

Using Entity-Relationship Model Data Model History

Announcements

Project I

- Select a team
- Part I Step I before meeting staff
- Meet staff this week!
- Useful to read the all parts (rest up “soon”)

Waiting list:

- 10 enrolled without HW0
- Will finalize Jan 27th (tomorrow) at 8 PM

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, for a given owning entity

e.g. Wall Post: Timestamp attribute

e.g. Song: Title Name attribute

Diagram: Dashed underline



Entity



Attribute



Relationship



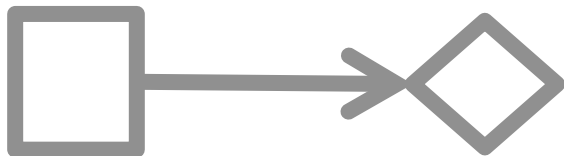
At most one (key constraint)



At least one (participation con.)



Exactly one



Weak Entity



Class Hierarchy



Aggregation

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

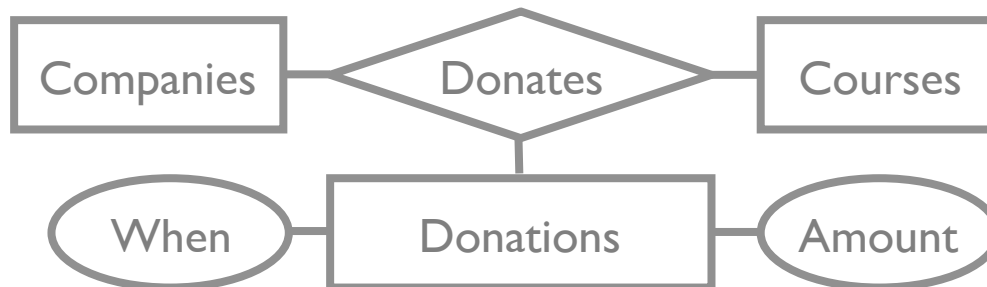
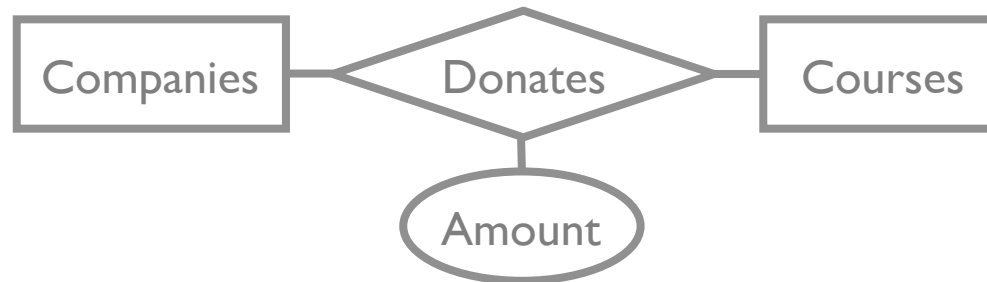
alternative: use multiple attributes? DRY?

Entity or Attribute?

> 1 instance of relationship attribute: Use entity

A company can't donate multiple amounts (top fig)

Use ternary relationship (bottom fig)



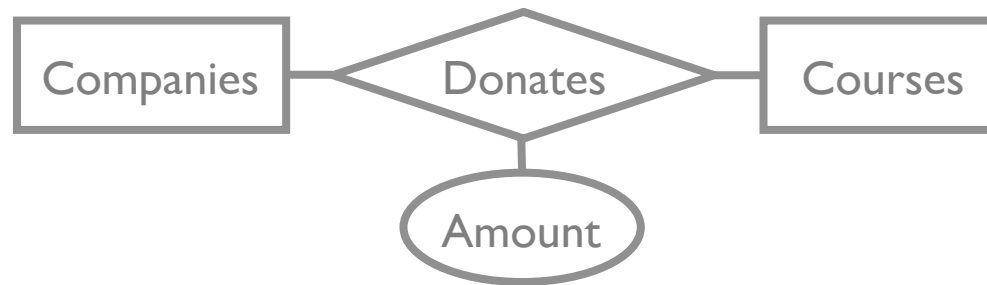
Entity or Relationship?

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

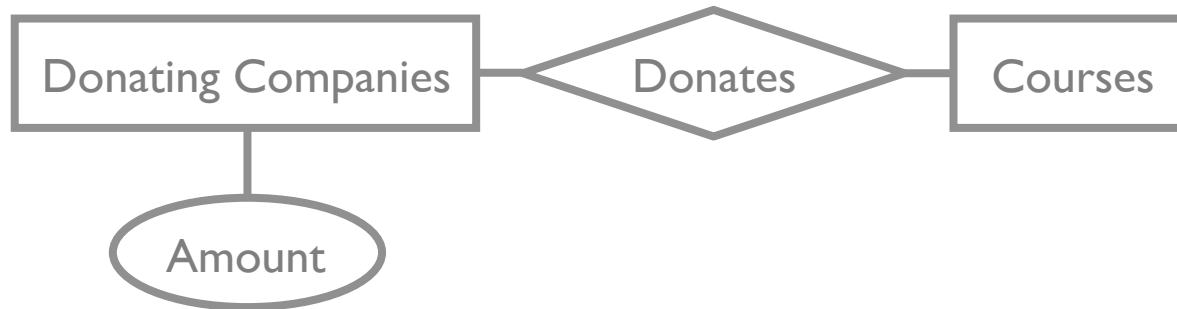


| Company | Course | Amount |
|---------|--------|--------|
| Amazon | 4111 | 2000 |
| Amazon | 4112 | 2000 |
| Amazon | 5111 | 2000 |

} These amounts are logically the same (redundant)!

