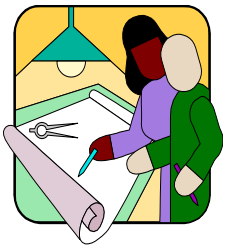
Chapter 4
Normalization

Data Normalization

- Formal process of decomposing relations with anomalies to produce smaller, ***well-structured*** and ***stable*** relations
- Primarily a tool to validate and improve a logical design so that it satisfies certain constraints that ***avoid unnecessary duplication of data***



Well-Structured Relations

- A relation that contains minimal data redundancy and allows users to insert, delete, and update rows without causing data inconsistencies
- Goal is to avoid (*minimize)* anomalies
  - **Insertion Anomaly** – adding new rows forces user to create duplicate data
  - **Deletion Anomaly** – deleting a row may cause loss of other data representing completely different facts
  - **Modification Anomaly** – changing data in a row forces changes to other rows because of duplication

General rule of thumb: a table should not pertain to more than one entity type

3

Example – Figure 4.2b

EMPLOYEE2

EmpID	Name	DeptName	Salary	CourseTitle	DateCompleted
100	Margaret Simpson	Marketing	48,000	SPSS	6/19/201X
100	Margaret Simpson	Marketing	48,000	Surveys	10/7/201X
140	Alan Beeton	Accounting	52,000	Tax Acc	12/8/201X
110	Chris Lucero	Info Systems	43,000	Visual Basic	1/12/201X
110	Chris Lucero	Info Systems	43,000	C++	4/22/201X
190	Lorenzo Davis	Finance	55,000		
150	Susan Martin	Marketing	42,000	SPSS	6/19/201X
150	Susan Martin	Marketing	42,000	Java	8/12/201X

Question – Is this a relation?

Answer – Yes: unique rows and no multivalued attributes

Question – What's the primary key?

Answer – Composite: EmpID, CourseTitle

4

Anomalies in this Table

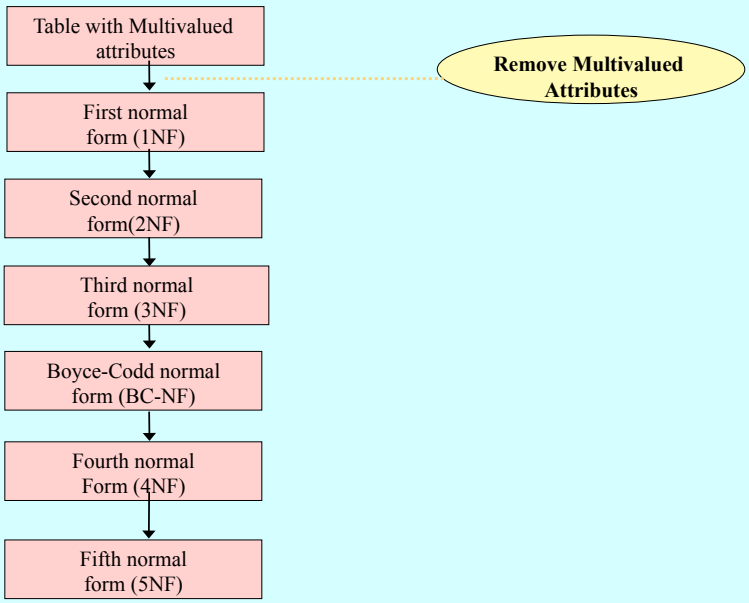
- **Insertion** – can't enter a new employee without having the employee take a class
- **Deletion** – if we remove employee 140, we lose information about the existence of a Tax Acc class
- **Modification** – giving a salary increase to employee 100 forces us to update multiple records

Why do these anomalies exist?

