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 in 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 restructure 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 → B

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 → email address

Functional Dependencies

EmpNum	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

91.2914

If EmpNum is the PK then the FDs:

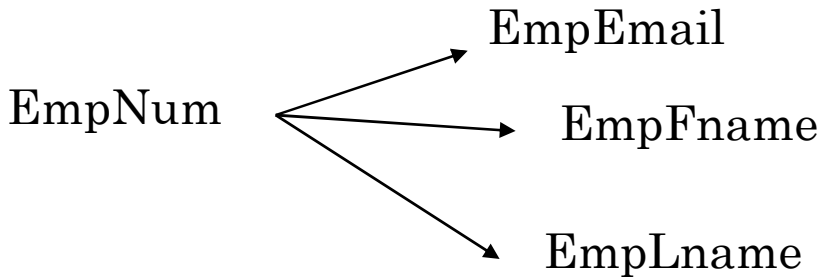
EmpNum → EmpEmail
EmpNum → EmpFname
EmpNum → EmpLname

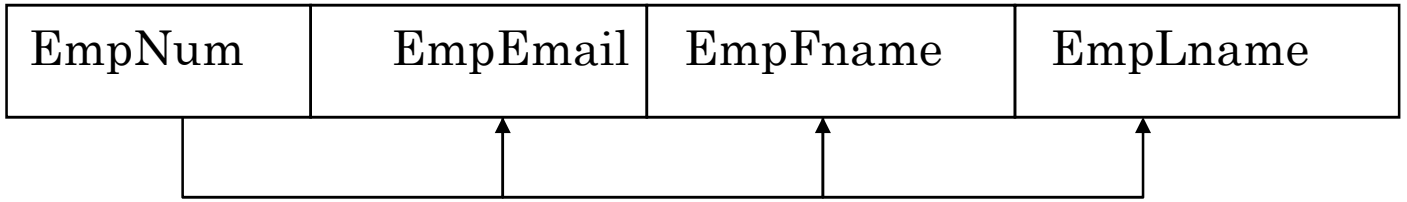
must exist.

FUNCTIONAL DEPENDENCIES

EmpNum → EmpEmail
EmpNum → EmpFname
EmpNum → EmpLname

3 different ways you might see FDs depicted





DETERMINANT

Functional Dependency

EmpNum → EmpEmail

91.2914

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 → B and B → C.

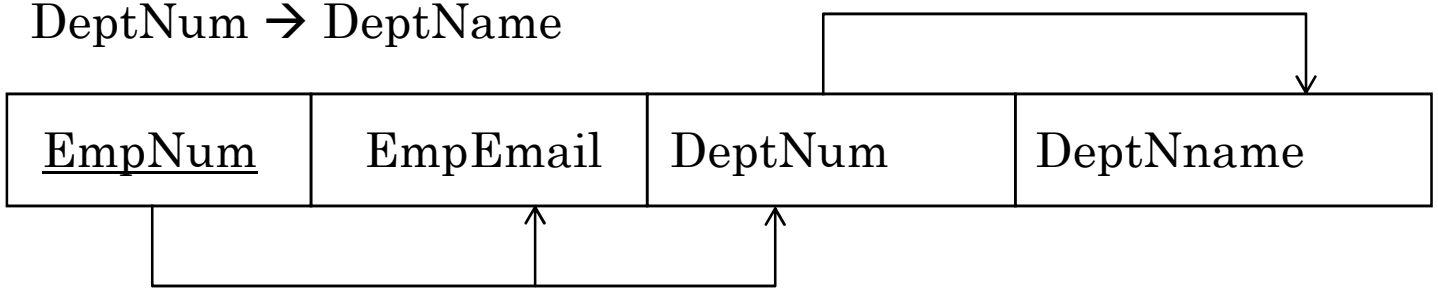
Functional dependencies are transitive, which means that we also have the functional dependency A → C

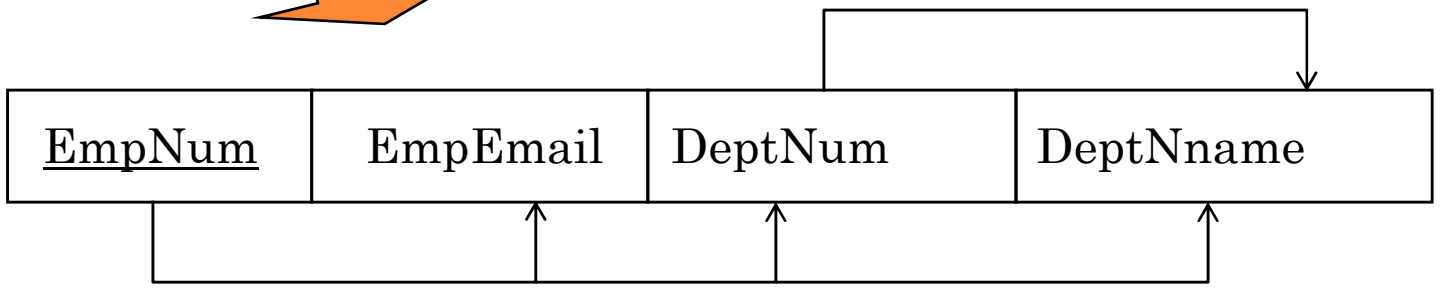
We say that C is transitively dependent on A through B.

