INFS7901
Database Principles

Introduction

Rocky Chen
tong.chen@uq.edu.au

About Me

Rocky (Tong) Chen

2021 – Now: Lecturer, Data Science Discipline, UQ

Research Interests:

- **Recommender Systems**

  Provide tailored experience for individual e-commerce customers

- **Lightweight Machine Learning**

  Develop data-driven machine learning solutions for low-resource environments

- **Predictive Analytics**

  Informed decision-making and business management with automated predictions

My Google Scholar: https://scholar.google.com/citations?user=07cqSMsAAAAJ&hl=en

Meet Our Tutors

Jason (Xurong) Liang

Yunwen Zhao

Rischan Mafrur

Talia Garrett-Benson

Course Related Information

An Overview of Relational Database Management Systems (DBMSs)

Our Goals

1. Reason with the logical foundation of the relational data model and understand the fundamental principles of correct relational database design.

2. Express natural language queries using relational algebra and SQL.

3. Analyse the fundamental techniques and algorithms applied for sorting, tree manipulation, and hashing on structured data.

4. Reason with the logical foundation on how data is indexed and how a query is executed and optimised.

5. Design relational databases with considerations of data integrity and system performance.

Course Introduction

Relational Databases (Most of INFS1200)	
Duration	6 weeks
Main Topics	Relational Database Design, Relational Model, Functional Dependencies and Normal Forms, SQL

Data Access Methods (An intro to many of the topics covered in COMP3506)	
Duration	4 weeks
Main Topics	Asymptotic Analysis, Searching and Sorting, Abstract Data Types (Dictionary), Binary Search Trees, Hashing

Indexing and Query Processing (Some of the topics covered in INFS2200)	
Duration	2 weeks
Main Topics	Tree- and Hash-Based Indexes, Relational Algebra, Query Optimization,

Pracs/Tutorial

Time slots (from Week 2 to Week 11)

Pracs – please sign-up for one:
- Tuesday 9.00am – 10.00am, in-person
- Tuesday 10.00am – 11.00am, in-person
- Tuesday 11.00am – 12.00pm, in-person
- Tuesday 12.00pm – 1.00pm, online

Course Assessments

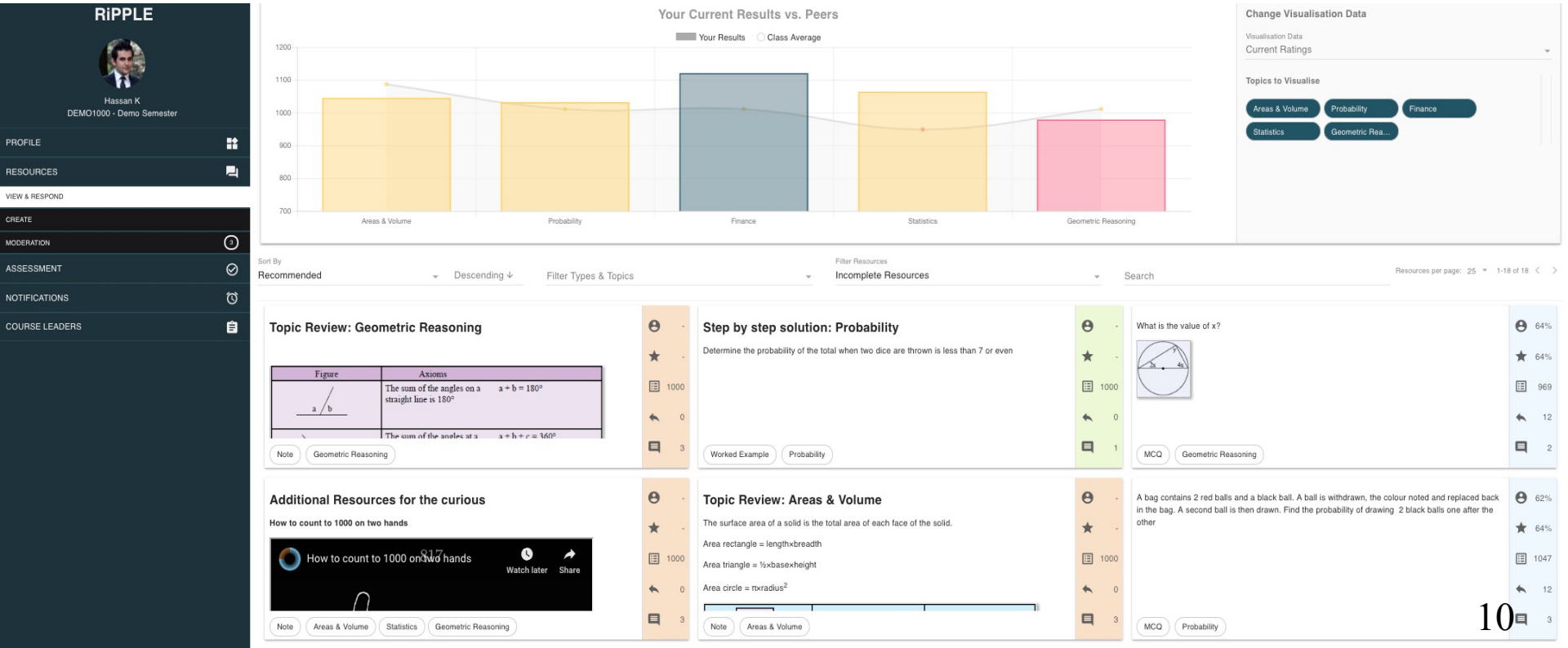
Item	Weighting	Notes
Project	40%	Two milestones
Final Exam	50%	Open-book
RiPPLE	10%	An alternative to quizzes; four milestones

Project

- You are required to select an application that would benefit from a database and build a database application from start to finish.
- Two deliverables/milestones: proposal + final report (with code).
- Project documents will be made available this Friday. Please make sure you are aware of the key dates and deliverables – plan ahead!

The RiPPLE Platform

- RiPPLE recommends personalised learning resources to you based on your knowledge from a repository of learning resources that are generated by instructors and the students themselves
- More info will become available this week



Ed Discussion Board

- Ed Discussion is a Q&A web service.
- It will be used by the teaching team to communicate with you ASAP
- You can use it to ask questions, and respond to others' questions.
  - Please actively participate and answer questions yourselves.
  - Please only use email for personal matters.

Additional Learning Resources

Fundamentals of Database Systems

Available in digital at UQ Library

Online Resource: Problem Solving with Algorithms and Data Structures using Python
http://interactivepython.org/runestone/static/pythonds/index.html

Course Related Information

An Overview of Relational Database Management Systems (DBMSs)

Data is the
new fuel for the
global economy

WHERE is this data

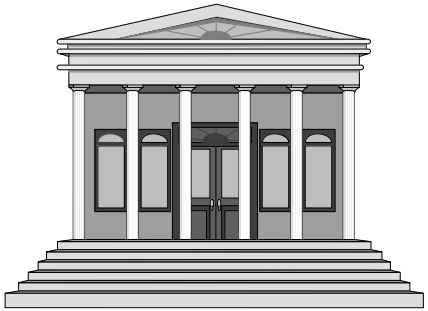
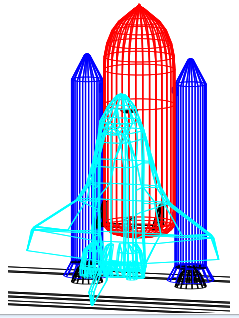
WHAT can you do with it

WHY is data so important

WHO owns this data

HOW can you use it

Impact

Commerce	Government	Science and Engineering
		
Cash management	Law Enforcement	Health and Medicine
Pricing	Election Commission	Space Exploration
Employee records	Taxation	Geography (GIS)
Purchasing	Legal Systems	Architecture
Reordering	Transport & Utilities	Military and Defense
Trend analysis	City Councils	Telecommunications

Impact

Or... Managing Ed Sheeran’s concert tickets?

A Motivating Example

Suppose you are building a system to store the information pertaining to a university from scratch.

You have access to an operating system of your choice, but that's it.

What are some the things you need to consider for building this system?

Potential Responses

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)

What is a DBMS?

- A Database Management System (DBMS) is a software system designed to store and manage databases. It is used to:
  - Maintain integrity (Week 3)
  - Control redundancy (Week 4)
  - Create, modify, and query a database (Week 5)
  - Provide support for decision making
  - Control access
  - Permit concurrent access
  - Provide loading, backup, and recovery

Covered in future INFS courses

Questions

Which of the following is not a function of the DBMS

A. Enforce integrity constraints
B. Design the database to be used
C. Backup and recovery of database
D. Provide secure access to the database

Questions

Which of the following is not a function of the DBMS

A. Enforce integrity constraints

B. Design the database to be used

C. Backup and recovery of database

D. Provide secure access to the database

INFS7901
Database Principles

The Entity-Relationship Model

Rocky Chen

Databases

The Entity-Relationship Model

Entity and Relationship

Relationship Constraints

Specialization and Generalization

Design Choices

- Who is the watcher?

  A **database** is an organized collection of related data, usually stored on disk

- Who watches the watcher?

