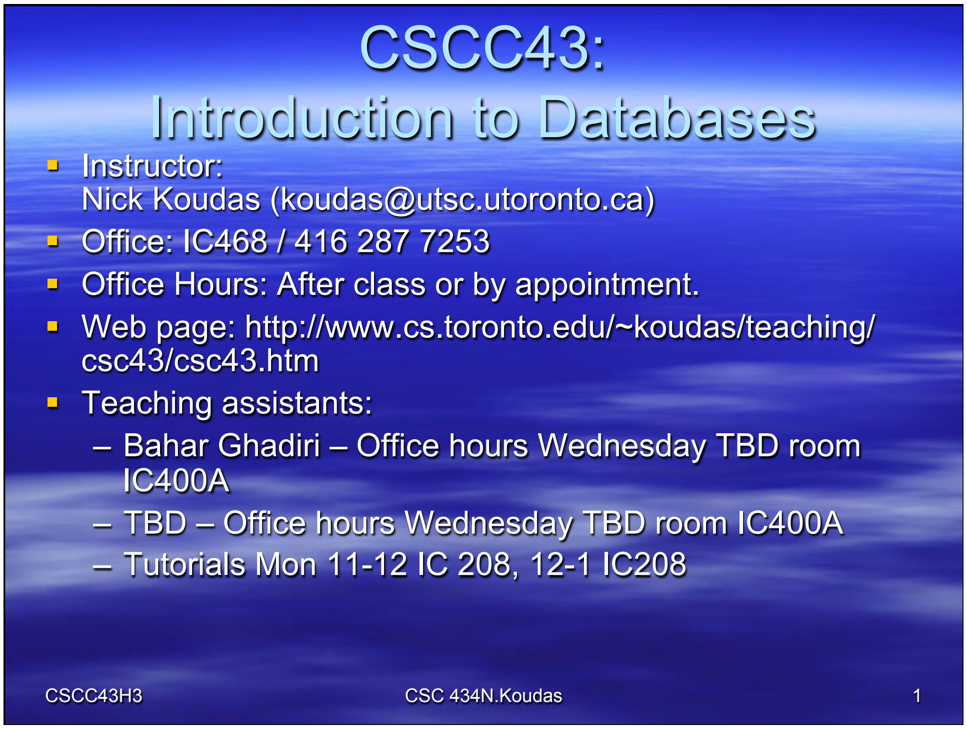
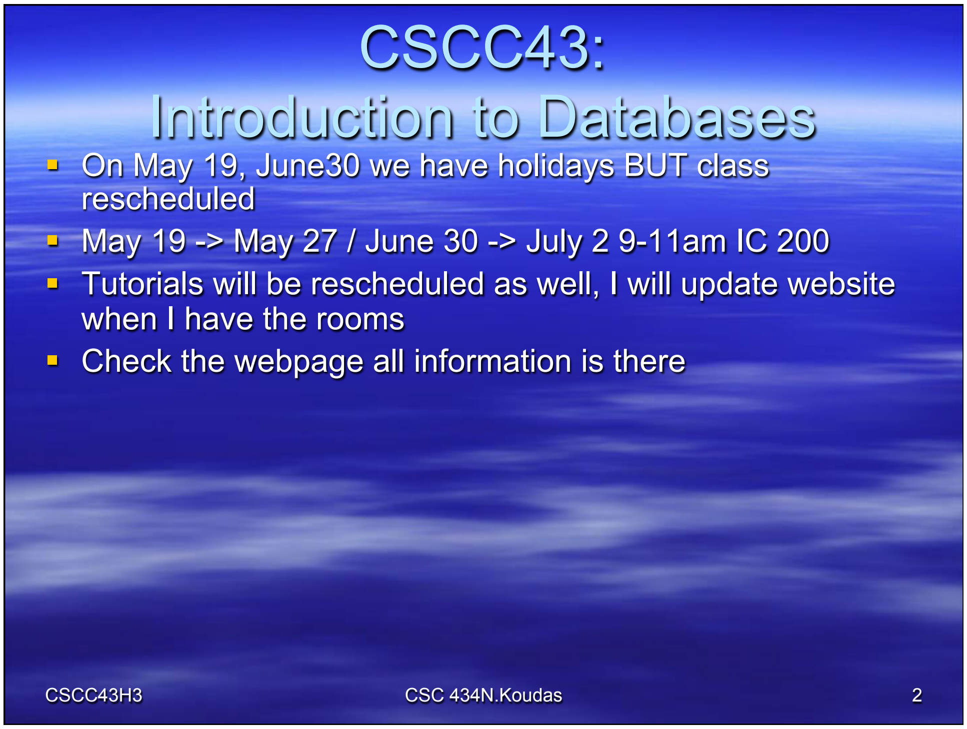


CSCC43: Introduction to Databases

- Instructor:
  Nick Koudas (koudas@utsc.utoronto.ca)
- Office: IC468 / 416 287 7253
- Office Hours: After class or by appointment.
- Web page: http://www.cs.toronto.edu/~koudas/teaching/csc43/csc43.htm
- Teaching assistants:
  – Bahar Ghadiri – Office hours Wednesday TBD room IC400A
  – TBD – Office hours Wednesday TBD room IC400A
  – Tutorials Mon 11-12 IC 208, 12-1 IC208

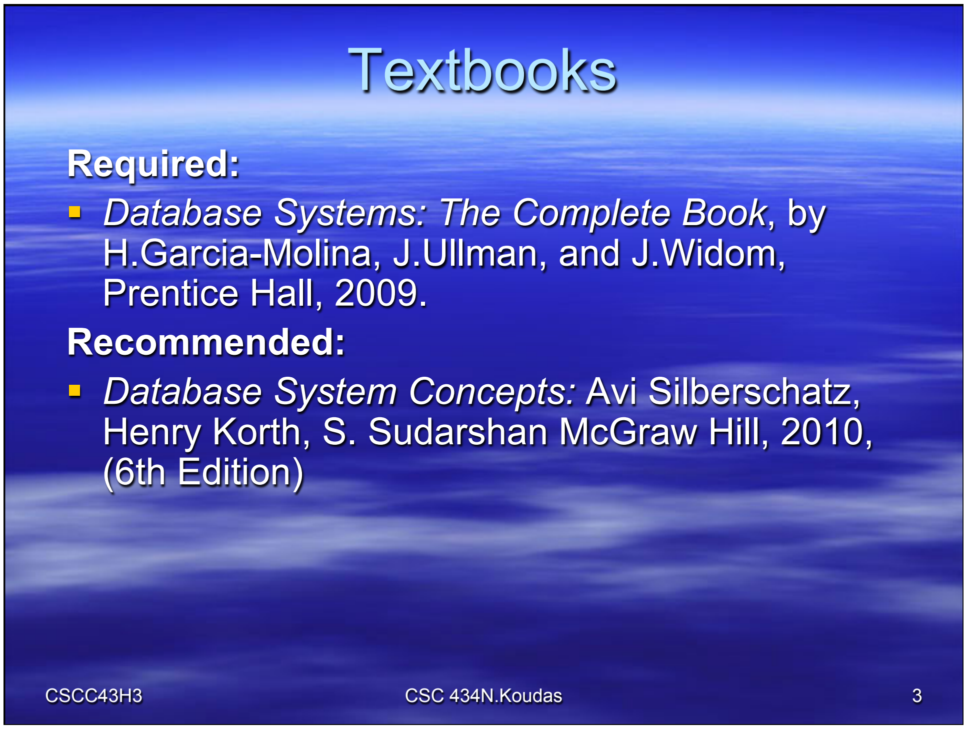
CSCC43H3 CSC 434N.Koudas 1



CSCC43: Introduction to Databases

- On May 19, June30 we have holidays BUT class rescheduled
- May 19 -> May 27 / June 30 -> July 2 9-11am IC 200
- Tutorials will be rescheduled as well, I will update website when I have the rooms
- Check the webpage all information is there

CSCC43H3 CSC 434N.Koudas 2



Textbooks

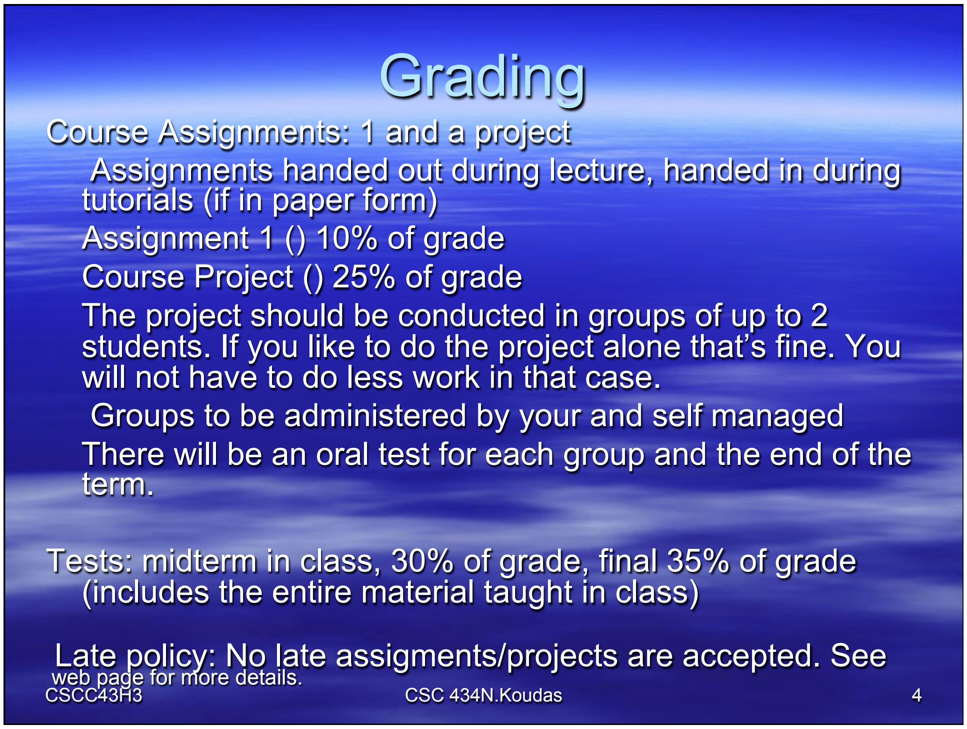
Required:

- *Database Systems: The Complete Book*, by H.Garcia-Molina, J.Ullman, and J.Widom, Prentice Hall, 2009.

Recommended:

- *Database System Concepts:* Avi Silberschatz, Henry Korth, S. Sudarshan McGraw Hill, 2010, (6th Edition)

CSCC43H3 CSC 434N.Koudas 3



Grading

Course Assignments: 1 and a project

Assignments handed out during lecture, handed in during tutorials (if in paper form)

Assignment 1 () 10% of grade

Course Project () 25% of grade

The project should be conducted in groups of up to 2 students. If you like to do the project alone that's fine. You will not have to do less work in that case.

