Normalization in DBMS (Up to 3NF)

1. Introduction to Normalization

- **Definition:** Normalization is the process of organizing data in a database to minimize redundancy and improve data integrity.
- **Objectives of Normalization:**
  - Eliminate redundant data.
  - Ensure data dependencies are logical.
  - Improve database performance.

---

2. Normal Forms Overview

- **1NF (First Normal Form)**
- **2NF (Second Normal Form)**
- **3NF (Third Normal Form)**

---

3. First Normal Form (1NF)

- **Definition:** A table is in 1NF if:
  1. Each column contains atomic (indivisible) values.
  2. Each column contains values of a single type.
  3. Each column has a unique name.

Example (Unnormalized Table):

OrderID	Customer	Items
1	John Doe	Pen, Pencil
2	Alice	Notebook

Conversion to 1NF:

OrderID	Customer	Item
1	John Doe	Pen
1	John Doe	Pencil
2	Alice	Notebook

---

4. Second Normal Form (2NF)

- **Definition:** A table is in 2NF if:
  1. It is in 1NF.
  2. All non-key attributes are fully functionally dependent on the primary key (i.e., no partial dependencies).

Example (1NF Table):

OrderID	Customer	Item	Price
1	John Doe	Pen	10
1	John Doe	Pencil	5

Issue: `Customer` depends only on `OrderID`, not on `Item`, leading to partial dependency.

Conversion to 2NF (Decomposing into two tables):

Orders Table:

OrderID	Customer
1	John Doe
2	Alice

OrderDetails Table:

OrderID	Item	Price
1	Pen	10
1	Pencil	5

---

5. Third Normal Form (3NF)

- **Definition:** A table is in 3NF if:
  1. It is in 2NF.
  2. There are no transitive dependencies (i.e., no attribute depends on another non-key attribute).

Example (2NF Table):

OrderID	CustomerID	CustomerName	City
1	C101	John Doe	New York
2	C102	Alice	Chicago

Issue: `CustomerName` and `City` depend on `CustomerID`, not `OrderID`, leading to transitive dependency.

`A ->B AND B -> C => A -> C`

Conversion to 3NF (Decomposing into separate tables):

Orders Table:

OrderID	CustomerID
1	C101

OrderID	CustomerID
2	C102

Customers Table:

CustomerID	CustomerName	City
C101	John Doe	New York
C102	Alice	Chicago

---

6. Summary of Normal Forms

Normal Form	Key Criteria
1NF	Eliminate duplicate columns, ensure atomicity
2NF	Eliminate partial dependencies
3NF	Eliminate transitive dependencies

Normalization Practice Questions

Question 1: Convert to 1NF

Consider the following table and convert it to First Normal Form (1NF):

OrderID	CustomerName	Products	TotalAmount
101	John Doe	Laptop, Mouse	1500
102	Alice Smith	Phone, Charger	800

Question 2: Identify Partial Dependencies for 2NF

Analyze the following table and identify any partial dependencies that need to be removed to achieve Second Normal Form (2NF):

StudentID	CourseID	StudentName	CourseName	Instructor
S01	C101	John Doe	Math	Dr. Smith
S02	C102	Alice Brown	Science	Dr. Johnson

Question 3: Normalize to 3NF

The following table contains transitive dependencies. Convert it to Third Normal Form (3NF):

EmployeeID	Name	Department	Dept_Location	Manager
201	Bob	Sales	New York	Tom
202	Alice	IT	San Francisco	Jerry

Question 4: Identify the Normal Form

Determine the highest normal form achieved by the following table:

ProjectID	ProjectName	EmployeeID	EmployeeName	StartDate
P01	Alpha	E01	John Doe	2024-01-10
P02	Beta	E02	Alice Brown	2024-02-15

Question 5: Normalize to 2NF

Consider the table and remove any partial dependencies to achieve 2NF:

InvoiceNo	ProductID	ProductName	CustomerID	CustomerName
5001	P01	Laptop	C01	John Doe
5002	P02	Phone	C02	Alice Brown

Question 6: Normalize the Following Table

Analyze the below data and normalize it up to 3NF:

CourseID	CourseName	StudentID	StudentName	Instructor	InstructorContact
C101	Physics	S001	John	Dr. Brown	123-456-7890
C102	Math	S002	Alice	Dr. Smith	987-654-3210

Question 7: Remove Redundancy

The following table contains redundant data. Suggest a way to normalize it:

OrderID	CustomerName	CustomerPhone	ProductName	ProductPrice
1	John Doe	123-456-7890	Laptop	1000
2	John Doe	123-456-7890	Mouse	50

Question 8: Split the Table to Achieve 3NF

Given the following table, break it down into smaller tables to achieve 3NF:

BookID	Title	Author	Publisher	PublisherAddress
B01	DBMS Concepts	John King	Pearson	123 St, NY
B02	Algorithms	Alice Doe	Wiley	456 Rd, LA

Question 9: Identify the Dependency Issues

Analyze the table below and identify dependency-related issues:

EmployeeID	Name	DOB	Department	DepartmentHead	HeadPhone
E01	Bob	1990-01-01	Sales	Tom	111-222-333
E02	Alice	1992-05-10	IT	Jerry	444-555-666

Question 10: Create 3NF from the Following Table

Normalize the following table to 3NF and suggest a better schema:

OrderID	ProductName	Category	Supplier	SupplierPhone
101	Laptop	Electronics	BestBuy	111-222-333
102	Chair	Furniture	Ikea	444-555-666

Solution

Question 1: Convert to 1NF (First Normal Form)

Given Table:

OrderID	CustomerName	Products	TotalAmount
101	John Doe	Laptop, Mouse	1500
102	Alice Smith	Phone, Charger	800

Why it’s Not in 1NF:

- The `Products` column contains multiple values (Laptop, Mouse) and (Phone, Charger).
- A table violates 1NF if it contains **non-atomic (multivalued) attributes.**
- To be in 1NF, each column should contain atomic (indivisible) values.

Solution: Convert to 1NF

Break down multi-valued columns into separate rows:

OrderID	CustomerName	Product	TotalAmount

OrderID	CustomerName	Product	TotalAmount
101	John Doe	Laptop	1500
101	John Doe	Mouse	1500
102	Alice Smith	Phone	800
102	Alice Smith	Charger	800

Now the table is in 1NF because all attributes contain atomic values.

Question 2: Identify Partial Dependencies for 2NF

Given Table:

StudentID	CourseID	StudentName	CourseName	Instructor
S01	C101	John Doe	Math	Dr. Smith
S02	C102	Alice Brown	Science	Dr. Johnson

Identifying Partial Dependencies:

- The composite key is `(StudentID, CourseID)`.
- **Partial dependency** occurs when a non-key attribute depends on part of the composite key instead of the whole.
- `StudentName` depends only on `StudentID`.
- `CourseName` and `Instructor` depend only on `CourseID`.

Solution: Remove Partial Dependencies

Decompose into separate tables:

1. **Student Table (StudentID as primary key):**

StudentID	StudentName
S01	John Doe
S02	Alice Brown

2. **Course Table (CourseID as primary key):**

CourseID	CourseName	Instructor
C101	Math	Dr. Smith
C102	Science	Dr. Johnson

3. **Enrollment Table (Composite key StudentID, CourseID):**

StudentID	CourseID
S01	C101
S02	C102

Now the tables are in 2NF, as all partial dependencies have been removed.

Question 3: Normalize to 3NF (Third Normal Form)

Given Table:

EmployeeID	Name	Department	Dept_Location	Manager
201	Bob	Sales	New York	Tom
202	Alice	IT	San Francisco	Jerry

Identifying Transitive Dependencies:

- `Dept_Location` and `Manager` depend on `Department` rather than directly on `EmployeeID`.
- A table is in 3NF if it has no **transitive dependencies**, meaning no non-prime attribute should depend on another non-prime attribute.