  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

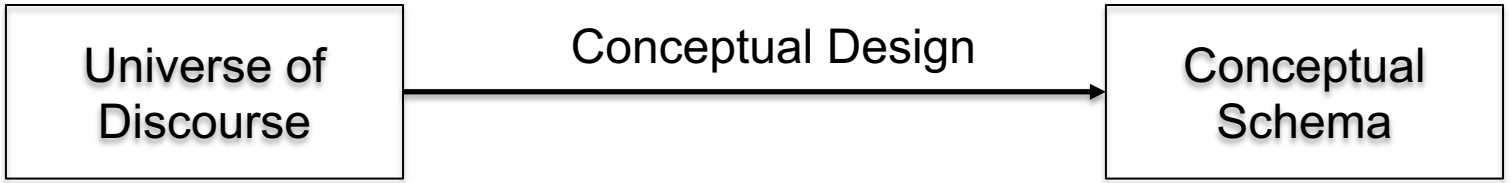
Specialization and Generalization

Design Choices

Conceptual Database Design

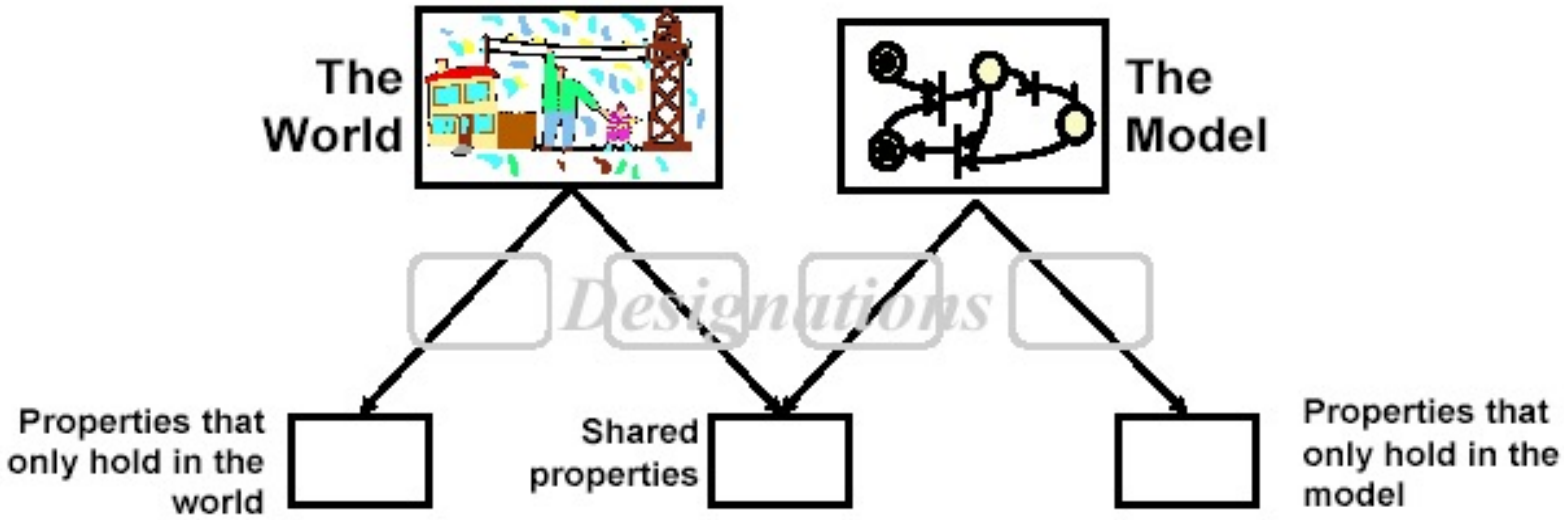
- Conceptual database design is a very important phase in designing a successful database application.
- Step 1: Identify the "Universe of Discourse" (UoD)
  - The database to be built will not model everything in the world, but rather some "mini-world" or "Universe of Discourse".

- Step 2: Convert the UoD to a conceptual schema, which can be captured by a database.



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



How the customer explained it

How the project leader understood it

How the analyst designed it

How the customer was billed

What the customer really needed

What are the implications for database designs?

Entity-Relationship (ER) Diagram

- In relational databases, the conceptual schema is generally encoded in an **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 developing conceptual database designs.

Databases

The Entity-Relationship Model

Entity and Relationship

Relationship Constraints

Specialization and Generalization

Design Choices

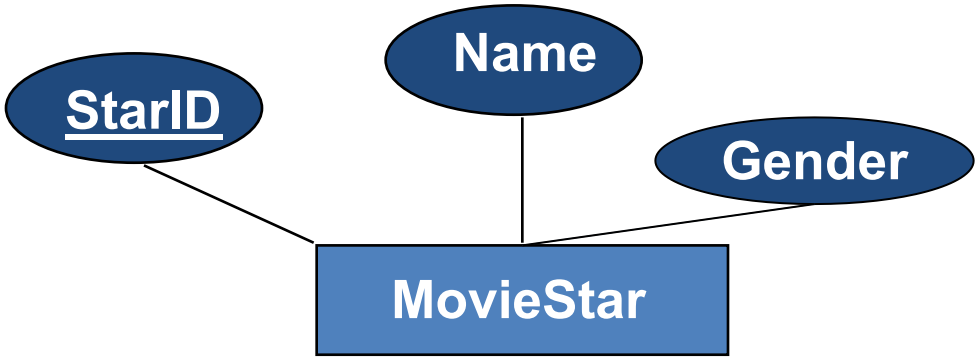
Entity

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



- Each attribute has a ***domain***. *(e.g., float, date, int)*
- Each entity set has a ***key.***

Entity Set

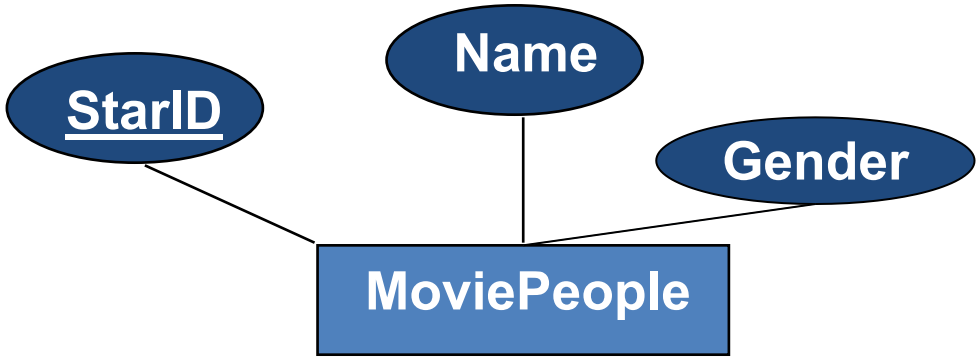
- ***Entity Set:*** The collection of all entities of a particular entity type in the database at any point in time
  - E.g., a set of movie stars
  - An entity set can easily be mapped to a table.

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

Key Attributes

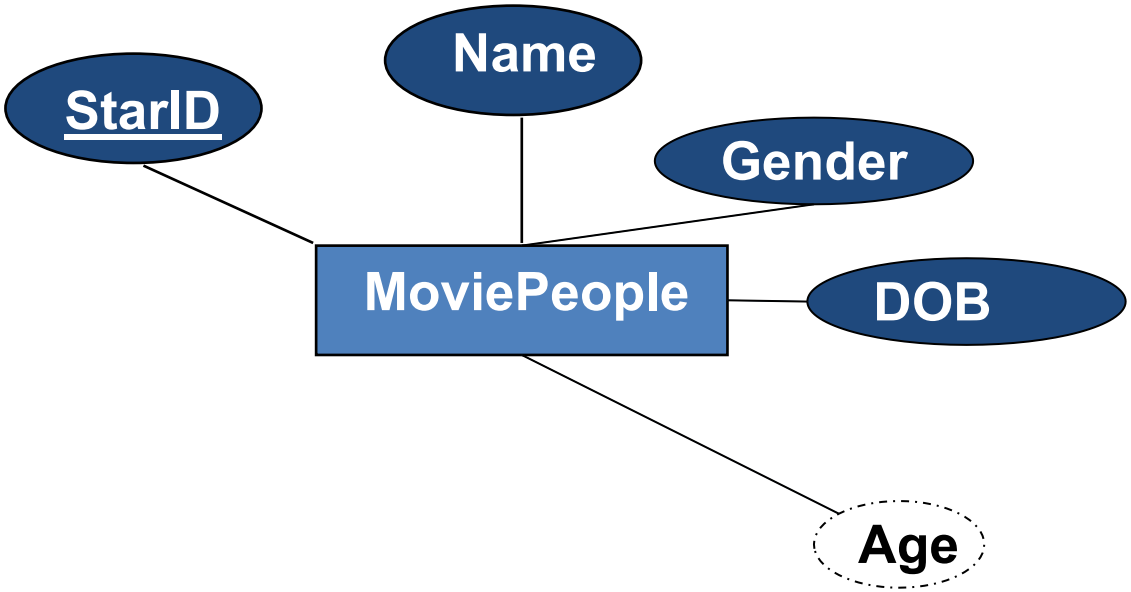
- A *key* is the minimal set of one or more attributes which, taken collectively, uniquely identify an entity in an entity set. (Underlined in ER diagram)

- A *primary key* is the key chosen as the principal means to identify entities in an entity set



Derived vs. Stored Attributes

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



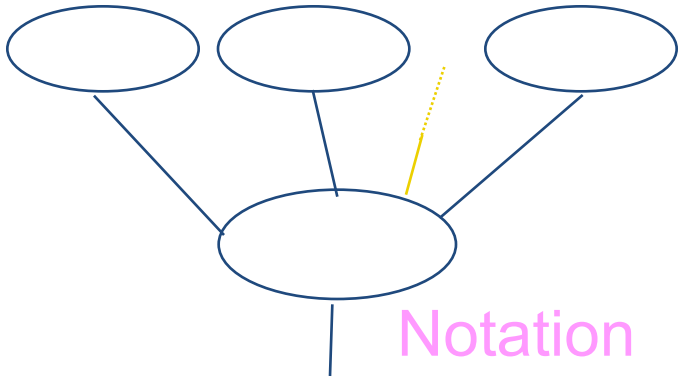
Null Valued Attributes

- 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

- Simple Attribute: Customer_Name
- Complex Attribute: Shipping_Address which is comprised of
  - Street Number
  - Street Name
  - Suburb Name
  - City
  - Postcode
  - …