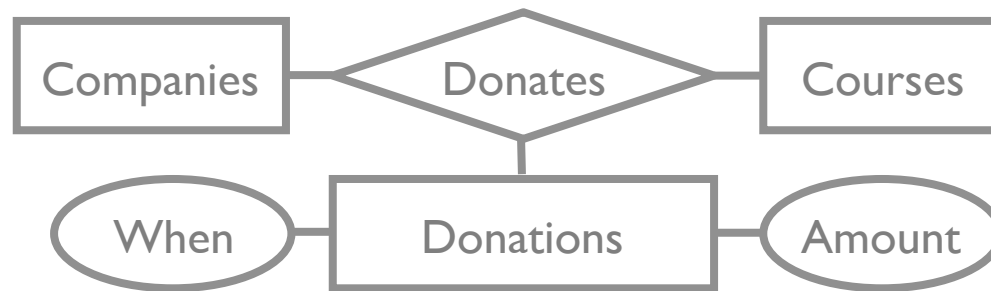
Entity or Relationship?

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

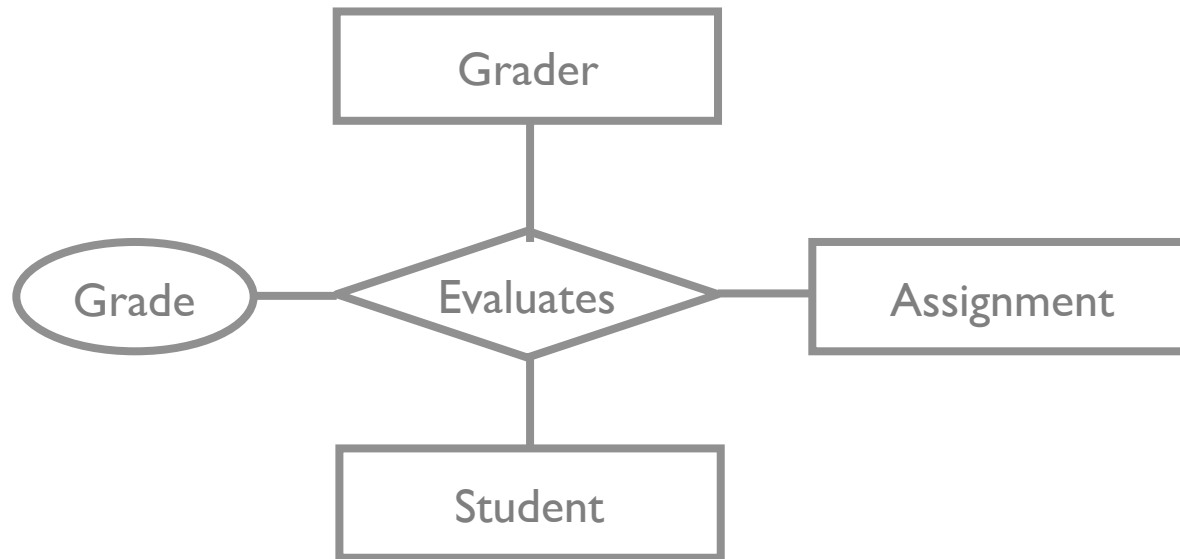


| Company | Course | Donation |
|---------|--------|----------|
| Amazon | 4111 | 1 |
| Amazon | 4112 | 1 |
| Amazon | 5111 | 1 |

| Donation | When | Amount |
|----------|-------|--------|
| 1 | Today | 2000 |

Binary or Ternary Relationship?

What if assignments have at most one grader?

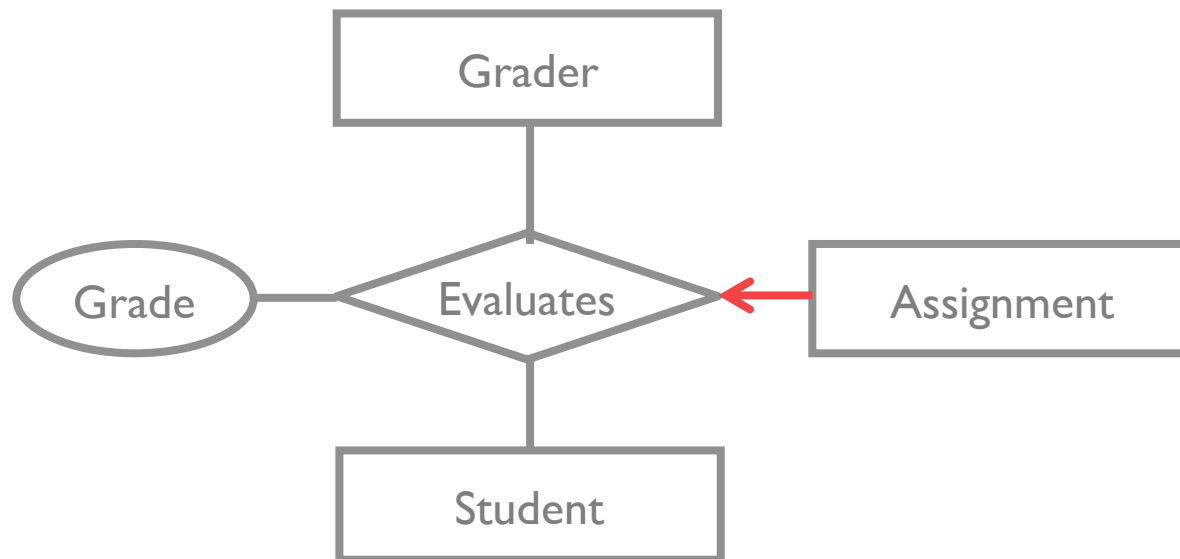


Binary or Ternary Relationship?

