Data models (1/2)

- A model is an abstract representation
  - We care about data, so we have data models
- Do you know your data?
  - Is there something missing?
  - Are there redundancies?
  - How do various components of your data fit together?

Data models (2/2)

- To design a database, we first need a schema
  - Remember logical/physical independence ?
- Use a data model to design the schema
  - We will study two data models: ER and relational

ER MODEL

Entity-Relationship model (2/2)

- Entity sets
  - Actors, movies, studios
- Attributes
  - *Atomic values*
  - Name, age, height ← Actors
- Relationships
  - Between two entity sets
  - Actors *actsIn* Movies

Go back and revise from the text book. Almost all material is from there.

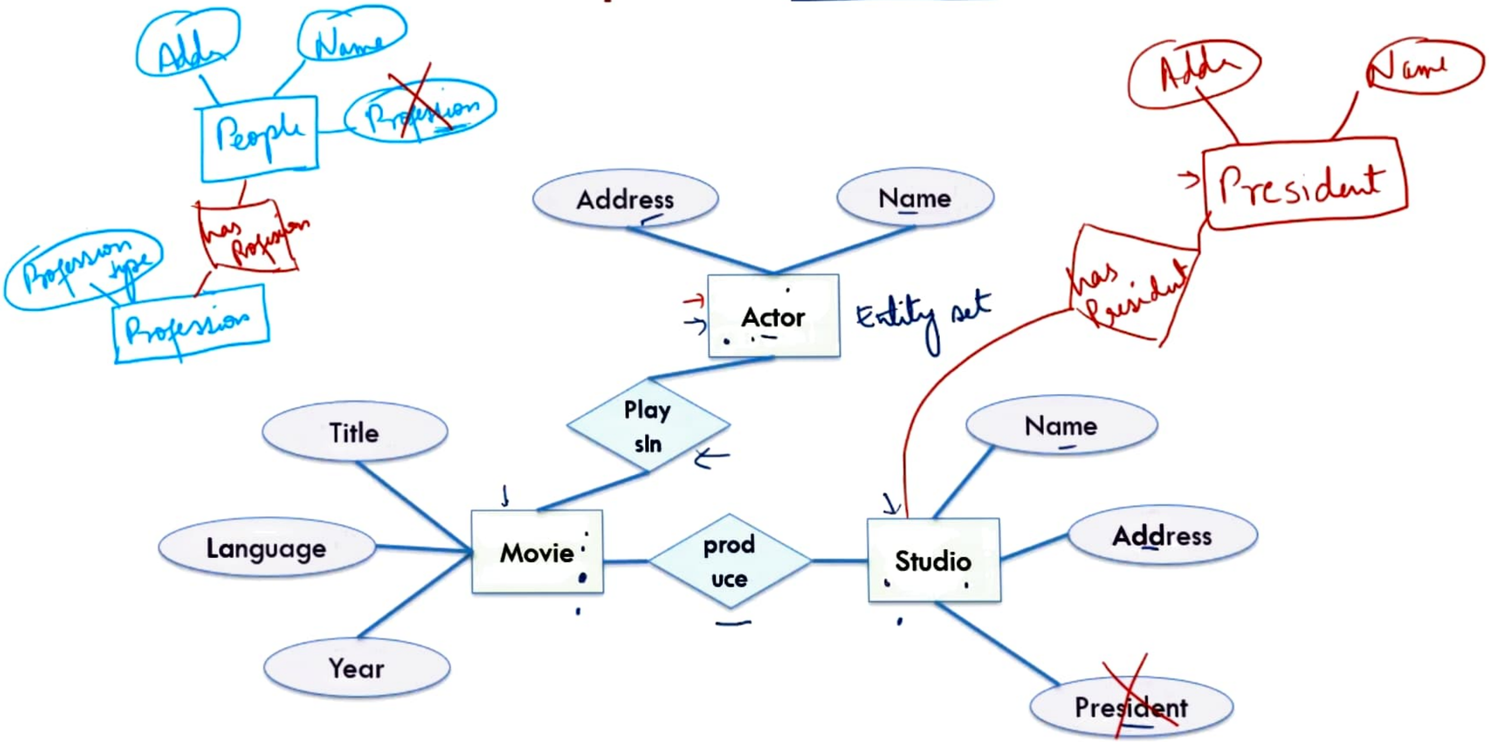
ER diagram

Entity set

Attribute

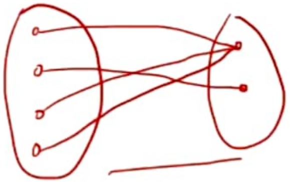
Relationship

Example – Movie DB



Types of relationships (1/2)

