CPSC 304
Introduction to Database Systems

Conceptual Database Design
The Entity-Relationship Model

Textbook Reference

Database Management Systems: Chapter 2

Hassan Khosravi

Borrowing many slides from Rachel Pottinger

Databases: the continuing saga…

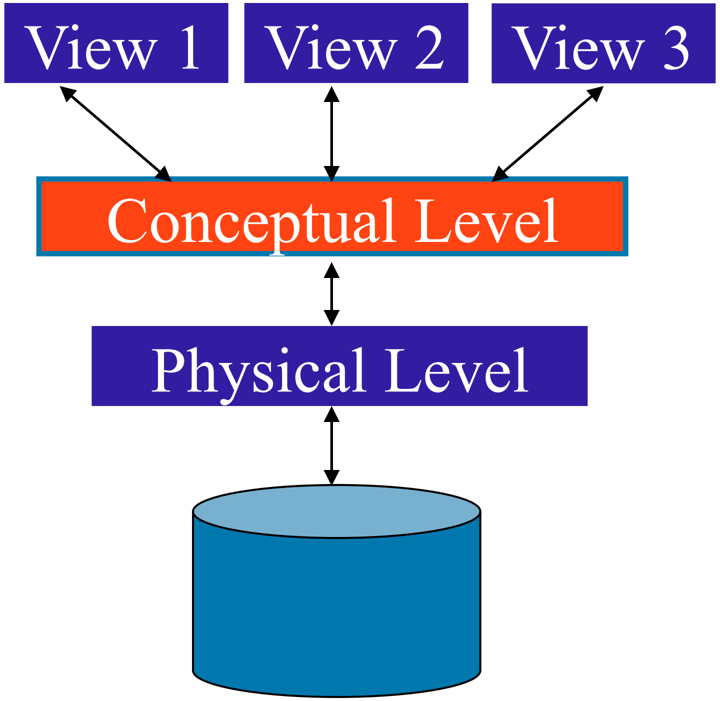
- In our last exciting episode, we motivated that databases were great because they:
  - Store large amounts of data
  - Handle transactions
  - Allow efficient querying
  - *And many, many more classic favorites!*
- Before we can do all of these, we must design the database

Learning Goals

- Explain the purpose of an ER diagram, and list the major components.
- Given a problem description, create an ER diagram given a specification. Justify the decisions you make for entities, relationships, keys, key constraints, participation constraints, weak entities, is-a relationships, and aggregations.
- Given a problem description, identify alternative representations of the problem concepts and evaluate the choices
- Compare alternative ER models for the same domain and identify their strengths and weaknesses

Levels of Abstraction

- A major purpose of a DB system is to provide an abstract view of the data.
- Three abstraction levels:
  - **Physical level**: how data is actually stored
  - **Conceptual (or Logical) level**: how data is perceived by the users
  - **External (or View) level**: describes different part of the database to different users
    - convenience, security, etc.
    - Compare views of student, registrar, & database admin.



Schema and Instances

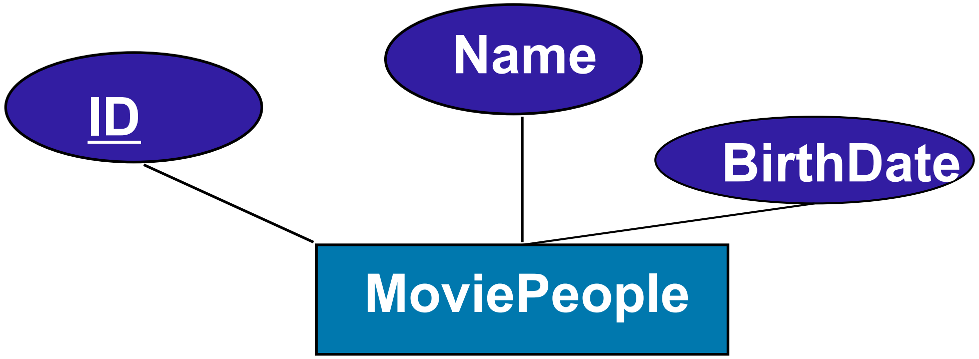
- We create the **schema** – the logical structure of the database (e.g., students take courses)
  - **Conceptual (or logical) schema**: db design at the logical level
  - **Physical schema**: db design at the physical level; indexes, etc
- Later we'll populate **instances** – the content of the database at a particular point in time
  - E.g., currently there are no grades for CPSC 304 2016 Summer term 1

Conceptual Database Design

- What are the entities and relationships in the enterprise?
  - Entities are usually nouns, but avoid irrelevant nouns
    - e.g., Students and courses
  - Relationships are statements about 2 or more objects. Often, verbs.
    - e.g., a prof teaches a course
- What information about these entities and relationships should we store in the database?
- What integrity constraints or other rules hold?
- In relational databases, this is generally encoded in an **Entity-Relationship (ER) Diagram**

ER Model Basics: Entities

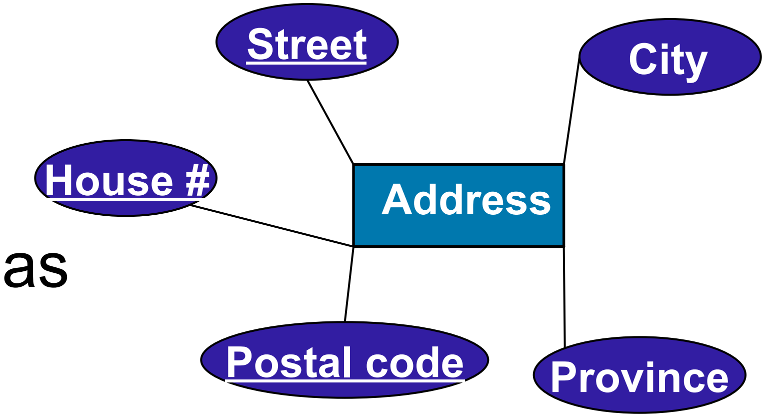
- ***Entity:*** Real-world object distinguishable from other objects.
- An entity is described using a set of ***attributes***.



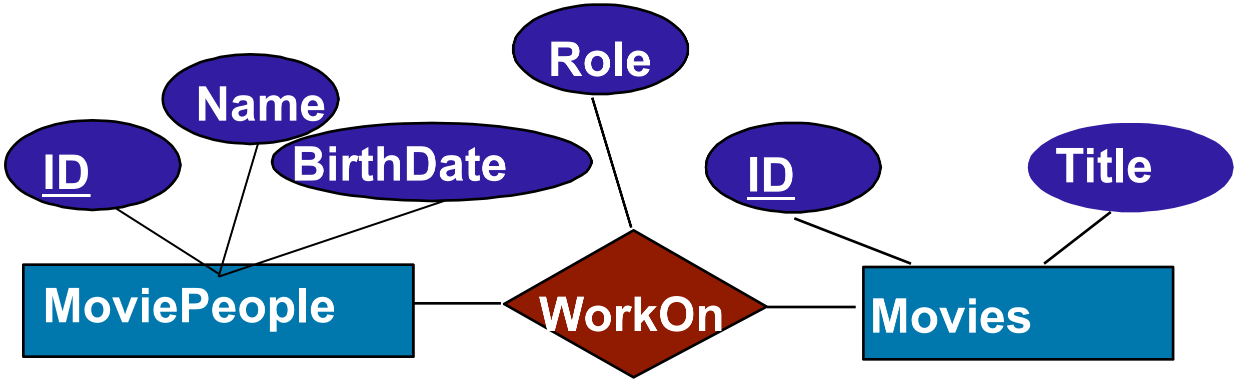
- ***Entity Set:*** A collection of similar entities. E.g., all Movie People.
  - All entities in an entity set have the same set of attributes. (Until we consider ISA hierarchies, anyway!)
  - Each attribute has a ***domain***. *(e.g.., float, date, int)*
  - Each entity set has a ***key***.

Keys

- Distinguish entities
- A *key* is the minimal set of one or more attributes which, taken collectively, identify uniquely an entity in an entity set.
- A *primary key* is the key chosen as the principal means to identify entities in an entity set
- Only primary keys are shown in ER diagrams
- We'll discuss superkeys when we consider normal forms (for now, don't worry about them)



ER Model Basics (Cont.)



Role

Name

ID

BirthDate

ID

Title

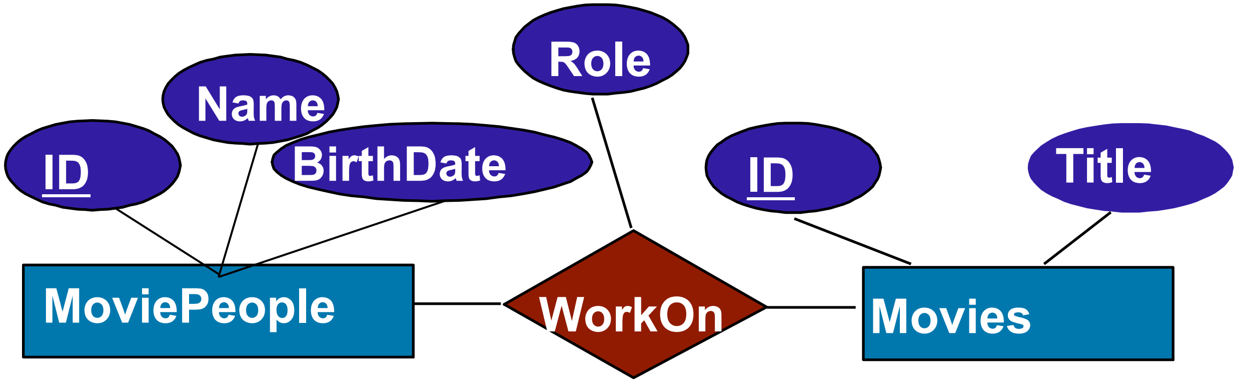
MoviePeople

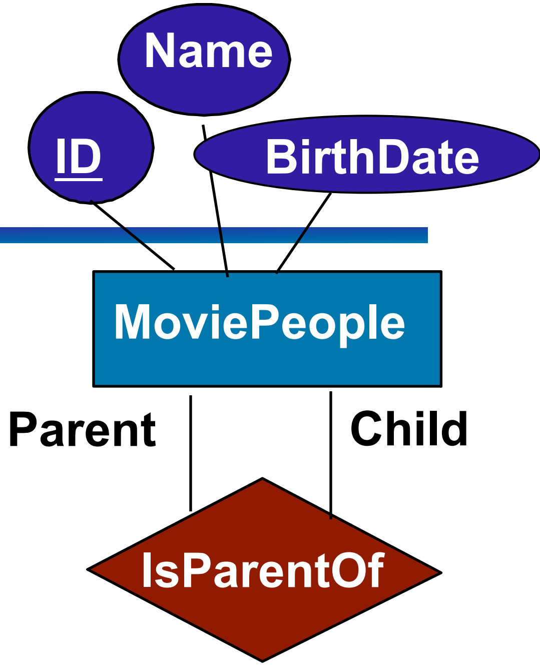
WorkOn

Movies

- ***Relationship**:* Association among two or more entities.
  - E.g., George Clooney worked on Gravity.

ER Model Basics (Cont.)

