CPSC 304 – Sept 6/10, 2024 Administrative notes

- Please sign up for Piazza
  - Link is working now
- Please sign up for iClickers (we'll try to use them today!):
  - Section 101: https://join.iclicker.com/IUPB
  - Section 103: https://join.iclicker.com/MSUN
- Start forming your project groups
  - Read the Project and Milestone 0 description
- First "In Class" exercise will likely be next class
- Office hours to be posted soon (mine are up)
- Tutorials start the week of the 16th

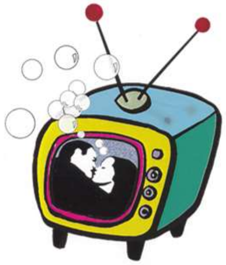
CPSC 304
Introduction to Database Systems

Conceptual Database Design
The Entity-Relationship Model

Textbook Reference

Database Management Systems: Chapter 2

Databases: the continuing saga…



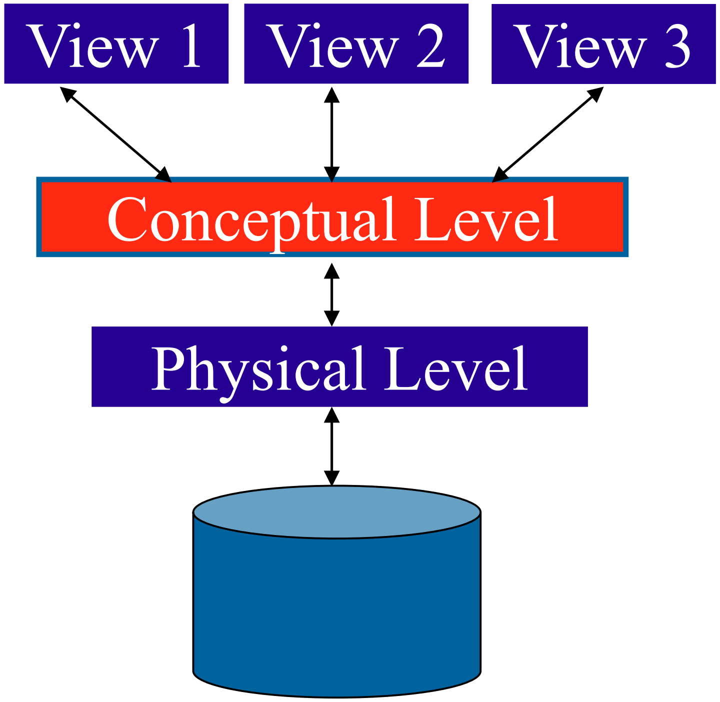
- We motivated that databases were great because they:
  - Store large amounts of data
  - Handle transactions
  - Allow efficient querying
  - *And many, many more classic favourites!*
- Before we can do all of these, we must design the database

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:
  - **External (or View) level**: describes different part of the database to different users
    - convenience, security, etc.
    - Compare views of student, registrar, and database admin.
  - **Conceptual (or Logical) level**: how data is perceived by the users
  - **Physical level**: how data is actually stored (404) – covers things like indexes, bits on disk



Schema and Instances

- We create the conceptual **schema** – the logical structure of the database (e.g., students take courses)
- Later we'll populate **instances** – the content of the database at a particular point in time

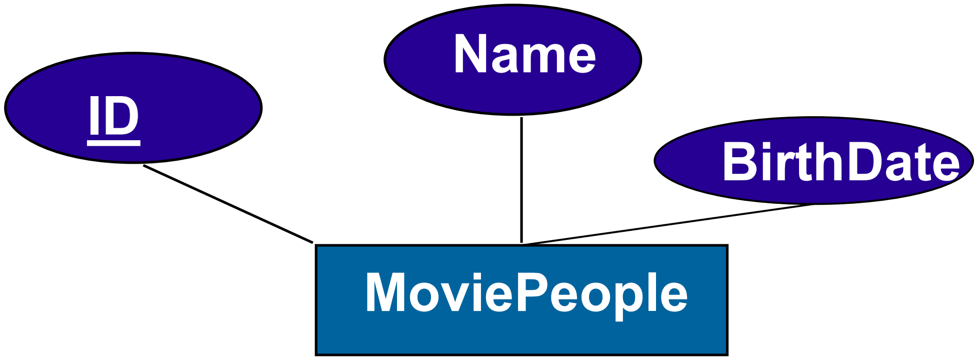
- E.g., Gradebook schema is set, but currently there are no grades for CPSC 304

Conceptual Database Design

- What are the entities and relationships in the enterprise?
  - Entities are usually nouns
    - e.g., Students and Courses
  - Relationships are statements about 2 or more objects. Often, verbs.
    - e.g., an instructor 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

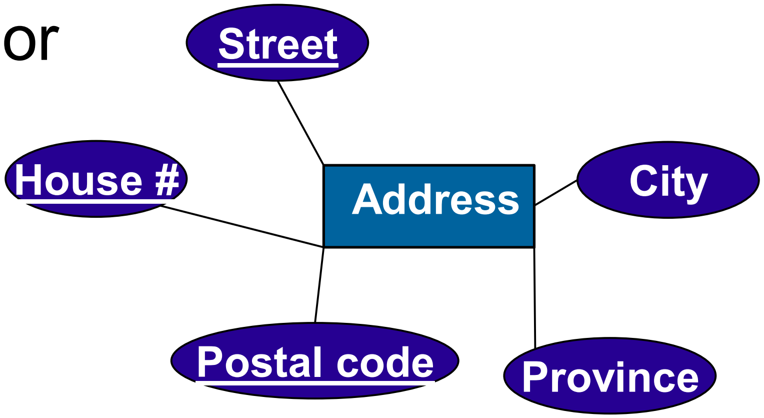
- ***Entity:*** Real-world object distinguishable from other objects.
- An entity is described using a set of ***attributes***.



