Module 3: Normalization

- Now you know about:
  - What Entity Relationship Models are.
  - Converted Entity Relationship Diagrams to Relational Models
- This module will introduce:
  - Data redundancy problems.
  - Functional Dependencies
  - Normal Form, and Normalization Process

Learning Objectives

- Finishing this module, you will be able to:
  - Explain the problems associated with Data redundancy.
  - Explain and identify Functional Dependencies.
  - Explain different Normal Form,
  - Normalize relations to 3NF.

Data Redundancy

- What is data redundancy?
  - Major aim of relational database design is to group attributes into relations to minimize data redundancy.

EmpID	FirstName	LastName	DoB	Position	Department	StoreID	Address
#20399	John	Ford	1998/2/12	Manager	HR	#1506	1200 W Dillon Rd, Louisville
#30123	Anne	Brand	2001/3/12	Intern	Marketing	#1546	1600 29th Street, Boulder
#12524	David	Biden	2000/2/20	Assistant	Sales	#1524	1271 Sheridan Blvd, Broomfield
#14517	William	Potter	2001/9/12	Senior Manager	HR	#1506	1200 W Dillon Rd, Louisville
#15214	Mary	Alexander	2001/9/12	Assistant	IT	#1524	1271 Sheridan Blvd, Broomfield
#11032	Rose	Smith	1999/1/21	Intern	IT	#1503	10003 Grant Street, Thornton
#02012	Julie	Smith	1977/12/1	Senior Manager	IT	#1503	10003 Grant Street, Thornton
#78123	Angela	White	1967/4/4	Senior Manager	HR	#1546	1600 29th Street, Boulder
#21342	John	Ford	1983/11/11	Manager	IT	#1546	1600 29th Street, Boulder

Data Redundancy

- What are the problems?
  - Insert, Delete, Update

EmpID	FirstName	LastName	DoB	Position	Department	StoreID	Address
#20399	John	Ford	1998/2/12	Manager	HR	#1506	1200 W Dillon Rd, Louisville
#30123	Anne	Brand	2001/3/12	Intern	Marketing	#1546	1600 29th Street, Boulder
#12524	David	Biden	2000/2/20	Assistant	Sales	#1524	1271 Sheridan Blvd, Broomfield
#14517	William	Potter	2001/9/12	Senior Manager	HR	#1506	1200 W Dillon Rd, Louisville
#15214	Mary	Alexander	2001/9/12	Assistant	IT	#1524	1271 Sheridan Blvd, Broomfield
#11032	Rose	Smith	1999/1/21	Intern	IT	#1503	10003 Grant Street, Thornton
#02012	Julie	Smith	1977/12/1	Senior Manager	IT	#1503	10003 Grant Street, Thornton
#78123	Angela	White	1967/4/4	Senior Manager	HR	#1546	1600 29th Street, Boulder
#21342	John	Ford	1983/11/11	Manager	IT	#1546	1600 29th Street, Boulder

A Better Design

EmpID	FirstName	LastName	DoB	Position	Department	StoreID
#20399	John	Ford	1998/2/12	Manager	HR	#1506
#30123	Anne	Brand	2001/3/12	Intern	Marketing	#1546
#12524	David	Biden	2000/2/20	Assistant	Sales	#1524
#14517	William	Potter	2001/9/12	Senior Manager	HR	#1506
#15214	Mary	Alexander	2001/9/12	Assistant	IT	#1524
#11032	Rose	Smith	1999/1/21	Intern	IT	#1503
#02012	Julie	Smith	1977/12/1	Senior Manager	IT	#1503
#78123	Angela	White	1967/4/4	Senior Manager	HR	#1546
#21342	John	Ford	1983/11/11	Manager	IT	#1546

StoreID	Street	City	Zip
#1506	1200 W Dillon Rd	Louisville	80027
#1546	1600 29th Street	Boulder	80301
#1524	1271 Sheridan Blvd	Broomfield	80020
#1517	7125 W 88th Ave	Westminster	80021
#1548	16420 Washington Street	Thornton	80023
#1503	10003 Grant Street	Thornton	80229
#1502	5215 Wadsworth Blvd	Arvada	8002

Normalization

- Normalization is a technique for producing a set of suitable relations that support the data requirements of an enterprise.

