



COMPUTER SCIENCE

12

(MS Access and C)

CHAPTER 4: Data Integrity and Normalization

Topics

- Data Integrity
- Entity Integrity
- Referential Integrity
- Problems in Relation
- Database Anomalies
- Normalization
- Functional Dependency
- First Normal Form (1NF)
- Partial Dependency
- Second Normal Form (2NF)
- Transitive Dependency
- Third Normal Form (3NF)

Data Integrity

- ✓ Data integrity means reliability and accuracy of data
- ✓ Integrity rules are designed to keep the data consistent and correct
- ✓ These rules act like a check on the incoming data



Data Integrity

Examples

- ✓ If an employee id is entered as "123", this value should not be entered again
- ✓ The ID should not be assigned to two or more employees
- ✓ Fee of the student should not be greater than 10000
- ✓ Age of the employee should be less than or equal to 60

Entity Integrity

No duplicate records

- Entity Integrity ensures that there are no duplicate records within the relation

Unique and Not Null

- The attribute that identifies each record within the relation is unique and never null

Primary Key

- The existence of the Primary Key is the core of the entity integrity.

Entity Integrity

- ✓ To ensure entity integrity, it is required that every relation have a primary key
- ✓ The attribute that is used as primary key cannot contain null value
- ✓ It is not possible to uniquely identify a record in relation if primary key contains null value
- ✓ DBMS displays an error message if the user attempts to save a record without a valid value in the primary key attribute

Referential Integrity

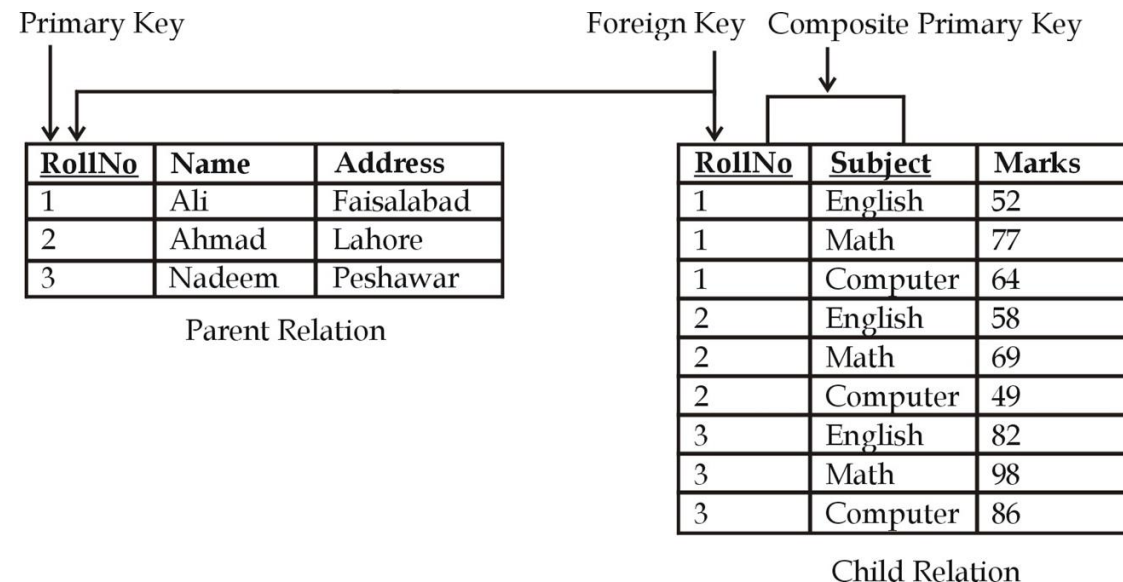
- ✓ A rule that states that if a foreign key exists in a relation, the foreign key value must match the primary key value of some tuples in its parent relation
- ✓ **Referential integrity** preserves the defined relationship between relations when records are added or deleted.
- ✓ It ensures that key values are consistent across the relations

