DATABASE - 1

3RD CLASS

COMPUTER SCIENCE DEPARTMENT

6th Lecture – Introduction to Normalization

Sunday 27th of Oct. 2024

LECTURER :

DR. RAYAN YOUSIF ALKHAYAT

Normalization :

is a formal process for deciding which attributes should be grouped together in a relation.

Before we preceding with physical design, we need a method to validate the logical design to this point.

Normalization is a tool to validate and improve a logical design.

So that it satisfies certain constrains that avoid unnecessary duplication of data.

We have presented an intuitive discussion of well-structured relations; however, we need a formal definitions of such relations , together with a process for designing them.

Normalization is a process of decomposing relations with anomalies to produce smaller, well-structured relations.

For example, we use the principles of normalization to convert the EMPLOYEE2 table with its redundancy to (EMPLOYEE1) .

Steps in Normalization:

- Normalization can be accomplished and understood in stages, each that corresponds to a normal form .
- **A Normal Form** is a state of relations that results from applying simple rules regarding functional dependencies (or relationships between attributes) to that relation .

we describe these rules briefly in this section and then detail in the following section .

- First Normal Form .
- Second Normal Form.
- Third Normal Form.
- Boyce/Codd Normal .
- Fourth Normal Form.
- Fifth Normal Form.