What if assignments have at most one grader?

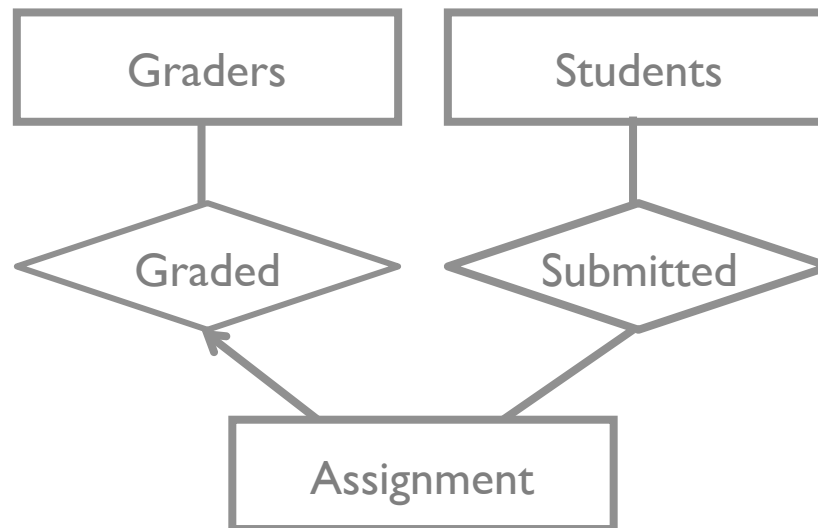
Only one student can complete HW0!

Actually two separate relationships

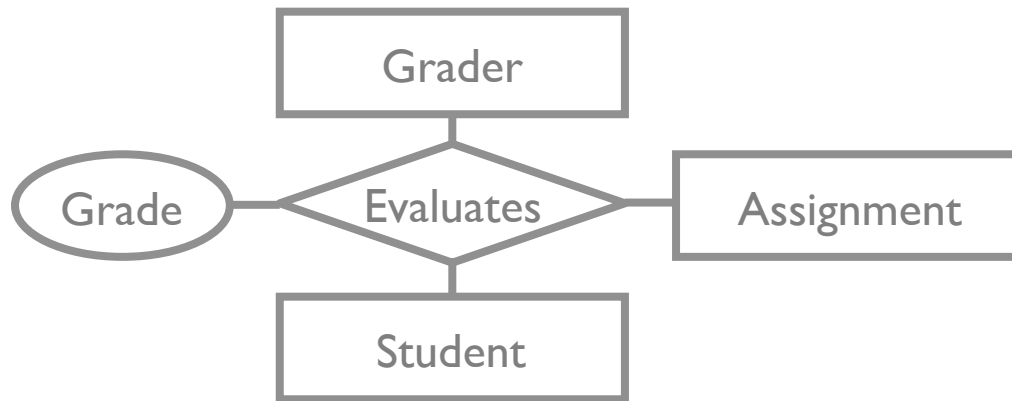


Binary or Ternary Relationship?

Binary relationships allows additional constraints



Ternary relationship and constraints



Grader

Jane

Neha

...

Student

Alice

Jinyang

Sarah

...

Assignment

Homework 0

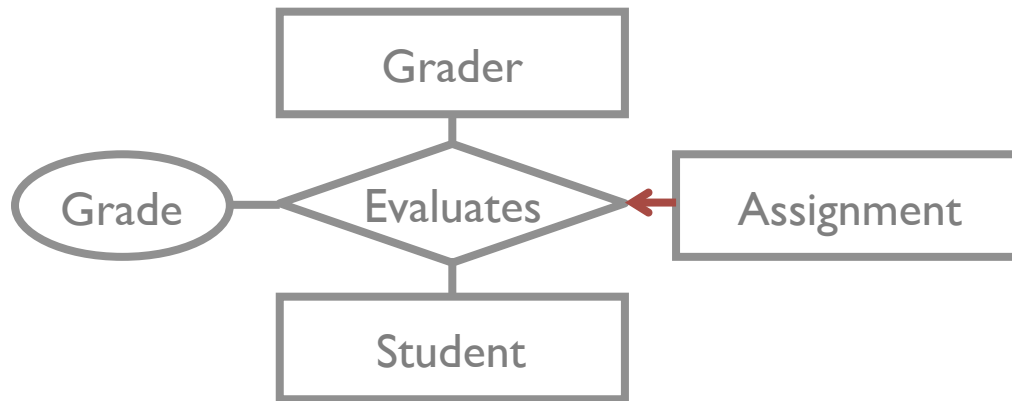
Project I

...