- ***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***. The key is composed of all underlined attributes

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.
  - In Canada, ~50 addresses share the same postal code

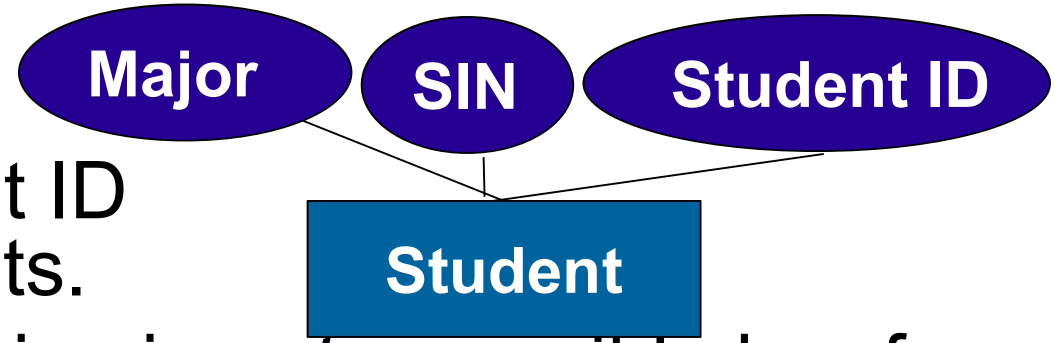


- A *primary key* is the key chosen as the principal means to identify entities in an entity set
- The only keys shown in ER diagrams are primary keys (do not worry about this for now)
- We'll discuss superkeys when we consider normal forms (for now, don't worry about them)

Keys: clicker question

Consider a student entity loosely based on the real world:

You may assume both SIN & Student ID uniquely ID students.



Which of the following is *not* a possible key for the student entity?

A. SIN & Student ID together

B. SIN

C. Student ID

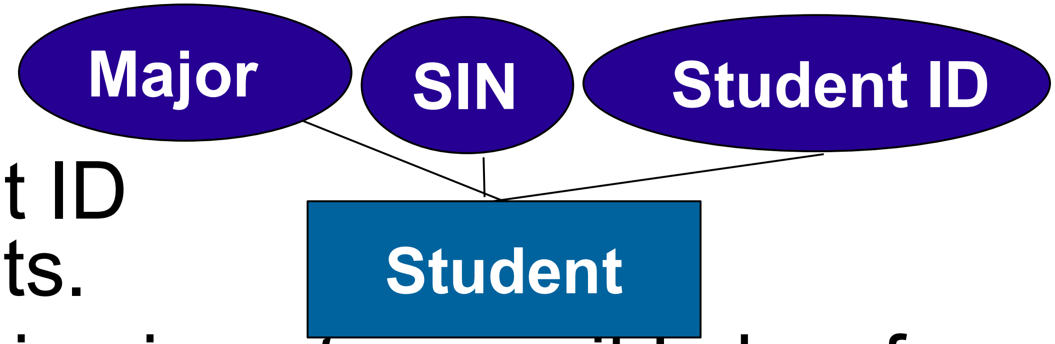
D. All are possible

E. More than one is not possible

Keys: clicker question

Consider a student entity loosely based on the real world:

You may assume both SIN & Student ID uniquely ID students.

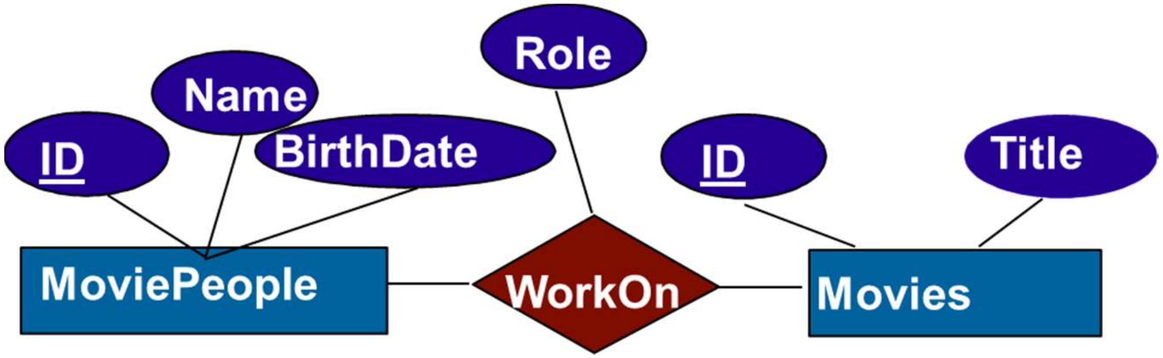


Which of the following is *not* a possible key for the student entity?

