The Relational Model

Only one structure – relation

- A relation is both a mathematical concept and just a table of values

- The relational model models “everything” as relations

Example

Actor-Movies

Name	**Movie**	**Character**
Priyanka Chopra	Baywatch	Victoria Leeds
Tom Cruise	MI-I	Ethan Hunt
Anthony Hopkins	Thor: Ragnarok	Odin

Schema of the relation (without types):

Actor-Movies (Name, Movie, Character)

More about relations

- A relation is a *set* of tuples, not a *bag*
- Permuting the order of attributes does not matter

Name	**Movie**	**Character**
Priyanka Chopra	Baywatch	Victoria Leeds
Tom Cruise	MI-I	Ethan Hunt
Anthony Hopkins	Thor: Ragnarok	Odin

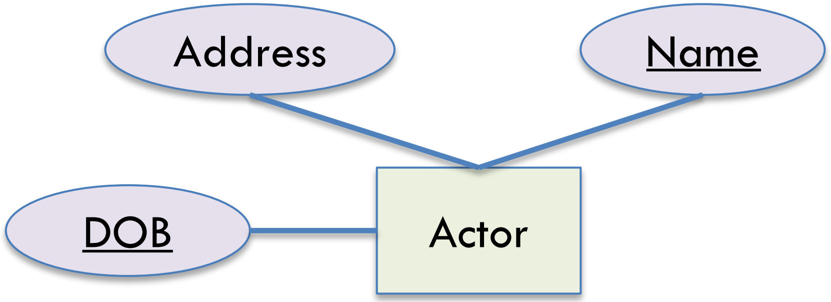
=

Character	**Movie**	**Name**
Victoria Leeds	Baywatch	Priyanka Chopra
Ethan Hunt	MI-I	Tom Cruise
Odin	Thor: Ragnarok	Anthony Hopkins

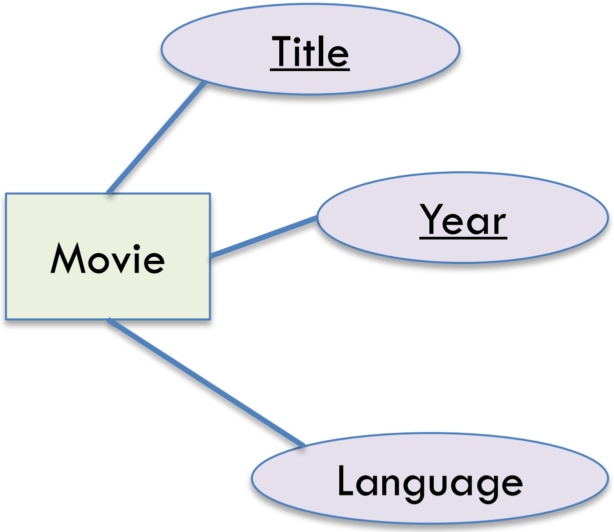
ER to relational

- ER diagrams are easy to comprehend and closer to how we think
- Relational model is powerful because it is simple – only one kind of object
  - Any operation on the relation, results in yet another relation
- So, let’s convert our ER diagrams to relational!

