Logical Database Design and the Relational Model - II

Lecture 11 - 12

CSE 303: Database Management System

Basic Definitions

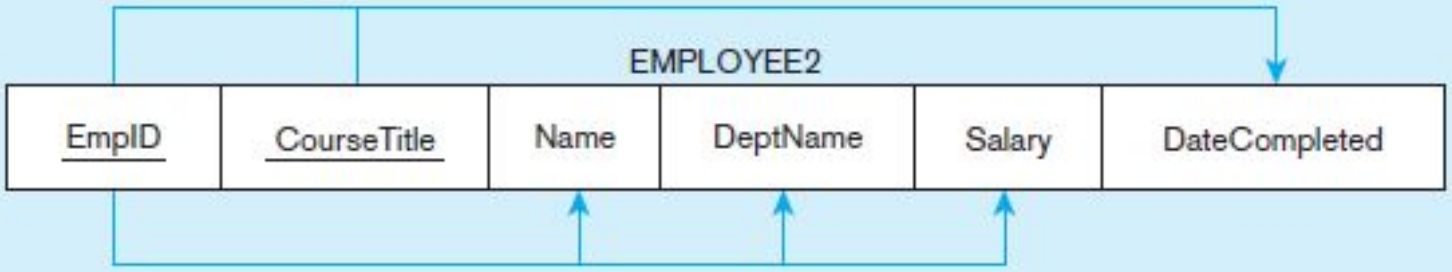
- **Normalization** The process of decomposing relations with anomalies to produce smaller, well-structured relations.
  - *During logical database design* You should use normalization concepts as a quality check for the relations that are obtained from mapping E-R diagrams.
  - *When reverse-engineering older systems* Many of the tables and user views for older systems are redundant and subject to the anomalies
- **Normal form** A state of a relation that requires that certain rules regarding relationships between attributes (or functional dependencies) are satisfied.

Basic Definitions

- **Functional dependency** A constraint between two attributes in which the value of one attribute is determined by the value of another attribute.
- **Determinant** The attribute on the left side of the arrow in a functional dependency.
  - For any relation **R**, attribute **B** is functionally dependent **(A→B)** on attribute A if, for every valid instance of A, that value of **A** uniquely determines the value of **B**.
  - A functional dependency is not a mathematical dependency: B cannot be computed from A. Rather, if you know the value of A, there can be only one value for B.
  - An attribute may be functionally dependent on a combination of two (or more) attributes rather than on a single attribute.