A. [✓ SIN & Student ID together]
B. SIN
C. Student ID
D. All are possible
E. More than one is not possible

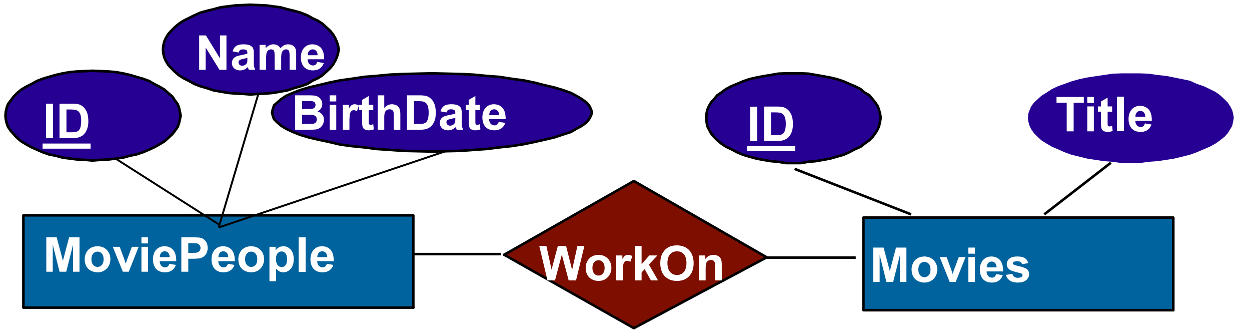
Keys must be minimal!

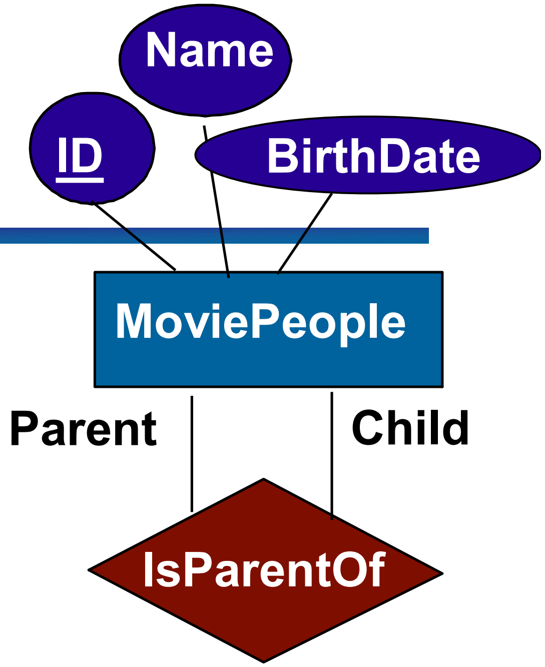
ER Model Basics (Cont.)



- ***Relationship*:** Association among two or more entities.
  - E.g., Michelle Yeoh worked on Everything Everywhere All at Once.

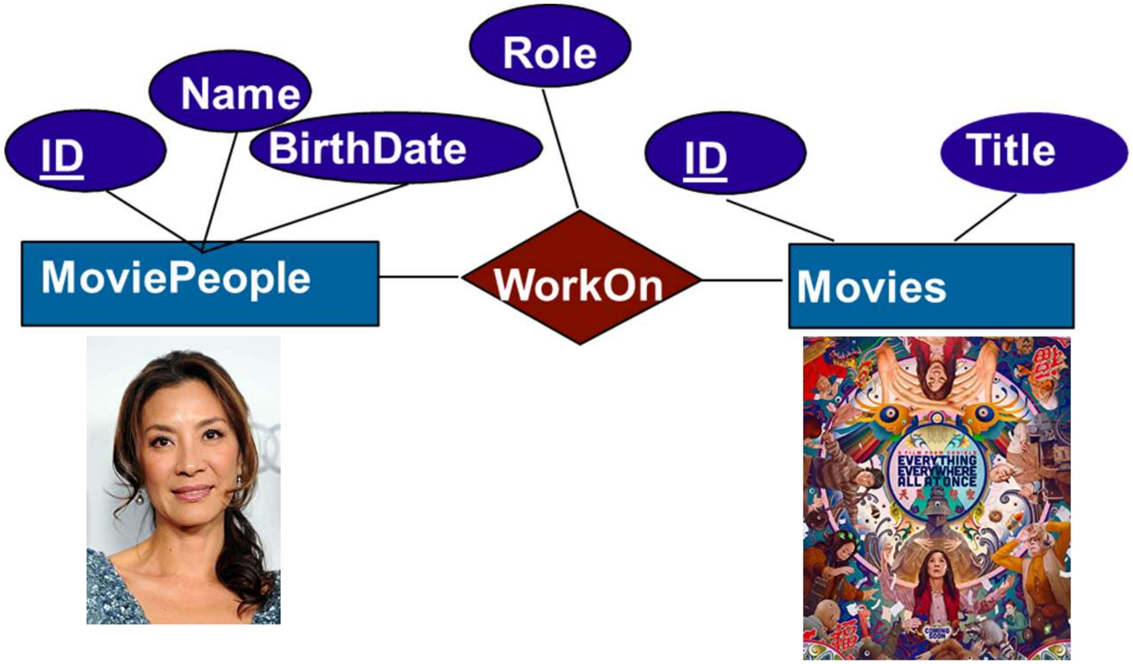
ER Model Basics (Cont.)





- ***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.)