Because there are two themes (entity types – what are they?) in this one relation (two themes, entity types, were combined). This results in duplication, and an unnecessary dependency between the entities

5

Figure: 4-22 Steps in Normalization



6

First Normal Form (1NF)

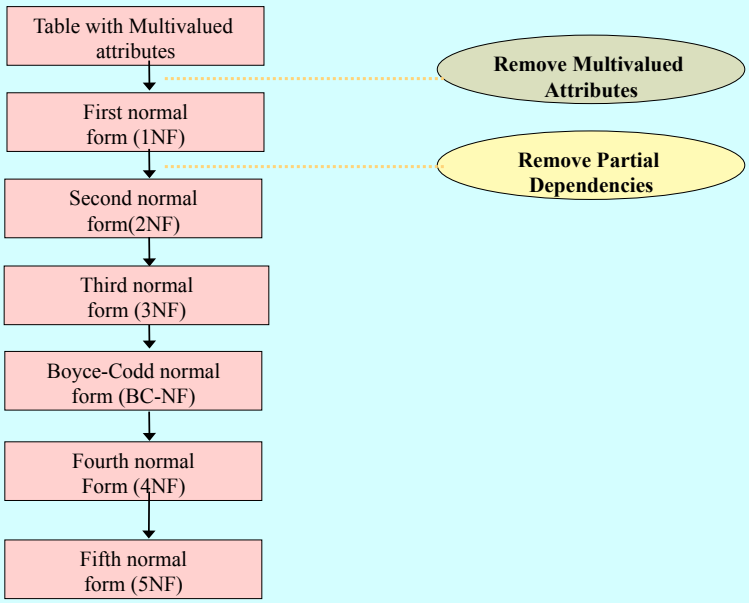
- Only atomic attributes (simple, single-value)
- A primary key has been identified
- ***Every relation* is in 1NF** by definition
- 1NF example:

Student

StudentId	**StuName**	**CourseId**	**CourseName**	**Grade**
100	Mike	112	C++	A
100	Mike	111	Java	B
101	Susan	222	Database	A
140	Lorenzo	224	Graphics	B

7

Figure: 4-22 Steps in Normalization



8

Functional Dependencies

- **Functional Dependency**: The value of one attribute (the *determinant*) determines the value of another attribute.
  - A→B reads "Attribute B is functionally dependent on A"
  - A→B means if two rows have same value of A they necessarily have same value of B
  - FDs are determined by **semantics: You can't** say that a FD exists just by looking at data. But can say whether it does not exist by looking at data.

9

Quick Check

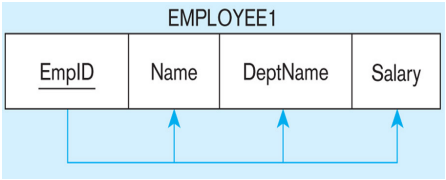
Id	Name	Gender	Age
1	Orlando	Male	35
2	John	Male	35
3	Jane	Female	31
4	Jane	Female	30

- Id → Name?
- Age → Gender?
- Name → Id?
- Name, Age → Id?

10

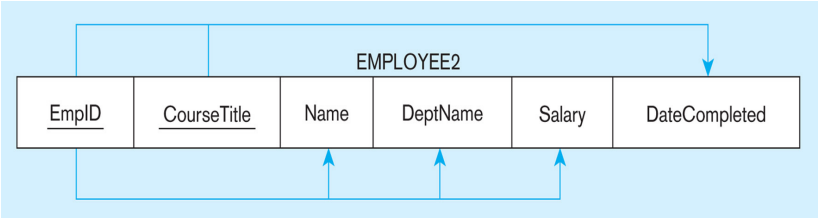
Functional Dependencies and Keys

- **Functional Dependency**: The value of one attribute (the *determinant*) determines the value of another attribute.
- **Candidate Key**
  - Attribute that uniquely identifies a row in a relation
  - Could be a combination of (*non-redundant*) attributes
  - Each non-key field is functionally dependent on every candidate key



12

Figure 4-23: Representing Functional Dependencies (cont.)



