

# **DATABASE NORMALIZATION**



**Department of Computer Sciences  
Birzeit University  
Dr. Ahmad Abusnaina**

# INTRODUCTION

- Normalization is a process that “improves” a database design by generating relations that are of higher normal forms.
- **Normalization** is the process of removing redundant data from your tables to improve storage efficiency, data integrity, and scalability.
-

# DATABASE NORMALIZATION

- In the relational model, there are methods exist for quantifying how efficient a database is. These classifications are called normal forms (or NF).
- Normalization generally involves splitting existing tables into multiple ones, which must be re-joined or linked each time a query is issued.

# DATABASE NORMALIZATION

- The main goal of Database Normalization is to restruct the logical data model of a database to:
  - Eliminate redundancy
  - Organize data efficiently
  - Reduce the potential for data anomalies.

# DATA ANOMALIES

- Data anomalies are inconsistencies in the data stored in a database as a result of an operation such as update, insertion, and/or deletion.
- Such inconsistencies may arise when have a particular record stored in multiple locations and not all of the copies are updated.
- We can prevent such anomalies by implementing 7 different level of normalization called Normal Forms (NF)

# FUNCTIONAL DEPENDENCIES

- We say an attribute, **B**, has a functional dependency on another attribute, **A**;
- If for any two records, which have the same value for **A**, then the values for **B** in these two records must be the same.

We illustrate this as:

$A \rightarrow B$

**A determines B**

**B depends on A**

**Example:** Suppose we keep track of employee email addresses, and we only track one email address for each employee. Suppose each employee is identified by their unique employee number. We say there is a functional dependency of email address on employee number:

employee number  $\rightarrow$  email address

# FUNCTIONAL DEPENDENCIES

<u>EmpNum</u>	EmpEmail	EmpFname	EmpLname
123	jdoe@abc.com	John	Doe
456	psmith@abc.com	Peter	Smith
555	alee1@abc.com	Alan	Lee
633	pdoe@abc.com	Peter	Doe
787	alee2@abc.com	Alan	Lee

If EmpNum is the PK then the FDs:

EmpNum  $\rightarrow$  EmpEmail

EmpNum  $\rightarrow$  EmpFname

EmpNum  $\rightarrow$  EmpLname

must exist.

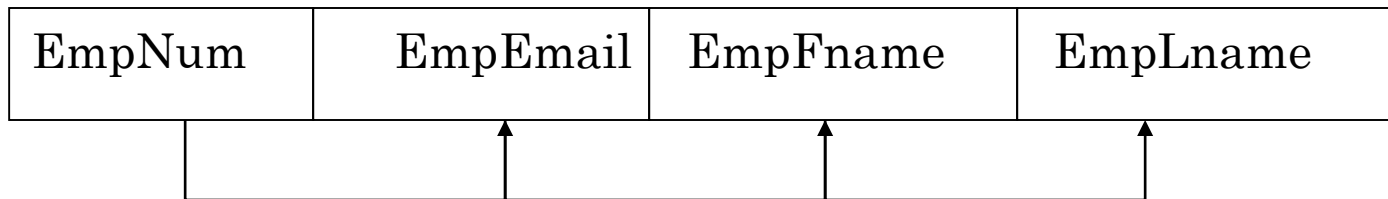
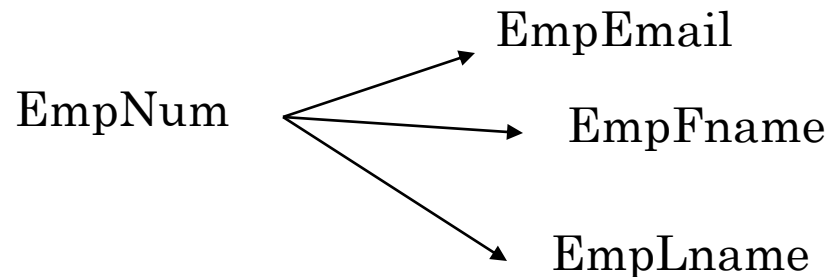
# FUNCTIONAL DEPENDENCIES

$\text{EmpNum} \rightarrow \text{EmpEmail}$

$\text{EmpNum} \rightarrow \text{EmpFname}$

$\text{EmpNum} \rightarrow \text{EmpLname}$

*3 different ways you might see FDs depicted*





# DETERMINANT

Functional Dependency

$\text{EmpNum} \rightarrow \text{EmpEmail}$

Attribute on the LHS is known as the *determinant*

- EmpNum is a determinant of EmpEmail

# TRANSITIVE DEPENDENCY

Consider attributes A, B, and C, and where

$$A \rightarrow B \text{ and } B \rightarrow C.$$

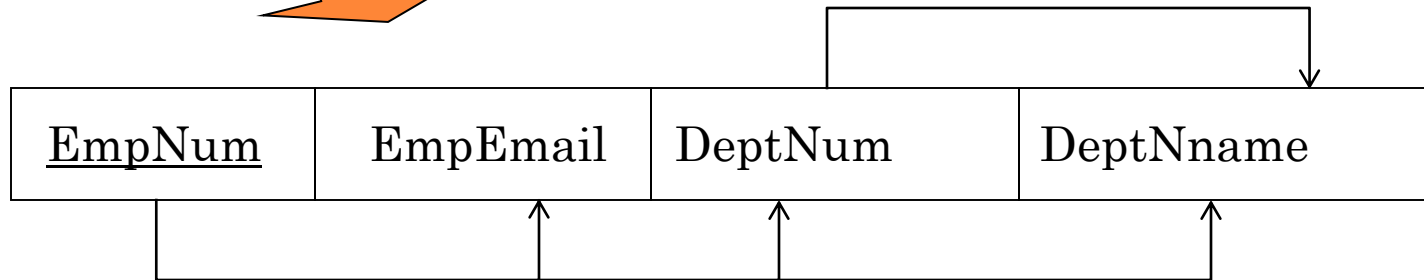
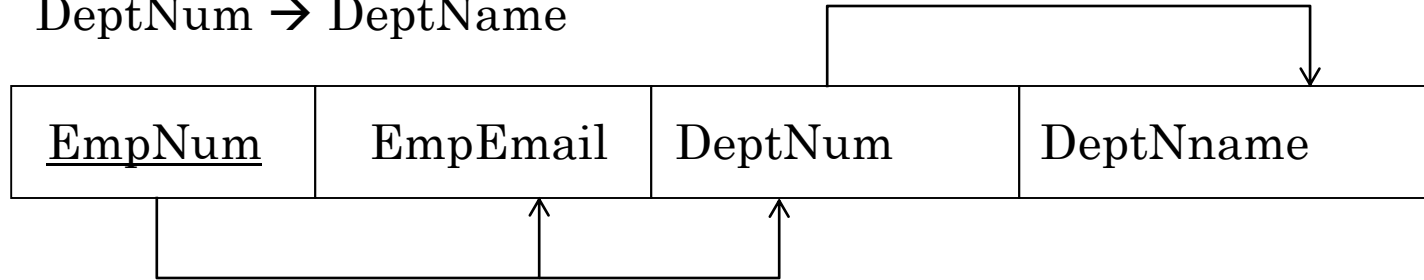
Functional dependencies are transitive, which means that we also have the functional dependency  $A \rightarrow C$

We say that C is transitively dependent on A through B.