TRANSITIVE DEPENDENCY

EmpNum → DeptNum

DeptNum → DeptName





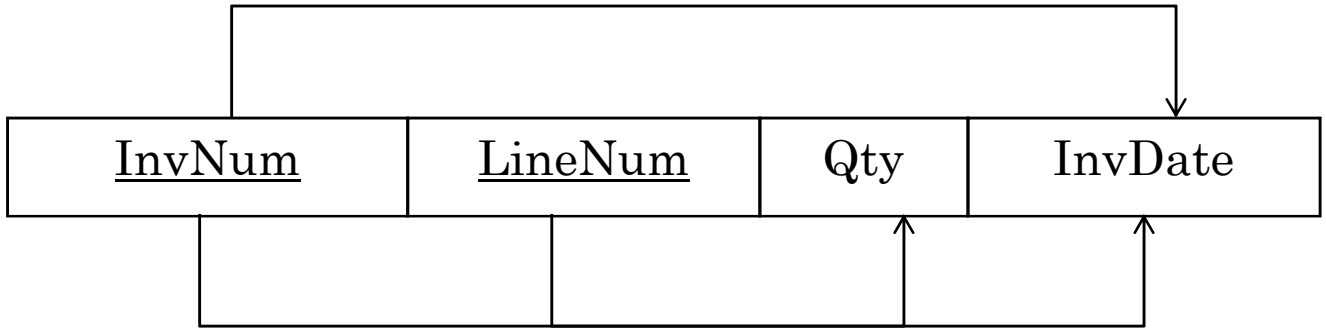
DeptName is *transitively dependent* on EmpNum via DeptNum

EmpNum → DeptName

91.2914

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

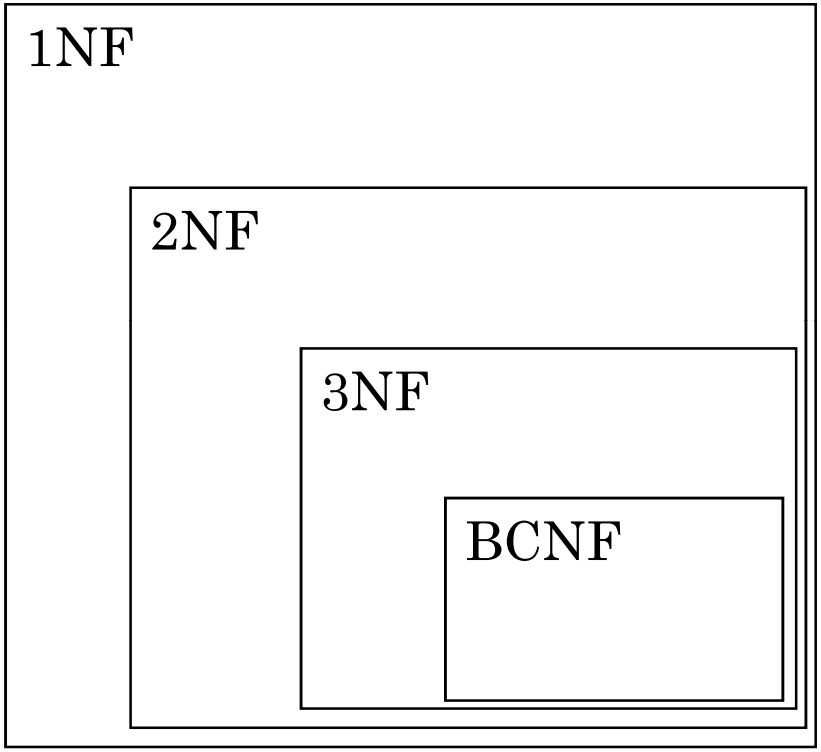
NORMALIZATION

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 3rd NF we must have 2nd NF and in order to have 2nd NF we must have 1st 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

91.2914

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 1st 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 in **not** in 1NF