Entity sets/attributes

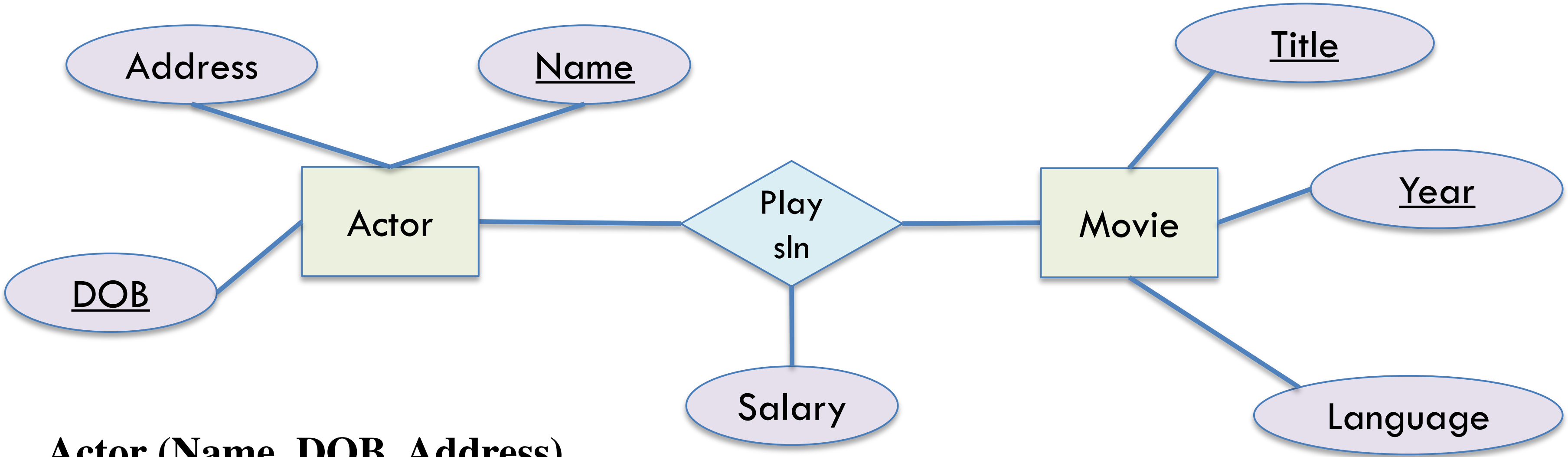


Actor (Name, DOB, Address)



Movie (Title, Year, Language)

Relationships to relations



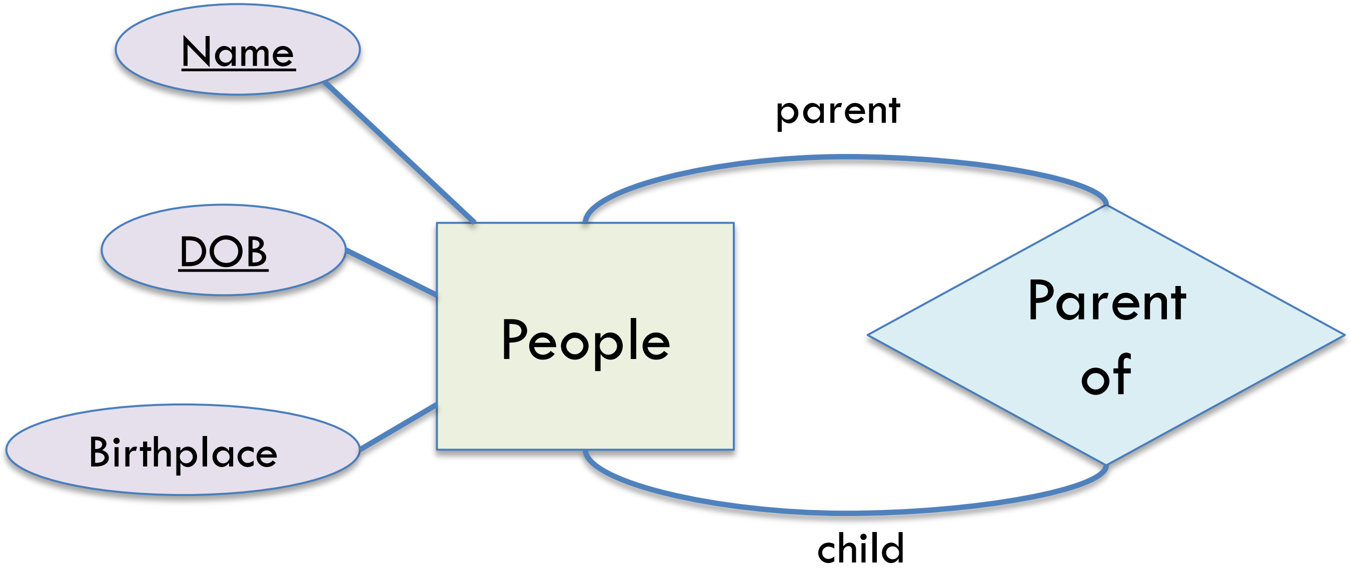
Actor (Name, DOB, Address)