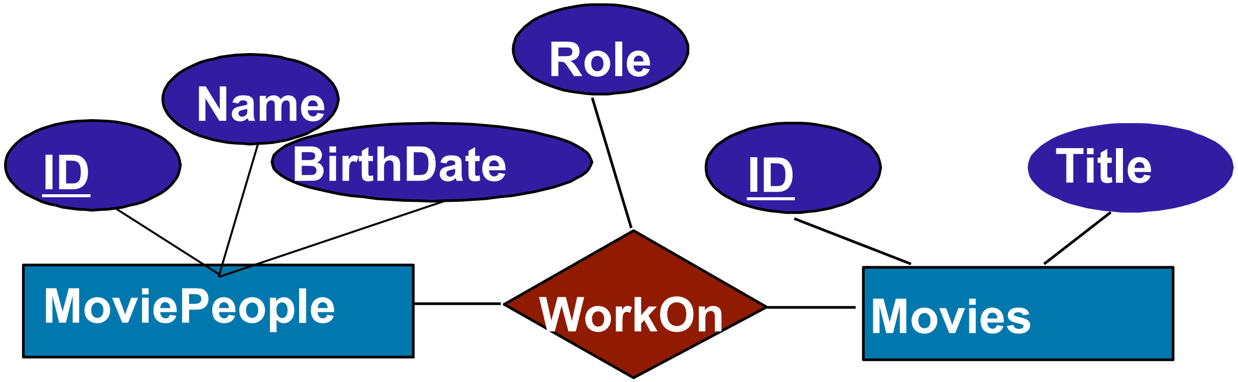


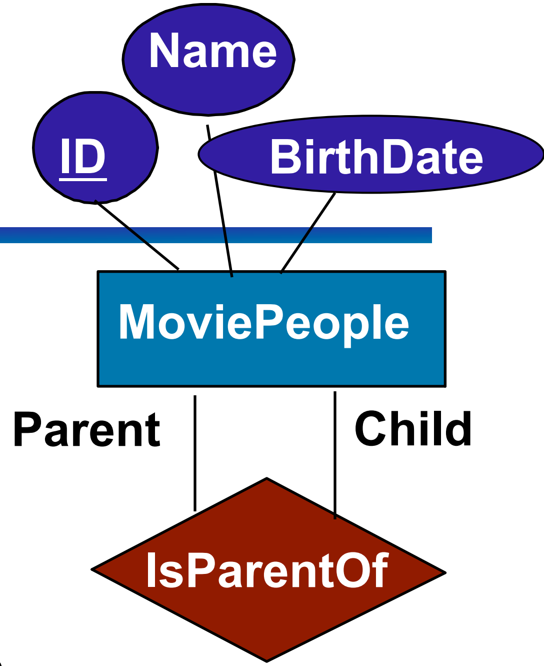


- ***Relationship Set:*** Collection of similar relationships.
  - Collection of all moviePeople that have worked in Movies.

- Same entity set could participate in different relationship sets, or in different “roles” in same set. (Kirk Douglas isParentOf Michael Douglas)

ER Model Basics (Cont.)





- ***Relationship Set:*** Collection of similar relationships.
  - Collection of all moviePeople that have worked in Movies.
- Same entity set could participate in different relationship sets, or in different “roles” in same set.
- A relationship set may have ***descriptive attributes*** (like since).
- An n-ary relationship set R relates n entity sets E_1 ... E_n; each relationship in R involves entities $e_1 \in E_1, \ldots, e_n \in E_n$
  - ***Degree*** or ***arity:*** # of entity sets in relationship (binary, ternary, etc.)

In-class Exercise: Registrar’s database

- Design a registrar's database to store information about students, courses, the courses students have taken, and the grades students have gotten in these courses. Some relevant details are: Courses have a number, a department, and a title. for example, "CPSC111: Introduction to Computing" has department = CPSC, number = 111, and title = "Introduction to Computing."
- Numbers are assigned by departments, and different departments may use the same number.
- Students are represented by their (unique) student ID and their name.
- "Enrollments" each consist of a course, a student who took that course, and the grade the student got in the course.
- You should draw one or more E/R diagrams that represent this database structure correctly.

Which of the following might you find in a correct E/R diagram?

A. Entity set Students with attribute ID not underlined and name underlined.

B. Entity set Students with attributes ID and name underlined.

C. Entity set Courses with attributes department and number underlined and title not underlined.

D. Entity set Courses with attribute department underlined and attributes number and title not underlined.

E. None of the above

Which of the following might you find in a correct E/R diagram?

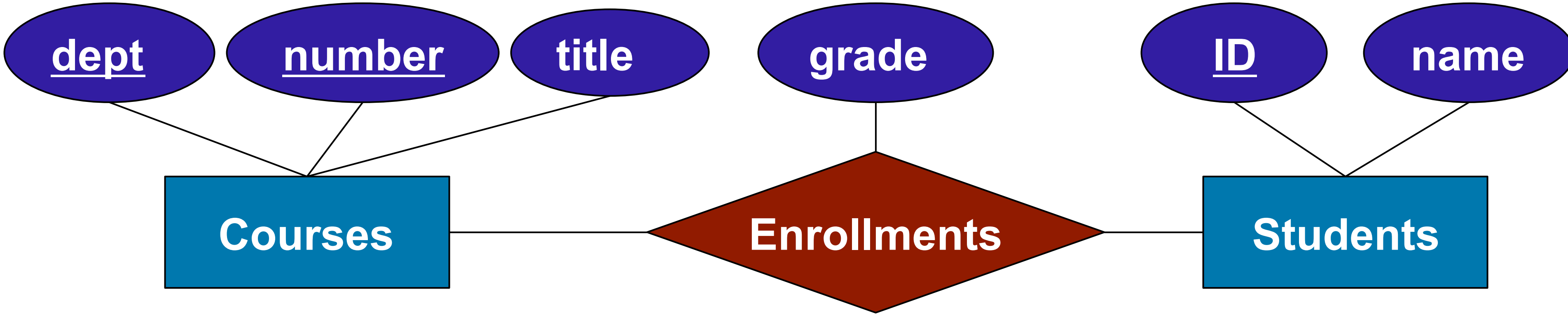
A. Entity set Students with attribute ID not underlined and name underlined.

B. Entity set Students with attributes ID and name underlined.

C. Entity set Courses with attributes department and number underlined and title not underlined. Correct answer

D. Entity set Courses with attribute department underlined and attributes number and title not underlined.

E. None of the above



Can a student take a course twice?

Okay, that worked pretty well…

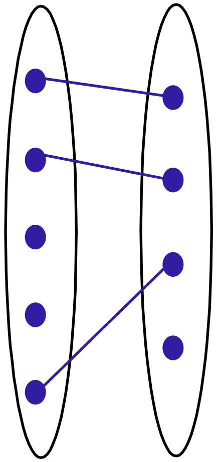
- But what if we wanted to add instructors to courses and restrict that each could only have one instructor?

Cardinalities

- A **cardinality ratio** for a relationship set specifies the number of relationships in the set that an entity can participate in.

Let R be a relationship set between sets A and B.
R can have 1 of 4 cardinalities:

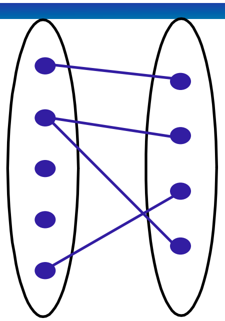
1. ***one-to-one*** from A to B:
   - an entity in A is associated with at most one entity in B and vice versa
   - e.g. A: driver, B: driver's license



1-to-1

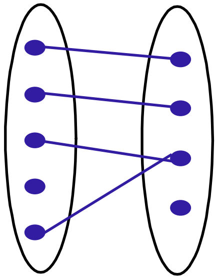
Cardinalities (cont’)

2. ***one-to-many*** from A to B:
   - an entity in A is associated with any number of entities in B
   - an entity in B is associated with at most one entity in A
   - e.g. A: biological-mother, B: children



1-to Many

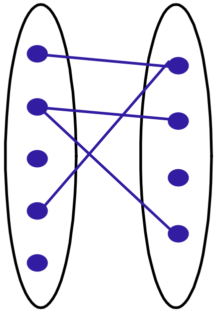
3. ***many-to-one** from A to B: switch A and B above*



Many-to-1

Cardinalities (cont')

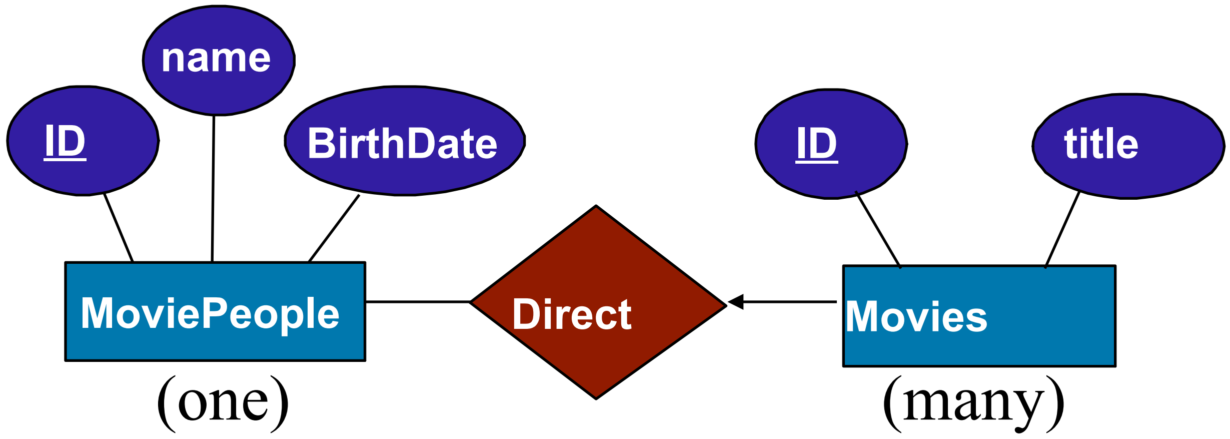
- ***many-to-many*** *from A to B:*
  - *an entity in A is associated with any number of entities in B and vice versa*
  - *e.g. A: students, B: courses*

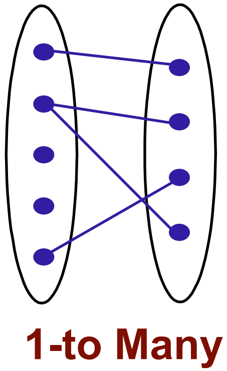


Many-to-Many

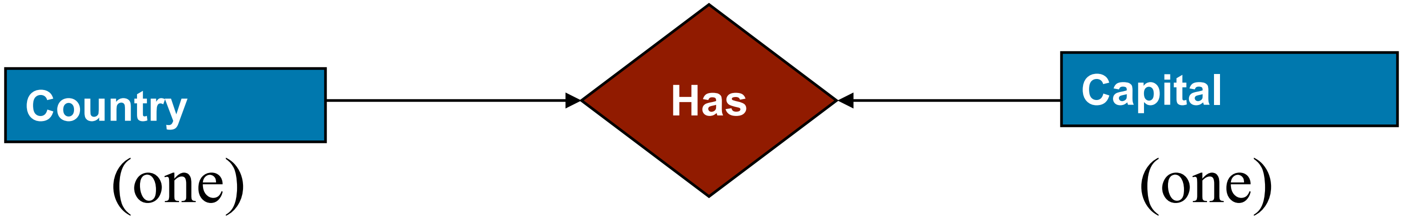
Key Constraints

- The restriction imposed by a 1-to-1 and 1-to-many ratios are examples of ***key constraints***.
- A key constraint is shown with an arrow in the ER diagram.
- Important on insertions



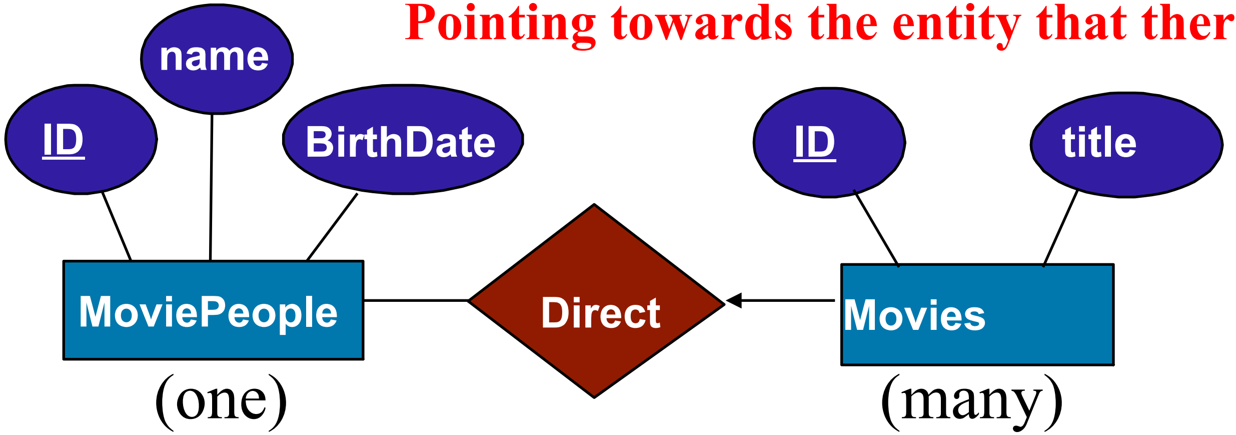


or

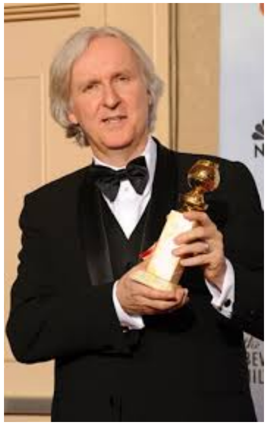


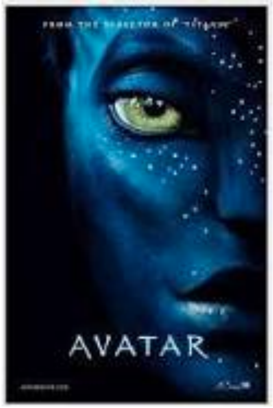
A brief digression on notation

- The ER notation we use can be read: “if you know the entity with the arrow, then you know the relationship (and the other entities involved)”