EmpNum	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

EmpNum	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

EmpNum	EmpPhone
123	233-9876
333	233-1231
679	233-1231

EmployeeDegree

EmpNum	EmpDegree
333	BA
333	BSc
333	PhD
679	BSc
679	MSc

91.2914

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 ⟶ ProdNum, Qty

InvLine is only in **1NF**

InvNum ⟶ InvDate

InvNum	LineNum	ProdNum	Qty	InvDate

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

SECOND NORMAL FORM

InvLine

InvNum	LineNum	ProdNum	Qty	InvDate

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

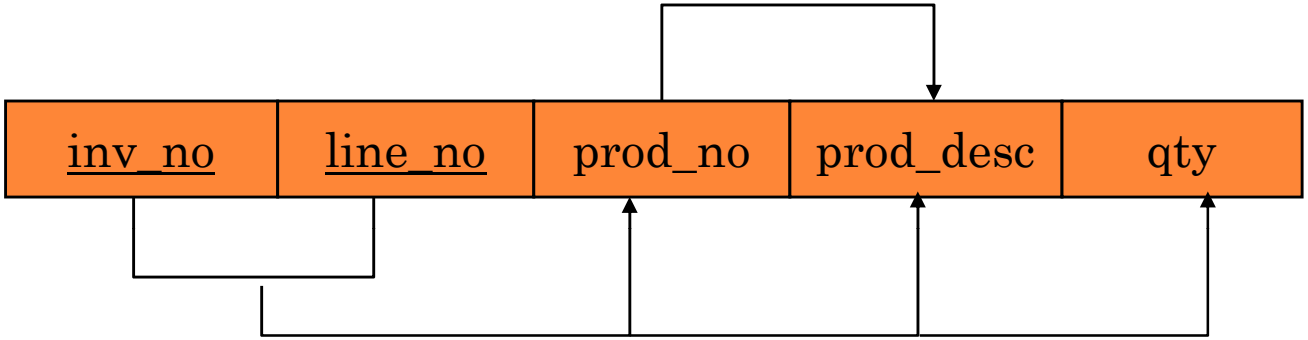
We can *improve* the database by decomposing the relation into two relations:

InvNum	LineNum	ProdNum	Qty

InvNum	InvDate

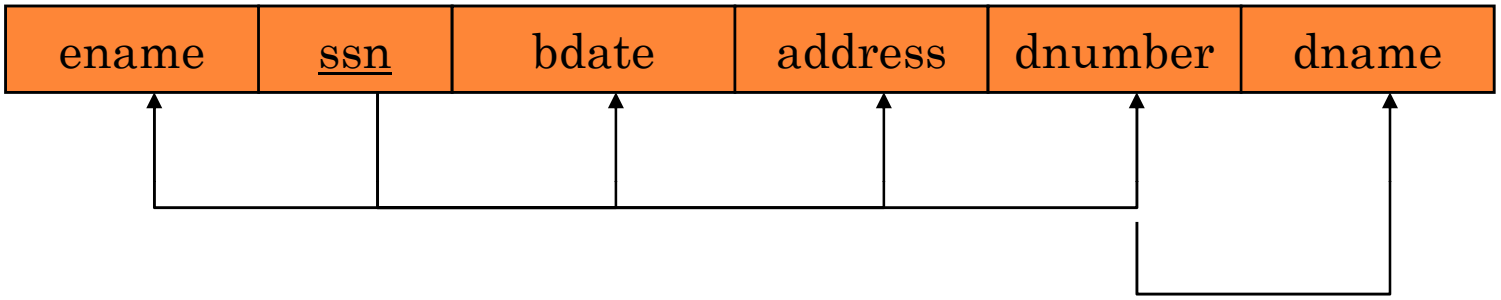
91.2914

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:

dnumber → dname.

3RD NORMAL FORM
THE REQUIREMENTS

- The requirements to satisfy the 3rd NF:
  - All requirements for 2nd NF must be met.
  - Eliminate fields that do not depend on the primary key;
    - 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.

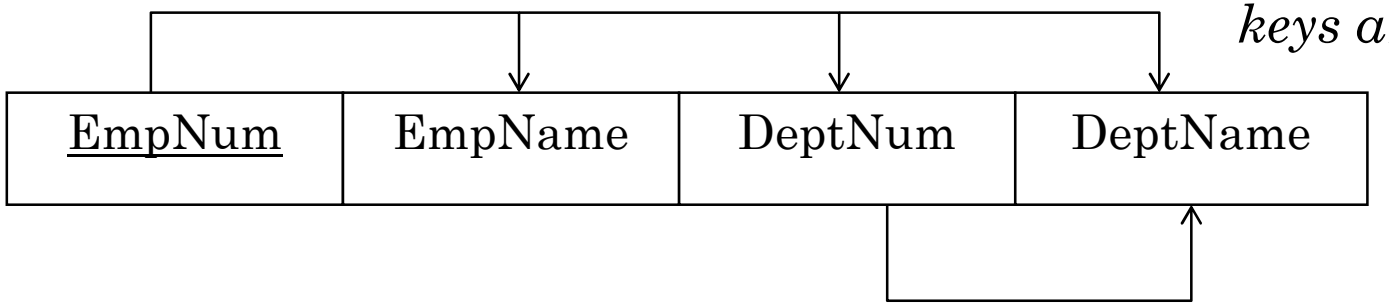
- That is, for any functional dependency: $X \rightarrow Y$, where Y is a non-key attribute (or a set of non-key attributes), X is a candidate key.