Achieving Referential Integrity

- Can be achieved by creating relationship between the relations
- One relation is called **parent relation** and other is called **child relation**
- The Parent relation contains primary key and child relation contains foreign key

Achieving Referential Integrity

- Referential integrity ensures:
 - A value entered in the foreign key field of child relation must exist in the primary key field of parent relation
 - Record cannot be deleted from a child relation if matching records exist in parent relation
 - Primary key value cannot be changed in the parent relation if its corresponding records exist in the child relation



Major Problems / Errors in Relations

1

• Synonym

2

• Homonym

3

• Redundancy

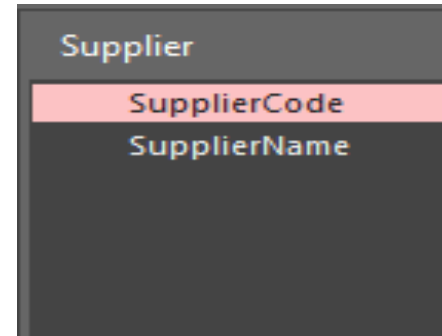
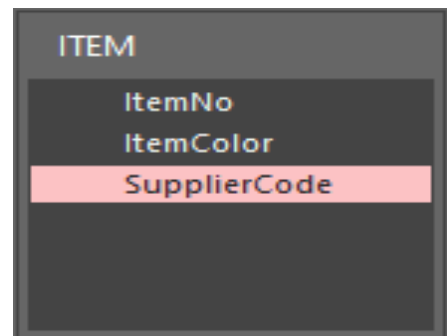
4

• Mutual Exclusiveness of Data

Major Problems / Errors in Relations

1- Synonym

- A type of problem in the relations that occurs when two different names are used for the same attribute
- **ITEM** (ItemNo, ItemColor, SupplierCode)
- **SUPPLIER** (SupplierID, SupplierName)
- The name of attribute must be same if it exists in two or more relations



Major Problems / Errors in Relations

2- Homonym

- A type of problem in the relations that occurs when the same name is used for two different attributes
- **Customer** (CustomerID, CompanyName)
- **Supplier** (SupplierID, CompanyName)
- Different names should be used for different attributes
- CompanyName attribute used in both relations
- Different names should be used for different attributes

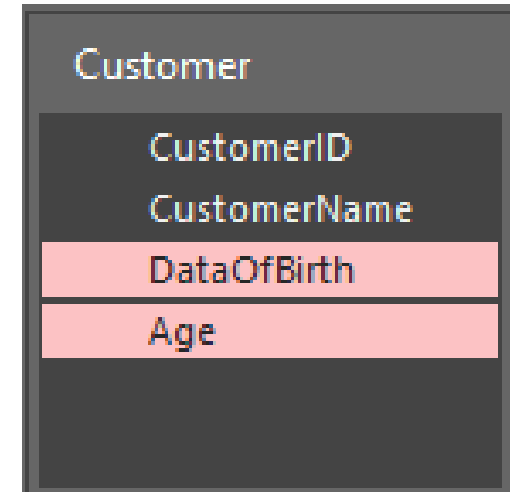
Customer	
CustomerID	
CompanayName	
DataOfBirth	
Age	

Supplier	
SupplierID	
SupplierCompany	

Major Problems / Errors in Relations

3- Redundancy

- A type of problem in the relations that occurs when same information is unnecessarily stored in two different ways or forms
- Customer relation have redundancy problem
 - Age and DateOfBirth stores the same type of information
 - Age can be calculated from the value of DateOfBirth attribute
 - Age attribute creating redundancy in relation



Major Problems / Errors in Relations

4- Mutual Exclusiveness of Data

- Problem created when values of attributes are stored as “True/False”
- Sometimes, two or more such attributes cannot be true or false at the same time

EMPLOYEE (EmployeeID, Married, Single)