- A relationship must be uniquely defined by the entities involved (it may not be a key because it may not be minimal, as we'll see shortly)
- A relationship set may have ***descriptive attributes*** (like Role).
- An n-ary relationship set R relates n entity sets $E_1 \dots E_n$; each relationship in R involves entities $e_1 \in E_1, \dots, e_n \in E_n$
  - ***Degree*** or ***arity:*** # of entity sets in relationship (binary, ternary, etc.)

Exercise (not to turn in): Workday Student – (minus minus)

- Design a database for part of Workday Student: 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, "CPSC100: Computational Thinking" has department = CPSC, number = 100, and title = “Computational Thinking."
- Numbers are assigned by departments, and different departments may use the same number.
- Students are represented by their (unique) student ID and their name.
- "Enrolls" 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?

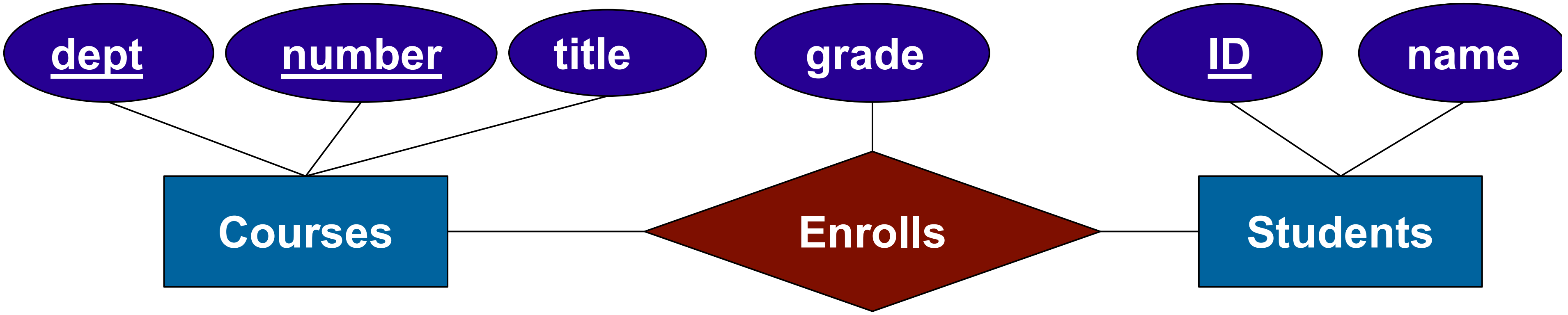
A. Entity set Students with attribute ID not underlined and name underlined.

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



Can a student take a course twice? A. Yes [✓ B. No]

Okay, that worked pretty well…

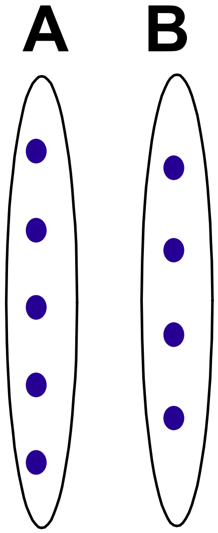
But what if we wanted to add instructors to courses and restrict that each could only have one instructor?

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: student, B: student ID #



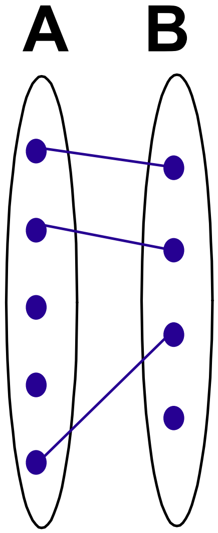
Entity sets A and B

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:

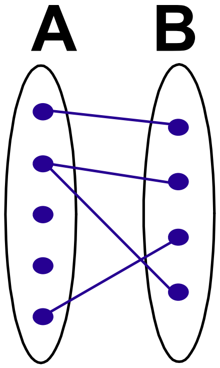
1. ***one-to-one*** from A to B:
   - an entity in A is associated with **at most one** entity in B and vice versa
   - e.g. A: student, B: student ID #



1-to-1

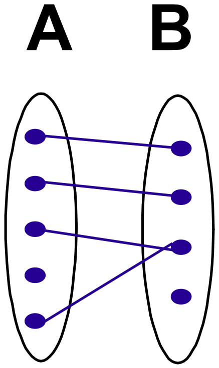
Cardinalities (cont')

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)*
   - an entity in B is associated with any number of entities in A
   - an entity in A is associated with at most one entity in B



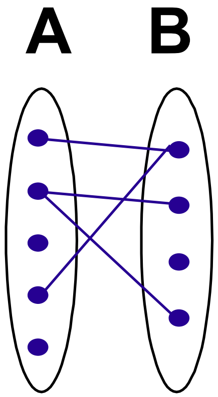
Many-to-1

Cardinalities (cont’)

zero one many
cardinality

4. ***many-to-many*** *from A to B:*

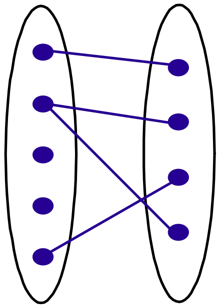
- *an entity in A is associated with any number of entities in B and vice versa*
- *e.g. A: students, B: courses*



Many-to-Many

Cardinalities exercise #1

What cardinality constraint does this diagram best represent:



A. One-to-one

[✓ B. One-to-many]