Movie (Title, Year, Language)

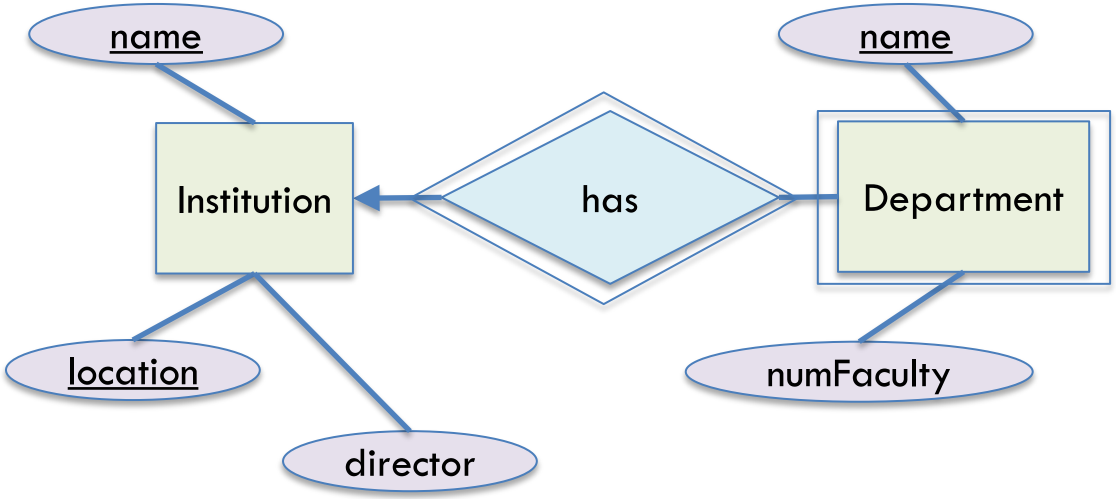
PlaysIn (Name, DOB, Title, Year)

PlaysIn (Name, DOB, Title, Year, Salary)