EmpID → _____

EmpID, CourseTitle → _____

EmpID	Name	DeptName	Salary	CourseTitle	DateCompleted
100	Margaret Simpson	Marketing	48,000	SPSS	6/19/201X
100	Margaret Simpson	Marketing	48,000	Surveys	10/7/201X

13

Practice Exercise #7, page #193

TABLE 4-3 Sample Data for Parts and Vendors

Part No	Description	Vendor Name	Address	Unit Cost
1234	Logic chip	Fast Chips	Cupertino	10.00
		Smart Chips	Phoenix	8.00
5678	Memory chip	Fast Chips	Cupertino	3.00
		Quality Chips	Austin	2.00
		Smart Chips	Phoenix	5.00

1. Convert this table to a relation (named PART SUPPLIER) in 1NF
2. Draw a relational schema for PART SUPPLIER and show the functional dependencies. Identify a candidate key.
3. Identify each of the following: an insert anomaly, a delete anomaly, and a modification anomaly.

14

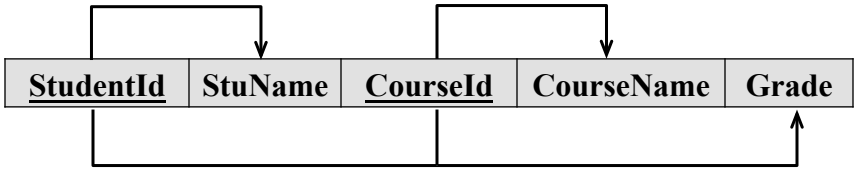
15

Second Normal Form (2NF)

- 1NF PLUS ***every non-key attribute is fully functionally dependent on the ENTIRE primary key***
  - Every non-key attribute must be defined by the entire key, not by only part of the key
  - No partial functional dependencies

16

Functional Dependencies in Student



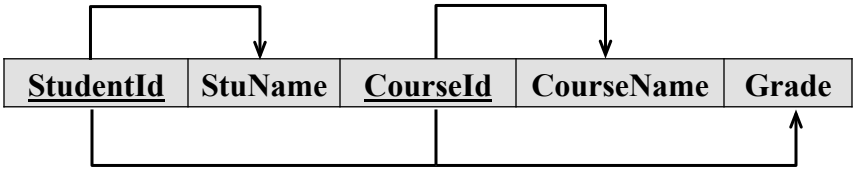
Can represent FDs with arrows as above, or

- StudentId → StuName,
- CourseId → CourseName
- StudentId,CourseId → Grade (and StuName, CourseName)

Any partial FDs ?

17

Functional Dependencies in Student



Can represent FDs with arrows as above, or

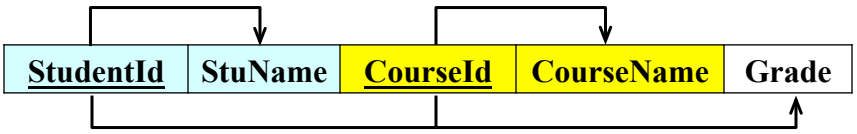
- StudentId →StuName,
- CourseId → CourseName
- StudentId,CourseId → Grade (and StuName, CourseName)

Therefore, NOT in 2nd Normal Form!!

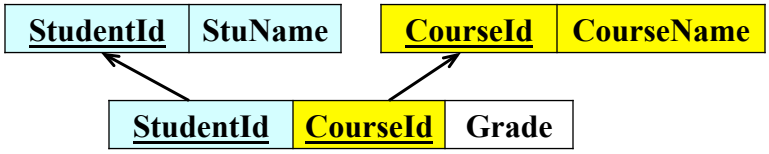
18

2NF: Normalizing

- How do we convert the partial dependencies into normal ones ? By breaking into more tables.



- Becomes … (notice above arrows mean functional dependency, below they mean FK constraints)