C. Many-to-one

D. Many-to-many

Cardinalities exercise #2

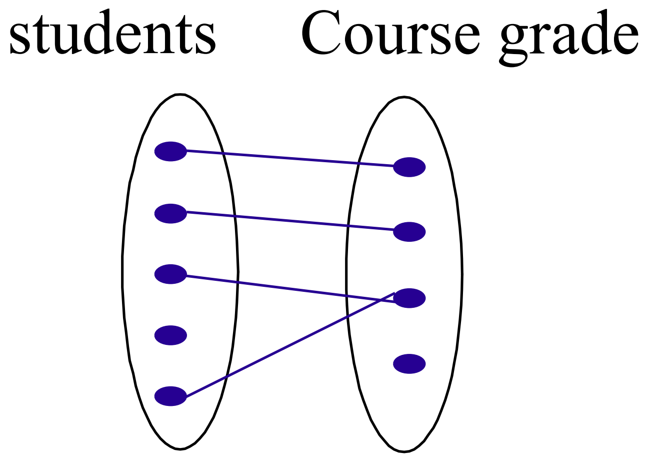
Which cardinality constraint is most appropriate for the relationship of students to overall course grades in a section of CPSC 304?

A. One-to-one

[✓ B. Many-to-one]

C. One-to-many

D. Many-to-many

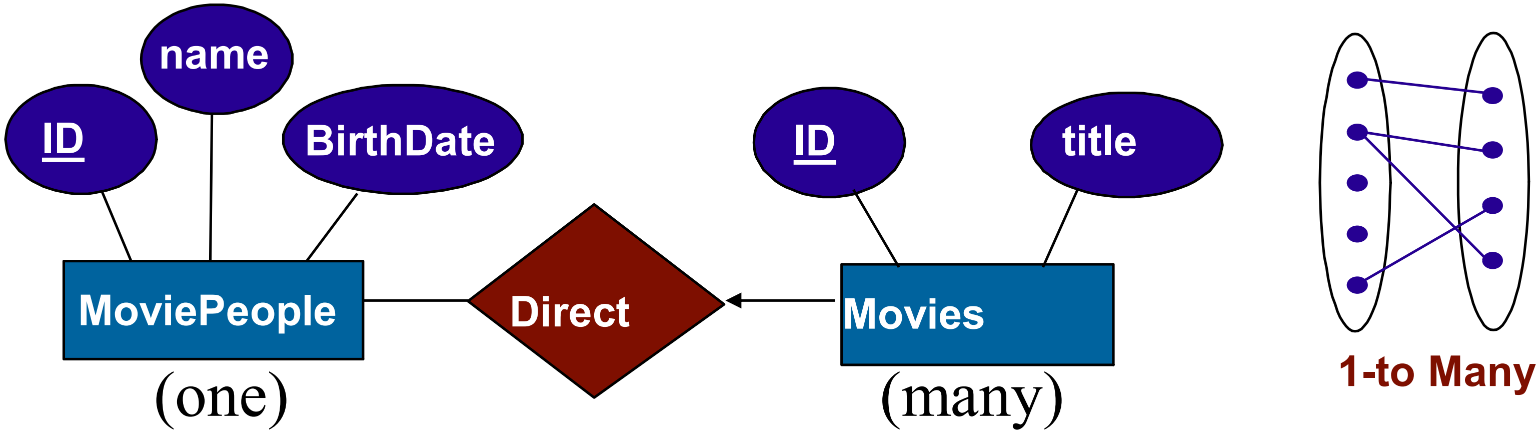


Many-to-1

Careful! This assumes that each student only gets a grade in CPSC 304 once – so everyone pass so this stays true, eh?

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



name

ID

BirthDate

MoviePeople

(one)

Direct

ID

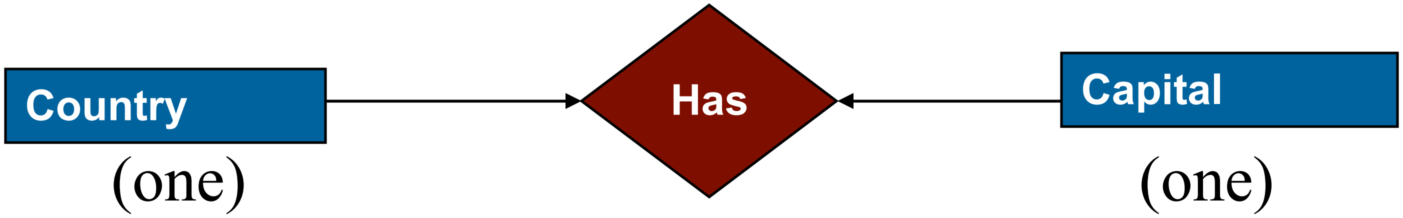
title

Movies

(many)

1-to Many

or



Country

(one)