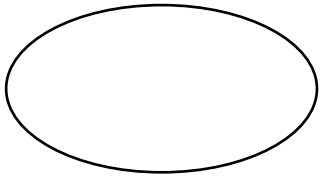


Single vs. Multivalued Attributes

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

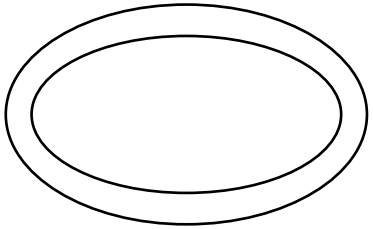
- Single-valued: Gender = F

Notation



- Multivalued: Degree = {BSc, MInfTech}

Notation



In-class Exercise

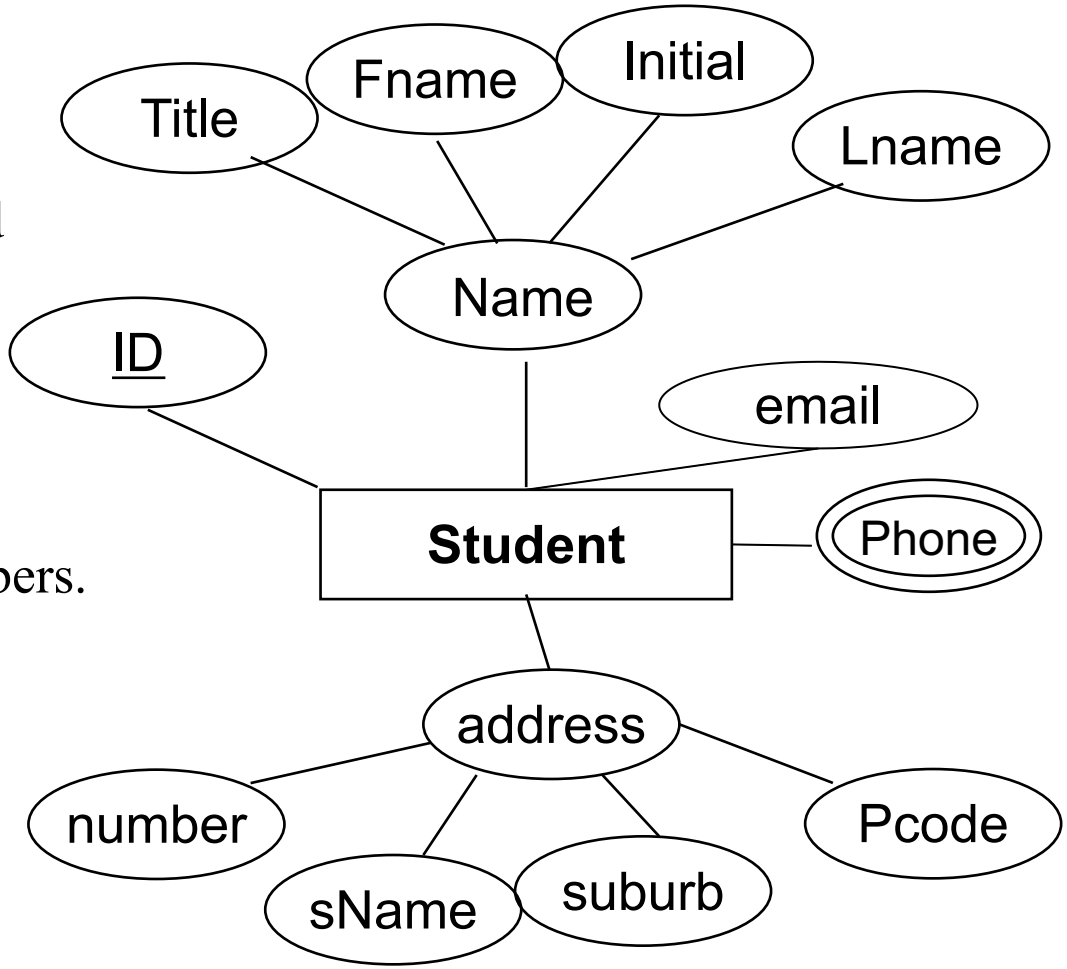
Every student has a unique id, name (composed of title, first name, middle initial and last name), email, address (composed of number, streetname, suburb and postcode), and one or more phone numbers. Draw an ER diagram for the above description.

In-class Exercise - Solution

Every student has a unique id,

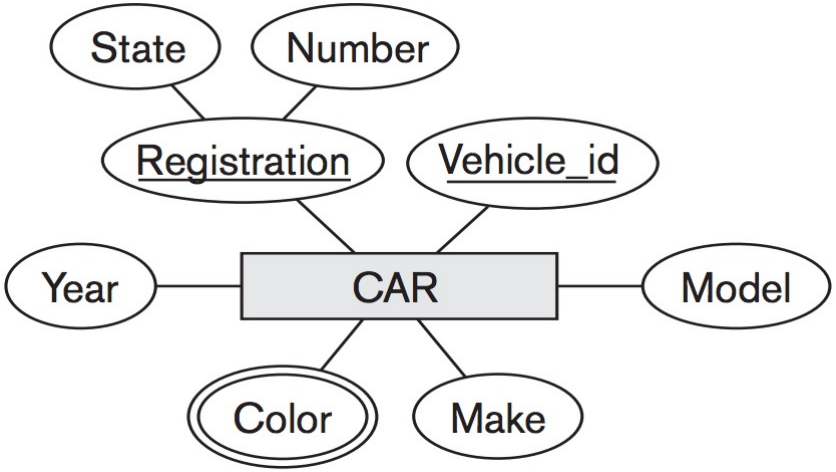
name (composed of title, first name, middle initial and last name), email, address (composed of number, streetname, suburb and postcode),

and one or more phone numbers.



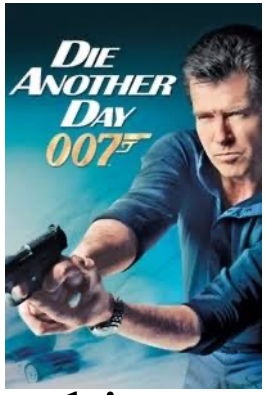
Several Attribute Keys

- Sometimes several attributes together form a key, meaning that the combination of the attribute values must be distinct for each entity.
- If a set of attributes possess this property, the proper way to represent this in the ER model that we describe here is to define a composite attribute.

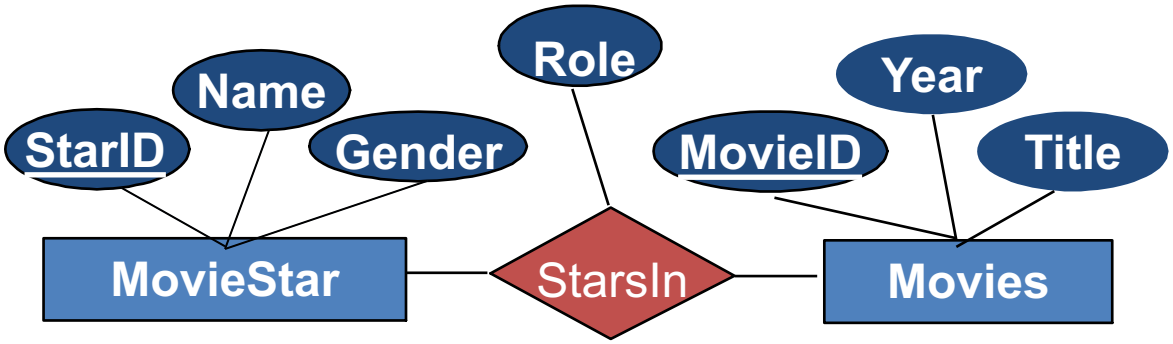


Relationship Types and Sets

- ***Relationship:*** Association among two or more entities.
  - E.g., Pierce Brosnan starred in 007-Die Another Day.



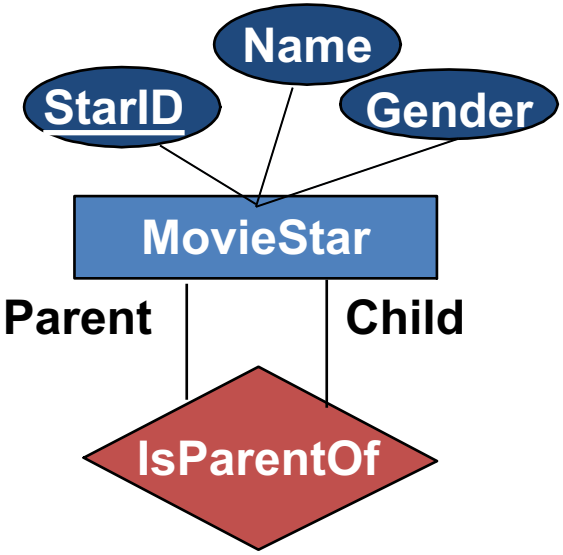
- ***Relationship Type*** defines the relationship



- ***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.

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 a composite attribute representing its key.
D. Entity set Courses with attribute department underlined and attributes number and title not underlined.

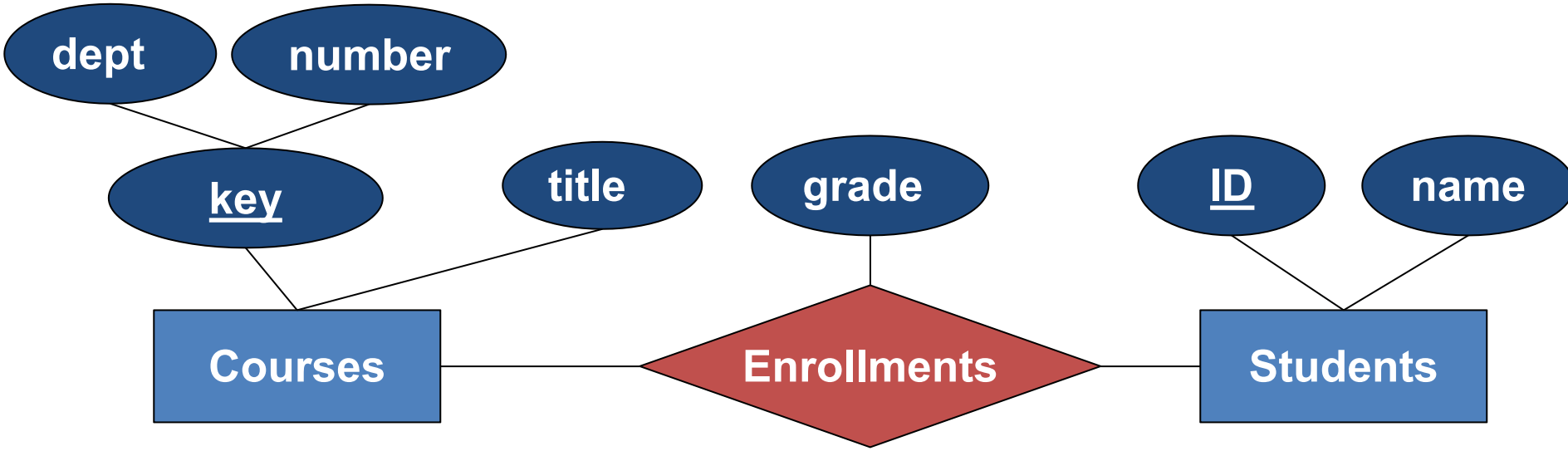
Clicker Question

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 a composite attribute for its key. Correct Answer

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



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 coordinators to courses and restrict that each could only have one coordinator?

