ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT - HÀN
Vietnam - Korea University of Information and Communication Technology

# Database Systems

## *Chapter 3: Relational Model Normalization*

## session 2: Normal forms

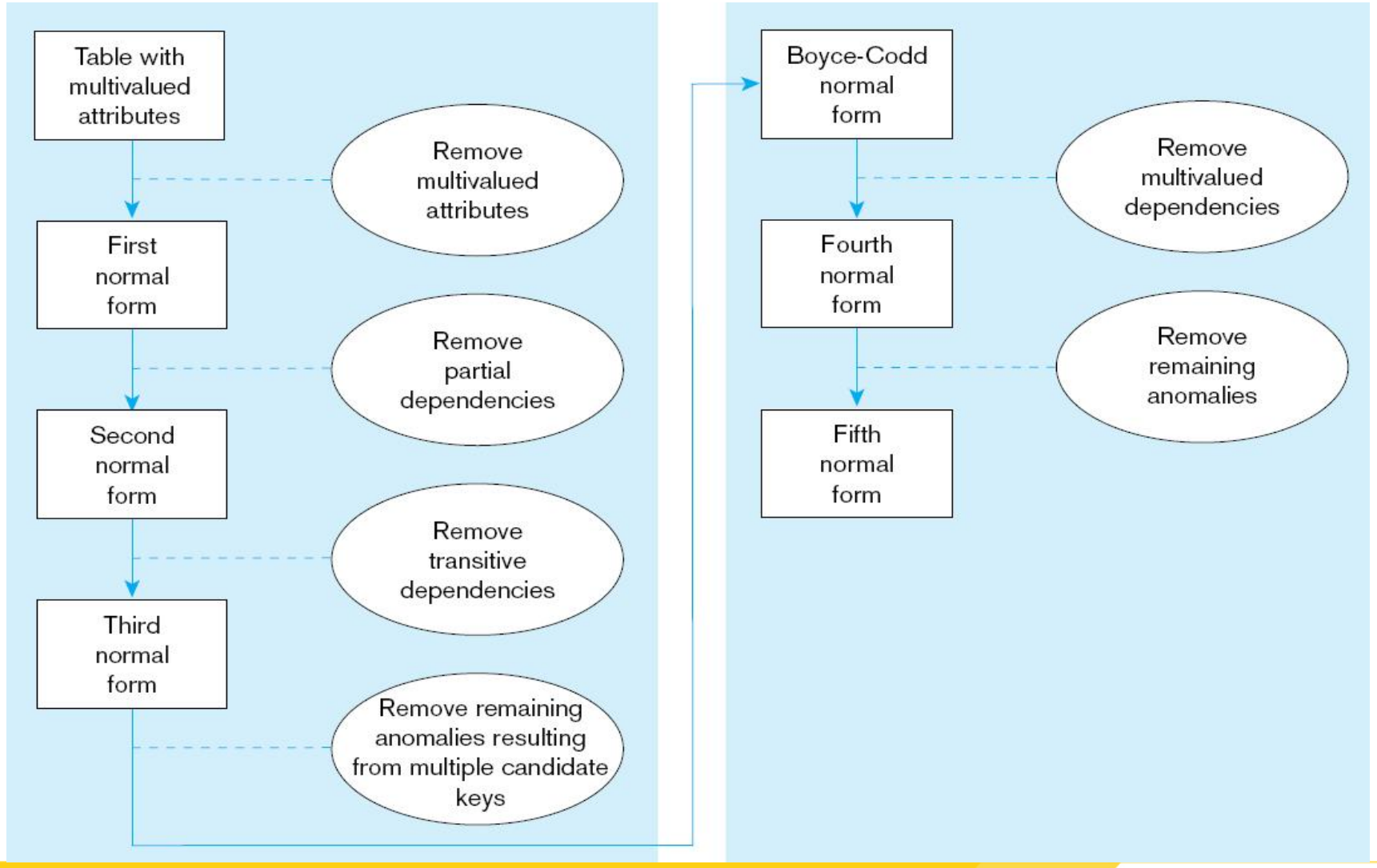
# Outline

1. Introduction to Normal Forms
2. The process of normalization
3. 1NF, 2NF, 3NF
4. BCNF

# Normal forms

- Normalization works through a series of stages called normal forms.
- The first normal form (1NF), second normal form (2NF), and third normal form (3NF).
- From a structural point of view, 2NF is better than 1NF, and 3NF is better than 2NF.
- For most purposes in business database design, 3NF is as high as you need to go in the normalization process.
- Highest level of normalization is not always most desirable

# The process of normalization



# First Normal Form (1NF)

- A relation is in first normal form (1NF) if:
  - No multivalued attributes or no repeating groups in the table. In other words, each row/column intersection contains one and only one value, not a set of values (Every attribute value is **atomic**)
  - A primary key has been defined, which uniquely identifies each row in the relation
- ***All relations* are in 1st Normal Form**

# First Normal Form (1NF)

- Example: Invoice tabular

**Table with multivalued attributes, not in 1st normal form**

| OrderID | Order Date | Customer ID | Customer Name | Customer Address | ProductID | Product Description | Product Finish | Product StandardPrice | Ordered Quantity |
|---|---|---|---|---|---|---|---|---|---|
| 1006 | 10/24/2010 | 2 | Value Furniture | Plano, TX | 7 | Dining Table | Natural Ash | 800.00 | 2 |
| | | | | | 5 | Writer's Desk | Cherry | 325.00 | 2 |
| | | | | | 4 | Entertainment Center | Natural Maple | 650.00 | 1 |
| 1007 | 10/25/2010 | 6 | Furniture Gallery | Boulder, CO | 11 | 4–Dr Dresser | Oak | 500.00 | 4 |
| | | | | | 4 | Entertainment Center | Natural Maple | 650.00 | 3 |

**Note: this is NOT a relation**

# First Normal Form (1NF)

- **Removing repeating groups**