What Is Suitable?

- Characteristics of suitable relations are:
  - the minimal number of attributes necessary to support the data requirements of the enterprise;
  - minimal redundancy with each attribute represented only once with the important exception of attributes that form all or part of foreign keys.
  - attributes with a close logical relationship are found in the same relation;

Benefit of Normalization

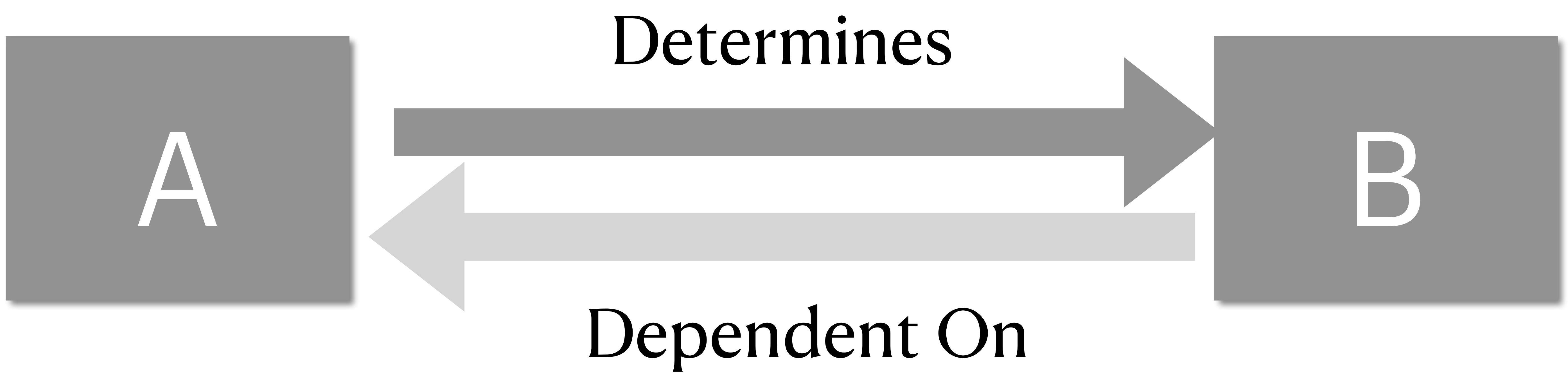
- Remove problems caused by data redundancy
- Easier for users to access and maintain data
- Take up minimal storage space

Functional Dependency

- In order to understand Normal Forms and Normalization Process, we need to understand the Functional Dependency first.
- Functional Dependencies describe the relationship among attributes in the same relation.

A Graphic View

- If A determines B, then B is functional dependent on A.
- A → B is a deterministic relationship, and is a functional dependency.



Example

- EmpID determines FirstName, so EmpID → FirstName
- Does FirstName determine EmpID?
- No. Thus, There is no ~~FirstName → EmpID~~

EmpID	FirstName	LastName	DoB	Position	Department	StoreID
#20399	John	Ford	1998/2/12	Manager	HR	#1506
#30123	Anne	Brand	2001/3/12	Intern	Marketing	#1546
#12524	David	Biden	2000/2/20	Assistant	Sales	#1524
#14517	William	Potter	2001/9/12	Senior Manager	HR	#1506
#15214	Mary	Alexander	2001/9/12	Assistant	IT	#1524
#11032	Rose	Smith	1999/1/21	Intern	IT	#1503
#02012	Julie	Smith	1977/12/1	Senior Manager	IT	#1503
#78123	Angela	White	1967/4/4	Senior Manager	HR	#1546
#21342	John	Ford	1983/11/11	Manager	IT	#1546