Movie produced by *at most* one studio, but a studio produces many movies



An actor plays at most one role, a role is played by at most one actor

Double roles

A movie has many actors, an actor plays in many movies

Types of relationships (2/2)

- one-one
  - an entity of one entity set is related to *at most* one entity of another entity set *and vice-versa*
- many-one
  - many entities of one entity set are related to *at most* one entity of another entity set
- many-many

Remember the maths

Entity sets

$$A = \{a_1, a_2, a_3\}$$

$$B = \{b_1, b_2\}$$

$$A \times B = \{(a_1, b_1), (a_1, b_2), (a_2, b_1), (a_2, b_2), (a_3, b_1), (a_3, b_2)\}$$

$R \subseteq A \times B$

relation

one-one

many-one

man

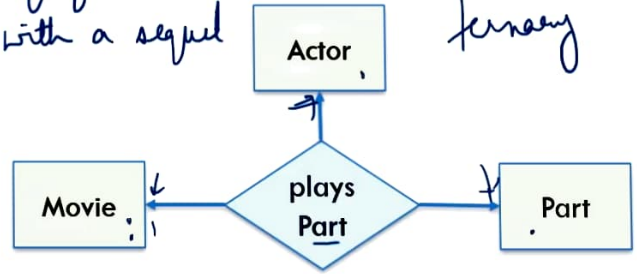
ER model: Relationships

Multi-way relationships

- Non-binary relationships

(1) Actor playing a double role
(2) Movie with a sequel

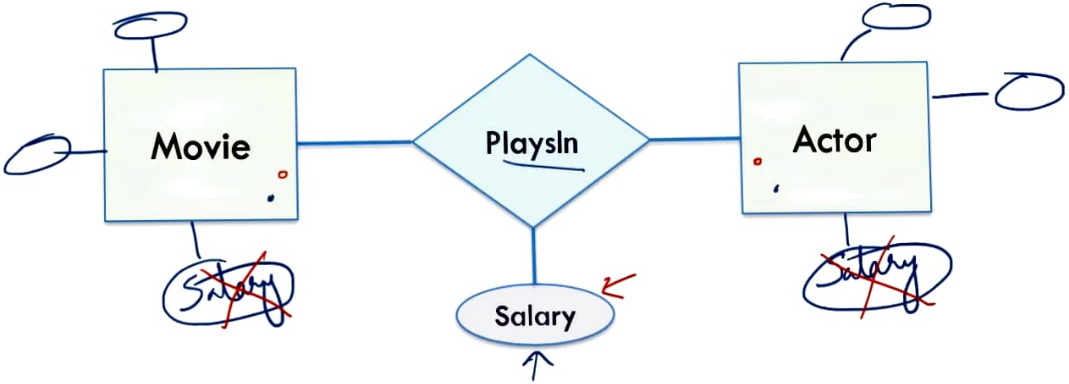
ternary



Priyanka Chopra	Baywatch	Victoria Leeds