# TRANSITIVE DEPENDENCY

$\text{EmpNum} \rightarrow \text{DeptNum}$

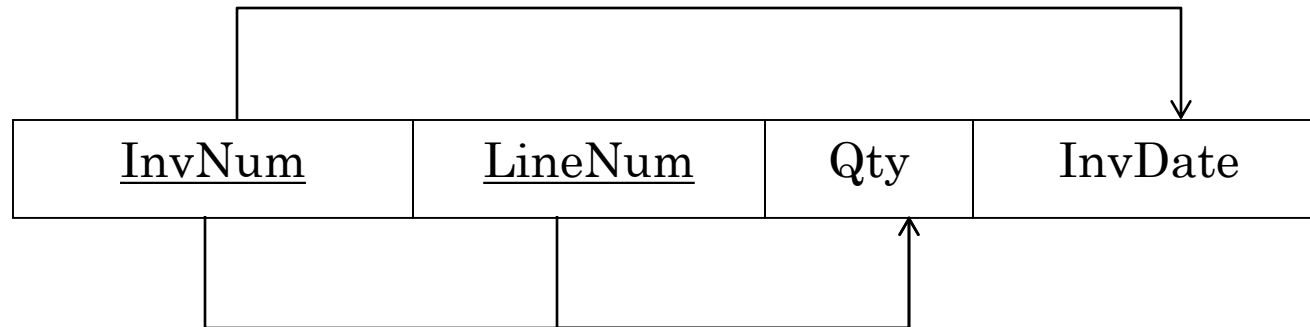
$\text{DeptNum} \rightarrow \text{DeptName}$



DeptName is *transitively dependent* on EmpNum via DeptNum  
 $\text{EmpNum} \rightarrow \text{DeptName}$

# PARTIAL DEPENDENCY

A **partial dependency** exists when an attribute B is functionally dependent on an attribute A, and A is a component of a multipart candidate key.



Candidate keys: {InvNum, LineNum}

InvDate is *partially dependent* on {InvNum, LineNum} as

**InvNum is a determinant of InvDate and InvNum is part of a candidate key**

# NORMALIZATION

There is a sequence to normal forms:

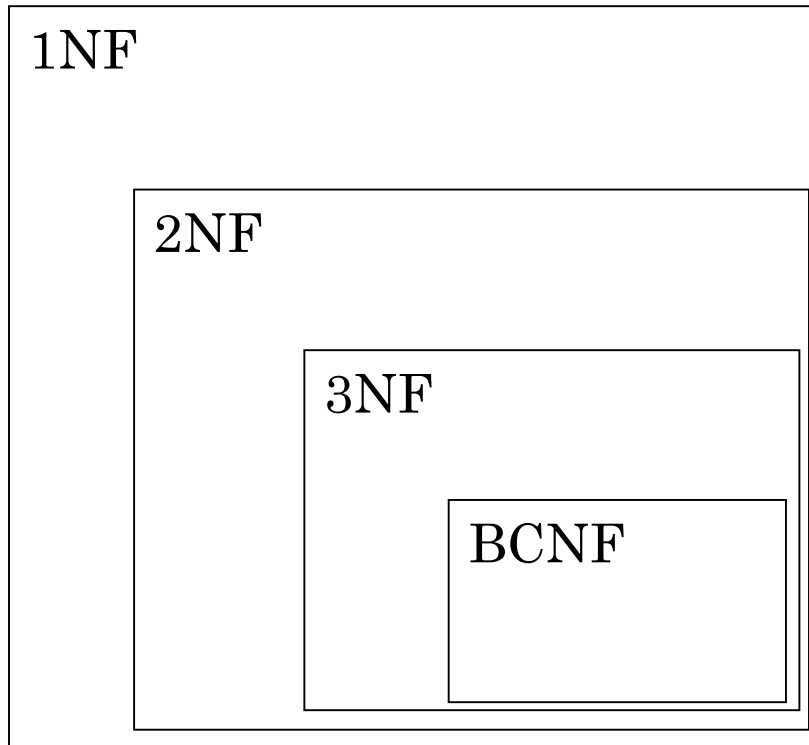
1NF is considered the weakest,  
2NF is stronger than 1NF,  
3NF is stronger than 2NF, and  
BCNF is considered the strongest

Also,

any relation that is in BCNF, is in 3NF;  
any relation in 3NF is in 2NF; and  
any relation in 2NF is in 1NF.

One of the key requirements to remember is that Normal Forms are progressive. That is, in order to have 3<sup>rd</sup> NF we must have 2<sup>nd</sup> NF and in order to have 2<sup>nd</sup> NF we must have 1<sup>st</sup> NF.

# NORMALIZATION



*a relation in BCNF, is also in 3NF*

*a relation in 3NF is also in 2NF*

*a relation in 2NF is also in 1NF*

# NORMAL FORMS:

- Unnormalized – There are multivalued attributes or repeating groups
- 1 NF – No multivalued attributes or repeating groups.
- 2 NF – 1 NF plus no partial dependencies
- 3 NF – 2 NF plus no transitive dependencies

# 1ST NORMAL FORM

## THE REQUIREMENTS

- The requirements to satisfy the 1<sup>st</sup> NF:
  - Each table has a primary key: minimal set of attributes which can uniquely identify a record.
  - The values in each column of a table are atomic (No multi-value attributes allowed).
  - There are no repeating groups: two columns do not store similar information in the same table.
  - Redundant data across multiple rows of a table must be moved to a separate table.
    - The resulting tables must be related to each other by use of foreign key.



# FIRST NORMAL FORM

The following is **not** in 1NF

<u>EmpNum</u>	EmpPhone	EmpDegrees
123	233-9876	
333	233-1231	BA, BSc, PhD
679	233-1231	BSc, MSc

EmpDegrees is a multi-valued field:

employee 679 has two degrees: *BSc* and *MSc*

employee 333 has three degrees: *BA*, *BSc*, *PhD*

# FIRST NORMAL FORM

<u>EmpNum</u>	EmpPhone	EmpDegrees
123	233-9876	
333	233-1231	BA, BSc, PhD
679	233-1231	BSc, MSc

To obtain 1NF relations we must, without loss of information, replace the above with two relations.