Example

EmpID	FirstName	LastName	DoB	Position	Department	StoreID
#20399	John	Ford	1998/2/12	Manager	HR	#1506
#30123	Anne	Brand	2001/3/12	Intern	Marketing	#1546
#12524	David	Biden	2000/2/20	Assistant	Sales	#1524
#14517	William	Potter	2001/9/12	Senior Manager	HR	#1506
#15214	Mary	Alexander	2001/9/12	Assistant	IT	#1524
#11032	Rose	Smith	1999/1/21	Intern	IT	#1503
#02012	Julie	Smith	1977/12/1	Senior Manager	IT	#1503
#78123	Angela	White	1967/4/4	Senior Manager	HR	#1546
#21342	John	Ford	1983/11/11	Manager	IT	#1546

- Find FDs in the relation:
  - EmpID → FirstName, LastName, DoB, Position, Department, StoreID
  - FirstName + LastName → Position?

FD Holds for All Time

- Examine sample data is helpful to reject Functional Dependencies.
- However, in order to establish a Functional Dependency, we need to consider All possible data.
- Even in the sample data, FirstName + LastName determine position, in real life, it is not true.
  - Thus, FirstName + LastName → Position is not true.

Practice

- Let's do more practice in Lab1.

Special Types of FDs

- We need to understand some special types of Functional Dependencies for normalization:
  - Full / Partial FDs
  - Transitive FDs.

Full / Partial FDs

- In one relation, if a set of attributes A: $(a_1, a_2, \ldots, a_n)$ determines attribute B:
  - A: $(a_1, a_2, \ldots, a_n) \rightarrow B$
  - If there is no proper subset of A also determines B, then A: $(a_1, a_2, \ldots, a_n) \rightarrow B$ is a Full functional dependency. It is full because we need all attributes in A to determine B.
  - If there is a proper subset of A determines B, then A: $(a_1, a_2, \ldots, a_n) \rightarrow B$ is a Partial functional dependency.
  - To achieve Full FD, we need to minimize A by removing unnecessary attributes.

Example

EmpID	FirstName	LastName	DoB	Position	Department	StoreID
#20399	John	Ford	1998/2/12	Manager	HR	#1506
#30123	Anne	Brand	2001/3/12	Intern	Marketing	#1546
#12524	David	Biden	2000/2/20	Assistant	Sales	#1524
#14517	William	Potter	2001/9/12	Senior Manager	HR	#1506
#15214	Mary	Alexander	2001/9/12	Assistant	IT	#1524
#11032	Rose	Smith	1999/1/21	Intern	IT	#1503
#02012	Julie	Smith	1977/12/1	Senior Manager	IT	#1503
#78123	Angela	White	1967/4/4	Senior Manager	HR	#1546
#21342	John	Ford	1983/11/11	Manager	IT	#1546

- EmpID + FirstName + LastName → DoB, is it a Full or Partial FD?
- EmpID + FirstName → DoB, is it a Full or Partial FD?
- EmpID → DoB, is it a Full or Partial FD?
  - Hint: If A → B, and A has only one attribute, then it must be full FD.

Example

- Relation R (<u>A</u>, <u>B</u>, <u>C</u>, D, E, F, G) has following FDs:
  - FD1: A, B, C → D, E, F, G
  - FD2: A → D
  - FD3: B, C → E
  - FD4: F → G
- FD1 is Full or Partial FD?
- FD3 is Full or Partial FD?

Transitive FDs

- In one relation, if an attribute A determines attribute B, and B determines attribute C (C is not A):
  - FD1: A → B
  - FD2: B → C
  - We can have A → C because the FD1 and FD2 form a Transitive Functional Dependency.

Example

- Relation R (A, B, C, D, E, F, G) has following FDs:
  - FD1: A, B, C → D, E, F, G
  - FD2: A → D
  - FD3: B, C → E
  - FD4: F → G
- Are there any Transitive Functional Dependencies?