- 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.

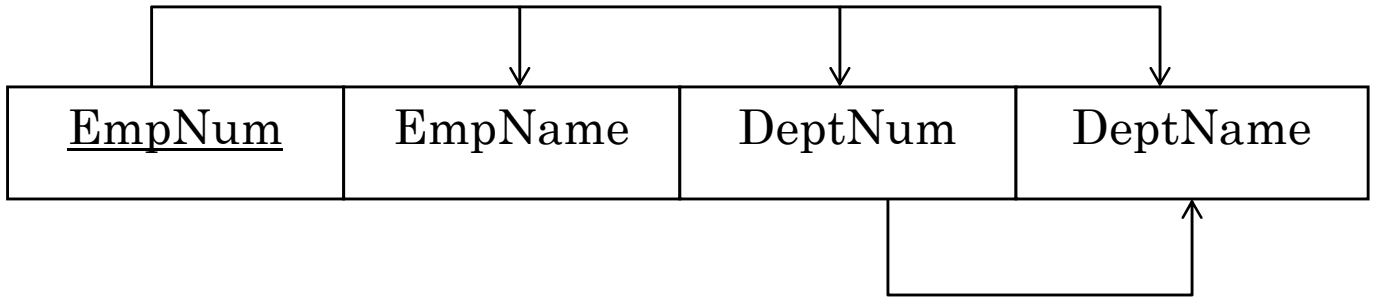
Is the relation in BCNF? … no

Is the relation in 3NF? … no

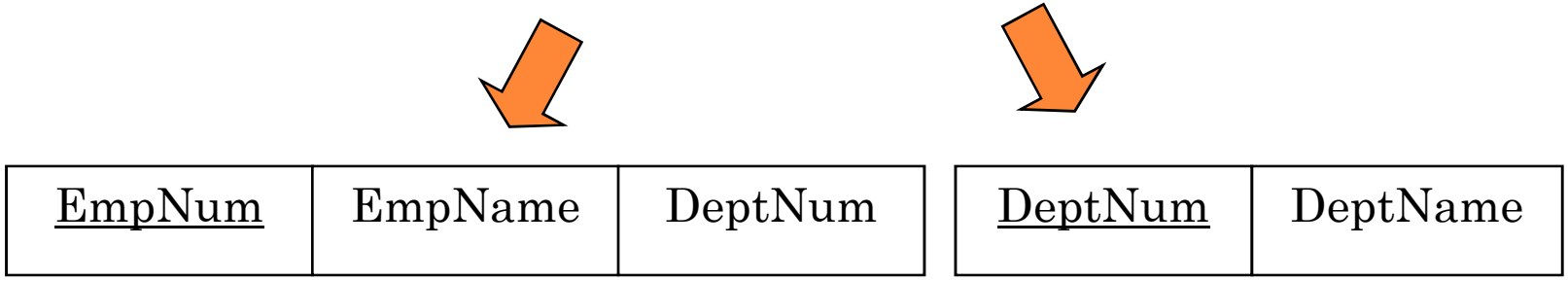
Is the relation in 2NF? … yes, no partial dependency

91.2914

THIRD NORMAL FORM



We correct the situation by decomposing the original relation into two 3NF relations. Note the decomposition is *lossless*.

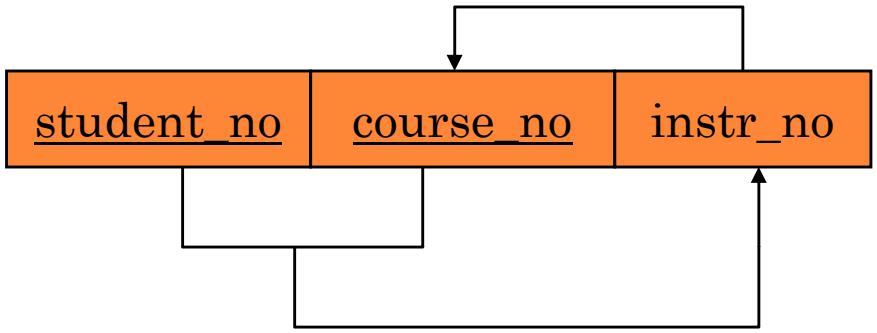


Verify these two relations are in 3NF.

No transitive dependency

91.2914

In 3NF, but not in BCNF:



Instructor teaches one course only.

Student takes a course and has one instructor.

