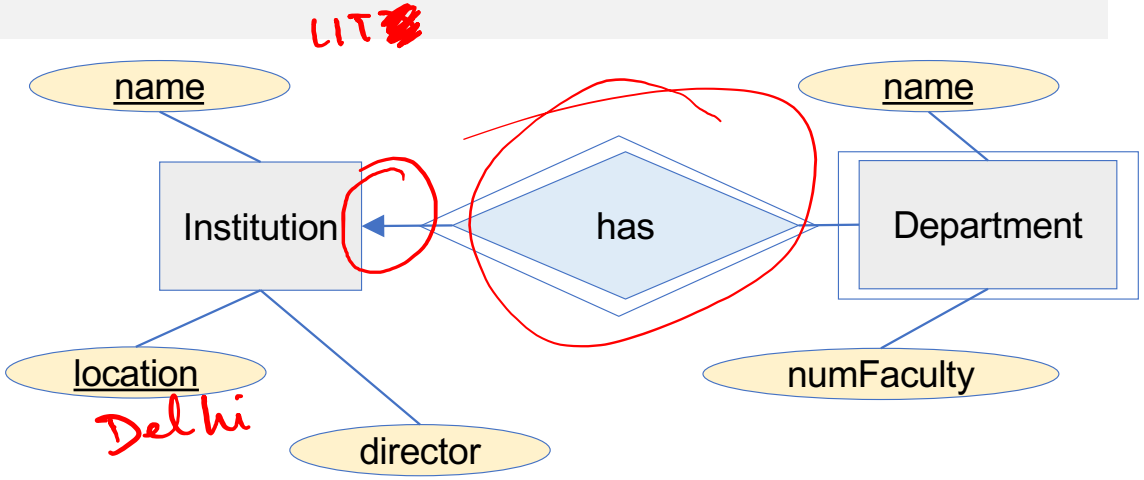
COL 362 & COL 632

Relational

11 Jan 2023

Weak entity sets to relations

- Relation for a **weak entity set** must include attributes for its **complete key** (including those belonging to other entity sets), as well as its own, nonkey attributes.
- A supporting relationship is redundant and yields no relation (unless *it* has attributes).