  =>**Invoice** table with no multivalued attributes and unique rows

| OrderID | Order Date | Customer ID | Customer Name | Customer Address | ProductID | Product Description | Product Finish | Product StandardPrice | Ordered Quantity |
|---|---|---|---|---|---|---|---|---|---|
| 1006 | 10/24/2010 | 2 | Value Furniture | Plano, TX | 7 | Dining Table | Natural Ash | 800.00 | 2 |
| 1006 | 10/24/2010 | 2 | Value Furniture | Plano, TX | 5 | Writer's Desk | Cherry | 325.00 | 2 |
| 1006 | 10/24/2010 | 2 | Value Furniture | Plano, TX | 4 | Entertainment Center | Natural Maple | 650.00 | 1 |
| 1007 | 10/25/2010 | 6 | Furniture Gallery | Boulder, CO | 11 | 4–Dr Dresser | Oak | 500.00 | 4 |
| 1007 | 10/25/2010 | 6 | Furniture Gallery | Boulder, CO | 4 | Entertainment Center | Natural Maple | 650.00 | 3 |

# First Normal Form (1NF)

## ❑Select the primary key

There are four determinants in INVOICE, and their functional dependencies are the following:

OrderID → OrderDate, CustomerID, CustomerName, CustomerAddress
CustomerID → CustomerName, CustomerAddress
ProductID → ProductDescription, ProductFinish, ProductStandardPrice
OrderID, ProductID → OrderedQuantity

- The only candidate key for INVOICE is the composite key consisting of the attributes OrderID and ProductID.

**=>The invoice is in 1NF, but not a well-structured one**

# First Normal Form (1NF)

- Anomalies in the table:
  - **Insertion**–if new product is ordered for order 1007 of existing customer, customer data must be re-entered, causing duplication
  - **Deletion**–if we delete the Dining Table from Order 1006, we lose information concerning this item's finish (Natural Ash) and price ($800.00).
  - **Update**–changing the price of product ID 4 requires update in several records

  **Why do these anomalies exist?**

  Because there are multiple entity types in one relation. This results in duplication and an unnecessary dependency between the entities

# First Normal Form (1NF)

- Ex: Is the supplier table in 1NF?

