INFS1200/7900
Introduction to Information Systems

Relational Databases and the Entity-Relationship Model

Hassan Khosravi

Learning Outcomes

Description	Tag
Explain the purpose of an ER diagram.	
Describe the concept and notation of entities in ERs.	ER-model
Describe the concept and notation of relationships in ERs.	
Describe the concept and notation of weak entities in ERs.	
Describe the concept and notation of specialization and generalization In EERs.	
Define Cardinality Constraint and use it in ERs.	
Define Participation Constrain and use it in ERs.	
Given a problem description, create an ER diagram, justifying the decisions you make.	
Given a problem description, identify alternative representations of the problem concepts and evaluate the choices.	ER-compare-models
Compare alternative ER models for the same domain and identify their strengths and weaknesses.	

Databases

The Entity-Relationship Model

Entity and Relationship

Relationship Constraints

Specialization and Generalization

Design Choices

What is a database?

- A **database** is an organized collection of related data, usually stored on disk. It is typically:
  - – Important data
  - – Shared
  - – Secured
  - – Well-designed (minimal redundancy)
  - – Variable size

Student

sid	name	address	phone	major
99111120	G. Jones	…	…	CPSC
…	…	….	…	…

Grade

sid	dept	course#	mark
99111120	CPSC	122	80
…	…	….	…

What is a database?

- A **database** is an organized collection of related data, usually stored on disk. It is typically:
  - Important data
  - Shared
  - Secured
  - Well-designed (minimal redundancy)
  - Variable size
- A DB typically models some real-world ***enterprise***
  - Entities (e.g., students, courses)
  - Relationships (e.g., Ting got 95% in CPSC 221)

Who watches the watchers?

- A ***Database Management System (DBMS)*** is a software system designed to store and manage databases. It is used to:
  - Define, modify, and query a database
  - Control access
  - Permit concurrent access
  - Maintain integrity
  - Provide loading, backup, and recovery
  - Provide support for decision making

Databases

The Entity-Relationship Model

Entity and Relationship

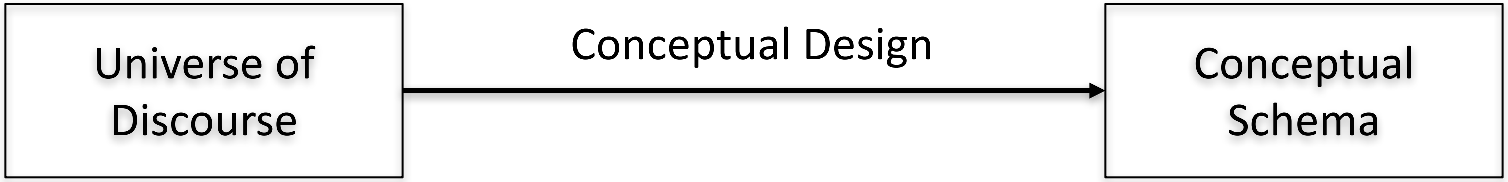
Relationship Constraints

Specialization and Generalization

Design Choices

Conceptual Database Design

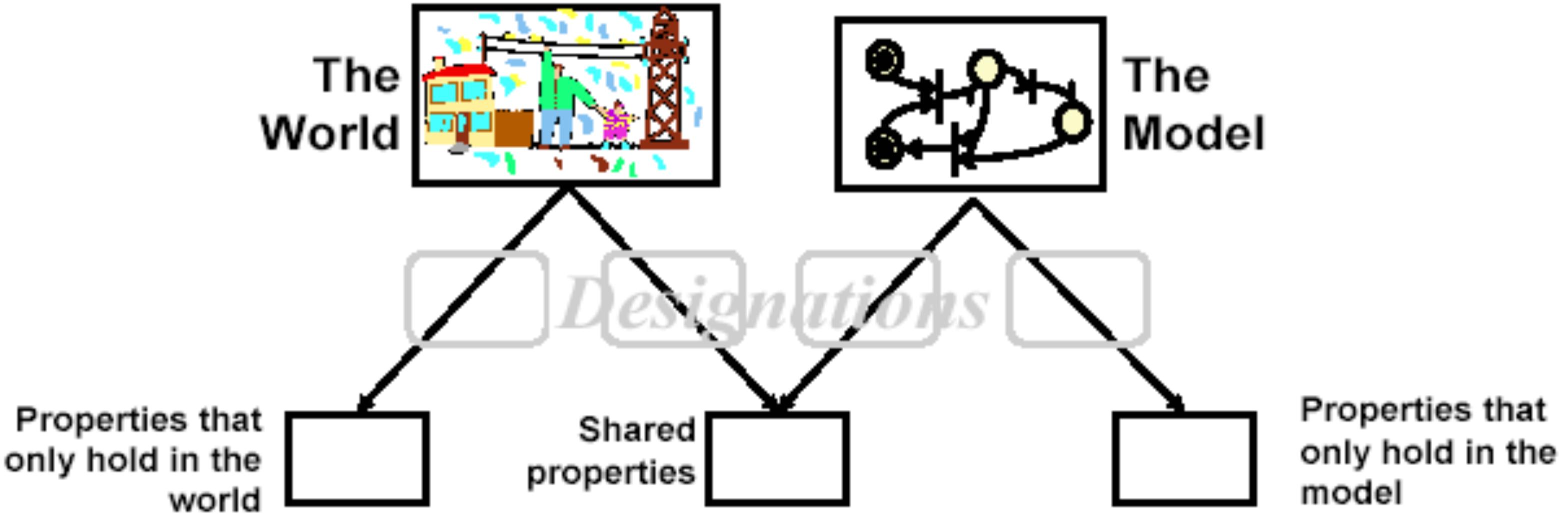
- Conceptual database design is a very important phase in designing a successful database application.
- Step 1: identify the "Universe of Discourse" (UoD)
  - The database to be built will not model everything in the world, but rather some "mini-world" or "Universe of Discourse".
- Step 2: Convert the UoD to a conceptual schema, which can be captured by a database.



- In relational databases, the conceptual schema is generally encoded in an **Entity-Relationship (ER) Diagram**

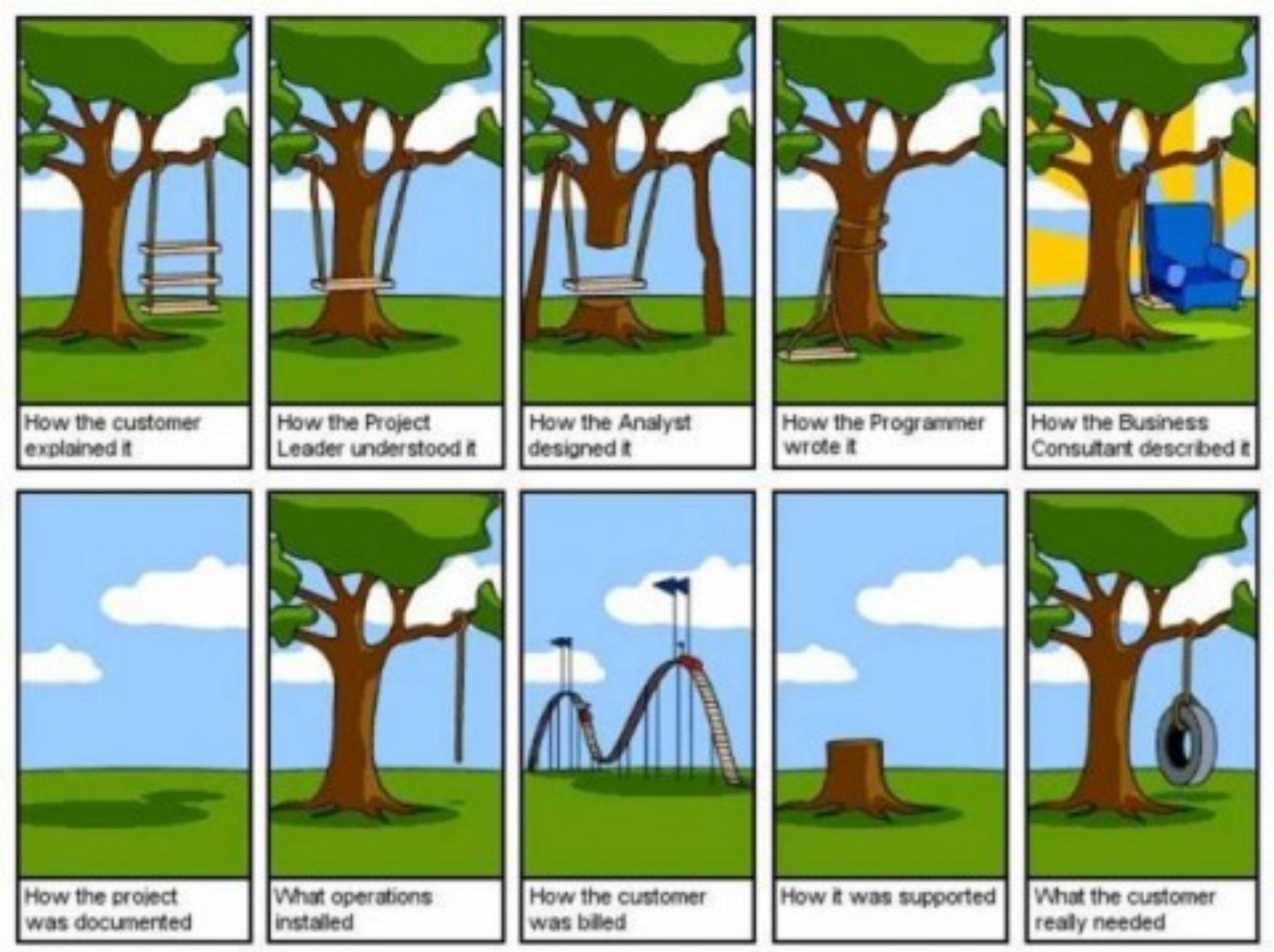
Dangers of Modelling

- A Model is never perfect
  - Phenomena in the model that are not present in the application domain
  - Phenomena in the application domain that are not in the model


The World
The Model
Designations
Properties that only hold in the world
Shared properties
Properties that only hold in the model

purpose: maximize this part

www.cs.toronto.edu/~sme/CSC2106S/


How the customer explained it
How the Project Leader understood it
How the Analyst designed it
How the Programmer wrote it
How the Business Consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
What the customer really needed

Entity-Relationship (ER) Diagram

- An Entity-Relationship (ER) diagram is a graphical data modeling technique that represents the main entities and their corresponding relationships within a system or an enterprise.

- ER diagrams can help define business processes and are used as the foundation for a developing conceptual database designs.

Databases

The Entity-Relationship Model

Entity and Relationship

Relationship Constraints

Specialization and Generalization

Design Choices

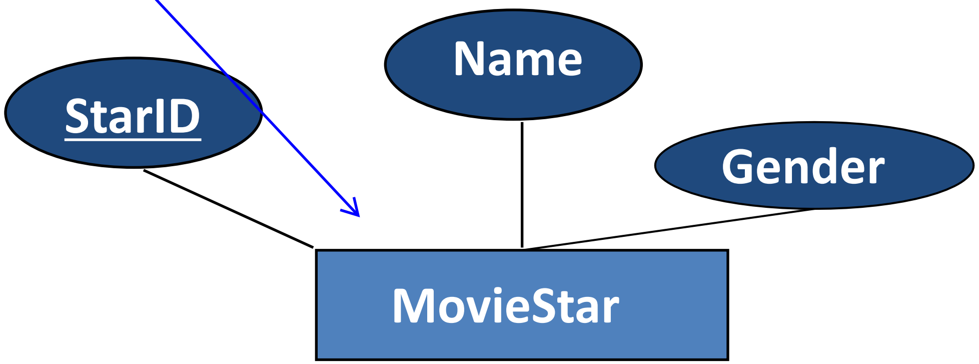
Entity

- ***Entity:*** Real-world object distinguishable from other objects (e.g., a movie star). An entity is described using a set of attributes.