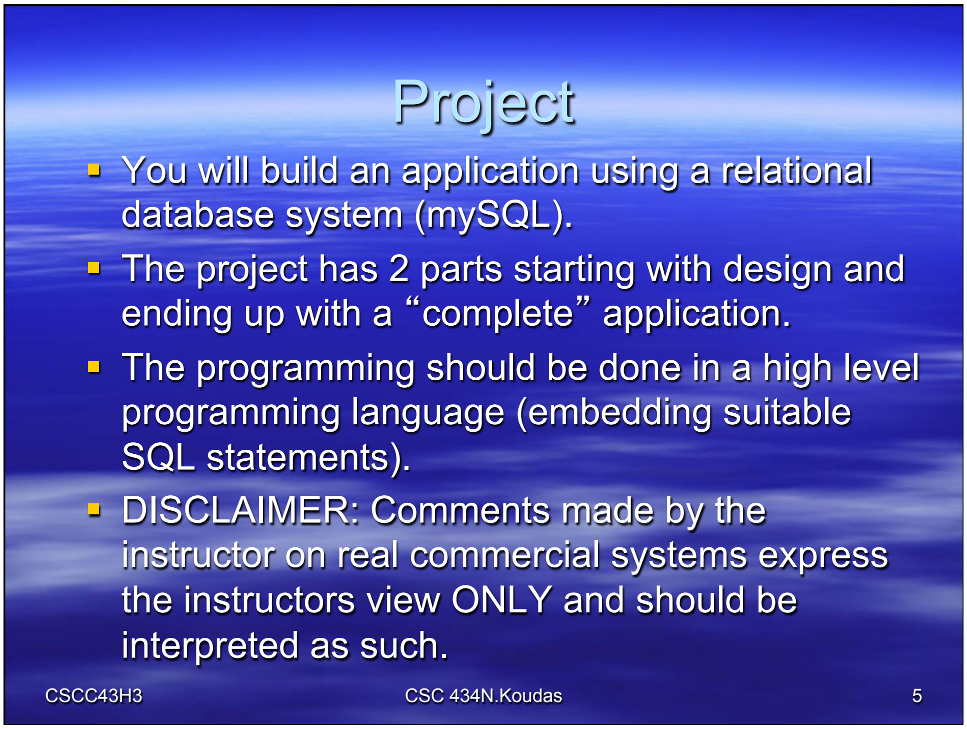
Groups to be administered by your and self managed

There will be an oral test for each group and the end of the term.

Tests: midterm in class, 30% of grade, final 35% of grade (includes the entire material taught in class)

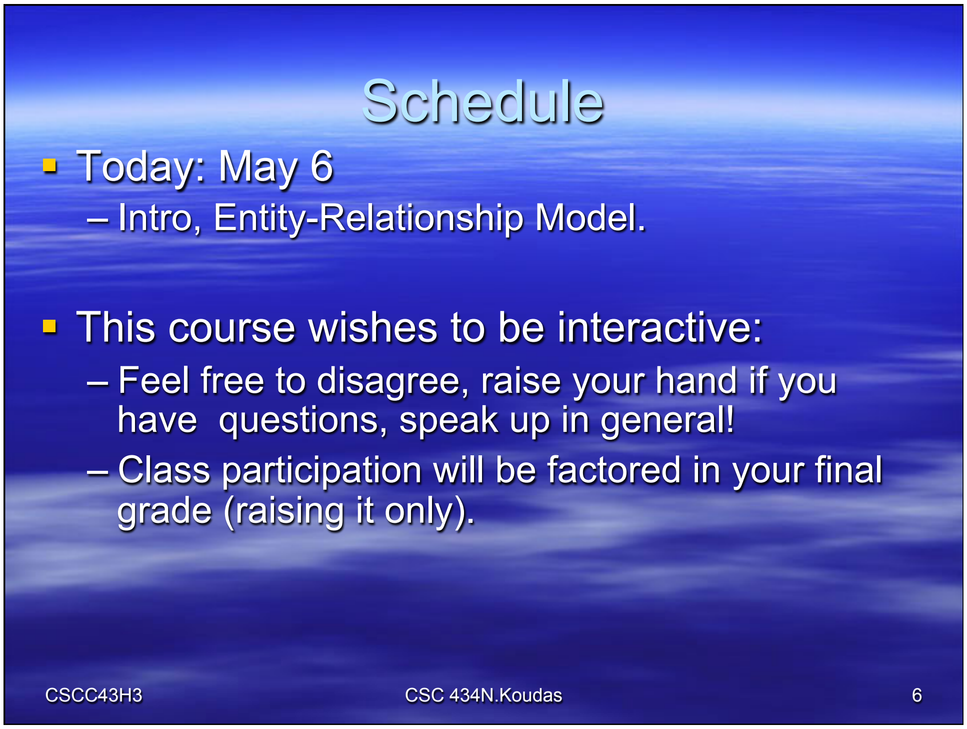
Late policy: No late assigments/projects are accepted. See web page for more details.

CSCC43H3 CSC 434N.Koudas 4



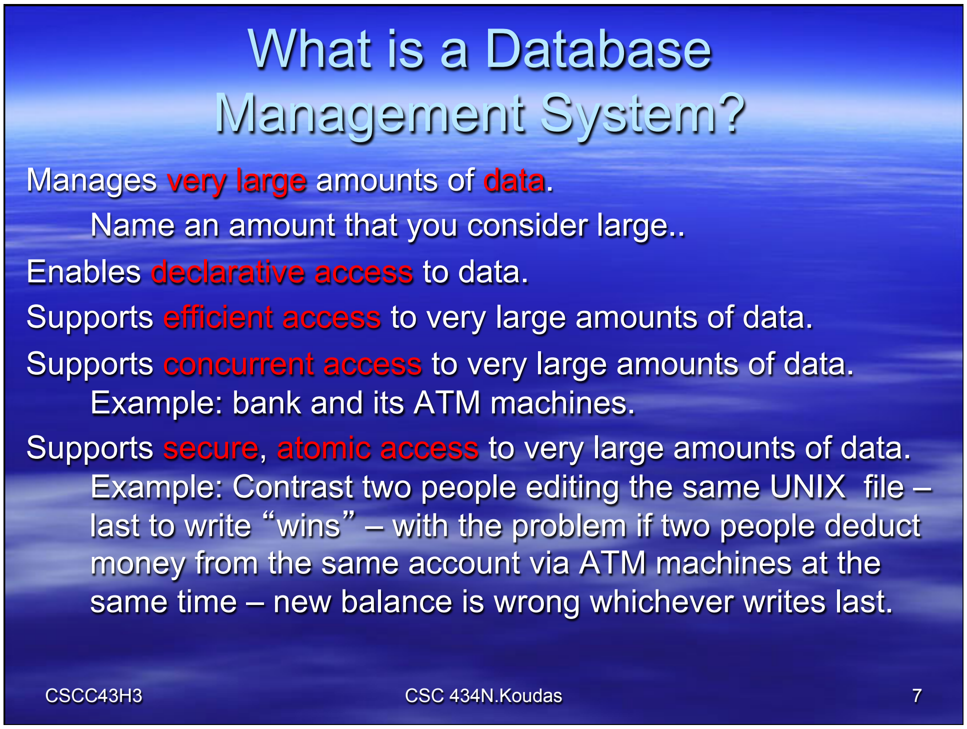
Project

- You will build an application using a relational database system (mySQL).
- The project has 2 parts starting with design and ending up with a “complete” application.
- The programming should be done in a high level programming language (embedding suitable SQL statements).
- DISCLAIMER: Comments made by the instructor on real commercial systems express the instructors view ONLY and should be interpreted as such.



Schedule

- Today: May 6
  - Intro, Entity-Relationship Model.
- This course wishes to be interactive:
  - Feel free to disagree, raise your hand if you have questions, speak up in general!
  - Class participation will be factored in your final grade (raising it only).



What is a Database Management System?

Manages very large amounts of data.

Name an amount that you consider large..

Enables declarative access to data.

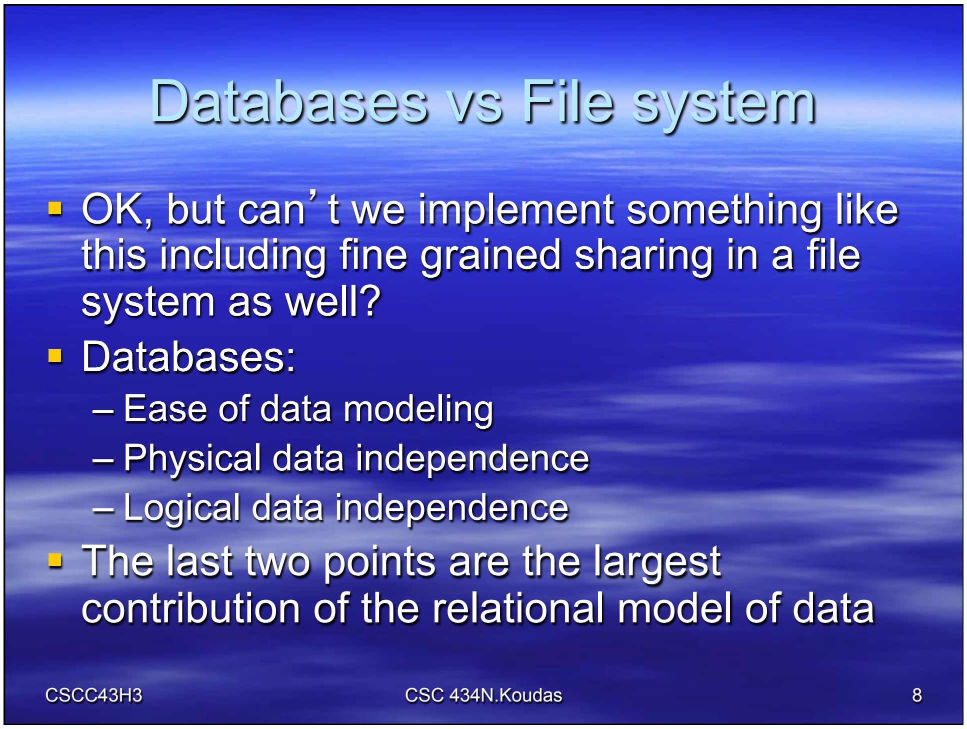
Supports efficient access to very large amounts of data.

Supports concurrent access to very large amounts of data.

Example: bank and its ATM machines.

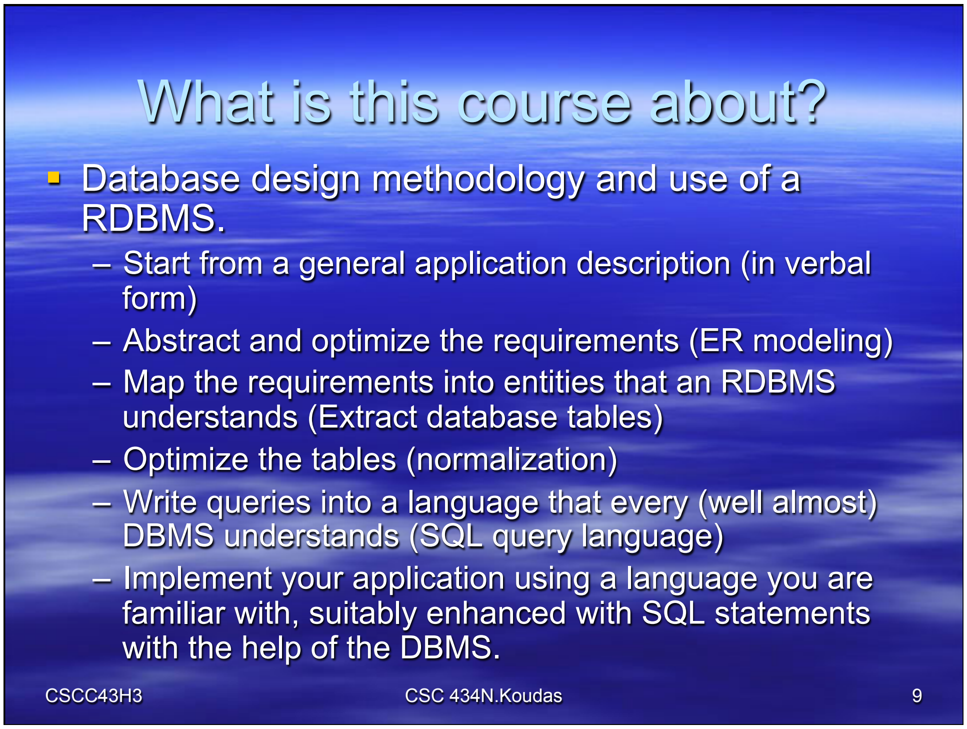
Supports secure, atomic access to very large amounts of data.