Part of class
syllabus, *not* a
specific
submission

| Assignment | Student | Grader | Grade |
|------------|---------|--------|-------|
| Homework 0 | Jinyang | Jane | 8 |
| Homework 0 | Alice | Jane | 7 |
| Homework I | Jinyang | Neha | 7 |
| Homework 0 | Sarah | Neha | 6 |

Ternary relationship and constraints

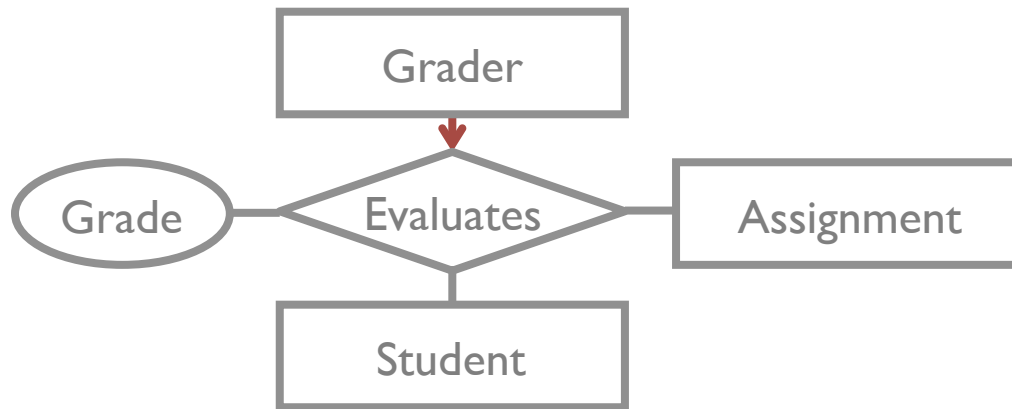


At most one grader per assignment?

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 |

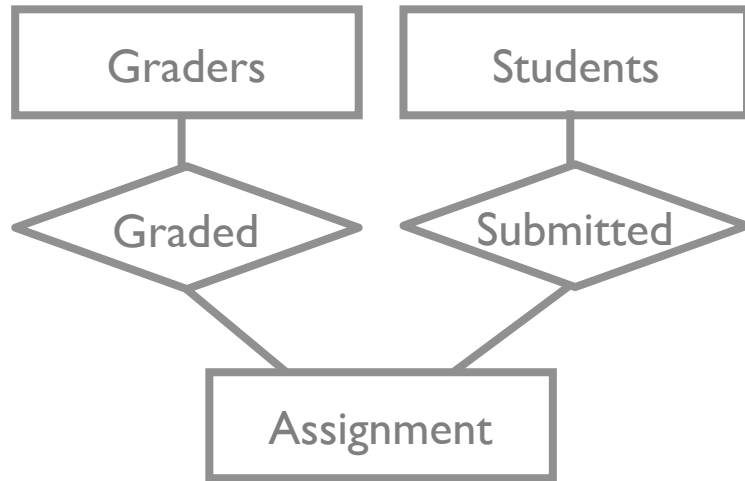
Ternary relationship and constraints



At most one grader per assignment?

| 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

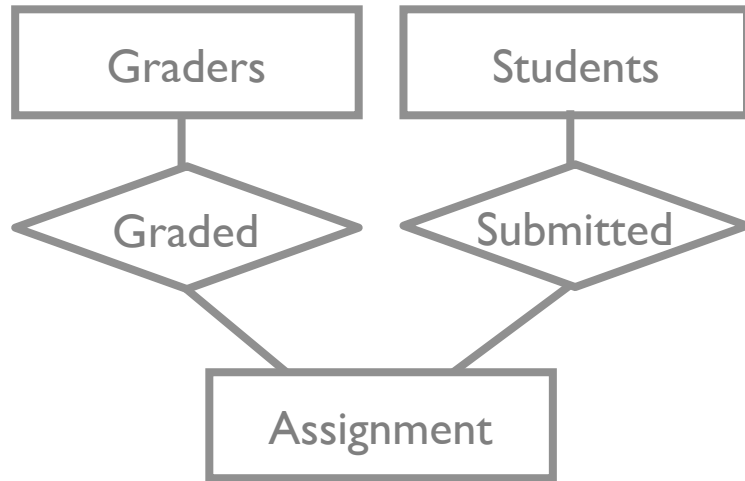


| Assignment | Grader |
|------------|--------|
| Homework 0 | Jane |
| Homework 1 | Neha |
| Homework 0 | Neha |

| 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

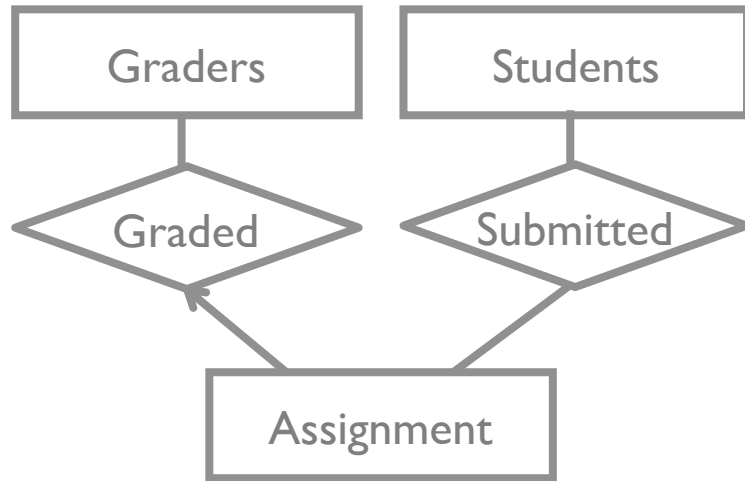


At most one grader per assignment?

| 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 |

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!