# FIRST NORMAL FORM

## Employee

<u>EmpNum</u>	EmpPhone
123	233-9876
333	233-1231
679	233-1231

## EmployeeDegree

<u>EmpNum</u>	<u>EmpDegree</u>
333	BA
333	BSc
333	PhD
679	BSc
679	MSc

# SECOND NORMAL FORM

A relation is in **2NF** if it is in 1NF, and every non-key attribute is fully dependent on each candidate key.

That is, we **don't have any partial functional dependency**.

- 2NF (and 3NF) both involve the concepts of key and non-key attributes.
- A key attribute is any attribute that is part of a key; any attribute that is not a key attribute, is a non-key attribute.
- A relation in 2NF will not have any partial dependencies

# SECOND NORMAL FORM

Consider this **InvLine** table (in 1NF):

There are two candidate keys: InvNum, LineNum

InvNum, LineNum  $\longrightarrow$  ProdNum, Qty

InvNum  $\longrightarrow$  InvDate

InvLine is only in 1NF

<u>InvNum</u>	<u>LineNum</u>	ProdNum	Qty	InvDate
---------------	----------------	---------	-----	---------

InvLine is **not in 2NF** since there is a partial dependency of InvDate on InvNum

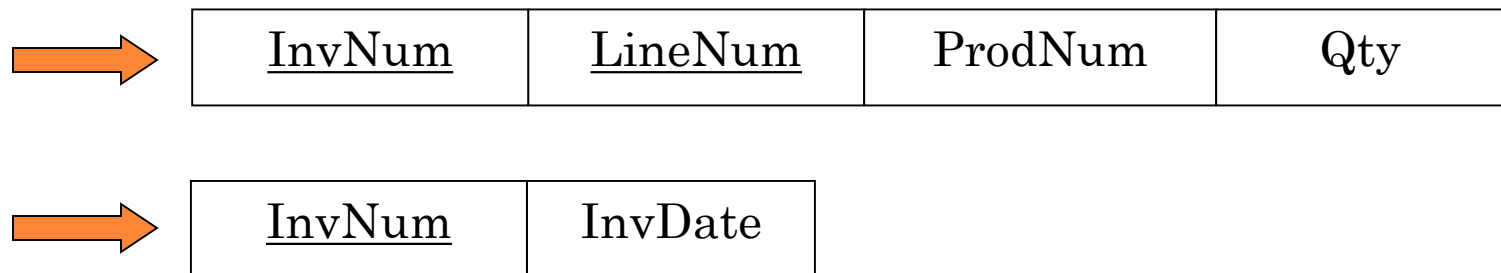
# SECOND NORMAL FORM

## InvLine

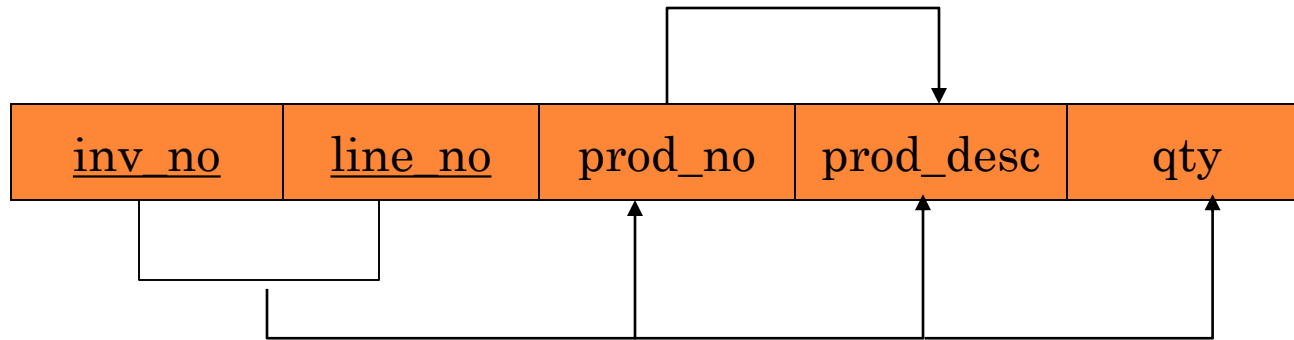
<u>InvNum</u>	<u>LineNum</u>	ProdNum	Qty	InvDate
---------------	----------------	---------	-----	---------

The above relation has redundancies: the invoice date is repeated on each invoice line.

We can *improve* the database by decomposing the relation into two relations:

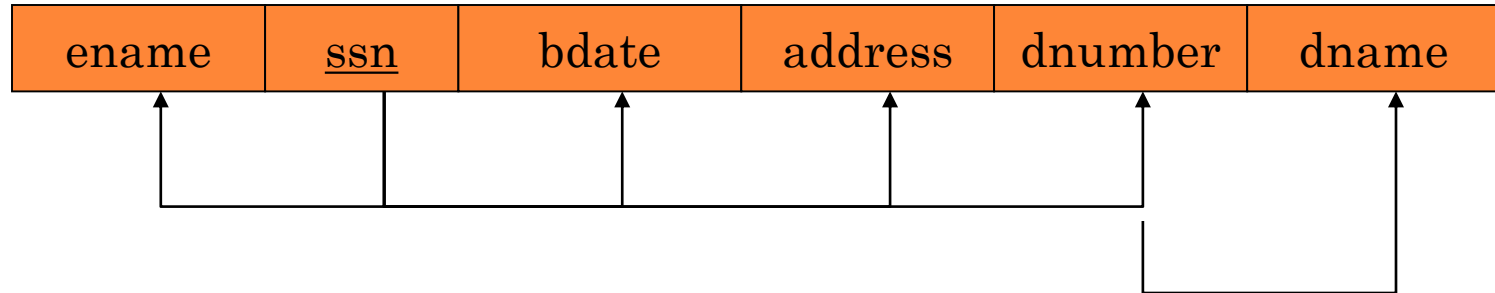


Is the following relation in 2NF? **Yes**



Answer: yes in 2NF, but not in 3NF, nor in BCNF:

### EmployeeDept



since `dnumber` is not a candidate key and we have:

`ssn` → `dnumber`

`dnumber` → `dname`.