Example: Contrast two people editing the same UNIX file – last to write “wins” – with the problem if two people deduct money from the same account via ATM machines at the same time – new balance is wrong whichever writes last.



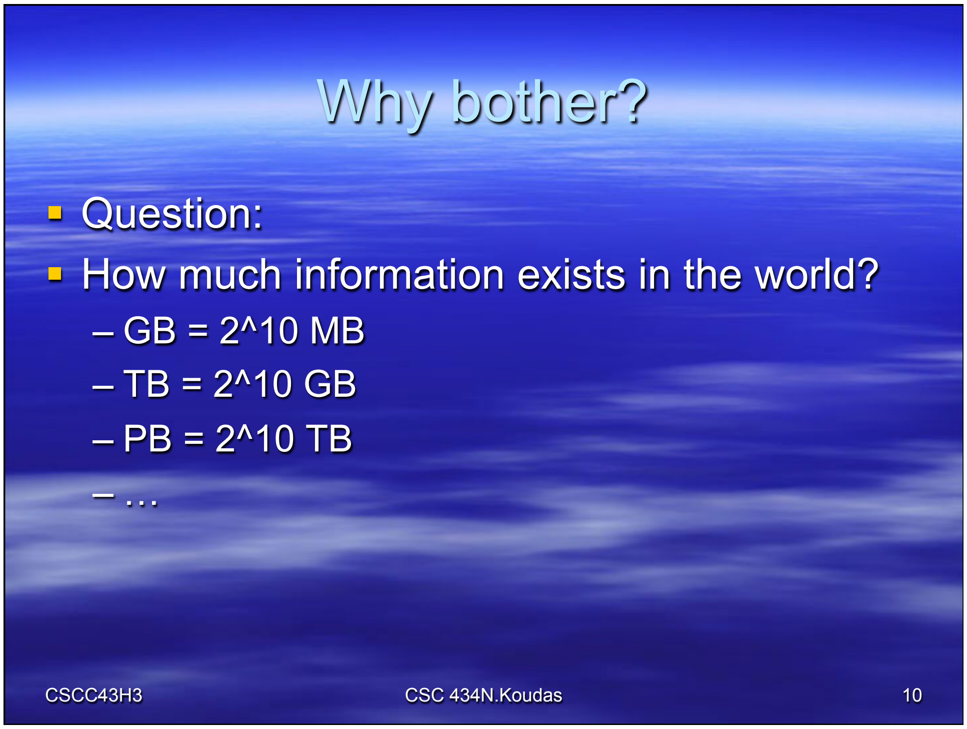
Databases vs File system

- OK, but can’t we implement something like this including fine grained sharing in a file system as well?
- Databases:
  - – Ease of data modeling
  - – Physical data independence
  - – Logical data independence
- The last two points are the largest contribution of the relational model of data



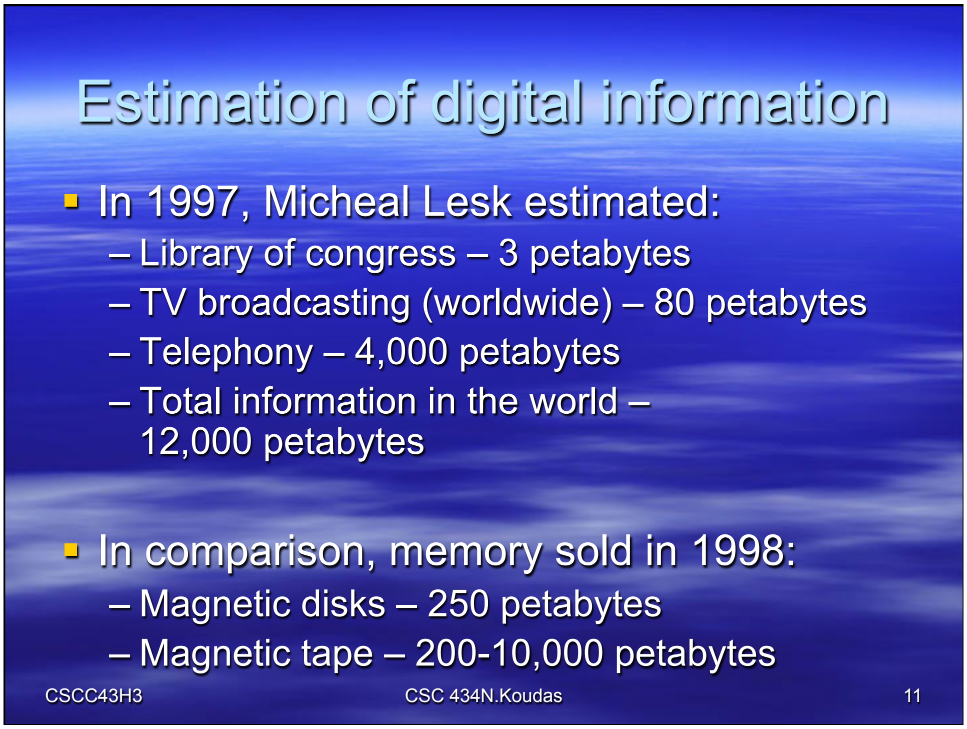
What is this course about?

- Database design methodology and use of a RDBMS.
  - Start from a general application description (in verbal form)
  - Abstract and optimize the requirements (ER modeling)
  - Map the requirements into entities that an RDBMS understands (Extract database tables)
  - Optimize the tables (normalization)
  - Write queries into a language that every (well almost) DBMS understands (SQL query language)
  - Implement your application using a language you are familiar with, suitably enhanced with SQL statements with the help of the DBMS.



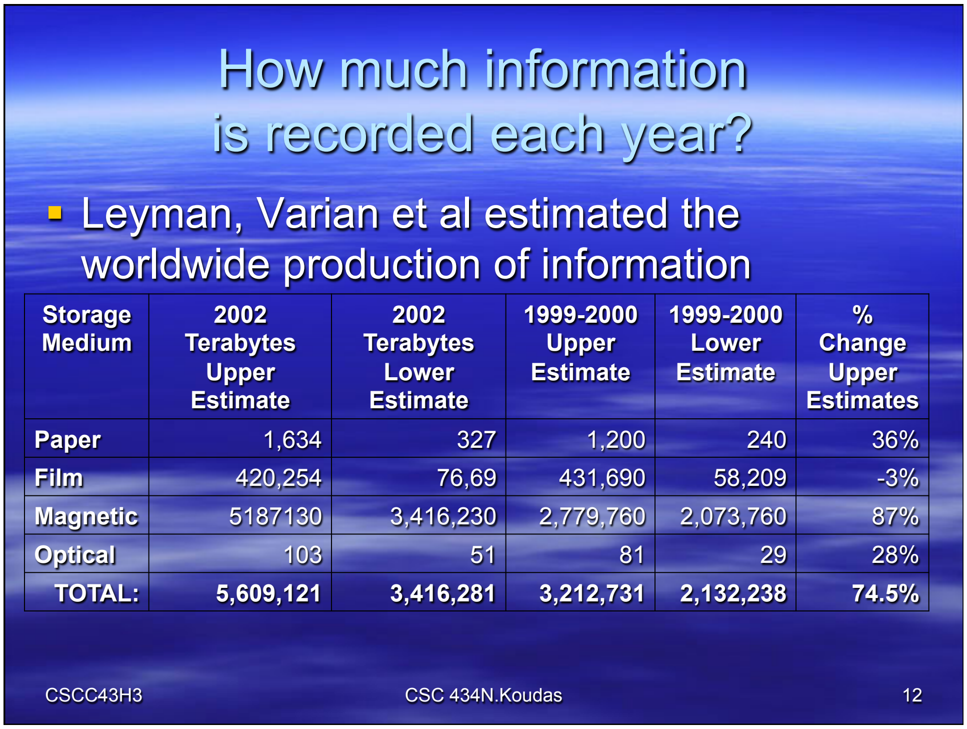
Why bother?

- Question:
- How much information exists in the world?
  - GB = 2^10 MB
  - TB = 2^10 GB
  - PB = 2^10 TB
  - …



Estimation of digital information

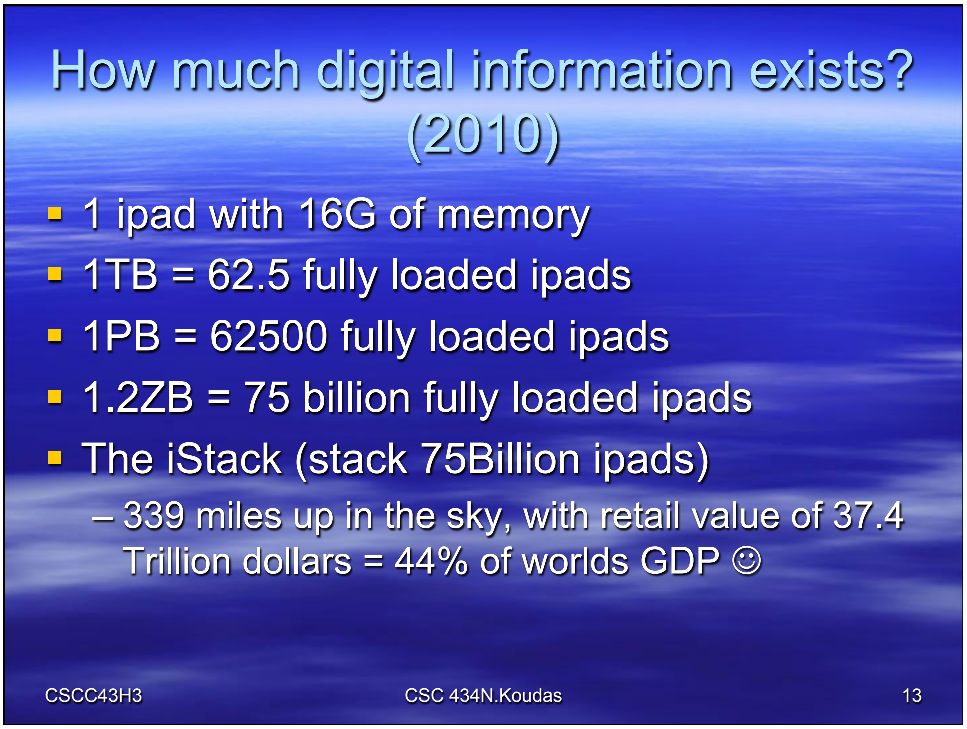
- In 1997, Micheal Lesk estimated:
  - Library of congress – 3 petabytes
  - TV broadcasting (worldwide) – 80 petabytes
  - Telephony – 4,000 petabytes
  - Total information in the world – 12,000 petabytes
- In comparison, memory sold in 1998:
  - Magnetic disks – 250 petabytes
  - Magnetic tape – 200-10,000 petabytes



How much information is recorded each year?