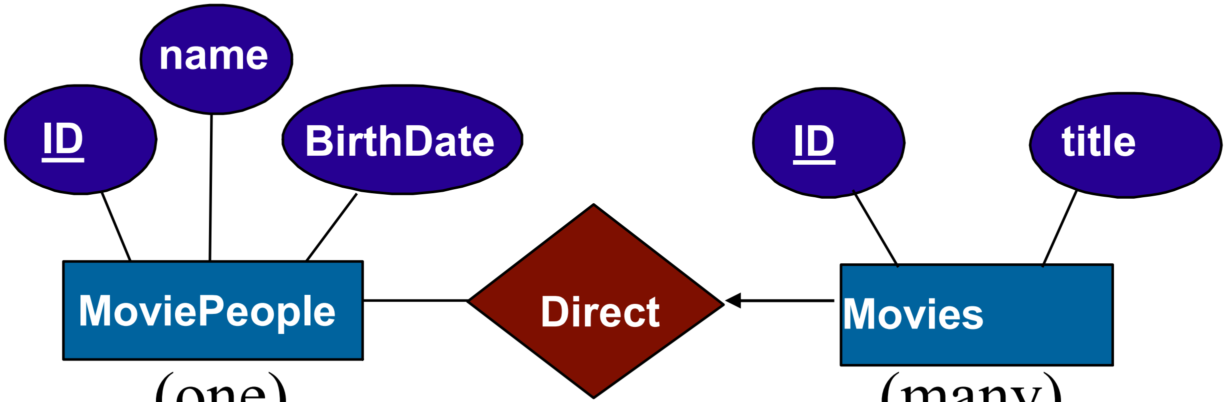
Has

Capital

(one)

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)” – the arrow points to the thing there is only one of



name

ID

BirthDate

MoviePeople

Direct

ID

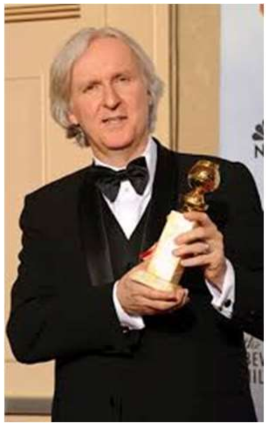
title

Movies

(one)

(many)

James Cameron

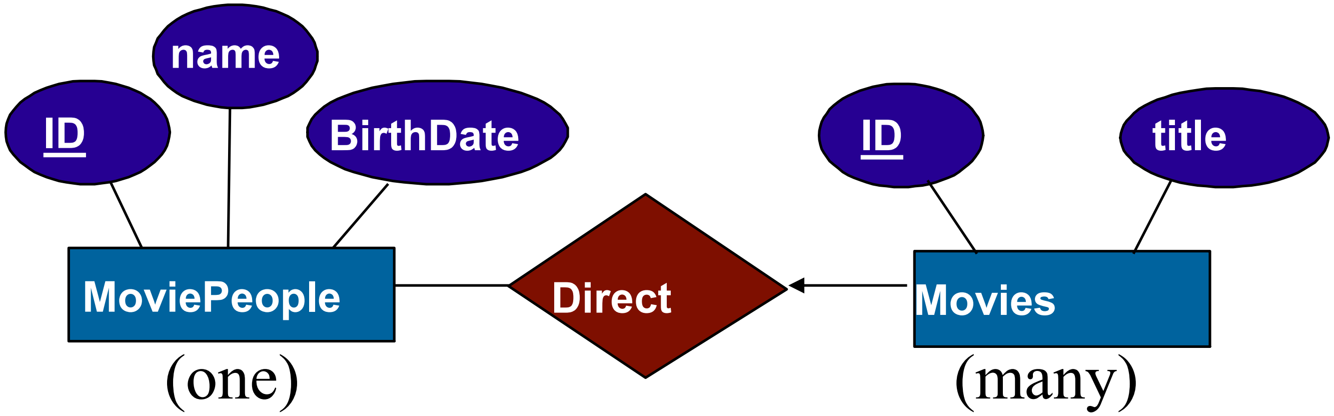


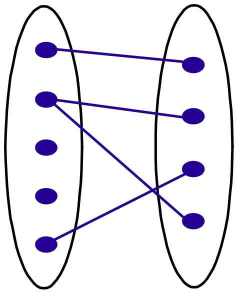
Clicker Exercise

In a one to many relationship like the one below, which statement is true?

[✓ A. If I know the movie, I definitely know who directed it]

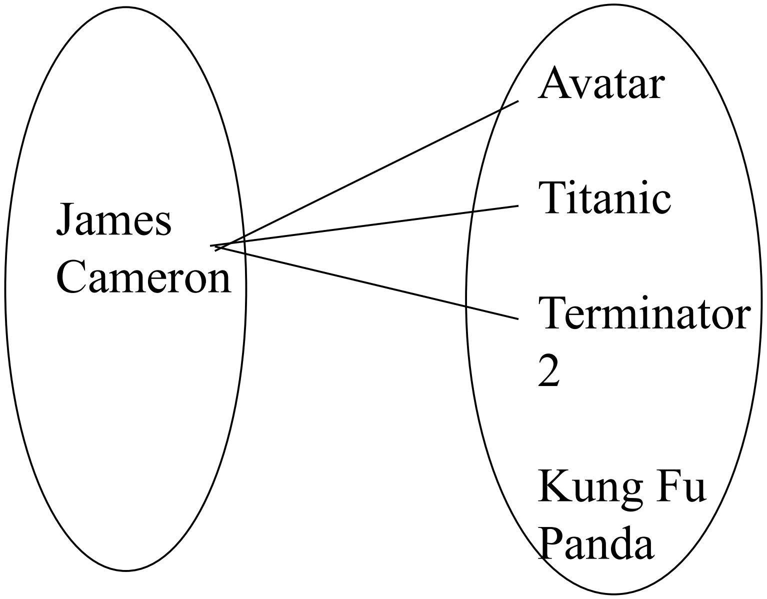
B. If I know the director, I definitely know which movie is being referred to in the Direct relationship