- Attributes **Married** and **Single** cannot be true or false because
 - Person cannot be married and single at the same time
- The solution by combining both these attributes(Married, Single) into one attribute **MartialStatus**
- Possible value for this attribute can be **M** for Married and **S** for Single

Database Anomalies

- The problems in relations that occur due to redundancy in the relations
- Redundancies in a relation may result
 - Errors or inconsistencies (called anomalies)
- These anomalies affect the process of
 - Inserting, deleting and modifying data in the relations
- Some important data may be lost if a relation is updated that contains database anomalies

Types of anomalies

1

• Insertion Anomaly

2

• Deletion Anomaly

3

• Modification Anomaly

Database Anomalies

1. Insertion Anomaly

- The insertion anomaly occurs when a new record is inserted in the relation
- In this anomaly, the user cannot insert a fact about an entity until he has an additional fact about another entity

CustomerID	Name	SalesMan	Region
10	Ahsan	Ahmad	South
20	Babar	Bashir	West
30	Ali	Ahmad	South
40	Daood	Khalid	East
50	Raza	Bashir	West
60	Farooq	Munir	North
		Abid	North

Database Anomalies

2. Deletion Anomaly

- The deletion anomaly occurs when a record is deleted from the relation
- In this anomaly, the deletion of facts about an entity automatically deletes the fact of another entity

CustomerID	Name	SalesMan	Region
10	Ahsan	Ahmad	South
20	Babar	Bashir	West
30	Ali	Ahmad	South
40	Daood	Khalid	East
50	Raza	Bashir	West
60	Farooq	Munir	North

Database Anomalies

3. Modification Anomaly

- The modification anomaly occurs when the record is updated in the relation
- In this anomaly, the modification in the value of specific attribute requires modification in all records in which that value occurs

CustomerID	Name	SalesMan	Region
10	Ahsan	Ahmad	South
20	Babar	Bashir	West
30	Ali	Ahmad	South
40	Daood	Khalid	East
50	Raza	Bashir	West
60	Farooq	Munir	North

Normalization

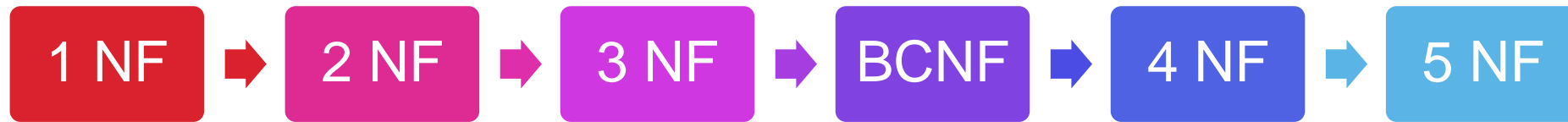
- Process of producing a simpler and more reliable database structure

Advantages of Normalization

- Reduces data redundancy
- Allows users to
 - Insert, delete, and update rows without causing data inconsistencies (anomalies)
- Used to create a suitable set of relations /tables for storing data and link them using relationship