19

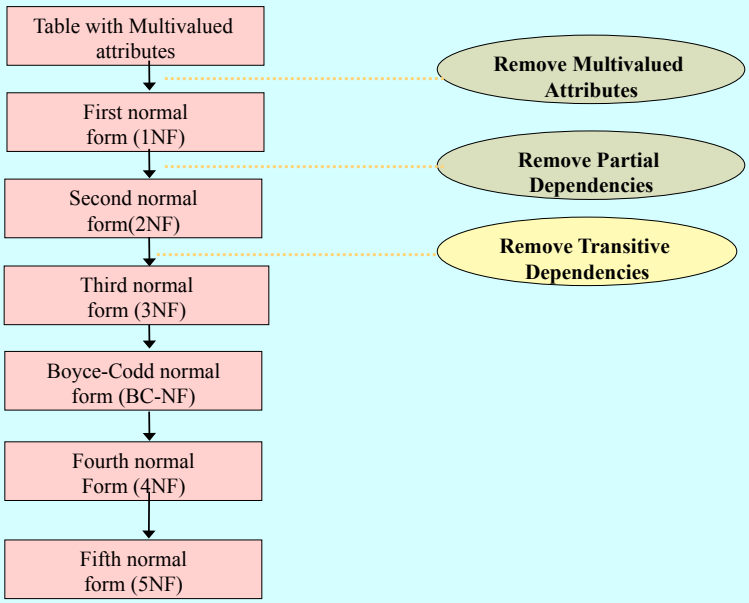
You Try …

SeriesId	EpisodeId	SeriesTitle	EpisodeTitle	AiringDate

- List all FDs
- Eliminate partial FDs, if any

20

Figure: 4-22 Steps in Normalization



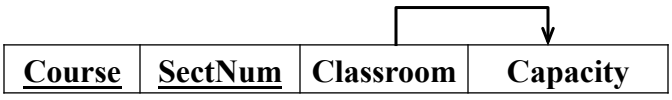
22

Third Normal Form

- 2NF and no transitive dependencies
- A *transitive dependency* is when a non-key attribute depends on another non-key attribute
- Note: This is called transitive, because the primary key is a determinant for another attribute, which in turn is a determinant for a third attribute

23

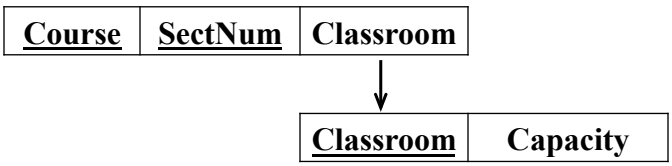
3NF Example



- Classroom → Capacity TRANSITIVE
- Any partial FDs? NO
- Any transitive FDs? YES !
  - How do we eliminate it?
  - By breaking into its own table

25

3NF Normalization



27

You Try …

StudentId	ProgramId	StudentName	ProgramName

- Partial FDs? Eliminate, if any.
- Transitive FDs? Eliminate, if any.

28

Practice Exercise #15, page #196

TABLE 4-8 Shipment Relation

Shipment#	Origin	Destination	Distance
409	Seattle	Denver	1,537
618	Chicago	Dallas	1,058
723	Boston	Atlanta	1,214
824	Denver	Los Angeles	975
629	Seattle	Denver	1,537

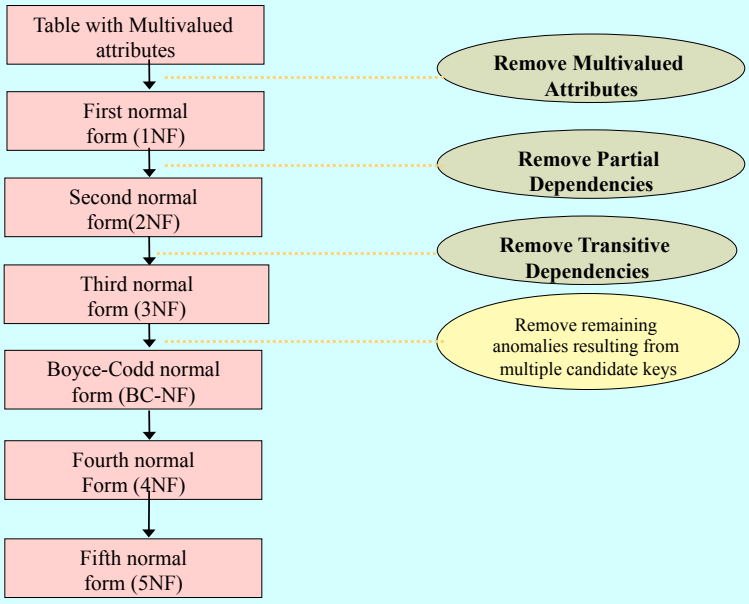
Insertion anomaly?
Deletion anomaly?
Modification anomaly?