- The same entity may have different prominence in different UoDs
  - The salary of a movie star may play an important role in one database application and may not be of importance in another application.

Entity type

- ***Entity Type:*** Defines set of entities that have the same attributes (e.g. moviestar). Each Entity Type is described by its **NAME** and attributes.



similar to type of variables

- Each attribute has a ***domain***. *(e.g.., float, date, int)*
- Each entity set has a ***key***. e.g. in Uni, key for student is student ID

Entity Set

- ***Entity Set*:** The collection of all entities of a particular entity type in the database at any point in time
  - E.g., a set of movie stars
  - An entity set can easily be mapped to a table.

StarID	Name	Gender
1	Harrison Ford	Male
2	Vivian Leigh	Female
3	Judy Garland	Female

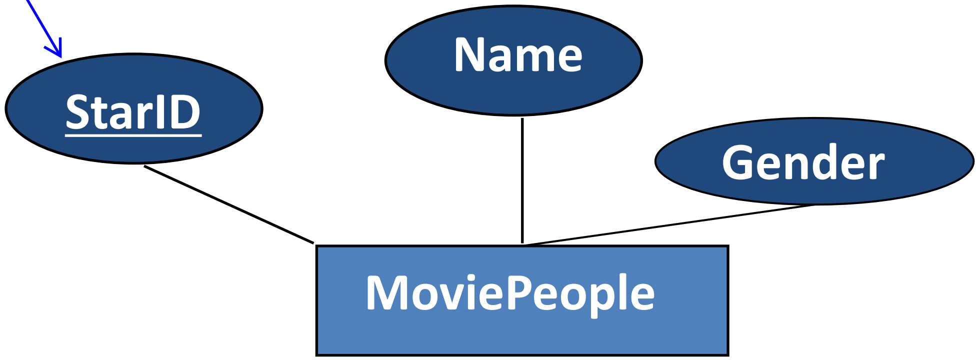
Key Attributes

- 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 e.g. student ID, course code

- Only primary keys are shown in ER diagrams

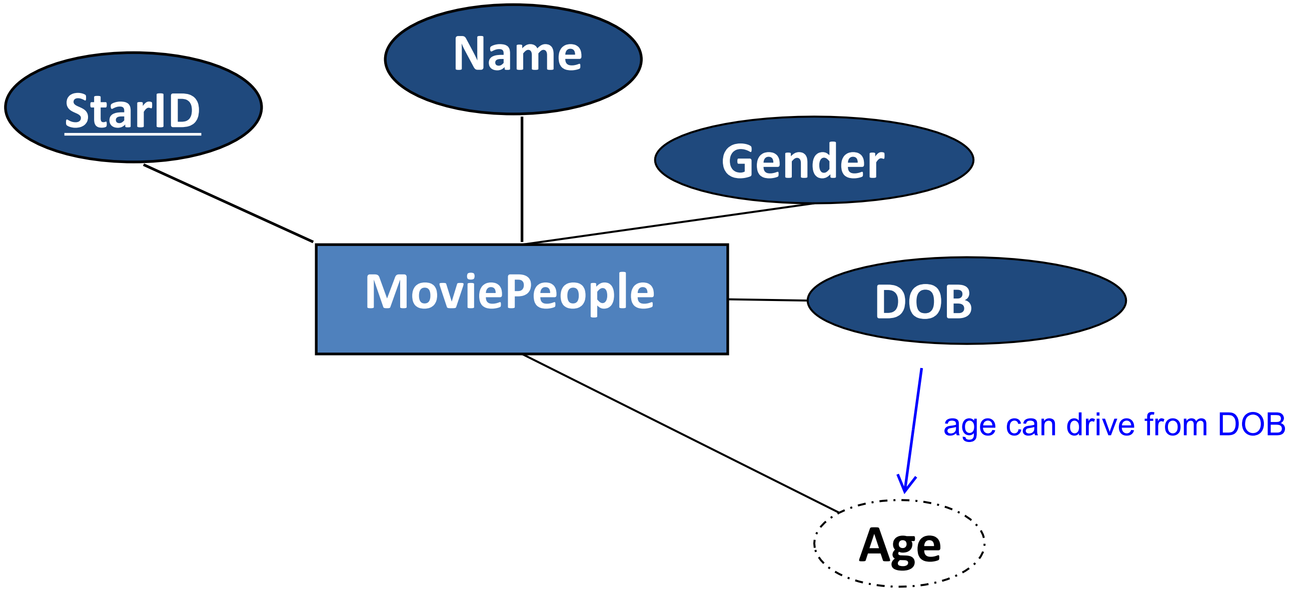
using underline



- We'll discuss superkeys when we consider normal forms (for now, don't worry about them)

Derived Vs. Stored Attributes

- Some attribute values can be derived from attributed values of *related entities*



Null Valued Attributes

- A particular entity may not have an applicable value for an attribute
  - Tertiary-Degree: Not applicable for a person with no university education
  - Home-Phone: Not known if it exists
  - Height: Not known at present time
- Type of Null Values
  - Not Applicable
  - Unknown
  - Missing

Composite Vs. Simple Attributes

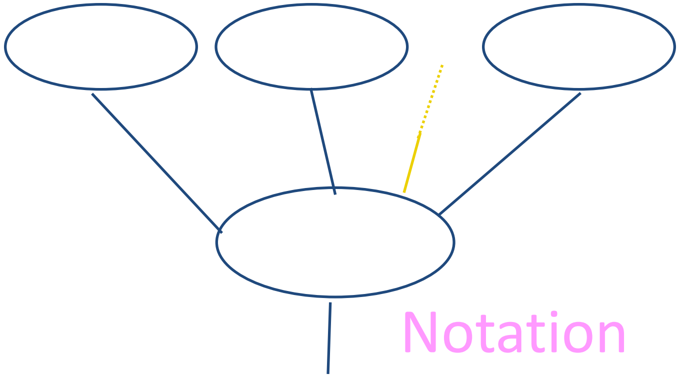
- Composite attributes can be divided into smaller parts which represent simple attributes with independent meaning (e.g. simple attributes)

  - Simple Attribute: Aircraft-Type
  - Complex Attribute: Aircraft-Location which is comprised of

    Aircraft-Latitude

    Aircraft-Longitude

    Aircraft-Altitude



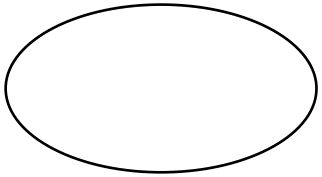
Notation

Single Vs. Multivalued Attributes

Simple attributes can either be single-valued or multi-valued

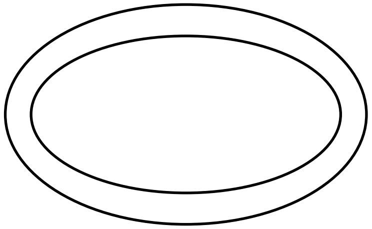
- Single-valued: Gender = F

Notation



- Multivalued: Degree = {BSc, MInfTech}

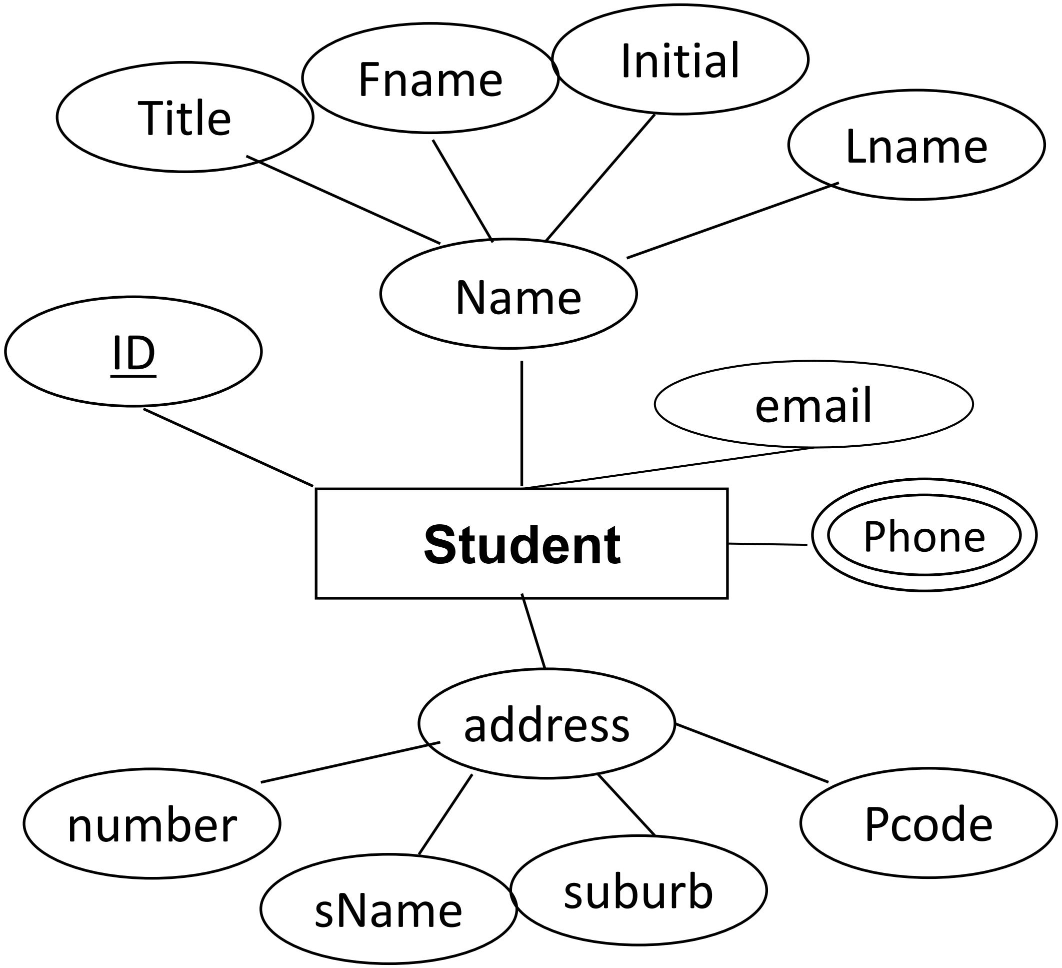
Notation



In-class Exercise

Every student has a unique id, name (composed of title, first name, middle initial and last name), email, address (composed of number, streetname, suburb and postcode), and one or more phone numbers. Draw an ER diagram for the above description.