# 3<sup>RD</sup> NORMAL FORM

## THE REQUIREMENTS

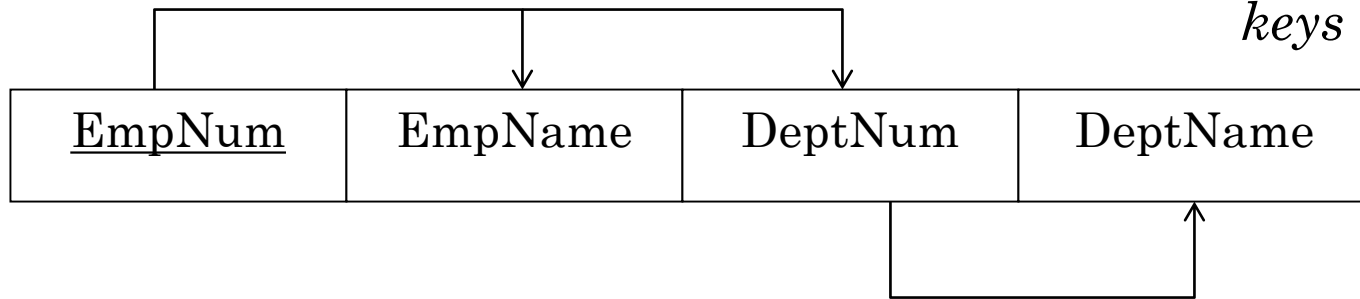
- The requirements to satisfy the 3<sup>rd</sup> NF:
  - All requirements for 2<sup>nd</sup> NF must be met.
  - Eliminate transitive dependency
    - That is, any field that is dependent not only on the primary key but also on another field must be moved to another table.

# THIRD NORMAL FORM

- A relation is in 3NF if the relation is in 2NF and all determinants of non-key attributes are candidate keys.
- That is, for any functional dependency:  $X \rightarrow Y$ , where  $Y$  is a non-key attribute (or a set of non-key attributes),  $X$  is a candidate key.
- A relation in 3NF will not have any transitive dependencies of non-key attribute on a candidate key through another non-key attribute.

# THIRD NORMAL FORM

Consider this **Employee** relation



*Candidate  
keys are? ...*

EmpName, DeptNum, and DeptName are non-key attributes.

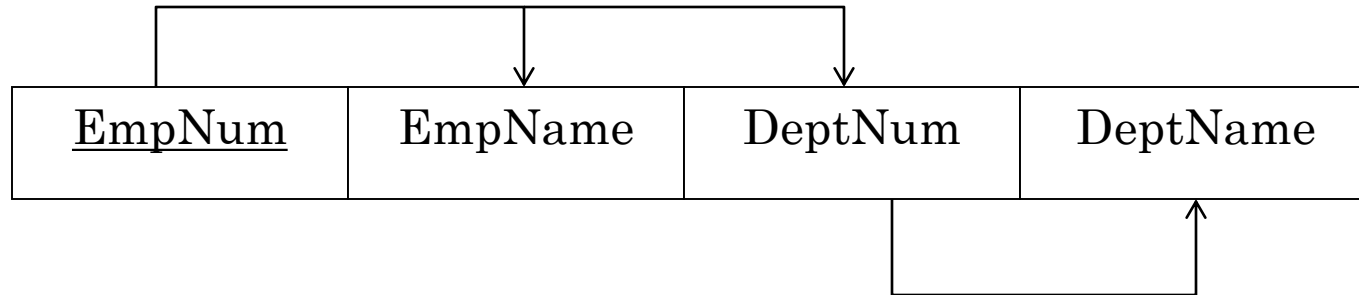
DeptNum determines DeptName, a non-key attribute, and DeptNum is not a candidate key.

Is the relation in BCNF? ... no

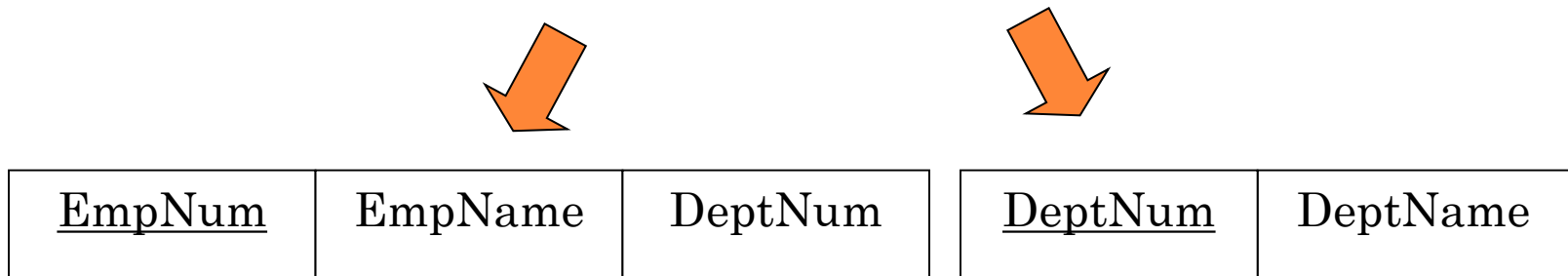
Is the relation in 3NF? ... no

Is the relation in 2NF? ... yes, no partial dependency

# THIRD NORMAL FORM



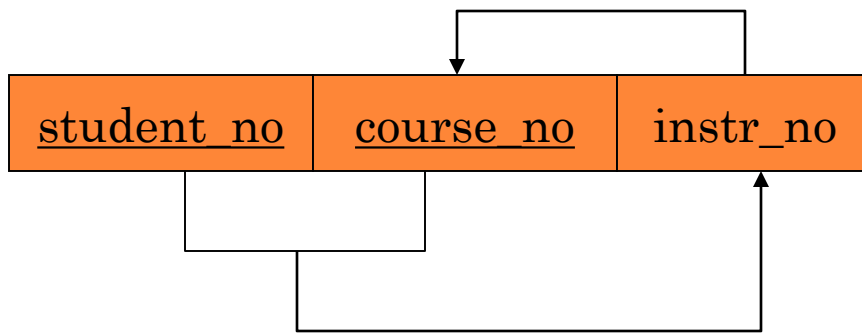
We correct the situation by decomposing the original relation into two 3NF relations.



Verify these two relations are in 3NF.