1. Develop a diagram that shows the functional dependencies in the SHIPMENT relation.
2. In what normal form is SHIPMENT? Why?
3. Convert SHIPMENT to 3NF if necessary. Show the resulting table(s) with the sample data presented in SHIPMENT.

30

31

Figure: 4-22 Steps in Normalization



33

Further Normalization

- Boyce-Codd Normal form (BCNF)
  - Slight difference with 3NF
  - To be in 3NF but not in BNF, needs two composite candidate keys, with one attribute of one key depending on one attribute of the other
  - Not very common ☺
  - If a table contains only one candidate key, the 3NF and the BCNF are equivalent.
- Fourth Normal Form (4NF)
  - To break it, need to have multivalued dependencies, a generalization of functional dependencies
- Usually, if you're in 3NF you're in BCNF, 4NF, …

34

BCNF Example

- **Assume that**
  - For each subject, each student is taught by one Instructor
  - Each Instructor teaches only one subject
  - Each subject is taught by several Instructors

Course, Student → Instructor

Instructor → Course

Course	Instructor	Student
CS 121	Dr. A. James	Bill Payne
CS 121	Dr. A. James	Tony Perez
CS 121	Dr. A. James	James Atkinson
CS 121	Dr. A. James	Linda Lee

Course	Instructor	Student
CS 141	Dr. T. Watson	Linda Lee
CS 141	Dr. T. Watson	Judith San
CS 141	Dr. T. Watson	Bill Jones
CS 141	Dr. P. Hold	Bill Payne
CS 141	Dr. P. Hold	A. White

Course	Instructor	Student
CS 101	Dr. M. Jones	Linda Lee
CS 101	Dr. M. Jones	Tony Perez
CS 101	Dr. M. Jones	Bill Payne

BCNF: Decompose into (Instructor, Course) and (Student, Instructor)

35

BCNF

- **Boyce-Codd normal form (BCNF)**

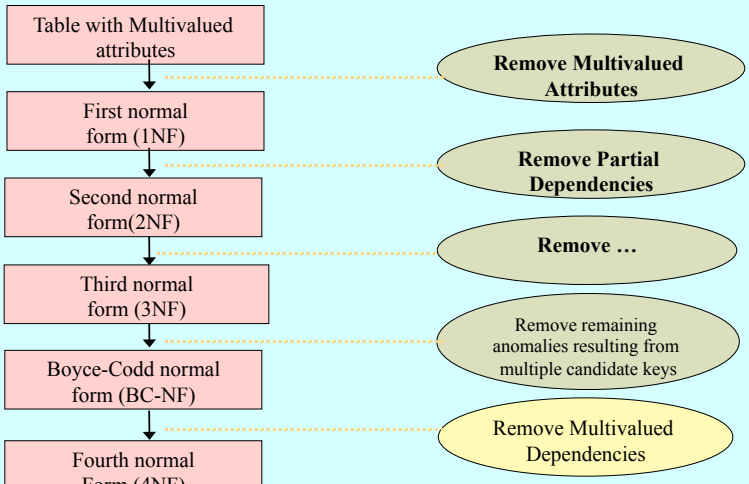
  A relation is in BCNF, if and only if, every determinant is a candidate key.

- **The difference between 3NF and BCNF** is that for a functional dependency A → B, 3NF allows this dependency in a relation if B is a primary-key attribute and A is not a candidate key,

  whereas BCNF insists that for this dependency to remain in a relation, A must be a candidate key.

36

Figure: 4-22 Steps in Normalization



37

4NF

- **A multi-valued dependency exists when**
  - There are at least 3 attributes A, B, C in a relation and
  - For each value of A there is a well defined set of values for B, and a well defined set of values for C,
  - But the set of values for B is independent on the set of values for C

- **4NF = 3NF with no multi-valued dependency**

38

4NF Example

- **Assume that**
  - Each subject is taught by many Instructors
  - The same books are used in many subjects
  - Each Instructor uses a different book

Course, Instructor → Text

Course, Text → Instructor

Course	Instructor	Text
CS 121	Dr. A. James	Int to Com Science
CS 121	Dr. P. Hold	Comp Scien Int

Course	Instructor	Text
CS 141	Dr. T. Watson	Int to Com Science
CS 141	Dr. P. Hold	Comp Scien Int
CS 101	Dr. M. Jones	COMP SCIEN

4NF: Decompose into (Course, Instructor) and (Course, Text)

39

Textbook Example

40

The Normalization Example in the Text Book
Figure 4-24 INVOICE (Pine Valley Furniture Company)

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

41

Figure 4-25 INVOICE Data
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. WHY?