{student_no, course_no} → instr_no
instr_no → course_no

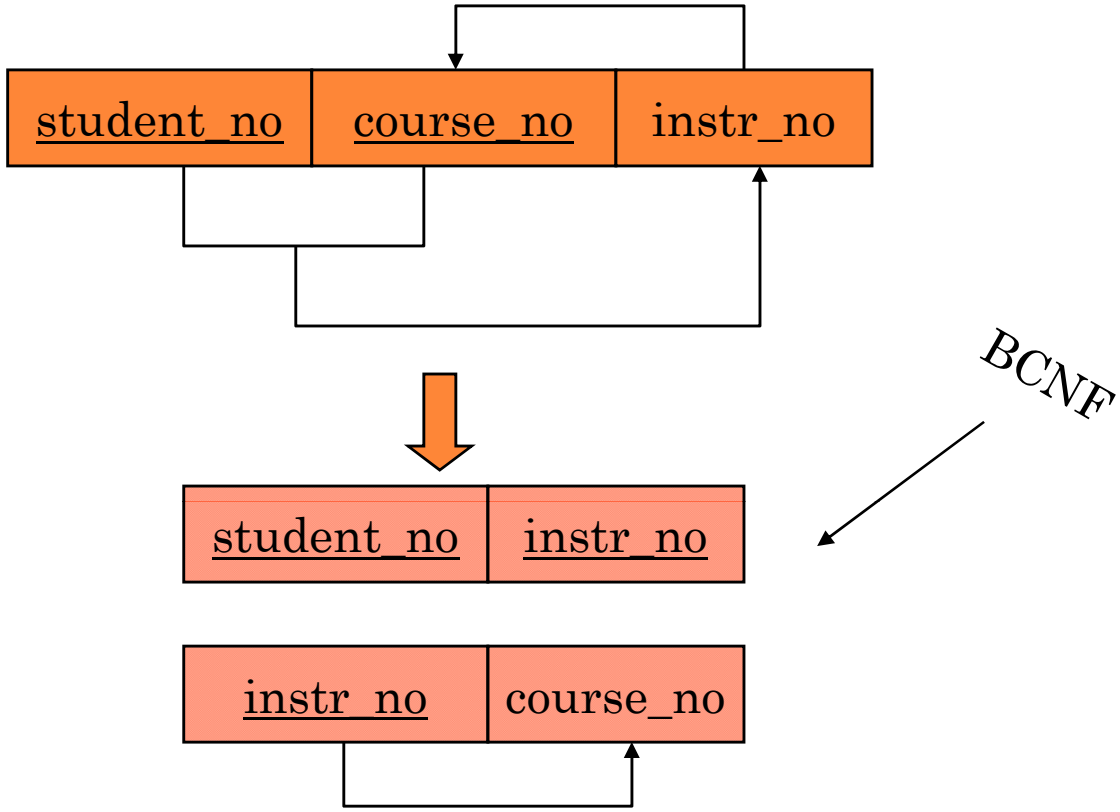
since we have instr_no → course-no, but instr_no is not a Candidate key.

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.



instr_no → course_no

More Examples

Un-normalized Students table:

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

Normalized Students table:

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

1ST NORMAL FORM EXAMPLE

Students table

Student#	AdvID	AdvName	AdvRoom
123	123A	James	555
124	123B	Smith	467

Registration table

Student#	Class#
123	102-8
123	104-9
124	209-0
124	102-8

Student# → Class#
Student# → AdvID
AdvID → {AdvName, AdvRoom}

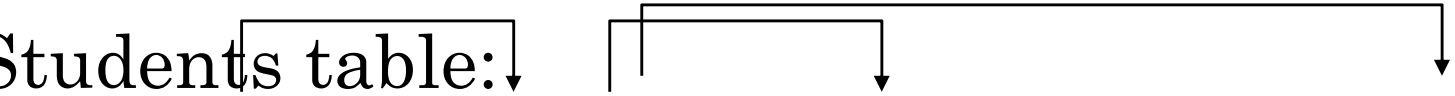
Is this in 2NF?

Yes, why?

No partial dependency

3RD NORMAL FORM EXAMPLE

Students table:



Student#	AdvID	AdvName	AdvRoom
123	123A	James	555
124	123B	Smith	467

Student table:

Student#	AdvID
123	123A
124	123B

Is this in 3NF?

No, why?

transitive dependency

3RD NORMAL FORM EXAMPLE CONT.