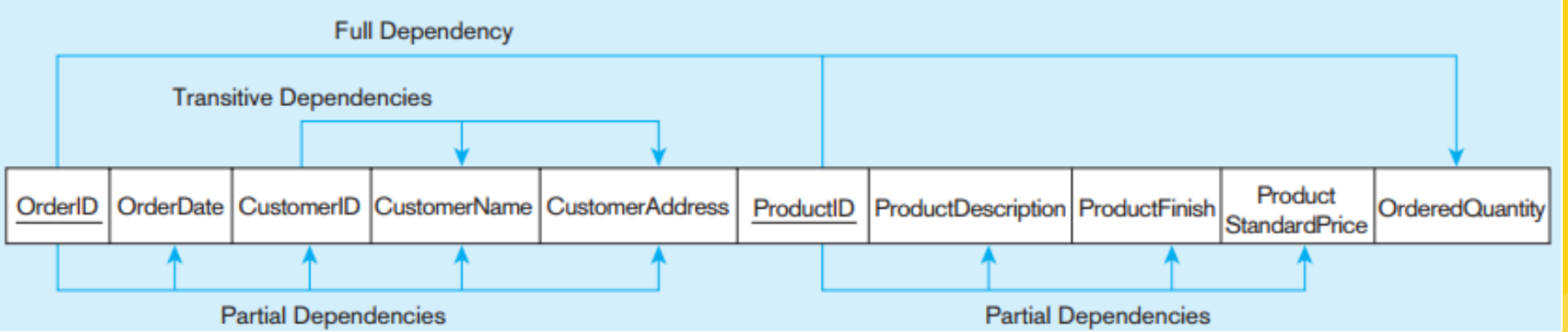
| s_id | status | city | part_id | quantity |
|---|---|---|---|---|
| S1 | 20 | London | P1 | 300 |
| | | | P2 | 200 |
| | | | P3 | 400 |
| | | | P4 | 200 |
| | | | P5 | 100 |
| | | | P6 | 100 |
| S2 | 10 | Paris | P1 | 300 |
| | | | P2 | 400 |
| s3 | 10 | Paris | P2 | 200 |
| S4 | 20 | London | P2 | 200 |
| | | | P4 | 300 |
| | | | P5 | 500 |

# Second Normal Form (2NF)

- A relation is in second normal form (2NF) if:
  - It is in 1NF
  - There is no partial dependency.
- A **partial functional dependency** exists when a non-key attribute is functionally dependent on part (but not all) of the primary key
- If a relation in 1NF has a primary key with single attribute, it is automatically in 2NF.

# Second Normal Form (2NF)

- Functional dependency diagram for INVOICE




- Primary key for relation is (OrderID,ProductID)
- There are following partial dependencies

OrderID → OrderDate, CustomerID, CustomerName, CustomerAddress
ProductID → ProductDescription, ProductFinish, ProductStandardPrice