42

Figure 4-26 INVOICE relation (1NF)
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

Note: this is relation, but not a well-structured one. WHY?

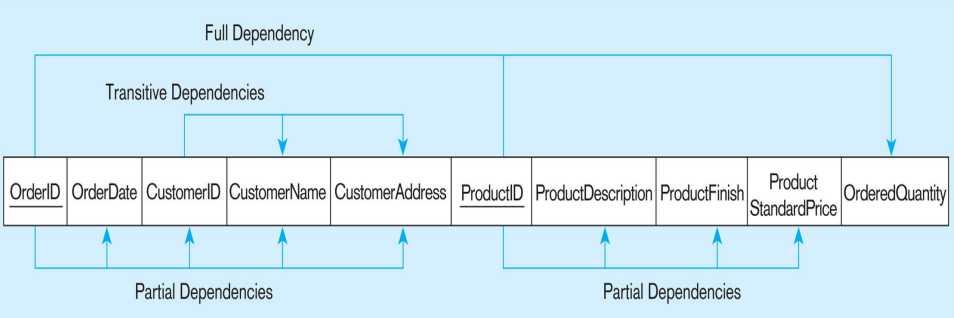
43

Anomalies in this 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 and price
- **Update**–changing the price of product ID 4 requires update in several records

44

Figure 4-27 Functional dependency diagram for INVOICE

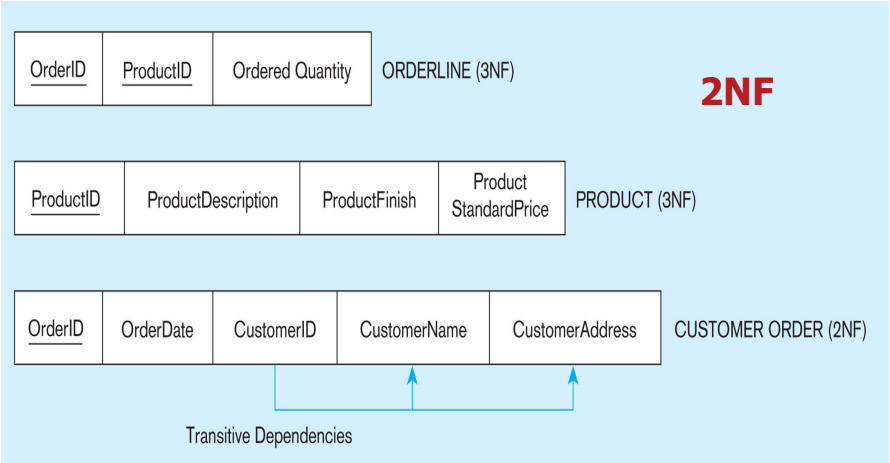


Order_ID ➔ Order_Date, Customer_ID, Customer_Name, Customer_Address
Customer_ID ➔ Customer_Name, Customer_Address
Product_ID ➔ Product_Description, Product_Finish, Unit_Price
Order_ID, Product_ID ➔ Order_Quantity

Therefore, NOT in 2nd Normal Form

45

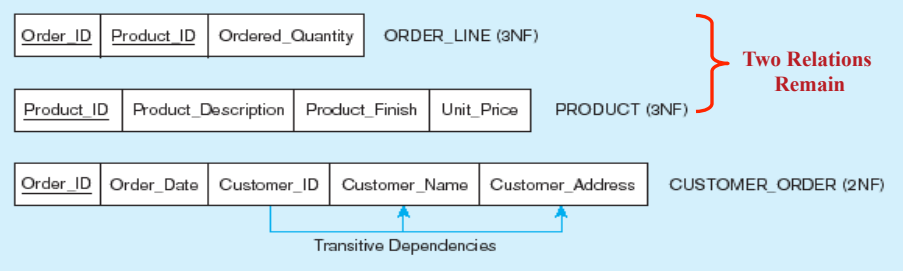
Figure 4-28 Partial Dependencies were Removed (2NF)

