Using Entity-Relationship Model
Data Model History

ER Definition Review

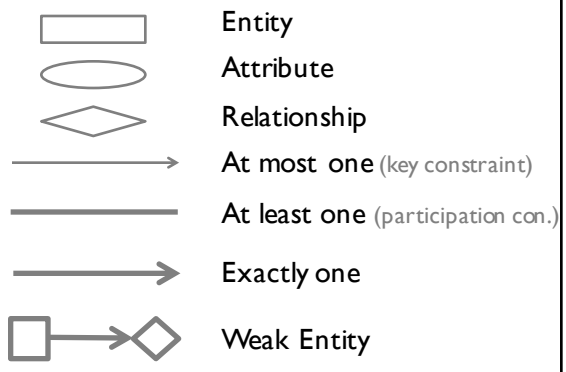
Entity: App "object" distinguishable from other objects
Attribute: Information that describes the entity
Attribute *domain*: Range of permissible values
  e.g. integers 1-20, 20 character strings, timestamp
Entity Set: Collection of entities with same attributes
Relationship: Association between entities
Relationship Set: Collection of similar relationships

Keys

Minimal set of attributes that uniquely identify an entity
May be multiple candidate keys
  e.g. User: both uid and email may be unique
May involve multiple attributes
  e.g. Class identified by both number and section
Primary key: designated unique identifier
Most entities have a key (except weak entities)
Diagram: Underlined

Weak Entity

Entity without a key: attributes are not unique
Identifying relationship distinguishes them
  e.g. Wall Post: User who posted it
  e.g. Song: Album where first released
Partial key: attributes that identify the weak entity, if you know the owning entity
  e.g. Wall Post: Timestamp attribute
  e.g. Song: Title Name attribute
Diagram: Dashed underline



Entity
Attribute
Relationship
At most one (key constraint)
At least one (participation con.)
Exactly one
Weak Entity

Class Hierarchy

Aggregation

You will practice these in HW1 & Project 1 part 1

Using the ER Model

Explore design choices for a concept

Entity or Attribute?
Entity or Relationship?
Binary or Ternary relationship?
Aggregation or Ternary relationship?

Entity or Attribute?

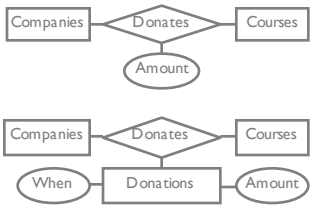
Is address an attribute of Users or an entity connected to Users by a relationship?

>1 instance of attribute: must be an entity
e.g. home and work addresses?

Attribute has structure, use entity:
e.g., search for users by city, state, or zip
alternative: use multiple attributes? DRY?

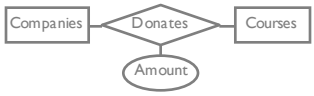
Entity or Attribute?

>1 instance of relationship attribute: Use entity
A company can't donate multiple amounts (top fig)
Use ternary relationship (bottom fig)



Entity or Relationship?

OK if company donates to courses individually
What if company donates once for a set of courses?
Redundancy of *amount*, need to remember to update every one
Misleading implies *amount* tied to *each* donation individually

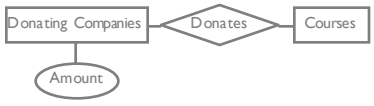


Company	Course	Amount
Amazon	4111	2000
Amazon	4112	2000
Amazon	5111	2000

These amounts are logically the same (redundant)!

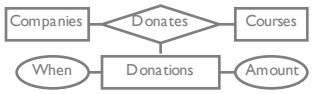
Entity or Relationship?

Add "Donating Company," move amount to attribute
Need ISA with Company: companies without donations



Entity or Relationship?

If company donates once to school for data related courses.
Refactor amount into an entity

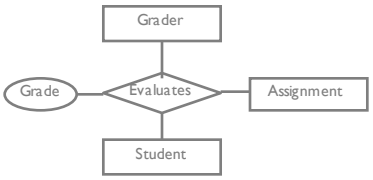


Company	Course	Donation
Amazon	4111	1
Amazon	4112	1
Amazon	5111	1

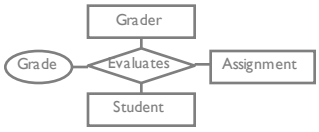
Donation	When	Amount
1	Today	2000

Binary or Ternary Relationship?

What if assignments have at most one grader?