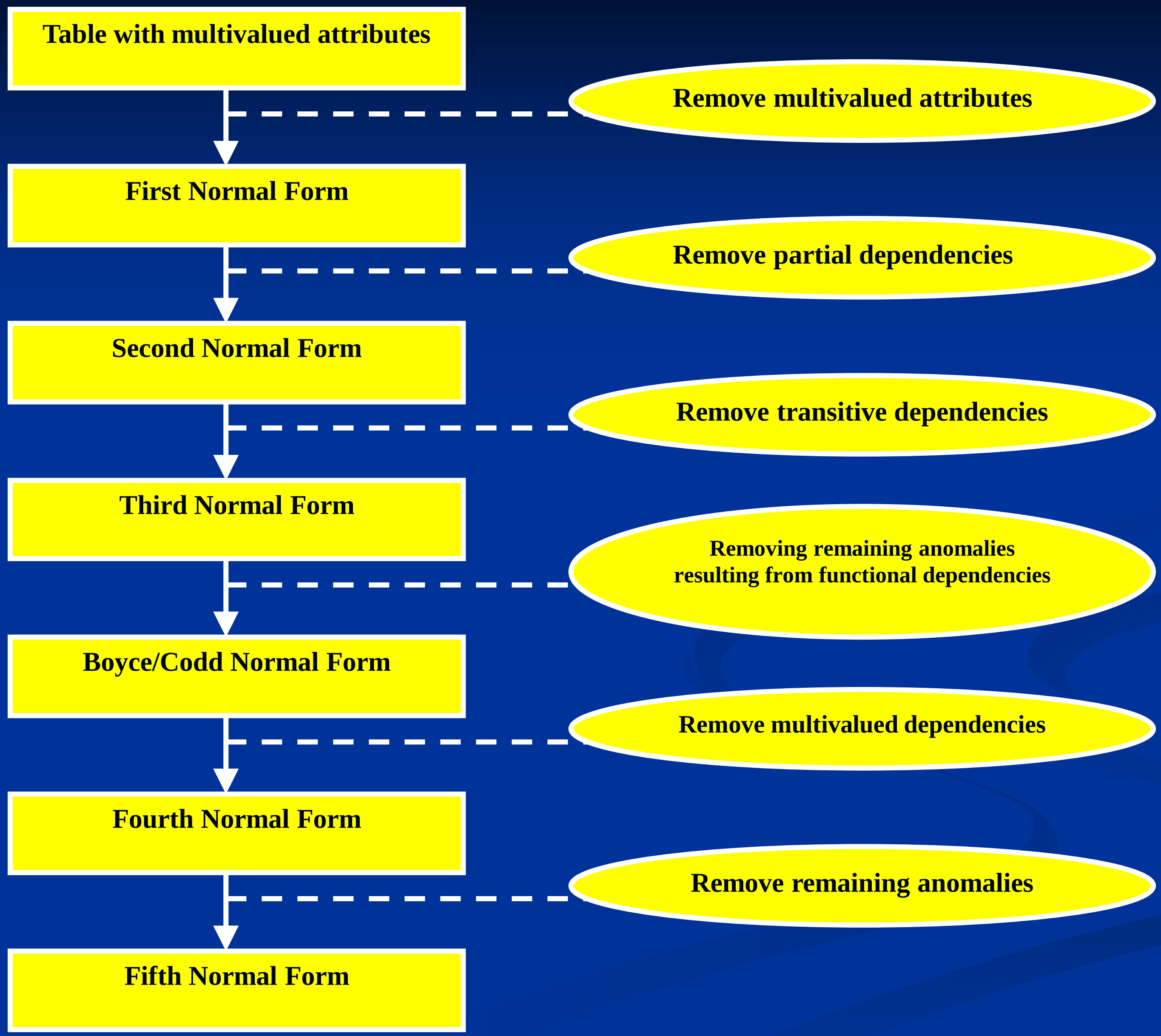


Table with multivalued attributes
Remove multivalued attributes
First Normal Form
Remove partial dependencies
Second Normal Form
Remove transitive dependencies
Third Normal Form
Removing remaining anomalies resulting from functional dependencies
Boyce/Codd Normal Form
Remove multivalued dependencies
Fourth Normal Form
Remove remaining anomalies
Fifth Normal Form