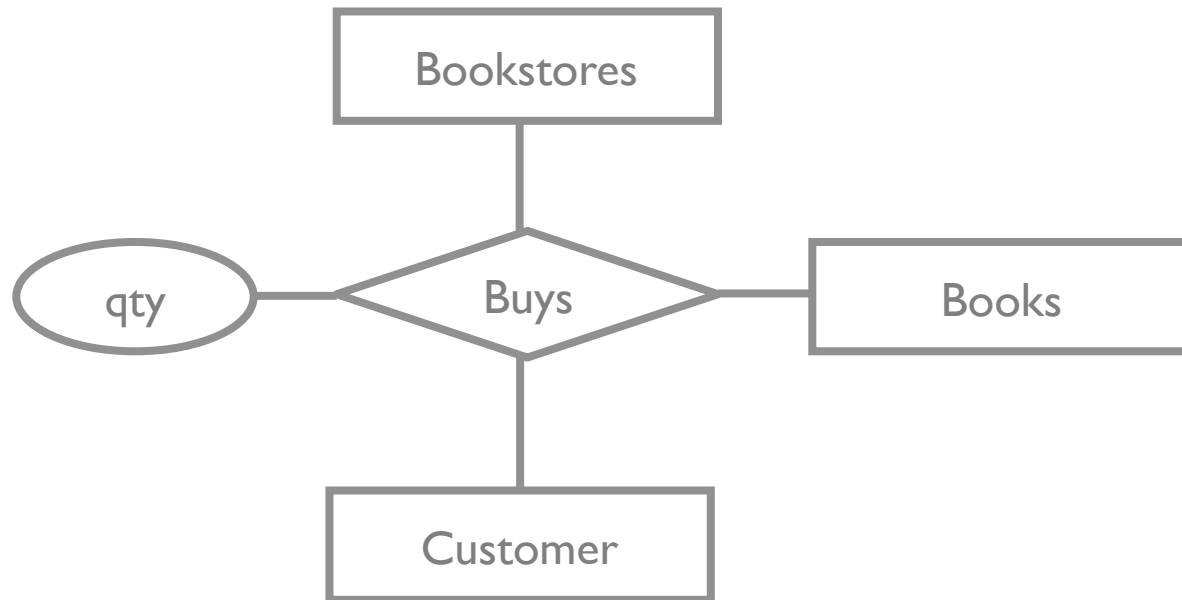
Relationships are sets

No weak entities here

| 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

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

The Relational Model



Neil Conway @neil_conway · 20m

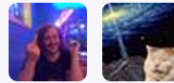
Dear twitter folks, what's the most interesting paper you read recently?

RETWEET

1

FAVORITE

1



4:44 PM - 14 Sep 2015 · Details



Reply to @neil_conway



Saurabh Sharan @saurabhsharan · 15m

@neil_conway "What Goes Around Comes Around" by Stonebraker & Hellerstein
- good overview of different data models mitpress.mit.edu/sites/default/...



★ 1



Background

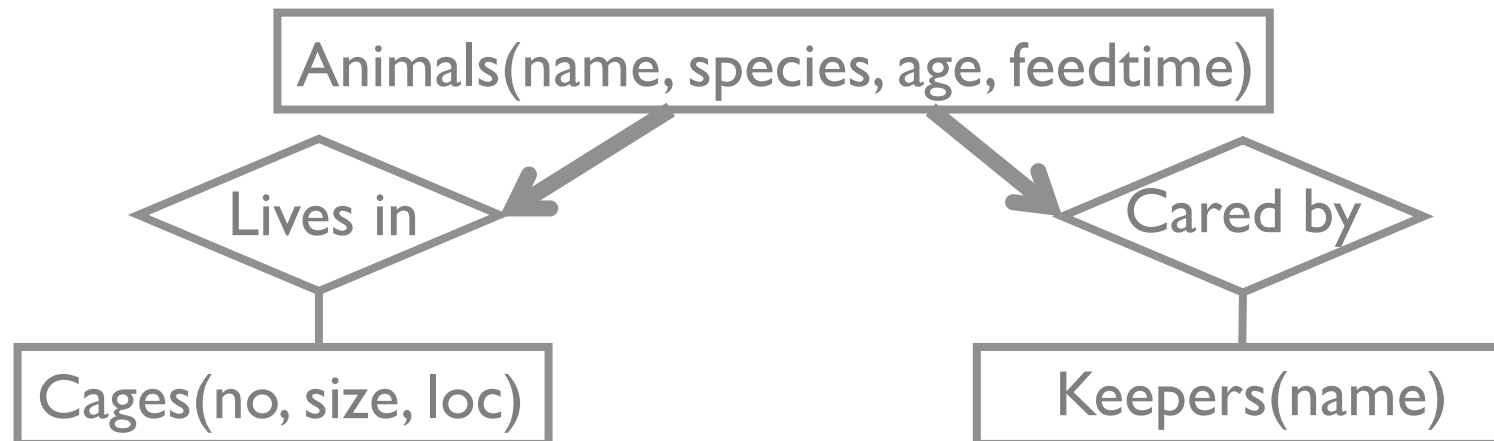
- Most widely used data model in the world
- Legacy Models
 - IMS hierarchical
 - CODASYL network
- “NoSQL”: various recent flexible models

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)
 - Relational model (SQL/QUEL)
- 70s
- 80-90s