Ternary relationship and constraints



Grader
Jane
Neha
…

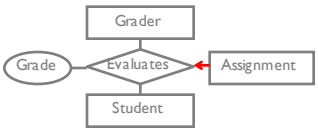
Student
Alice
Jinyang
Sarah
…

Assignment
Homework 0
Project 1
…

Part of class syllabus, *not* a specific submission

Assignment	Student	Grader	Grade
Homework 0	Jinyang	Jane	8
Homework 0	Alice	Jane	7
Homework 1	Jinyang	Neha	7
Homework 0	Sarah	Neha	6

Ternary relationship and constraints

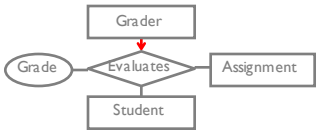


At most one grader per assignment?

HW0 can appear at most once! Also at most one student

Assignment	Student	Grader	Grade
Homework 0	Jinyang	Jane	8
~~Homework 0~~	~~Alice~~	~~Jane~~	~~7~~
Homework 1	Jinyang	Neha	7
~~Homework 0~~	~~Sarah~~	~~Neha~~	~~6~~

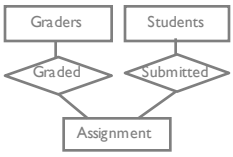
Ternary relationship and constraints



What does this say?

Assignment	Student	Grader	Grade
Homework 0	Jinyang	Jane	8
~~Homework 0~~	~~Alice~~	~~Jane~~	~~7~~
Homework 1	Jinyang	Neha	7
~~Homework 0~~	~~Sarah~~	~~Neha~~	~~6~~

Ternary relationship and constraints

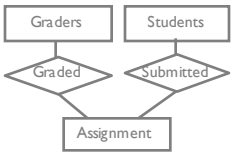


Assignment	Grader
Homework 0	Jane
Homework 1	Neha
Homework 0	Neha

Assignment	Student
Homework 0	Jinyang
Homework 0	Alice
Homework 1	Jinyang
Homework 0	Sarah

Assignment	Student	Grader	Grade
Homework 0	Jinyang	Jane	8
Homework 0	Alice	Jane	7
Homework 1	Jinyang	Neha	7
Homework 0	Sarah	Neha	6

Ternary relationship and constraints

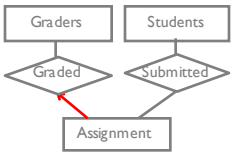


Assignment	Grader
Homework 0	Jane
Homework 1	Neha
Homework 0	Neha

At most one grader per assignment?

Assignment	Student
Homework 0	Jinyang
Homework 0	Alice
Homework 1	Jinyang
Homework 0	Sarah

Ternary relationship and constraints



Assignment	Grader
Homework 0	Jane
Homework 1	Neha
Homework 0	Neha

At most one grader per assignment?

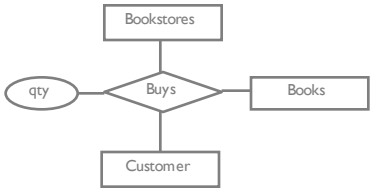
Students can only submit a given assignment once?

Not representable!

Assignment	Student
Homework 0	Jinyang
Homework 0	Alice
Homework 1	Jinyang
Homework 0	Sarah

Binary or Ternary Relationship?

Sometimes have true ternary relationship that is defined by all three entities.



Summary

Requirements
- what are you going to build?

Conceptual Database Design
- high-level description

(Today) ER Modeling

Logical Design
- formal database schema

Schema Refinement:
- fix potential problems, normalization

Physical Database Design
- use sample of queries to optimize for speed/storage

Summary

ER design is subjective based on usage and needs
- Today we saw multiple ways to model same idea

ER design is not complete/perfect
- Doesn't capture semantics (what does "instructor" *mean?*)
- Doesn't capture processes/state machines

ER design is a useful way of thought

Real World?

Visual, high-level tool to explore and explain
Many variants (all sorts of weird arrows)
Used, but not often

Important ideas:
- Go from high-level to details
- Relationships: Related data is very powerful
- Constraints: many-to-many, one-to-many

PPT/Illustrator/Keynote are simple ER diagramming tools.

