Database Management Systems

Prof. H V. Jagadish jag@umich.edu



Course Outline – EECS 484

- GOAL: Basic introduction to database management systems.
- Two perspectives:
 - **External** (*Database user*)
 - Data models, ER model, relational model, SQL, database design ...
 - Java/JDBC Project: Common platform for building database applications
 - **Internal** (*Database implementer*)
 - File organizations, access methods, sorting, concurrency control, recovery, ...
 - Minirel Project: Build components of a Relational Database System
- Textbook “Database Management Systems”, by Raghu Ramakrishnan & Johannes Gehrke. 3rd ed.
 - Textbook is required.
 - Weekly reading ~50 pp. You will not regret it.



Java

- Databases are most often accessed via a declarative query language, SQL.
- SQL is usually embedded in, and called from, a traditional (procedural) programming language.
- Java is common choice, and so you will be using that in a project.



Course Policies

Make sure you have access to

- Canvas
- Piazza
- Discord
- Office Hour Queue
- Oracle (soon)
- Textbook



Groups

- Four projects + six homeworks
- Highly recommended: project group of size 2.
- Single-person project submission is allowed (but discouraged!)
- Start looking for partners now!



Project Grading

- Mostly autograder, some human.
- Limited number of submissions, even for autograded portion.
 - Make sure to test extensively.
- Both partners are expected to contribute to and be familiar with all aspects of the project.



Course Policies

- Projects
 - Due by **11:55 PM** on a Thursday.
 - 4-day late submission period.
 - You may take two late days without penalty, total for the whole semester.
 - 5 percentage points per late day beyond the two free, taken off the top from the possible score for the project. (Since each project is worth 10% in the semester grade, this is effectively 0.5% taken off your total score for the semester).
 - Note that office hours etc. are scheduled to maximize help on Wednesdays and Thursdays, with much less Friday through Sunday.



Course Policies

- Homework assignments
 - Due by **11:55 PM** every Thursday without a project/exam.
 - No late submissions accepted, but lowest score dropped.
 - Excuses for late or missed homework generally not accepted.
 - This is the reason to let you drop one HW.



Course Grading

First Half Exam	25%
Second Half Exam	25%
Six homework assignments @ 2% ea. Drop lowest	10%
Four projects @ 10% ea.	40%

A bonus of up to +/-2% may be awarded for "service to the class", such as asking good questions during class, answering peer questions on piazza, etc. Or disruptive behavior, such as reasking the same question on piazza.



Exams

- Two exams: midterm and final
 - Non-cumulative.
- Details in course calendar and Piazza and ...



This week

- Yes, we have a discussion.
 - Java Tutorial + Associated Tools
 - Project 1 Intro
 - Don't miss it
- No office hours this week.
- The regular schedule for office hours starts next week.
- Info in course calendar.



Discussion Sections

- Not optional!
- Project and homework discussion.
- Discussions sometimes run ahead of lectures or cover additional relevant topics.



Lectures

- Video tape of lecture posted on canvas (only for lecture section 002).
- Lecture notes posted on canvas
- Sometimes updated after lecture.
 - To fix errors
 - To add clarifications



Office Hours

Regular office hours will be virtual only.
(On zoom).

Special office hours will be used, in place of regular office hours, immediately before projects are due. These will be in person only.



Registration Issues, Wait Lists

- If you are a CSE student in any program, and wait listed before Aug 20, you should have received permission to register.
- We cannot expand the class size further, so anyone wait listing after Aug 20 will get in only as registered students drop.



Cross Attendance

- You may attend either lecture section, as long as space is available. Content is identical. Lec 001 (MW at 3pm) is in a smaller class room.
- You may attend any discussion section.
 - There are 9, all on Thursday or Friday.
 - But, please attend the same one each week so that the instructor and you get a chance to know each other..



Honor Code – Course Policies

- CoE Honor Code for all students
- Key principle: No unfair advantage
- Your work must be original – no peeking at old solutions, sharing of code, or discussing the projects beyond your group
- **No public posting of solutions**, e.g., even after the course.
- **Private repos** to share with your partner or a potential employer are OK.
- Posting questions on Piazza or using office hours for help is fine
- Questions? – Just ask.
- Also see Canvas for link to CoE Honor Code.