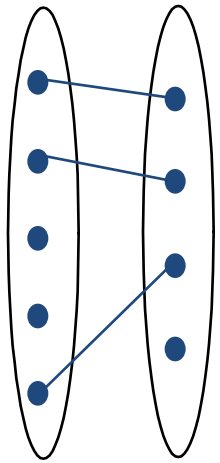
- Two kinds of constraints can be defined in the ER Model
  - Cardinality Constraint
  - Participation Constraints

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:

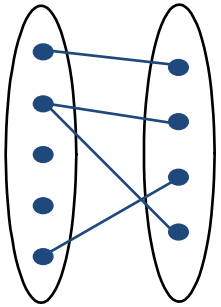
1. ***one-to-one*** from A to B:
   - an entity in A is associated with at most one entity in B and vice versa
   - e.g. A: driver, B: driver's license



1-to-1

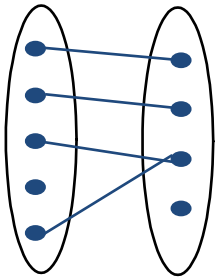
Cardinalities

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

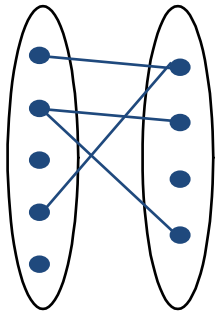
3. ***many-to-one*** *from A to B:*
   switch A and B above



Many-to-1

Cardinalities

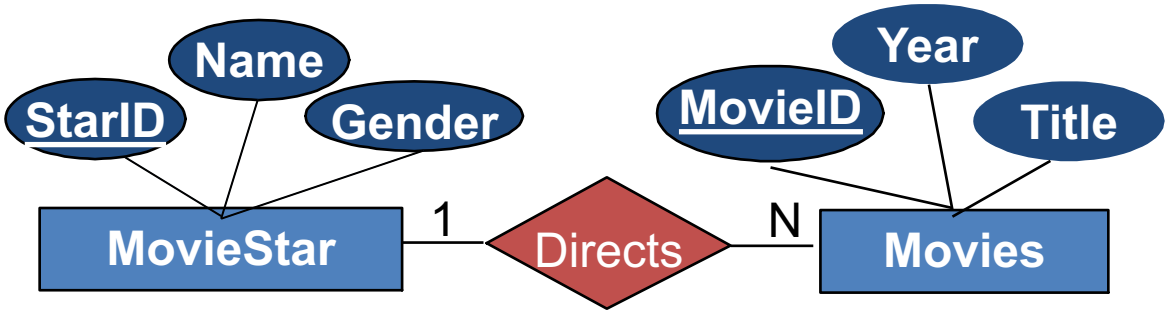
- ***many-to-many*** *from A to B:*
  - *an entity in A is associated with any number of entities in B and vice versa*
  - *e.g. A: students, B: courses*



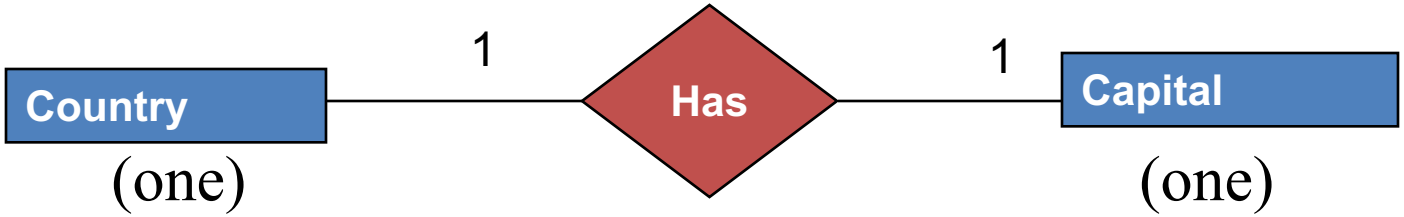
Many-to-Many

Key Constraints

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

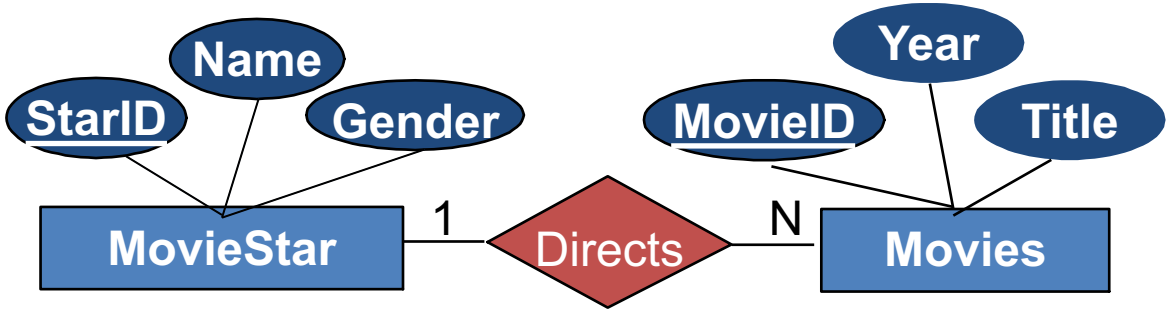


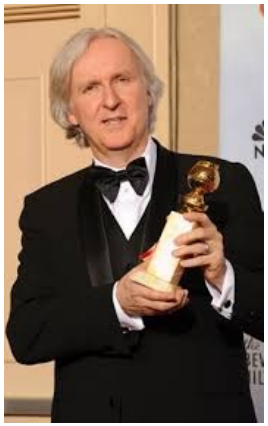
or

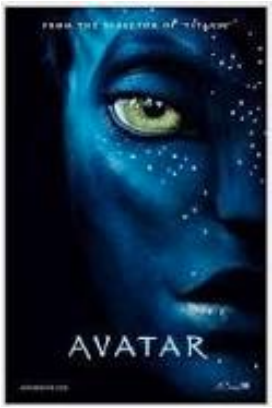


Example

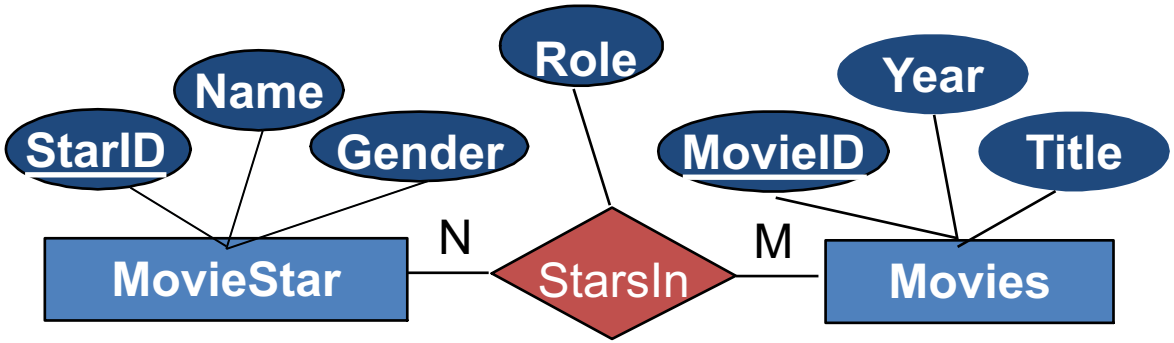
- In this example, if you know the movie, you would be able to determine the director



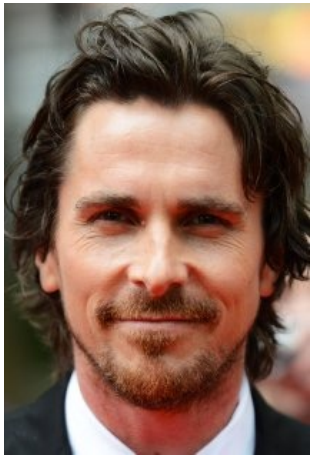




Key in Many-to-Many Relationships



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

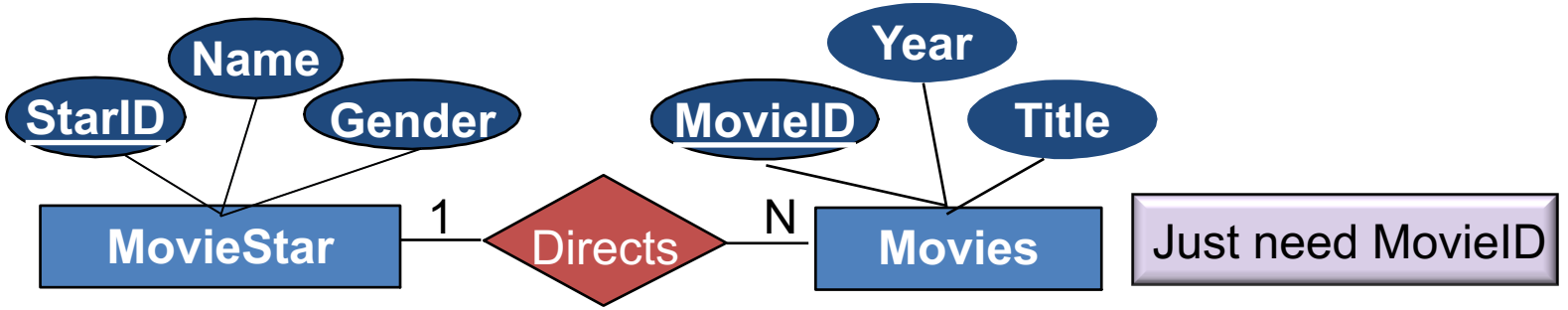


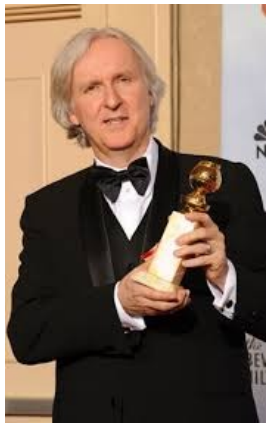
Christian Bale
as
Bruce Wayne

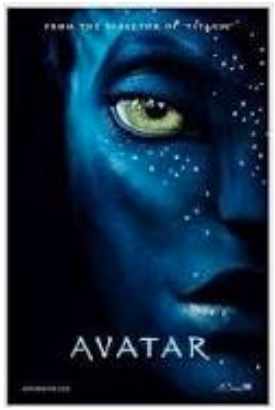
Need StarID and MovieID

Can the same person have multiple roles in one movie?