Tables with multivalued attributes

Tables with no multivalued attributes

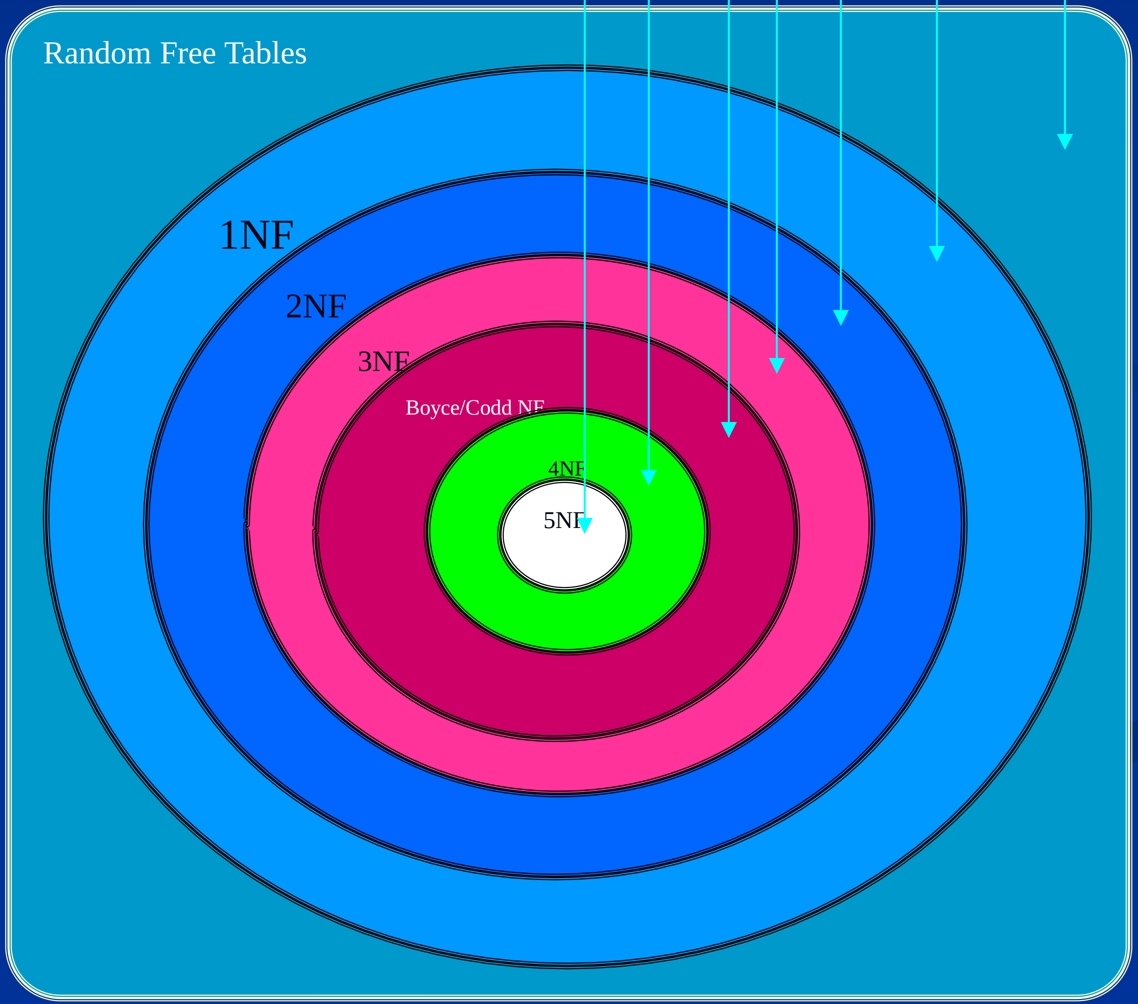
Tables with no partial dependencies

Tables with no transitivedependencies

Tables after removing anomalies resulting from functional dependencies

Tables after removing multivalued dependencies

Tables with no anomalies



Functional Dependencies and Keys:

Normalization is based on the analysis of *Functional Dependencies*

- **Functional Dependency: is a constraint between two attributes or two sets of attributes.**
- **For any relation R , attribute B is functionally dependent on attribute A if , for every valid Instance of A , that value of A uniquely determines the value of B . The functional dependency of B on A represented by an arrow , as follows : A→ B . An attribute may be functionally dependent on two (or more) attributes , rather than on a single attribute .**

For Example, consider the relation
EMP_COURSE (Emp_ID, Course_Name, Date_Completed)
shown in figure below :

EMP_COURSE

Emp_ID	Course_Title	Date_Completed
100	SPSS	6/19/2016
100	Cisco	10/7/2016
140	CNC	12/8/2015
110	Tax Acc	1/12/2016
110	SPSS	4/22/2015
150	SPSS	6/16/2015
150	Java	8/12/2016

We represent the functional dependencies in this relation as follows :
Emp_ID, Course_Name→Date_Completed

The functional dependencies in this statement implies that the date completed of a course is completely determined by the identity of the employee and the name of the course .

Common examples of functional dependences are the following:

- SSN → Name, Address, Birthdate A person's name, address, and birthdate are functionally dependent on the person Social Security Number.
- VIN → Make, Model, Color The make, model, and color of a vehicle are functionally dependent on the Vehicle Identification Number.
- ISBN → Title, First_Author_Name, The title of a book and the name of the first author are functionally dependent of the book's International Standard Book Number(ISBN).

- **Determinates**: the attribute on the left-hand side of the arrow in a functional dependency is called a determinant. SSN,VIN ,and ISBN are determinates (respectively) in the preceding three examples .

In the EMP_COURSE relation, the combination of the Emp_ID and Course_Name is a determinate.

Candidate Keys: A candidate key is an attribute or combination of attributes , that uniquely identifies a row in a relation .

A candidate key must satisfy the following properties ,which are subset of the six properties of a primary key :

1. ***Unique identification***: for every row ,the value of the key must uniquely identify that row .*This property implies that each Nonekey attribute is functionally dependent on that key.*

2. ***Nonredundancy*** : No attribute in the key can be deleted without destroying the property of unique identification.

By applying the preceding definition to identify candidate keys in two of the relations described in this section.

The EMPLOYEE1 relation has the following schema :

- EMPLOYEE1 (Emp_ID,Name,Dept_Name,Salary).
- Emp_ID is the only Determinant in this relation .
- The other attributes are functionally dependent on Emp_ID .

Therefore, Emp_ID is a candidate key and since there are no other candidate keys it is also is the primary key.

- For the relation EMPLOYEE2 notice that

(unlike EMPLOYEE1) , Emp_ID does not uniquely identify a row in the relation. For Example, there are two rows in the table for Emp_ID 100.

There are two functional dependences in this relation.

Emp_ID → Name,Dept_Name,Salary

Emp_ID,Course_Title→ Date_Completed

- The functional dependences indicate that the combination of Emp_ID and Course_Title is the only candidate key(and therefore the primary key) for EMPLOYEE2 in other words the primary key of EMPLOYEE2 is a composite key . Neither Emp_ID nor Course_Title uniquely identifies a row in this relation ,and therefore by property 1 cannot by itself be a candidate key .
  Examine the data in table below to verify that the combination of Emp_ID and Course_Title Does uniquely identify each row of EMPLOYE2.