1-to Many

Clicker Exercise Explained

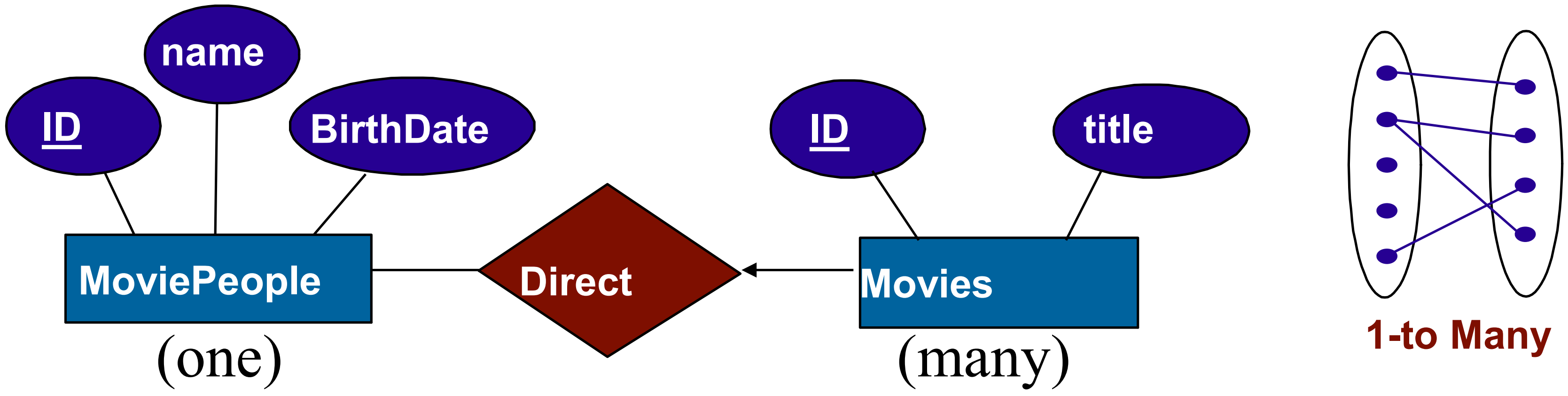


Imagine I wanted to talk about a specific MoviePeople Direct Movies relationship.

If you know which MoviePerson I am talking about (e.g., James Cameron), can you definitively determine which movie is participating in the relationship I am interested in?

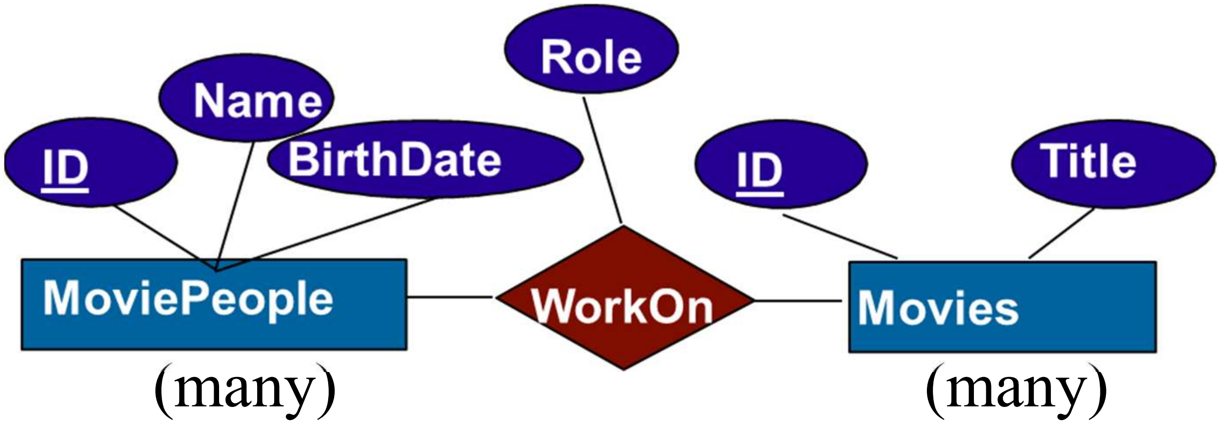
If you know the movie (e.g., Avatar), can you definitively say James Cameron was the director?

Clicker Exercise Explained



In other words, the key for a one to many relationship is the key of the entity that is on the many side!

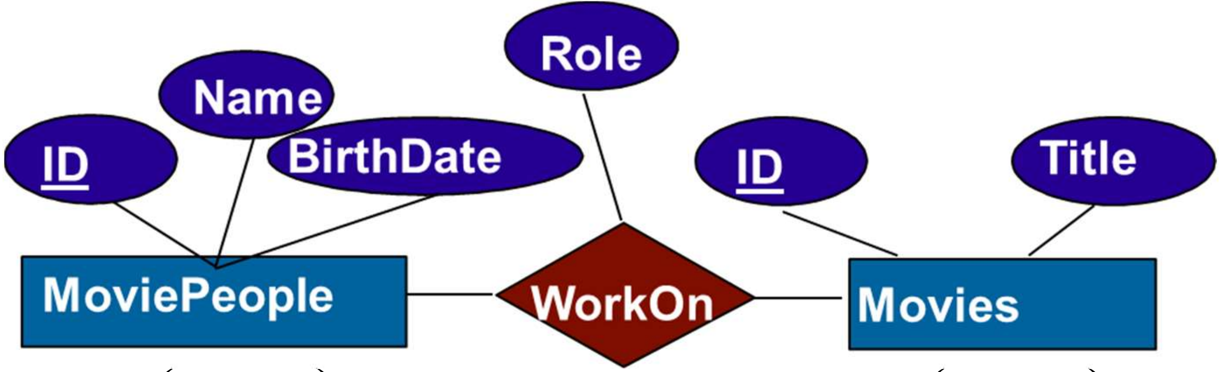
How can we uniquely identify a relationship?