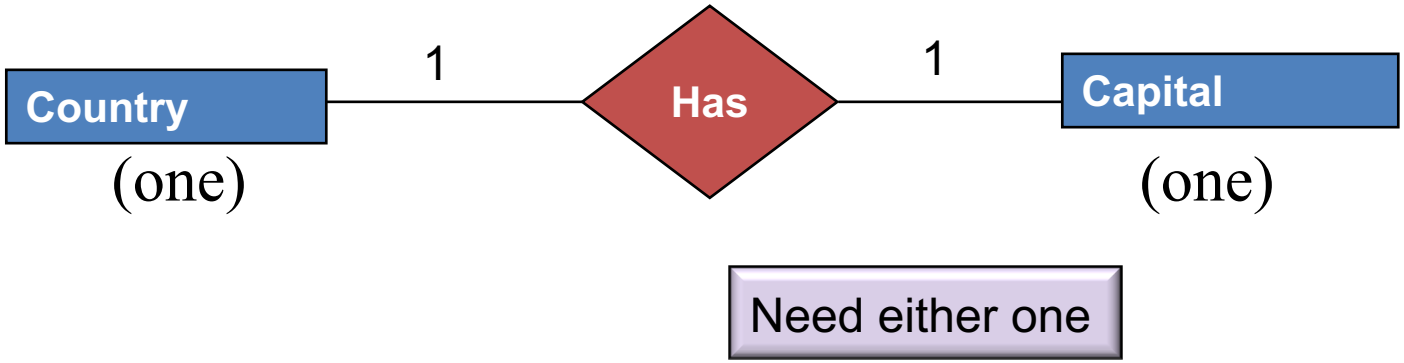
Key in One-to-Many Relationships







Key in One-to-One Relationships



Ottawa

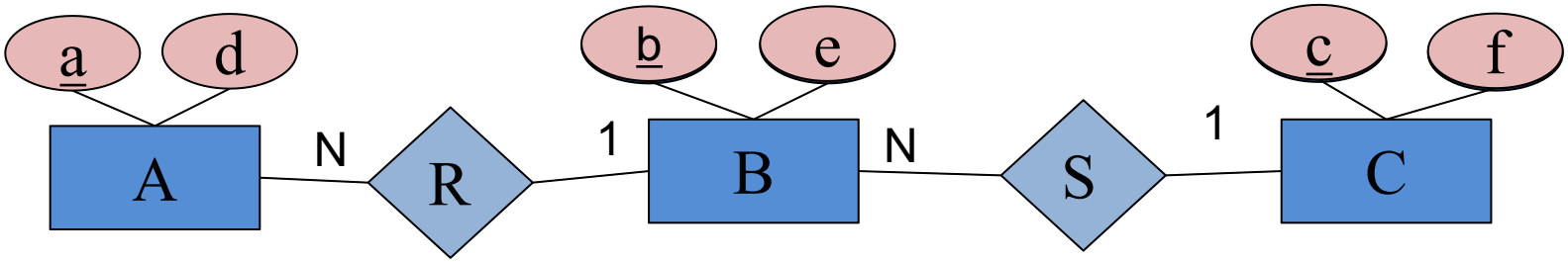
Primary Keys of Relationship Sets

- 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 A contains a1 and a2, B contains b1 and b2, and C contains c1 and c2.

Which of the following relationships for R and S are possible according to the diagram?

A.

R		S	
		b2	c1
		b2	c2

B.

R		S	
		b1	c2
		b2	c2

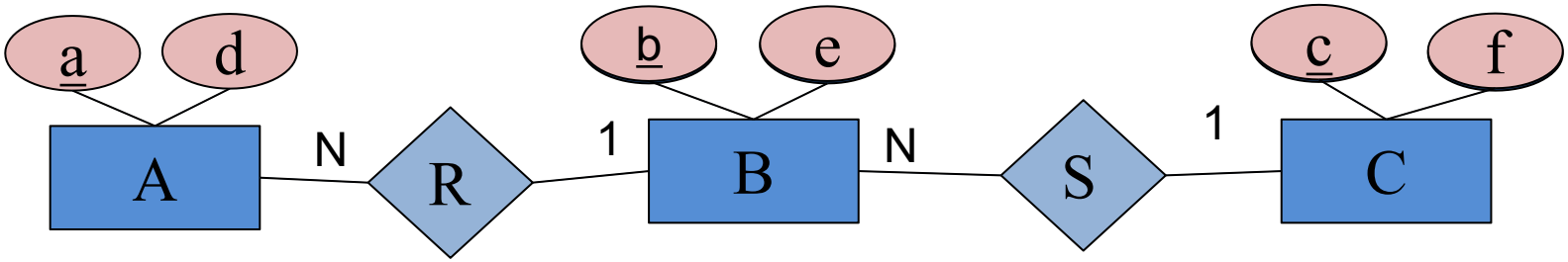
C.

R		S	
a2	b2	b2	c1
		b2	c2

D.

R		S	
a1	b2		
a2	b1		
a2	b2		

Clicker Exercise



Suppose that A contains a1 and a2, B contains b1 and b2, and C contains c1 and c2.

Which of the following relationships for R and S are possible according to the diagram?

A.

R		S	
		b2	c1
		b2	c2

B determines C

B.

R		S	
		b1	c2
		b2	c2

Correct

C.

R		S	
a2	b2	b2	c1
		b2	c2

B determines C

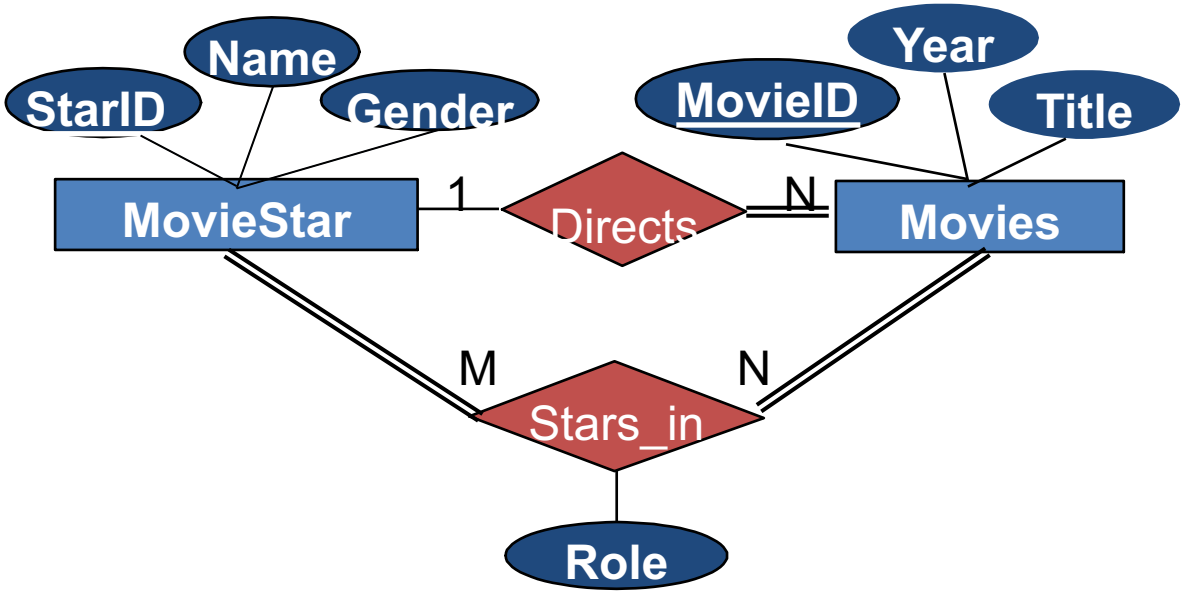
D.

R		S	
a1	b2		
a2	b1		
a2	b2		

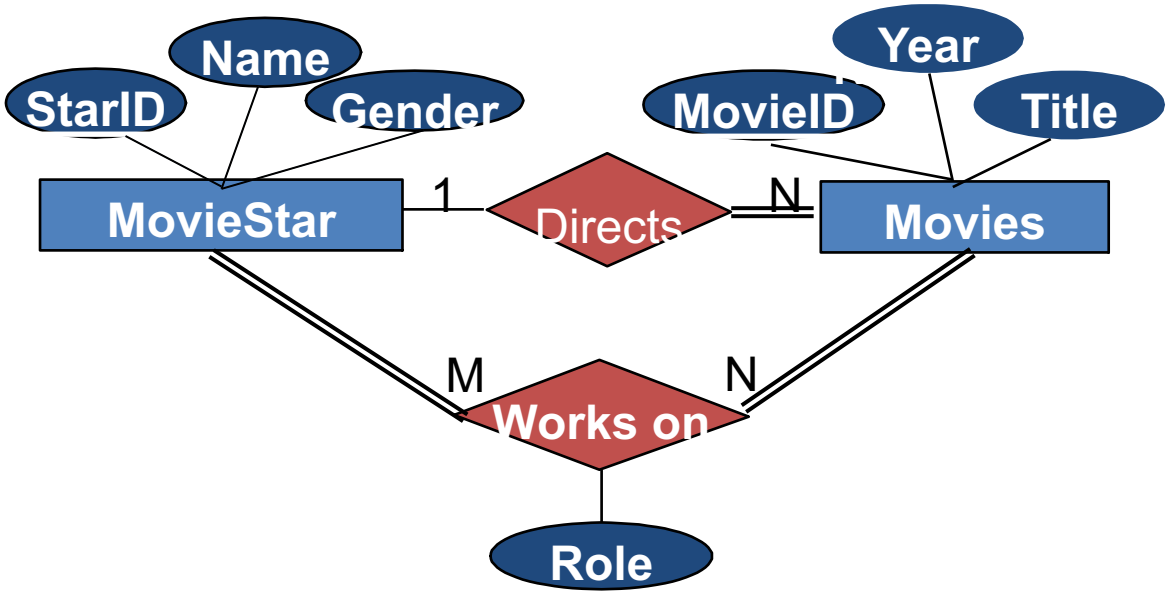
A determines B

Participation Constraints

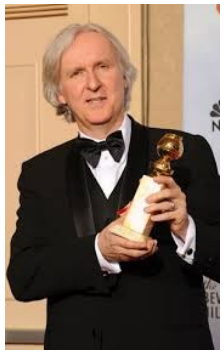
- Participation: Indicates if all entities participate in the relationship.
- An entity's participation can be ***total*** or ***partial*** .
- Requiring total participation is a ***participation constraint*** and it is shown with a ***double line***
  - Important on deletions
  - i.e., participation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