Overview of DBMS and Topics

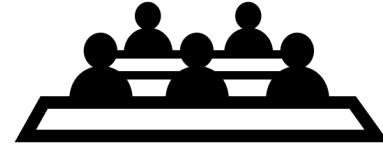


What Is a DBMS?

- DBMS = Database Management System
- Database: Large, structured collection of data.
- Models some real-world *enterprise*
 - Entities (e.g., students, courses)
 - Relationships (e.g., Lisa Simpson is taking EECS 484)
- **DBMS**: a software package designed to store and manage databases
- Oracle, MySQL, etc. are DBMS, but are informally called “databases”.

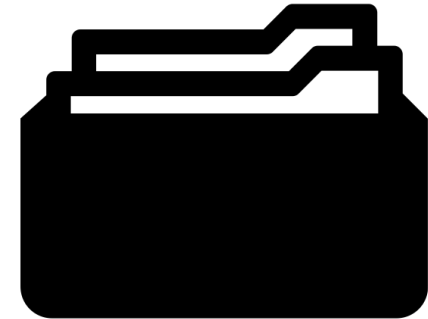


How to Store Student Data?



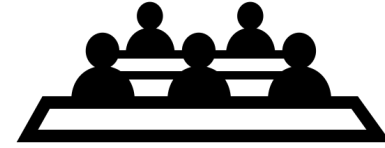
Old-time Solution: Physical Folders, Sorted by Lastname

- Advantages?
- Disadvantages?

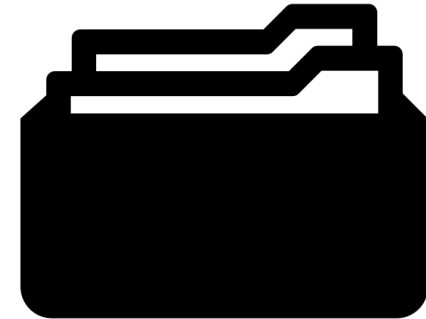




Old-time Solution: Sorted Student Folders

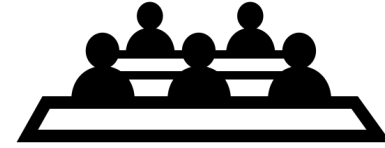


- Advantages?
 - cheap, universal, few dependencies
- Disadvantages?
 - Large weight & volume
 - Difficult to share
 - No ad-hoc queries

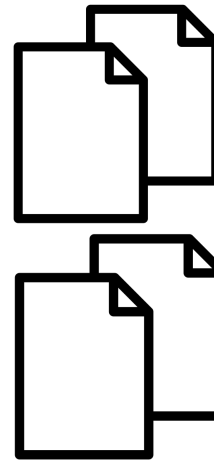




Other Solution: Flat Files

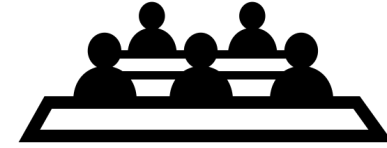


- Access?
 - using programs in C, Java, Python etc.
- Layout for the student records?





Other Solution: Flat Files



- Access?
 - using programs in C, Java, etc.
- Layout for the student records?

CSV: *Comma separated file*

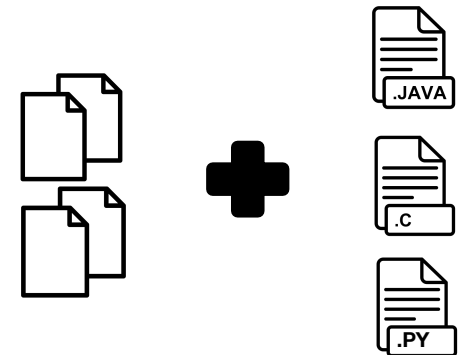
Brown, Lisa, lbrown, db, A, os, B

Smith, Bart, bsmith

Tompson, Mary, mtom, vis, B+, db, A-

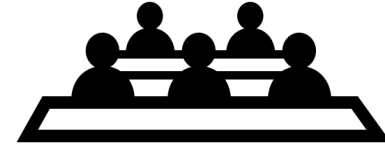
...

...

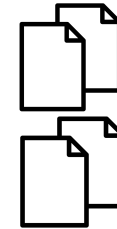




Other Solution: Flat Files



- Access?
 - using programs in C, Java, etc.
- Layout for the student records?