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

Hierarchical Model (IMS, circa 1968)

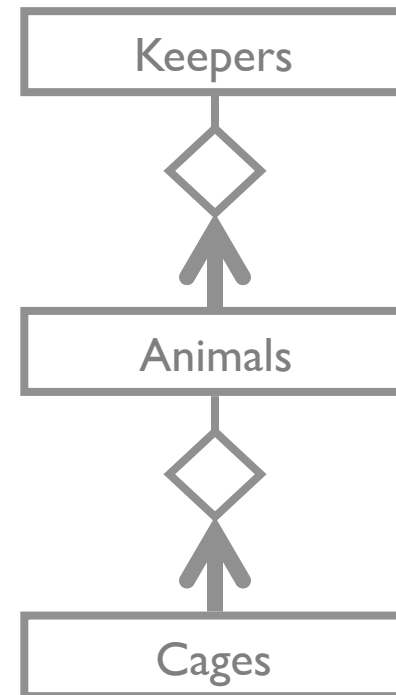
- IBM Information Management System:
- **Apollo program for BOM Saturn V rocket**
- Segment types (objects / entity sets) with fields (attrs)
- Segment instances (records)
- Segment types form a tree





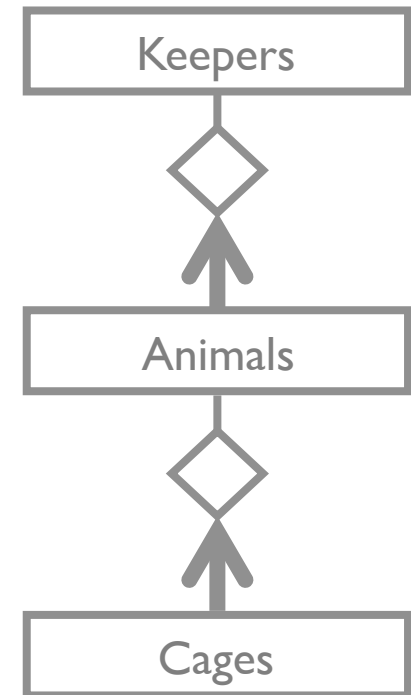
Hierarchical Model (IMS, circa 1968)

- IBM Information Management System:
- **Apollo program for BOM Saturn V rocket**
- Segment types (entity sets) with fields (attributes)
- Segment instances (entities)
- 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

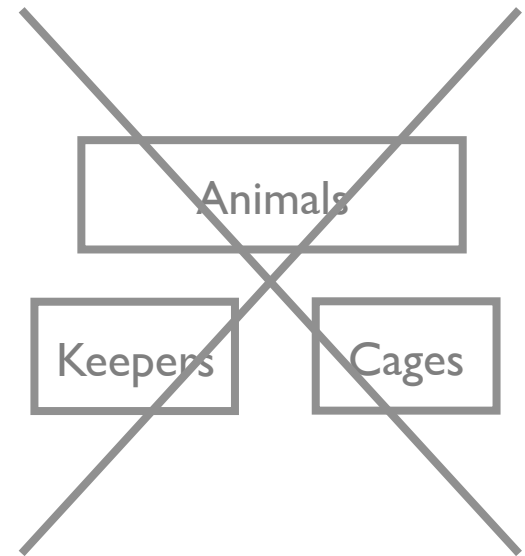
Hierarchical Model Limitations



Repeats cage data
(>1 animals in a cage)



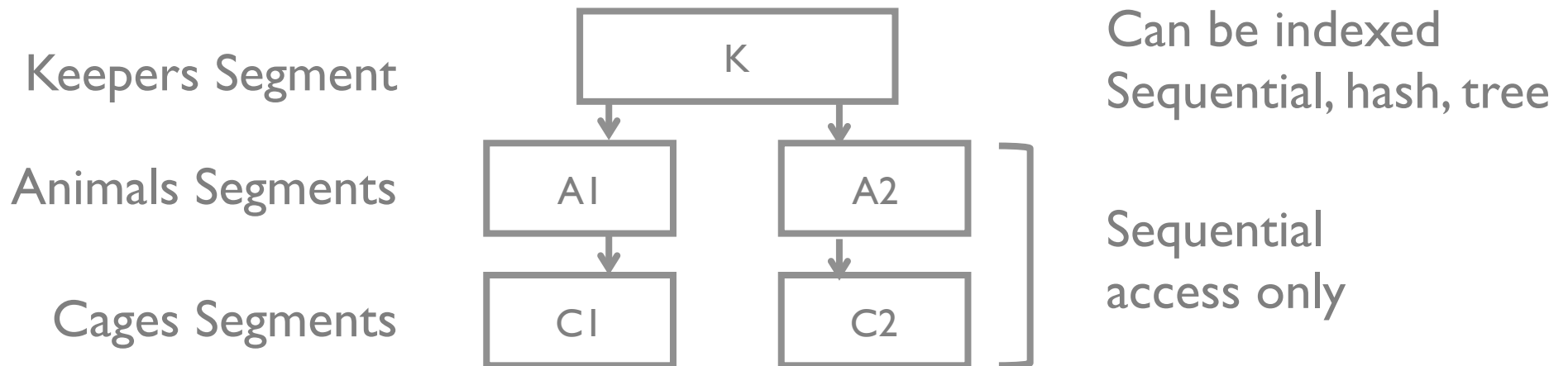
Repeats keeper data
(>1 animals/keeper)



Disallowed

Physical Storage

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



Hierarchical Querying: DL-I

- *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

IBM – IMS – Information M: x

www-01.ibm.com/software/data/ims/

United StatesWelcome [IBM Sign in / Register]

IBMIndustries & solutionsServicesProductsSupport & downloadsMy IBMSearch

IBM Software > Information Management > Data Management >

IBM IMS

OverviewProductsWhat's new?IMS User GroupsDownloadsResources

“Our 90-day development cycles require enormous flexibility at the architectural level, and the ability to bring new products to market very rapidly. The maturity of IMS on IBM zEnterprise makes it the ideal platform for achieving this speed to market.”