The Relational Model

Background

Most widely used data model in the world

Legacy Models
IMS hierarchical
CODASYL network

"NoSQL": various recent flexible models

Key Principles

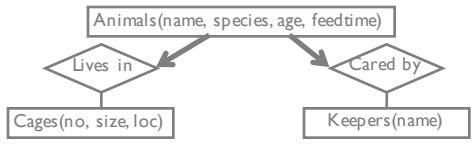
Data redundancy (or how to avoid it)
Physical data independence
- programs don't worry about physical structure

Logical data independence
- logical structure can change and legacy programs can continue to run

High level languages

Historical Context (not on test)

Hierarchical model (IMS)
Network model (CODASYL) — 70s
Relational model (SQL/QUEL) — 80-90s



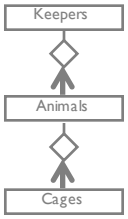
T/F: empty cage permitted? Animal with two keepers?

Hierarchical Model (IMS, circa 1968)

IBM Information Management System:
Apollo program for BOM SaturnV rocket
Segment types (objects / entity sets) with fields (attrs)
Segment instances (records)
Segment types form a tree

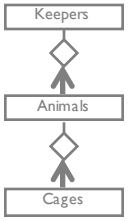
Hierarchical Model (IMS, circa 1968)

IBM Information Management System:
Apollo program for BOM SaturnV rocket
Segment types (objects / entity sets) with fields (attrs)
Segment instances (records)
Segment types form a tree
Sub-records must have "parent"



Hierarchical Model (IMS, circa 1968)

```
Jane (Keeper) (HSK 1)
   Bob, iguana, … (2)
      1, 100ft², … (3)
   Joe, student, … (4)
      1, 100ft², … (5)
…
```

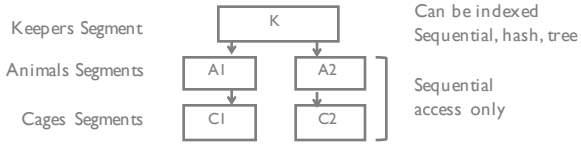


What's repeated?
Inconsistencies possible, lack of protections

Hierarchical Model Limitations

Physical Storage

Stored hierarchically
Only root segment can be indexed
Other segments only accessed sequentially



Hierarchical Querying: DL-1

Navigational Querying through a tree structure
Core operations
GX(seg, pred) general form, takes seg type and a predicate

Get Unique (GU) start at parent (root) segment
Get Next (GN) next record in HSK order in database
Get Next in Parent (GNP) next in HSK order until end of subtree

Fetch cages that Eugene entered

```
GU(Keeper, name = Eugene)
Until no more records
    cage = GNP(Cage)
    print cage.no
```

Problems

Duplicates data
Low level programming interface
Almost no physical data independence
- Change root from tree to hash index causes programs with GN on root to *fail*
- Inserts into sequential root structures disallowed

Lacks logical data independence
- Changing schema requires changing program

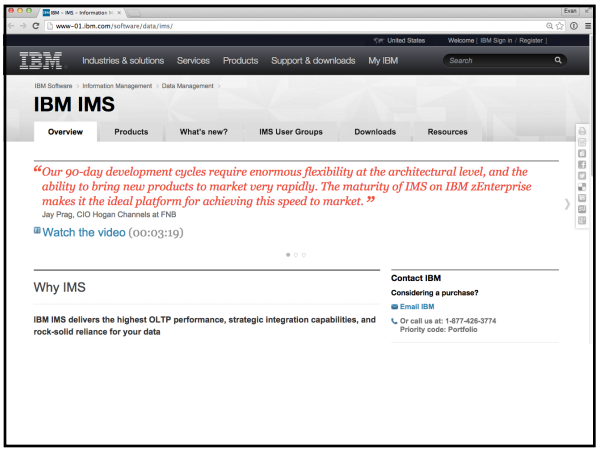
Violates many desirable properties of a proper DBMS

More Problems

Schema changes require program changes because pointers after GN calls now different

In reality, schemas change all the time

- Keepers now responsible for a whole cage
- Hummingbirds require multiple feedings
- Merge with another zoo



What is IMS?

- IMS = **I**nformation **M**anagement **S**ystem

- IMS is two products:
  - IMS Transaction Manager – IMS TM
    - Previously called IMS DC (Data Communications) or IMS TP (teleprocessing)
  - IMS Database Manager – IMS DB
    - Hierarchical Database Management System

- All products run on the mainframe (z/OS) only
- Proven track record of 46 years!!!

"Watts joined IBM in 1956 and worked at IBM's Silicon Valley development labs until his death on April 4, 2009.[2] *He had continuously worked on IMS since the 1960s*."