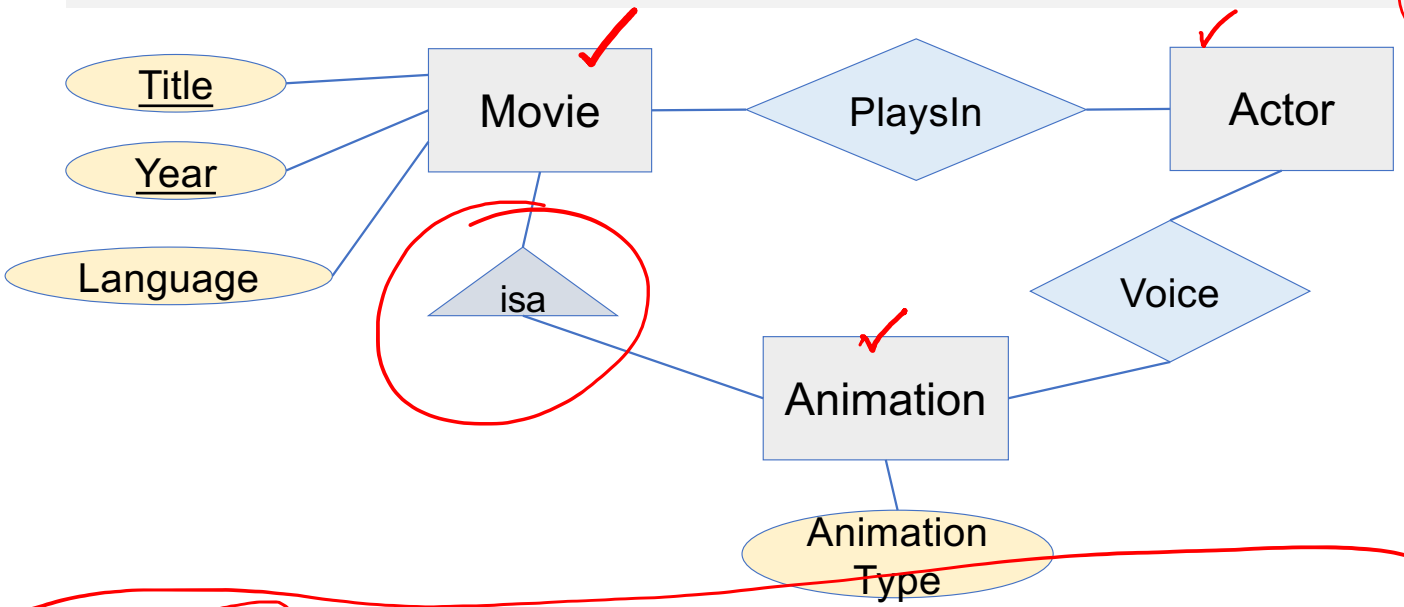


Institution (Name, Location, Director)

Department (Name, InstName, Instlocation, NumFaculty)

~~**Has (DeptName, InstName, Instlocation)**~~

Hierarchies to relations



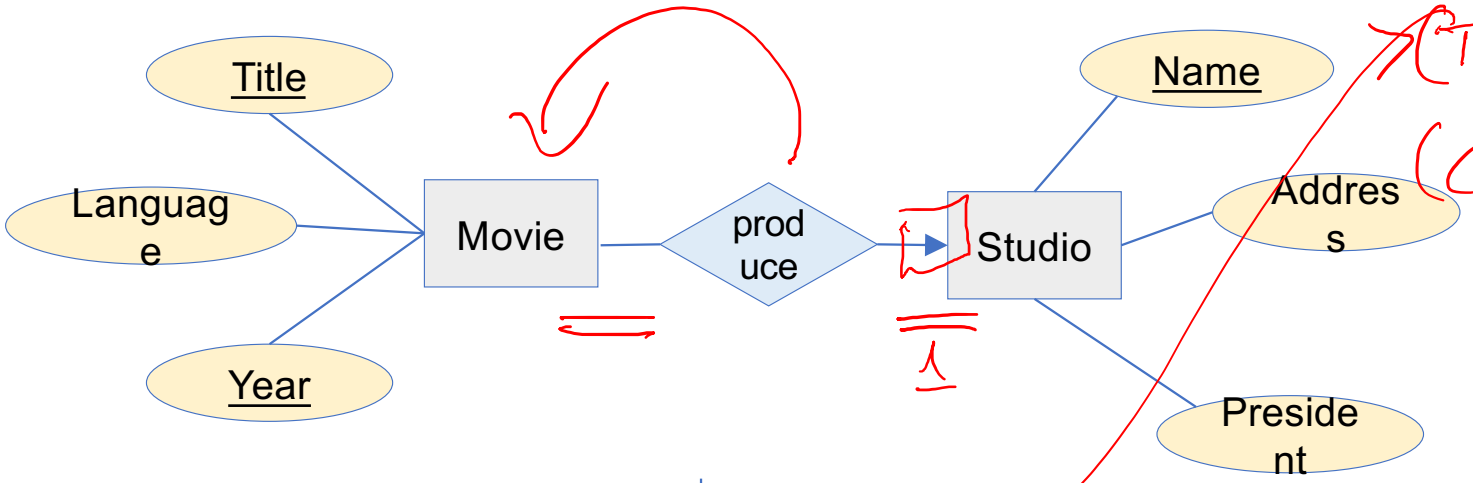
Movie (Title, Year, Language)

Actor (Name, DOB, City)

Animation (Title, Year, AnimationType)

AllMovies (Title, Year, Language, AnimationType)

Combining relations (1/2)



Movie (Title, Year, Language)