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

Robert Pattinson
as
Bruce Wayne

How can we uniquely identify a relationship?



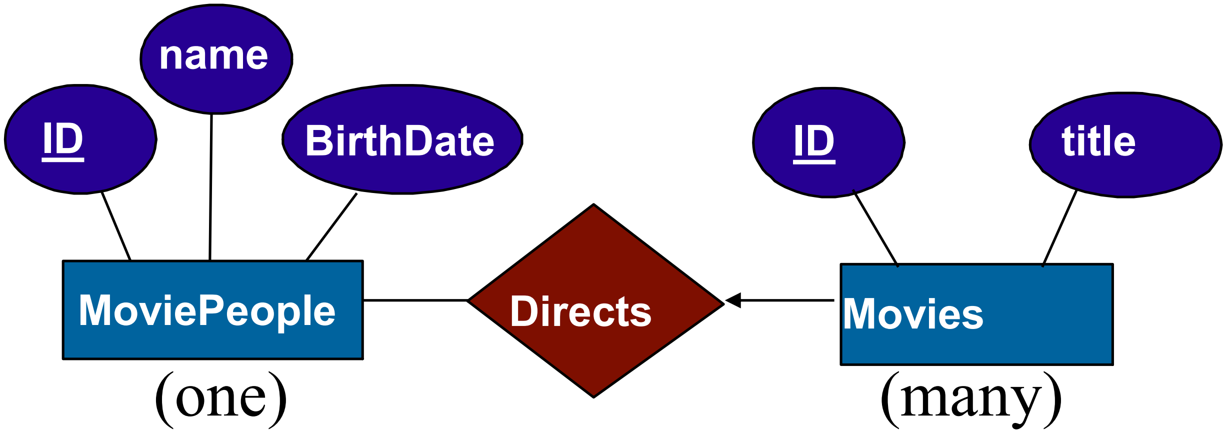
(many) (many)

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

Robert Pattinson
as
Bruce Wayne

Clicker question: can this diagram represent that Robert plays both Bruce Wayne and Batman in the "The Batman"? A. Yes. [✓ B. No.]

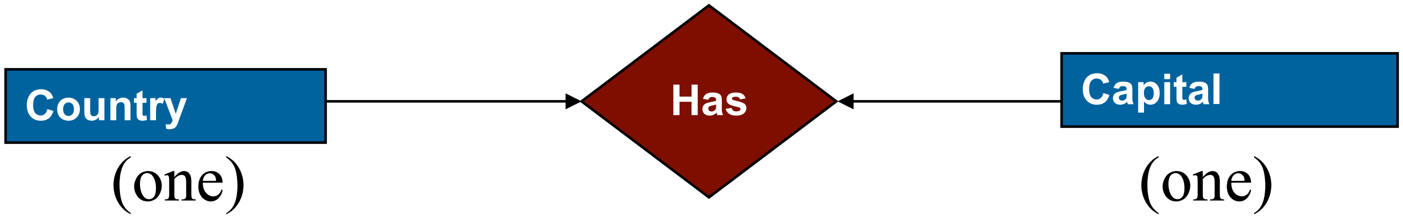
How can we uniquely identify a relationship?



James Cameron

The key of a many to one relationship is the key of the entity on the many side.

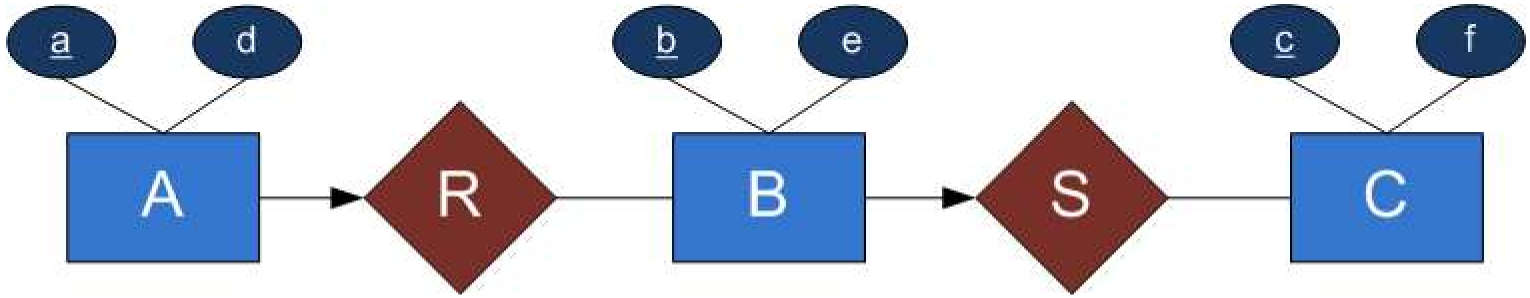
How can we uniquely identify a relationship?



Ottawa

The key of a one-to-one relationship is the key of ONE of 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

Got to here