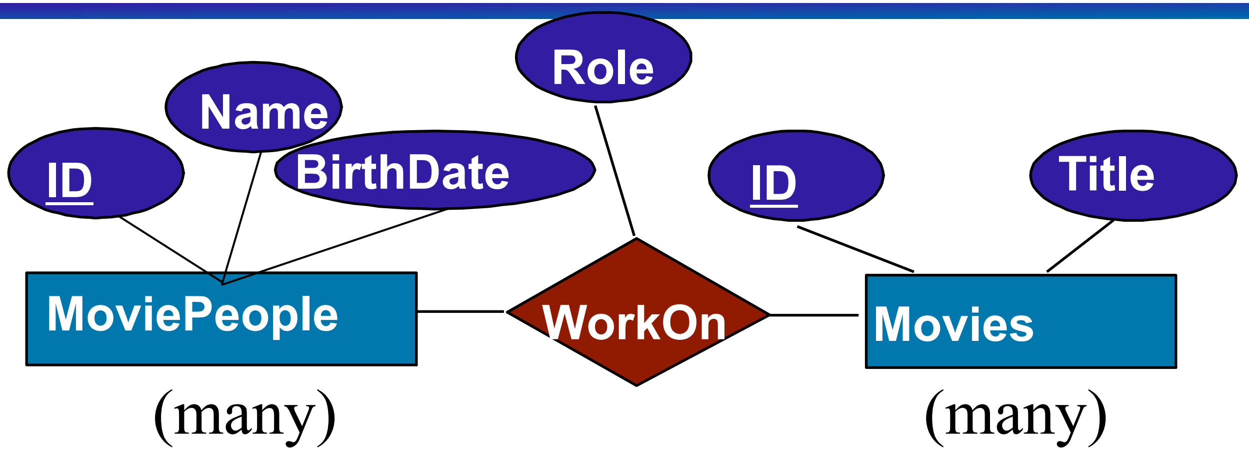


Pointing towards the entity that there is only one of

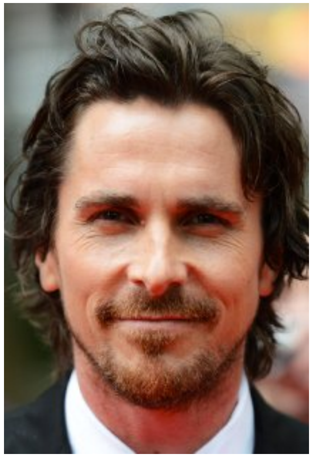




How can we uniquely identify a relationship?



- How can we identify the role of a specific MoviePerson in a specific movie

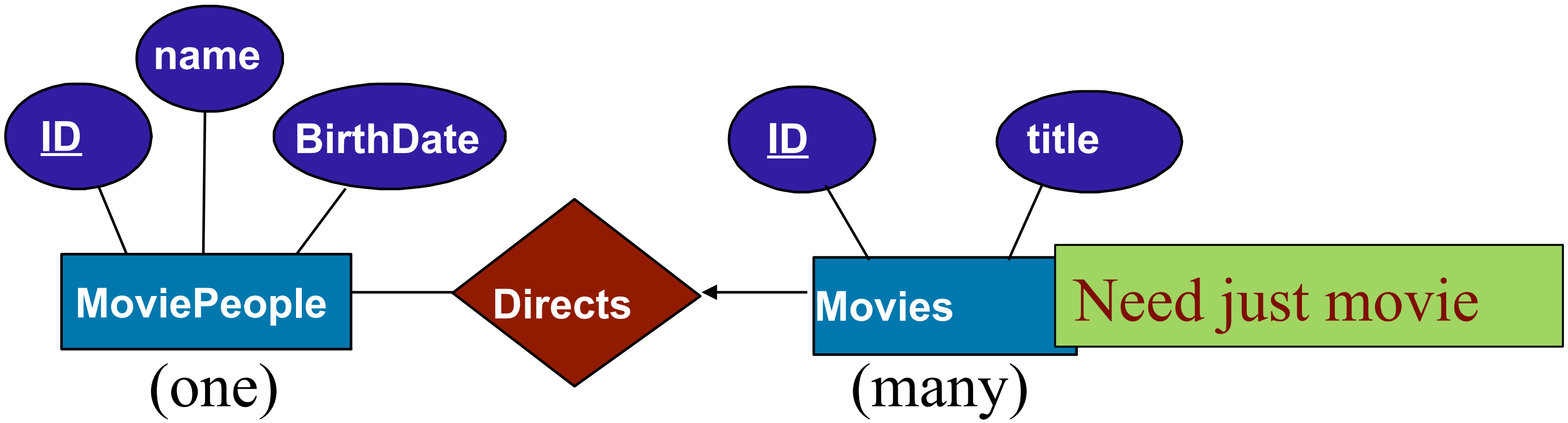


Christian Bale
as
Bruce Wayne

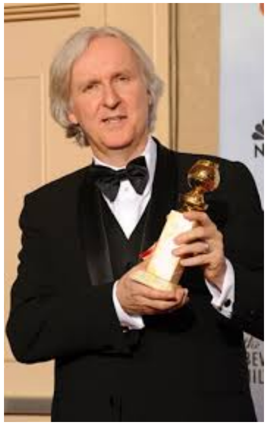
WELCOME TO A WORLD WITHOUT RULES
BALE CAINE LEDGER OLDMAN ECKHART GYLLENHAAL FREEMAN
THE DARK KNIGHT
JULY 18

Can the same person have multiple roles?

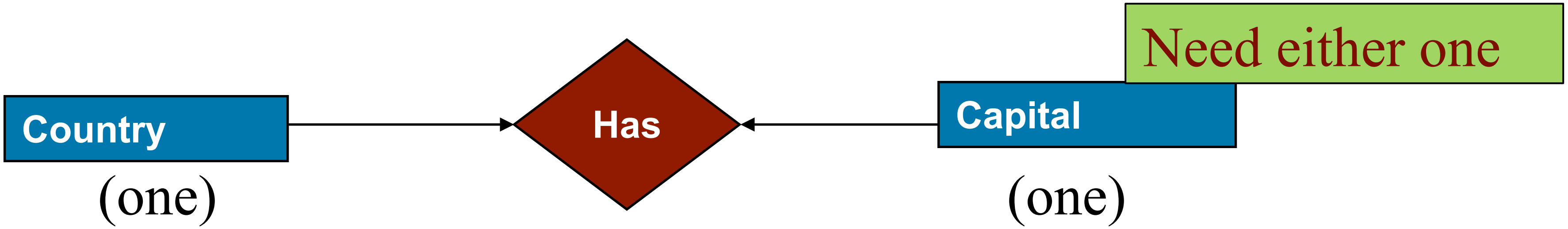
How can we uniquely identify a relationship?



Need just movie



How can we uniquely identify a relationship?



Need either one

Ottawa

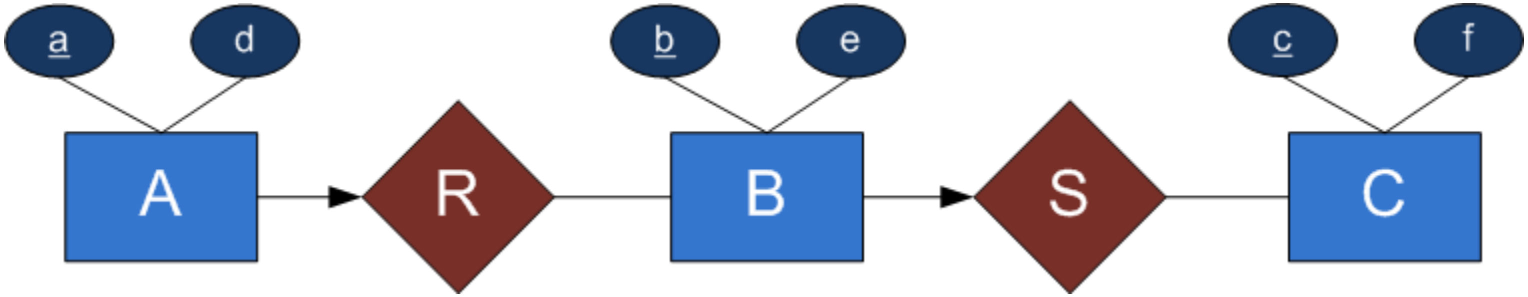
Summary:
Primary Keys of Relationship Sets

- Let R be a relationship set between sets A and B. R's primary key is:

TYPE OF R	PRIMARY KEY OF R
one-to-one	primary key of A or primary key of B
one-to-many from A to B	primary key of B
many-to-many	primary key of A + primary key of B

- R may have its own key, in addition to the key it inherits from the entities.

rel-name = {(e1, f1), (e2,f2)} means that a relationship between e1 and f1 exists in the relationship set for rel-name, as does a relationship between e2 and f2.



Suppose that a1 and a2 are the only entities of A, b1 and b2 are the only entities of B, and c1 and c2 are the only entities of C.

Which of the following relationship sets for R and S are possible according to the diagram, where T = {(e1,f1)} means a relationship between e1 and f1 exists in relationship set T

A. R = {}; S = {(b2, c1), (b2, c2)}

B. R = {}; S = {(b1, c2), (b2, c2)}

C. R = {(a2, b2)}; S = {(b2, c1), (b2, c2)}

D. R = {(a1, b2), (a2, b1), (a2, b2)}; S = {}

E. None of the above

B defines C (S)

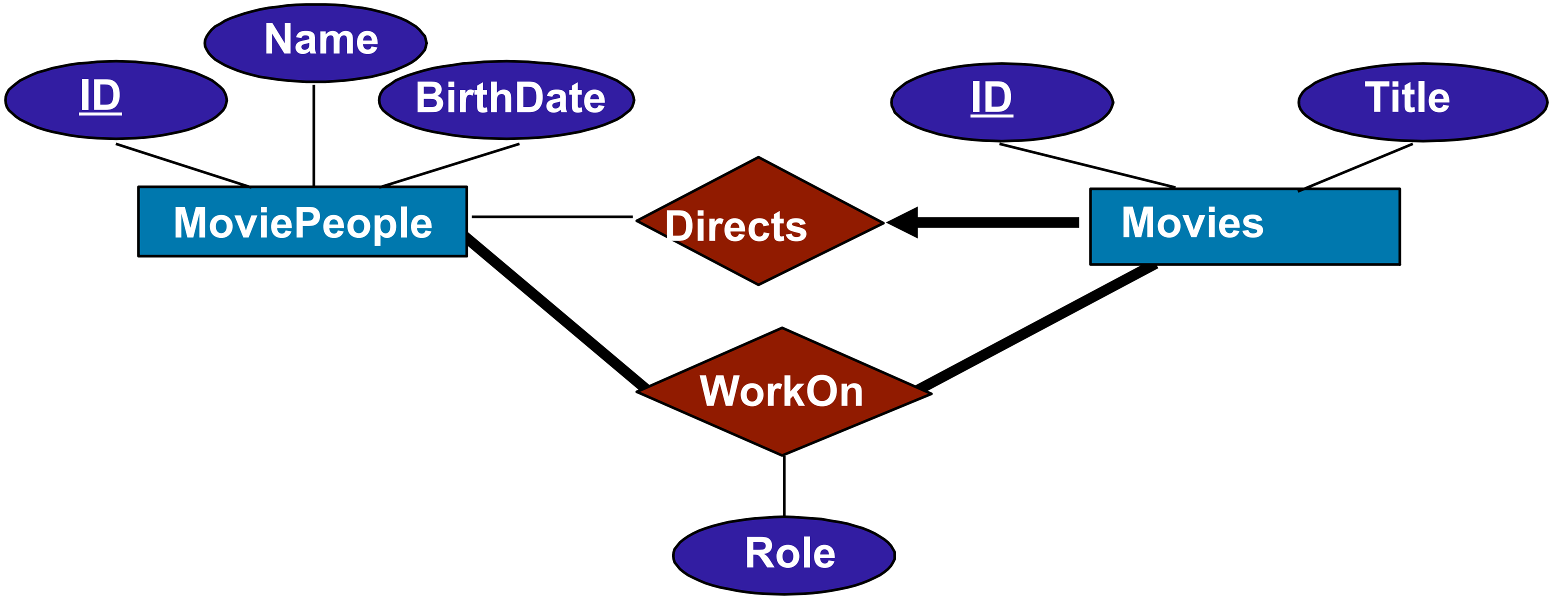
correct

B defines C (S)