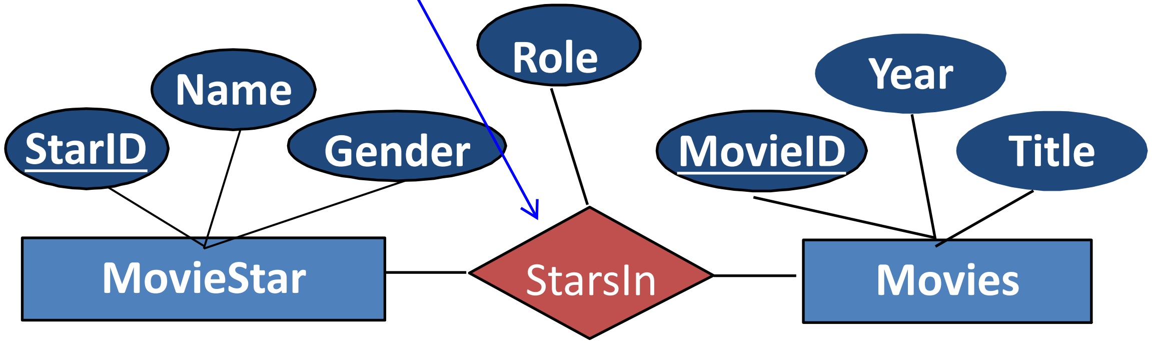
In-class Exercise - Solution



Relationship Types and Sets

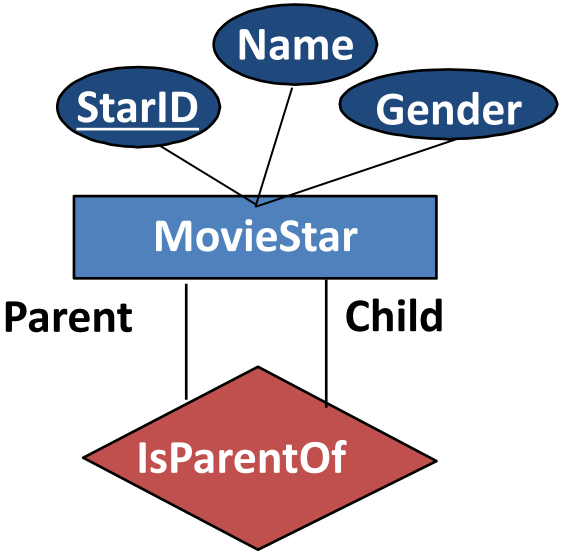
- ***Relationship:*** Association among two or more entities.
  - E.g., George Clooney starred in Gravity.

- ***Relationship Type*** defines the relationship and a ***Relationship Set*** represents the set of relationships of a particular type at any point in time.



Recursive Relationships

- Same entity type can participate more than once in the same relationship type under different "roles"Such relationships are called "**Recursive Relationships**"
  - e.g., Kirk Douglas isParentOf Michael Douglas)



- A relationship set may have ***descriptive attributes*** (like since).

Relationship Degree

- The degree of a relationship type is the number of participating entity types
- 2 entities: Binary Relationship
  3 entities: Ternary Relationship
  n entities: N-ary Relationship
- Same entity type could participate in multiple relationship types

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.

Clicker Question

- 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

Clicker Question

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

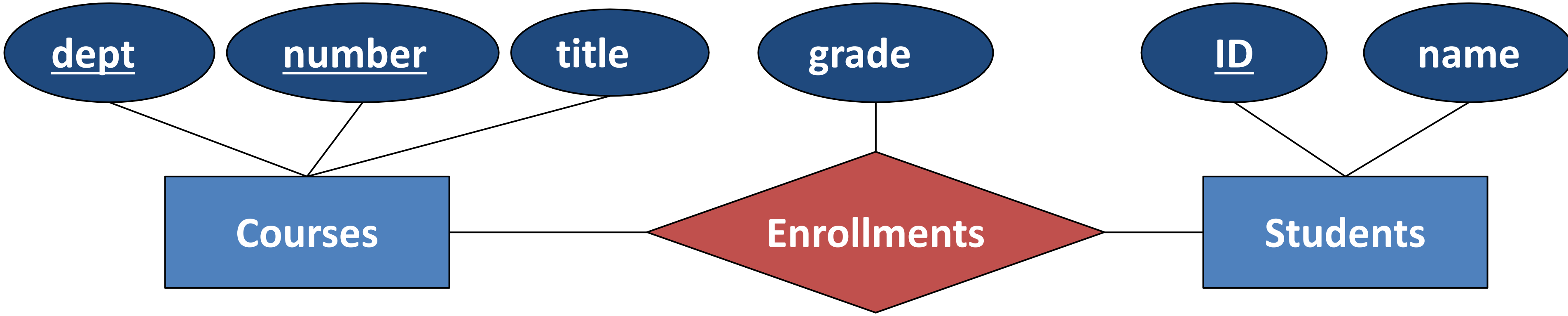
B. Entity set Students with attributes ID and name underlined.

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?

Databases

The Entity-Relationship Model

Entity and Relationship

Relationship Constraints

Specialization and Generalization

Design Choices

Relationship Constraints

- Constraints on the relationship type limit the possible combination of entities that may participate in the corresponding relationship set
  - For example, what if we wanted to add instructors to courses and restrict that each could only have one instructor?

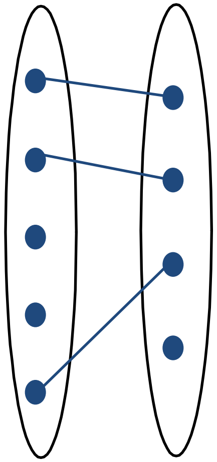
- Two kinds of constraints can be defined in the ER Model
  - Cardinality Constraint
  - Participation Constraints

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:

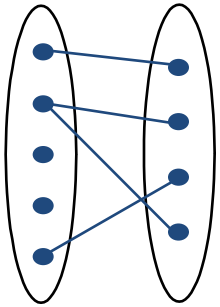
1. *one-to-one* from A to B:
   - an entity in A is associated with at most one entity in B and vice versa
   - e.g. A: driver, B: driver's license



1-to-1

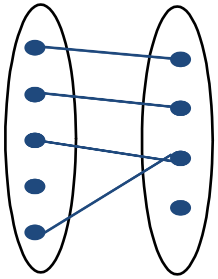
Cardinalities

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

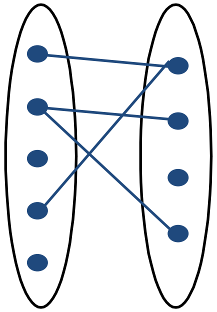
3. ***many-to-one*** *from A to B: switch A and B above*



Many-to-1

Cardinalities

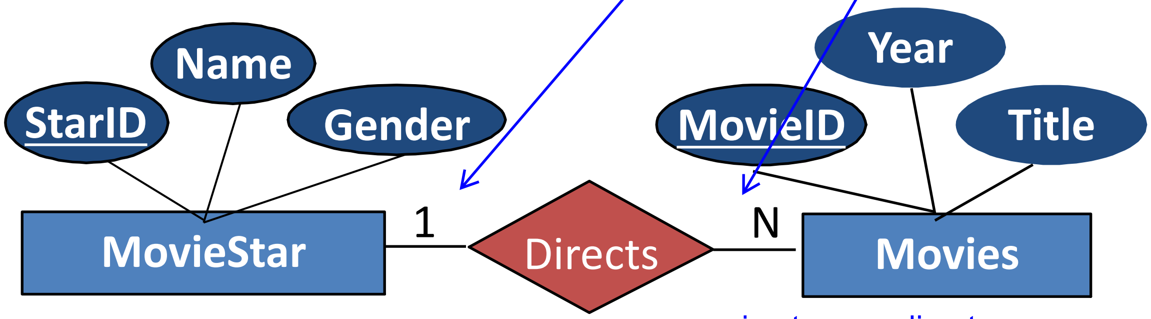
- ***many-to-many*** *from A to B:*
  - *an entity in A is associated with any number of entities in B and vice versa*
  - *e.g. A: students, B: courses*



Many-to-Many

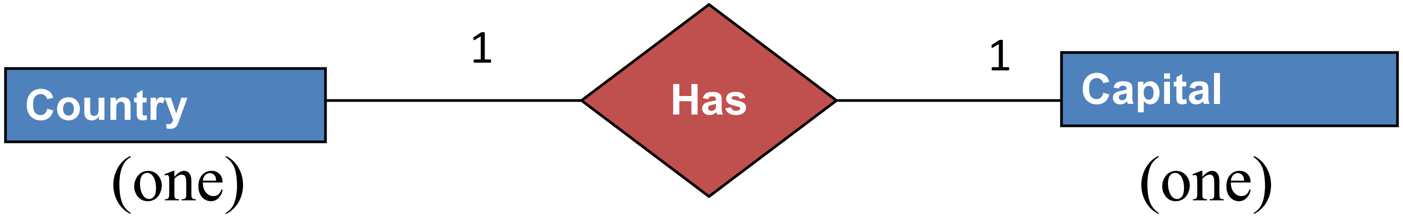
Key Constraints

- The restriction imposed by a 1-to-1 and 1-to-many ratios are examples of ***key constraints***.
- Important on insertions



movie star can directs many movies, movies have only one directors

or

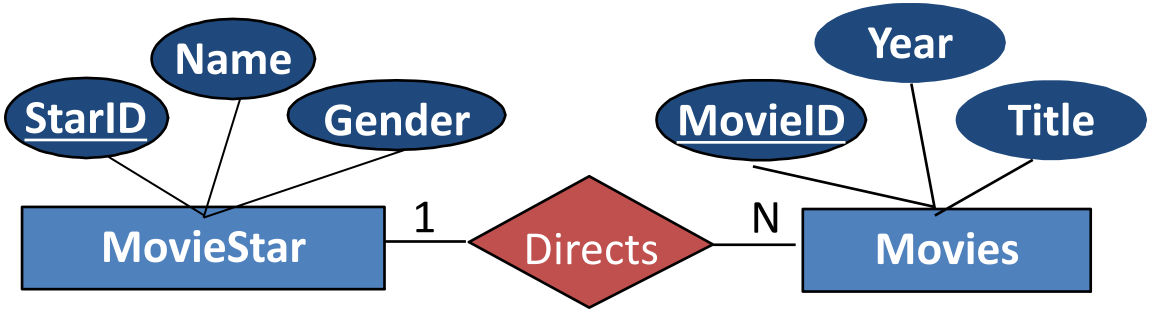


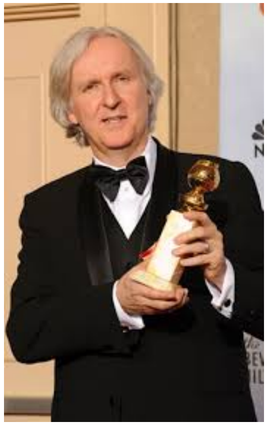
(one)

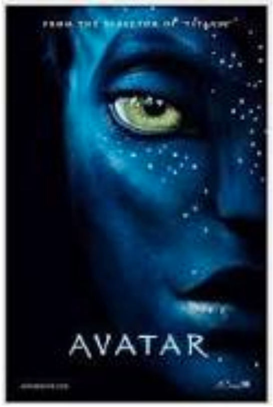
(one)