Roles to relations



Weak entity sets to relations

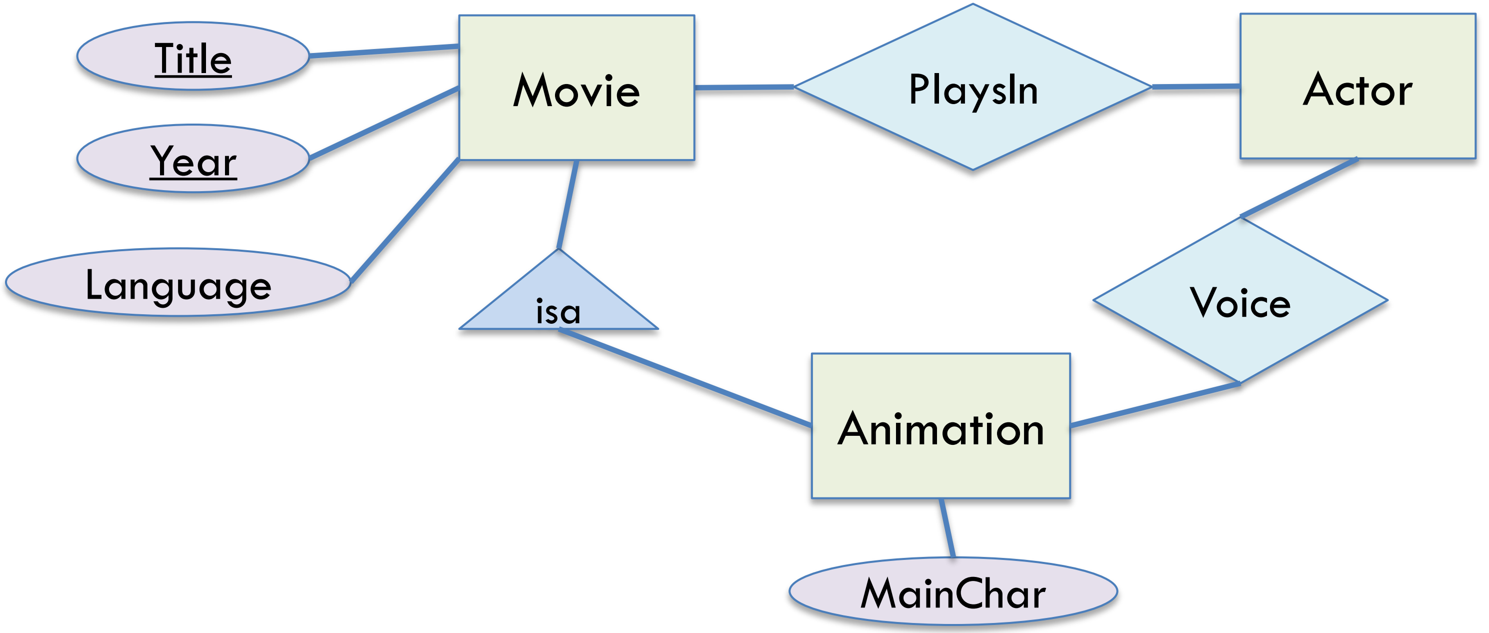


Institution (Name, Location, Director)

Department (Name, InstName, Instlocation, NumFaculty)

Has (DeptName, InstName, Instlocation)

Hierarchies to relations



Movie (Title, Year, Language)

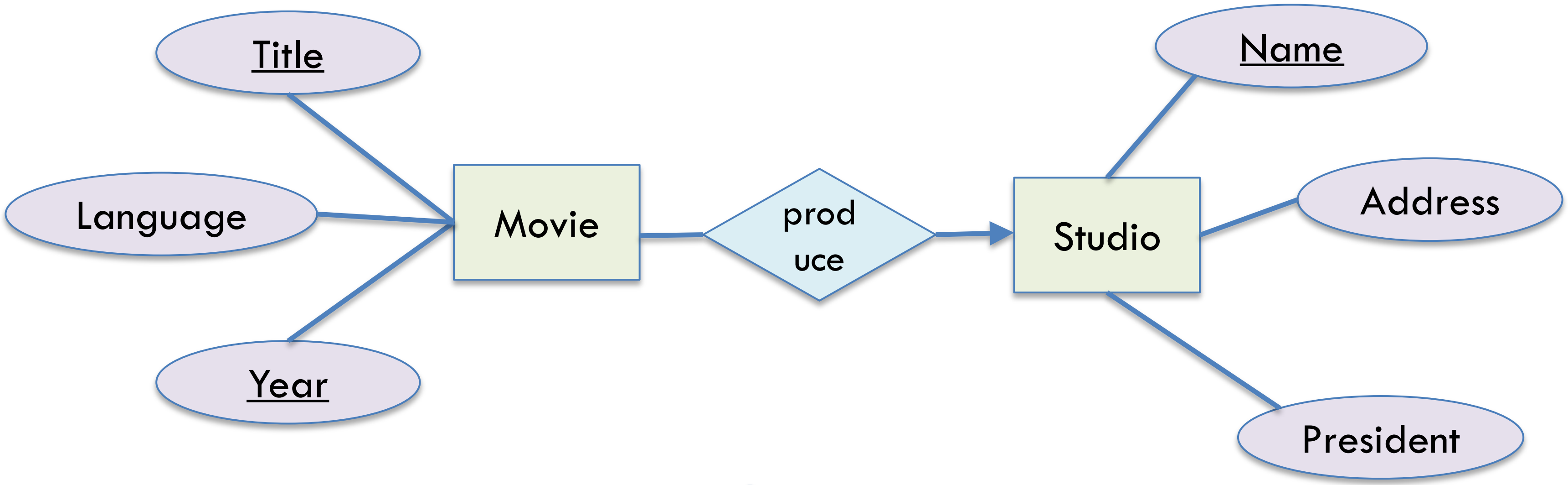
Actor (Name, DOB, City)