(a) Functional dependencies in EMPLOYEE1

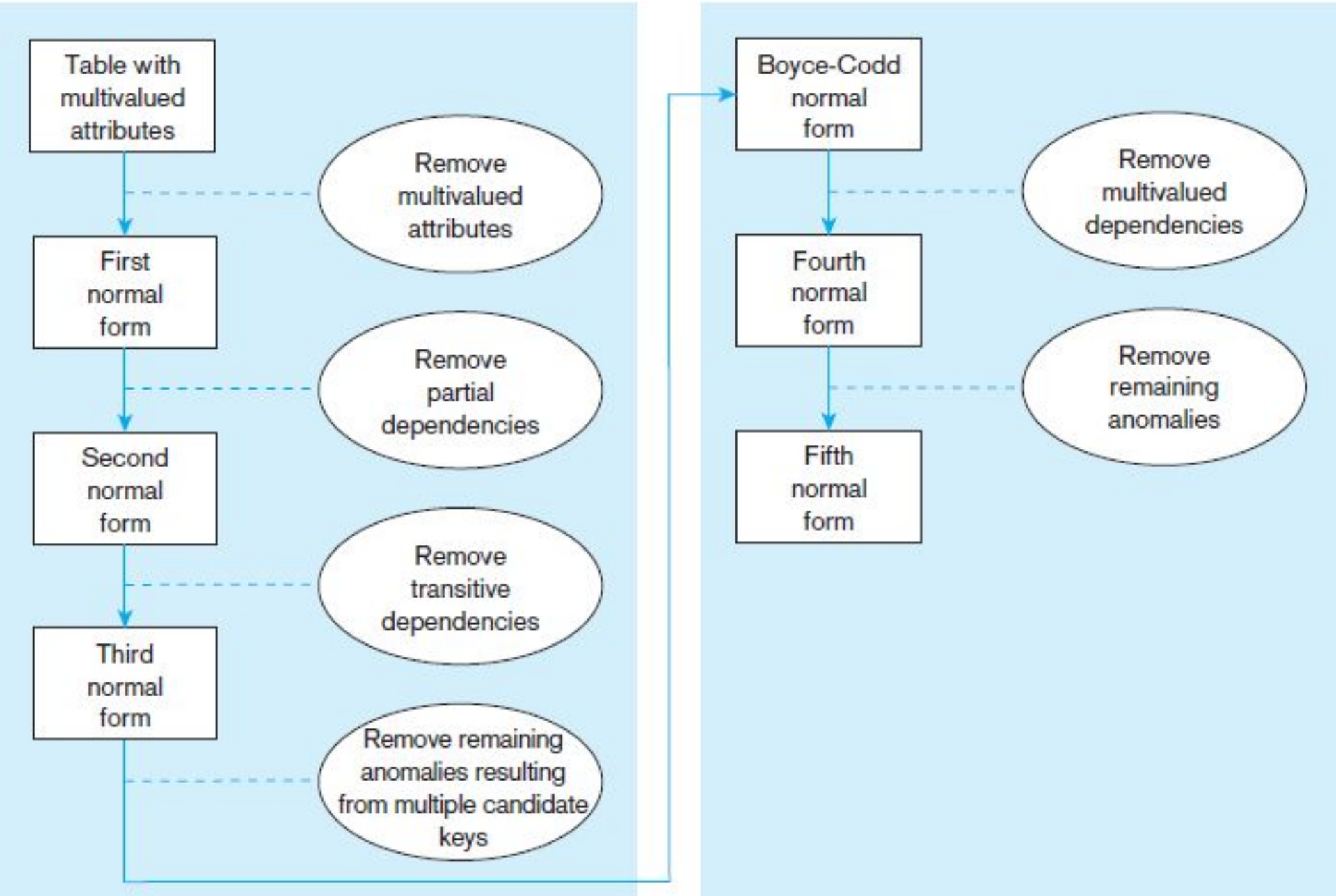
(b) Functional dependencies in EMPLOYEE2



Basic Definitions

- **Candidate key** An attribute, or combination of attributes, that uniquely identifies a row in a relation.
  - *Unique identification* For every row, the value of the key must uniquely identify that row. This property implies that each nonkey attribute is functionally dependent on that key.
  - *Nonredundancy* No attribute in the key can be deleted without destroying the property of unique identification.
- A candidate key is always a determinant, whereas a determinant may or may not be a candidate key. For example, in EMPLOYEE2, EmpID is a determinant but not a candidate key.

Steps in normalization



Step 0: Represent the View in Tabular Form

PVFC Customer Invoice

Customer ID	2	**Order ID**	1006
Customer Name	Value Furniture	**Order Date**	10/24/2010
Address	15145 S.W. 17th St. Plano TX 75022		

Product ID	Product Description	Finish	Quantity	Unit Price	Extended Price
7	Dining Table	Natural Ash	2	$800.00	$1,600.00
5	Writer's Desk	Cherry	2	$325.00	$650.00
4	Entertainment Center	Natural Maple	1	$650.00	$650.00
				Total	$2,900.00

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

Step 1: Convert to First Normal Form

- **First normal form (1NF)** A relation that has a primary key and in which there are no repeating groups.
  - There are no repeating groups in the relation (thus, there is a single fact at the intersection of each row and column of the table).
  - A primary key has been defined, which uniquely identifies each row in the relation.
- **Normalization Steps**
  1. Repeating groups can be removed by filling relevant data values into previously vacant cells.
  2. Primary key can be formed using the functional dependencies of the case study.

Step 1.1: Removing repeating groups

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

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

Step 1.2: Selection of primary key

- Primary key should determine all the attributes. Below we have the functional dependencies.
  - OrderID → OrderDate, CustomerID, CustomerName, CustomerAddress
  - CustomerID → CustomerName, CustomerAddress
  - ProductID → ProductDescription, ProductFinish, ProductStandardPrice
  - OrderID, ProductID → OrderedQuantity
- Thus OrderID and ProductID would be the primary key.