Anthony Hopkins	Thor: Ragnarok	Odin

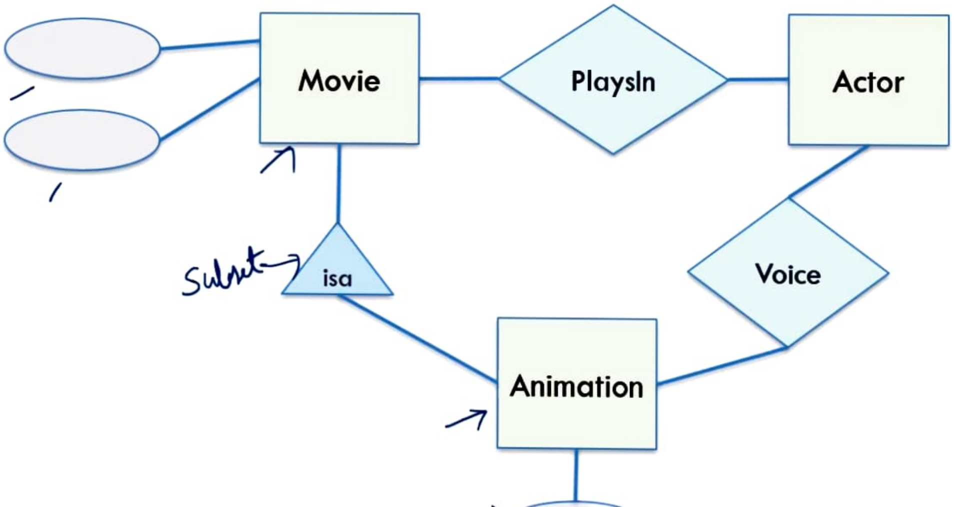
Attributes on relationships

- An attribute depends on a combination of entities, not a single entity
  - Relationships are how entities are combined!

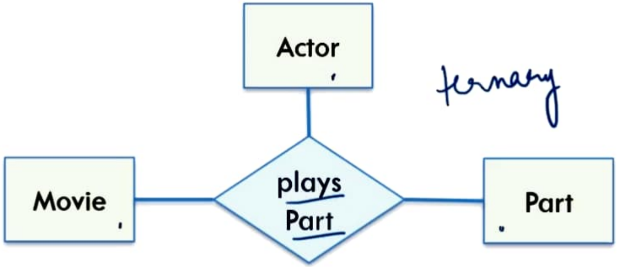


Modeling subsets

- ER model allows for hierarchies
  - Sound familiar?



N-ary to binary

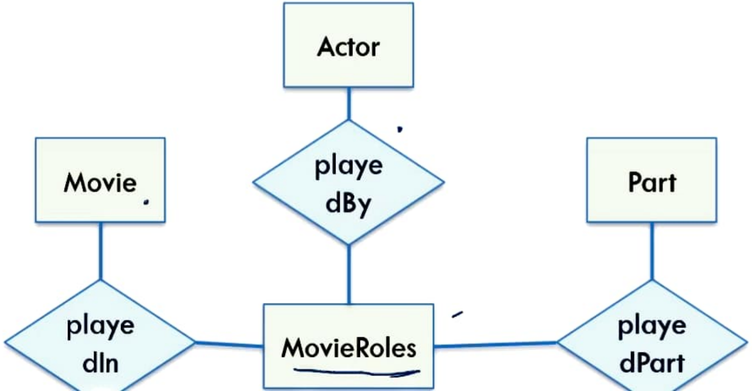


ternary

MRI
MRI
MRI

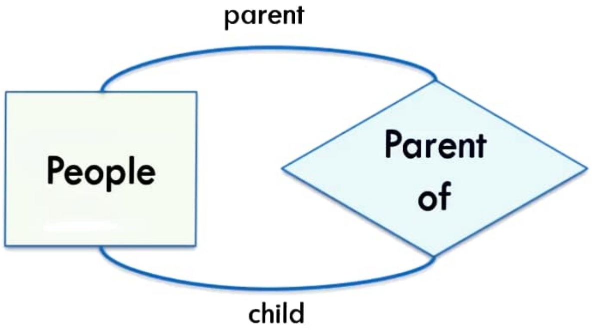
X- Movie
Y- Actor
Z- Part

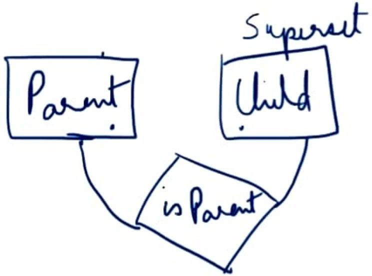
MRI	X	Y	Z



"Parallel" relationships

- A relationship from an entity set to itself
  - Each edge indicates a *role*





ER model

CONSTRAINTS

Modeling constraints

- Constraints model restrictions on the data
  - Data may be erroneous
  - Mistakes may be made during the data entry process 650 0, 100
- Modeling the correct constraints are part of the design process

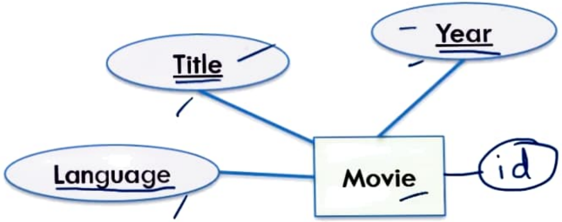
Common constraints (1/2)