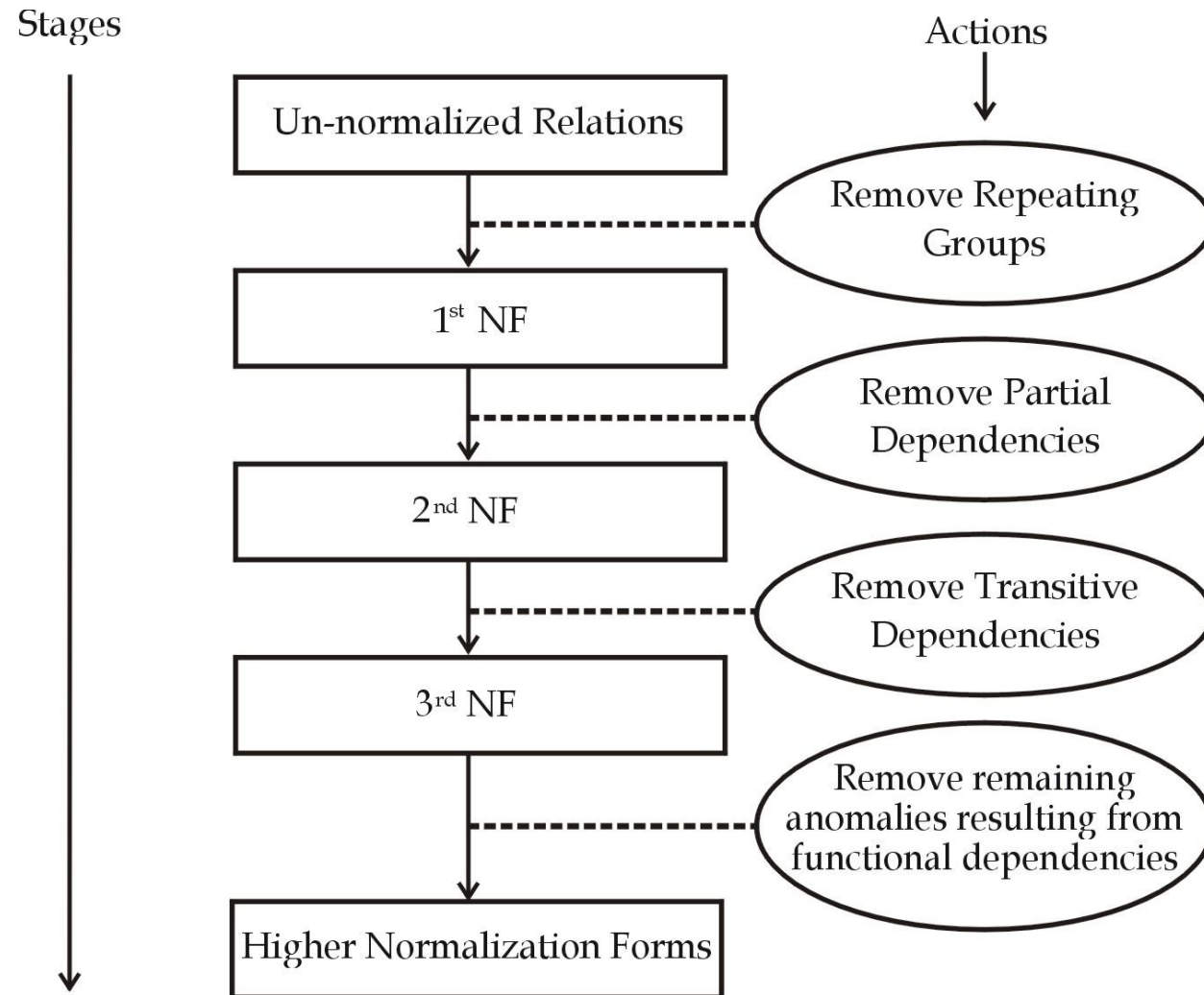
Normal Forms

- Normal Form is a series of stages done in Normalization



- Each normal form has certain requirements or condition
- These conditions have to fulfilled to bring the database in that particular normal form
- If a relation satisfies the conditions of a normal form, it is said to be in that normal form

Normal Forms



Functional Dependency

Functional dependency is a relationship between attributes

The value of one attribute determines the value of another attribute

Notation:  (arrow)

Functional Dependency

Example 1

- Relation **STUDENT** with following attributes:
STUDENT (RollNo, StudentName, Marks, Email)
- Functional dependency

RollNo \rightarrow StudentName , Marks, Email

Attribute RollNo represents determinant

Functional Dependency

Example 2

- Relation **BOOK** with following attributes:

BOOK (ISBN, Title, Author)

- Functional dependency

ISBN -> Title, Author

Functional Dependency

Example 3

- An attribute can also be functionally dependent on two or more attributes:

COURSE (RollNo, CourseID, DateCompleted)

- Functional dependency

RollNo, CourseID → DateCompleted

- **DateCompleted** is functionally dependent on both **RollNo** and **CourseID**
- It can be determined only if both **RollNo** and **CourseID** are known

First Normal Form (1NF)

A relation is in first normal form if every intersection of row and column contains atomic(single)values only

Each field in a relation should contain only one value

It means that the relation does not contain any repeating group

First Normal Form (1NF)

Repeating Group

A repeating group is a set of one or more data items that may occur a variable number of times in a tuple.