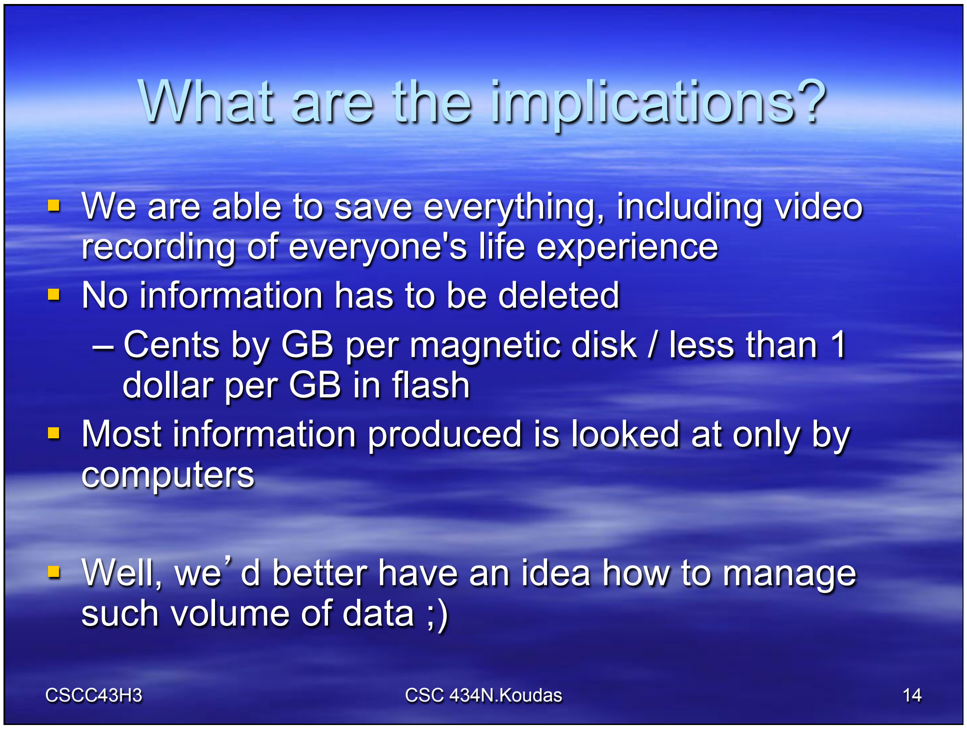
- Leyman, Varian et al estimated the worldwide production of information

Storage Medium	2002 Terabytes Upper Estimate	2002 Terabytes Lower Estimate	1999-2000 Upper Estimate	1999-2000 Lower Estimate	% Change Upper Estimates
Paper	1,634	327	1,200	240	36%
Film	420,254	76,69	431,690	58,209	-3%
Magnetic	5187130	3,416,230	2,779,760	2,073,760	87%
Optical	103	51	81	29	28%
TOTAL:	5,609,121	3,416,281	3,212,731	2,132,238	74.5%



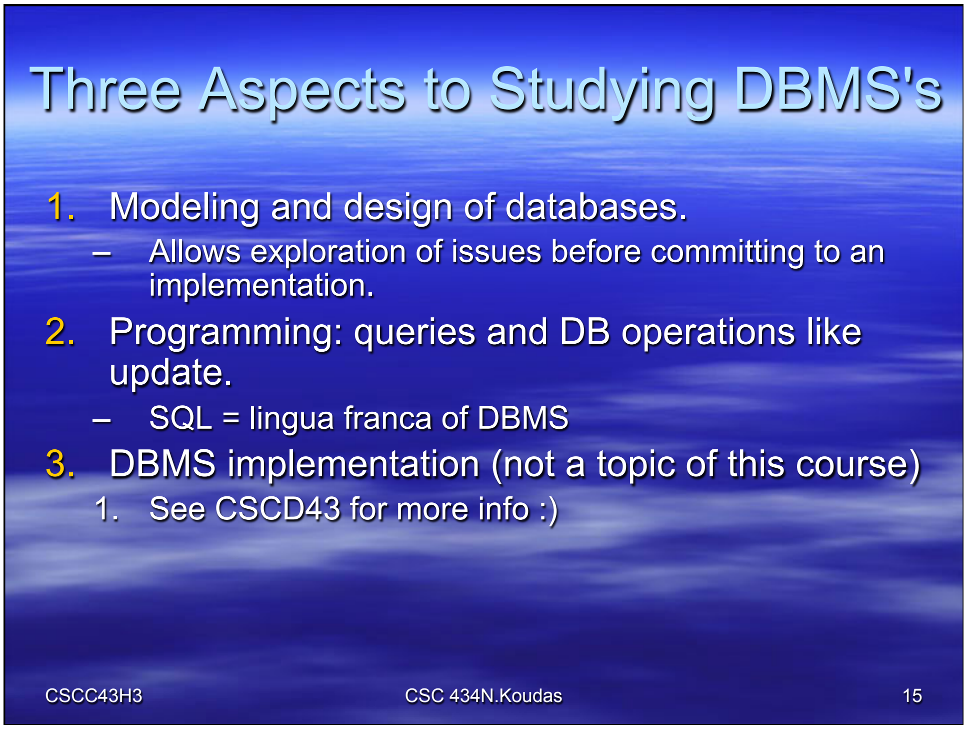
How much digital information exists? (2010)

- 1 ipad with 16G of memory
- 1TB = 62.5 fully loaded ipads
- 1PB = 62500 fully loaded ipads
- 1.2ZB = 75 billion fully loaded ipads
- The iStack (stack 75Billion ipads)
  - 339 miles up in the sky, with retail value of 37.4 Trillion dollars = 44% of worlds GDP ☺



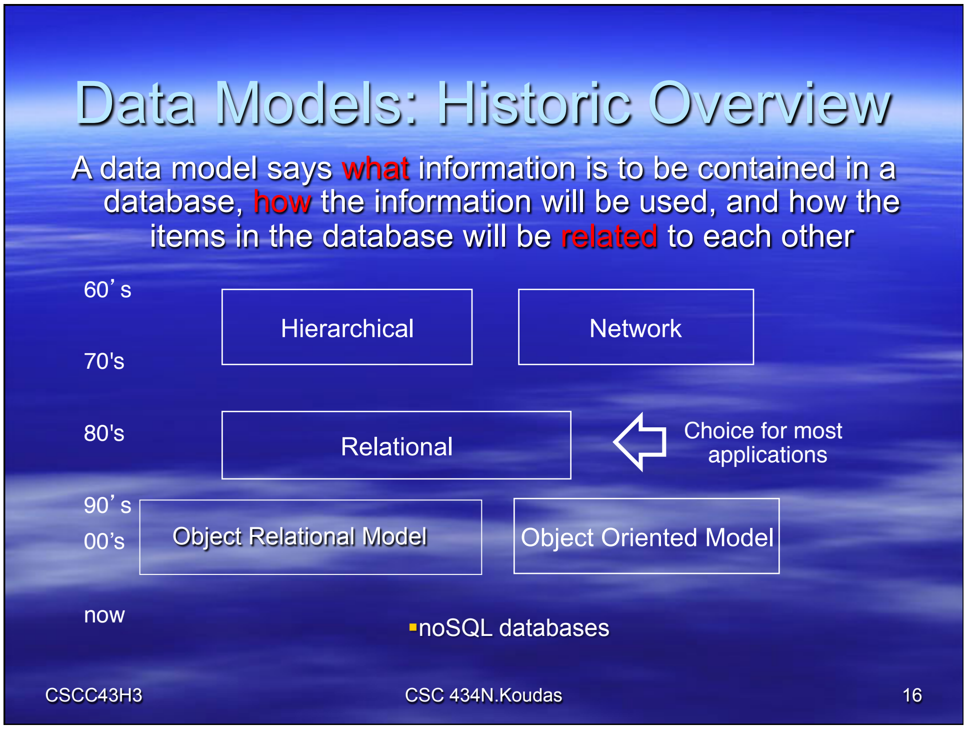
What are the implications?

- We are able to save everything, including video recording of everyone's life experience
- No information has to be deleted
  - Cents by GB per magnetic disk / less than 1 dollar per GB in flash
- Most information produced is looked at only by computers
- Well, we’d better have an idea how to manage such volume of data ;)



Three Aspects to Studying DBMS's

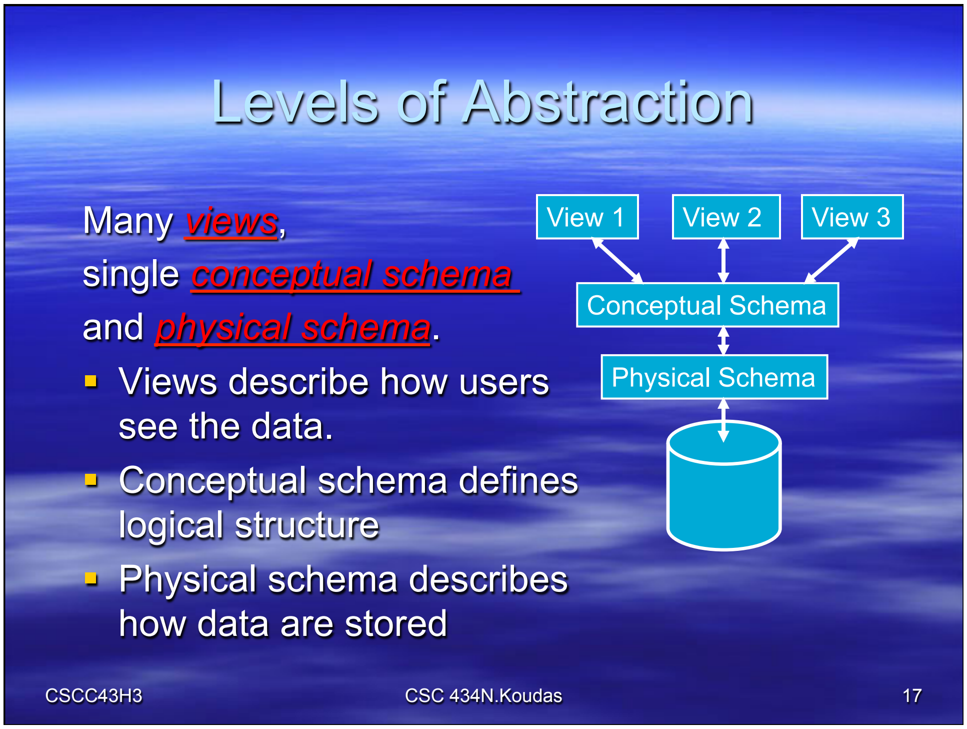
1. Modeling and design of databases.
   - Allows exploration of issues before committing to an implementation.
2. Programming: queries and DB operations like update.
   - SQL = lingua franca of DBMS
3. DBMS implementation (not a topic of this course)
   1. See CSCD43 for more info :)



Data Models: Historic Overview

A data model says what information is to be contained in a database, how the information will be used, and how the items in the database will be related to each other

- noSQL databases

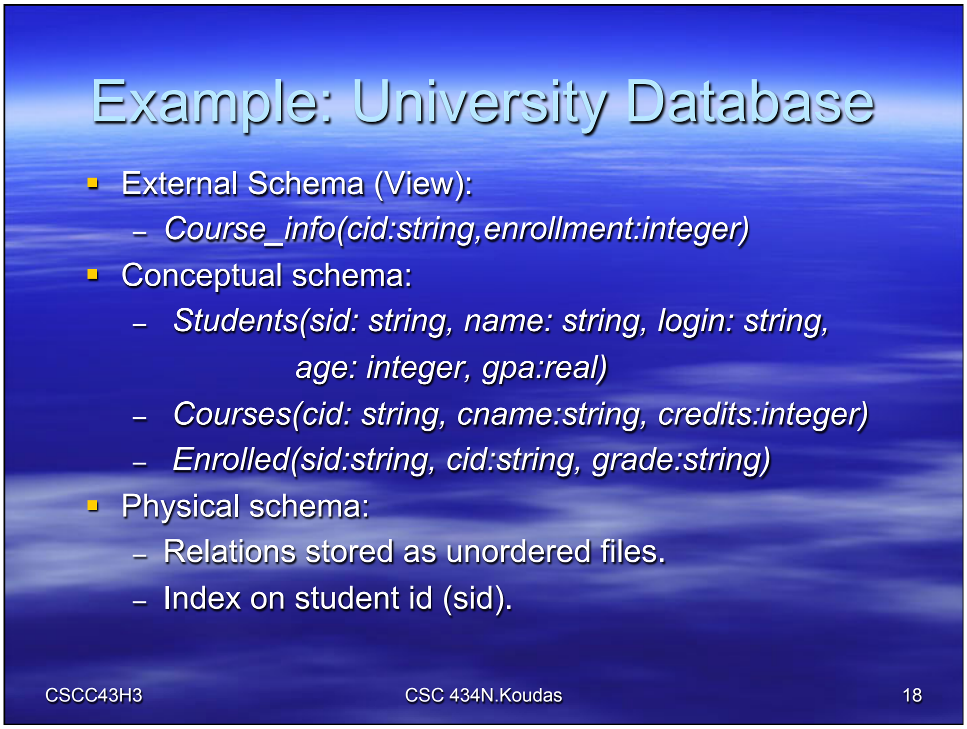


Levels of Abstraction

Many *views*,
single *conceptual schema*
and *physical schema*.