Solution: Remove Transitive Dependencies

Decompose into separate tables:

1. **Employee Table (EmployeeID as primary key):**

EmployeeID	Name	Department
201	Bob	Sales
202	Alice	IT

2. **Department Table (Department as primary key):**

Department	Dept_Location	Manager
Sales	New York	Tom
IT	San Francisco	Jerry

Now the tables are in 3NF, as transitive dependencies have been eliminated.

Question 4: Identify the Normal Form

Given Table:

ProjectID	ProjectName	EmployeeID	EmployeeName	StartDate
P01	Alpha	E01	John Doe	2024-01-10
P02	Beta	E02	Alice Brown	2024-02-15

Step 1: 1NF Check

- No multi-valued attributes, so the table is in 1NF.

Step 2: 2NF Check

- The composite key might be `(ProjectID, EmployeeID)`, and `EmployeeName` depends only on `EmployeeID`.
- There is a partial dependency, meaning the table is **not in 2NF**.

Step 3: Conclusion

- Since partial dependency exists, the highest normal form achieved is **1NF**.

Question 5: Normalize to 2NF (Second Normal Form)

Given Table:

InvoiceNo	ProductID	ProductName	CustomerID	CustomerName
5001	P01	Laptop	C01	John Doe
5002	P02	Phone	C02	Alice Brown

Identifying Partial Dependencies:

- The composite key is `(InvoiceNo, ProductID, CustomerID)`.
- `ProductName` depends only on `ProductID`.
- `CustomerName` depends only on `CustomerID`.

Solution: Remove Partial Dependencies

Decompose into separate tables:

1. **Invoice Table (InvoiceNo, CustomerID):**

InvoiceNo	CustomerID
5001	C01
5002	C02

2. **Customer Table (CustomerID as primary key):**

CustomerID	CustomerName
C01	John Doe
C02	Alice Brown

3. **Product Table (ProductID as primary key):**

ProductID	ProductName
P01	Laptop
P02	Phone

4. **Invoice_Product Table (InvoiceNo, ProductID):**

InvoiceNo	ProductID
5001	P01
5002	P02

Now the tables are in 2NF, as all partial dependencies have been removed.

Let's go through the normalization process for each question step by step.

Question 6: Normalize the Following Table to 3NF

Table:

CourseID	CourseName	StudentID	StudentName	Instructor	InstructorContact
C101	Physics	S001	John	Dr. Brown	123-456-7890
C102	Math	S002	Alice	Dr. Smith	987-654-3210

Step 1: 1NF (First Normal Form)

- The table is in 1NF because all columns have atomic values (no repeating groups or arrays).

Step 2: 2NF (Second Normal Form)

- Identify the primary key: `CourseID` and `StudentID` together are the composite key.
- All non-key attributes must depend on the entire composite key, but `Instructor` and `InstructorContact` only depend on `CourseID`. Hence, they violate 2NF.
- We can split this table into two:
  1. **Course-Student Table** (Student enrollments in courses)
  2. **Instructor Table** (Course-Instructor mapping)

New Tables after 2NF:

1. **CourseStudent Table**

CourseID	StudentID	StudentName
C101	S001	John
C102	S002	Alice

2. **Instructor Table**

CourseID	Instructor	InstructorContact
C101	Dr. Brown	123-456-7890
C102	Dr. Smith	987-654-3210

Step 3: 3NF (Third Normal Form)

- There is no transitive dependency in both tables, so they are in 3NF.
- The data is now normalized.

Question 7: Remove Redundancy and Normalize

Table:

OrderID	CustomerName	CustomerPhone	ProductName	ProductPrice
1	John Doe	123-456-7890	Laptop	1000
2	John Doe	123-456-7890	Mouse	50

Step 1: 1NF

- The table is already in 1NF.