Jay Prag, CIO Hogan Channels at FNB

Watch the video (00:03:19)

Why IMS

IBM IMS delivers the highest OLTP performance, strategic integration capabilities, and rock-solid reliance for your data

Contact IBM

Considering a purchase?

Email IBM

Or call us at: 1-877-426-3774
Priority code: Portfolio

PrintEmailFacebookTwitterLinkedInYouTube

facebook

[Sign Up](#)

Email or Phone

Password

[Log In](#)☐ Keep me logged in[Forgot your password?](#)[Create Page](#)**Recent**

2015

2014

2013

2012

2011

2010

2009

2008

1968

**IBM IMS**
Software**IBM IMS**
is on Facebook.

To connect with IBM IMS, sign up for Facebook today.

[Sign Up](#)[Log In](#)http://linkd.in/IBM_IMS[Twitter.com/IBM_IMS](https://twitter.com/IBM_IMS)[Youtube.com/RethinkIMS](https://www.youtube.com/RethinkIMS)**Timeline**[About](#)[Photos](#)[Likes](#)[More ▾](#)**PEOPLE****1,265 likes****ABOUT**

The official IBM IMS page, managed by Tori Gonzalez and Sandy Sherrill and follows the IBM Social Computing Guidelines.

<http://www.ibm.com/ims>

**IBM IMS**

January 22 at 4:32pm ·

It's time for the IMS Technical Symposium in beautiful Wiesbaden, Germany! Yes, you should come!

Are you an IBM z Systems Mainframe technical professional with responsibility for IMS?

Calling all IMS DBAs and System Programmers, Enterprise and Application Architects who integrate with and connect to IMS! Sharpen your competitive edge by joining the IMS Development team in

LINKEDIN.COM

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 ...

© Copyright IBM Corporation 2014

Major US
Insurance
Company

Handelsbanken



CREDIT SUISSE



CATERPILLAR®

Scotiabank

IT für Banken



FIDUCIA
Ihr IT-Partner

bbk

SIGNAL IDUNA

UniCredit

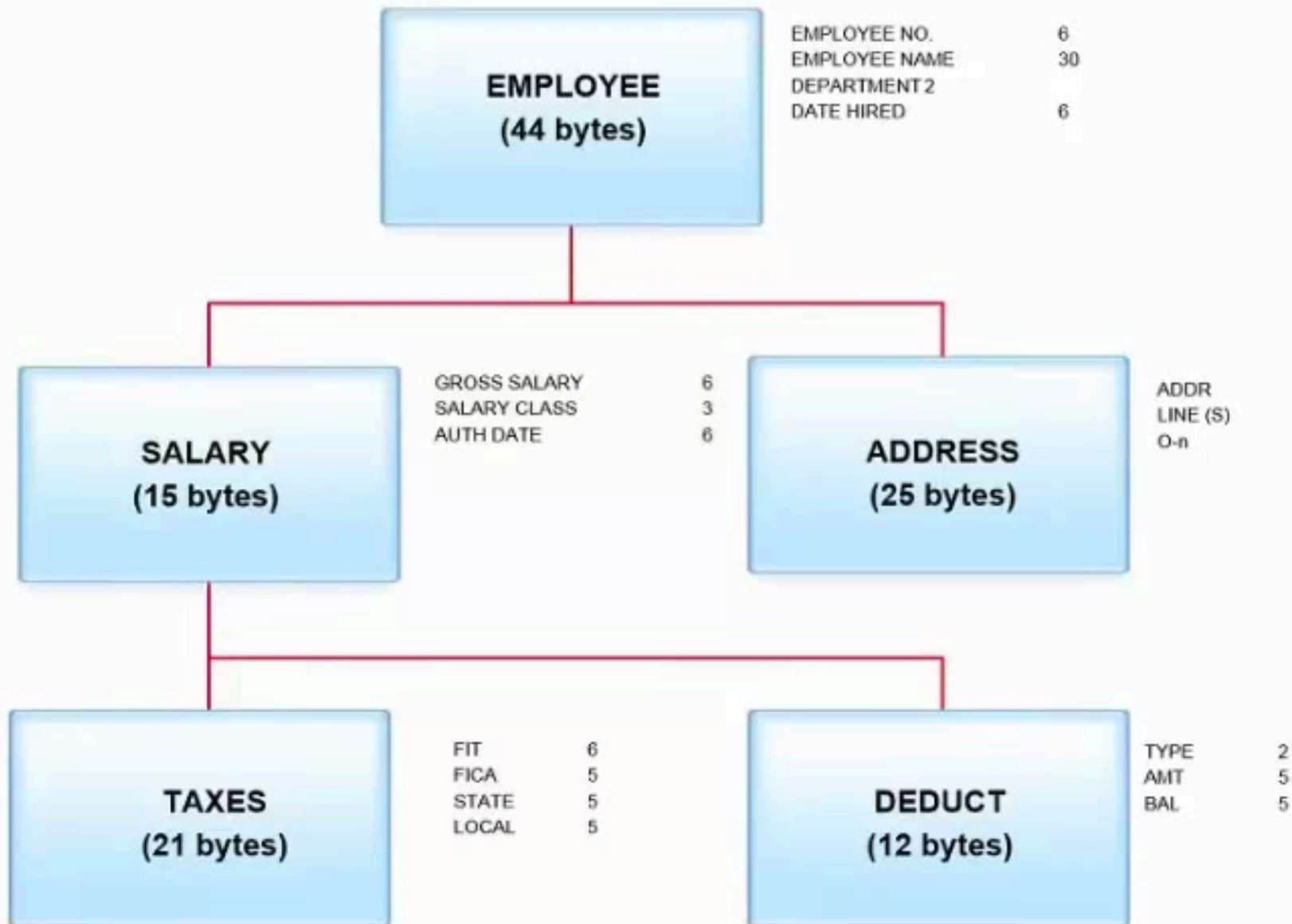
"la Caixa"