Multiple files:



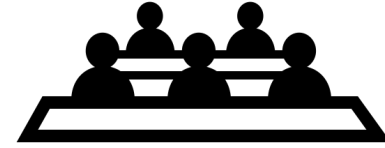
Brown, Lisa, lbrown
Smith, Bart, bsmith
Tompson, Mary, mtom
...
...



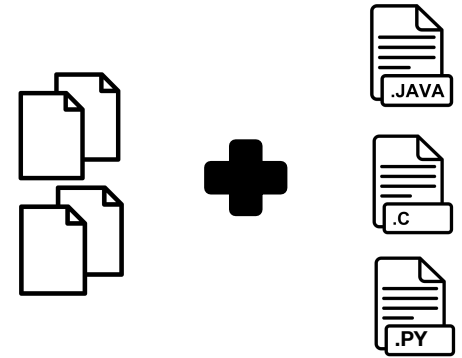
lbrown, db, A
lbrown, os, B
mtom, vis, B+
mtom, db, A-
...
...



Other Solution: Flat Files

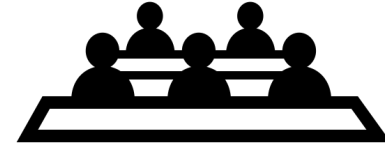


- Problems?

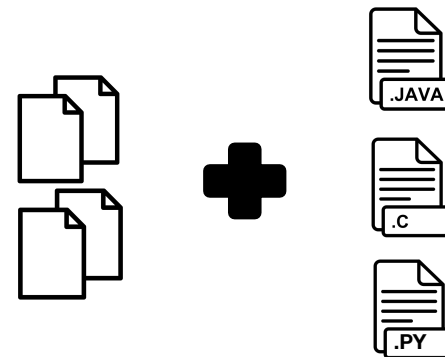




Other Solution: Flat Files



- Problems?
 - Inconvenient access to data
 - requires programming experience and knowledge of file layout
 - Data redundancy
 - Integrity problems
 - Atomicity problems (concurrent access issues)
 - Security problems



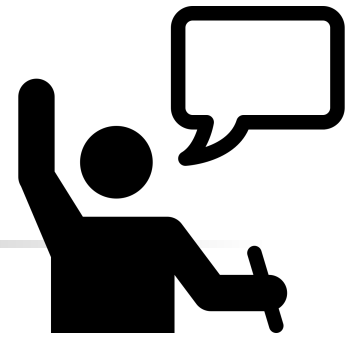


Why use a DBMS?



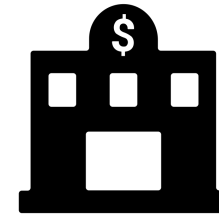
- It solves ALL these problems!
 - Data independence
 - Apps need a view of the data, not info about internal representation and storage
 - Efficient storage and access
 - Centralized data administration
 - Data integrity and security
 - Concurrent access, recovery from crashes
 - Reduced application dev time

Who uses a DBMS?



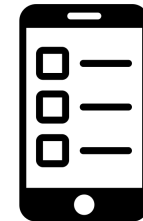
Who uses a DBMS?

- Everyone!
 - Your bank
 - Your university
 - Your coffee shop
 - Your favorite hotel
 - Your favorite website
 - Your phone
 - Your government
- Critical foundation for any AI system
- How many databases have you used so far today?



amazon

 **canvas**



- “Optimal” pricing of an airline ticket

Select your departure to Cancun Fri, Jan 8

Prices are one way per person, include all taxes and fees, but do not include baggage fees.

Filter your results by

Stops

☐ Nonstop (5)
 ☐ 1 Stop (54)
 ☐ 2+ Stops (1)

Airlines included


☐ American Airlines (17)
 ☐ Delta (14)
 ☐ Aeromexico (12)


Departure time

☐ Morning (5:00a - 11:59a)
 ☐ Afternoon (12:00p - 5:59p)

Sort by:

Price (Lowest)

From:	<div>  <div> <div>10:00a - 7:00p</div> <div>Air Canada</div> </div> </div>	<div> <div>9h 0m</div> <div>DTW - CUN</div> </div>	<div> <div>1 stop</div> <div>3h 40m in YYZ</div> </div>	<div> <div>\$230.07</div> <div>✓ Live one way</div> </div>
	Air Canada 8022 operated by Air Canada Express - Jazz Air Canada 1812 operated by Air Canada Rouge			
	Flight details and baggage fees »			

From:	<div>  <div> <div>7:05a - 7:00p</div> <div>Air Canada</div> </div> </div>	<div> <div>11h 55m</div> <div>DTW - CUN</div> </div>	<div> <div>1 stop</div> <div>6h 35m in YYZ</div> </div>	<div> <div>\$230.07</div> <div>✓ Live one way</div> </div>
	Air Canada 7281 operated by Air Canada Express - Air Georgian Air Canada 1812 operated by Air Canada Rouge			
	Flight details and baggage fees »			

- [illegible]



Data Models

Constructs.