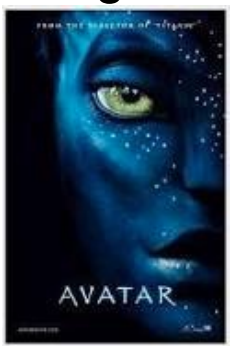


Why is Participation Constraint Important

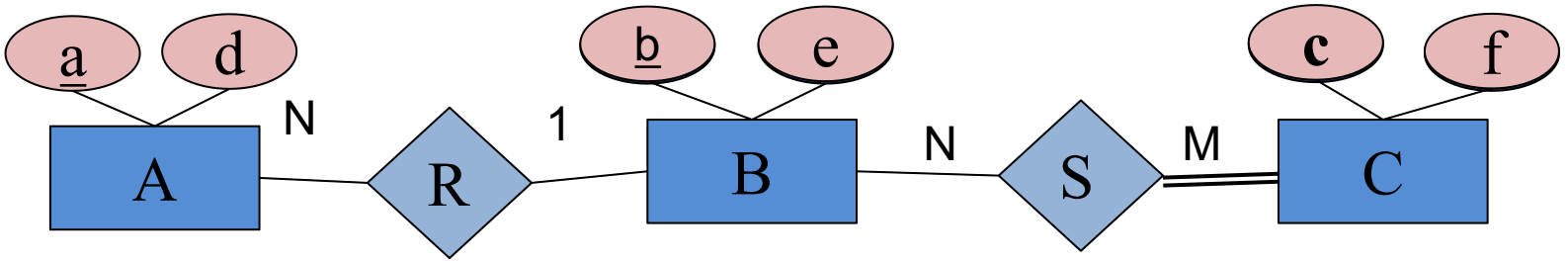


Would we be able to delete James Cameron without deleting Avatar?
Would we be able to delete Avatar without deleting James Cameron?





Clicker Exercise



Suppose that A contains a1 and a2, B contains b1 and b2, and C contains c1 and c2.

Which of the following relationships for R and S are possible according to the diagram?

A.

R		S	

B.

R		S	
a1	b1	b2	c2

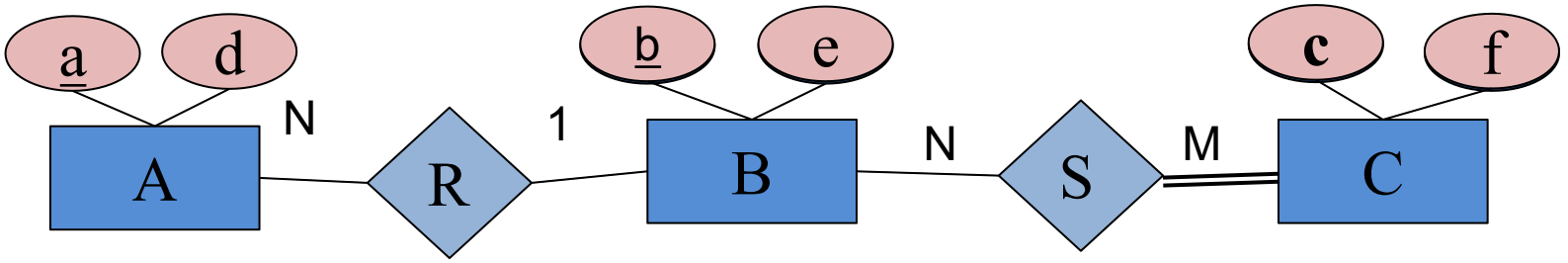
C.

R		S	
a1	b1	b1	c1
a1	b2	b2	c2

D.

R		S	
a1	b2	b1	c2
		b2	c1
		b1	c1

Clicker Exercise



Suppose that A contains a1 and a2, B contains b1 and b2, and C contains c1 and c2.

Which of the following relationships for R and S are possible according to the diagram?

A.

R		S	

C must participate

B.

R		S	
a1	b1	b2	c2

C must participate

C.

R		S	
a1	b1	b1	c1
a1	b2	b2	c2

A determines B

D.

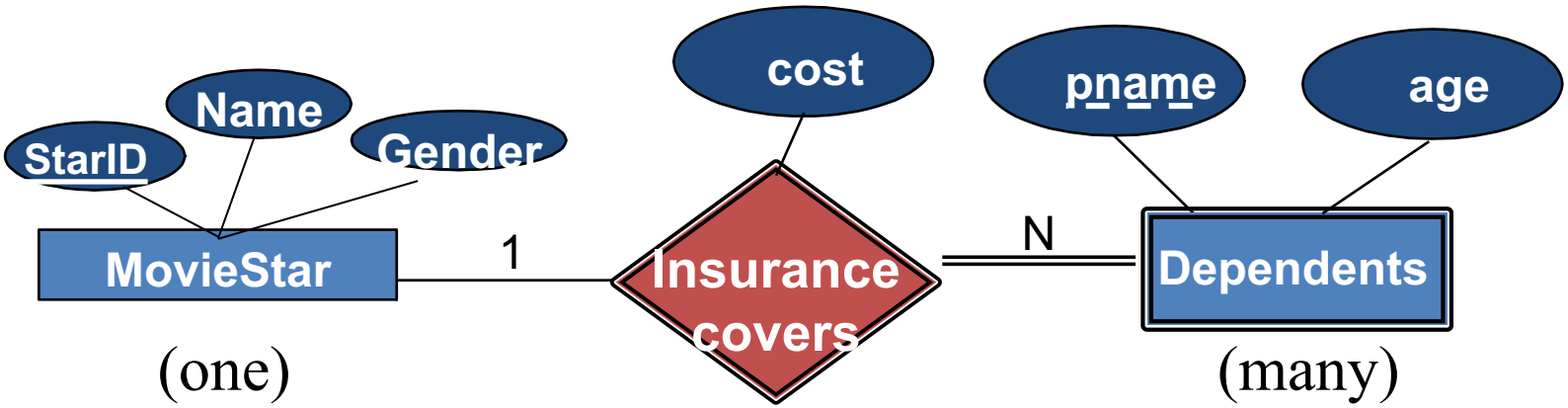
R	
a1	b2

S	
b1	c2
b2	c1
b1	c1

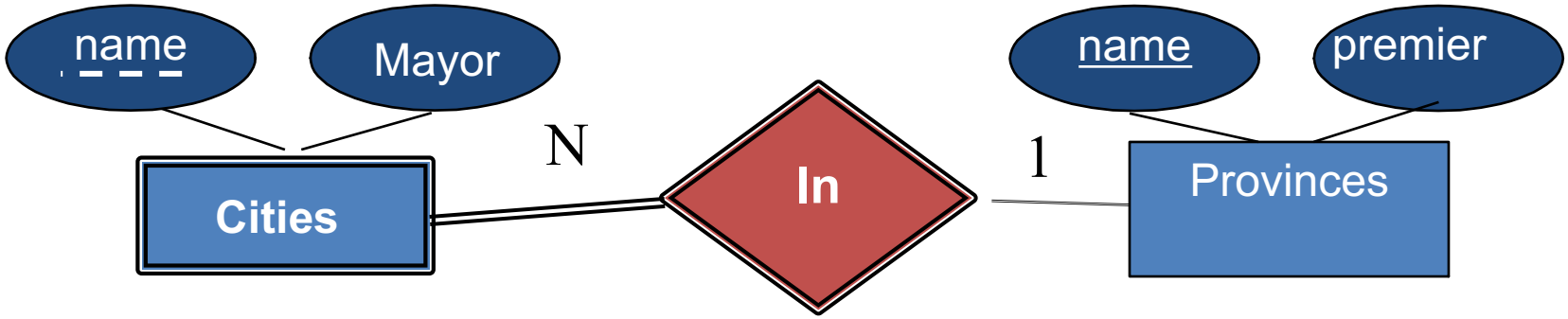
Correct

Weak Entities

- A ***weak entity*** can be identified uniquely only by considering the primary key of another (*owner*) entity.
  - Weak entities have a ***partial key*** that is unique when combined with the owner's key.
  - Owner entities and weak entities must participate in a one-to-many relationship (one owner, many weak entities).
  - Weak entities must have total participation in this ***identifying*** relationship.
  - Think of this as a "belongs to" relationship.
- Weak entity and their identifying relationship sets are shown with ***double lines***.



Clicker Exercise



Which of the following is necessarily true:

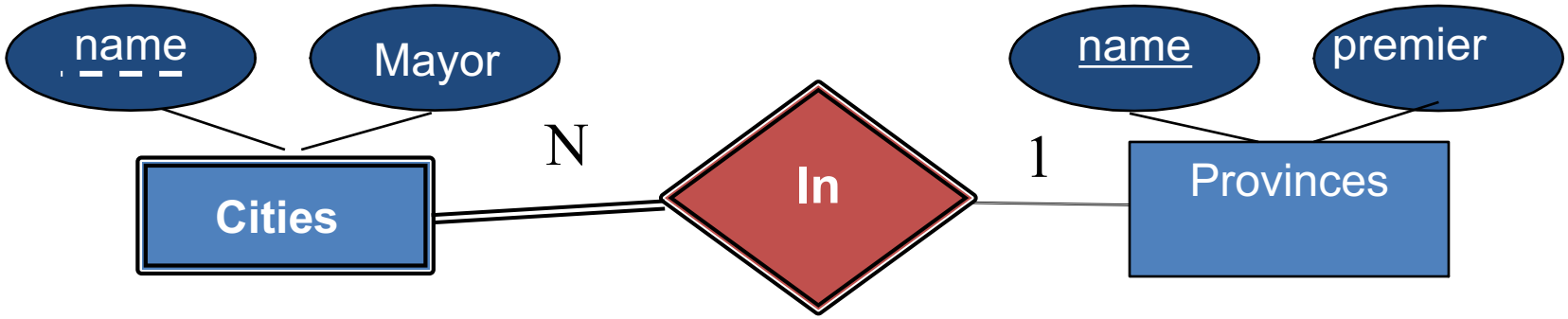
A. No two provinces can have premiers with the same name.