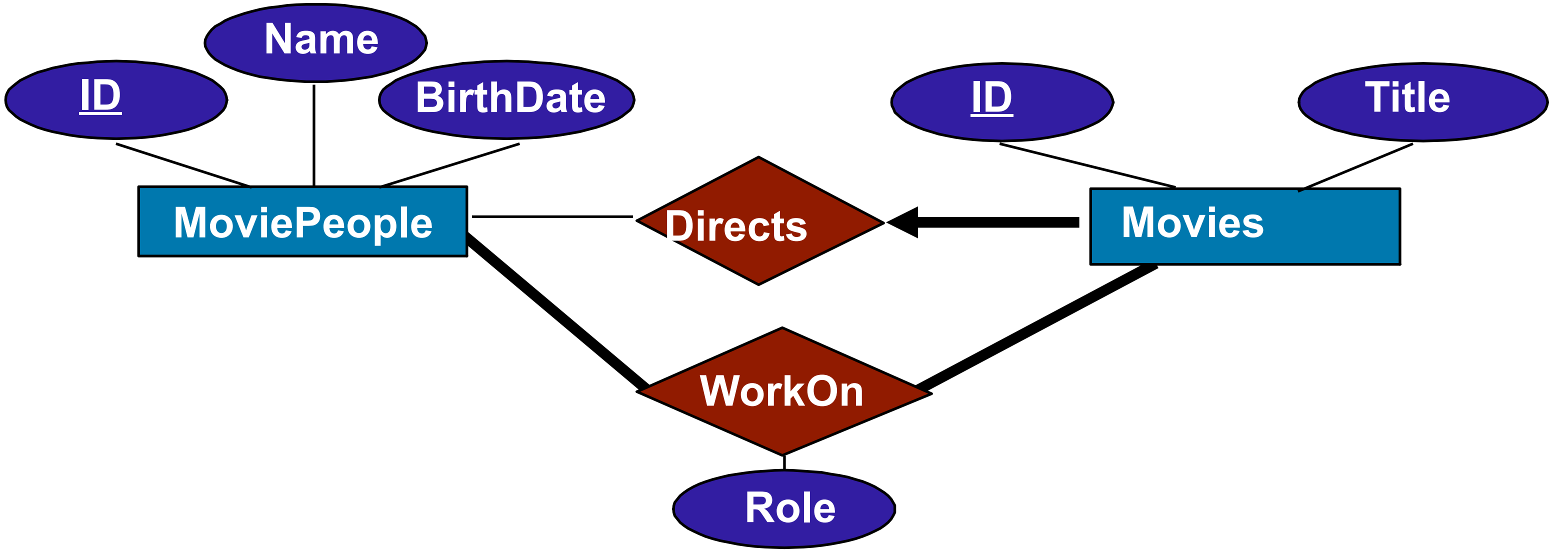
A defines B (R)

Participation Constraints

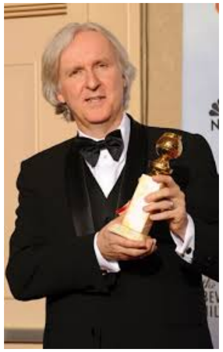
- Participation : Indicates if all entities participate in the relationship.
- An entity's participation can be ***total*** or ***partial*** .
- Requiring total participation is a ***participation constraint*** and it is shown with a thick line
  - Important on deletions
  - i.e., participation of Movie in Directs is total (thick line)
    - Every movie must appear in some relationship in the Directs set

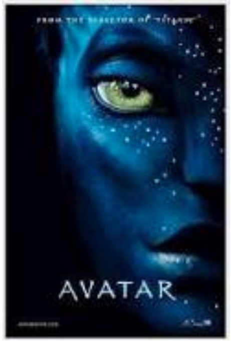


Why is participation constraint important



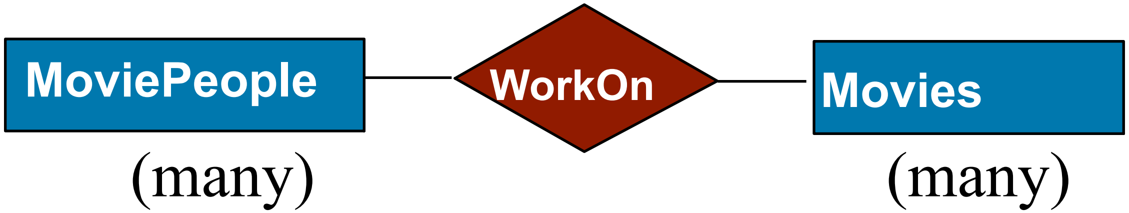
Would I be able to delete James Cameron without deleting Avatar?
Would I be able to delete Avatar without deleting James Cameron?



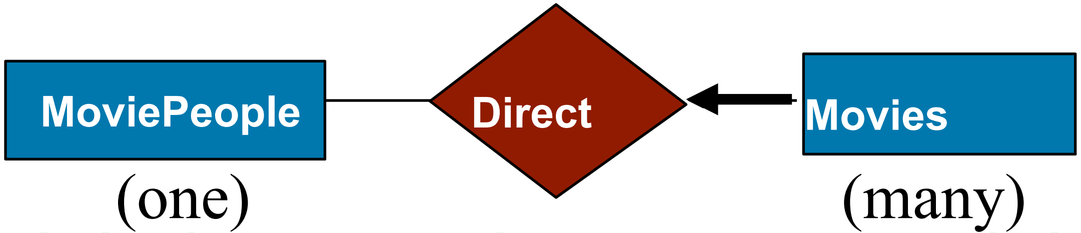


Line types summarized

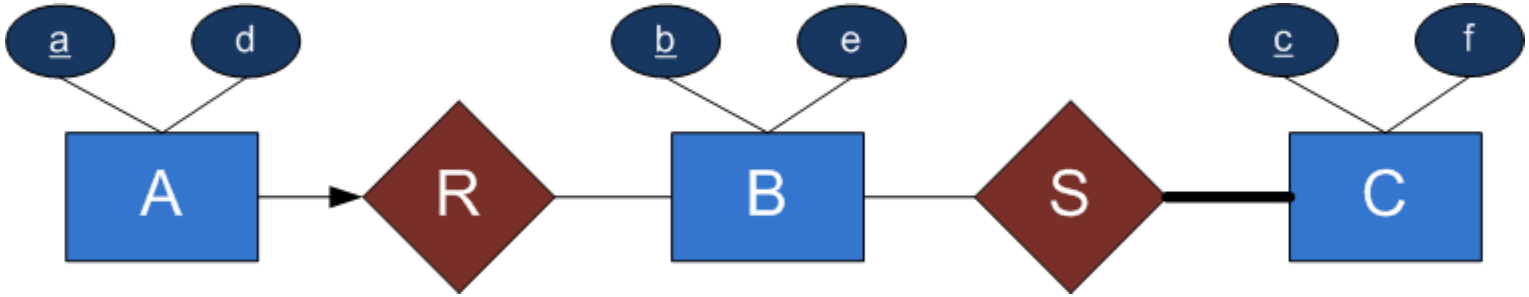
- Plain lines mean many to many:



- Arrows mean the other side has a cardinality of one



- A thick line requires total participation and can be added to any line, arrow or not rel-name = {(e1, f1), (e2,f2)} means that a relationship between e1 and f1 exists in the relationship set for rel-name, as does a relationship between e2 and f2.



Suppose that a1 and a2 are the only entities of A, b1 and b2 are the only entities of B, and c1 and c2 are the only entities of C.

Which of the following relationship sets for R and S are possible according to the diagram, where T = {(e1,f1)} means a relationship between e1 and f1 exists in relationship set T

A. R = {}; S = {} — C must participate

B. R = {(a1,b1)}, S = {(b2,c2)} — C must participate

C. R = {(a1,b1), (a1,b2)}; S = {(b1,c1), (b2,c2)} — A → B (R)

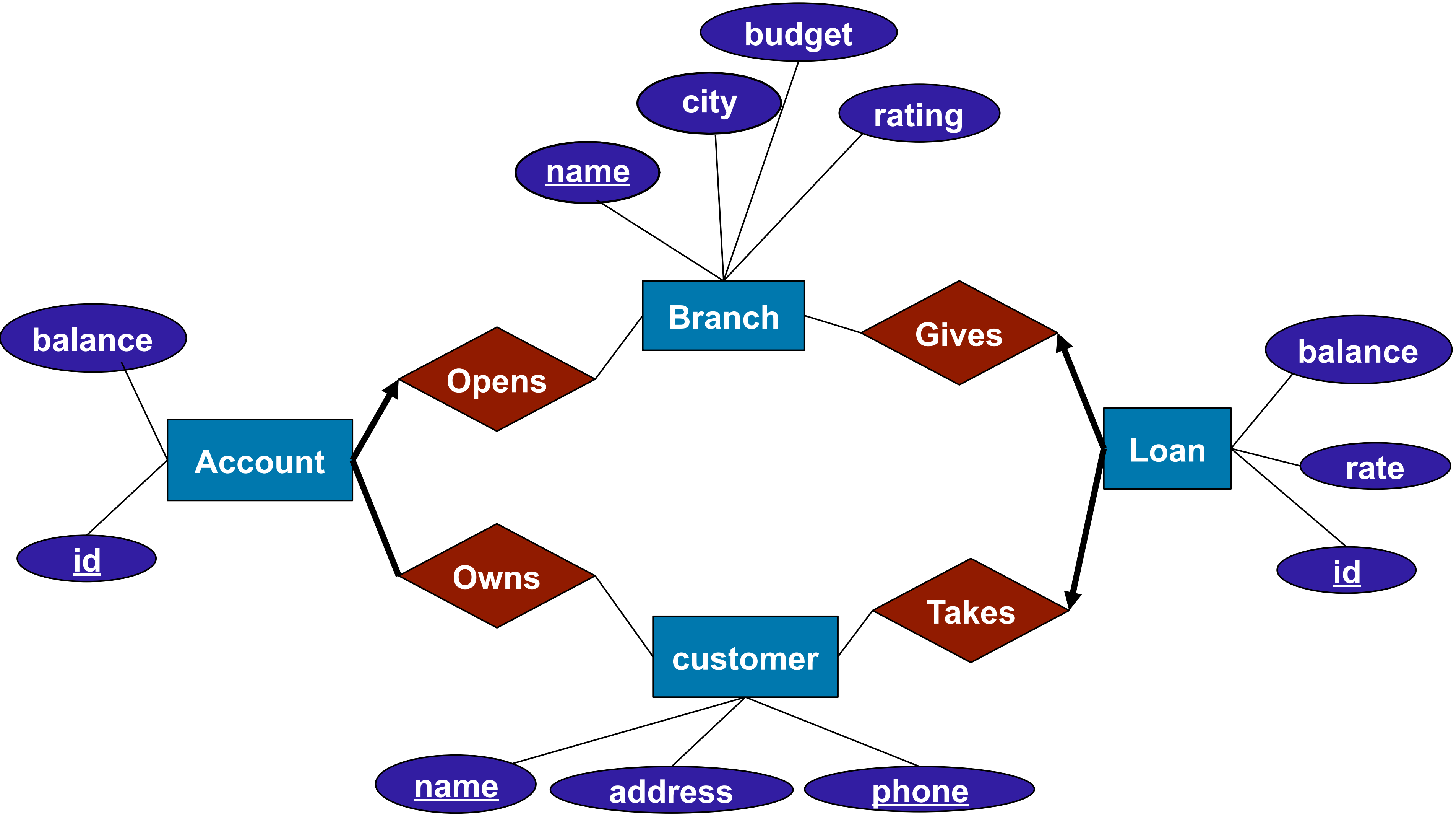
D. R = {(a1, b2)}; S = {(b1,c2), (b2, c1), (b1,c1)} — correct