Example

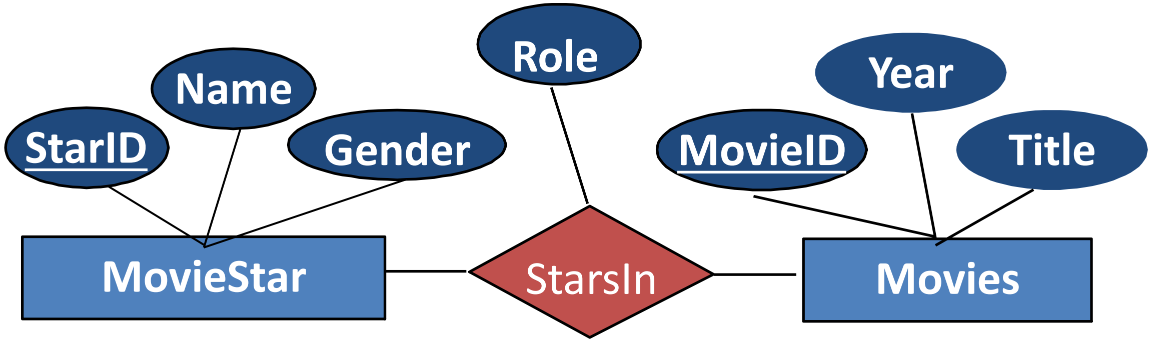
- In this example, if you know the movie, you would be able to determine the director



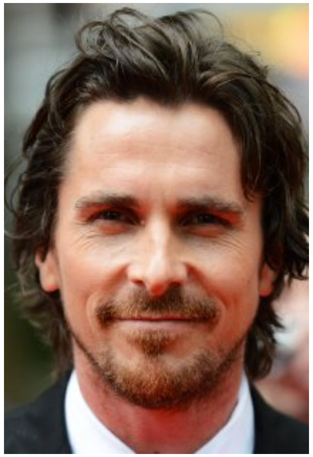




Key in Many-to-Many Relationships



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



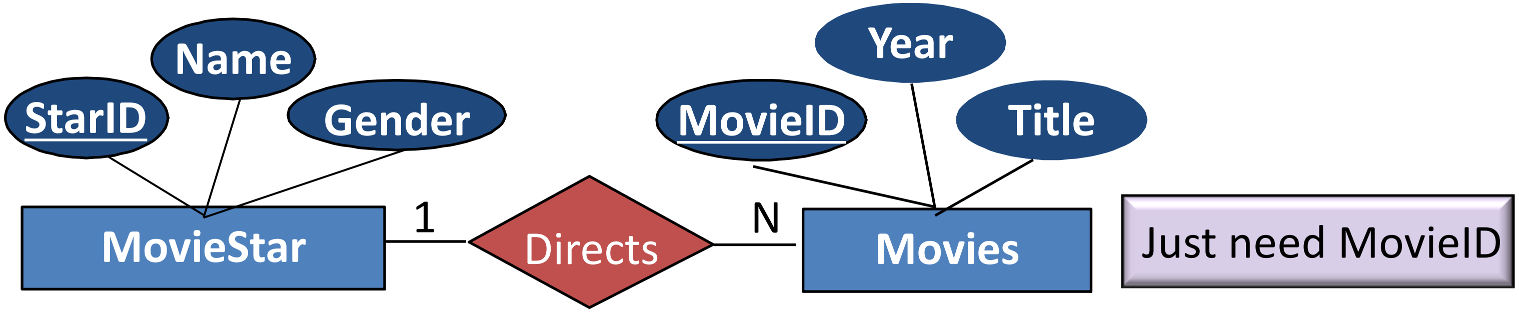
Christian Bale
as
Bruce Wayne

Need StarID and MovieID

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

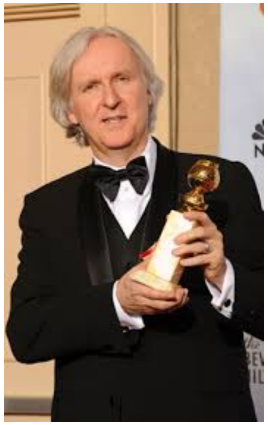
Can the same person have multiple roles?

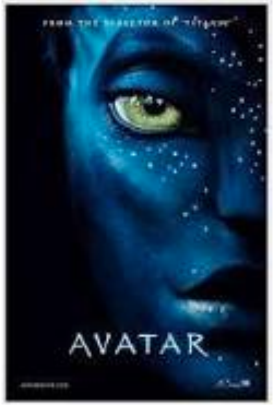
Key in One-to-Many Relationships



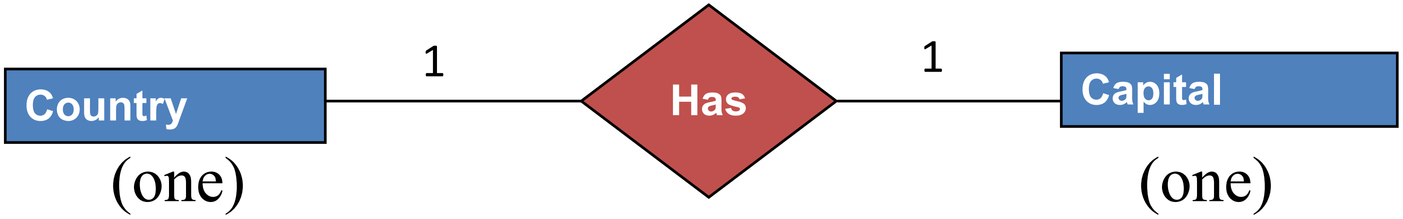
Just need MovieID

movie star to movies
1 to N





Key in One-to-One Relationships



Need either one

Ottawa

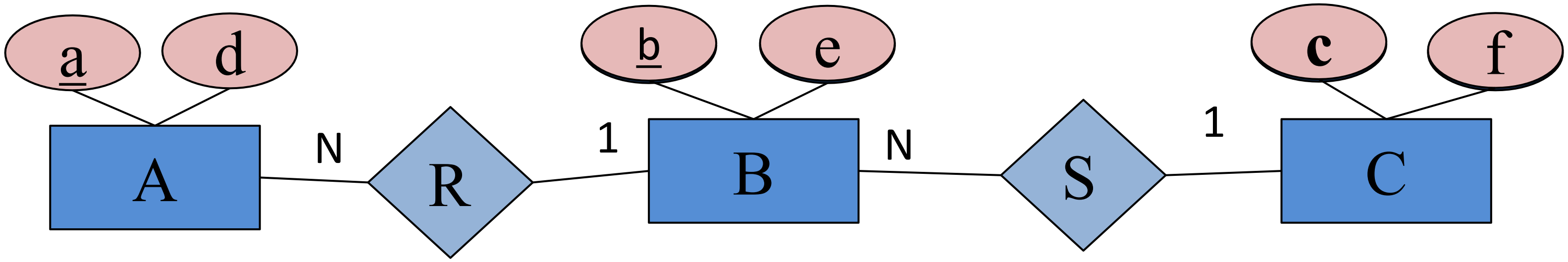
Primary Keys of Relationship Sets

- 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.

Clicker Exercise



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)}

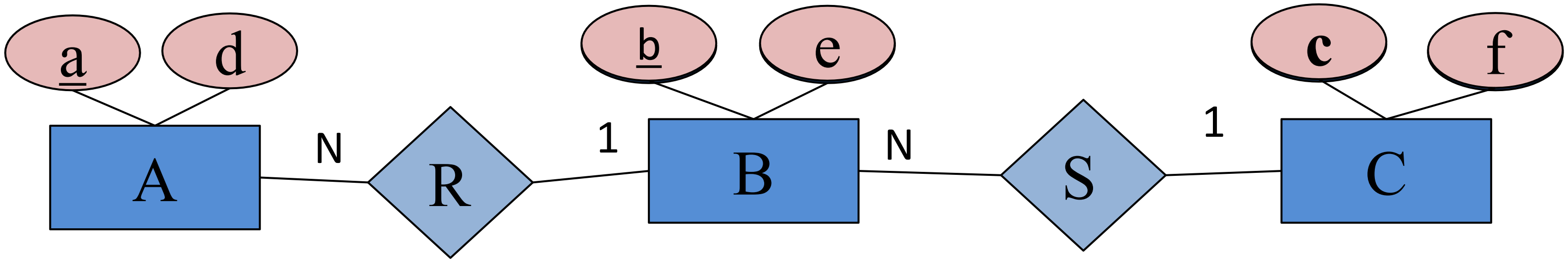
B. R = {}; S = {(b1, c2), (b2, c2)}

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

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

E. None of the above

Clicker Exercise



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)

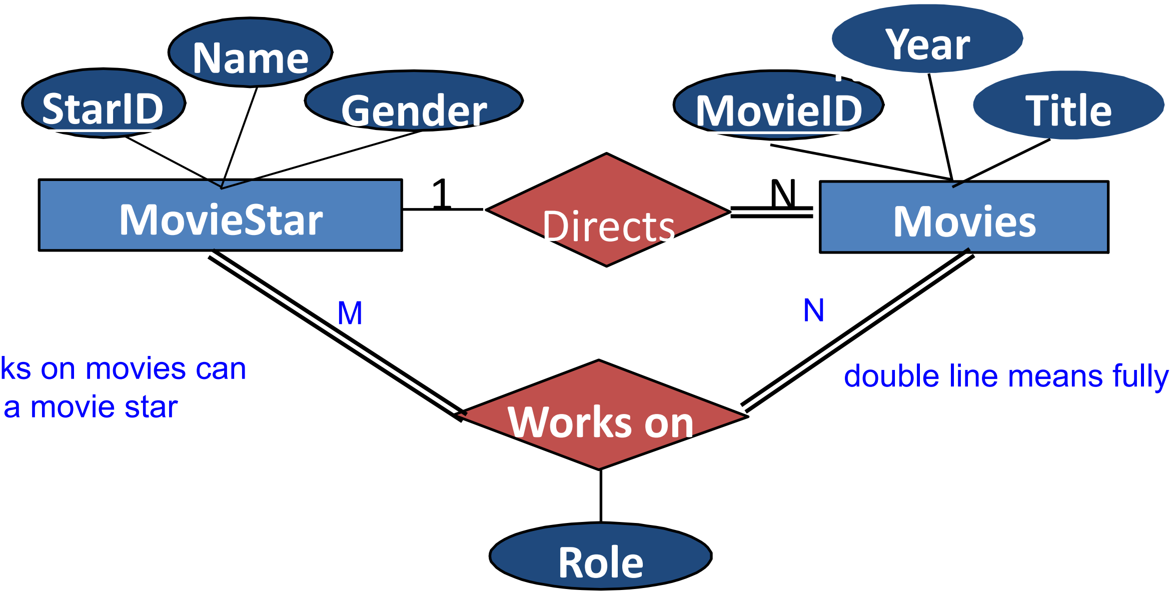
Correct

B defines C (S)

A defines B (R)