saarstahl



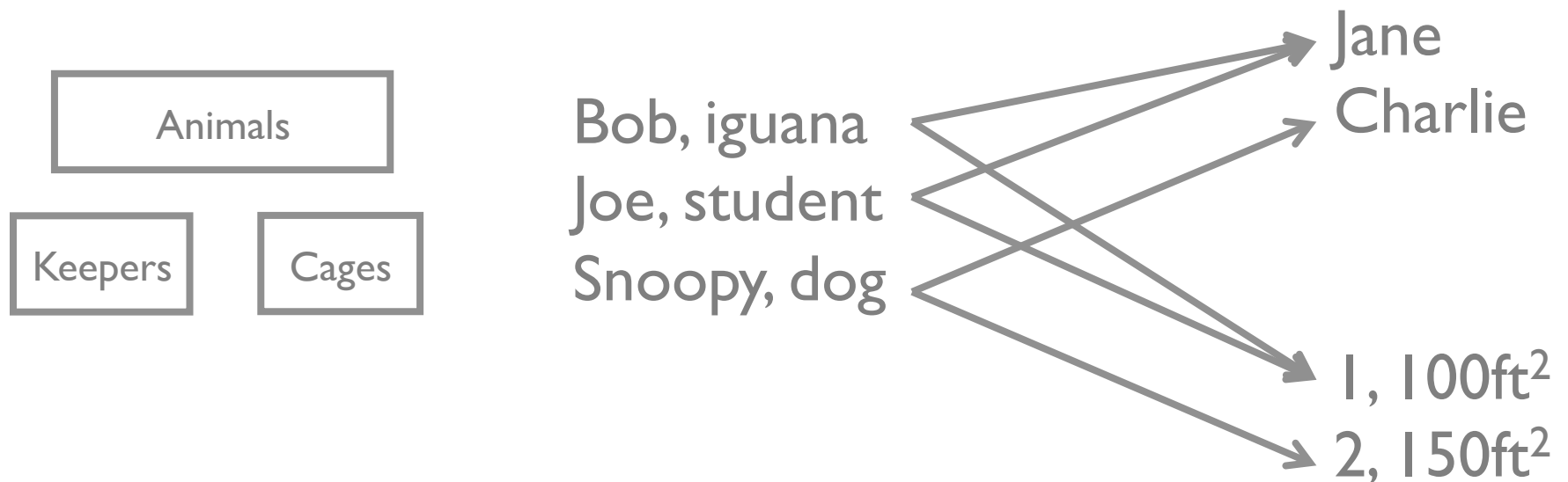
WELLPOINT

IMS DB: Hierarchical DBMS



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

| |
|-----------------------------|
| Database 12c |
| Database In-Memory |
| Multitenant |
| Options |
| Application Development |
| Big Data Appliance |
| Data Warehousing & Big Data |
| Database Appliance |
| Database Cloud |
| Exadata Database Machine |
| High Availability |
| Manageability |
| Migrations |
| Security |
| Unstructured Data |
| Upgrades |
| Windows |
| Database Technology Index |

Oracle CODASYL DBMS

Enhancements to Oracle CODASYL DBMS continue to focus on high availability, performance, and very large database support the very features that have given CODASYL DBMS a reputation for stability and reliability.

Oracle CODASYL DBMS

Oracle CODASYL DBMS is a multi-user, general purpose database management system that runs on the OpenVMS operating system. DBMS can be used to access and administer databases ranging in complexity from simple hierarchies to complex networks with multi-level relationships. It supports full concurrent access in a multi-user environment without compromising the integrity and security of the user's databases. The majority of DBMS installations occur through application providers using DBMS as the underlying database.

Oracle CODASYL DBMS is designed for users working in a structured application environment. The users are programmers, analysts, designers, or administrators who use conventional planning and coding techniques to design, create, and maintain long-term applications for corporate use. The characteristics include high availability and throughput coupled with system management features that minimize downtime. Examples are large-scale applications such as manufacturing and shop floor systems that require a stable environment not subject to change.

DBMS7: Strategic Enhancements

Strategic enhancements for Release 7 of DBMS included very large memory addressing support. Shared record cache allows DBMS to use as much physical memory as a computer system can support, so that frequently accessed records can be stored in memory to reduce disk I/O. The advantages include:

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