E. None of the above

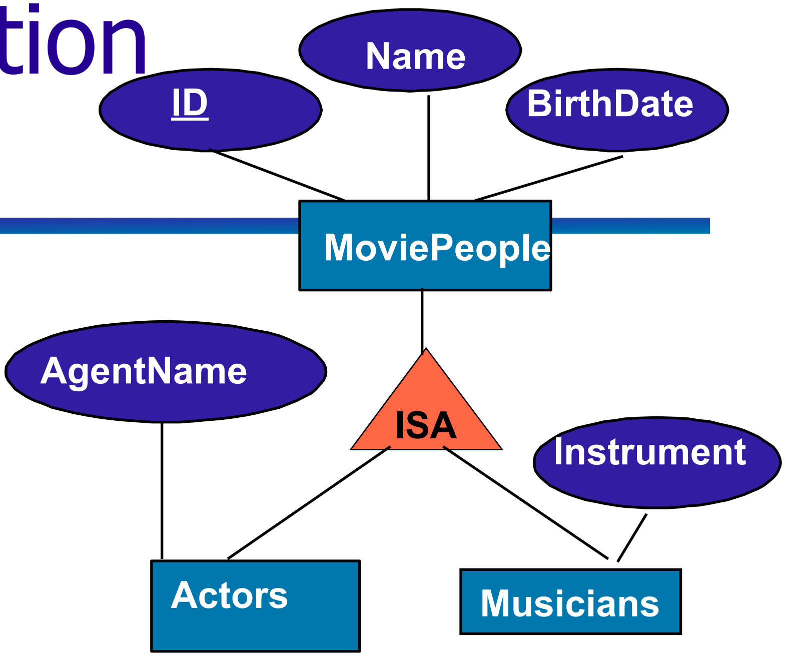
Exercise: ABC Banks

- The bank is organized into branches. Each branch is located in a particular city and is identified by a unique name. Each year the bank’ board defines the yearly budget and the rating (which is a number from 1 to 10) for each branch.
- A bank customer is identified by their customer name and phone number. The bank also keeps track of each customer’s current address.
- The bank offers accounts and loans to its customers. Each account and loan has a unique number and is created and maintained by a single branch.
- Each account is assigned to one or more customers and its balance can never be negative.
- A loan is always assigned to a single customer, has a fixed interest rate and its balance cannot be negative either.

Sample solution



Generalization/Specialization (ISA relationships)

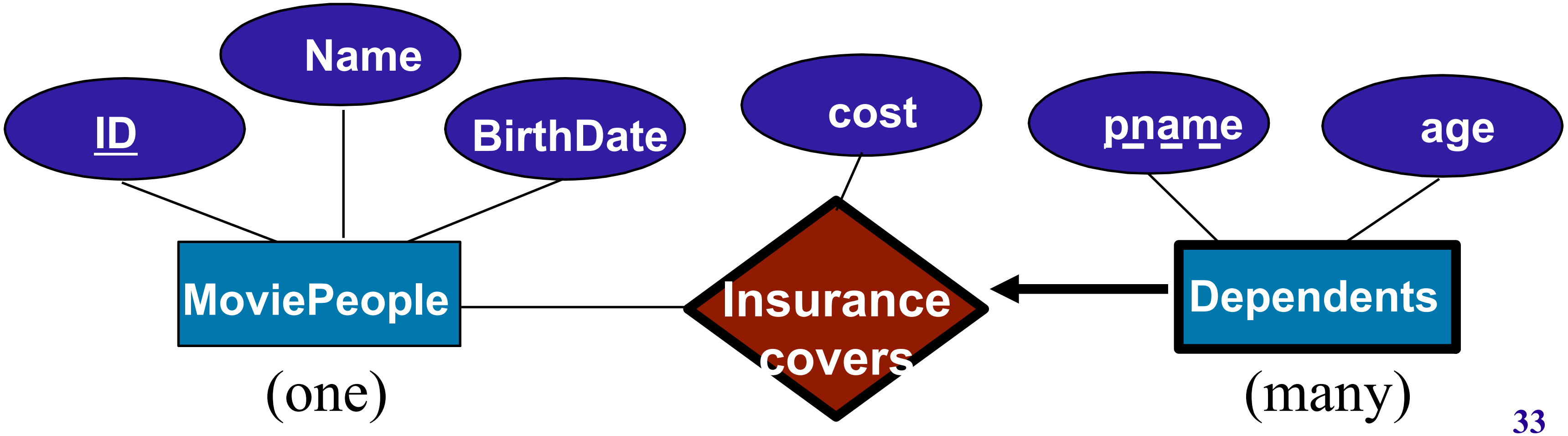


- As in Java, or other PLs, attributes can be inherited.
- If we declare A ***ISA*** B, every A entity is a B entity.
- Reasons for using ISA:
  - To add descriptive attributes specific to a subclass.
  - To restrict entities that participate in a relationship.

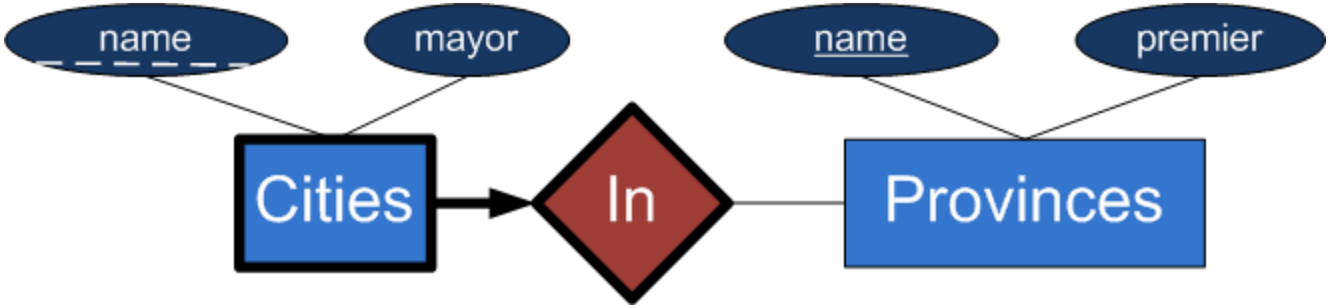
- How is "ISA" different from a relationship?
  - Hard to define a key for Actors here

Weak Entities

- A ***weak entity*** can be identified uniquely only by considering the primary key of another (*owner*) entity.
  - Owner entity set and weak entity set must participate in a one-to-many relationship set (one owner, many weak entities).
  - Weak entity set must have total participation in this ***identifying*** relationship set.
  - Think of this as a “belongs to” relationship.
- Weak entity sets and their identifying relationship sets are shown with thick lines.



Clicker exercise



Which of the following is necessarily true:

A. No two provinces can have premiers with the same name.

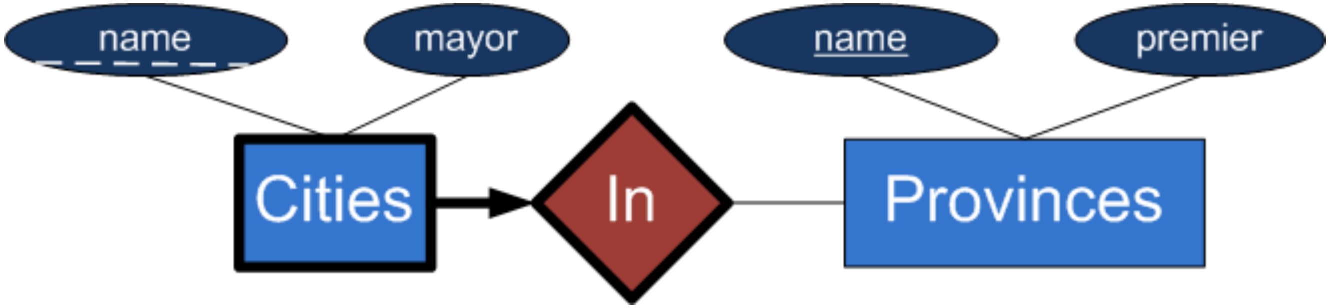
B. No two cities can have mayors with the same name.

C. No two cities can have the same name.

D. No person can be the mayor of Cities In two different provinces.

E. None of the above

Clicker exercise



Which of the following is necessarily true:

A. No two provinces can have premiers with the same name.

B. No two cities can have mayors with the same name.

C. No two cities can have the same name.

D. No person can be the mayor of Cities In two different provinces.

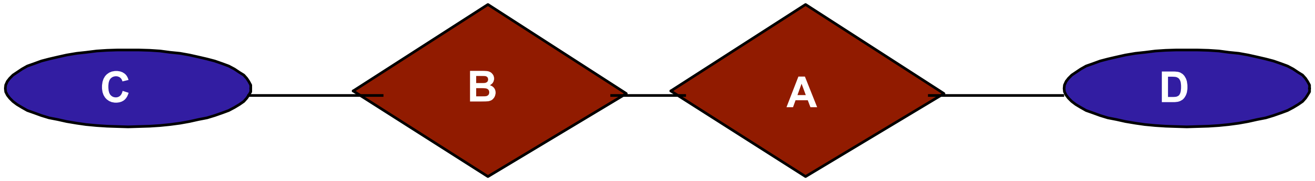
E. None of the above

E is correct

Key: name of City + Name of province (Victoria, BC)

Aggregation

- Having a relationship between relationships is forbidden.

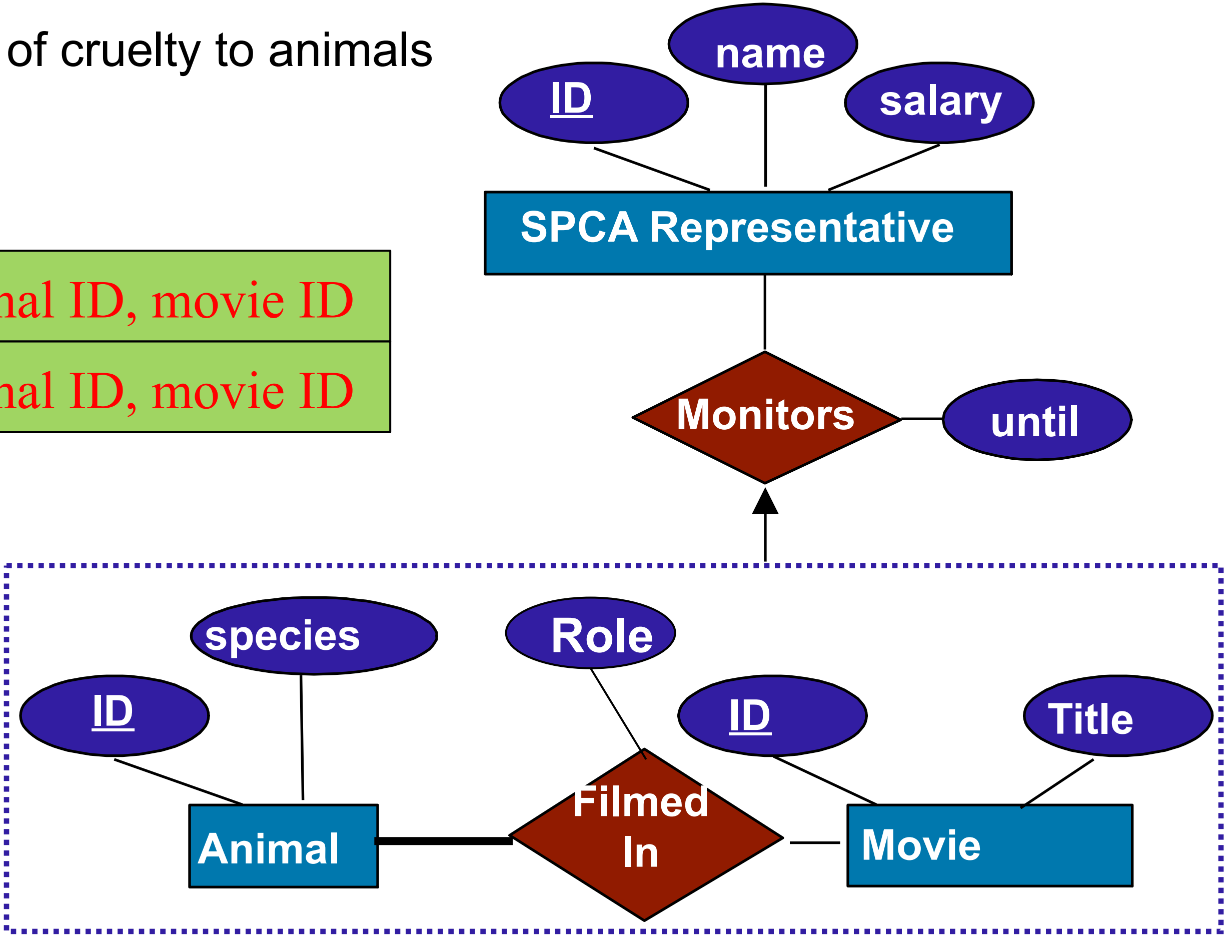


- *Aggregation* allows us to treat a relationship set as an entity set for purposes of participation in (other) relationships

Aggregation

society for the prevention of cruelty to animals

Key for FilmedIn?	Animal ID, movie ID
Key for Monitors?	Animal ID, movie ID



- Each sponsorship is monitored by at most one SPCA Representative.