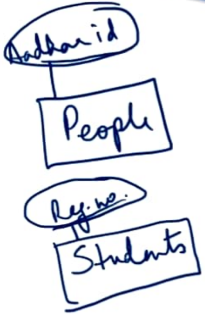
- Keys
  - How is an entity uniquely identified?
  - (Name, Year of birth) identifies an actor uniquely
- Single-value constraints
  - Unique values in a given context
  - Place of birth has to be unique
  - Null values

Profession

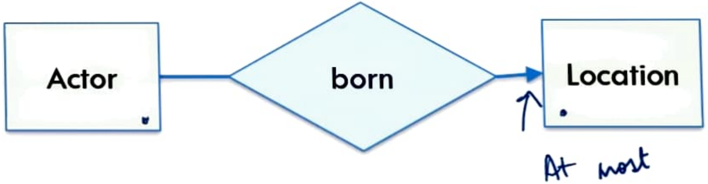
Key constraints and single value constraints

- *Every entity set* should have a key
- There could be more than one key ←





Keys



Common constraints (2/2)

- Referential integrity
  - Remove the null, insist on the value
  - If an actor acts in a movie, then that movie *has to* exist in the database
- Domain constraints
  - Restricting the value set of attributes
  - Age in range from 0 to 100 (or is it 0 to 25?)
- General constraints

Representing constraints (2/2)

Referential Integrity

Other

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

Attributes

Schema of the relation:

Actor-Movies (Name varchar(20), Movie varchar(50), Character varchar(20))

Name of relation/table

Actor-Movies instance of the relation

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

tuple

More about relations

$R \subseteq A \times B$

set   sets

- 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

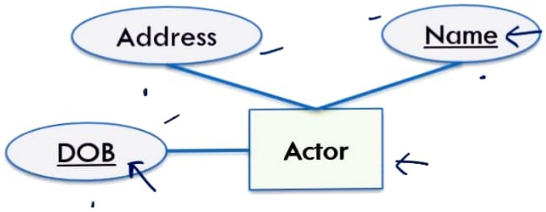
=

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

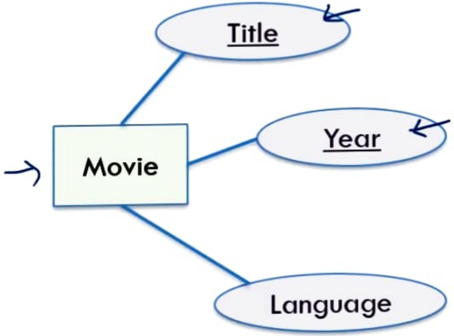
ER to relational

- 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

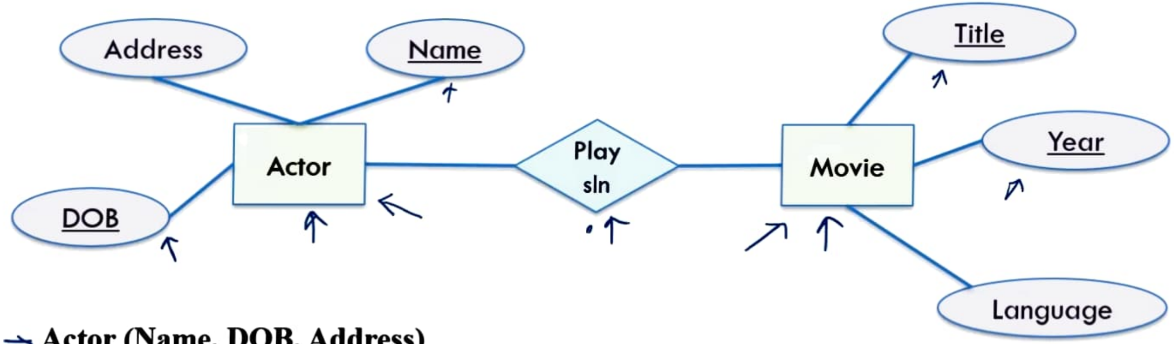


Actor (Name, DOB, Address)



Movie (Title, Year, Language)

Relationships to relations



Actor (<u>Name</u>, <u>DOB</u>, Address)