**No transitive dependency**

**In 3NF, but not in BCNF:**



*Instructor teaches one course only.*

*Student takes a course and has one instructor.*

$\{\text{student\_no}, \text{course\_no}\} \rightarrow \text{instr\_no}$   
 $\text{instr\_no} \rightarrow \text{course\_no}$

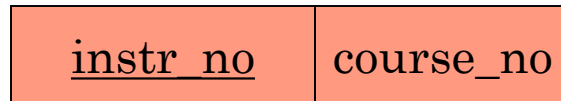
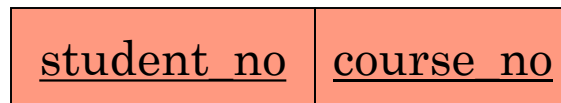
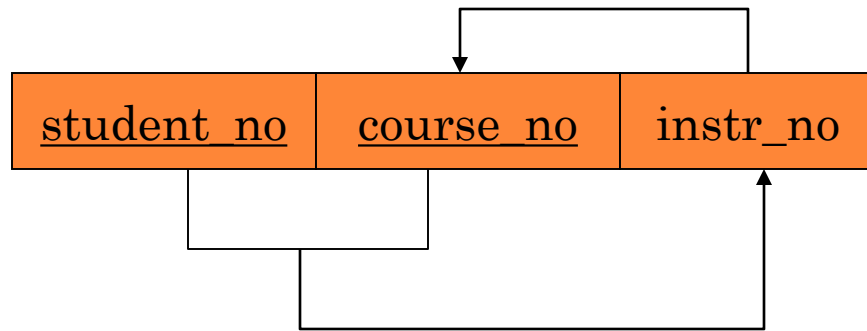
since we have  $\text{instr\_no} \rightarrow \text{course\_no}$ , but  $\text{instr\_no}$  is not a Candidate key.

# BOYCE-CODD NORMAL FORM

BCNF is defined very simply:

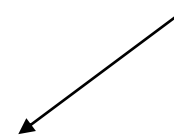
a relation is in BCNF if it is in 3NF and if every determinant is a candidate key.

If our database will be used for OLTP (on line transaction processing), then BCNF is our target. Usually, we meet this objective. However, we might denormalize (3NF, 2NF, or 1NF) for performance reasons.



$\text{instr\_no} \rightarrow \text{course\_no}$

BCNF



# MORE EXAMPLES

Un-normalized Students table:

<u>Student#</u>	AdvID	AdvName	AdvRoom	Class
123	123A	James	555	102-8, 104-9
124	123B	Smith	467	209-0, 102-8

Normalized Students table:

<u>Student#</u>	AdvID	AdvName	AdvRoom	<u>Class#</u>
123	123A	James	555	102-8
123	123A	James	555	104-9
124	123B	Smith	467	209-0
124	123B	Smith	467	102-8





# 1<sup>ST</sup> NORMAL FORM EXAMPLE

Students table

<u>Student#</u>	AdvID	AdvName	AdvRoom
123	123A	James	555
124	123B	Smith	467

Registration table

<u>Student#</u>	<u>Class#</u>
123	102-8
123	104-9
124	209-0
124	102-8

Student# → Class#

Student# → AdvID

AdvID → {AdvName, AdvRoom}

Is this in 2NF?


Yes, why?

No partial dependency



# 3<sup>RD</sup> NORMAL FORM EXAMPLE

Students table:



<u>Student#</u>	AdvID	AdvName	AdvRoom
123	123A	James	555
124	123B	Smith	467

Is this in 3NF?

No, why?

transitive dependency



# 3<sup>RD</sup> NORMAL FORM EXAMPLE CONT.

Students table:

<u>Student#</u>	AdvID
123	123A
124	123B

Registration table:

<u>Student#</u>	<u>Class#</u>
123	102-8
123	104-9
124	209-0
124	102-8

Advisor table:

<u>AdvID</u>	AdvName	AdvRoom
123A	James	555
123B	Smith	467



# NORMAL FORMS: REVIEW

- Unnormalized – There are multivalued attributes or repeating groups
- 1 NF – No multivalued attributes or repeating groups.
- 2 NF – 1 NF plus no partial dependencies
- 3 NF – 2 NF plus no transitive dependencies

# DEPENDENCIES: DEFINITIONS

- *Partial Dependency* – when a non-key attribute is determined by a part, but not the whole, of a **COMPOSITE** primary key.

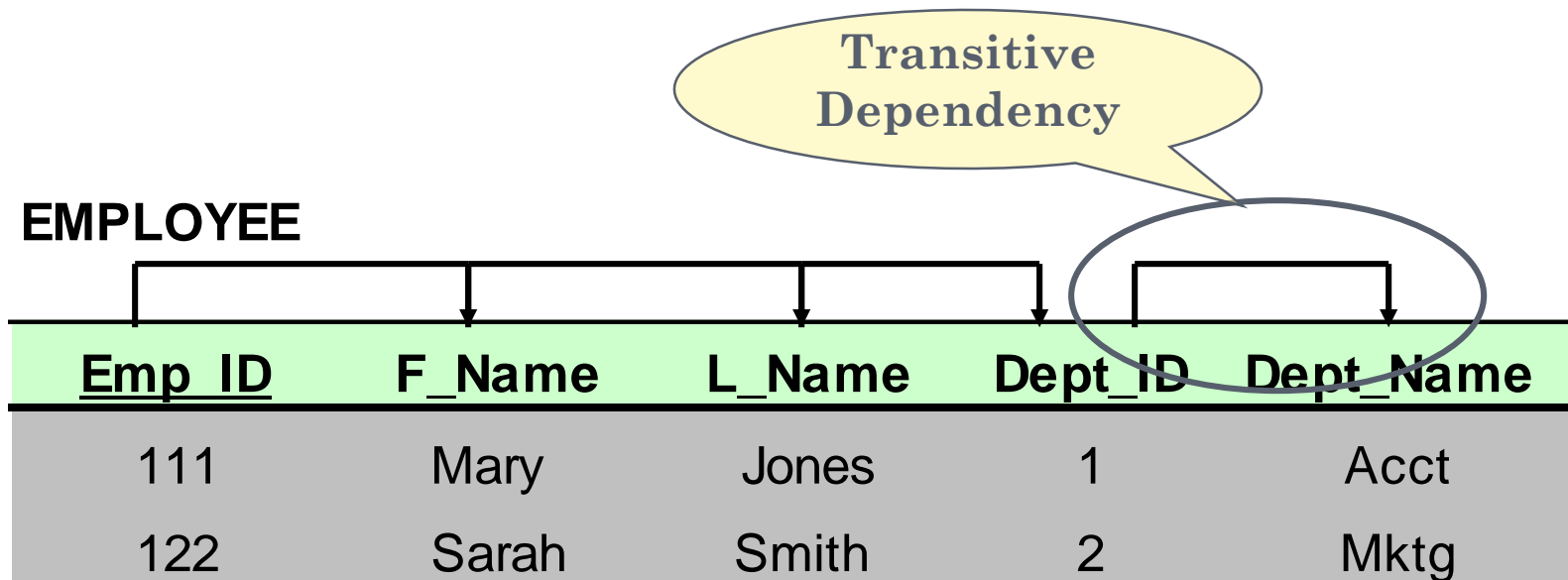
**CUSTOMER**

Partial Dependency

<u>Cust ID</u>	Name	<u>Order ID</u>
101	AT&T	1234
101	AT&T	156
125	Cisco	1250

# DEPENDENCIES: DEFINITIONS

- **Transitive Dependency** – when a non-key attribute determines another non-key attribute.