Clicker Exercise

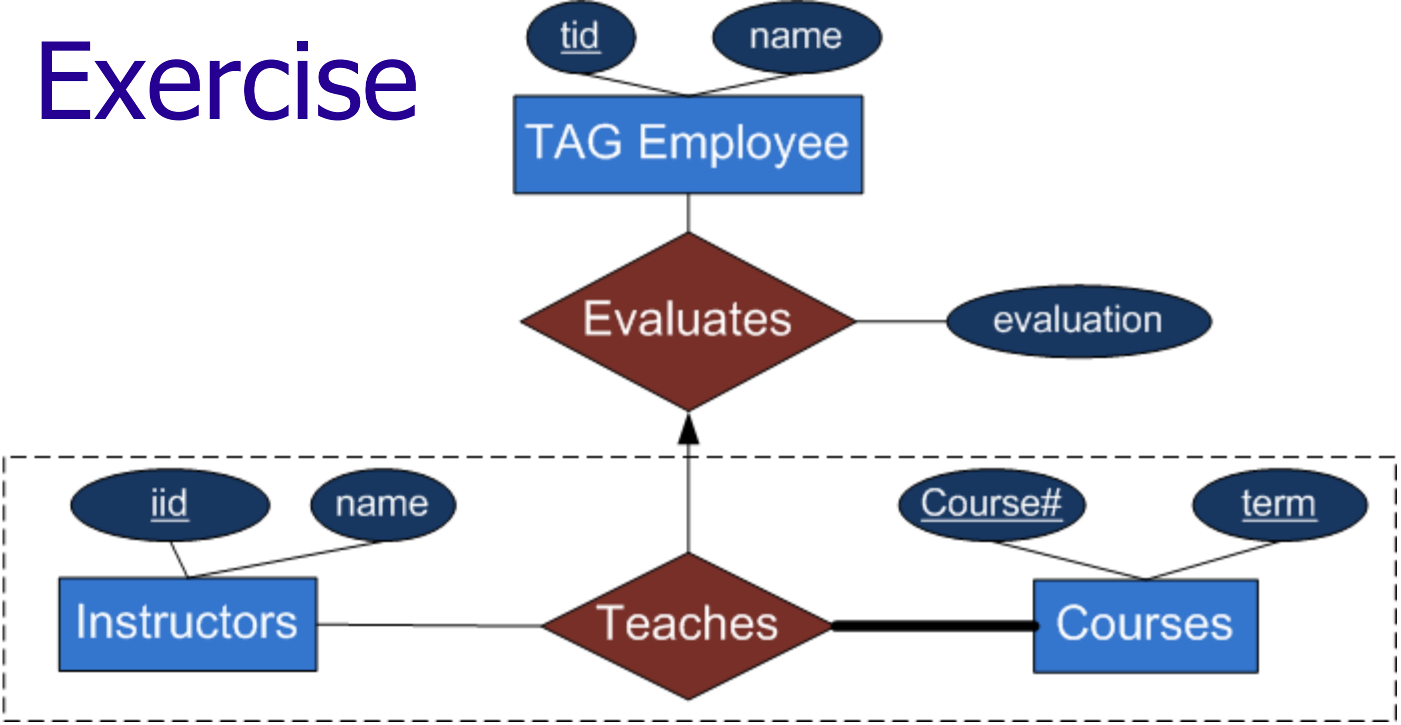


Figure out the (minimal) keys for each entity set and each relationship set in the above diagram.

Clicker Exercise

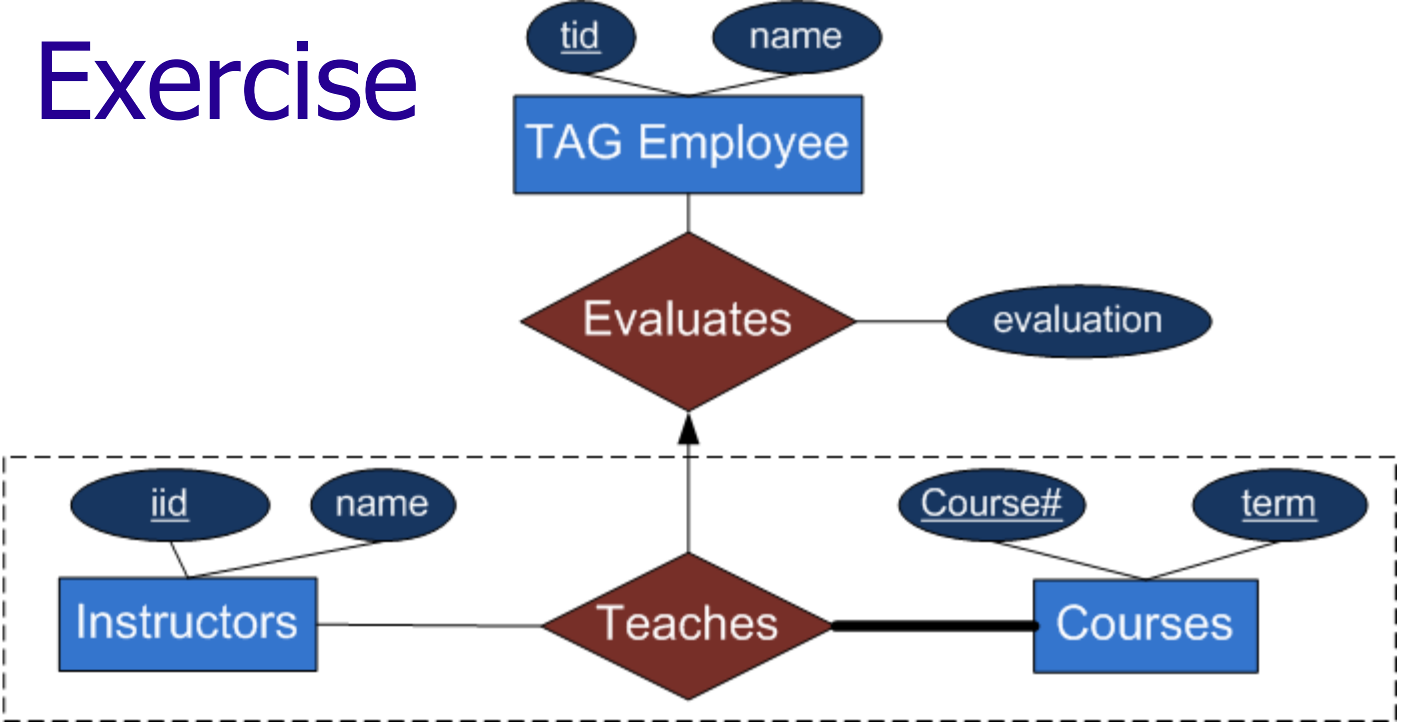


Figure out the (minimal) keys for each entity set and each relationship set in the above diagram.

Choose the correct choice of (minimal) key from the options below:

A. The (minimal) key of Evaluates is tid

B. The (minimal) key of Evaluates is iid + course# + term.

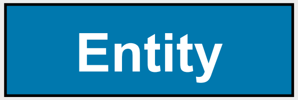
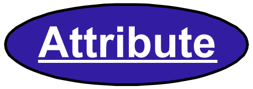
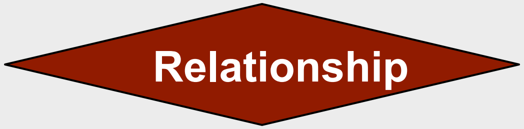
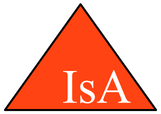
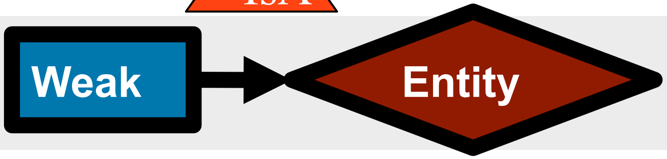
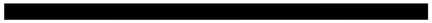
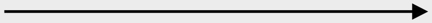
C. The (minimal) key of Evaluates is iid + course# + term + tid

D. The (minimal) key of Evaluates is iid+ course# + term + evaluation

E. None of the above

B – the key of teaches

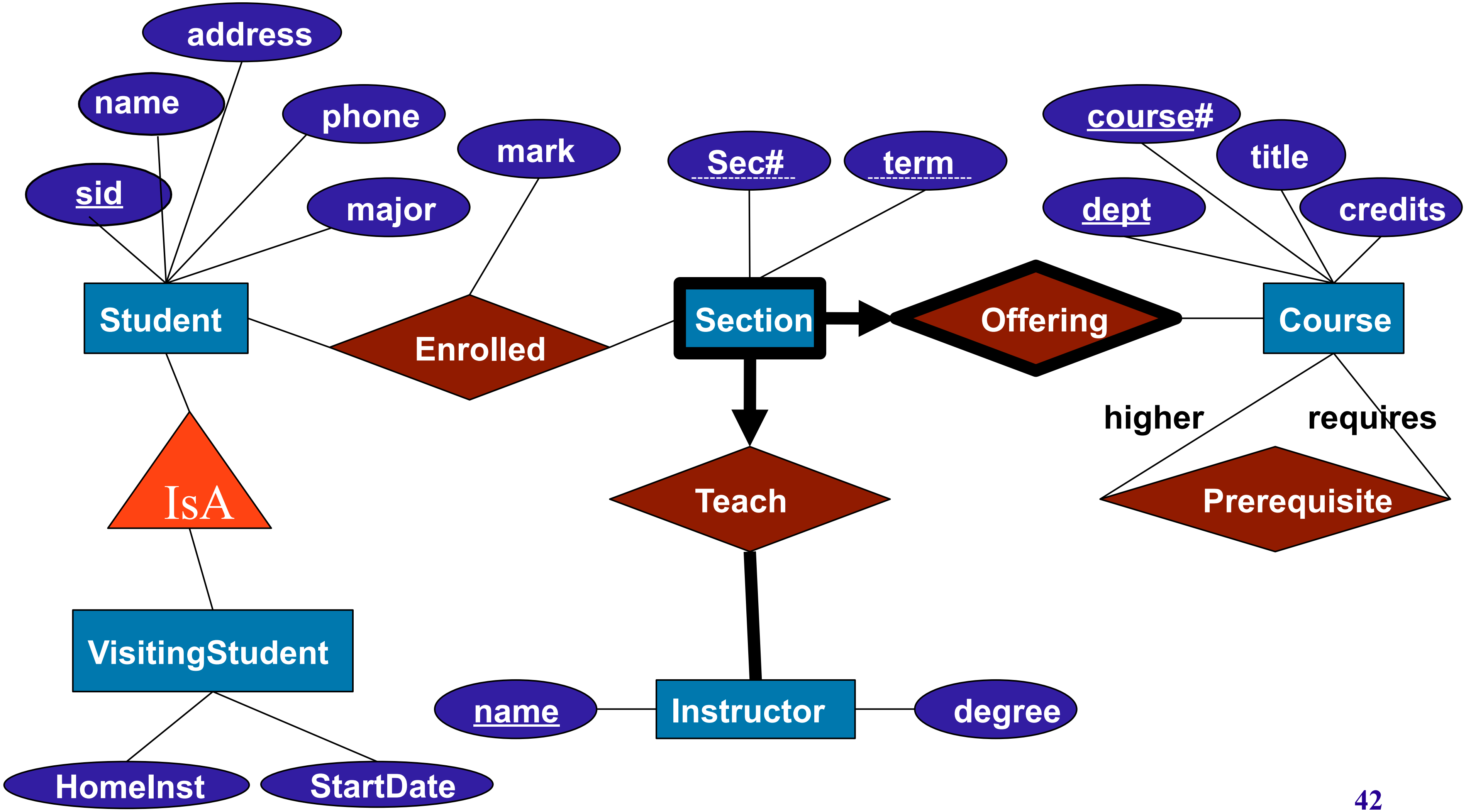
Summary

Name	Symbol
Entity	
Attribute	
Relationship	
Generalization/Specialization	
Weak Entity	
Participation Constraints	
key constraints	