For Example: Skill, Phone Number, Degree etc.

Methods:

Method 1

- In this method, only single values are permitted at the intersection of each row and column; hence, there are no repeating groups

Method 2

- Place all items that appear in the repeating group in a new table
- Designate a primary key for each new table produced.
- Duplicate in the new table the primary key of the table from which the repeating group was extracted or vice versa

First Normal Form (1NF)

Applying Method 1 Example

DEPARTMENT (DeptNo, DeptName, EmpNo, EmpName)

DeptNo	DeptName	EmpNo	EmpName
10	Management	E01 E02	Usman Khalil Abdullah
20	Finance	E10 E11	Ali Ahmed Mahmood Abbas
30	IT	E25	Hamid Ali

First Normal Form (1NF)

Applying Method 1

- The attributes **EmpNo** and **EmpName** are being repeated for single occurrence of DeptNo

DeptNo	DeptName	EmpNo	EmpName
10	Management	E01	Usman Khalil
		E02	Abdullah
20	Finance	E10	Ali Ahmed
		E11	Mahmood Abbas
30	IT	E25	Hamid Ali

First Normal Form (1NF)

Applying Method 1

- Repeating groups are removed by entering proper data in the blank fields:

<u>DeptNo</u>	DeptName	<u>EmpNo</u>	EmpName
10	Management	E01	Usman Khalil
10	Management	E02	Abdullah
20	Finance	E10	Ali Ahmed
20	Finance	E11	Mahmood Abbas
30	IT	E25	Hamid Ali

- The above relation is now in 1NF as each field contains a single value
- It will have composite key that consists of DeptNo and EmpNo

First Normal Form (1NF)

Applying Method 2

The attributes **EmpNo** and **EmpName** are being repeated for single occurrence of **DeptNo**.


Steps are taken to remove the repeating groups:

1. The repeating group must be removed from the relation
 - It can be stored in a separate relation to represent an entity
2. Create a relationship between the new relations

First Normal Form (1NF)

<u>DeptNo</u>	DeptName
10	Management
20	Finance
30	IT

Table: The EMP relation



<u>EmpNo</u>	EmpName	DeptNo
E01	Usman Khalil	10
E02	Abdullah	10
E10	Ali Ahmed	20
E11	Mahmood Abbas	20
E25	Hamid Ali	30

Partial Dependency

A type of dependency in which one or more non-key attributes are functionally dependent on a part of primary

Creates problem of data redundancy(duplication) in the relation

How? We will explain with example

Partial Dependency

Example

- Relation Employee with following attributes:
Employee (EmpID, Name, DeptName, Salary, CourseTitle, DateCompleted)
- Contains composite primary key **EmpID** and **CourseTitle**

<u>EmpID</u>	Name	DeptName	Salary	<u>CourseTitle</u>	DateCompleted
100	Ahmad	Marketing	25,000	Advertising	19/06/2019
100	Ahmad	Marketing	25,000	Surveys	10/09/2019
140	Nazir	Accounting	19,000	MS Excel	12/08/2019
110	Hamid	IT	24,000	Oracle	14/07/2019
110	Hamid	IT	24,000	Java	22/09/2019
190	Rashid	Finance	30,000	Investment	20/06/2020
150	Hussain	Marketing	25,000	Advertising	19/06/2020
150	Hussain	Marketing	25,000	Ecommerce	20/09/2020