- **Data model**: a collection of concepts for describing data.
- **Schema**: a description of a particular collection of data, using a given data model.
- **Relational model**: the most widely-used model today.
- **Entity-Relationship (ER) model**: A “semantic” data model, i.e., a higher-level more user-intuitive model
 - A (relational) DBMS understands only the relational model, so we will translate an ER schema to a relational schema



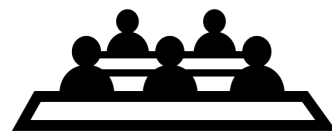
Relational and Other Data Models

- **DBMS using the relational DM**
(‘70s-‘80s)

- IBM DB2
- Informix
- Oracle
- Sybase
- Microsoft Access
- Tandem
- Teradata
- ...

- **Other data models**

- ✧ Hierarchical (mid ‘60s-‘70s)
 - IBM IMS
- ✧ Network (‘70s)
 - IDMS, IDS
- ✧ Object-oriented (~‘90s)
 - ObjectStore
- ✧ Object-relational (relational model + object DB concepts)
 - Oracle
- ✧ ...



Relational (Data) Model

- The most widely-used model today
 - A collection of **relations**
 - **Relation** = set of records, naturally represented as a table with rows and (named, typed) columns

Students

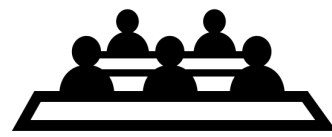
sid	name	login	age
13	Lisa	lsimp	40
41	Bart	bart	20

Courses

cid	cname	cred.
E-484	EECS484	4
E-584	EECS584	3

Enrolled

sid	cid	grade
41	E-484	A-
13	E-584	A+



Relational (Data) Model

- **Schema** = a description of data in terms of a data model

→ of the table structure.

- Every relation has a **schema**
- Specifies the **name** of the **relation**, the **name** and **type** of the **columns** (or *fields* or *attributes*)

Students(sid:string, name:string, login:string, age:integer)