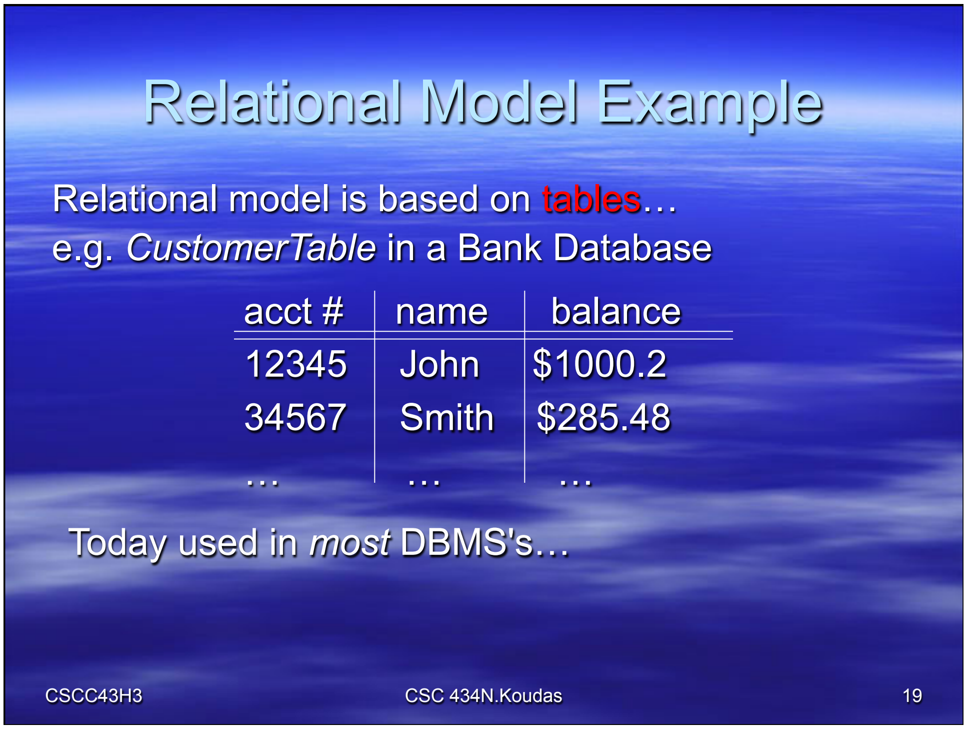
- Views describe how users see the data.
- Conceptual schema defines logical structure
- Physical schema describes how data are stored

View 1
View 2
View 3
Conceptual Schema
Physical Schema



Example: University Database

- External Schema (View):
  - *Course_info(cid:string,enrollment:integer)*
- Conceptual schema:
  - *Students(sid: string, name: string, login: string, age: integer, gpa:real)*
  - *Courses(cid: string, cname:string, credits:integer)*
  - *Enrolled(sid:string, cid:string, grade:string)*
- Physical schema:
  - Relations stored as unordered files.
  - Index on student id (sid).

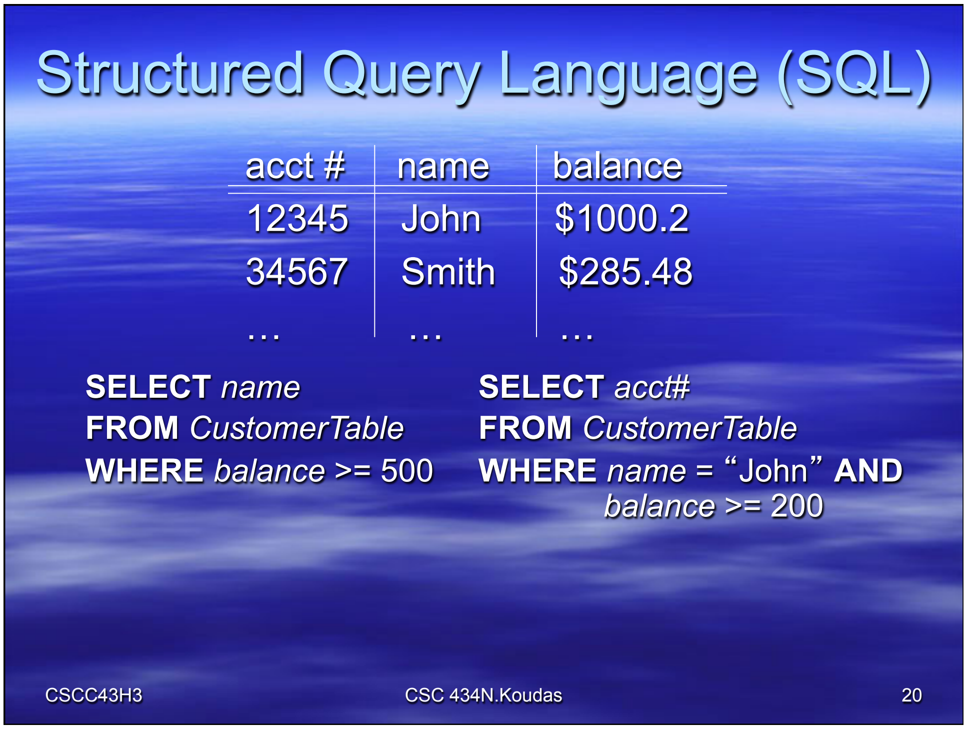


Relational Model Example

Relational model is based on tables…
e.g. *CustomerTable* in a Bank Database

acct #	name	balance
12345	John	$1000.2
34567	Smith	$285.48
…	…	…

Today used in *most* DBMS's…

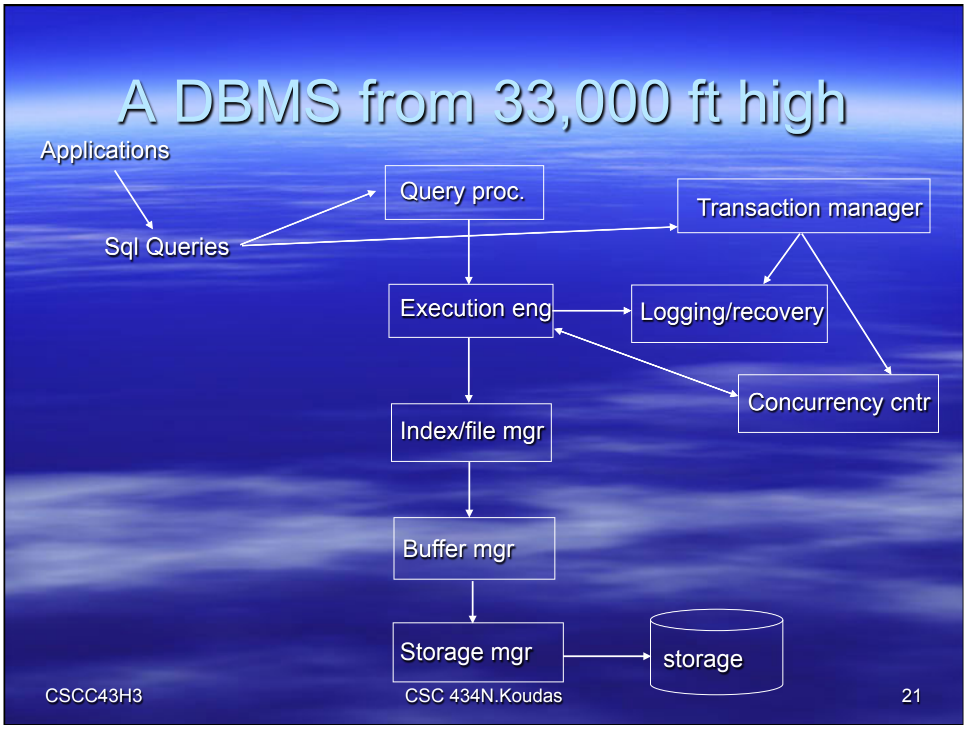


Structured Query Language (SQL)

acct #	name	balance
12345	John	$1000.2
34567	Smith	$285.48
…	…	…

SELECT *name*
FROM *CustomerTable*
WHERE *balance* >= 500

SELECT *acct#*
FROM *CustomerTable*
WHERE *name* = “John” **AND**
balance >= 200



A DBMS from 33,000 ft high

Applications

Sql Queries

Query proc.