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

C. No two cities can have the same name.

D. 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. None of the above D is correct

Databases

The Entity-Relationship Model

Entity and Relationship

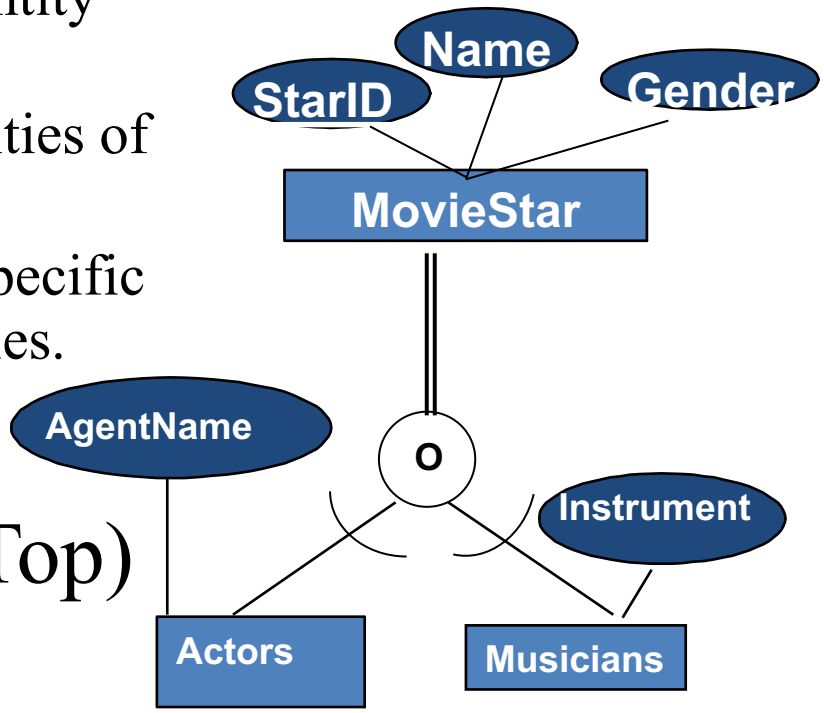
Relationship Constraints

Specialization and Generalization

Design Choices

Specialisation and Generalisation

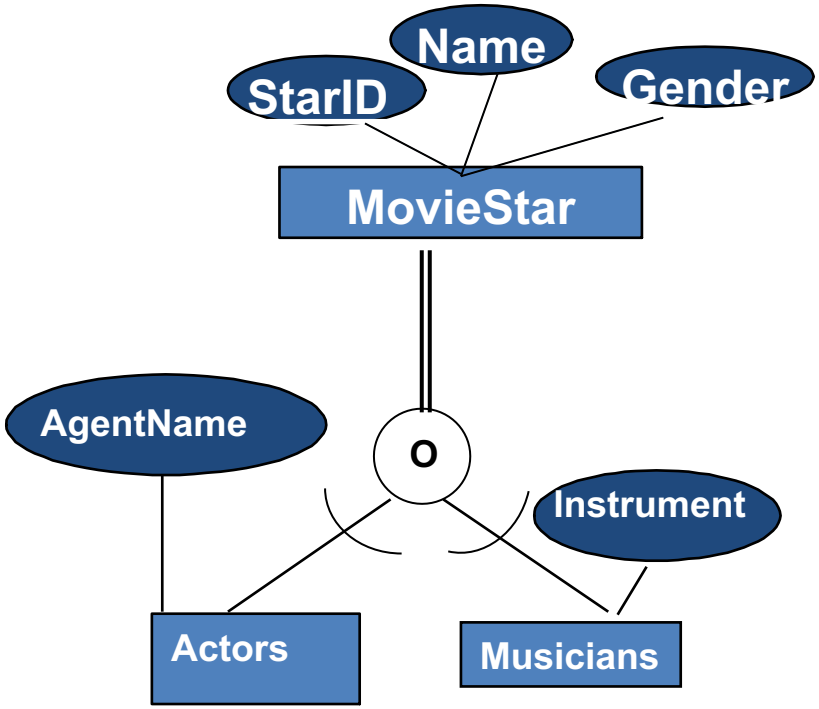
- Specialisation (Top-to-bottom)
  - Define a number of subclasses of an entity type.
  - Each subclass contains a subset of entities of the superclass.
  - A subclass is defined based on more specific distinguishing characteristics the entities.



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

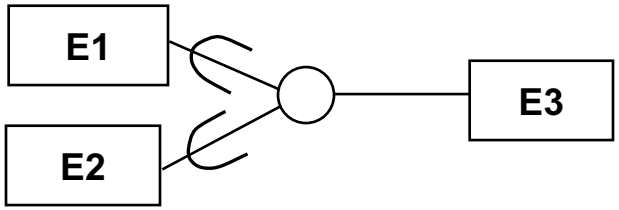
Extended ER (EER)

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



Specialization/Generalization

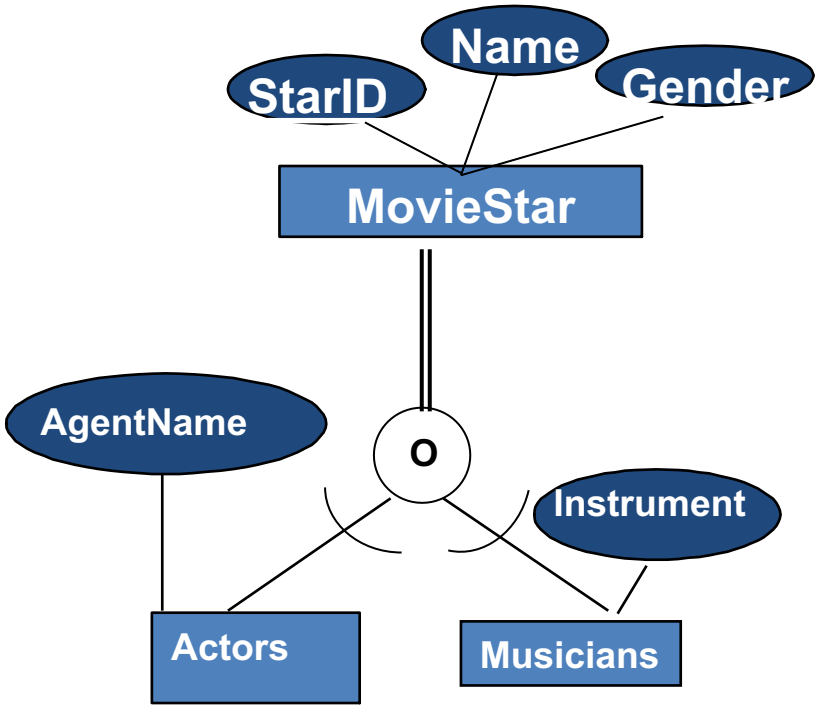
- Entity types refined into sub-classes and super-classes:



- Allows us to model:
  - attributes only applicable to entity subclasses
  - relationships only played by entity subclasses

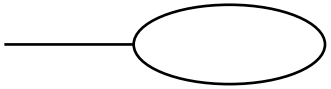
Constraints

- Specialization may be
  - total ═══
  - partial ———
- Subclass sets may be
  - overlapping (o)
  - disjoint (d)

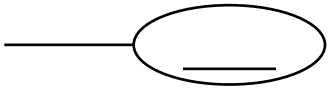


Notation Guide

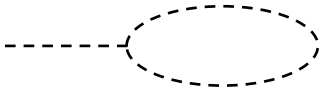
ENTITY TYPE



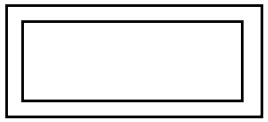
ATTRIBUTE



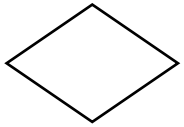
KEY ATTRIBUTE



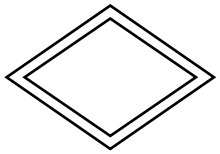
DERIVED ATTRIBUTE



WEAK ENTITY TYPE

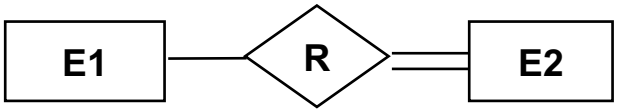


RELATIONSHIP TYPE

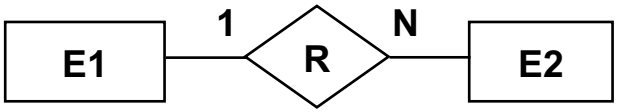


IDENTIFYING RELATIONSHIP TYPE

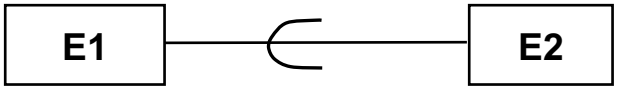
Notation Guide



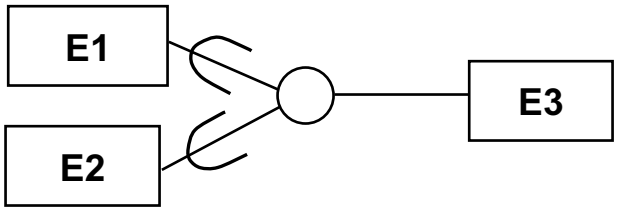
TOTAL PARTICIPATION OF E2 IN R



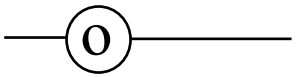
CARDINALITY RATIO 1:N FOR E1:E2 IN R



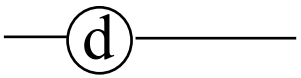
E1 IS A SUBCLASS OF E2



E1 and E2 ARE SUBCLASSES OF E3



Overlapping specialization



Disjoint specialization

Databases

The Entity-Relationship Model

Entity and Relationship

Relationship Constraints

Specialization and Generalization

Design Choices

Self-Practice: Draw an ER diagram

- The primary function of Uni is to offer 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 Uni
- Visiting students stay at Uni for a year.
- A course offered by Uni 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 registers in a course section and gets a mark for it.
- A course may have any number of other courses as prerequisites.