Step 2: 2NF

- The primary key should be `OrderID`, but `CustomerName`, `CustomerPhone`, `ProductName`, and `ProductPrice` have partial dependencies.
- `CustomerName` and `CustomerPhone` depend on `OrderID` but not on `ProductName`.
- `ProductName` and `ProductPrice` depend on `OrderID`.

To remove redundancy, we break the table into three:

1. **Customer Table**

CustomerID	CustomerName	CustomerPhone
C001	John Doe	123-456-7890

2. **Order Table**

OrderID	CustomerID

OrderID	CustomerID
1	C001
2	C001

3. **Product Table**

OrderID	ProductName	ProductPrice
1	Laptop	1000
2	Mouse	50

Step 3: 3NF

- All the tables are in 3NF as there is no transitive dependency. The final schema eliminates redundancy.

Question 8: Split the Table to Achieve 3NF

Table:

BookID	Title	Author	Publisher	PublisherAddress
B01	DBMS Concepts	John King	Pearson	123 St, NY
B02	Algorithms	Alice Doe	Wiley	456 Rd, LA

Step 1: 1NF

- The table is in 1NF.

Step 2: 2NF

- The primary key is `BookID`, but `Author`, `Publisher`, and `PublisherAddress` are partially dependent on `Publisher`.
- Split the table into three tables:

1. **Book Table**

BookID	Title	Author
B01	DBMS Concepts	John King
B02	Algorithms	Alice Doe

2. **Publisher Table**

PublisherID	Publisher	PublisherAddress
P001	Pearson	123 St, NY
P002	Wiley	456 Rd, LA

3. **BookPublisher Table** (to represent the relationship between Book and Publisher)

BookID	PublisherID
B01	P001
B02	P002

Step 3: 3NF

- There are no transitive dependencies. The tables are in 3NF now.

Question 9: Identify the Dependency Issues

Table:

EmployeeID	Name	DOB	Department	DepartmentHead	HeadPhone
E01	Bob	1990-01-01	Sales	Tom	111-222-333
E02	Alice	1992-05-10	IT	Jerry	444-555-666

Step 1: Identify Dependency Issues

- There is a transitive dependency here:
  - `DepartmentHead` and `HeadPhone` depend on `Department` but are not functionally dependent on `EmployeeID`.
  - We should separate the department information into its own table.

Step 2: Normalize to 3NF

1. **Employee Table**

EmployeeID	Name	DOB	DepartmentID
E01	Bob	1990-01-01	D001
E02	Alice	1992-05-10	D002

2. **Department Table**

DepartmentID	Department	DepartmentHead	HeadPhone
D001	Sales	Tom	111-222-333
D002	IT	Jerry	444-555-666

Question 10: Create 3NF from the Following Table

Table:

OrderID	ProductName	Category	Supplier	SupplierPhone
101	Laptop	Electronics	BestBuy	111-222-333
102	Chair	Furniture	Ikea	444-555-666

Step 1: 1NF

- The table is in 1NF.

Step 2: 2NF

- The primary key is `OrderID`, but `ProductName`, `Category`, `Supplier`, and `SupplierPhone` have partial dependencies.
- `ProductName`, `Category`, and `Supplier` are not fully dependent on `OrderID`.

Step 3: Normalize to 3NF

1. **Order Table**

OrderID	ProductID
101	P001
102	P002

2. **Product Table**

ProductID	ProductName	Category
P001	Laptop	Electronics
P002	Chair	Furniture

3. **Supplier Table**

SupplierID	Supplier	SupplierPhone
S001	BestBuy	111-222-333
S002	Ikea	444-555-666

4. **ProductSupplier Table** (to represent the relationship between Product and Supplier)

ProductID	SupplierID
P001	S001
P002	S002

Conclusion: The tables are now normalized up to 3NF, with redundancy removed and all dependencies managed properly.