Identifying FDs

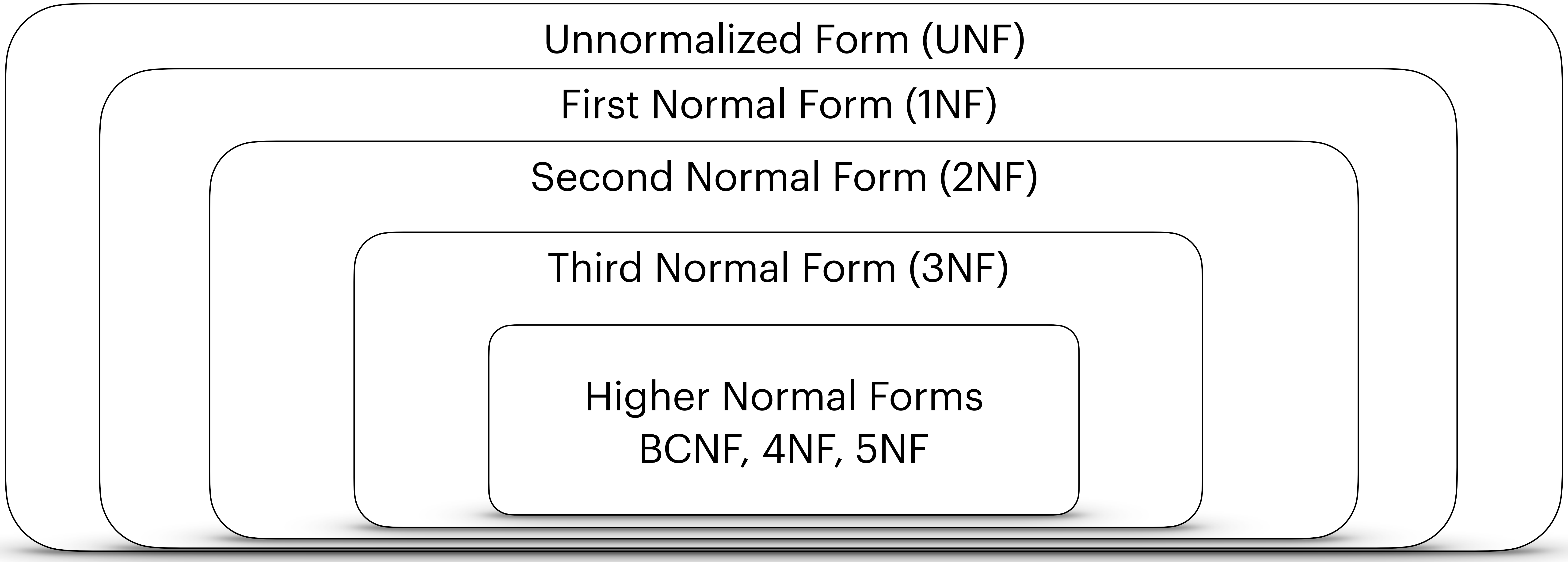
- It is simple if attributes and their relationships are well understood.
- The best case is all information is provided.
- Otherwise, use your common sense.

Practice

- Let's do more practice in Lab2.

Normalization Process

- Normalization Process is converting a relation from less restricted form to more restricted form.
- The level of restriction, is called Normal Form.



Unnormalized Form (UNF)

- A table that contains one or more repeating groups.

FirstName	LastName	DoB	Position	Department	StoreID
John	Ford	1998/2/12	Manager Vice President	HR	#1506 #1545
Anne	Brand	2001/3/12	Intern Assistant	Marketing	#1546 #1506
David	Biden	2000/2/20	Assistant	Sales	#1524
William	Potter	2001/9/12	Senior Manager	HR	#1506

First Normal Form (1NF)

- A relation in which the intersection of each row and column contains one and only one value.
- This is the requirement of a relation.
  - Each cell of relation contains exactly one value.
- Relation R (<u>A</u>, <u>B</u>, <u>C</u>, D, E, F, G) has following FDs:
  - FD1: A, B, C → D, E, F, G
  - FD2: A → D
  - FD3: B, C → E
  - FD4: F → G
- Relation R is in 1NF.

UNF to 1NF

- We need to eliminate the repeating values.