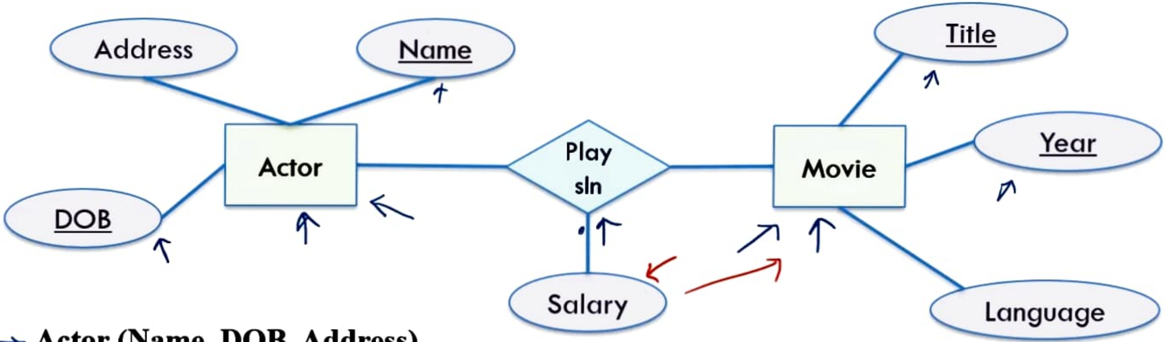
Movie (<u>Title</u>, <u>Year</u>, Language)

PlaysIn (Name, DOB, Title, Year)

Actor

Movie

Relationships to relations



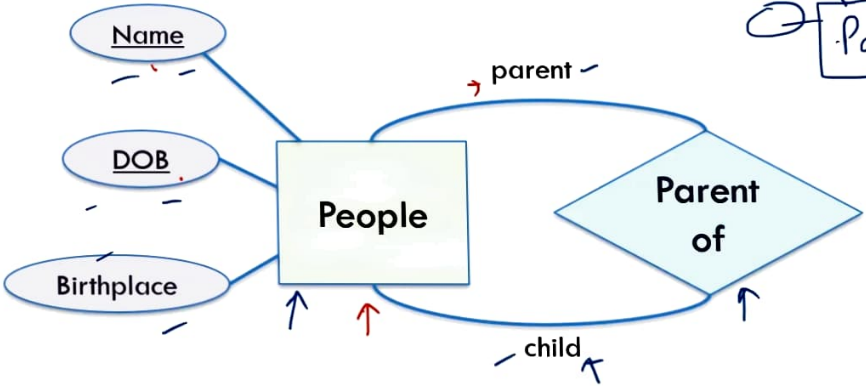
Actor (Name, DOB, Address)

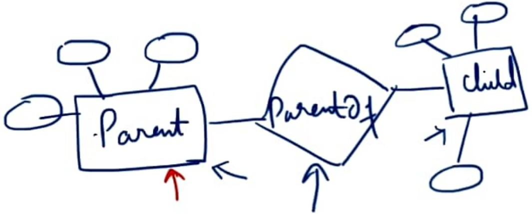
Movie (Title, Year, Language)

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

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

Roles to relations



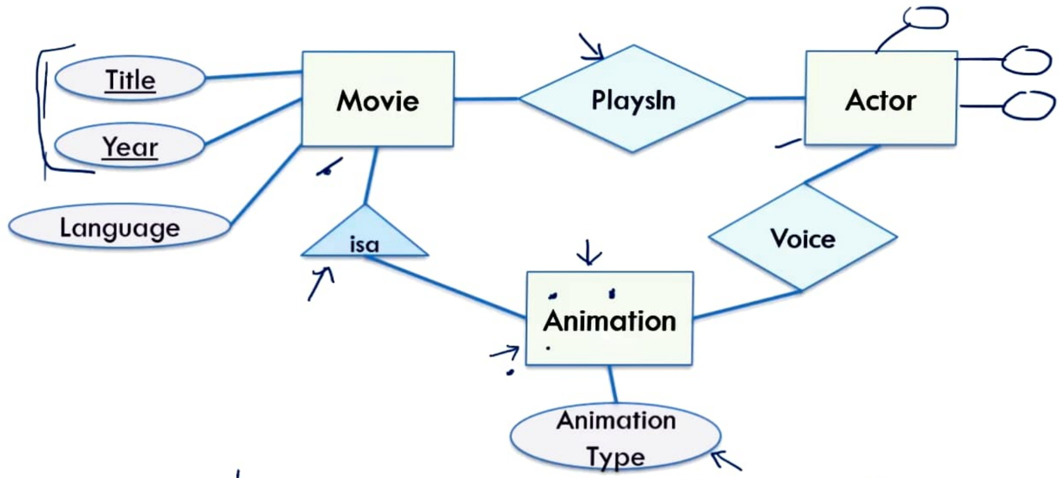


People (<u>Name</u>, <u>DOB</u>, Birthplace)

child

parentOf (parentName, parentDOB, childName, childDOB)