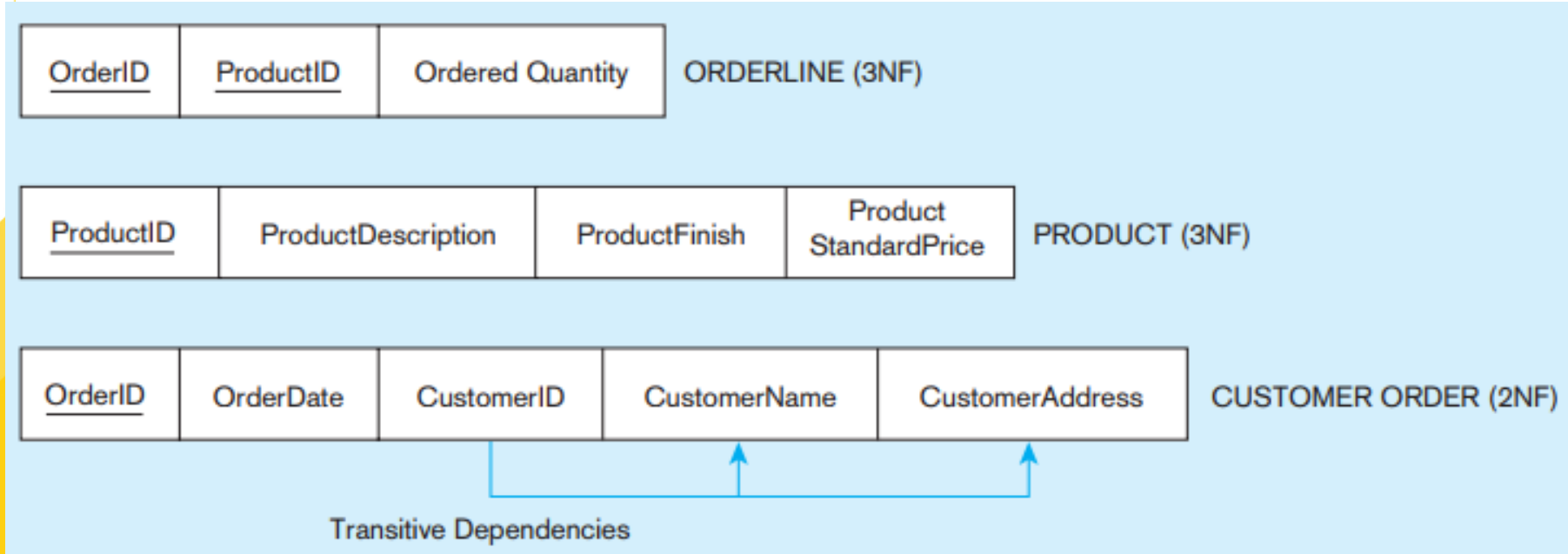
Therefore, it is NOT in 2nd Normal Form

# Second Normal Form (2NF)

- To convert a relation with partial dependencies to 2NF, the following steps are required:
  1. Create a new relation for each primary key attribute (or combination of attributes) that is a determinant in a partial dependency. That attribute is the primary key in the new relation.
  2. Move the non-key attributes that are dependent on this primary key attribute (or attributes) from the old relation to the new relation

# Second Normal Form (2NF)

- Removing partial dependencies




**Partial dependencies are removed, but there are still transitive dependencies**

# Exercises

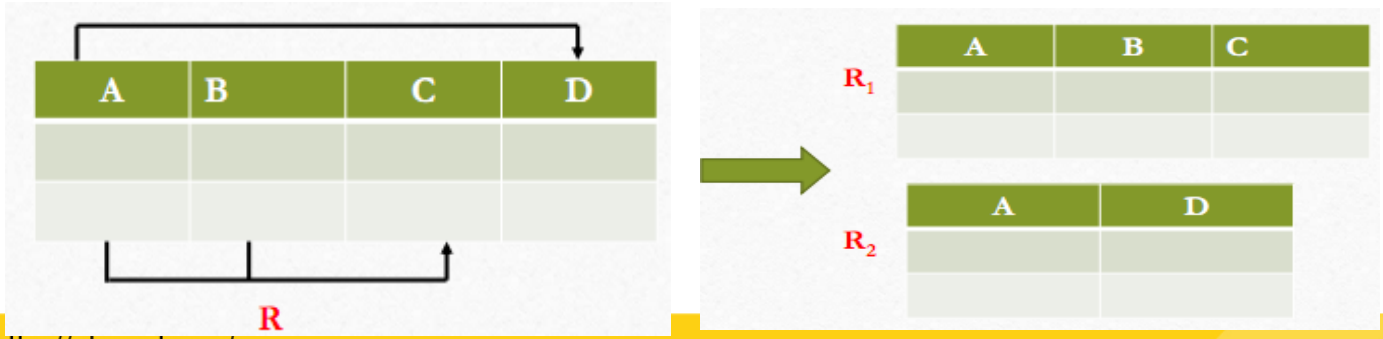
- **Ex1: Consider a relation R(A,B,C,D) with F = { AB -> C, A -> D}. Test whether R is in 2NF or not.**

=> AB is an only candidate key. Why?