FirstName	LastName	DoB	Position	Department	StoreID
John	Ford	1998/2/12	Manager	HR	#1506
John	Ford	1998/2/12	Vice President	HR	#1545
Anne	Brand	2001/3/12	Intern	Marketing	#1546
Anne	Brand	2001/3/12	Assistant	Marketing	#1506
David	Biden	2000/2/20	Assistant	Sales	#1524
William	Potter	2001/9/12	Senior Manager	HR	#1506

Second Normal Form (2NF)

- A relation is in 2NF if:
  - It is in 1NF
  - Every non-primary-key attribute is fully functionally dependent on the primary key.
  - In other words, if there is any non-primary-key attribute is partially functionally dependent on the primary key, a relation is not in 2NF.

Example

- Relation R (<u>A</u>, <u>B</u>, <u>C</u>, D, E, F, G) has following FDs:
  - FD1: A, B, C → D, E, F, G
  - FD2: A → D
  - FD3: B, C → E
  - FD4: F → G
- Is R in 2NF?
- No.
  - Because FD2: A is a proper subset of primary key (A, B, C), and A → D. Thus, D is partially dependent on (A, B, C).
  - Also FD3: (B, C) is a proper subset of primary key (A, B, C), and B, C → E. Thus, E is partially dependent on (A, B, C).

1NF to 2NF

- We need to eliminate the partial FDs.
- Since in relation R, FD2, and FD3 lead to partial FDs, we need to move them out to new relations.
- Determinants will be primary keys in new relations; and will be foreign keys in original relations.

1NF to 2NF

- Relation R (<u>A</u>, <u>B</u>, <u>C</u>, D, E, F, G):
  - FD1: A, B, C → D, E, F, G
  - FD2: A → D
  - FD3: B, C → E
  - FD4: F → G
- Relation R (<u>A(fk)</u>, <u>B(fk)</u>, <u>C(fk)</u>, ~~D~~, ~~E~~, F, G):
  - FD1: A, B, C → ~~D, E,~~ F, G
  - ~~FD2: A → D~~
  - ~~FD3: B, C → E~~
  - FD2: F → G
- Relation R1 (<u>A</u>, D):
  - FD1: A → D
- Relation R2 (<u>B</u>, <u>C</u>, E):
  - FD1: B, C → E

Third Normal Form (3NF)

- A relation is in 3NF if:
  - It is in 2NF
  - No non-primary-key attribute is transitively dependent on the primary key.
  - In other words, if there is any non-primary-key attribute is transitively dependent on the primary key, a relation is not in 3NF.

Example

- Relation R (<u>A</u>, <u>B</u>, <u>C</u>, F, G) has following FDs:
  - FD1: A, B, C → F, G
  - FD2: F → G
- Is R in 3NF?
- No.
  - Because FD2: G is a non-primary-key attribute, A, B, C → F, and F → G. Thus, G is is transitively dependent on (A, B, C).

2NF to 3NF

- We need to eliminate the transitive FDs.
- Since in relation R, FD2 lead to transitive FDs, we need to move them out to new relations.
- Determinants will be primary keys in new relations; and will be foreign keys in original relations.

2NF to 3NF

- Relation R (<u>A</u>, <u>B</u>, <u>C</u>, F, G):
  - FD1: A, B, C → F, G
  - FD2: F → G

- Relation R (<u>A(fk)</u>, <u>B(fk)</u>, <u>C(fk)</u>, F(fk), ~~G~~):
  - FD1: A, B, C → F, ~~G~~
  - ~~FD2: F → G~~

- Relation R3 (<u>F</u>, G):
  - FD1: F → G

Practice

- Let's do more practice in Lab3.

Assignment

- Let's do the assignment to assess your understanding of Functional Dependencies, Normal Forms, and Normalization Process.

Congratulations

- Now you finished Module 3!
- You should be comfortable to lay out the Entity Relationship Model, convert it to a Relational Model, and normalize it to 3NF.
- Now you will be able to implement it as a database!
- Next module, we are going to do a case study.
- See you soon!