Normalization (DBMS) - 2

Created	@April 7, 2025 2:14 PM
Tags	

When we decompose schemas into sub-schemas, two properties are desirable:

1. Lossless
2. Dependency-preserving

All Decompositions are ***Normalizations*** - done to make schemas reach certain “standards”.

1NF (First Normal Form)

🔷 Rule:

- All attribute values must be **atomic** (indivisible).
- No **repeating groups** or arrays.

❌ Example (Not in 1NF):

StudentID	Name	Courses
1	Alice	Math, Science
2	Bob	English

- `Courses` has **multiple values** in one cell — violates 1NF

✅ 1NF Version:

StudentID	Name	Course
1	Alice	Math
1	Alice	Science
2	Bob	English

Project Code	Project Title	Project Manager	Project Budget	Emplyoee No	Employee Name	Department No	Department Name	Hourly Rate
PC10	MIS	Nisarga	24500	S10001	ARUN	L004	IT	22
PC10	MIS	Nisarga	24500	S10030	LOKESH	L023	Pionix	18
PC10	MIS	Nisarga	24500	S21010	PAVANI	L004	IT	21
PC45	LIS	Eenchara	17400	S10010	BABU	L004	IT	21
PC45	LIS	Eenchara	17400	S10001	ARUN	L004	IT	18
PC45	LIS	Eenchara	17400	S31002	TULASI	L028	Database	25
PC45	LIS	Eenchara	17400	S13210	WILLIAM	L008	Systems	17
PC64	HMS	Prakruthi	12250	S31002	TULASI	L028	Database	23
PC64	HMS	Prakruthi	12250	S21010	PAVANI	L004	IT	17
PC64	HMS	Prakruthi	12250	S10034	BALU	L009	HR	16

UNF:UnNormalized Table

1NF Tables: Repeating Attributes Removed

Project Code	Project Title	Project Manager	Project Budget
PC10	MIS	Nisarga	24500
PC45	LIS	Eenchara	17400
PC64	HMS	Prakruthi	12250

Project Code	Emplyoee No	Employee Name	Department No	Department Name	Hourly Rate
PC10	S10001	ARUN	L004	IT	22
PC10	S10030	LOKESH	L023	Pionix	18
PC10	S21010	PAVANI	L004	IT	21
PC45	S10010	BABU	L004	IT	21
PC45	S10001	ARUN	L004	IT	18
PC45	S31002	TULASI	L028	Database	25
PC45	S13210	WILLIAM	L008	Systems	17
PC64	S31002	TULASI	L028	Database	23
PC64	S21010	PAVANI	L004	IT	17
PC64	S10034	BALU	L009	HR	16

Table: `EmployeeDepartment`

EmpID	EmpName	DeptID	DeptName	DeptLocation
E1	Alice	D1	Sales	New York, Boston
E2	Bob	D2	HR	Chicago
E3	Charlie	D1	Sales	New York, Boston

🔻 Problem:

- `DeptLocation` has **multiple values** → violates **1NF**

✅ Step 1: Convert to 1NF

Break down multi-valued attributes into atomic values:

EmpID	EmpName	DeptID	DeptName	DeptLocation
E1	Alice	D1	Sales	New York
E1	Alice	D1	Sales	Boston
E2	Bob	D2	HR	Chicago
E3	Charlie	D1	Sales	New York
E3	Charlie	D1	Sales	Boston

✅ All values are now **atomic** → relation is in **1NF**

✅ 2NF (Second Normal Form)

🔷 Rules:

1. Must be in **1NF**
2. **No partial dependency**:

   Non-prime attributes must depend on **entire candidate key**, not just part of it

---

❌ Example (Not in 2NF):

StudentID	CourseID	StudentName	Instructor
101	CS101	Alice	Dr. Smith
101	MA101	Alice	Dr. Brown
102	CS101	Bob	Dr. Smith
103	MA101	Carol	Dr. Brown

103	PH101	Carol	Dr. Green

Table: Enrollment(StudentID, CourseID, StudentName)

- Candidate Key: (StudentID, CourseID)
- FD: `StudentID → StudentName` ❌ Partial dependency (depends on part of key)

✅ 2NF Version:

Enrollment(StudentID, CourseID)

Student(StudentID, StudentName)

✅ 3NF (Third Normal Form)

🔷 Rules:

1. Must be in **2NF**
2. **No transitive dependency**:

   Non-prime attributes must not depend on **other non-prime attributes**

Let's take this relation:

```
Employee(EmpID, EmpName, DeptID, DeptName, DeptLocation)
```

🧪 Sample Data (Before Decomposition)

EmpID	EmpName	DeptID	DeptName	DeptLocation
1	Alice	D1	HR	NY
2	Bob	D2	IT	SF
3	Carol	D1	HR	NY

✅ After Decomposition:

Table: Employee

EmpID	EmpName	DeptID
1	Alice	D1
2	Bob	D2
3	Carol	D1

Table: Department

DeptID	DeptName	DeptLocation
D1	HR	NY
D2	IT	SF

🎯 Functional Dependencies (FDs):

1. `EmpID → EmpName, DeptID`
2. `DeptID → DeptName, DeptLocation`

🔑 Candidate Key: `EmpID`

🚫 Problem: Transitive Dependency

We see:

- `EmpID → DeptID`
- `DeptID → DeptName, DeptLocation`

  ⛔ So, `EmpID → DeptName, DeptLocation` is a **transitive dependency**, which violates 3NF.