A and B are called prime attributes and C and D are non-prime attributes.

AB -> C satisfies 2NF. But, A -> D is partial dependency.

So, R is not in 2NF. Therefore, decomposing relation into 2 relations R1(A,B,C) and R2(A, D). R1,R2 is in 2NF.



# Exercises

**Consider the following relations, check whether them are in 2NF or not. If not, decompose it into 2NF.**

- Ex2: The relation R(A,B,C,D) and
  F = { AB -> C,  B -> D}.
- Ex3: The relation R(A,B,C,D,E) and
  F = { AB -> C,  A -> D, B -> E}.

# Third Normal Form (3NF)

- A relation is in third normal form (3NF) if:
  - It is in 2NF
  - There is no **transitive dependency**.
- A **transitive dependency** in a relation is a functional dependency between the primary key and one or more *non-key attributes that are dependent on the primary key via another non-key attribute*
- In the other words, a transitive dependency exists when a *non-key* attribute can determine another non-key attribute

# Third Normal Form (3NF)

- There are 2 transitive dependencies in CustomerOrder relation:

$$\text{OrderID} \rightarrow \text{CustomerID} \rightarrow \text{CustomerName}$$
$$\text{OrderID} \rightarrow \text{CustomerID} \rightarrow \text{CustomerAddress}$$

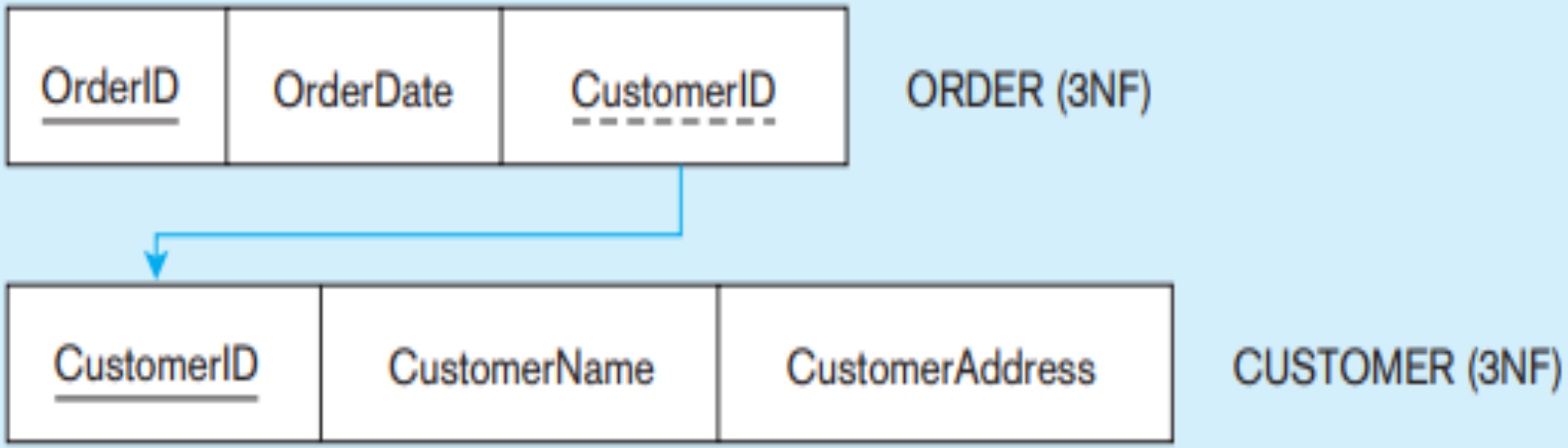
  - Both CustomerName and CustomerAddres are uniquely identified by CustomerID but CustomerID is not part of the primary key.