Partial Dependency

- Functional dependencies in Employee relation:

$\text{EmpID} \rightarrow \text{Name}, \text{DeptName}, \text{Salary}$

$\text{EmpID}, \text{CourseTitle} \rightarrow \text{DateCompleted}$

- The non-key attributes Name, DeptName and Salary are functionally dependent on a part of primary key EmpID
- They are not functionally dependent on complete key called partial dependency

Partial Dependency

Problems

- The partial dependency in the above relation creates redundancy
- It results in certain database anomalies when the relation is updated
 - Insertion Anomaly
 - Deletion Anomaly
 - Modification Anomaly

<u>EmpID</u>	Name	DeptName	Salary	<u>CourseTitle</u>	DateCompleted
100	Ahmad	Marketing	25,000	Advertising	19/06/2019
100	Ahmad	Marketing	25,000	Surveys	10/09/2019
140	Nazir	Accounting	19,000	MS Excel	12/08/2019
110	Hamid	IT	24,000	Oracle	14/07/2019
110	Hamid	IT	24,000	Java	22/09/2019
190	Rashid	Finance	30,000	Investment	20/06/2020
150	Hussain	Marketing	25,000	Advertising	19/06/2020
150	Hussain	Marketing	25,000	Ecommerce	20/09/2020

The EMPLOYEE relation

Second Normal Form (2NF)

A relation is in second normal form (2NF) if

```
graph TD; A[A relation is in second normal form (2NF) if] --> B[It is in 1NF]; A --> C[There is no Partial Dependency]; C --> D[Every non-key attribute is fully functionally dependent on the primary key. All non-key attributes must depend on all parts of the primary key];
```

It is in 1NF

There is no Partial Dependency

Every non-key attribute is fully functionally dependent on the primary key. All non-key attributes must depend on all parts of the primary key

Second Normal Form (2NF)

Example

EMPLOYEE (EmpID, Name, DeptName, Salary, CourseTitle, DateCompleted)

<u>EmpID</u>	Name	DeptName	Salary	<u>CourseTitle</u>	DateCompleted
100	Ahmad	Marketing	25,000	Advertising	19/06/2019
100	Ahmad	Marketing	25,000	Surveys	10/09/2019
140	Nazir	Accounting	19,000	MS Excel	12/08/2020
110	Hamid	IT	24,000	Oracle	14/07/2019
110	Hamid	IT	24,000	Java	22/09/2019
190	Rashid	Finance	30,000	Investment	20/06/2020
150	Hussain	Marketing	25,000	Advertising	19/06/2020
150	Hussain	Marketing	25,000	Ecommerce	20/09/2020

Second Normal Form (2NF)

Composite Primary Key in Relation

EMPLOYEE (EmpID, Name, DeptName, Salary, CourseTitle, DateCompleted)

EMPLOYEE (EmpID, CourseTitle)

Functional dependencies in relation

EmpID → Name, DeptName, Salary

EmpID, CourseTitle → DateCompleted

Second Normal Form (2NF)

Problems

- The partial dependency in the above relation creates redundancy
- It results in certain database anomalies when the relation is updated.
 - Insertion Anomaly
 - Deletion Anomaly
 - Modification Anomaly



Second Normal Form (2NF)

Remove Partial Dependency

Create Separate Tables for Each Dependency

EmpID → Name, DeptName, Salary

<u>EmpID</u>	Name	DeptName	Salary
100	Ahmad	Marketing	25,000
140	Nazir	Accounting	19,000
110	Hamid	IT	24,000
190	Rashid	Finance	30,000
150	Hussain	Marketing	25,000

The EMPLOYEE relation

EmpID, CourseTitle → DateCompleted

<u>EmpID</u>	<u>CourseTitle</u>	DateCompleted
100	Advertising	19/06/2019
100	Surveys	10/09/2019
140	MS Excel	12/08/2020
110	Oracle	14/07/2019
110	Java	22/09/2019
190	Investment	20/06/2020
150	Advertising	19/06/2020
150	Ecommerce	20/09/2020

The COURSE relation

Transitive Dependency

Transitive dependency exists if a non-key attribute depends on any other non-key attribute.