Transaction manager

Execution eng

Logging/recovery

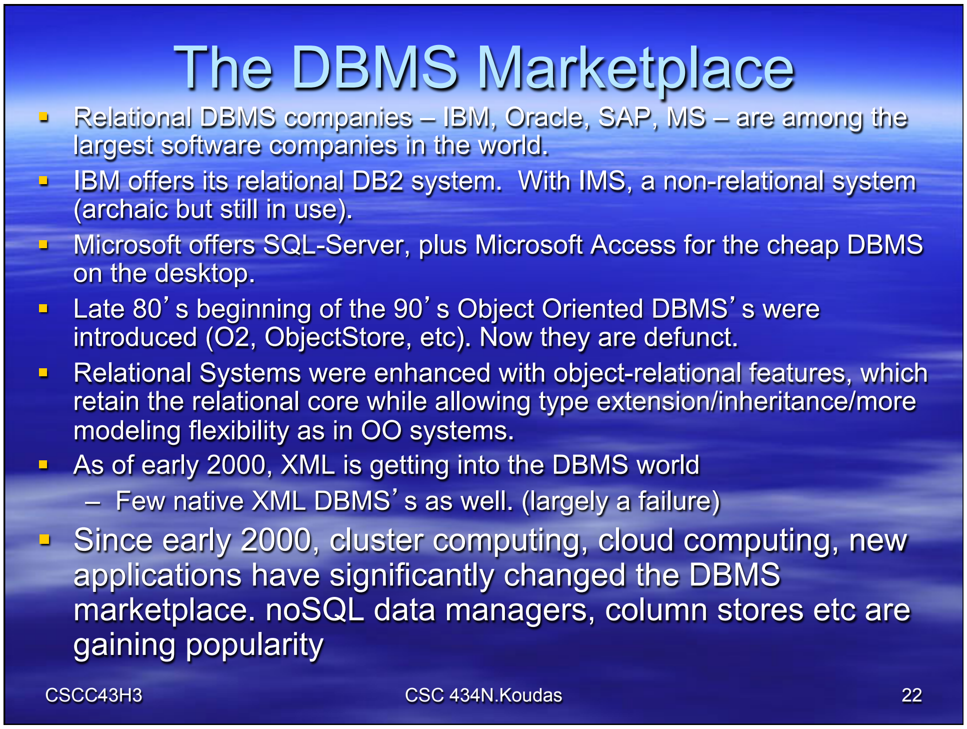
Concurrency cntr

Index/file mgr

Buffer mgr

Storage mgr

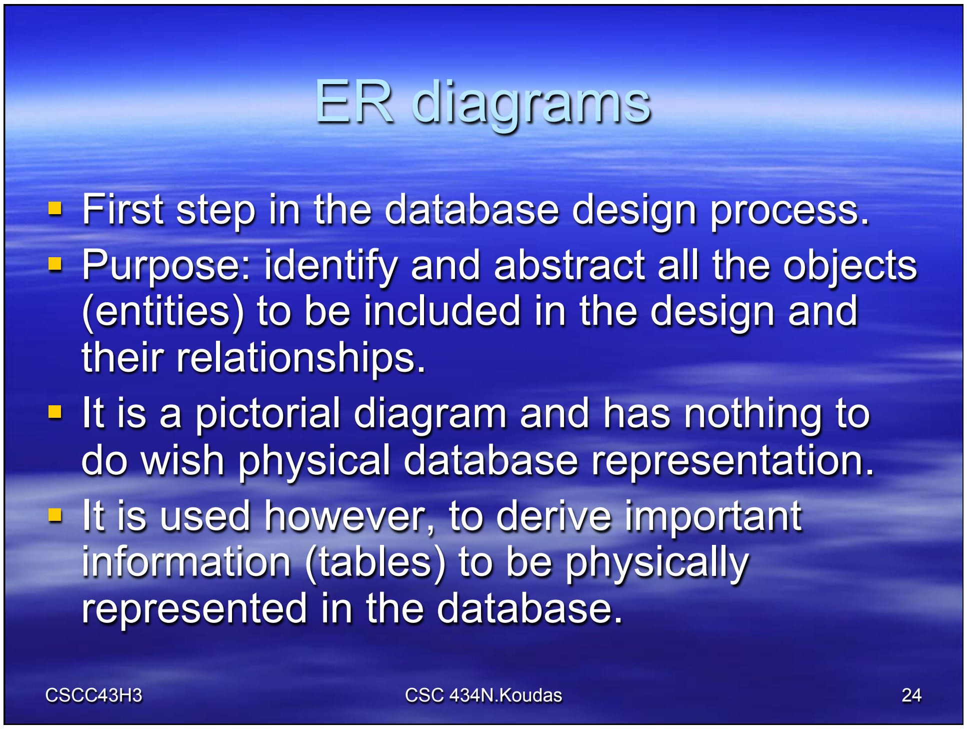
storage



The DBMS Marketplace

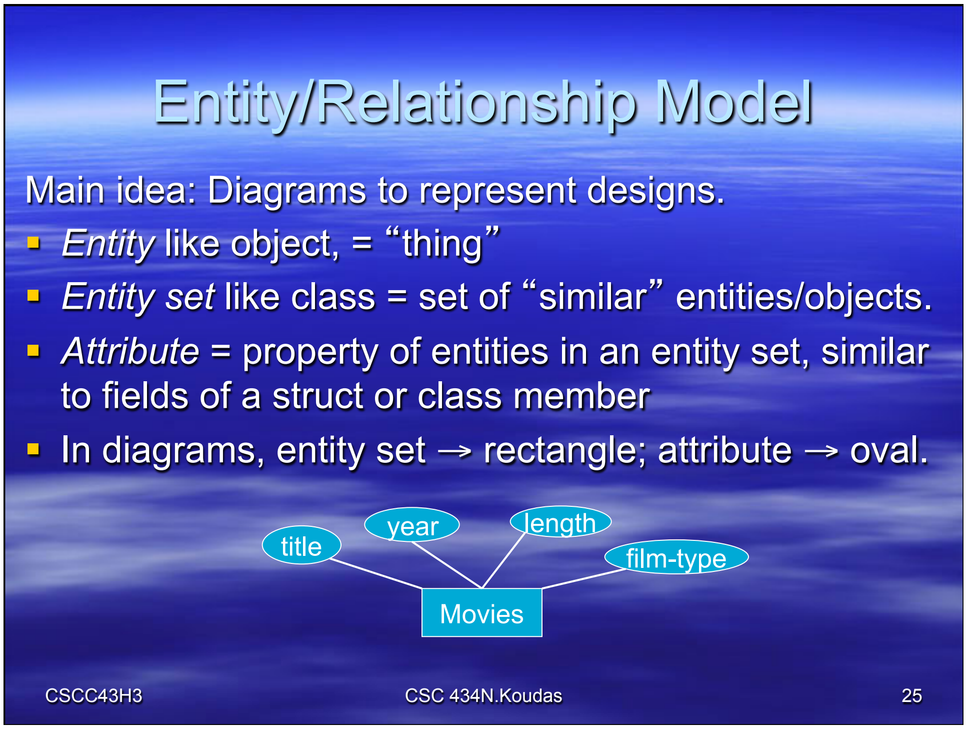
- Relational DBMS companies – IBM, Oracle, SAP, MS – are among the largest software companies in the world.
- IBM offers its relational DB2 system. With IMS, a non-relational system (archaic but still in use).
- Microsoft offers SQL-Server, plus Microsoft Access for the cheap DBMS on the desktop.
- Late 80's beginning of the 90's Object Oriented DBMS's were introduced (O2, ObjectStore, etc). Now they are defunct.
- Relational Systems were enhanced with object-relational features, which retain the relational core while allowing type extension/inheritance/more modeling flexibility as in OO systems.
- As of early 2000, XML is getting into the DBMS world
  - Few native XML DBMS's as well. (largely a failure)
- Since early 2000, cluster computing, cloud computing, new applications have significantly changed the DBMS marketplace. noSQL data managers, column stores etc are gaining popularity

The Entity Relationship Model



ER diagrams

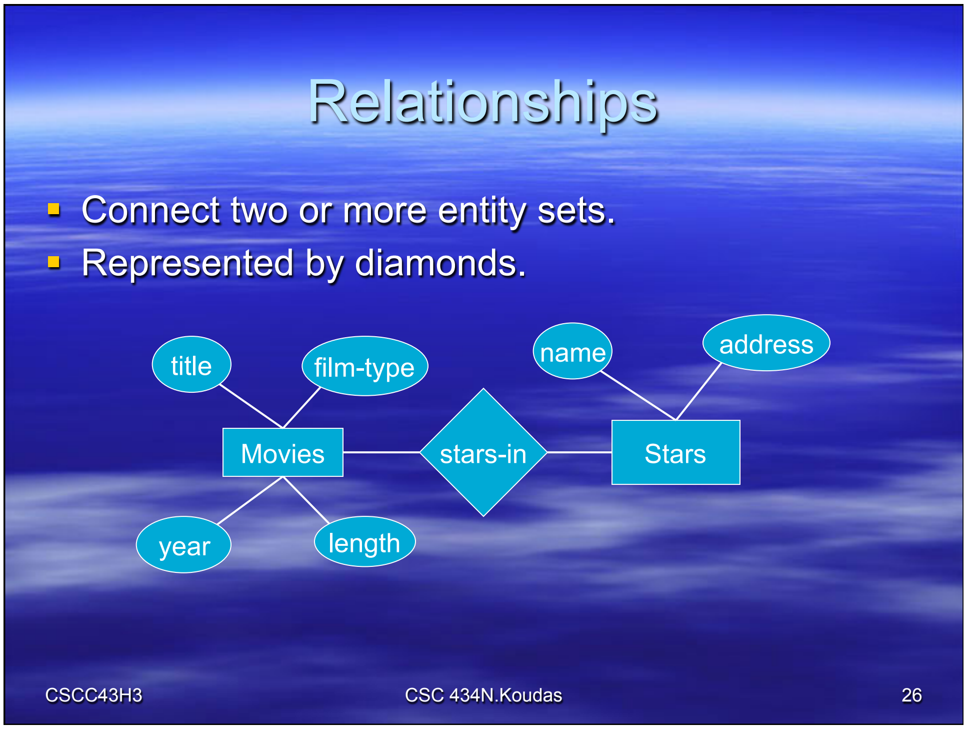
- First step in the database design process.
- Purpose: identify and abstract all the objects (entities) to be included in the design and their relationships.
- It is a pictorial diagram and has nothing to do wish physical database representation.
- It is used however, to derive important information (tables) to be physically represented in the database.



Entity/Relationship Model

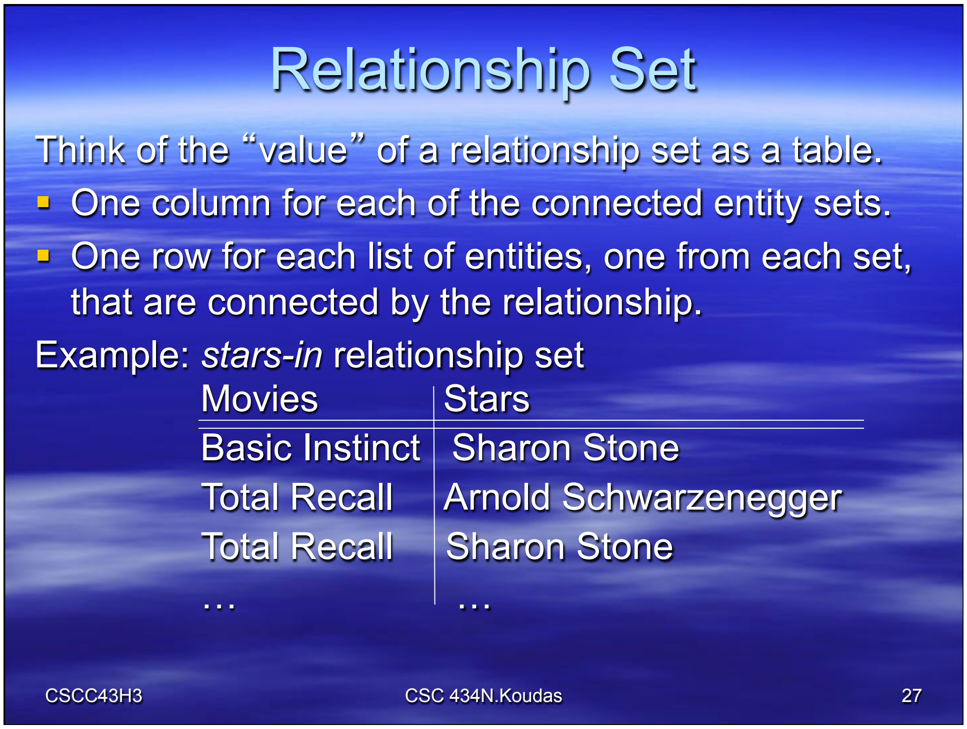
Main idea: Diagrams to represent designs.

- *Entity* like object, = “thing”
- *Entity set* like class = set of “similar” entities/objects.
- *Attribute* = property of entities in an entity set, similar to fields of a struct or class member
- In diagrams, entity set → rectangle; attribute → oval.



Relationships

- Connect two or more entity sets.
- Represented by diamonds.