Animation (Title, Year, MainChar)

AnimationOnly (Title, Year, Language, MainChar)

AllMovies (Title, Year, Language, MainChar)

Combining relations (1/2)



Movie (<u>Title</u>, <u>Year</u>, Language)

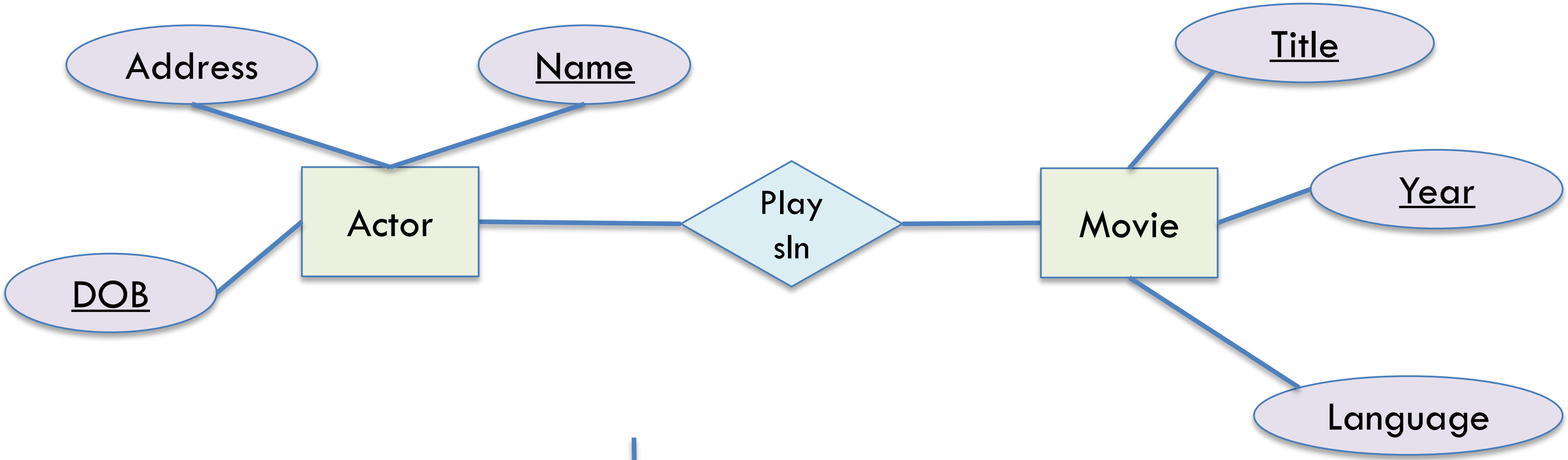
Studio (<u>Name</u>, Address, President)

Produce (Title, Year, StudioName)

Movie (<u>Title</u>, <u>Year</u>, Language, StudioName)

Studio (<u>Name</u>, 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)