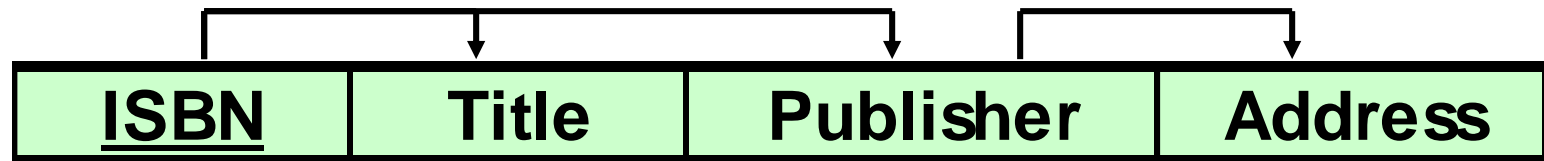


# EXAMPLE 1: DETERMINE NF

- ISBN → Title
- ISBN → Publisher
- Publisher → Address

All attributes are directly or indirectly determined by the primary key; therefore, the relation is at least in 1 NF

## BOOK

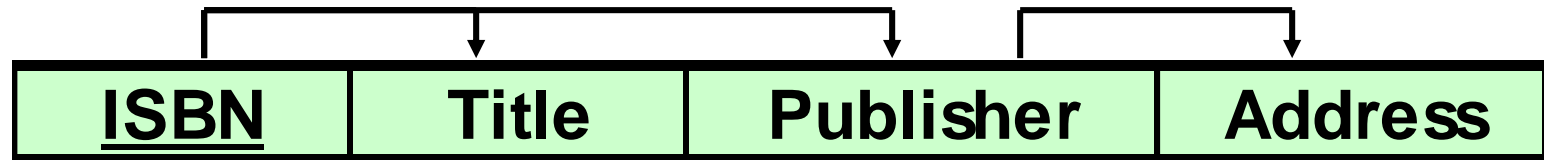


# EXAMPLE 1: DETERMINE NF

- ISBN → Title
- ISBN → Publisher
- Publisher → Address

The relation is at least in 1NF. There is no COMPOSITE primary key, therefore there can't be partial dependencies. Therefore, the relation is at least in 2NF

**BOOK**



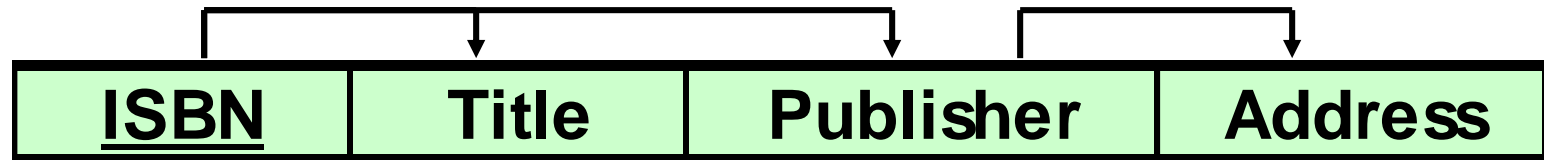


# EXAMPLE 1: DETERMINE NF

- ISBN  $\rightarrow$  Title
- ISBN  $\rightarrow$  Publisher
- Publisher  $\rightarrow$  Address

Publisher is a non-key attribute, and it determines Address, another non-key attribute. Therefore, there is a transitive dependency, which means that the relation is NOT in 3 NF.

## BOOK

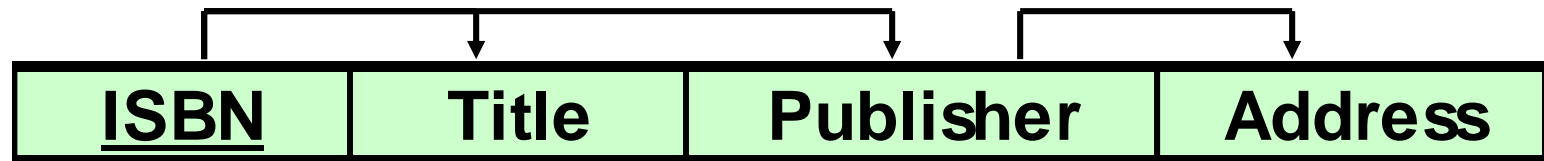


# EXAMPLE 1: DETERMINE NF

- ISBN → Title
- ISBN → Publisher
- Publisher → Address

We know that the relation is at least in 2NF, and it is not in 3NF. Therefore, we conclude that the relation is in 2NF.

## BOOK



# EXAMPLE 1: DETERMINE NF

- ISBN → Title
- ISBN → Publisher
- Publisher → Address

In your solution you will write the following justification:

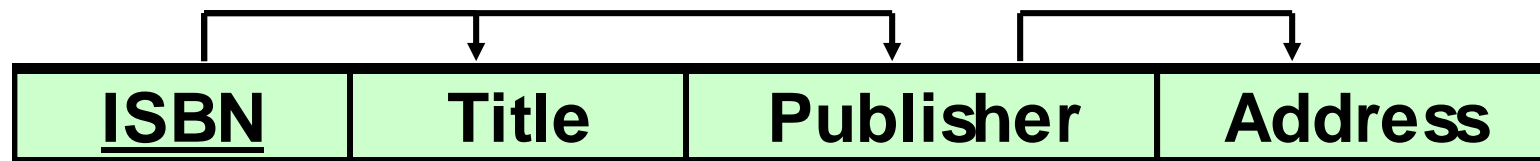
1) No M/V attributes, therefore at least 1NF

2) No partial dependencies, therefore at least 2NF

3) There is a transitive dependency (Publisher → Address), therefore, not 3NF

Conclusion: The relation is in 2NF

## BOOK



43

## EXAMPLE 2: DETERMINE NF

- Product\_ID → Description

All attributes are directly or indirectly determined by the primary key; therefore, the relation is at least in 1 NF

**ORDER**

<u>Order No</u>	<u>Product ID</u>	Description
-----------------	-------------------	-------------

## EXAMPLE 2: DETERMINE NF

- Product\_ID → Description

The relation is at least in 1NF.

There is a COMPOSITE Primary Key (PK) (Order No, Product ID), therefore there can be partial dependencies. Product\_ID, which is a part of PK, determines Description; hence, there is a partial dependency. Therefore, the relation is not 2NF. No sense to check for transitive dependencies!