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

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

- Each row also called a **tuple** or a **record**

Students

sid	name	login	age
13	Lisa	lsimp	40
41	Bart	bart	20

Courses

cid	cname	cred.
E-484	EECS484	4
E-584	EECS584	3

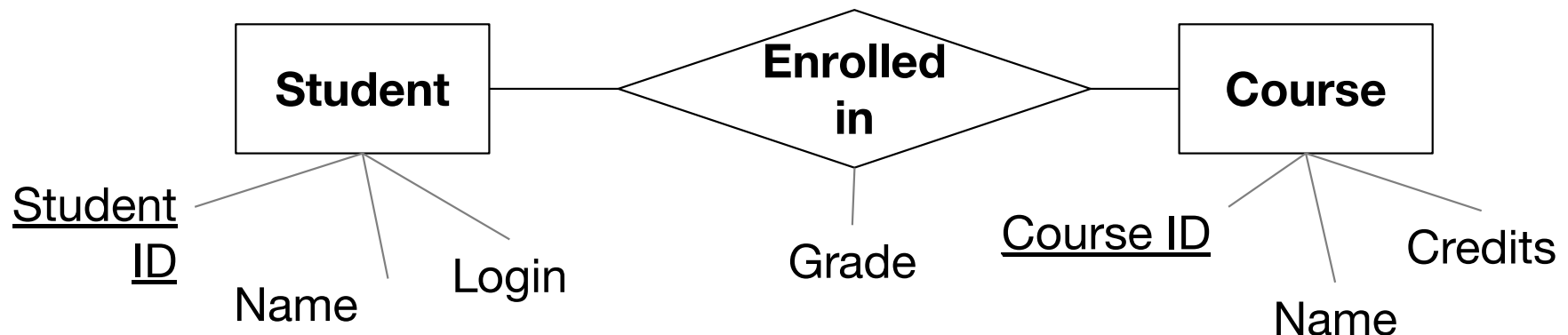
Enrolled

sid	cid	grade
41	E-484	A-
13	E-584	A+

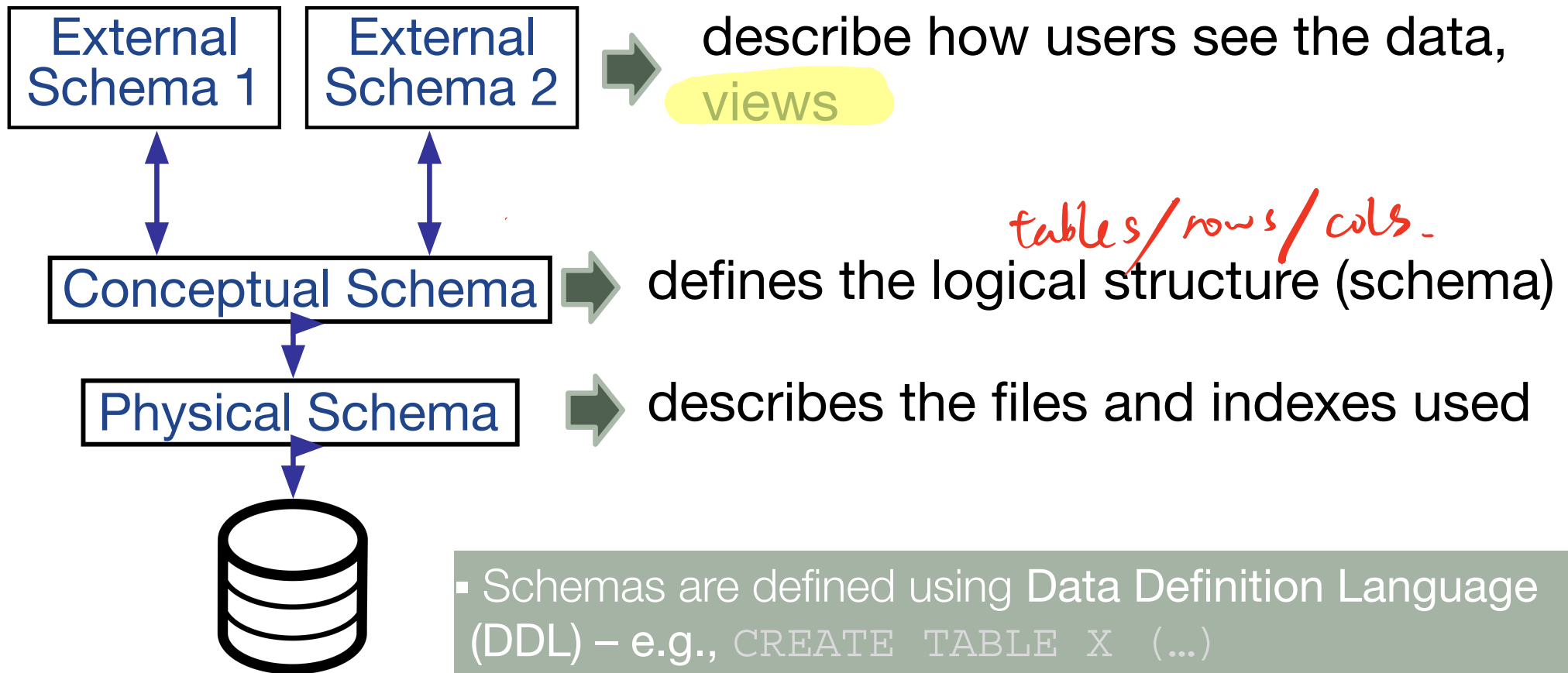


Entity-Relationship (ER) Model

- A “semantic” data model
 - a higher-level, user-intuitive model
- Entity-Relationship diagram:
 - ① • **Entities:** Student, Course
 - ② • **Relationship:** Enrolled_in