Students table:

Student#	AdvID
123	123A
124	123B

Registration table:

Student#	Class#
123	102-8
123	104-9
124	209-0
124	102-8

Advisor table:

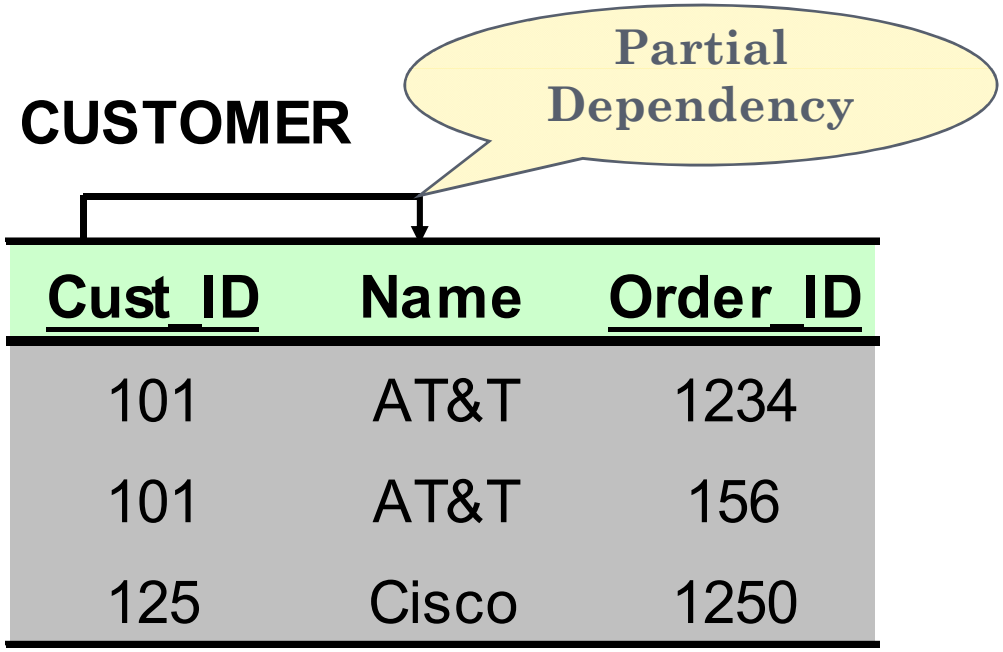
AdvID	AdvName	AdvRoom
123A	James	555
123B	Smith	467

Dependencies: Definitions

- ***Multivalued Attributes*** (or ***repeating groups***): non-key attributes or groups of non-key attributes the values of which are not uniquely identified by (directly or indirectly) (not functionally dependent on) the value of the Primary Key (or its part).