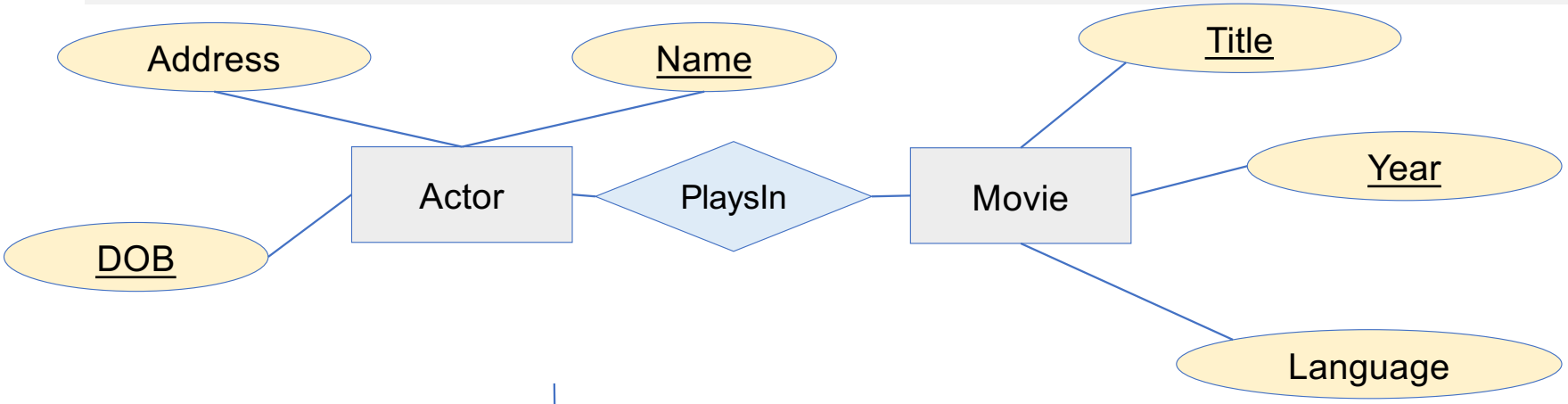
Studio (Name, Address, President)

Produce (Title, Year, StudioName)

Movie (Title, Year, Language, StudioName)

Studio (Name, Address, President)

Combining relations (2/2)



Actor (Name, DOB, Address)

Movie (Title, Year, Language)

PlaysIn (Name, DOB, Title, Year)

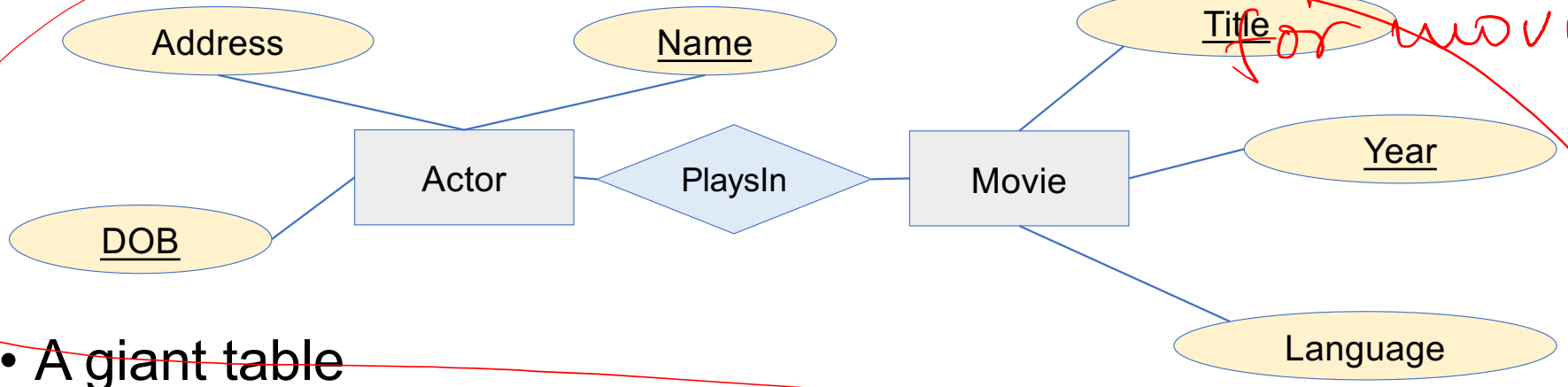
Actor (Name, DOB, Address, MovieTitle, Year)

Movie (Title, Year, Language)

Actor (Name, DOB, Address)

Movie (Title, Year, Language, ActorName, DOB)

Even More Extreme



- A giant table
  Actor (Name, DOB, Address, MovieTitle, Year, Language)

What is a good database design?