Participation Constraints

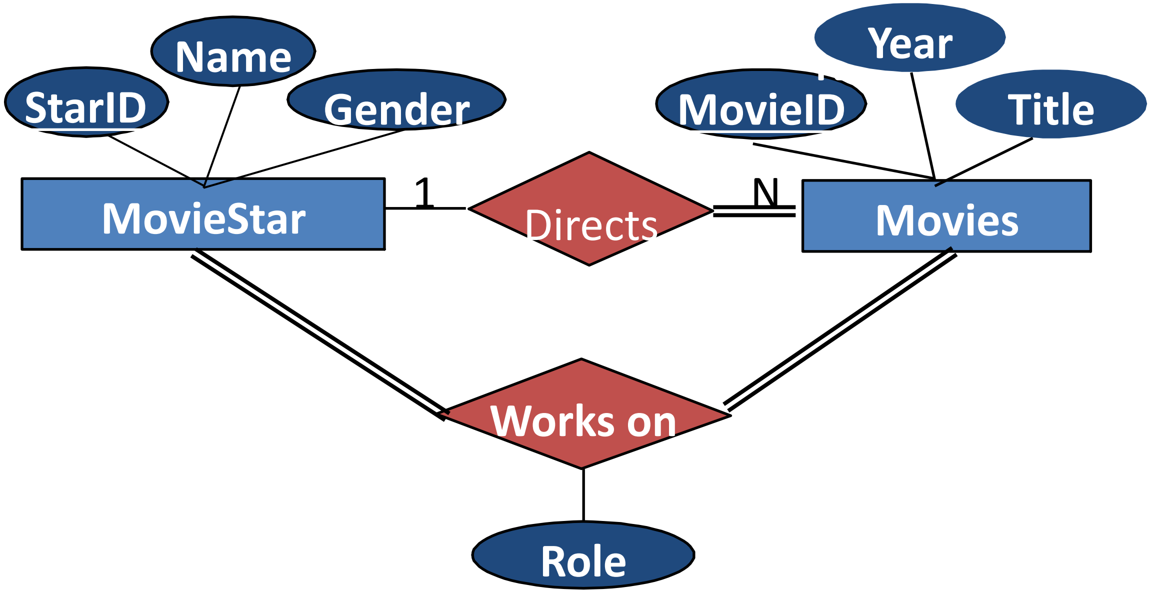
- Participation: Indicates if all entities participate in the relationship.
- An entity's participation can be ***total*** or ***partial*** .
- Requiring total participation is a ***participation constraint*** and it is shown with a double line
  - Important on deletions
  - i.e., participation of Movie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



only works on movies can become a movie star

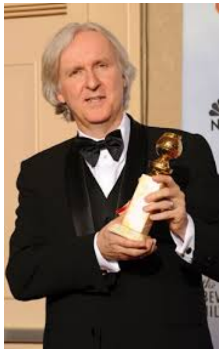
double line means fully participated

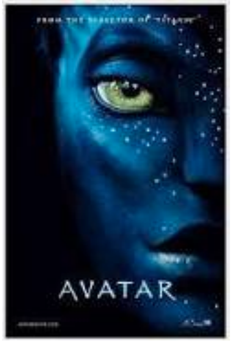
Why is Participation Constraint Important



Would I be able to delete James Cameron without deleting Avatar? No

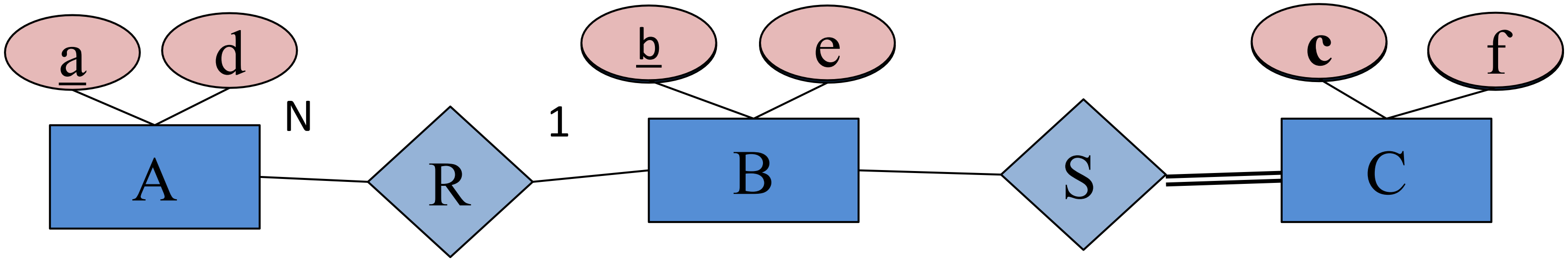
Would I be able to delete Avatar without deleting James Cameron? Yes





Maybe
as if James Cameron
has only one movie:
Avatar, then the
answer should be
wrong

Clicker Exercise



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 = {}

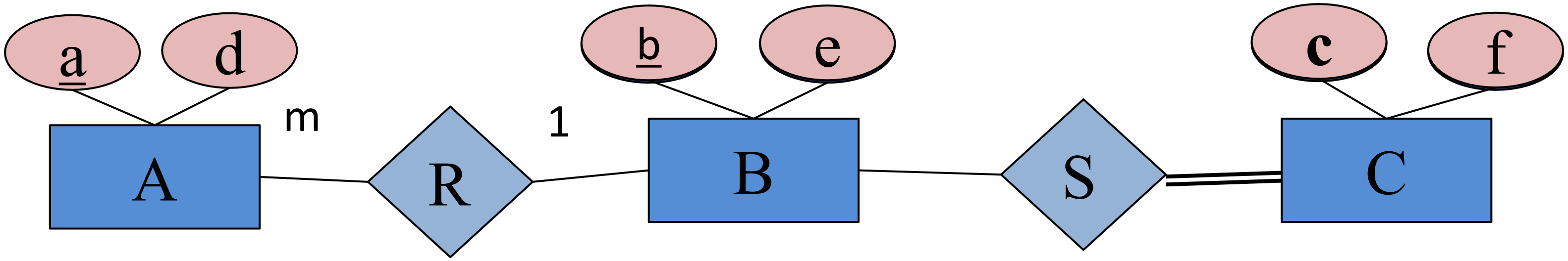
B. R = {(a1,b1)}, S = {(b2,c2)}

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

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

E. None of the above

Clicker Exercise



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 = {}

B. R = {(a1,b1)}, S = {(b2,c2)}

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

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

E. None of the above

C must participate

C must participate

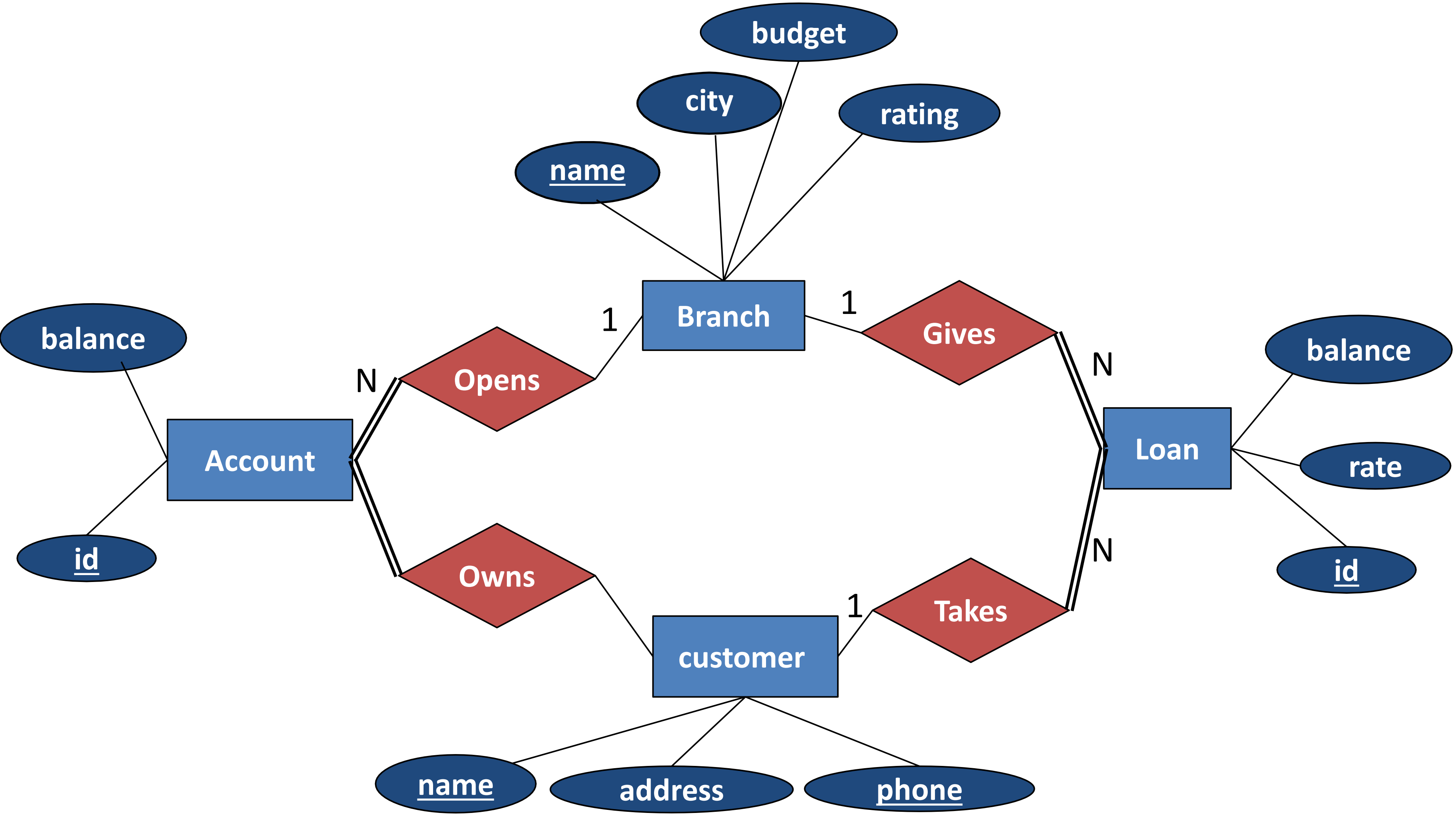
A → B (R)

Correct

Exercise: ABC Banks

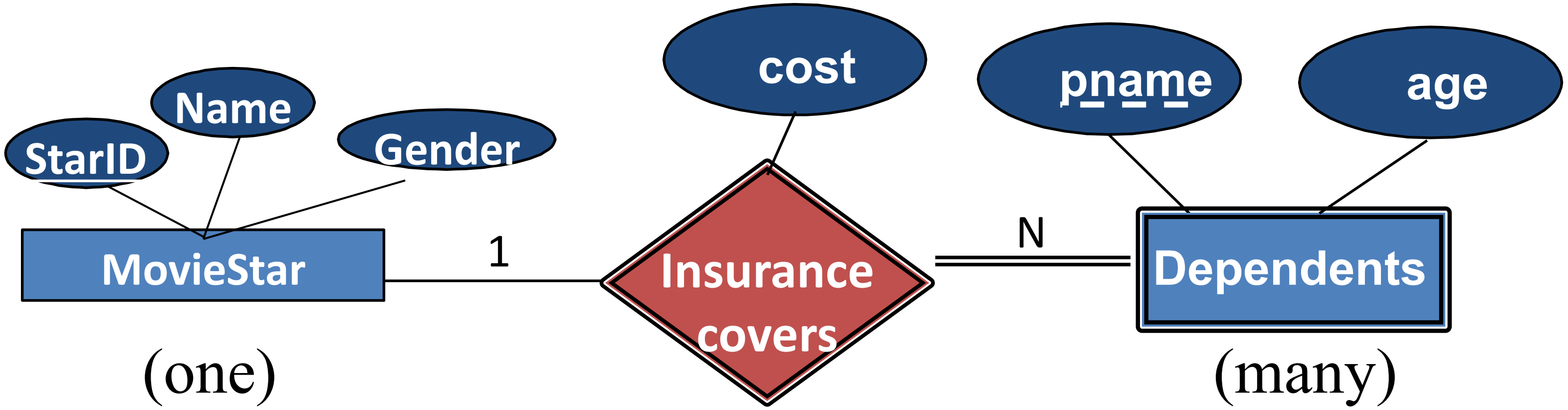
- The ABC bank is organized into branches. Each branch is located in a particular city and is identified by a unique name. Each year the bank' board defines the yearly budget and the rating (which is a number from 1 to 10) for each branch.
- A bank customer is identified by their customer name and phone 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