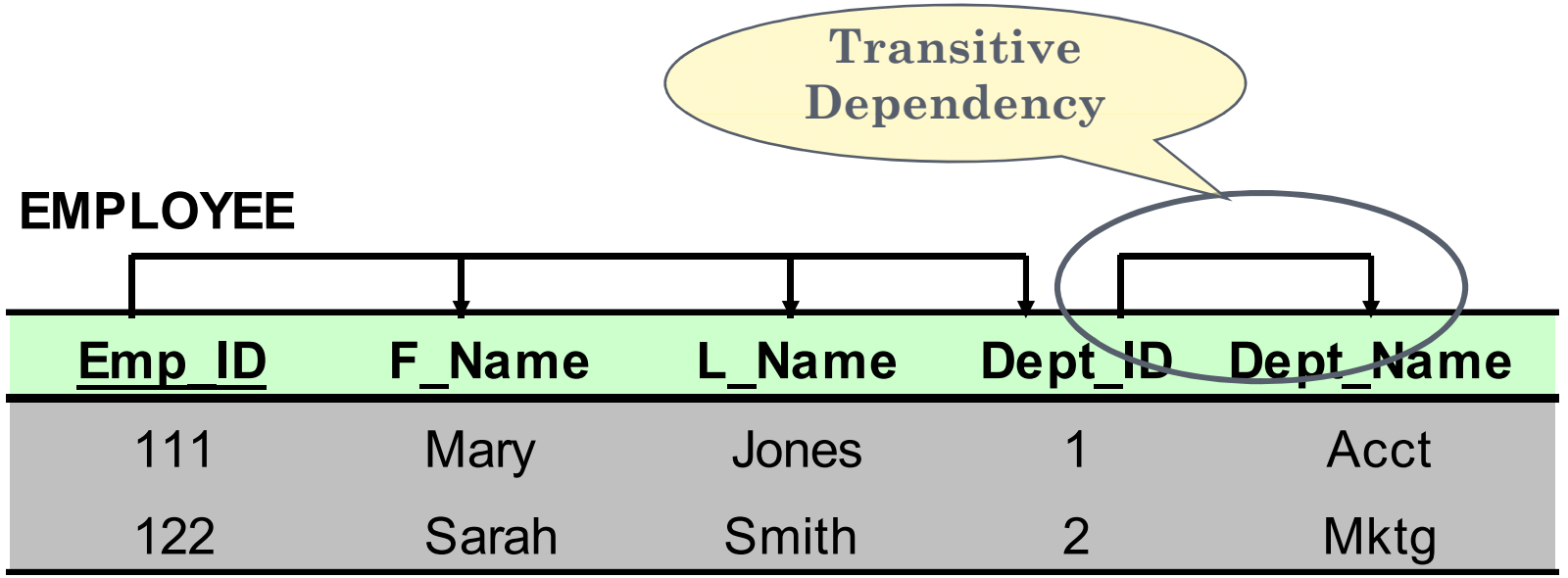
Dependencies: Definitions

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



Dependencies: Definitions

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



Normal Forms: Review

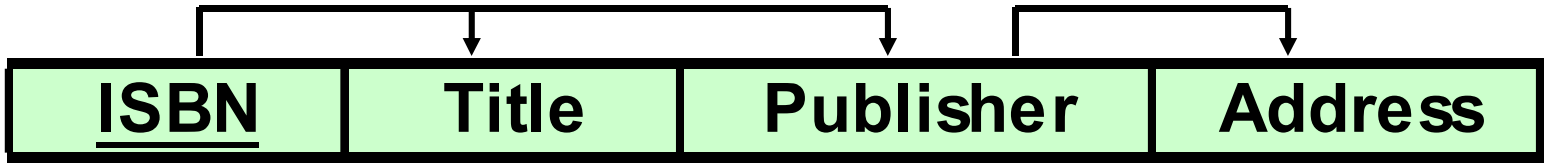
- Unnormalized – There are multivalued attributes or repeating groups
- 1 NF – No multivalued attributes or repeating groups.
- 2 NF – 1 NF plus no partial dependencies
- 3 NF – 2 NF plus no transitive dependencies