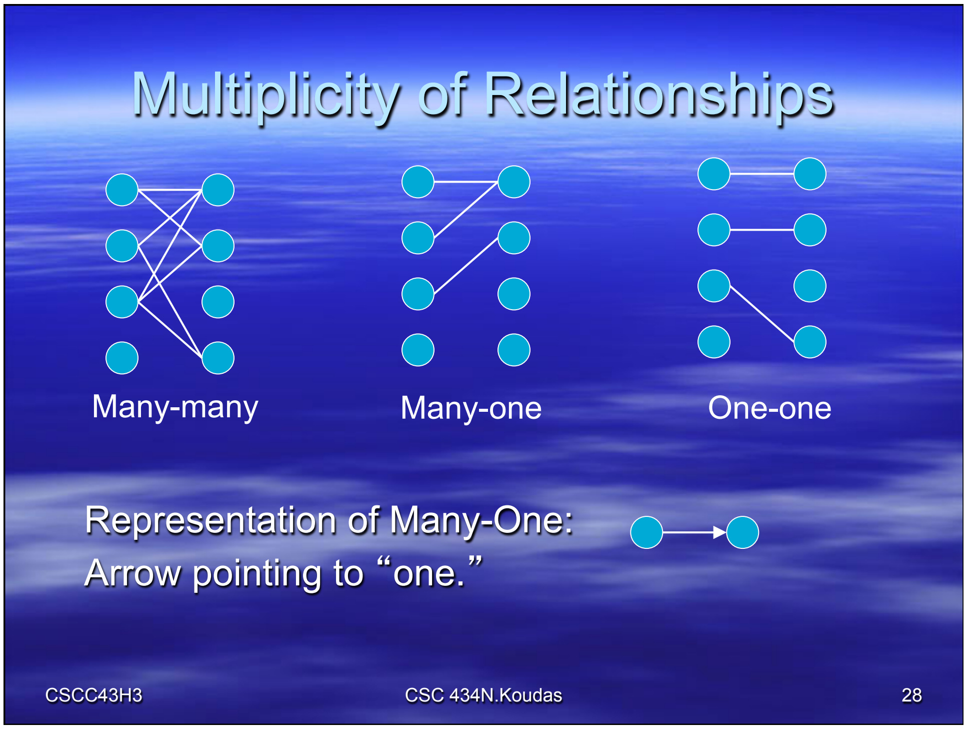
Relationship Set

Think of the "value" of a relationship set as a table.

- One column for each of the connected entity sets.
- One row for each list of entities, one from each set, that are connected by the relationship.

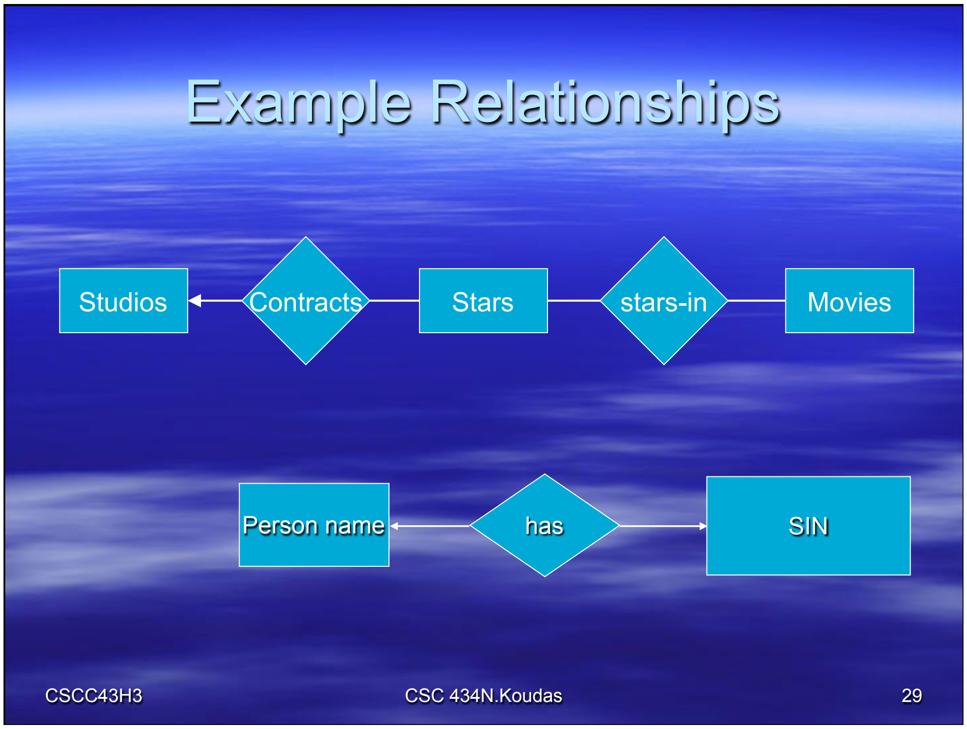
Example: *stars-in* relationship set

Movies	Stars
Basic Instinct	Sharon Stone
Total Recall	Arnold Schwarzenegger
Total Recall	Sharon Stone
…	…

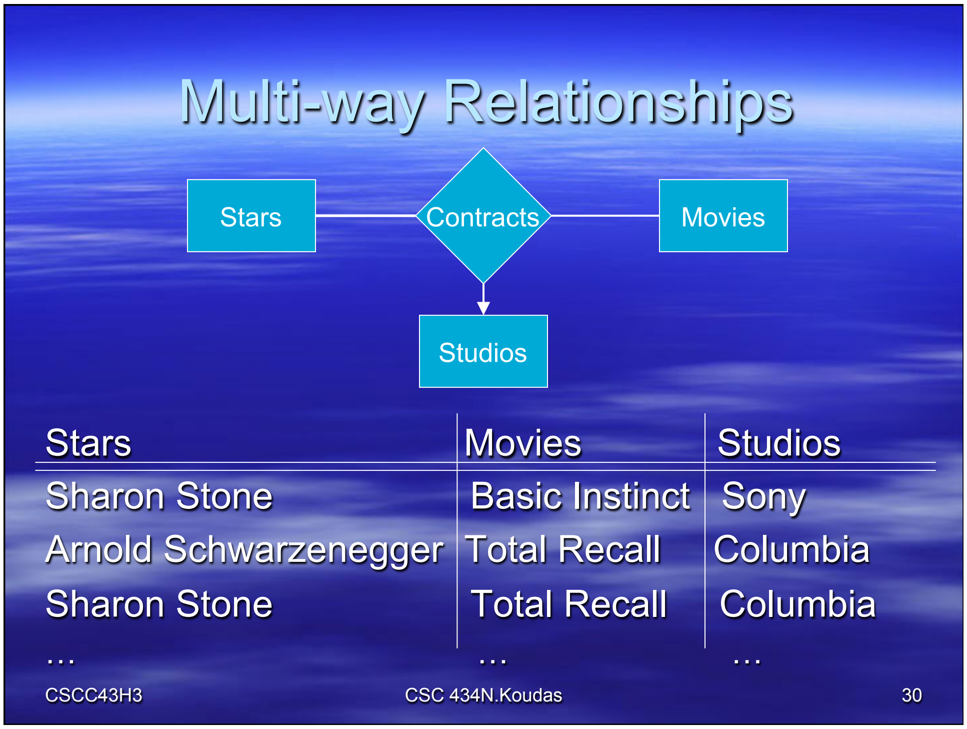


Multiplicity of Relationships

Representation of Many-One: Arrow pointing to "one."

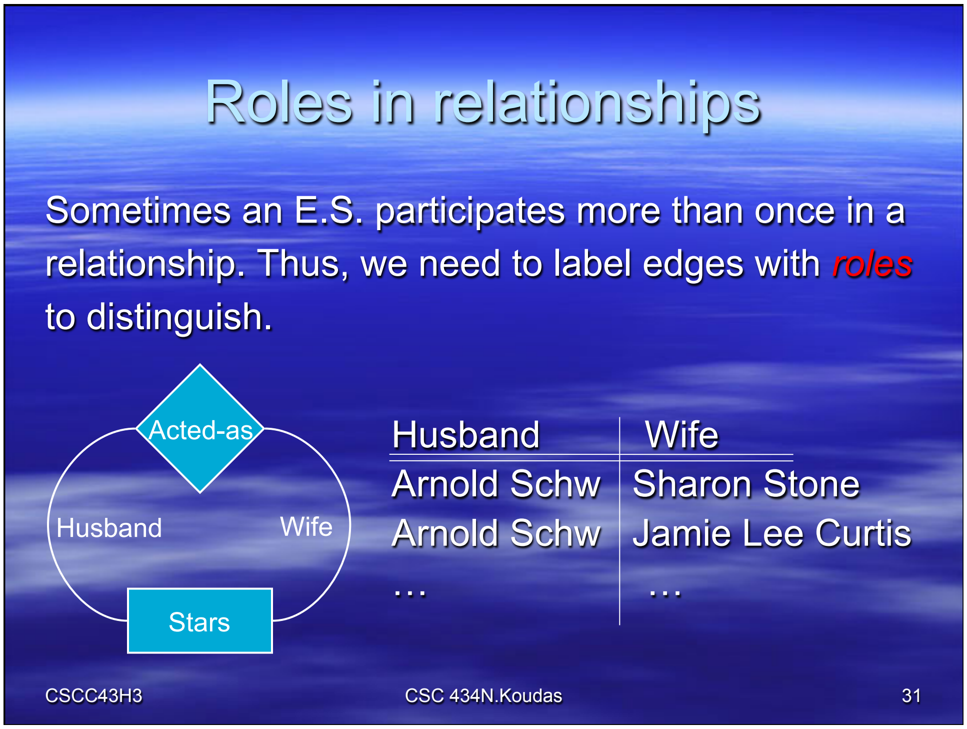


Example Relationships



Multi-way Relationships

Stars	Movies	Studios
Sharon Stone	Basic Instinct	Sony
Arnold Schwarzenegger	Total Recall	Columbia
Sharon Stone	Total Recall	Columbia
…	…	…

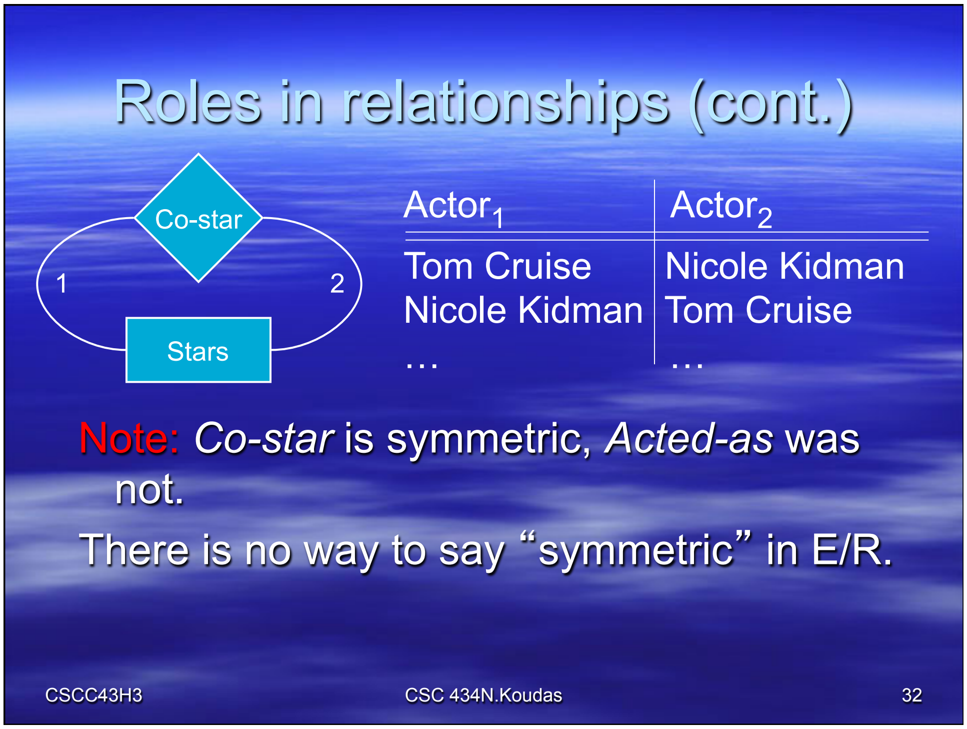


Roles in relationships

Sometimes an E.S. participates more than once in a relationship. Thus, we need to label edges with *roles* to distinguish.