Example Relation **SALES** with following attributes:

<u>CustomerID</u>	NAME	SALESMAN	REGION
10	Ahsan	Ahmad	South
20	Babar	Bashir	West
30	Ali	Ahmad	South
40	Daood	Khalid	East
50	Raza	Bashir	West
60	Farooq	Munir	North

Dependencies in the Relation

CustomerID → Name, SalesMan
SalesMan → Region

- Relation is in 2NF because the primary key consists of single attribute
- A transitive dependency exists in relation
 - Region is functionally dependent on **SalesMan** and **SalesMan** is functionally dependent on **CustomerID**
 - Region is transitively dependent on **CustomerID**

Transitive Dependency

Problems

- Insertion Anomaly
- Deletion Anomaly
- Modification Anomaly

<u>CustomerID</u>	NAME	SALESMAN	REGION
10	Ahsan	Ahmad	South
20	Babar	Bashir	West
30	Ali	Ahmad	South
40	Daood	Khalid	East
50	Raza	Bashir	West
60	Farooq	Munir	North

Third Normal Form (3NF)

A relation is in second normal form (3NF) if

```
graph TD; A["A relation is in second normal form (3NF) if"] --> B["It is in 2NF"]; A --> C["There is no Transitive Dependency"];
```

It is in 2NF

There is no Transitive Dependency

Third Normal Form (3NF)

How to Achieve 3NF?

- Remove all attributes from the 2NF record that depend on another non-key field
- Place them into a new relation with the other attribute as the primary key

Third Normal Form (3NF)

Example Relation **SALES** with following attributes:

SALES (CustomerID, Name, SalesMan, Region)

<u>CustomerID</u>	Name	SalesMan	Region
10	Ahsan	Ahmad	South
20	Babar	Bashir	West
30	Ali	Ahmad	South
40	Daood	Khalid	East
50	Raza	Bashir	West
60	Farooq	Munir	North

Third Normal Form (3NF)

Dependencies in the Relation

CustomerID → Name, SalesMan

SalesMan → Region

<u>CustomerID</u>	Name	SalesMan	Region
10	Ahsan	Ahmad	South
20	Babar	Bashir	West
30	Ali	Ahmad	South
40	Daood	Khalid	East
50	Raza	Bashir	West
60	Farooq	Munir	North

Third Normal Form (3NF)

Dependencies in the Relation

CustomerID → Name, SalesMan

SalesMan → Region

<u>CustomerID</u>	Name	SalesMan	Region
10	Ahsan	Ahmad	South
20	Babar	Bashir	West
30	Ali	Ahmad	South
40	Daood	Khalid	East
50	Raza	Bashir	West
60	Farooq	Munir	North

Third Normal Form (3NF)

Problems

- Insertion Anomaly
- Deletion Anomaly
- Modification Anomaly

<u>CustomerID</u>	NAME	SALESMAN	REGION
10	Ahsan	Ahmad	South
20	Babar	Bashir	West
30	Ali	Ahmad	South
40	Daood	Khalid	East
50	Raza	Bashir	West
60	Farooq	Munir	North



Third Normal Form (3NF)

Remove Transitive Dependency

Create Separate Tables for Each Dependency

SALES (CustomerID, Name, SalesMan)

<u>CustomerID</u>	Name	SalesMan
10	Ahsan	Ahmad
20	Babar	Bashir
30	Ali	Ahmad
40	Daood	Khalid
50	Raza	Bashir
60	Farooq	Munir

The SALES relation

SALESMAN (SalesMan, Region)

<u>SalesMan</u>	Region
Ahmad	South
Bashir	West
Khalid	East
Munir	North

The SALESMAN relation