# Third Normal Form (3NF)

- Removing transitive dependencies by the following steps:
  1. For each non- key attribute(s) that is a determinant in a relation, **create a new relation**. *That attribute (s) becomes the primary key of the new relation.*
  2. Move all of the attributes that are functionally dependent on the primary key of the new relation from the old to the new relation.
  3. Primary key in the new relation **serve as a foreign key** in the old relation.

# Third Normal Form (3NF)

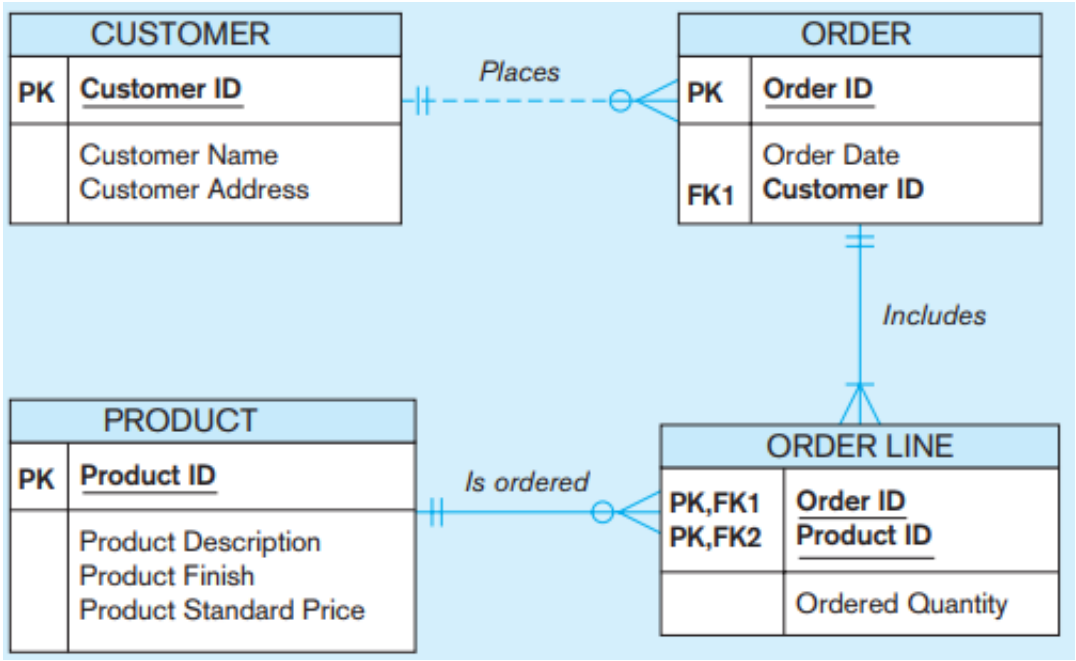
- The results of applying the steps to the relation CustomerOrder