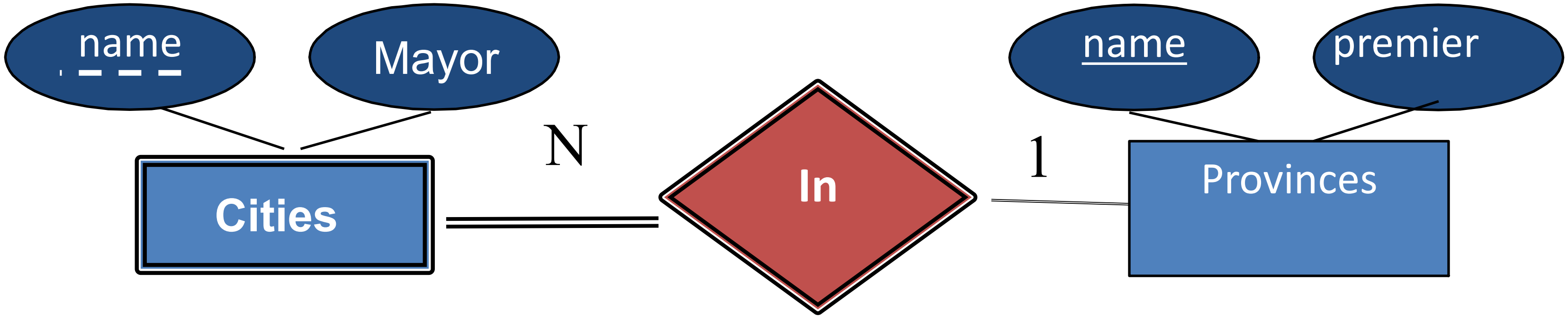
Weak Entities

- 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

name of provinces but not name of premier



Which of the following is necessarily true:

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

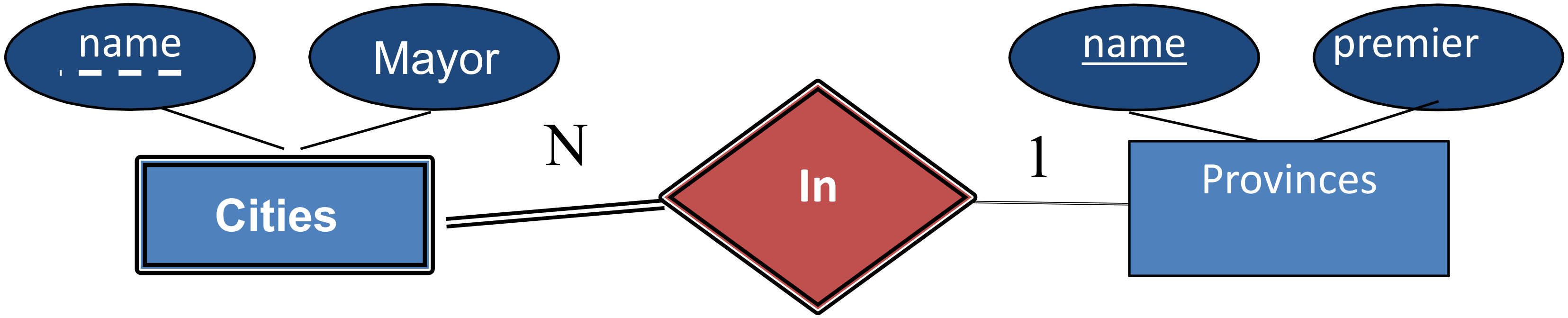
B. No two cities can have mayors with the same name.

C. No two cities can have the same name. ???? listen record

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

Databases

The Entity-Relationship Model

Entity and Relationship

Relationship Constraints

Specialization and Generalization

Design Choices

Specialization and Generalization

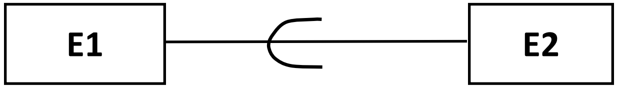
- Specialisation from abstract things to details
  - Define a number of subclasses of an entity type.
  - Each subclass contains a subset entities of the superclass.
  - A subclass is defined based on more specific distinguishing characteristic on entities of the super class.

- Generalisation from details to abstract things
  - Opposite process to specialisation.
  - Abstraction process of ignoring differences amongst some entity types (subclasses) and generalise them into a superclass.

Extended ER (EER)

- Entity Type is called class in EER
- Class can be Superclass, Subclass
- Entities in the same class have the same attributes
- Attributes of a superclass are inherited by the subclasses.
- Subclass can have its own specific attributes
- Subclass can have its own specific relationships
- Every entity in a subclass is a member of its super class(es)

Sub/Super Classes

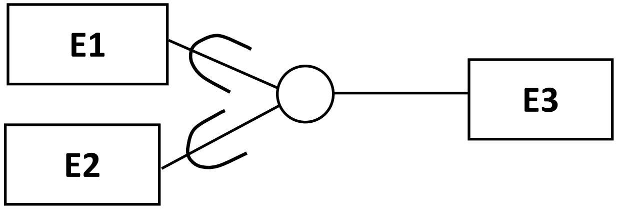
- Entity types refined into sub-classes and super-classes
- Notation: 

- Subclass entity types inherit
  - attributes
  - relationships

from superclass entity type

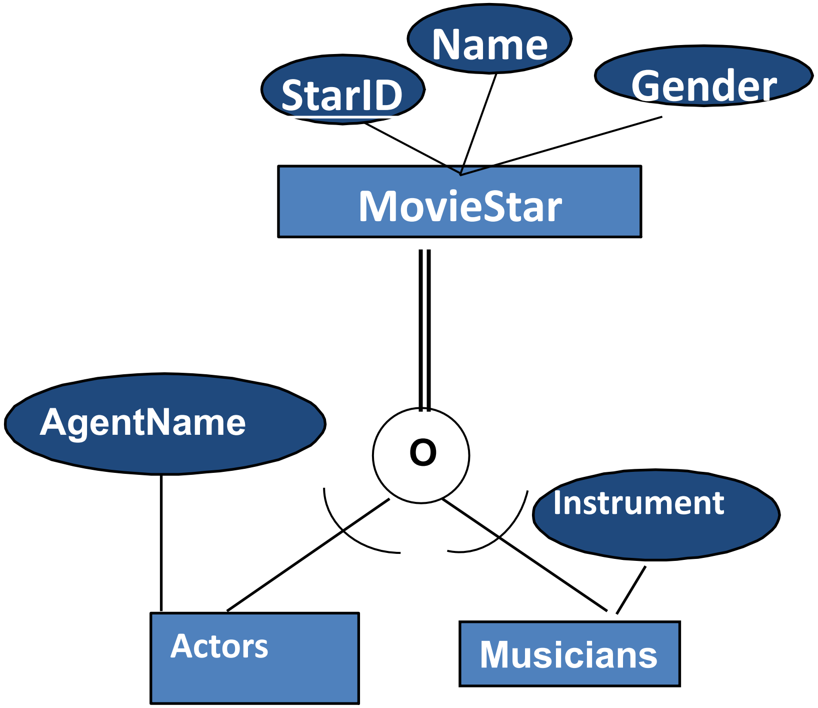
Specialization/Generalization

- Top Down vs Bottom Up
  - Sub classes are specializations of superclass
  - Superclass is generalization of subclasses
- Allows us to model:
  - attributes only applicable to entity subclasses
  - relationships only played by entity subclasses

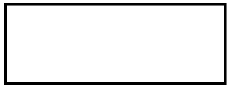


Constraints

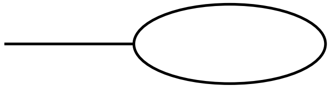
- Specialization may be
  - total ═══
  - partial ───
- Subclass sets may be
  - overlapping ⓞ
  - disjoint ⓓ