**ORDER**

<u>Order No</u>	<u>Product ID</u>	Description
-----------------	-------------------	-------------

## EXAMPLE 2: DETERMINE NF

- Product\_ID → Description

We know that the relation is at least in 1NF, and it is not in 2NF. Therefore, we conclude that the relation is in 1NF.

**ORDER**

<u>Order No</u>	<u>Product ID</u>	Description
-----------------	-------------------	-------------

## EXAMPLE 2: DETERMINE NF

- Product\_ID → Description

In your solution you will write the following justification:

1) No M/V attributes, therefore at least 1NF

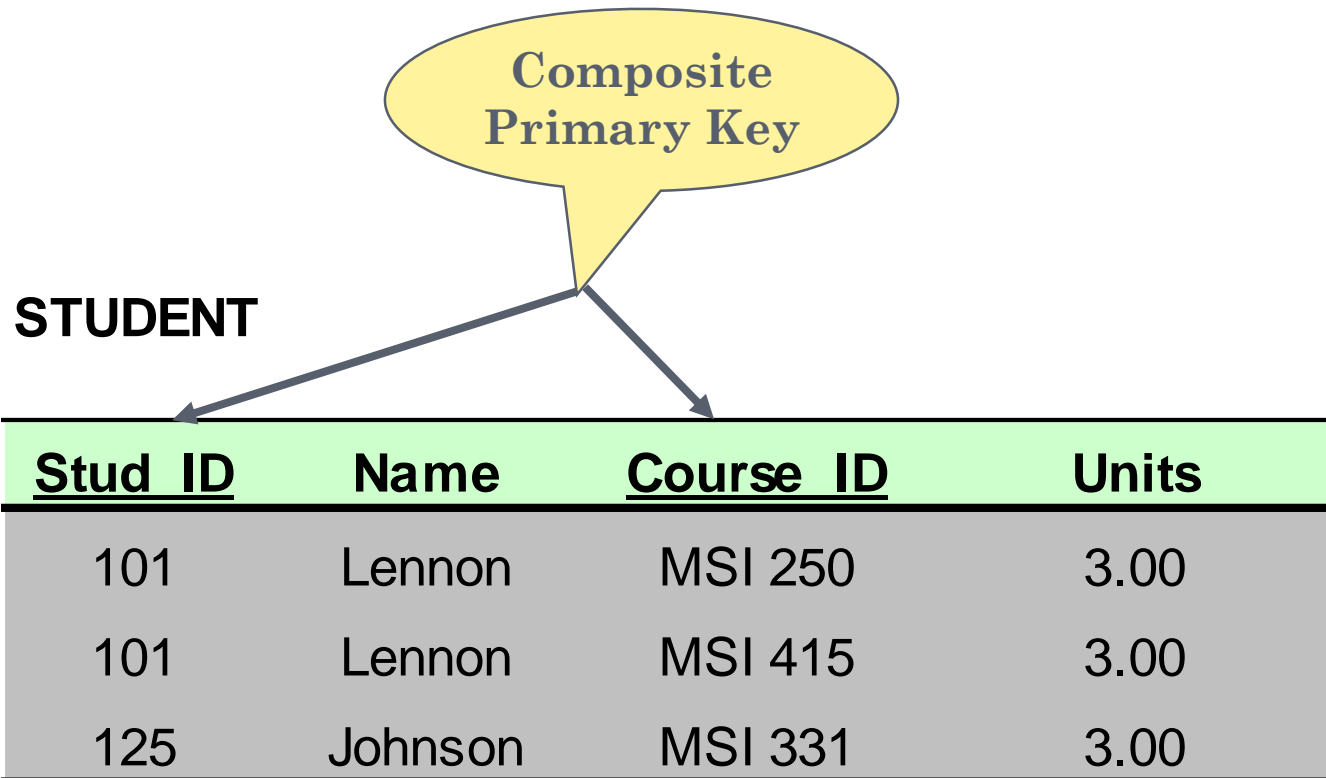
2) There is a partial dependency (Product\_ID → Description), therefore not in 2NF

Conclusion: The relation is in 1NF

**ORDER**

<u>Order No</u>	<u>Product ID</u>	Description
-----------------	-------------------	-------------