EMPLOYEE2

Emp_ID	Name	Dept_Name	Salary	Course_Title	Date_Completed
100	Mohammad zaki	Marketing	530.000	SPSS	6/19/2016
100	Mohammad zaki	Marketing	530.000	Cisco	10/7/2016
140	Sufyan salim	Accounting	480.000	CNC	12/8/2015
110	Sinan zohair	Info systems	225.000	Tax Acc	1/12/2016
110	Sinan zohair	Info systems	225.000	SPSS	4/22/2015
190	Muthanna Mohammad	Finance	400.000		
150	Mahmmod khalid	Marketing	75.000	SPSS	6/16/2015
150	Mahmmod khalid	Marketing	75.000	Java	8/12/2016

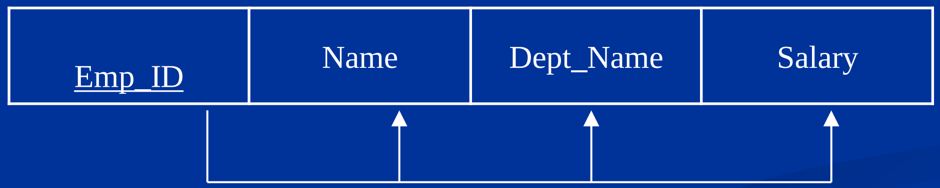
We represent the functional dependences for a relation using the notation :

The horizontal line portrays the functional dependences. A vertical line drops from the primary key and connects to this line.

Vertical arrows then point to each of the Nonekey attributes that are functionally dependent on the primary key.

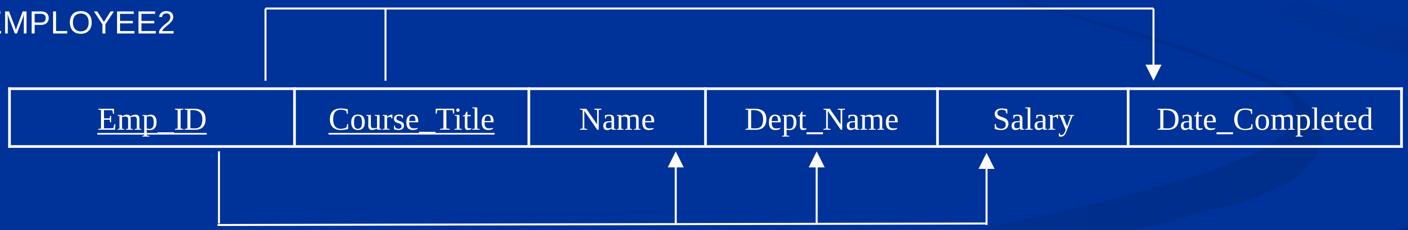
- We represent the functional dependencies in this relation in figure below . Notice that Date Completed is the only attribute that is functionally dependent on the full primary key consisting of the attributes Emp_ID and Course_Title

EMPLOYEE1



Part(a) Functional Dependencies in EMPLOYEE1

EMPLOYEE2



Part(b) Functional Dependencies in EMPLOYEE2

- We can summarize the relationship between the determinants and candidate keys as follows .
- **A candidate key** is always a determinant, while a determinant may or may not be a candidate key . For example in EMPLOYEE2 ,Emp_ID is a determinant but not a candidate key .
- **A candidate key** is a determinant that uniquely identifies the remaining (Nonekey) attributes in a relation.
- **A determinant** may be candidate key (such as Emp_Id in EMPLOYEE1), part of a composite candidate key such as Emp_ID in EMPLOYEE2)or a Nonekey attribute . We will describe examples of this shortly

THE BASIC NORMAL FORMS:

- **First Normal Form :**

  A relation is in first normal form (1NF) if it contains no multivalued attributes

THE BASIC NORMAL FORMS:

Second Normal Form :

A relation is in second normal form (2NF) if it is in first normal form and every Nonekey attribute is fully functionally dependent on the primary key . thus no Nonekey attribute is functionally dependent on part (but not all) of the primary key : A relation that is in first normal form will be in second normal form if any one of the following conditions applies :

1. The primary key consists of only one attribute .
2. No Nonekey attributes exist in the relation (thus all of the attributes in the relation are components of the primary key)
3. Every Nonekey attribute is functionally dependent of the full set of primary key attributes.