Anomalies

Name	DOB	Address	MovieTitle	Year	Language
Ayushman Khurana	1984	Mumbai	DoctorG	2022	Hindi
Ayushman Khurana	1984	Mumbai	Andhadhun	2018	Hindi
~~Henry~~ Cavill	1983	Beverley Hills	Man of Steel	2013	English
Tom Cruise	1962	LA	Top Gun: Maverick	2022	English
Peter Sellers	1925	London	~~Dr. Strangelove~~	1964	English

NULL

- Redundancy
- Update Anomalies
  - Ayushman Khurana changed his name to to Ayushmaan Khurrana
- Deletion Anomalies
  - Delete the movie “Dr. Strangelove” from the DB

Normalization is the process of systematically eliminating these anomalies => Leads us to better database designs

Functional Dependencies

Key Constraint- student in one dept
Instructor in at least one dept
Department has a HOD office

- There are usually a variety of constraints (rules) on the data in the real world.
- For example, some of the constraints that are expected to hold in a university database …

- **Legal instance of a relation:** an instance that satisfies all real-world constraints
  - A legal instance of a database = all relations are legal
- Require that the **value for a certain set of attributes** determines **uniquely the value for another set of attributes**.
  - A functional dependency is a generalization of the notion of a key.

Functional Dependencies

- A functional dependency is another kind of constraint
- If two tuples in a relation agree on the values of one set of attributes, then they must also agree on the values of another set of attributes.

$$R(A_1, A_2, A_3, B_1, B_2, B_3)$$

$$\begin{aligned} A_1A_2A_3 &\rightarrow B_1 \\ A_1A_2A_3 &\rightarrow B_2 \\ &\cdots \\ A_1A_2A_3 &\rightarrow B_1B_2B_3 \end{aligned}$$

Shorthand

Functional Dependencies

- Let R be a relation schema

  $$\alpha \subseteq R \quad and \quad \beta \subseteq R$$

- The **functional dependency**

  $\alpha \rightarrow \beta$ **holds on** R **if and only if**
  for any legal relations r(R), whenever any two tuples t_1 and t_2 of r agree on the attributes α, they also agree on the attributes β.

That is,

$$t_1[\alpha] = t_2[\alpha] \Rightarrow t_1[\beta] = t_2[\beta]$$

- Example: Consider r(A,B) with the following instance of r.

1	4
1	5
3	7

- On this instance, $B \rightarrow A$ hold; $A \rightarrow B$ does **NOT** hold,

Functional Dependencies

- Example (figure out the right FDs)

Actor (Name, DOB, Address)

Movie (Title, Year, Language, StudioName)

PlaysIn (Name, DOB, Title, Year)

Name DOB → Address
Title Year → Language StudioName
Title → Language
Language → Title Year
Name DOB → Title Year

Actor (Name, DOB, Address, MovieTitle, Year, Language)

Name DOB → Address
Name DOB Address → Language
Name DOB MovieTitle Year → Address
MovieTitle Year DOB → Name
DOB Address MovieTitle → Name
DOB Address MovieTitle Year → Name

Can you figure out the functional dependencies from the data?

Trivial and non-trivial FDs

Actor (Name, DOB, Address)

Movie (Title, Year, Language)

PlaysIn (Name, DOB, Title, Year)

Name DOB → Address
Name DOB → Name
Name DOB → Name Address

In general, $\alpha \rightarrow \beta$ is trivial if $\beta \subseteq \alpha$

Keys and superkeys

Movie (Title, Year, Language, Length)

{Title, Year}

Key
- Functionally determines *all* other attributes
- Minimal

Keys need not be unique

{Title, Year, Language}

Superkey
- Functionally determines *all* other attributes
- Not necessarily *minimal*

Terminology

- Key = Candidate Key
- Superkey
- Primary key
- Prime attribute

Inferring FDs

- Given a set of FDs, which other FDs follow from it?
- Example:
  - Given: Name DOB → Address
    Address → City
  - Inferred: Name, DOB → City

Inferred through *transitivity* of FDs

Armstrong's Axioms

- Reflexivity

  If B is a subset of A, then A → B
- Augmentation

  If A → B, then AC → BC
- Transitivity

  If A → B and B → C, then A → C

Closure of FDs

- Given: *S*, the set of FDs
- Output: S^+, the *closure* of S, containing all FDs derivable from S

- Example:

AB → C

C → ED

Basis

Is this set S^+

AB → ED

AB → E

ABC → ED

C → E

C → D