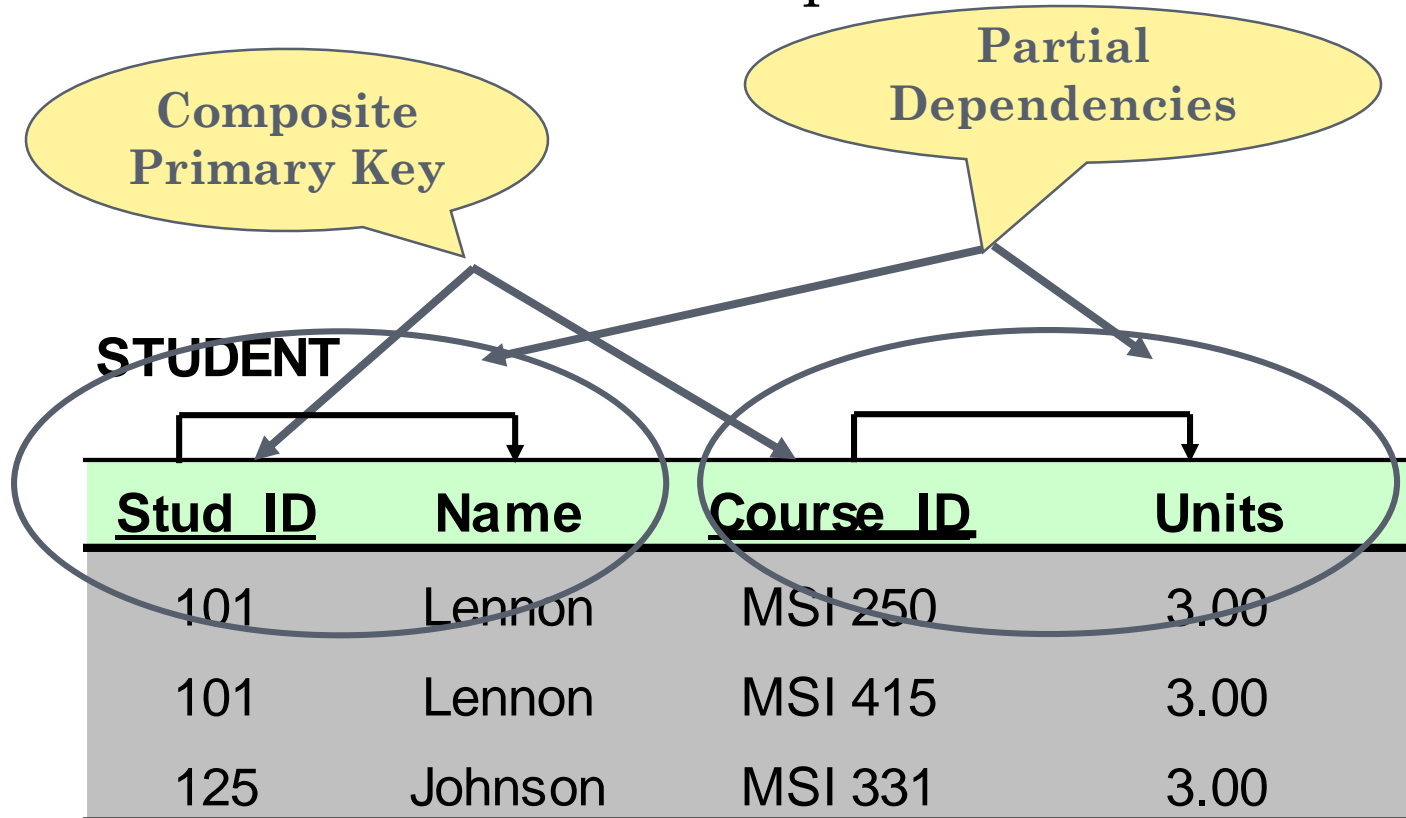
# BRINGING A RELATION TO 2NF





# BRINGING A RELATION TO 2NF

- Goal: Remove Partial Dependencies



# BRINGING A RELATION TO 2NF

- Remove attributes that are dependent from the part but not the whole of the primary key from the original relation. For each partial dependency, create a new relation, with the corresponding part of the primary key from the original as the primary key.

**STUDENT**

<u>Stud ID</u>	Name	<u>Course ID</u>	Units
101	Lennon	MSI 250	3.00
101	Lennon	MSI 415	3.00
125	Johnson	MSI 331	3.00

# BRINGING A RELATION TO 2NF

CUSTOMER

<u>Stud ID</u>	Name	<u>Course ID</u>	Units
101	Lennon	MSI 250	3.00
101	Lennon	MSI 415	3.00
125	Johnson	MSI 331	3.00

## STUDENT\_COURSE

<u>Stud ID</u>	<u>Course ID</u>
101	MSI 250
101	MSI 415
125	MSI 331

## STUDENT

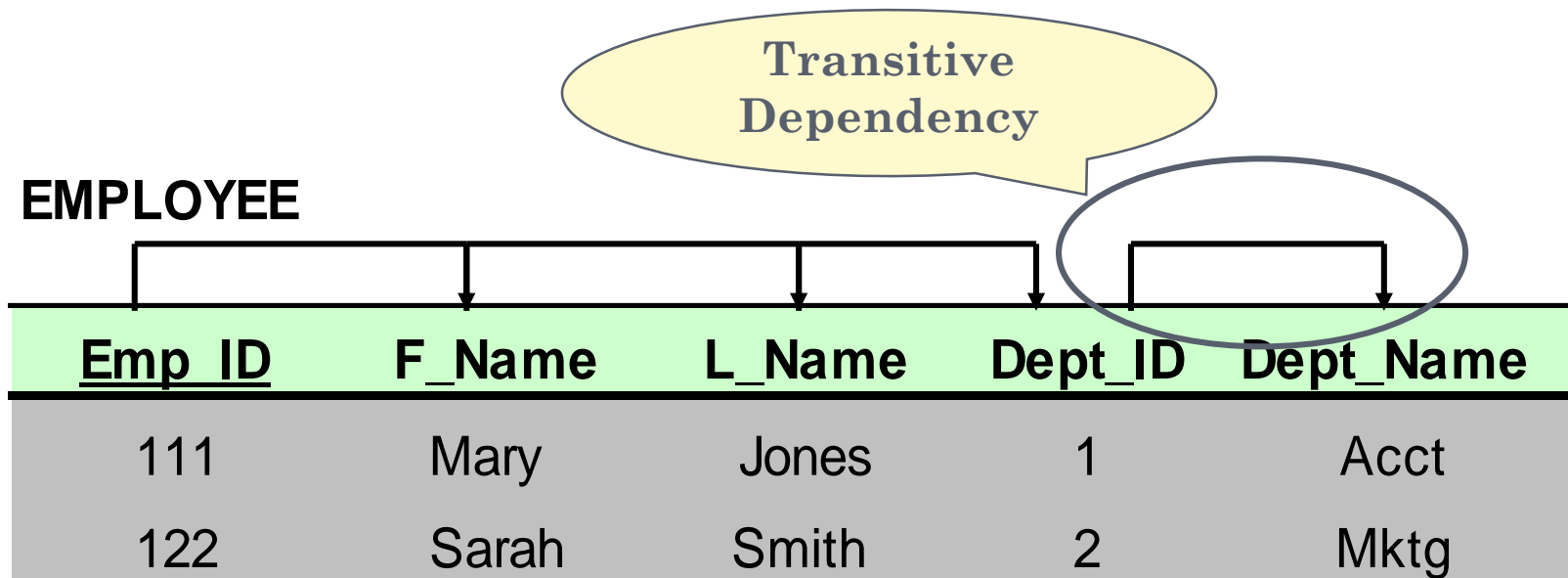
<u>Stud ID</u>	Name
101	Lennon
101	Lennon
125	Johnson

## COURSE

<u>Course ID</u>	Units
MSI 250	3.00
MSI 415	3.00
MSI 331	3.00

# BRINGING A RELATION TO 3NF

- Goal: Get rid of transitive dependencies.



# BRINGING A RELATION TO 3NF

- Remove the attributes, which are dependent on a non-key attribute, from the original relation. For each transitive dependency, create a new relation with the non-key attribute which is a determinant in the transitive dependency as a primary key, and the dependent non-key attribute as a dependent.

**EMPLOYEE**

<u>Emp_ID</u>	F_Name	L_Name	Dept_ID	Dept_Name
111	Mary	Jones	1	Acct
122	Sarah	Smith	2	Mktg

# BRINGING A RELATION TO 3NF

EMPLOYEE

<u>Emp ID</u>	F_Name	L_Name	Dept ID	Dept Name
111	Mary	Jones	1	Acct
122	Sarah	Smith	2	Mktg

EMPLOYEE

<u>Emp ID</u>	F_Name	L_Name	Dept ID
111	Mary	Jones	1
122	Sarah	Smith	2

DEPARTMENT

<u>Dept ID</u>	Dept Name
1	Acct
2	Mktg