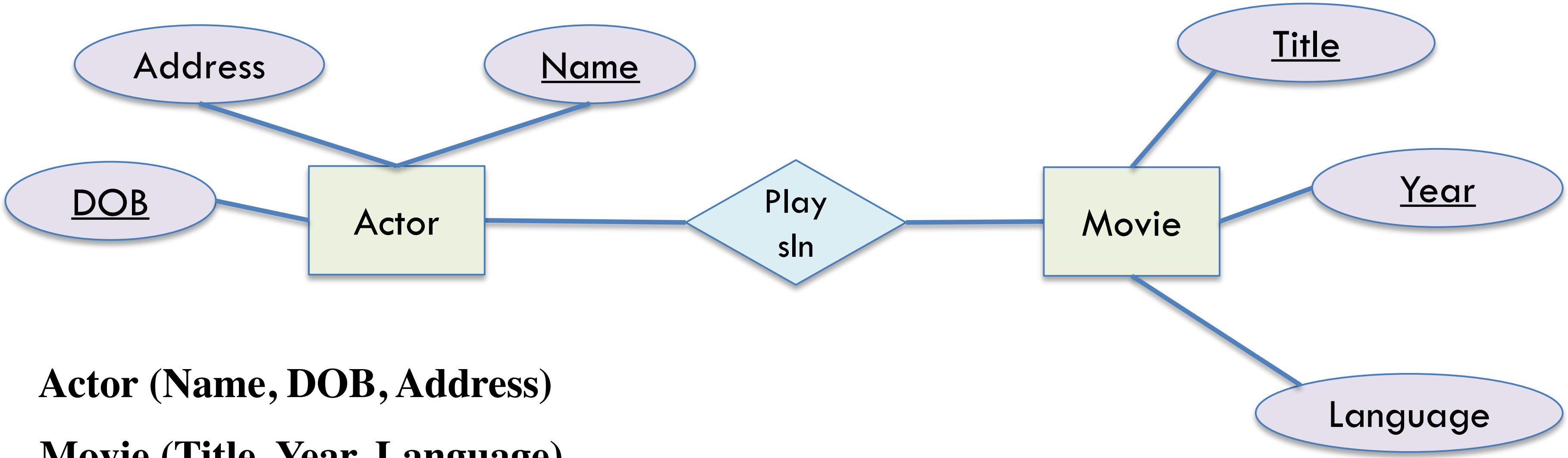
Movie (Title, Year, Language)

Actor (Name, DOB, Address)

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

NORMALIZATION

Designing good schemas



Actor (Name, DOB, Address)

Movie (Title, Year, Language)

PlaysIn (Name, DOB, Title, Year)

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

Anomalies

Name	DOB	Address	MTitle	Year	Language
Priyanka Chopra	1992	Mumbai	Don	2006	Hindi
Priyanka Chopra	1992	Mumbai	Don II	2011	Hindi
Anthony Hopkins	1937	LA	Thor: Ragnarok	2017	English
Tom Cruise	1962	LA	Valkyrie	2008	English
Bill Nighy	1949	LA	Valkyrie	2008	English

- Redundancy
- Update Anomalies
  - Priyanka Chopra’s changed to Priyankaa Chopra
- Deletion Anomalies
  - Delete the movie “Valkyrie” from the DB

Normalization is the process of systematically eliminating these anomalies

Functional Dependencies (1/2)

- 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.

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

$$\begin{aligned} A_1 A_2 A_3 &\rightarrow B_1 \\ A_1 A_2 A_3 &\rightarrow B_2 \\ &\dots \end{aligned}$$

Shorthand $\longrightarrow$ $$A_1 A_2 A_3 \rightarrow B_1 B_2 B_3$$

Functional Dependencies (2/2)

- Example (figure out the right FDs)

Actor (Name, DOB, Address)

Movie (Title, Year, Language)

PlaysIn (Name, DOB, Title, Year)

Name DOB → Address
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)