- EMPLOYEE2 is an example of a relation that is not in second normal form .
- The primary key for this relation is the composite key Emp_ID ,Course_Title.
- Therfore the Nonekey attribute Name ,dept_Name, Salary are functionally dependent on the part of the primary key (Emp_ID) but not on Course_Title .

- **A partial functional dependency** is a functional dependency in which one or more nonkey attributes (such as Name) are functionally dependent on part (but not all) of the primary key .
- The partial functional dependency in EMPLOYEE2 creates redundancy in that relation.
- Which results in anomalies when the table is updated as we noted before.

- To convert a relation to second normal form ,we decompose the relation into new relations that satisfy one (or more) of the conditions described above .
- EMPLOYEE2 is decomposed into the following two relations:

EMPLOYEE1(<u>Emp_id</u>,Name,<u>Dept_Name</u>,Salary).
This relation satisfies condition 1 above and is in second normal form.

EMPLOYEE1

<u>Emp_ID</u>	Name	Dept_Name	Salary
100	Mohammad zaki	Marketing	530.000
140	Sufyan salim	Accounting	480.000
110	Sinan zohair	Info systems	225.000
190	Muthanna Mohammad	Finance	400.000
150	Mahmmod khalid	Marketing	75.000

EMP_COURSE<u>(Emp_ID, Course_Title, </u>Date_Completed)

This relation satisfies condition 3 above and also in second normal form

EMP_COURSE

<u>Emp_ID</u>	<u>Course_Title</u>	Date_Completed
100	SPSS	6/19/2016
100	Cisco	10/7/2016
140	CNC	12/8/2015
110	Tax Acc	1/12/2016
110	SPSS	4/22/2015
150	SPSS	6/16/2015
150	Java	8/12/2016

Third normal form:

A relation is in third normal form (3NF) if it is in second normal form and there is no transitive dependences exist.

A transitive dependency in a relation is a functional dependency between two (or more) nonekey attributes.

For example consider the relation :

SALES (<u>Cust_ID</u> , Name ,Salesperson, Region)

Part (a) SALES relation with sample data:

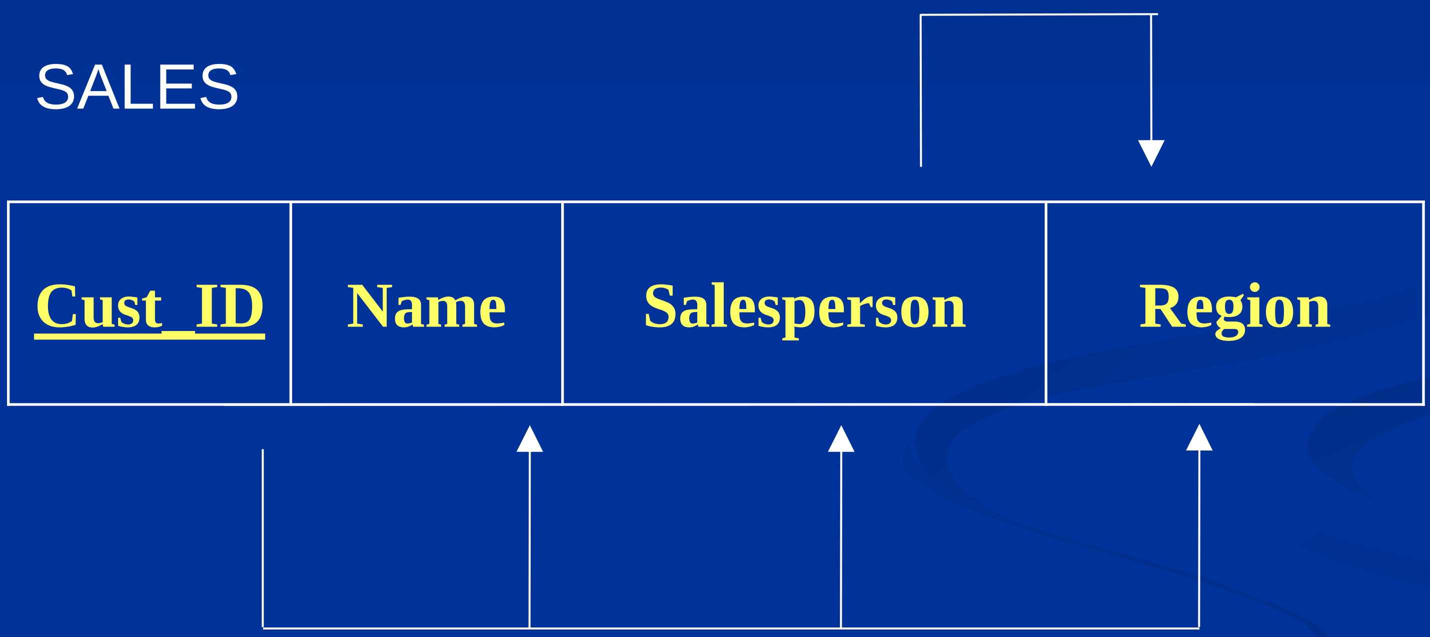
SALES

Cust_ID	Name	Salesperson	Region
South	Sami	Nawzat	8023
West	Jassam	Bassam	9167
South	Sami	Ahmed	7924
East	Hadi	Talal	6837
West	Jassam	Emad	8596
North	Fuaad	Arkan	7018

Part (b) Transitive dependency in SALES relation

:

SALES



- However there is a transitive dependency (Region is functional dependent on Salesperson and Salesperson is functional dependent of Cust_ID as a result there are update anomalies in SALES.

1. ***Insertion anomaly***: a new salesperson (Riad) assigned the north region cannot be entered until a customer has been assigned to that salesperson (since a value for Cust_ID must be provide to insert a row in the table) .
2. ***Deletion anomaly*** : if a customer number 6837 is deleted from the table , we lose the information that salesperson (Hadi) is assigned to the east region .
3. ***Modification Anomaly*** : if a sales person (Sami) is assigned to the east region , several rows must be changed to reflect that fact (two rows are shown in Part(a)

- These anomalies arise as a result of a transitive dependency .
- The transitive dependency can be removed by decomposing SALES into two relation as shown in figure below Part (a), note that salesperson which is a determinant in the transitive dependency in SALES becomes the primary key in the SPERSON, Salesperson becomes a foreign key in SALES1.
- As shown in figure Part(b) the new relations are now in third normal form since no transitive dependencies exists .
- You should verify that the anomalies that exists in SALES are not presented in SALES1 and SPERSON .

Part (a) decomposing SALES relation

SALES1

Cust_ID	Name	Salesperson
8023	Nawzat	Sami
9167	Bassam	Jassam
7924	Ahmed	Sami
6837	Talal	Hadi
8596	Emad	Jassam
7018	Arkan	Fuaad

SPERSON

Salesperson	Region
Sami	South
Jassam	West
Hadi	East
Fuaad	north

Part (b) Relation in 3NF :

SPERSON

Salesperson	Region

SALES1

Cust_ID	Name	Salesperson

- Transitive dependencies may also occur between sets of attributes in relation.
- For example the relation: SHIPMENT (Snum, Origin, Destination, Distance) could be used to record shipments according to origin , the destination , and distance .
- Sample data data for this relation appear in figure below Part (a).
- The functional dependencies in the shipment relation are shown in Part(b).
- The primary key of SHIPMENT relation is the attribute Snum(or shipment number) .