Hierarchies to relations



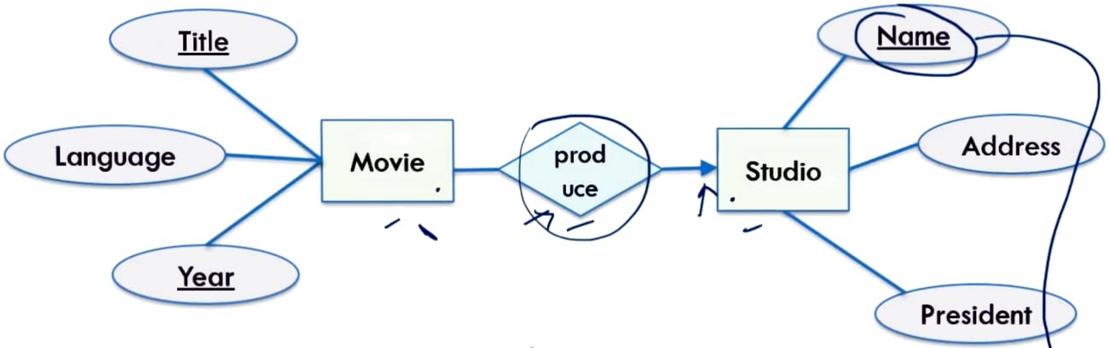
Movie (<u>Title</u>, <u>Year</u>, Language)

Actor (<u>Name</u>, <u>DOB</u>, City)

Animation (<u>Title</u>, <u>Year</u>, AnimationType)

AllMovies (<u>Title</u>, <u>Year</u>, Language, AnimationType)

Combining relations (1/2)



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

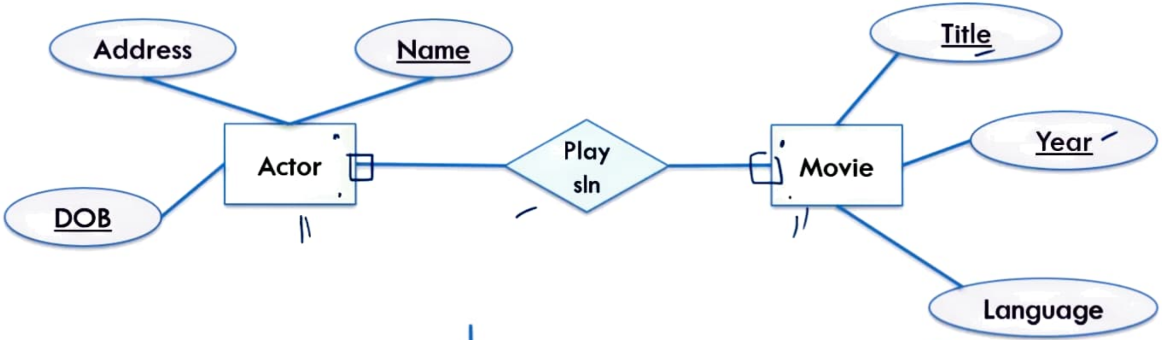
Studio (<u>Name</u>, Address, President)

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)

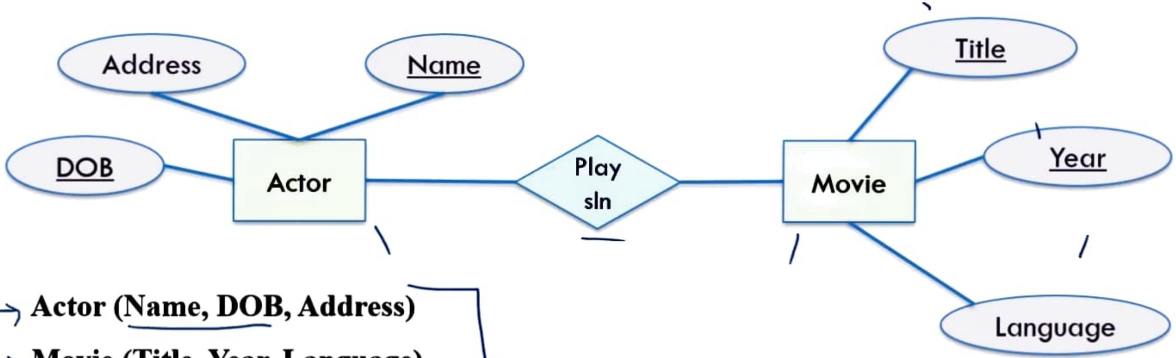
Movie (Title, Year, Language)