Conceptual Design Tips

- Design choices:
  - Should a concept be modeled as an entity or an attribute?
  - Should a concept be modeled as an entity or a relationship?
  - Relationships: Binary or ternary?
- 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

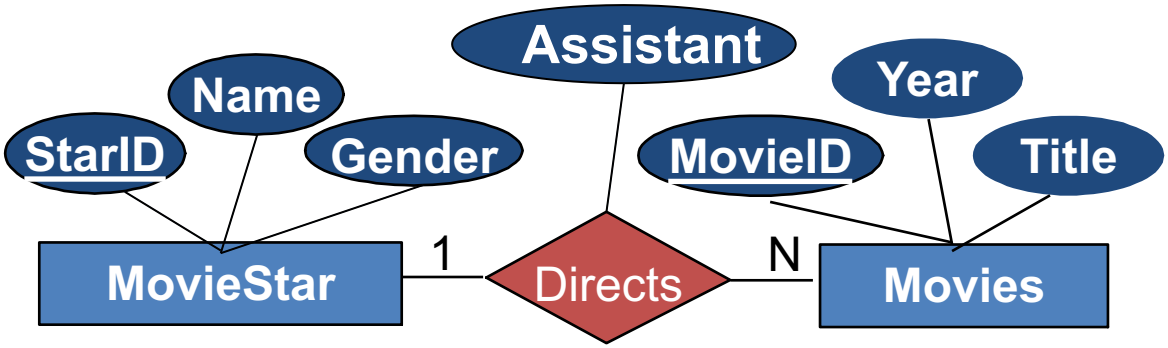
Tip 1: Entity vs. Attribute

- 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

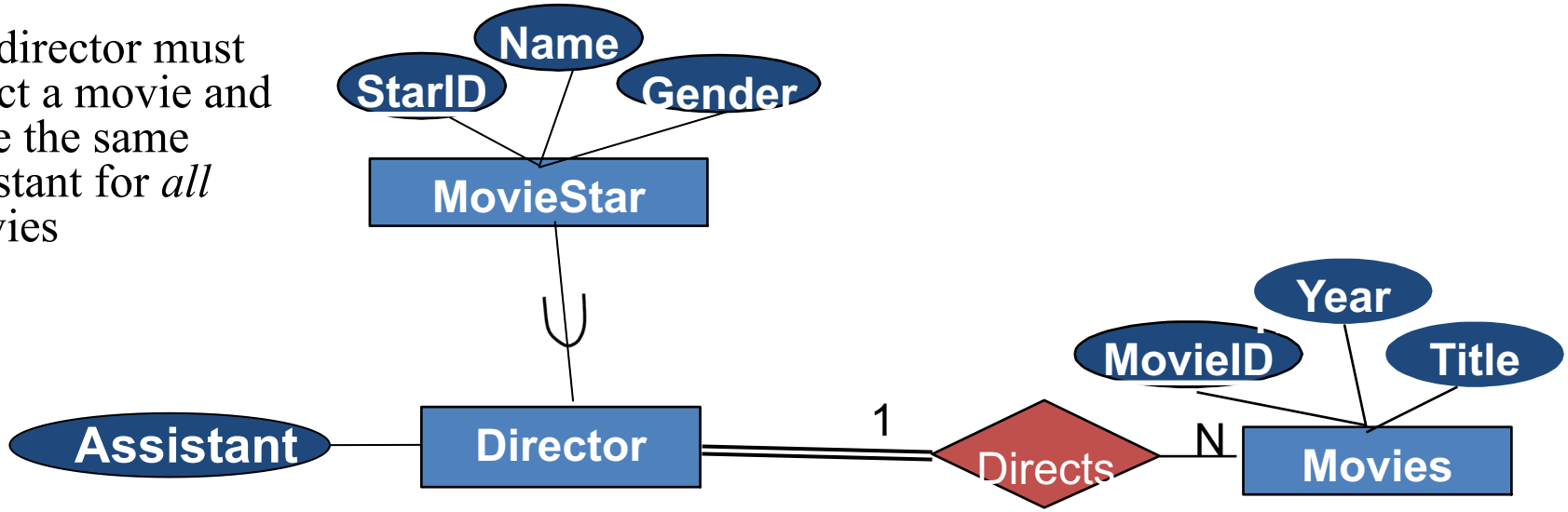
For example, if we have many MoviePepople entities that are couples/families and share the same address, then using attributes will lead to redundancy. In this case, modeling address via an entity is a better choice.

Tip 2: Entity vs. Relationship

- How are the two ER models different?
- Director can get a separate assistant for each movie.

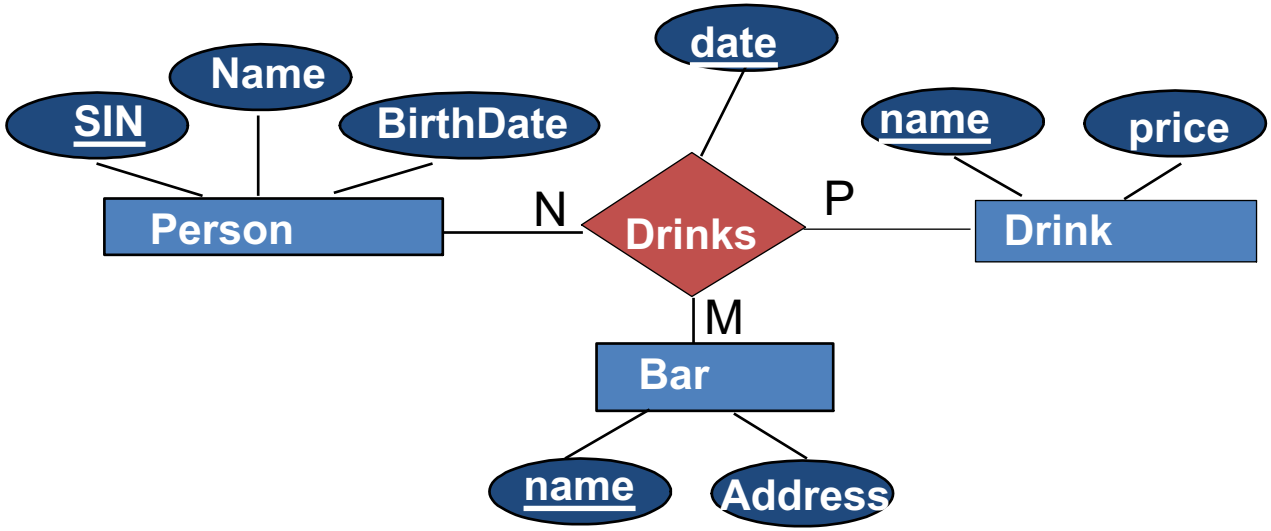


- All director must direct a movie and have the same assistant for *all* movies



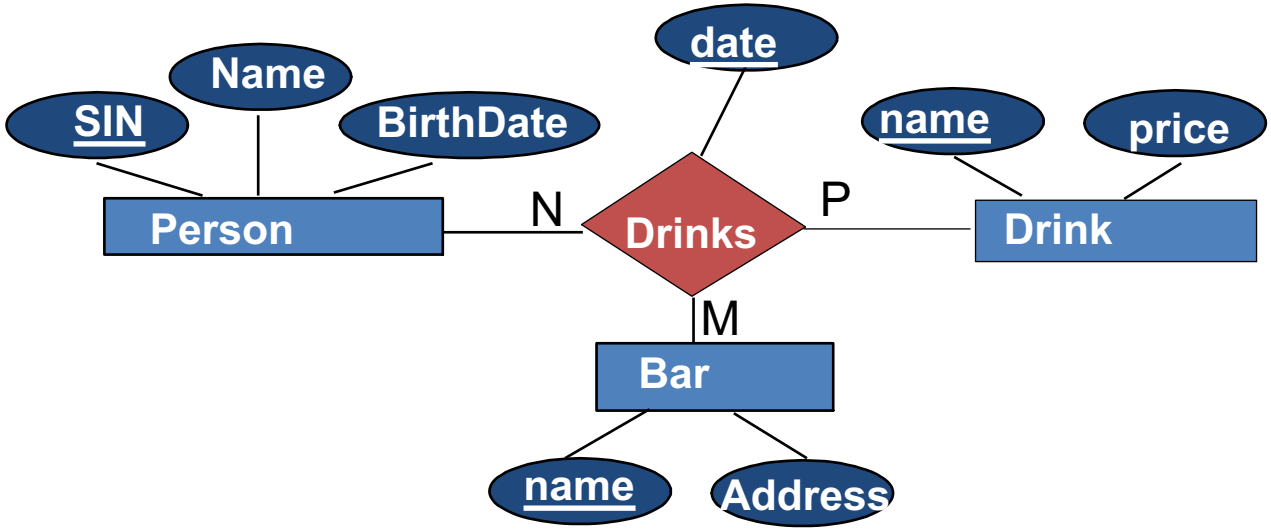
Tip 3: Binary vs. Ternary Relationships

- 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 binary relationships instead?

Tip 3: Binary vs. Ternary Relationships (Cont.)



- No combination of binary relationships is an adequate substitute:
  - Perons "likes" Drink, Person "visits" Bar, and Bar "provides" Drink does not imply that Person drinks Drink in Bar.
  - Also, how would we record *date*?

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*, *superclass & subclass relationships*.
- Note: There are many variations on ER model.

Summary of ER

- Several kinds of integrity constraints can be expressed in the ER model: *key constraints*, *participation constraints*, and *overlap/covering constraints* for "IS-A" 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 subclass hierarchies
- Ensuring good database design: resulting relational schema should be analyzed and refined further.

Sample solution