Acted-as
Husband
Wife
Stars

Husband	Wife
Arnold Schw	Sharon Stone
Arnold Schw	Jamie Lee Curtis
…	…



Roles in relationships (cont.)

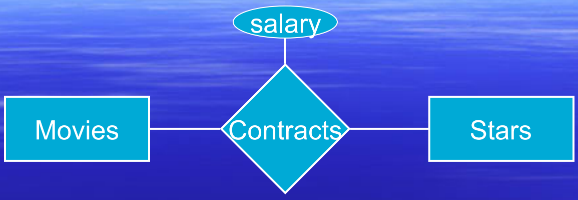
Co-star
1
2
Stars

Actor_1	Actor_2
Tom Cruise	Nicole Kidman
Nicole Kidman	Tom Cruise
…	…

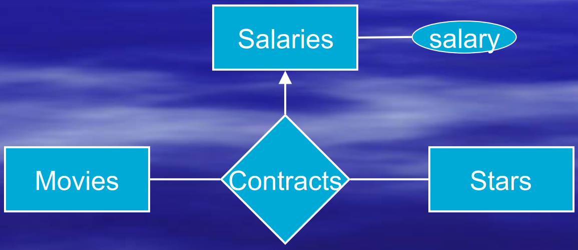
Note: *Co-star* is symmetric, *Acted-as* was not.

There is no way to say “symmetric” in E/R.

Attributes on Relationships



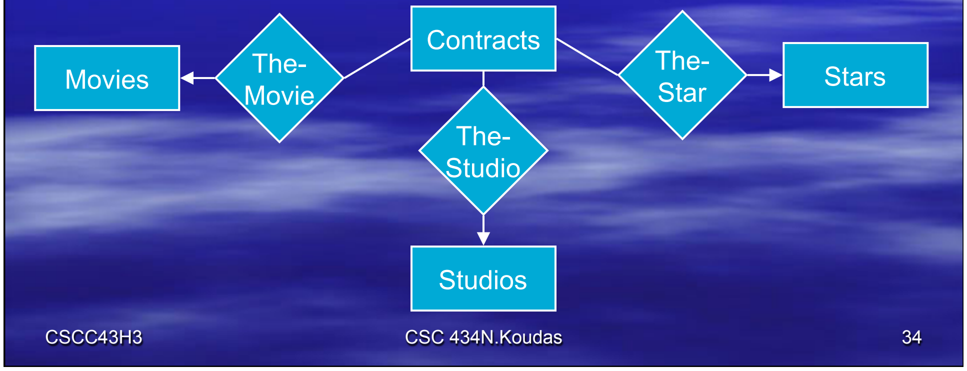
… a shorthand for the 3-way relationship:



Converting Multi-way to 2-Way

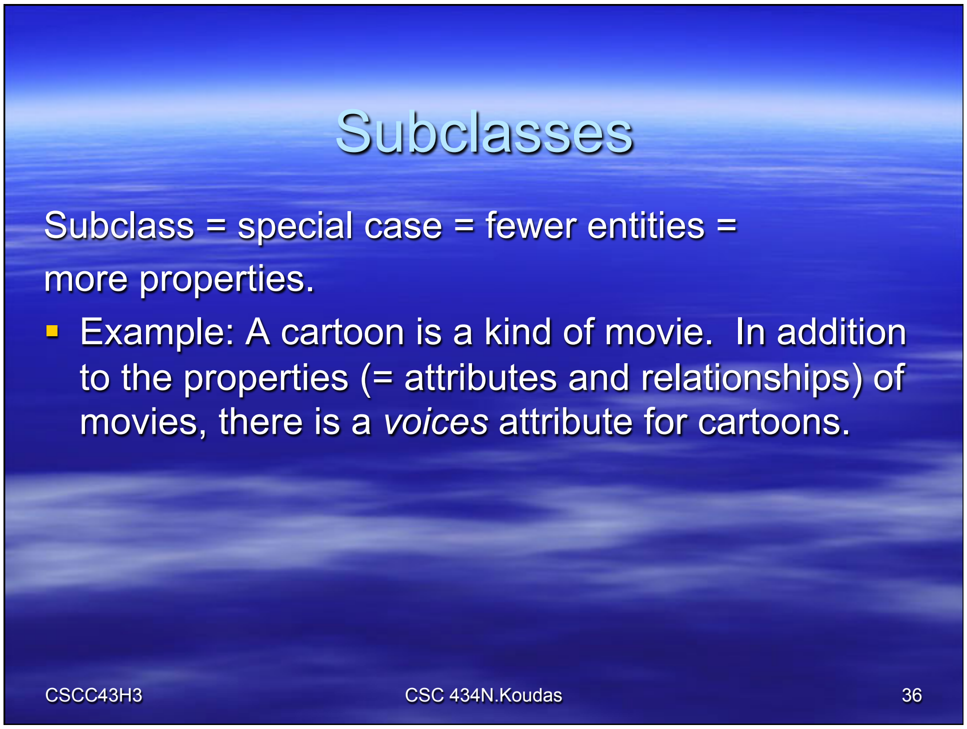
Baroque in E/R, but necessary in certain “object-oriented” models.

- Create a new E.S. to represent rows of a relationship set.
- Add many-one relationships from the new E.S. to the E.S.’s that participated in the original relationship.



More Design Issues

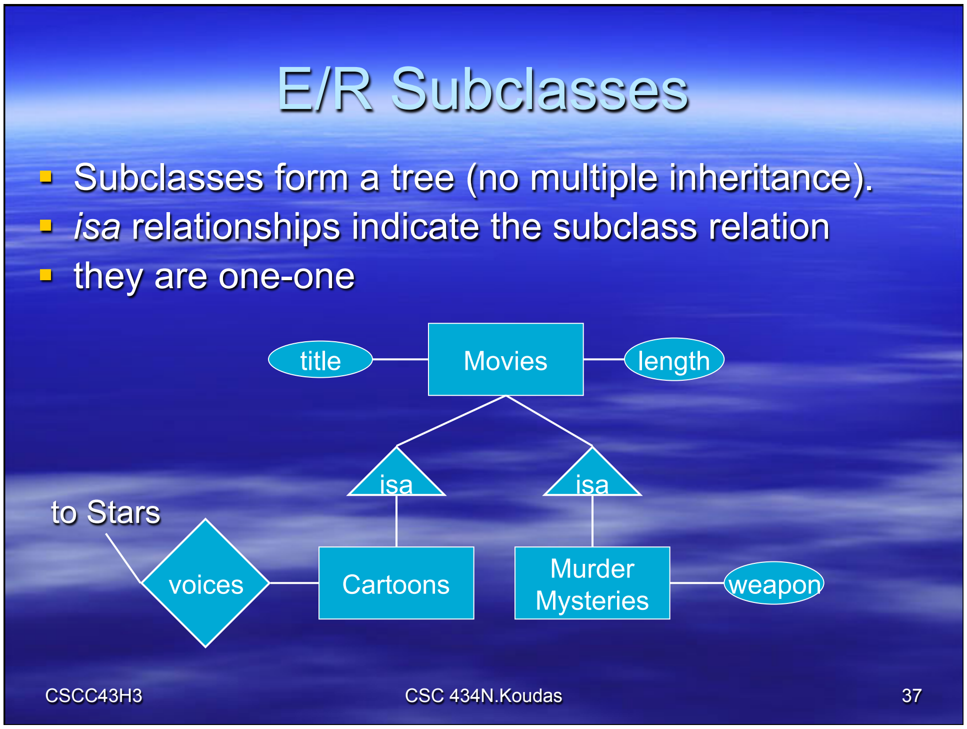
1. Subclasses.
2. Keys.
3. Weak entity sets.



Subclasses

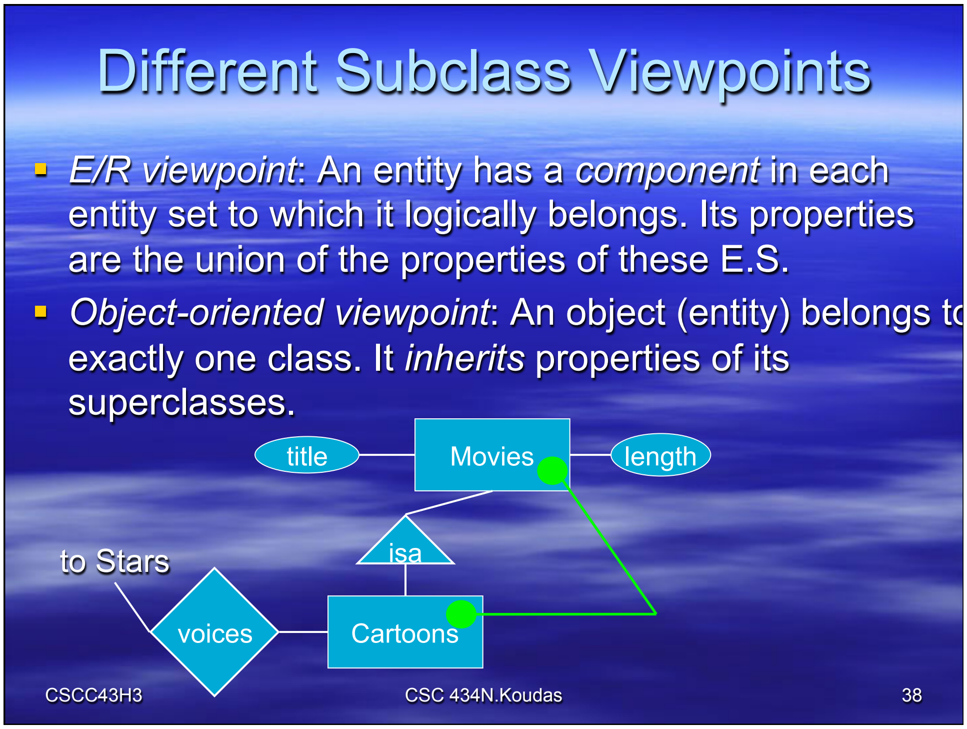
Subclass = special case = fewer entities = more properties.

- Example: A cartoon is a kind of movie. In addition to the properties (= attributes and relationships) of movies, there is a *voices* attribute for cartoons.



E/R Subclasses

- Subclasses form a tree (no multiple inheritance).
- *isa* relationships indicate the subclass relation
- they are one-one



Different Subclass Viewpoints

- *E/R viewpoint*: An entity has a *component* in each entity set to which it logically belongs. Its properties are the union of the properties of these E.S.
- *Object-oriented viewpoint*: An object (entity) belongs to exactly one class. It *inherits* properties of its superclasses.

That’s it for today…

CSCC43H3 CSC 434N.Koudas 39