Exercise: U of U University. Draw an ER diagram for the following:

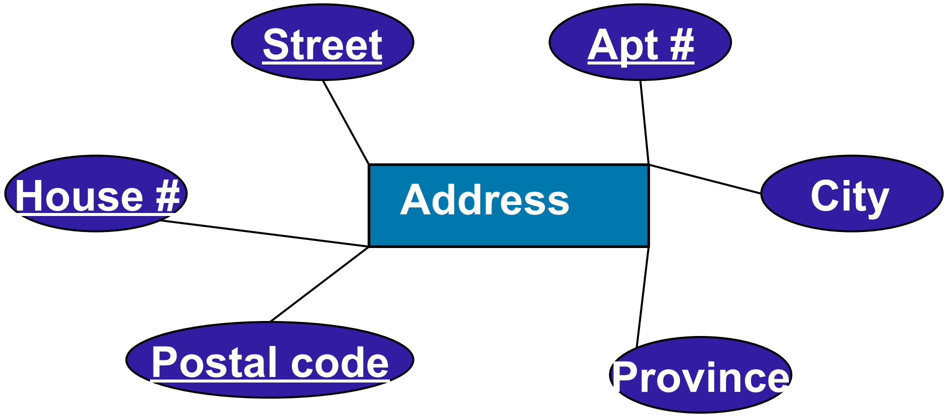
- The primary function of UofU is to offers courses to students.
- A student is identified by a unique student #, and has an address and a phone #. Each student is registered in a program at UofU
- Visiting students stay at UofU for a year.
- A course offered by UofU is identified by the department that offers the course and a course# which is unique within the department. We list our courses with their titles and the credits in our calendar.
- A course may be offered many times, even within the same term. Each offering is assigned a section # which is unique for a given course and year, and is taught by a single instructor.
- Each instructor is responsible for some section; there are no idle instructors. Instructors have unique names, and may teach a # of sections of different courses. For each instructor we like to keep info about their higher degree.
- A student register in a course section and gets a mark for the course.
- A course may have any number of other courses as prerequisites.

Sample solution



That’s all there is to it

- Some ER models differ in expressiveness
- They model *most* concepts people want
- They don’t model all of them, e.g.,
  - Functional dependencies – some attributes determine some other attributes



Postal code determines city & province

Conceptual Design Using the ER Model

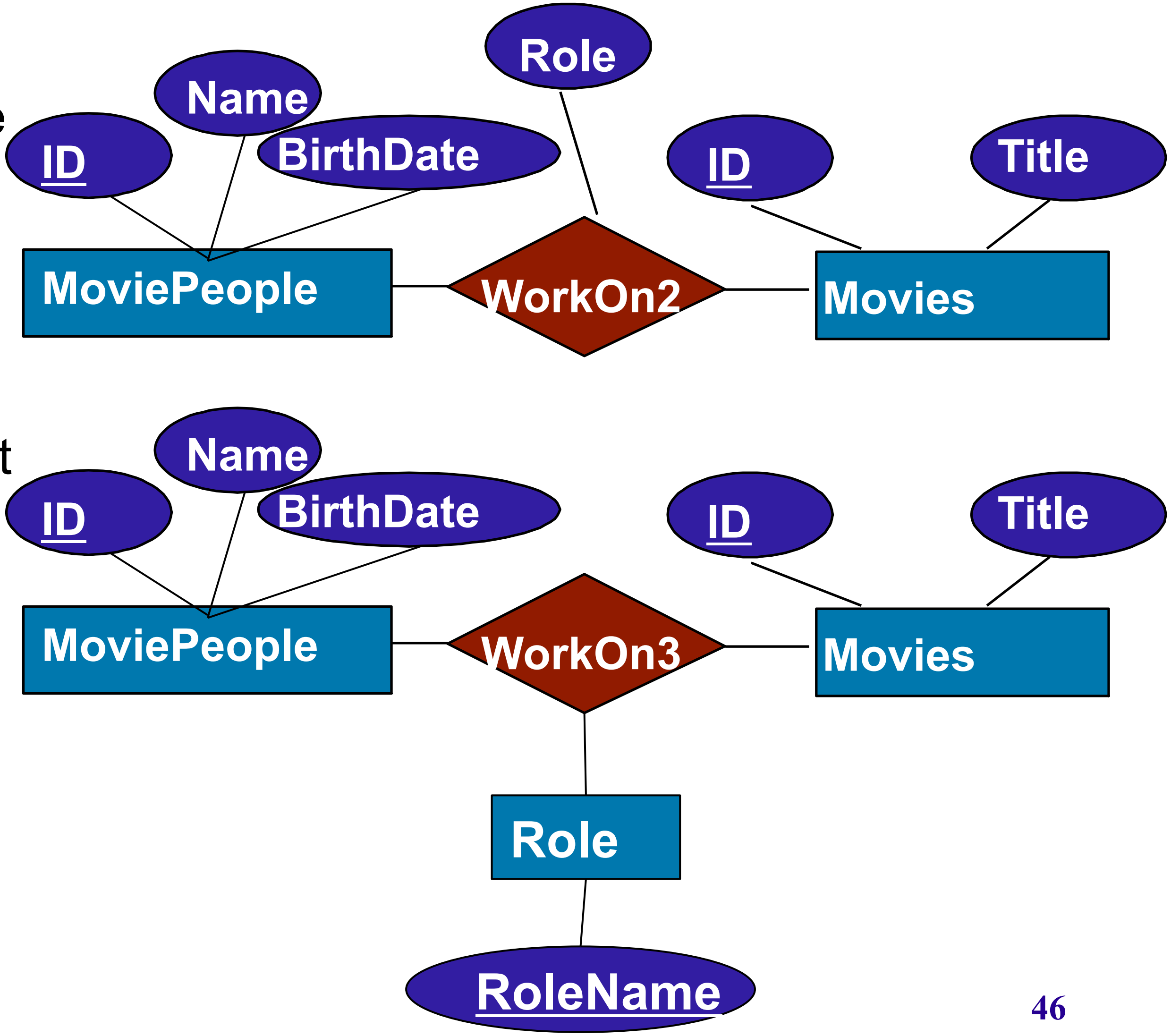
- Design choices:
  - Should a concept be modeled as an entity or an attribute?
  - Should a concept be modeled as an entity or a relationship?
  - Identifying relationships: Binary or ternary? Aggregation?
- Constraints in the ER Model:
  - A lot of data semantics can (and should) be captured.
  - But some constraints cannot be captured in ER diagrams.
    - i.e. domain constraints
    - dependencies

Entity vs. Attribute

- Should an *address* be an attribute of MoviePeople or an entity (connected to MoviePeople by a relationship)?
- Depends upon
  - the use we want to make of address information
  - the semantics of the data:
    - If we have several addresses per person, *address* must be an entity (since attributes cannot be set-valued).
    - If a person has only one street address, one city, one province, one postal code, etc. then these should simply be attributes.

Entity vs. Attribute (Cont.)

- WorkOn2 does not allow a person to have more than one role in the same movie.
- We want to associate the same pair (MoviePerson, Movie) with more than one set of values for the descriptive attributes?

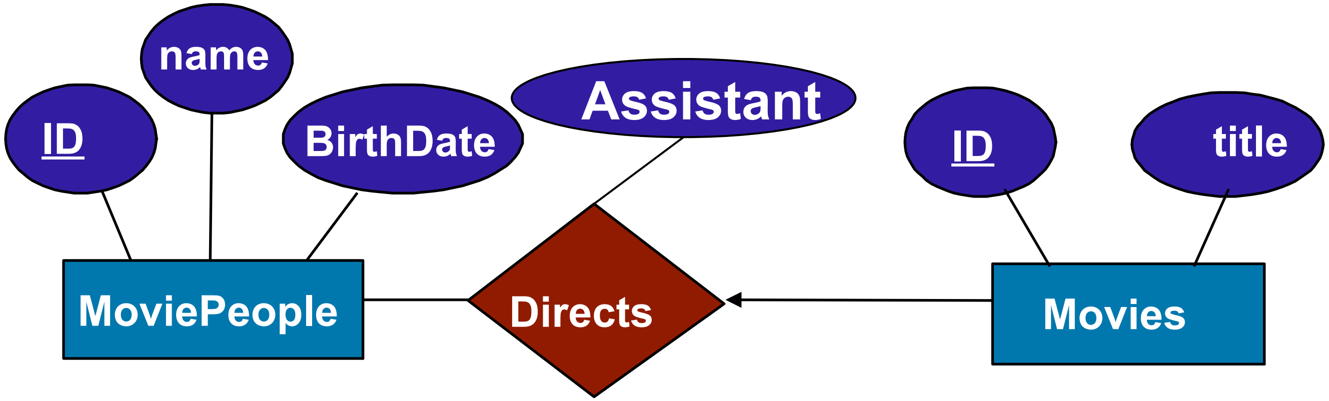


- Solution: change descriptive attributes into entities.

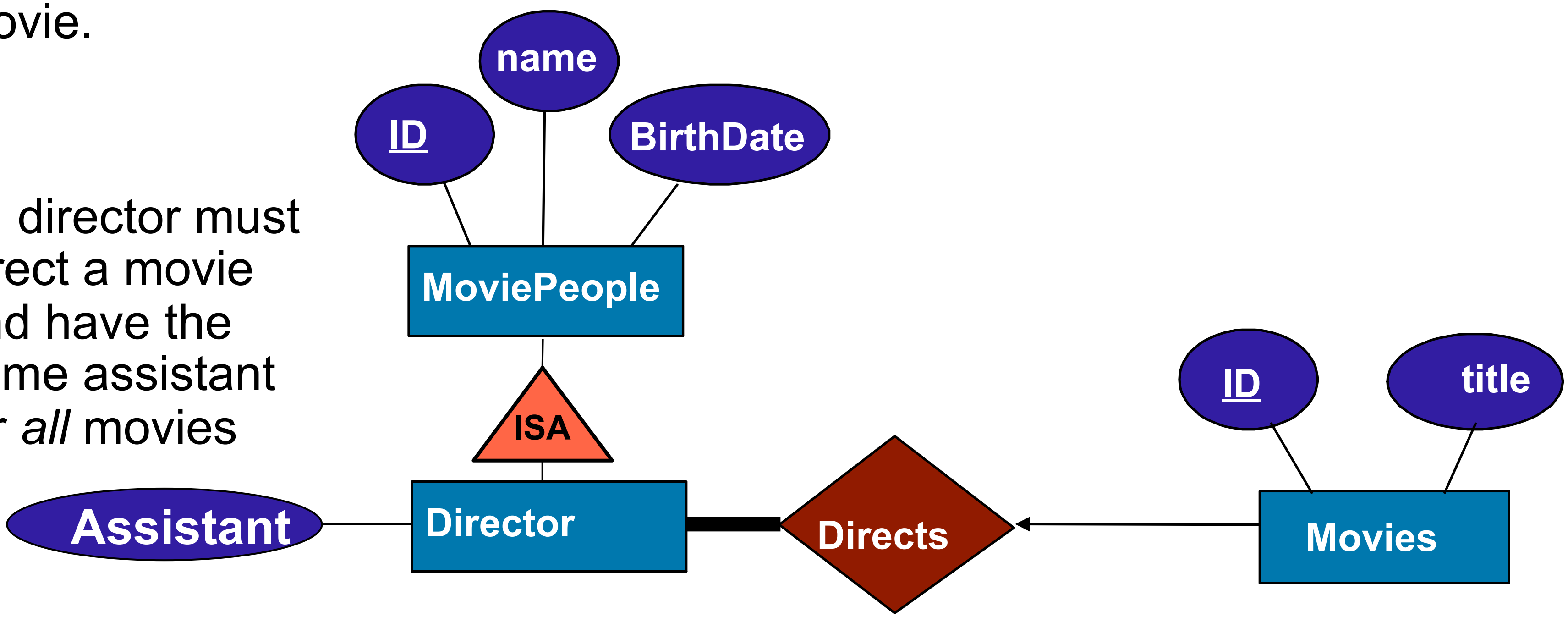
Entity vs. Relationship

- How are the two ER models different?