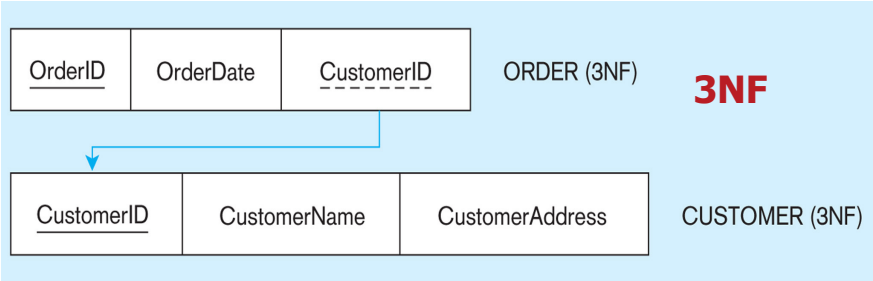


Partial dependencies are removed, but there are still transitive dependencies

46

Figure 4-29 Transitive Dependencies were Removed (3NF)





47

You Try …

TABLE 4-6 Parking Tickets at Millennium College

Parking Ticket Table

St ID	L Name	F Name	Phone No	St Lic	Lic No	Ticket #	Date	Code	Fine
38249	Brown	Thomas	111-7804	FL	BRY 123	15634	10/17/12	2	$25
						16017	11/13/12	1	$15
82453	Green	Sally	391-1689	AL	TRE 141	14987	10/05/12	3	$100
						16293	11/18/12	1	$15
						17892	12/13/12	2	$25

1. Convert to 1NF Relation
2. Draw dependency diagram showing all functional dependencies
3. Identify anomalies
4. Convert to 3NF Relations
5. Develop EER Diagram with appropriate cardinalities

48

49

Logical Database Design

You have just learned and completed one of the most important concepts and theories, **integrity constraints** and **normalization**, for developing a quality of database.

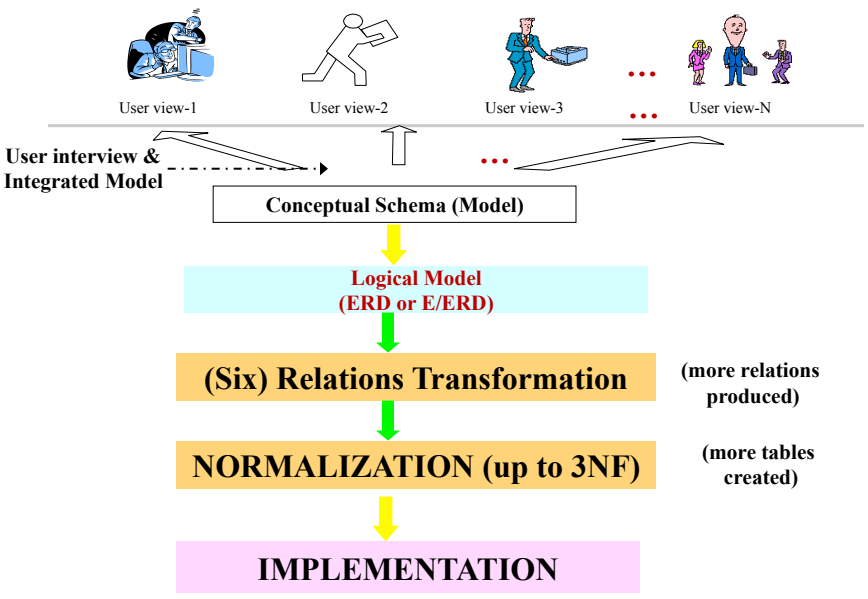
50

After learning one of most important database concepts and theories...

WHAT'S NEXT ?

51

Steps of Database Development



52