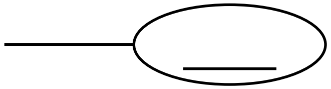
Notation Guide



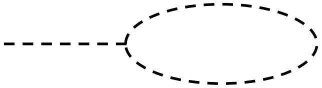
ENTITY TYPE



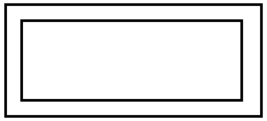
ATTRIBUTE



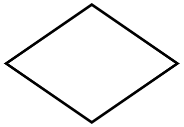
KEY ATTRIBUTE



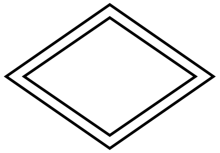
DERIVED ATTRIBUTE



WEAK ENTITY TYPE

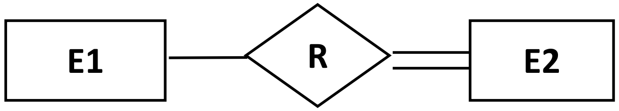


RELATIONSHIP TYPE

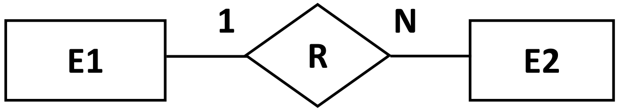


IDENTIFYING RELATIONSHIP TYPE

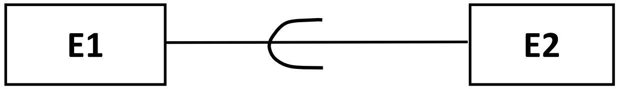
Notation Guide



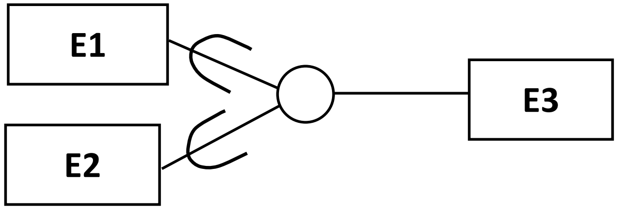
TOTAL PARTICIPATION OF E2 IN R



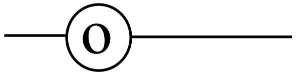
CARDINALITY RATIO 1:N FOR E1:E2 IN R



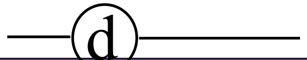
E1 IS A SUBCLASS OF E2



E1 and E2 ARE SUBCLASSES OF E3



Overlapping specialization



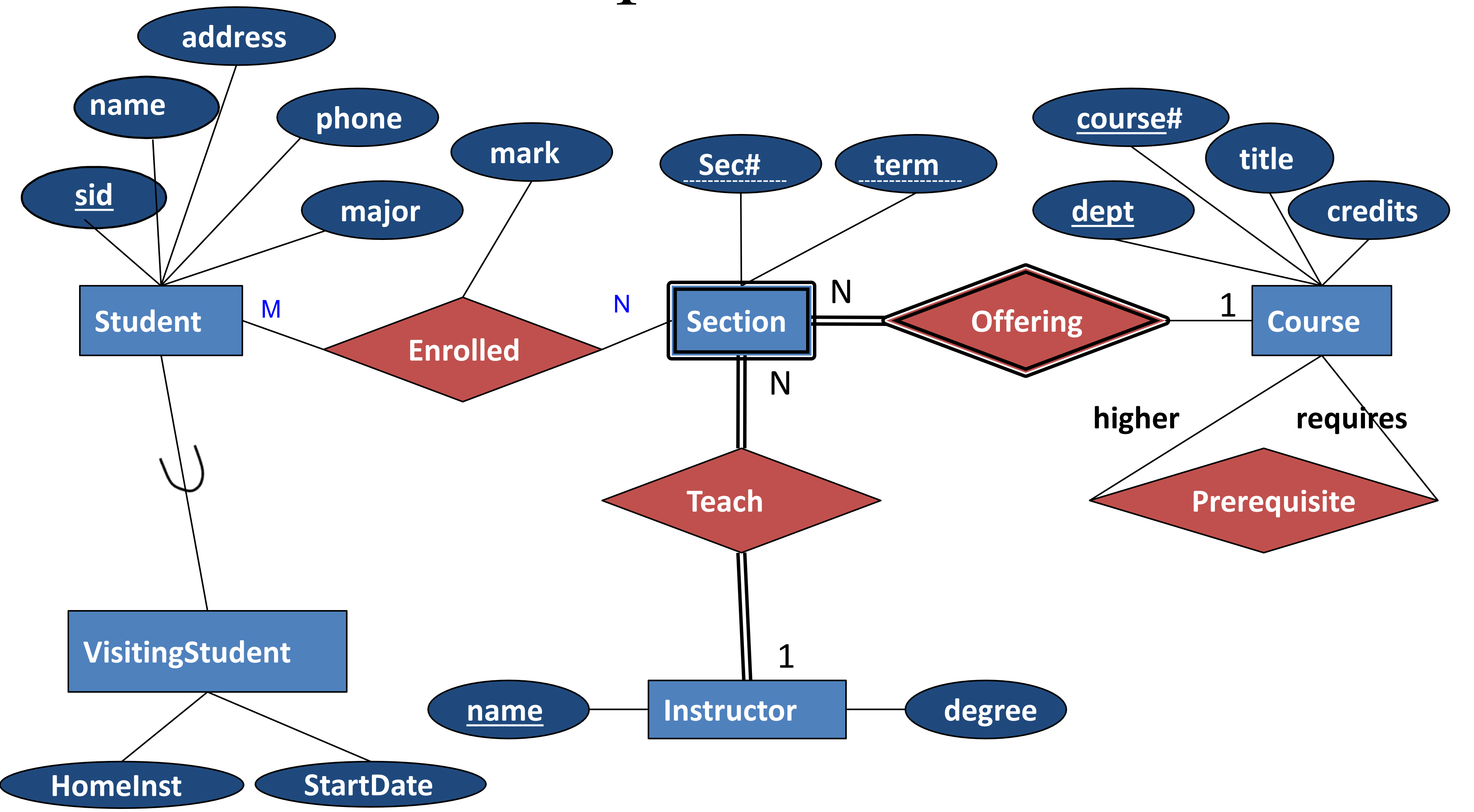
Disjoint specialization

mid term and final similar question

Draw an ER diagram for the following:

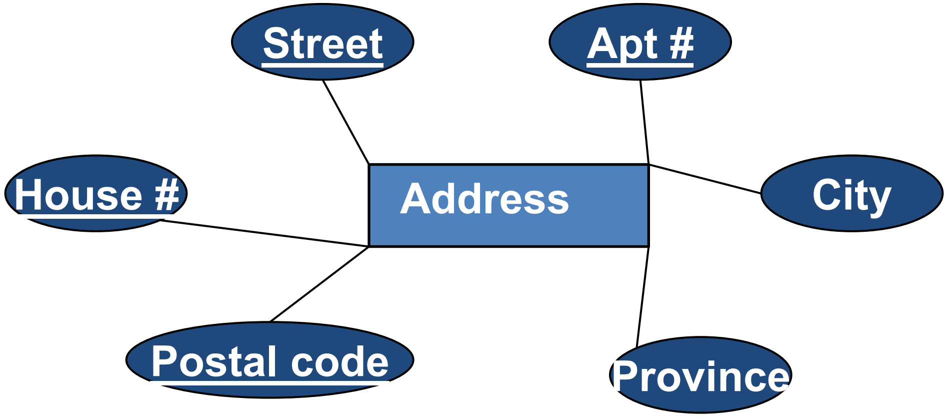
- The primary function of UofU is to offers courses to students.
- A student is identified by a unique student #, and has an address and a phone #. Each student is registered in a program at UofU
- Visiting students stay at UofU for a year.
- A course offered by UofU is identified by the department that 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

Databases

The Entity-Relationship Model

Entity and Relationship

Relationship Constraints

Specialization and Generalization

Design Choices

Conceptual Design Using the ER Model

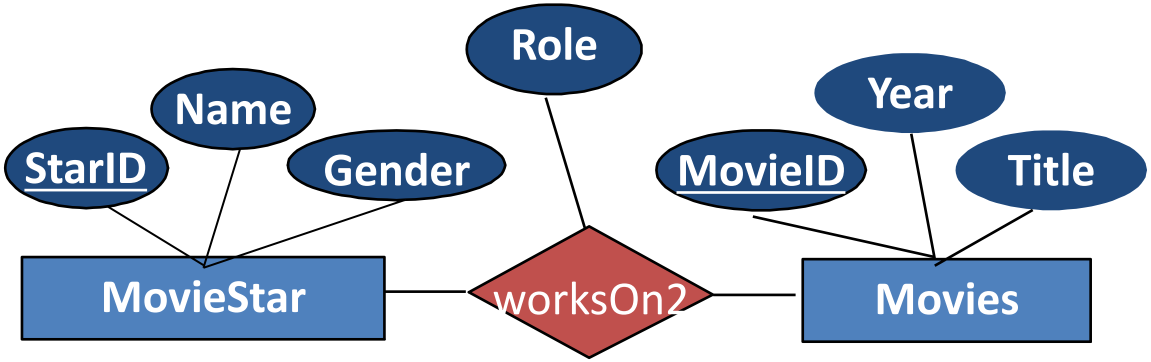
- Design choices:
  - Should a concept be modeled as an entity or an attribute?
  - Should a concept be modeled as an entity or a relationship?
  - Identifying relationships: Binary or ternary? Aggregation?
- Constraints in the ER Model:
  - A lot of data semantics can (and should) be captured.
  - But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

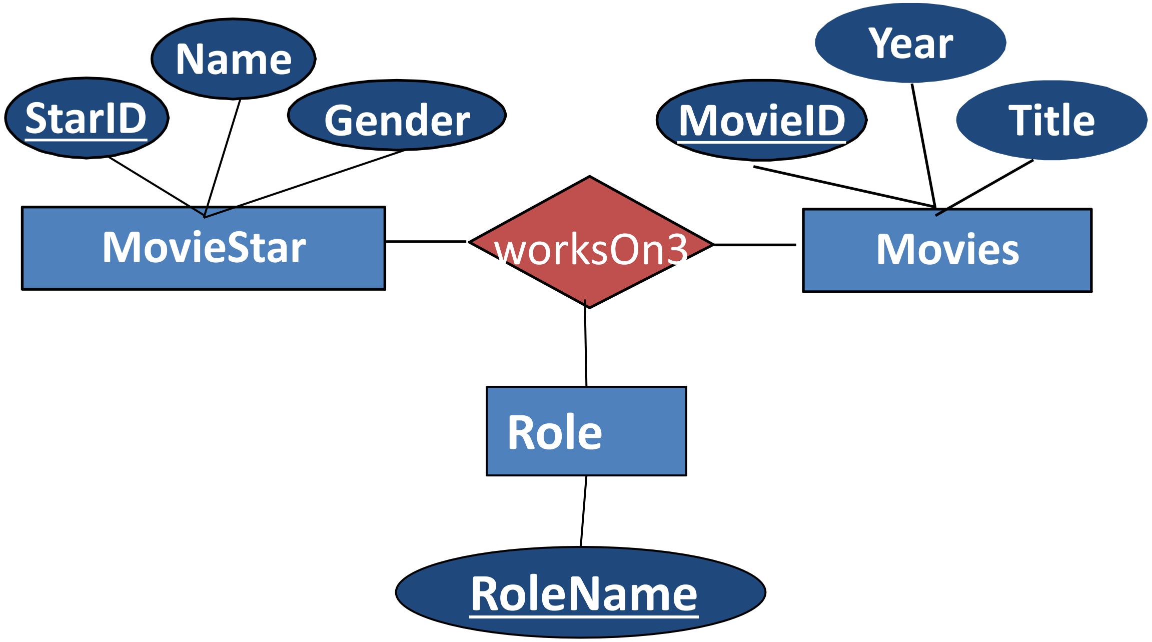
Entity vs. Attribute

- 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?





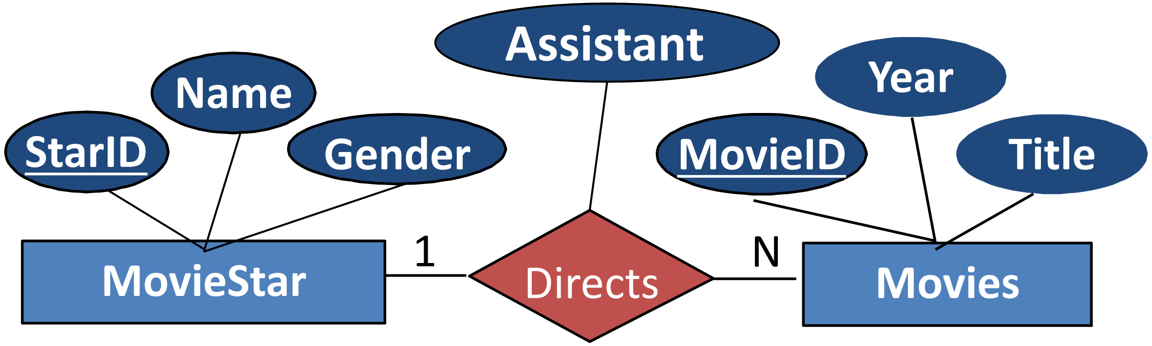
- Solution: change descriptive attributes into entities.