**The CustomerOrder relation is divided into 2 relations Customer and Order =>These relation are in 3NF**

# Third Normal Form (3NF)

- A relational schema showing these four relations and their associations in ERD



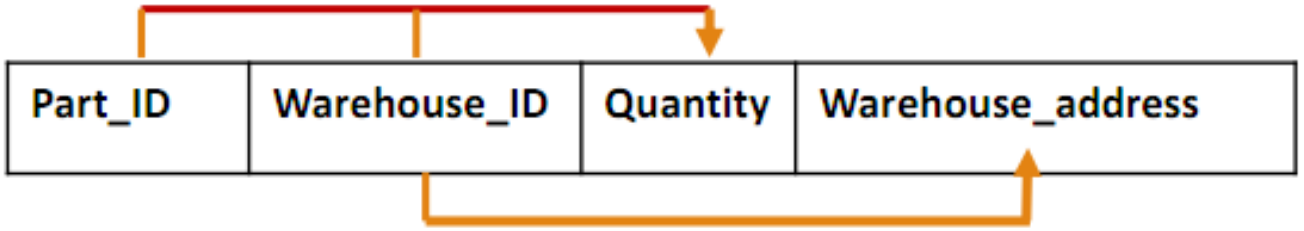
# Exercises

- Ex1: Is the **Inventory** relation in 3 NF?

Inventory

| Part_ID | Warehouse_ID | Quantity | Warehouse_address |
|---|---|---|---|
| 1 | 1 | 5 | Kathmandu |
| 1 | 2 | 3 | Lalitpur |
| 2 | 1 | 2 | Kathmandu |
| 3 | 1 | 2 | Kathmandu |
| 4 | 2 | 4 | Lalitpur |

- The dependency is shown below:

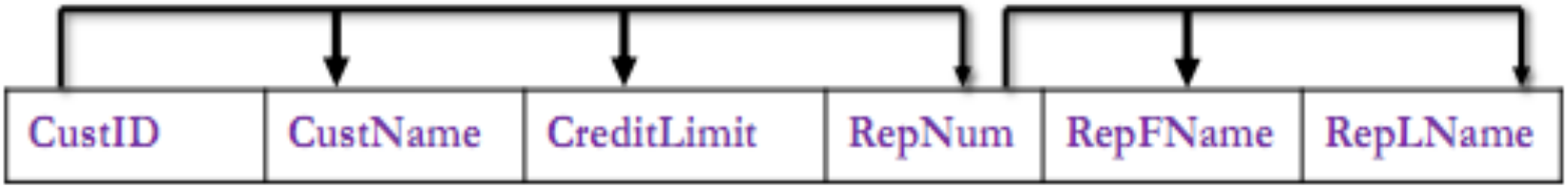


# Exercises

- Ex2: Is the **Customer** relation in 3 NF?

| CustID | CustName | Creditlimit | RepNum | RepFName | RepLName |
|---|---|---|---|---|---|
| 101 | KMC | 15000 | 15 | Ishan | Maskey |
| 202 | Teaching Hospital | 20000 | 25 | Santosh | Sharma |
| 303 | Manmohan Memorial | 18000 | 15 | Ishan | Maskey |
| 305 | Star Hospital | 12000 | 50 | Kiran | Chemjong |
| 505 | B and B | 20000 | 25 | Santosh | Sharma |

- The dependency is shown below:



# Exercises

Are the following relations in 3NF? If not, decompose them into 3NF.

- Ex3: Consider a relation R(A,B,C,D) with FDs { AB -> C, C -> D}.
- Ex4: Consider the relation R(A,B,C,D, E) and F = { AB -> C,  B ->D,  D ->E }.

# Boyce-codd Normal Form (BCNF)

- A relation is in BCNF if:
  - It is in 3NF
  - For every functional dependency X-> Y, X (called determinant) should be the **candidate key** of the relation.
- BCNF can be violated only when the table contains more than one candidate key.
- When a table contains only one candidate key, the 3NF and the BCNF are equivalent.

# Boyce-codd Normal Form (BCNF)

- **Example 1: Is the following relation in BCNF?**

  **R(A, B, C, D)**

  **F= { A ->BCD; BC ->AD;  D->B}**

⇒Above relation is in 3NF. Why?

⇒2 Candidate keys is A and BC. Why?

But FD: D->B, D is not a candidate key.

So the relation R violates BCNF.

Thus, we break the relation R into 2 relation R1(A, D, C) and R2 (D, B), then R1, R2 are in BCNF.