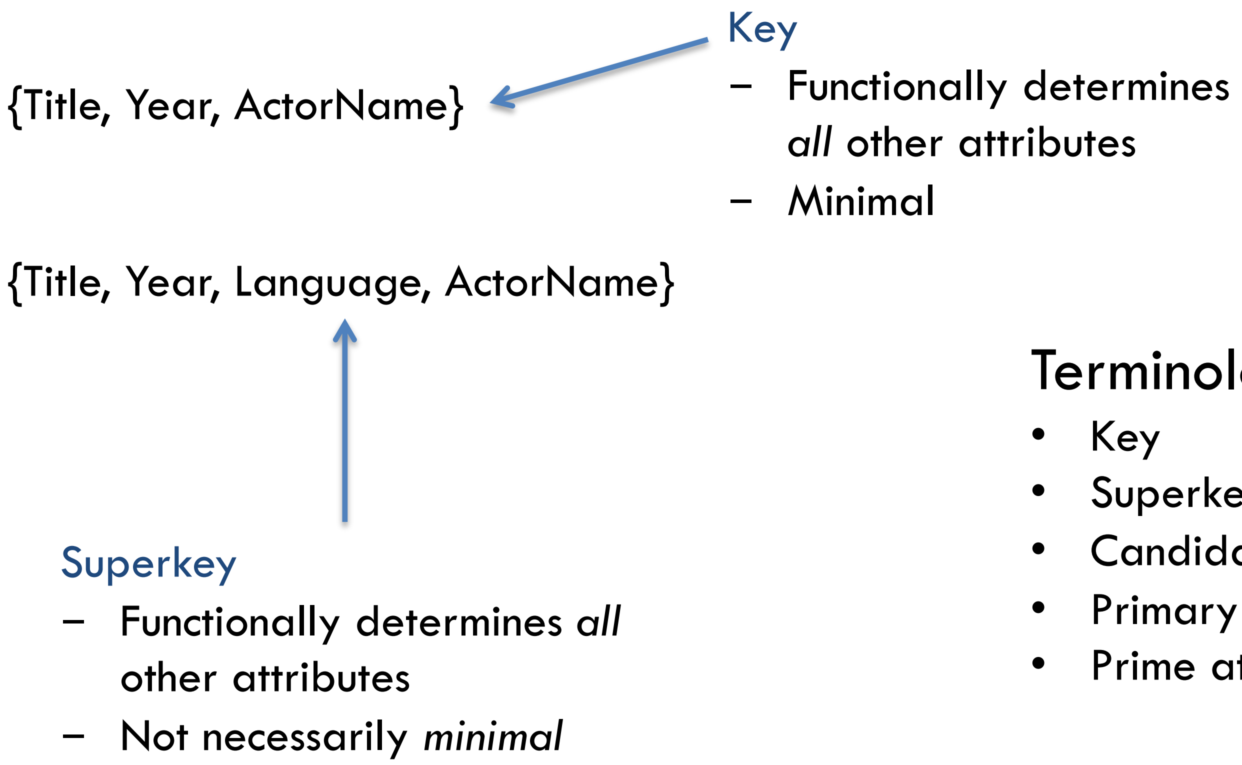
Movie (Title, Year, Language)

PlaysIn (Name, DOB, Title, Year)

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

Keys and superkeys

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



{Title, Year, ActorName}

Key

- Functionally determines *all* other attributes
- Minimal

{Title, Year, Language, ActorName}

Superkey

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

Terminology

- Key
- Superkey
- Candidate key
- 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

Rules involving FDs

- When does a set of FDs S follow from another set of FDs T?
- When are two sets of FDs S and T equivalent?

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

Closure of attributes (1/2)

- Let S be a set of FDs

  Is F, a new FD, derivable from S?

Example:

AB → C, BC → AD, D → E, CF → B

Does AB → D hold?

Closure of attributes (2/2)

AB → C, BC → AD, D → E, CF → B

AB → D ?

Compute $\{A,B\}^+$:

AB → C: {A,B,C}

BC → AD: {A,B,C,D}

D → E: {A,B,C,D,E}

Yes, AB → D

```
Z = Initial set of attributes
do {
  for each FD X → Y in S {
    if X subset of Z
       Add Y to Z
    fi
  }
  if Z unchanged, quit
}
```

Relation bt.
FD closure &
attribute closure?
↓
Why are they
useful?

NORMAL FORMS

Relation decomposition

- Breaking up a relation into two or more
- Tuples are *projected* accordingly
- Lossy and lossless decomposition
  - Can the original table be recovered from the decomposed tables?