Step 1.2: Selection of primary key

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

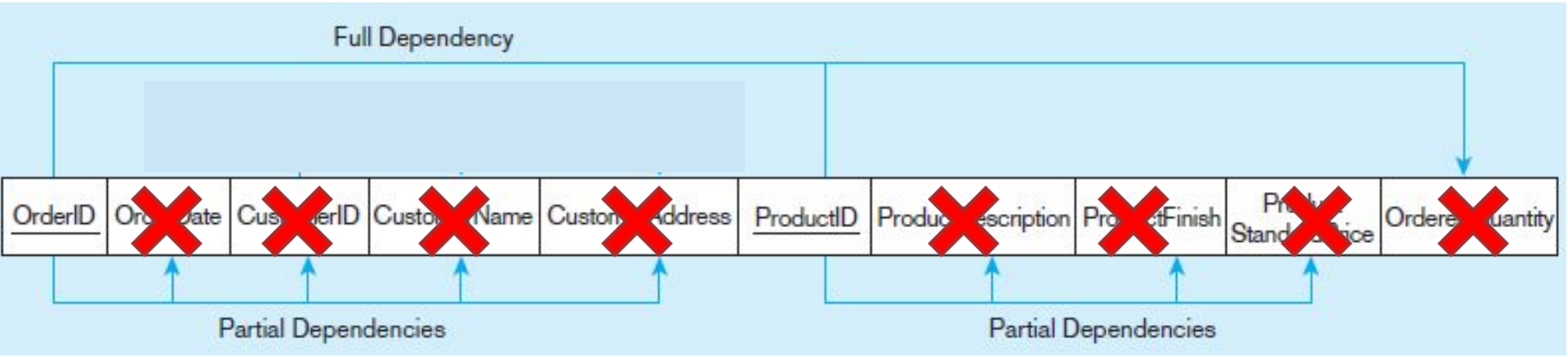
Anomalies in 1NF

- **Insertion** – can't enter a new product without having the order from a customer
- **Deletion** – if we remove a order then some product detail would be lost.
- **Modification** – changing price of a single product forces to update multiple records.
- Why do these anomalies exist?
  - Because we've combined two themes (entity types) into one relation. This results in duplication, and an unnecessary dependency between the entities

Step 2: Convert to Second Normal Form

- **Second normal form (2NF)** A relation in first normal form in which every nonkey attribute is fully functionally dependent on the primary key.
  - **Partial functional dependency** A functional dependency in which one or more nonkey attributes are functionally dependent on part (but not all) of the primary key.
- **Normalization steps**
  - 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.
  - Move the nonkey attributes that are dependent on this primary key attribute (or attributes) from the old relation to the new relation.