# Boyce-codd Normal Form (BCNF)

- **Ex2: Is the following relation in BCNF?**

  **R(A, B, C, D)**

  **F= { AB ->CD; AC ->BD;  C->B}**

⇒Above relation is in 3NF (no partial dependencies and no transitive dependencies - the C->B: one key/prime attribute determines another key/prime attribute)

⇒2 Candidate keys is AB and AC

And FD: C->B, C is not a candidate key.

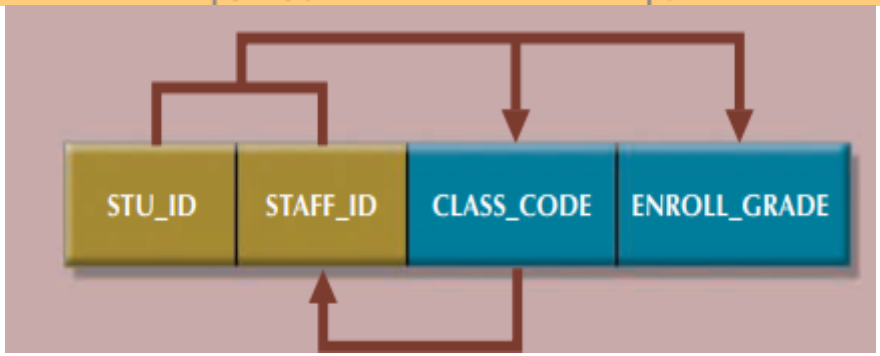
So the relation R violates BCNF. Thus, we break the relation R into 2 relation R1 and R2, then R1, R2 are in BCNF with R1(A, C, D), R2 (C,B)

# Boyce-codd Normal Form (BCNF)

## ❑Ex3: Is the Student relation in BCNF?

| STU_ID | STAFF_ID | CLASS_CODE | ENROLL_GRADE |
|---|---|---|---|
| 125 | 25 | 21334 | A |
| 125 | 20 | 32456 | C |
| 135 | 20 | 28458 | B |
| 144 | 25 | 27563 | C |
| 144 | 20 | 32456 | B |

- With FDs:




{Stu_ID, Staff_ID} -> {Class_code, Enroll_Grade}

Class_Code ->Staff_ID

# Boyce-codd Normal Form (BCNF)

=> 2 problems with the Student relation:

- Update anomaly: if a different staff member is assigned to teach class 32456, two rows will require updates
- deletion anomaly: if student 135 drops class 28458, information about who taught that class is lost

- The relation has 2 candidate keys: {Stu_ID, Staff_ID} and {StudentID, Class_Code}
- And Class_Code ->Staff_ID violates BCNF.
- So, decomposition to BCNF into 2 relations with Student(Stu_ID, Class_Code, Enroll_Grade} and Class(Class_Code, Staff_ID}

# BCNF (Boyce-Codd Normal Form)

- **Ex4: Consider the relation R(A,B,C,D) and F = { AB -> C, C -> D}. Is the relation in BCNF?**

=>AB is only one candidate key. Why?

- A,B are key/prime attributes and C,D are non-key/non-prime attributes.
- C -> D is transitive dependency (a non-key attribute determine another non-key attribute). So, R is not in 3NF. Therefore, R is not in BCNF as well.
- To convert to 3NF, decompose R into two relations R1(A,B,C) and R2(C,D). Now the relations will be in BCNF as well.

# Exercises

Determine whether the following relations is in BCNF or not? If not, decompose the relation with its NF.

- Ex1: Determine whether relation R(A,B,C) with
  F = { AB -> C, C -> A}
- Ex2: The relation StudMajor( StdNo, Major, Advisor) with FDs {StdNo, Major -> Advisor, Advisor -> Major

| STDNO | MAJOR | ADVISOR |
|---|---|---|
| 123 | PHYSICS | EINSTEIN |
| 123 | MUSIC | MOZART |
| 456 | BIOLOGY | DARWIN |
| 789 | PHYSICS | BOHR |
| 999 | PHYSICS | EINSTEIN |

# Thank You !