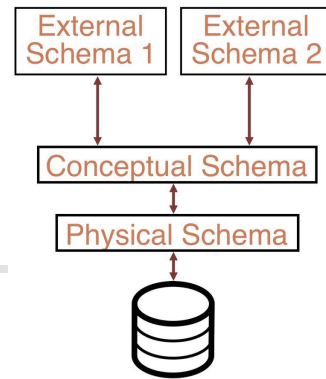
Levels of Abstraction



- Schemas are defined using Data Definition Language (DDL) – e.g., `CREATE TABLE X (...)`
- Data is modified/queried using Data Manipulation Language (DML) – e.g., `SELECT FROM X WHERE ...`

The DDL will specify the logical structure of the conceptual schema. The DML is a means to interact with the data after the database is defined. The external schema wouldn't really need to use DML because it's mainly how users VIEW the data or update it through input/enter methods. If someone were to be using DML to update a database, it would be best bet that they are interacting with the conceptual schema as well.

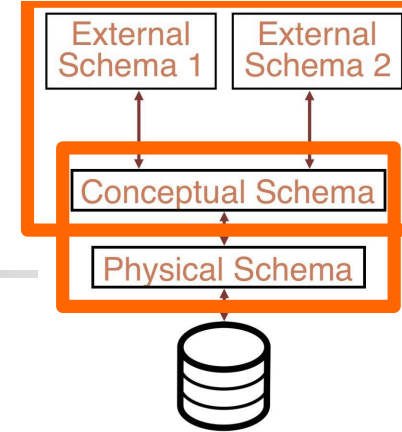
Example



- Conceptual schema (1):
 - `Students(sid:string, name:string, login:string, age:integer)`
 - `Courses(cid:string, cname:string, credits:integer)`
 - `Enrolled(sid:string, cid:string, grade:string)`
- Physical schema (1):
 - Relations stored as unordered files.
 - Index on first column of `Students`.
- External Schema (≥ 1):
 - View: `Course_info(cid:string, enrollment:integer)`
 - View: `Class_rank(sid:string, gpa:real, rank:integer)`

the view.

Data Independence



- Applications insulated from data format and storage details
- Logical data independence: Protection from changes in *logical* structure of data
 - External / Conceptual schema interface
- Physical data independence: Protection from changes in *physical* structure of data
 - Conceptual / Physical schema interface



CYU

- Which of these are more suitable for storing in a DBMS rather than files in an OS?
 - (a) Grades for students at the university
 - (b) Source code for a program
 - (c) Contents of a textbook



CYU

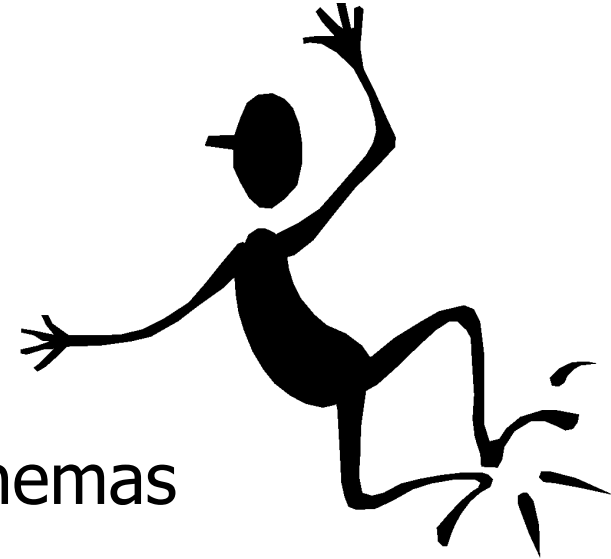
- Let's say UM provides you access to a relational table that gives just your grades in various courses. Does that relation represent:
 - a) An external schema?
 - b) A conceptual schema?
 - c) A physical schema?



- The relational table with student grade information is very large and stored on multiple servers for performance. Does the storage scheme represent:
 - a) An external schema?
 - b) A conceptual schema?
 - c) A physical schema?

Lots of People use DBMS ...

- DBMS vendors
- DB application programmers
 - E.g. smart webmasters
- *Database administrator (DBA)*
 - Designs external, logical, & physical schemas
 - Handles security and authorization
 - Data availability, crash recovery
 - Database tuning as needs evolve



DBA must understand how a DBMS works!



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.
- Data Management is a critical part of the ML pipeline.
 - Most of AI engineering is data engineering.
- DBAs hold responsible jobs and are **well-paid!** 😄
- DBMS R&D is one of the most exciting areas in CS. 🤔

Please read Chapter 2 (26 pp)

See you Wednesday