- Director can get a separate assistant for each movie.



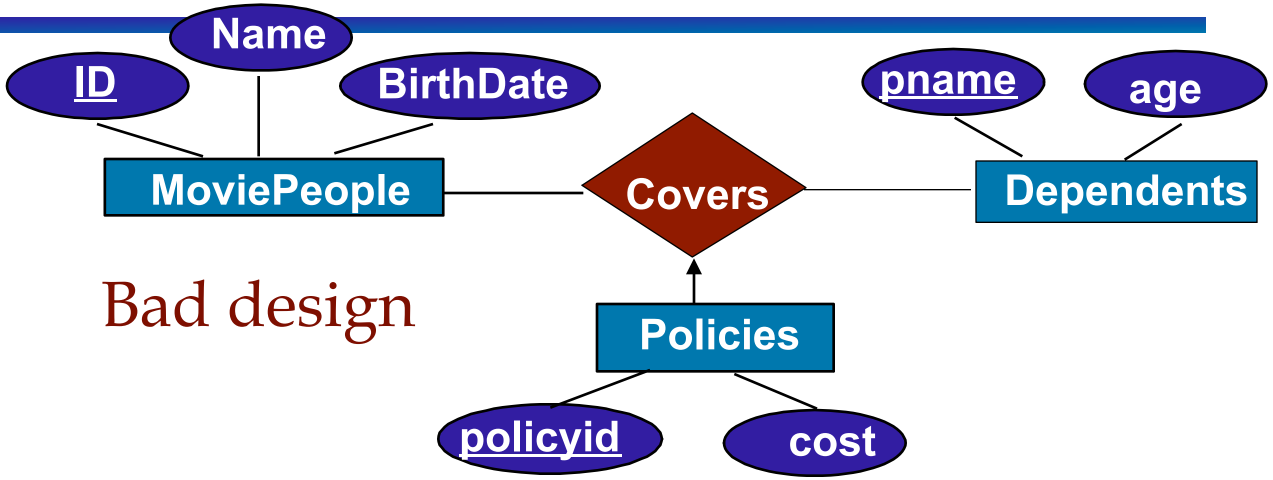
- All director must direct a movie and have the same assistant for *all* movies

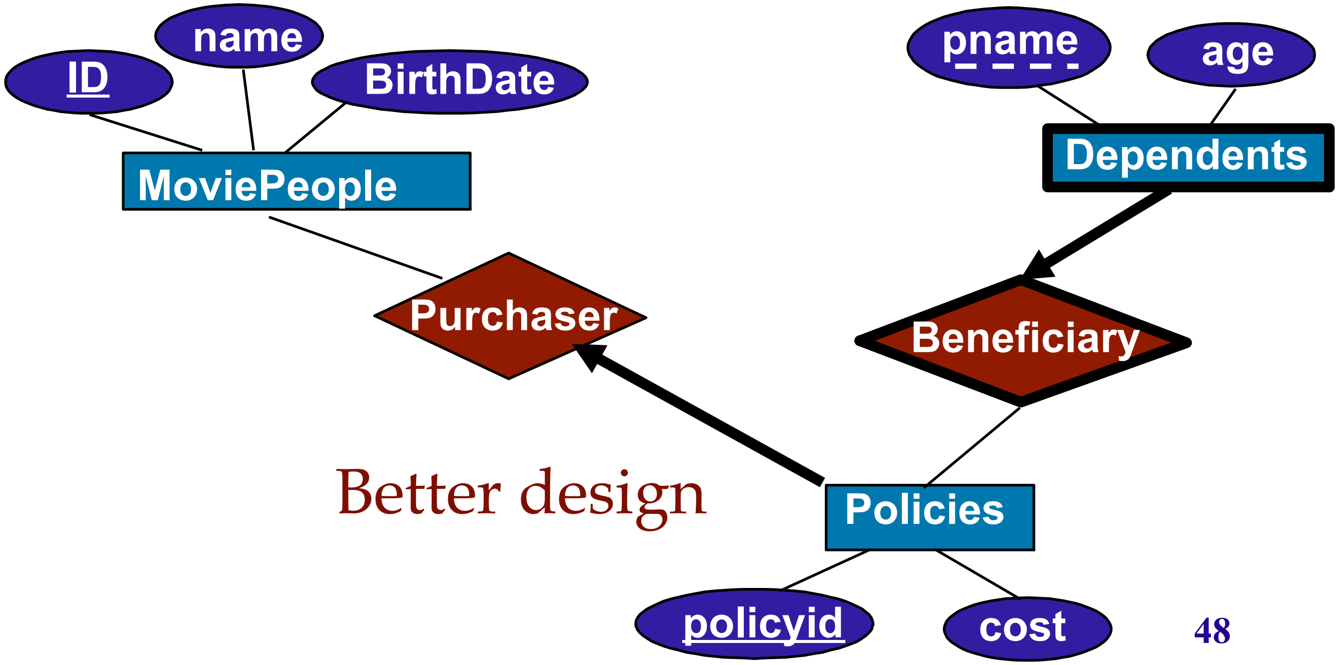


Binary vs. Ternary Relationships

- If each policy is owned by just 1 person:
  - Key constraint on Policies would mean policy can only cover 1 dependent!

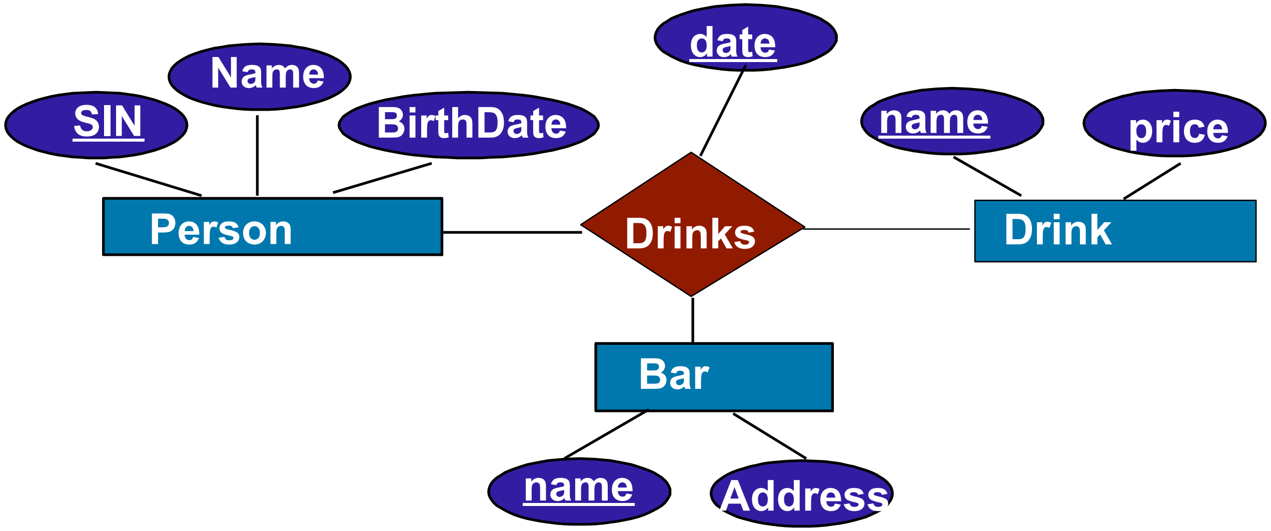
- What are the additional constraints in the 2nd diagram?





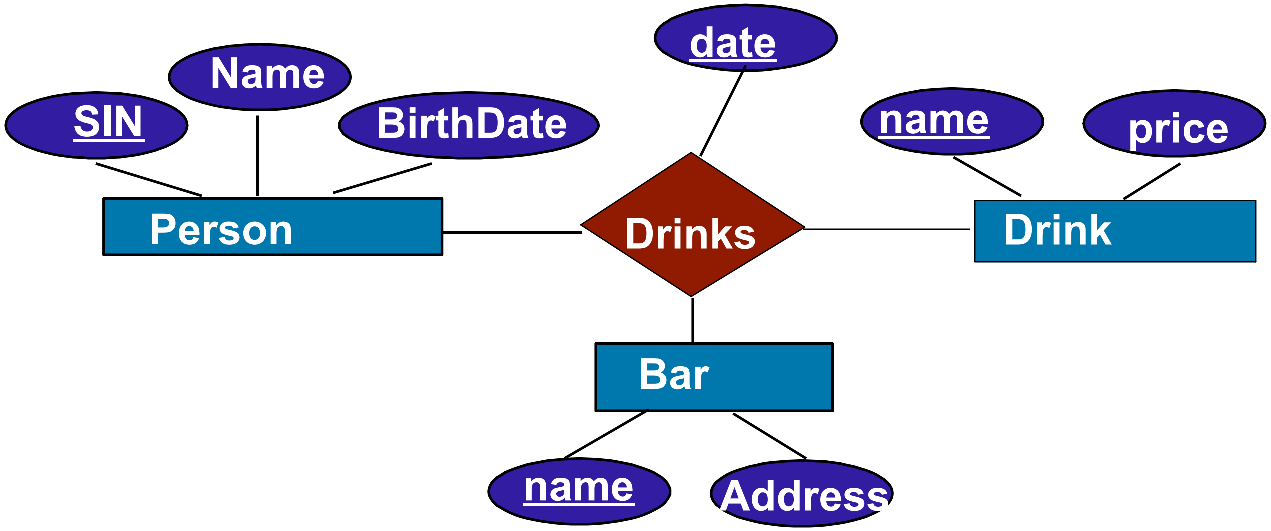
Binary vs. Ternary Relationships

- An example in the other direction: a ternary relation Drinks relates entity sets Person, Bar and Drink, and has descriptive attribute *date*.



- Can we use two of binary relationships instead?

Binary vs. Ternary Relationships vs. Aggregation



- No combination of binary relationships is an adequate substitute:
  - P "likes" D, P "visits" B, and B "provides" D does not imply that P drinks D in B.
  - Also, how would we record *date*?
- Aggregation can be used instead of a ternary relation if need to impose additional constraints:
  - I.e. a person cannot have more than one drink in the same bar

Summary of Conceptual Design

- *Conceptual design* follows *requirements analysis*,
  - Yields a high-level description of data to be stored
- ER model popular for conceptual design
  - Constructs are expressive, close to the way people think about their applications.
- Basic constructs: *entities*, *relationships*, and *attributes* (of entities and relationships).
- Some additional constructs: *weak entities*, *ISA relationships*, and *aggregation*.
- Note: There are many variations on ER model.

Summary of ER (Cont.)

- Several kinds of integrity constraints can be expressed in the ER model: *key constraints*, *participation constraints*, and *overlap/covering constraints* for ISA relationships. Some *foreign key constraints* are also implicit in the definition of a relationship set.
  - Some constraints (notably, *functional dependencies*) cannot be expressed in the ER model.
  - Constraints play an important role in determining the best database design for an enterprise.

Summary of ER (Cont.)

- ER design is *subjective*. There are often many ways to model a given scenario! Analyzing alternatives can be tricky, especially for a large enterprise. Common choices include:
  - entity vs. attribute
  - entity vs. relationship
  - binary or n-ary relationship
  - whether or not to use ISA hierarchies
  - whether or not to use aggregation
- Ensuring good database design: resulting relational schema should be analyzed and refined further.

Learning Goals revisited

- Explain the purpose of an ER diagram, and list the major components.
- Given a problem description, create an ER diagram given a specification. Justify the decisions you make for entities, relationships, keys, key constraints, participation constraints, weak entities, is-a relationships, and aggregations.
- given a problem description, identify alternative representations of the problem concepts and evaluate the choices
- compare alternative ER models for the same domain and identify their strengths and weaknesses