Step 2: Convert to Second Normal Form



Step 2: Convert to Second Normal Form

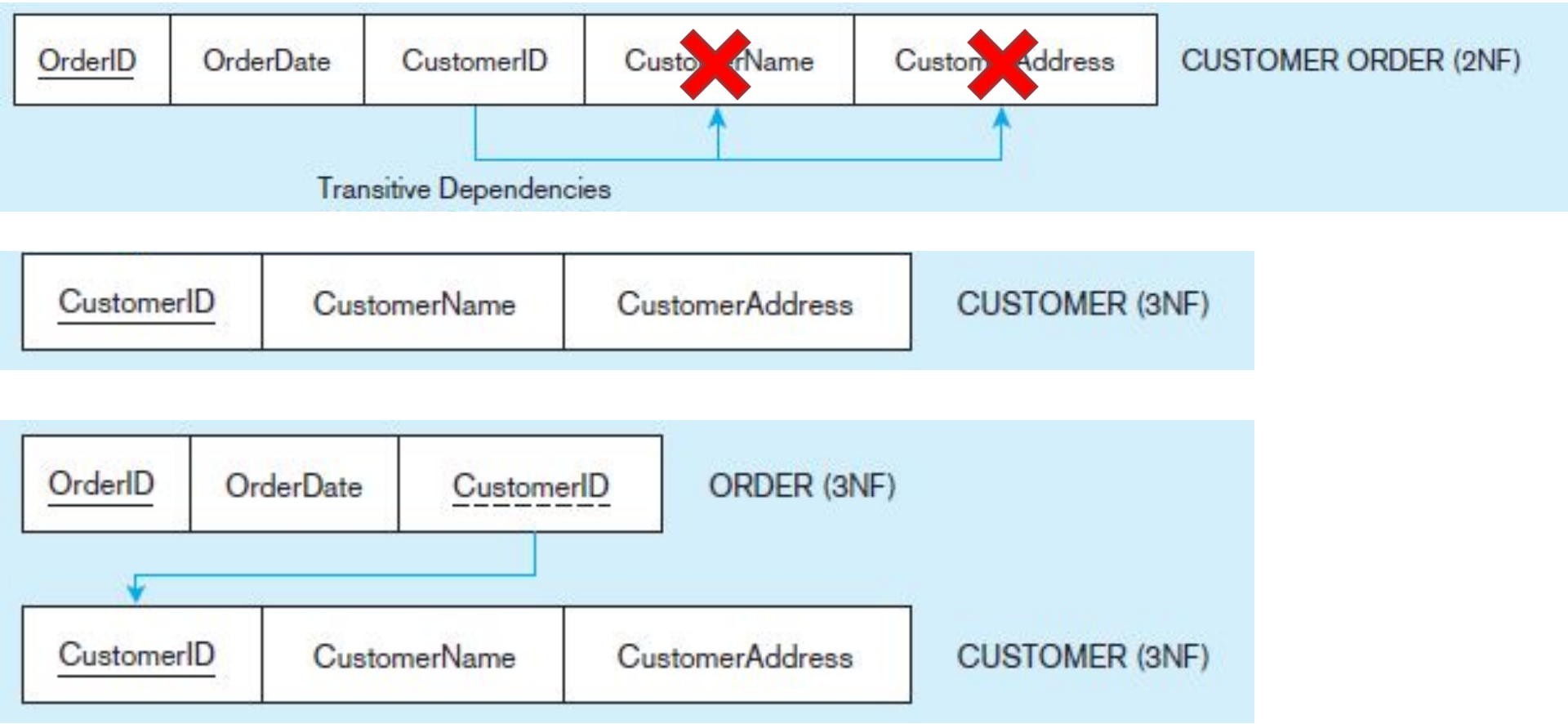
- A relation that is in first normal form will be in second normal form if any one of the following conditions applies:
  - The primary key consists of only one attribute. By definition, there cannot be a partial dependency in such a relation.
  - No nonkey attributes exist in the relation (thus all of the attributes in the relation are components of the primary key). There are no functional dependencies in such a relation.
  - Every nonkey attribute is functionally dependent on the full set of primary key attributes (e.g., the attribute OrderedQuantity in the ORDER LINE relation).

Step 3: Convert to Third Normal Form

- **Third normal form (3NF)** A relation that is in second normal form and has no transitive dependencies.
  - **Transitive dependency** A functional dependency between the primary key and one or more nonkey attributes that are dependent on the primary key via another non-key attribute.
- Steps of removing Transitive dependencies.

1. For each nonkey attribute (or set of attributes) that is a determinant in a relation, create a new relation. That attribute (or set of attributes) 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. Leave the attribute that serves as a primary key in the new relation in the old relation to serve as a foreign key that allows you to associate the two relations.

Step 3: Convert to Third Normal Form



Example (non-normalized relation)

ID	Name	Add	Dept	Dept_Head	Advisor	C_No	C_Name	C_Credit	GPA
1	Ihtesham	Baridhara	CSE	M. A. Amin	Mr. ABC	C401	DB Mgt	3	3.0
						C203	Data struct	3	2.0
						C212	Micro Proc.	3	3.3
2	Muntaser	Uttara	CSE	M. A. Amin	Mr. DEF	C401	DB Mgt	3	3.0
						C203	Data struct	3	2.0
						C212	Micro Proc.	3	3.3
3	Karim	Malibag	CSE	M. A. Amin	Mr. XYZ	C203	Data struct	3	2.0
						C212	Micro Proc.	3	3.3

Functional Dependencies:

ID → Name, Add, Dept, Dept_Head, Advisor
Dept → Dept_Head
C_No → C_Name, C_Credit
ID, C_No → GPA