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

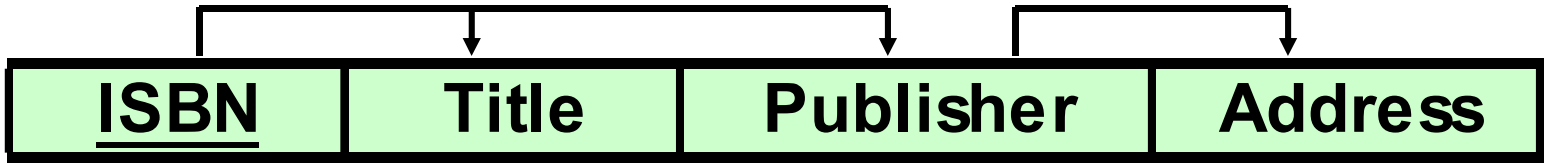


Example 1: Determine NF

- ISBN → Title
- ISBN → Publisher
- Publisher → Addre

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

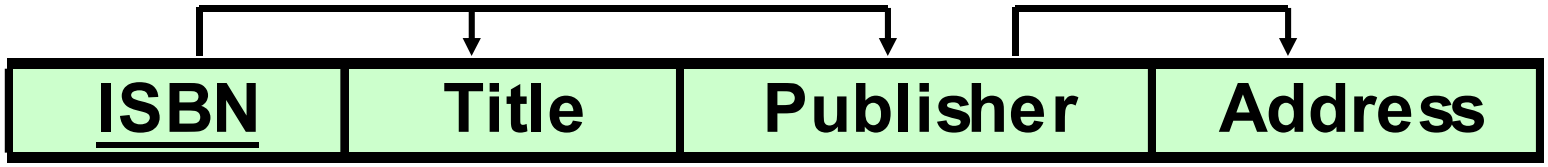


Example 1: Determine NF

- ISBN → Title
- ISBN → Publisher
- Publisher → Addres

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

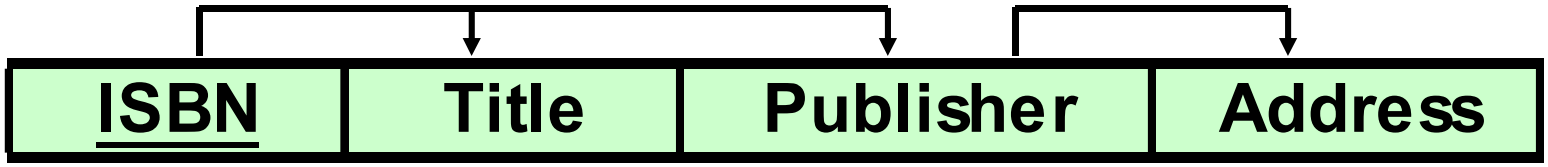


Example 1: Determine NF

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

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

BOOK



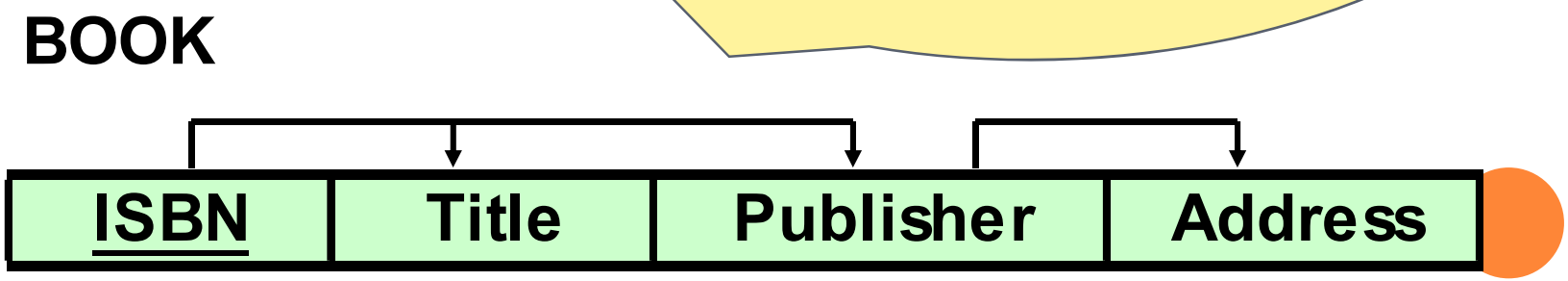
EXAMPLE 1: DETERMINE NF

- ISBN → Title
- ISBN → Publisher
- Publisher → Addres

In your solution you will write the following justification:

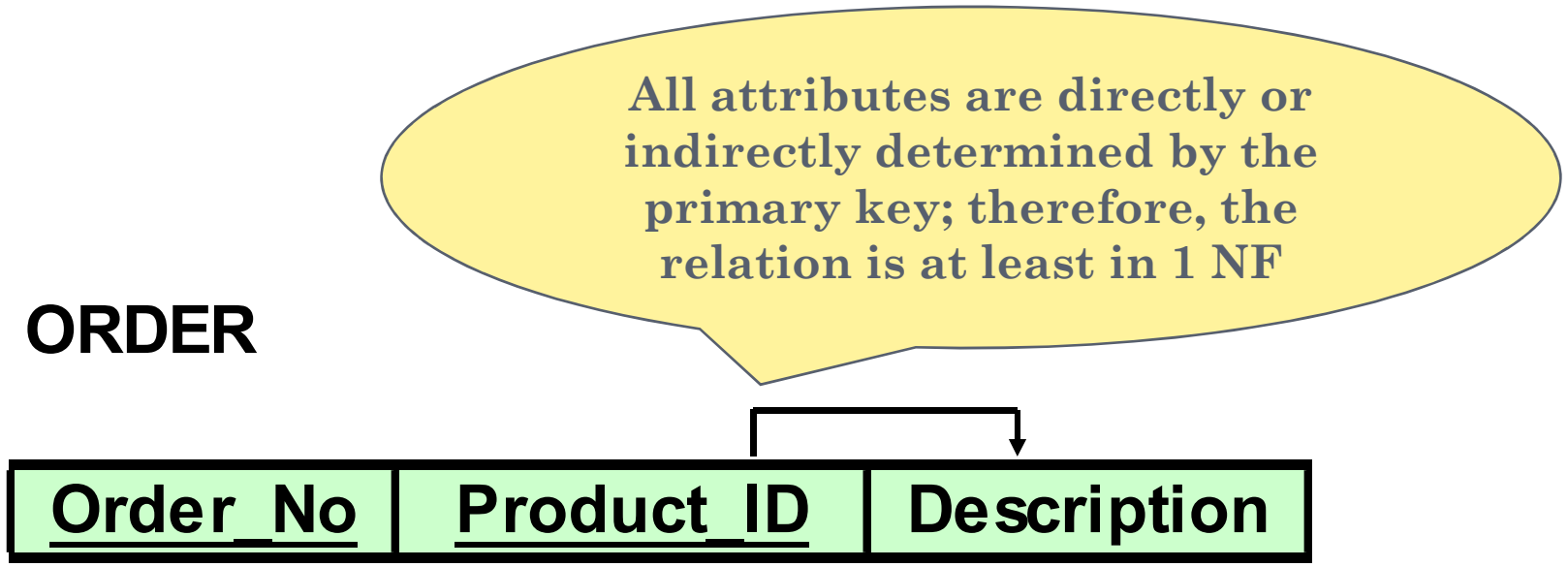
1) **No M/V attributes, therefore at least 1NF**
2) **No partial dependencies, therefore at least 2NF**
3) **There is a transitive dependency (Publisher → Address), therefore, not 3NF**

Conclusion: The relation is in 2NF



Example 2: Determine NF

- Product_ID → 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

Example 2: Determine NF

- Product_ID → Description

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

ORDER

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