Wikipedia: https://en.wikipedia.org/wiki/IBM_Information_Management_System

IMS runs the world's most critical workloads

- 2000 customers worldwide run IMS
- 75% of the top 100 banks worldwide run IMS
- The top 5 US banks run IMS
- The top 5 European banks run IMS

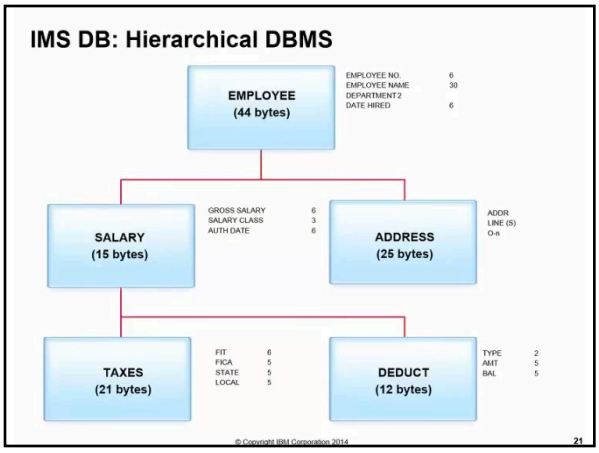
16 petabytes of production data managed by IMS

$3.0 trillion ($US) per day is transferred through IMS.....by one customer

300+ million users served every day

500 million accounts.....for one customer

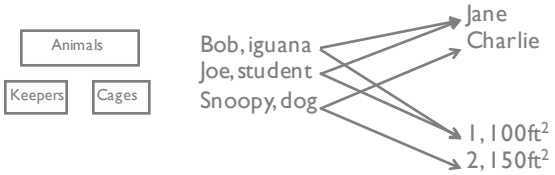
Plus ...



Network Models (CODASYL, 1969)

Abstraction

- Types of Records
- Connected by named sets (one to many relationships)
- Modeled as a *graph*: arbitrary connections



Network Models: Queries

Queries are programs that follow pointers

```
Find Keeper (name = 'Eugene')
until no more
   Find next Animal in cares_for
   Find Cage in lives_in
   Get current record
```

Very Smart people (Charles Bachman, '73 Turing Award) strongly defended this model but…

Network Models: Problems

Very complex due to navigational programming

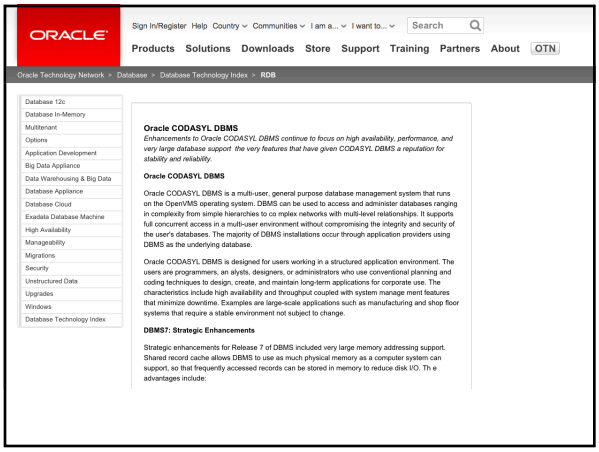
- Need to track where you are in a graph

Still no physical nor logical data independence

Implementations were limiting

Must load all data at once

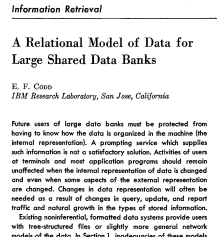
Trades off increased programmer pain for modeling non-hierarchical data



Relational Model (1970)

Ted Codd, 1970
Reaction to IMS maintenance cost

Key properties:
1. simple representation: table
2. set oriented model
3. no physical data model needed



Information Retrieval

A Relational Model of Data for Large Shared Data Banks

E. F. Codd
IBM Research Laboratory, San Jose, California

Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation). A prompting service which supplies such information is not a satisfactory solution. Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed and even when some aspects of the external representation are changed. Changes in data representation will often be needed as a result of changes in query, update, and report traffic and natural growth in the types of stored information.

Existing noninferential, formatted data systems provide users with tree-structured files or slightly more general network models of the data. In Section 1, inadequacies of these models

Optional Reading

What Goes Around Comes Around
Stonebraker and Hellerstein
Overview of the history of different data models

Michael Stonebraker Turing Lecture
History of Ingres/PostgreSQL