1NF

ID	Name	Add	Dept	Dept_Head	Advisor	C_No	C_Name	C_Credit	GPA
1	Ihtesham	Baridhara	CSE	M. A. Amin	Mr. ABC	C401	DB Mgt	3	3.0
1	Ihtesham	Baridhara	CSE	M. A. Amin	Mr. ABC	C203	Data struct	3	2.0
1	Ihtesham	Baridhara	CSE	M. A. Amin	Mr. ABC	C212	Micro Proc.	3	3.3
2	Muntaser	Uttara	CSE	M. A. Amin	Mr. DEF	C401	DB Mgt	3	3.0
2	Muntaser	Uttara	CSE	M. A. Amin	Mr. DEF	C203	Data struct	3	2.0
2	Muntaser	Uttara	CSE	M. A. Amin	Mr. DEF	C212	Micro Proc.	3	3.3
3	Karim	Malibag	CSE	M. A. Amin	Mr. XYZ	C203	Data struct	3	2.0
3	Karim	Malibag	CSE	M. A. Amin	Mr. XYZ	C212	Micro Proc.	3	3.3

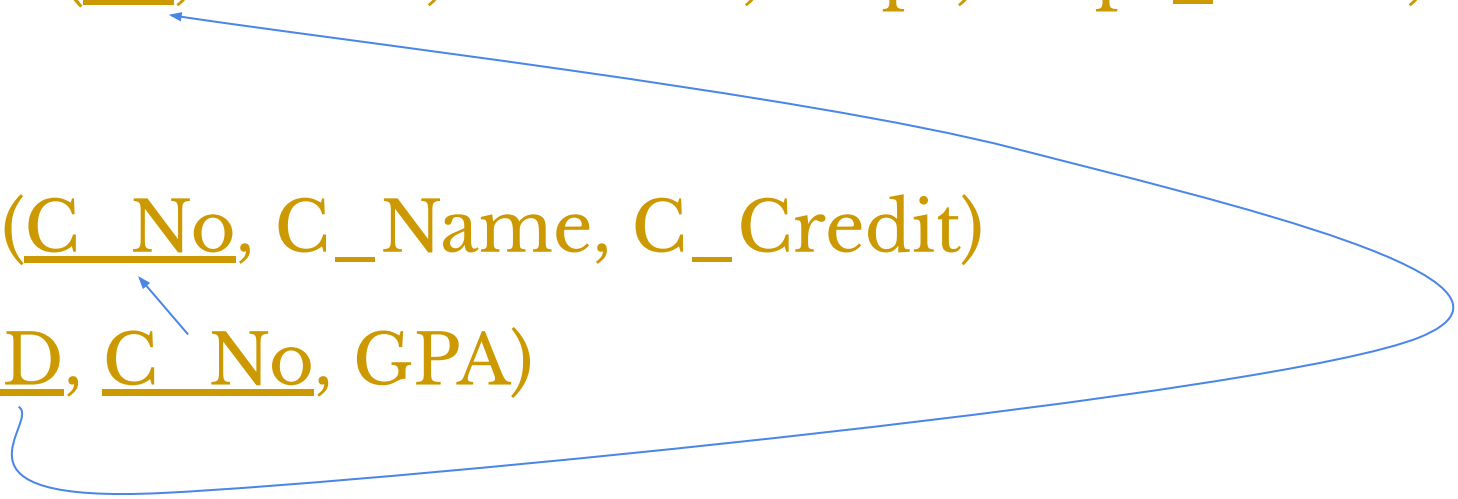
ID → Name, Add, Dept, Dept_Head, Advisor

C_No → C_Name, C_Credit

ID, C_No → GPA

2NF

- STUDENT(ID, Name, Address, Dept, Dept_Head, Advisor, C_No, C_Name, C_Credit, GPA)
- STUDENT(ID, Name, Address, Dept, Dept_Head, Advisor)
- COURSE(C_No, C_Name, C_Credit)
- GRADE(ID, C_No, GPA)



3NF

- STUDENT(ID, Name, Address, Dept, Dept_Head, Advisor)
- STUDENT(ID, Name, Address, Dept, Advisor)
- DEPT(Dept, Dept_Head)

Dept → Dept_Head

Thank You