X	Y	Z
1	2	3
2	3	4
2	4	5

Y	X
2	1
3	2
4	2

X	Z
1	3
2	4
2	5

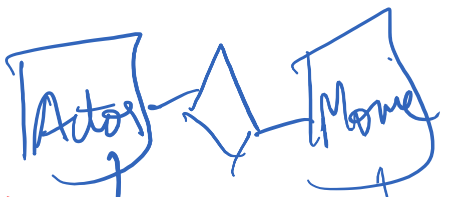
Y	X	Z
2	1	3
3	2	4
4	2	4
3	2	5
4	2	5

First normal form

- 1NF (First normal form)
  - A relation is in 1NF iff every tuple contains an atomic value for each attribute
  - Follows directly from definition of relation
  - Relation contains a key

Second normal form (1/2)

Universal relation



No non-prime attribute in the table is functionally dependent on a proper subset of any candidate key

Name DOB MTitle Year → Address

Name	DOB	Address	MTitle	Year	Language
Priyanka Chopra	1992	Mumbai	Don	2006	Hindi
Priyanka Chopra	1992	Mumbai	Don II	2011	Hindi
Tom Cruise	1962	LA	MI-IV	2011	English
Anthony Hopkins	1937	LA	Thor: Ragnarok	2017	English
Bill Nighy	1949	LA	Valkyrie	2008	English

What are we missing here?

Name DOB MTitle Year → Language

Name	DOB	Address
Priyanka Chopra	1992	Mumbai
Anthony Hopkins	1937	LA
Bill Nighy	1949	LA
Tom Cruise	1962	LA

MTitle	Year	Language
Don	2006	Hindi
Don II	2011	Hindi
MI-IV	2011	English
Valkyrie	2008	English
Thor: Ragnarok	2017	English

Second normal form (2/2)

Name	DOB	Address	MTitle	Year	Language
Priyanka Chopra	1992	Mumbai	Don	2006	Hindi
Priyanka Chopra	1992	Mumbai	Don II	2011	Hindi
Anthony Hopkins	1937	LA	MI-IV	2011	English
Anthony Hopkins	1937	LA	Valkyrie	2017	English
Bill Nighy	1949	LA	Valkyrie	2008	English

No non-prime attribute in the table is functionally dependent on a proper subset of any candidate key

Name | DoB | MTitle | Yes

ID	Name	DOB	Address
1	Priyanka Chopra	1992	Mumbai
2	Anthony Hopkins	1937	LA
3	Bill Nighy	1949	LA
4	Tom Cruise	1962	LA

playsIn

AID	MID
1	1
1	2
2	3
3	4
4	5
4	3

ID	MTitle	Year	Language
1	Don	2006	Hindi
2	Don II	2011	Hindi
3	MI-IV	2011	English
4	Valkyrie	2008	English
5	Thor: Ragnarok	2017	English

Third normal form

<u>Name</u>	<u>DOB</u>	Addr

<u>Addr</u>	Country

- 3NF (third normal form)
  - For a non-trivial FD X → Y,

    X is a superkey or *Y is prime*

Violates 3NF

Address → Country

Name, DOB → Address

Name, DOB → Country

<u>Name</u>	<u>DOB</u>	Address	Country
Priyanka Chopra	1992	Mumbai	India
Anthony Hopkins	1937	LA	USA
Bill Nighy	1949	LA	USA

<u>Name</u>	<u>DOB</u>	AID
Priyanka Chopra	1992	1
Anthony Hopkins	1937	2
Bill Nighy	1949	2

ID	Address	Country
1	Mumbai	India
2	LA	USA

Boyce-Codd normal form

BCNF

For a non-trivial FD X → Y,

X is a superkey

Addresses the following additional scenarios:

- Multiple candidate keys with intersecting elements
- All attributes are part of some key

Title	Theatre	City

Keys: Title, City

Theatre, Title

FDs:

Theatre → City

Title, City → Theatre

Theatre, Title → City

Violates BCNF, not 3NF

Lossless decomposition

A_1	A_2	A_3	A_4

- Algorithm:
  - If X → Y is a BCNF violation, then form two relations:

    with attributes from X U Y

    with attributes from X U (*all-X-Y*)

Multi-valued Dependencies

- What if no FDs hold?
- Redundancies can still occur
  - Clubbing together many-many relationships
- Homework
  - Find an example when this occurs
  - Study the definition of multi-valued dependency
  - Study 4NF