Entity vs. Relationship

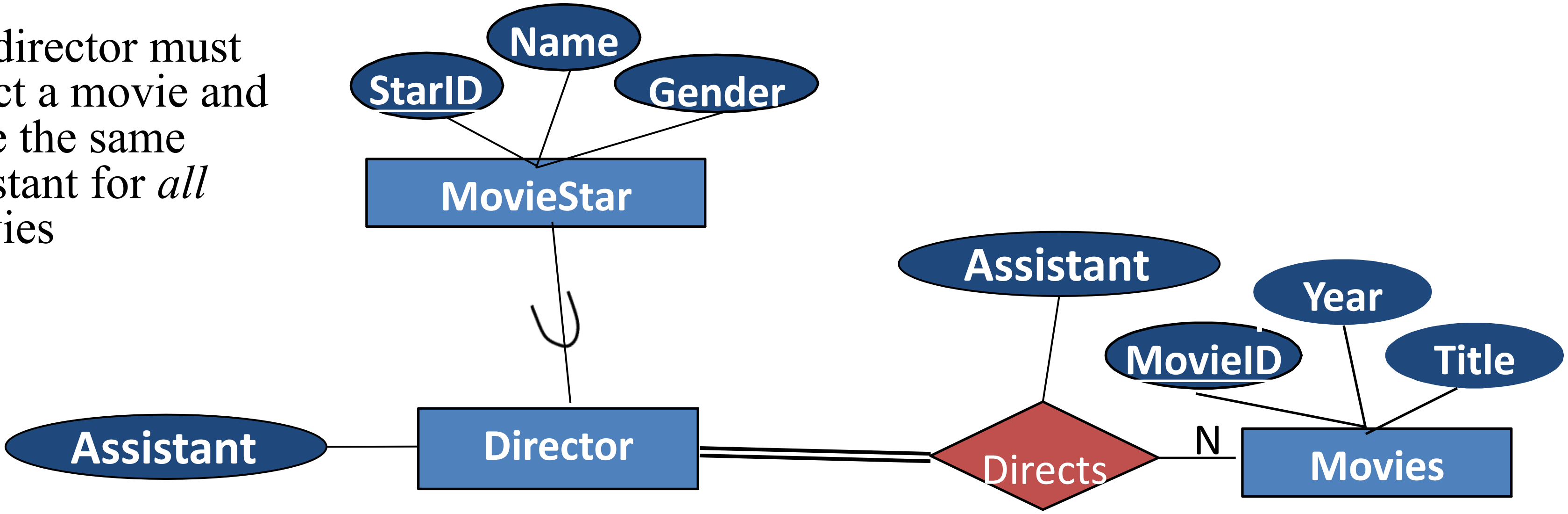
depends on which match the need of applications

- How are the two ER models different?

- Director can get a separate assistant for each movie.

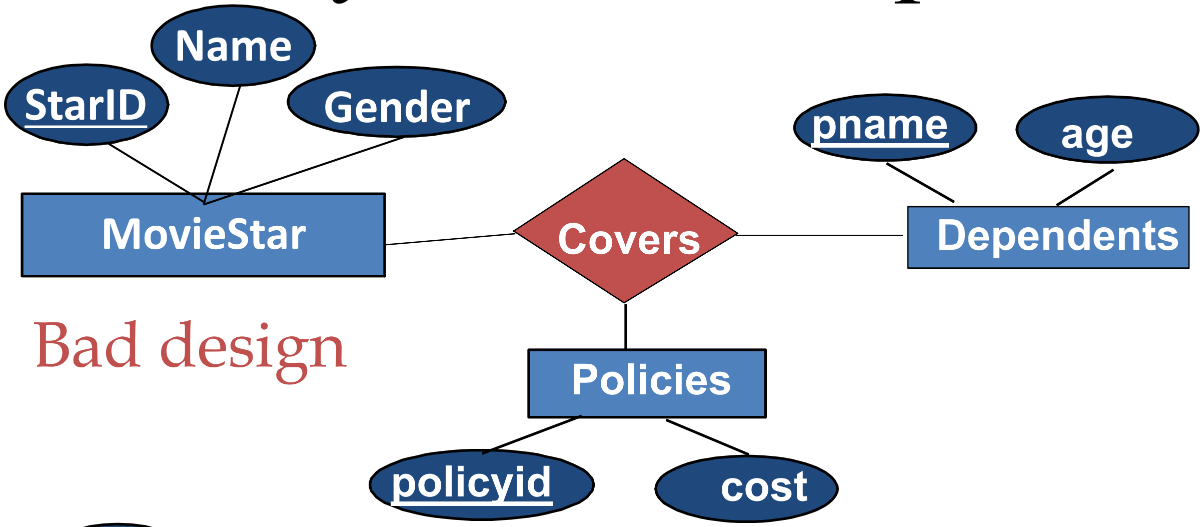


- All director must direct a movie and have the same assistant for *all* movies

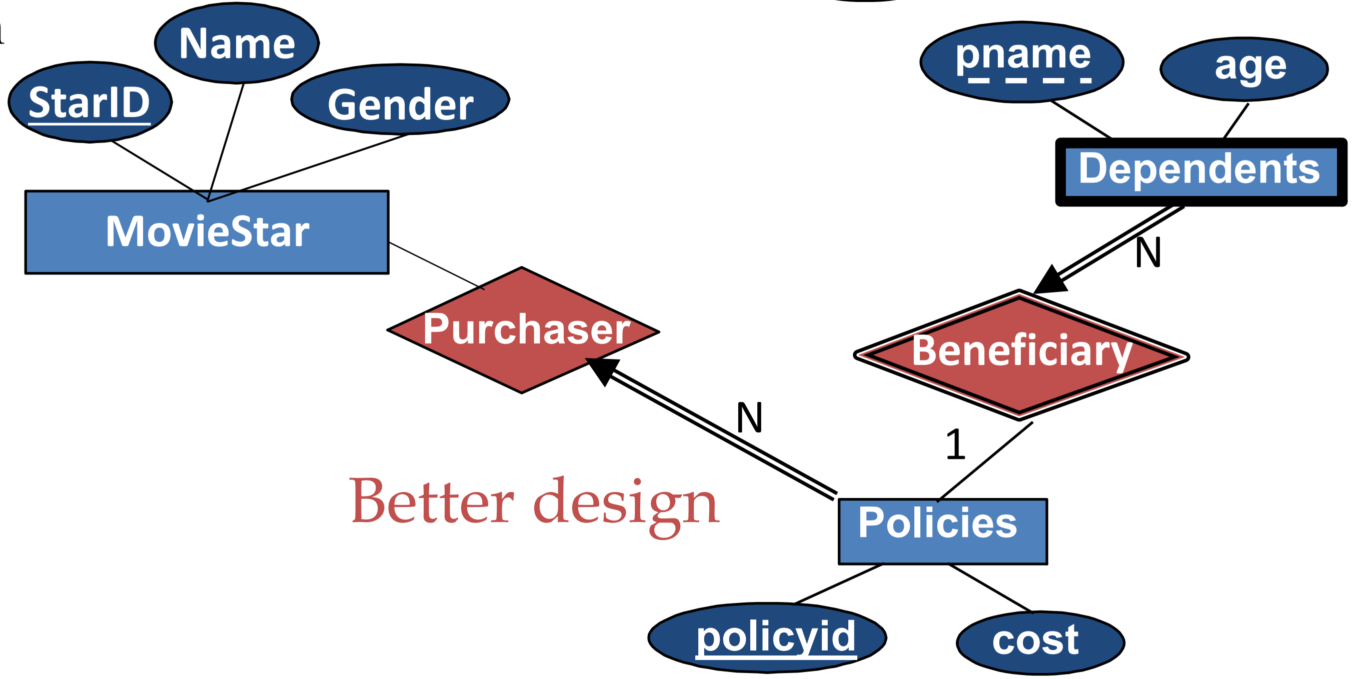


Binary vs. Ternary Relationships

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



Bad design

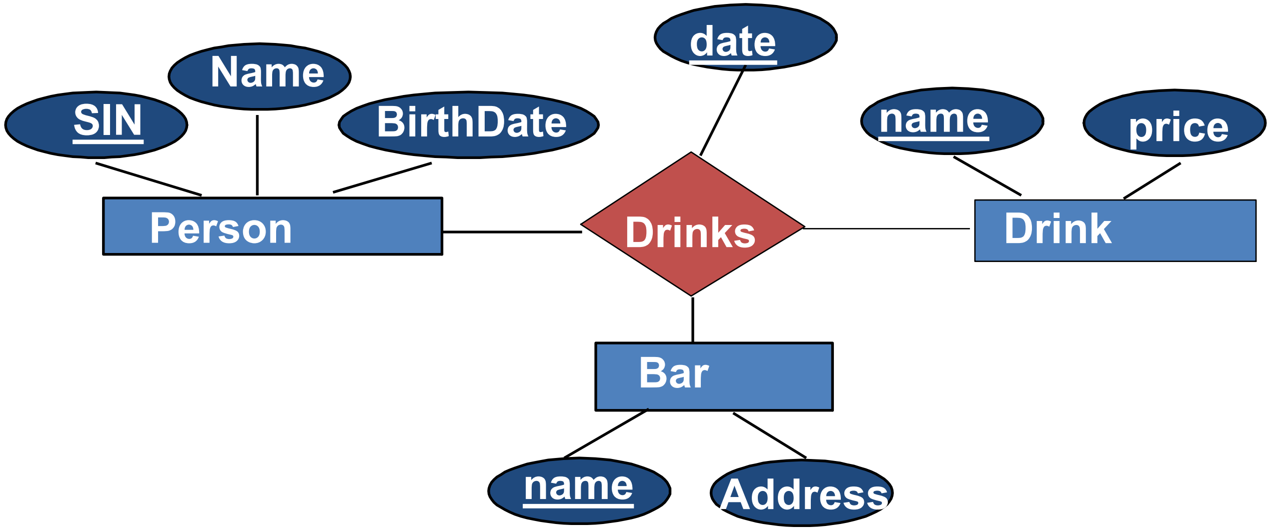


Better design

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

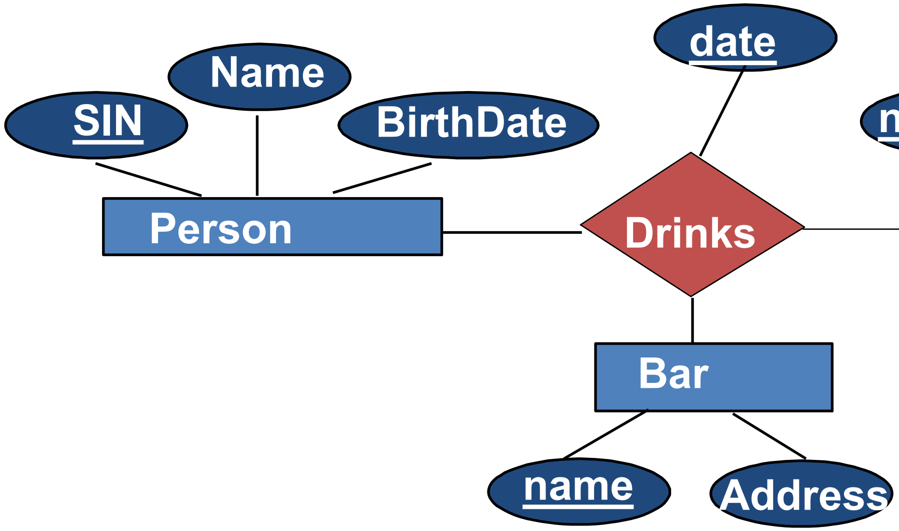
Binary vs. Ternary Relationships

- 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 Outcomes Revisited

Description	Tag
Define the term database.	DBMS
Explain the purpose of having a database.	
Explain the high-level objectives of having a DBMS.	
Explain the three-schema architecture, separating internal level, conceptual level and external level.	
Explain the purpose of an ER diagram.	
Describe the concept and notation of entities in ERs.	ER-model
Describe the concept and notation of relationships in ERs.	
Describe the concept and notation of weak entities in ERs.	
Describe the concept and notation of specialization and generalization In EERs.	
Define Cardinality Constraint and use it in ERs.	
Define Participation Constrain and use it in ERs.	
Given a problem description, create an ER diagram, justifying the decisions you make.	
Given a problem description, identify alternative representations of the problem concepts and evaluate the choices.	ER-compare-models
Compare alternative ER models for the same domain and identify their strengths and weaknesses.	