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)

"Universal Table"

Anomalies

Actor | Movie

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

Actor | Movie

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.

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

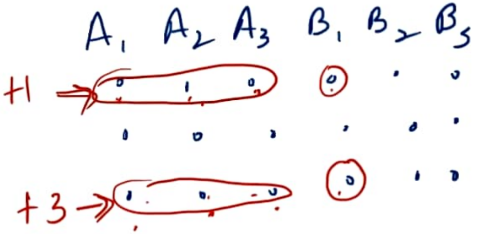
$$A_1A_2A_3 \xrightarrow{\text{functionally determines}} B_1$$

$$A_1A_2A_3 \rightarrow B_2$$

$$A_1\ A_2\ A_3 \rightarrow B_3$$

$$\ldots$$

Shorthand $\longrightarrow$ $A_1A_2A_3 \rightarrow B_1B_2B_3$



Functional Dependencies (2/2)

- Example (figure out the right FDs)

Universal table

Actor (Name, DOB, Address)

Movie (Title, Year, Language)

PlaysIn (Name, DOB, Title, Year)

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

Actor

Name DOB → Address

Title → Language

Movie

Language → Title Year

Name DOB → Title Year

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

Trivial and non-trivial FDs

Actor (Name, DOB, Address)

Movie (Title, Year, Language)

PlaysIn (Name, DOB, Title, Year)

Name DOB → Address ← Non-trivial FD

Name DOB → Name ← trivial FD

Name DOB → Name Address ← Non-trivial FD

Keys and superkeys

Movie (Title, Year, Language, Length)

T, Y → L, L

Key
- Functionally determines *all* other attributes
- Minimal

{Title, Year}

{Title, Year, Language}

{T, Y, L, L}

T,Y,L → L
T,Y,L → T
T,Y,L → Y

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

Terminology
- Key
- Superkey
- Candidate key — Keys
- Primary key — Only one candidate 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

  Trivial FD

  Closure of FDs

- 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:** {A, B, C, D, E}

$AB \rightarrow C$
$C \rightarrow ED$

Basis

Is this set S^+

$AB \rightarrow ED$
$AB \rightarrow E$
$ABC \rightarrow ED$
$C \rightarrow E$
$C \rightarrow D$

(1) Are they correctly derived?

(2) What is missing from this set

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

Lossy

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

Spurious tuple

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)

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

Universal relation

{Name DOB MTitle Year → Address ← Language

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

N, D → A

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

T, Y → L

MTitle	Year	Language
Don	2006	Hindi
Don II	2011	Hindi
MI-IV	2011	English
Valkyrie	2008	English

What are we missing here?

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

Lossless

ID	Name	DOB	Address
1	Priyanka Chopra	1992	Mumbai
2	Anthony Hopkins	1937	LA
3	Bill Nighy	1949	LA
4	[illegible]	[illegible]	[illegible]

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

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

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

    X is a superkey or *Y is prime*

key

$X \rightarrow Y$

Violation of 3NF ✓ $X A_1 A_2 A_3 \rightarrow Y$

✗ $A_1 \rightarrow Y$

✗

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

Address → Country
Name, DOB → Address
Name, DOB → Country

Addr → Country

Name	DOB	AID
Priyanka Chopra	1992	1
Anthony Hopkins	1937	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 [or Y is prime]

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

Violating BCNF

Lossless decomposition

- Algorithm:
  - If $X \rightarrow Y$ is a BCNF violation, then form two relations:

    with attributes from X U Y

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

$X \rightarrow Y$

$X \quad A_1 \quad A_2 \quad Y$

X	Y

X	A_1	A_2

X	Y	Z

X Y   X Z