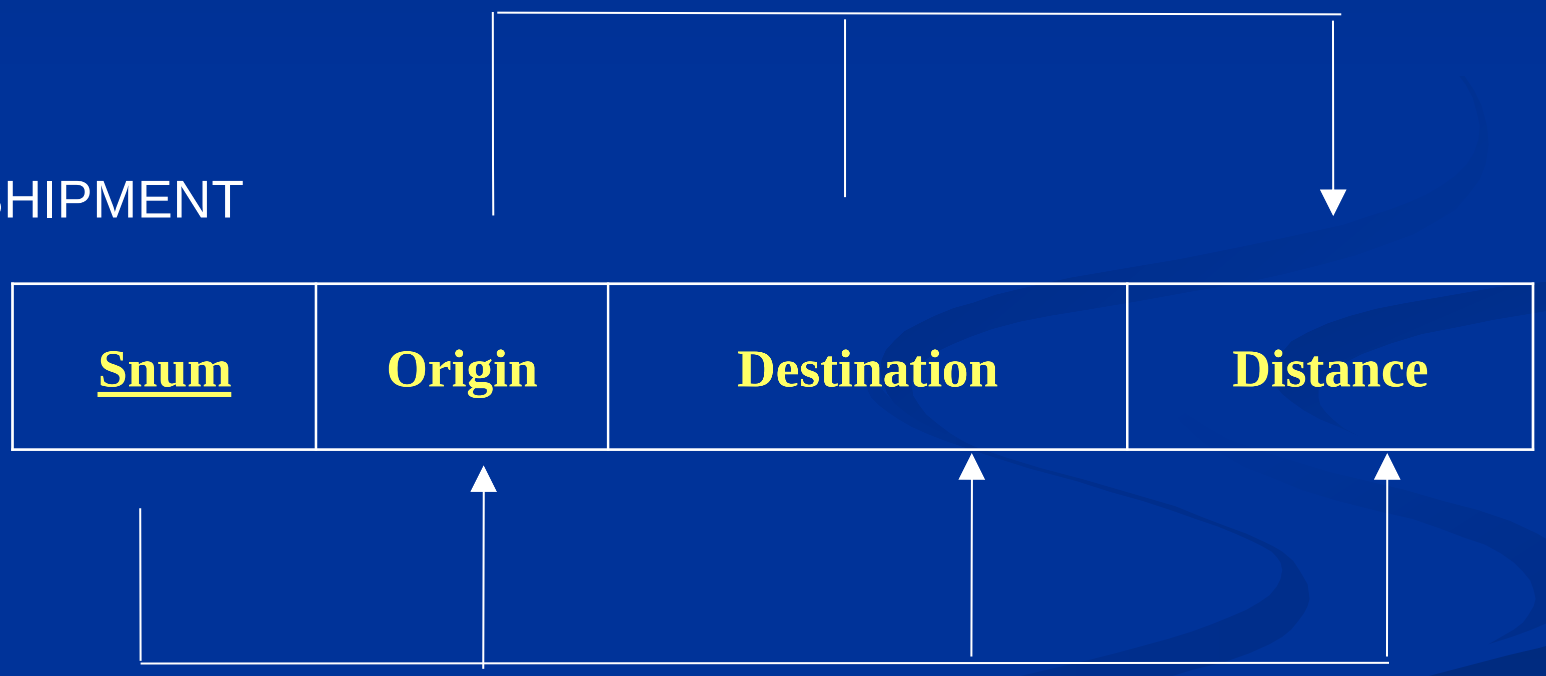
Part (a) SHIPMENT relation with sample data:

SHIPMENT

Snum	**Origin**	**Destination**	**Distance**
4.9	Seattle	Denver	1.537
618	Chicago	Dallas	1.053
723	Boston	Atlanta	1.214
824	Denver	Los angeles	1.150
629	Minneapolis	St Louis	587

Part (b) Functional dependence in SHIPMENT relation :

SHIPMENT



- We know that the relation is in second normal form, However, there is a transitive dependency in this relation: the distance attribute is a functional dependent on a pair of nonekey attributes origin and destination.
- As a result, there are anomalies in SHIPMENT (insertion , deletion , and modification anomalies). We can remove the transitive dependency in SHIPMENT by decomposing it into two relations that are in third normal form (3NF).

SHIPTO (<u>Snum</u>, Origin, Destination)

DISTANCES (<u>Origin</u>, <u>Destination</u>, Distance)

The first relations provides the origin and destination for any given shipment, the second provides distance between an origin and destination pair.

Sample data for this relation appear in figure Part(c)

Part (C) Relation in 3NF :

SHIPTO

Snum	Origin	Destination
4.9	Seattle	Denver
618	Chicago	Dallas
723	Boston	Atlanta
824	Denver	Los angeles
629	Minneapolis	St Louis

DISTANCES

Origin	Destination	Distance
Seattle	Denver	1.537
Chicago	Dallas	1.053
Boston	Atlanta	1.214
Denver	Los Angeles	1.150
Minneapolis	St Louis	587

END OF LECTURE 6