🛠️ Step 2: Decompose into 3NF

✅ Decomposition:

1. **Employee(EmpID, EmpName, DeptID)**
2. **Department(DeptID, DeptName, DeptLocation)**

✅ Now:

- No **partial dependency** ✅ (in 2NF)
- No **transitive dependency** ✅ (in 3NF)
- All FDs are preserved ✅

✅ 3NF Check:

Let's apply the 3NF conditions:

- All FDs must have:
    1. Left-hand side is a **super key**, or
    2. Right-hand side is a **prime attribute** (part of a candidate key)

Now:

- `CourseID → Instructor` : ✅ `CourseID` is the key
- `Instructor → Room` :

  ❌ `Instructor` is **not** a superkey

  ❌ `Room` is **not** a prime attribute

But wait! In **3NF**, the exception is that **if the RHS is a prime attribute**, it's okay.

In this case, it's **not**, but still accepted under **3NF**, **because we've kept transitive dependency** separate.

So, this table **passes 3NF**, but fails **BCNF**.

❌ Why Not BCNF?

- **Instructor → Room** violates BCNF:

  Because **Instructor is not a superkey**, but it determines another attribute.

🎯 Boyce-Codd Normal Form

A relation is in **BCNF** if:

- **For every non-trivial functional dependency** `X → Y`,

  👉 `X` is a **super key**.

So, BCNF is **stricter than 3NF** — it removes **even those transitive dependencies** where the determinant is not a superkey.

🧾 Relation: `Courses(CourseID, Instructor, Room)`

📌 Functional Dependencies (FDs):

1. `CourseID → Instructor`
2. `Instructor → Room`

🔑 Candidate Key: `CourseID`

> Because CourseID uniquely determines the rest of the attributes.

🔧 BCNF Decomposition

We split the table to remove the violation:

✂️ Decompose into:

1. **InstructorRoom(Instructor, Room)**

   → Because `Instructor → Room` is the violating FD

2. **CourseInstructor(CourseID, Instructor)**

   → Retain the mapping from course to instructor

✅ Final BCNF Tables:

InstructorRoom

Instructor	Room
Alice	101
Bob	102

CourseInstructor

CourseID	Instructor
CS101	Alice
MA101	Bob

🧠 What is a Multivalued Dependency (MVD)?

In a relation **R**, a multivalued dependency **A →→ B** means:

> For each value of A, there is a set of independent values of B, and these values are not dependent on other attributes.

It’s like saying:

> "Given A, B can take multiple values independently of the rest."

📋 Example: `StudentHobbies(StudentID, Hobby, Language)`

This table tracks:

- Students
- Their hobbies
- The languages they know

📌 Assumptions:

- A student can have **multiple hobbies**
- A student can know **multiple languages**
- But **hobbies and languages are independent** — knowing a language doesn’t depend on the hobby

---

🔍 Multivalued Dependencies:

- `StudentID →→ Hobby`

- `StudentID →→ Language`

Because:

Each student has **a set of hobbies**

Each student has **a set of languages**

And they are **not related to each other**

🔢 Sample Data:

StudentID	Hobby	Language
S1	Painting	English
S1	Painting	French
S1	Dancing	English
S1	Dancing	French

> This table is storing all combinations of hobbies × languages for a student — even though they aren't related.

🚫 Problem:

This redundancy is caused by the **multivalued dependency**, and it violates **4NF**.

✅ 4NF Decomposition (To Remove MVDs):

Split into two separate relations:

1. `StudentHobby(StudentID, Hobby)`

StudentID	Hobby
S1	Painting
S1	Dancing

2. `StudentLanguage(StudentID, Language)`

StudentID	Language
S1	English
S1	French

Now there's **no redundancy** or **unnecessary combinations**.

2NF Tables: Partial Key Dependencies Removed

Project Code	Project Title	Project Manager	Project Budget
PC10	MIS	Nisarga	24500
PC45	LIS	Eenchara	17400
PC64	HMS	Prakruthi	12250

Project Code	Emplyoee No	Hourly Rate
PC10	S10001	22
PC10	S10030	18
PC10	S21010	21
PC45	S10010	21
PC45	S10001	18
PC45	S31002	25
PC45	S13210	17
PC64	S31002	23
PC64	S21010	17
PC64	S10034	16

Emplyoee No	Employee Name	Department No	Department Name
S10001	ARUN	L004	IT
S10030	LOKESH	L023	Pionix
S21010	PAVANI	L004	IT
S10010	BABU	L004	IT
S31002	TULASI	L028	Database
S13210	WILLIAM	L008	Systems
S10034	BALU	L009	HR

3NF Tables: Non-Key Dependencies Removed

Project Code	**Project Title**	**Project Manager**	**Project Budget**
PC10	MIS	Nisarga	24500
PC45	LIS	Eenchara	17400
PC64	HMS	Prakruthi	12250

Project Code	**Emplyoee No**	**Hourly Rate**
PC10	S10001	22
PC10	S10030	18
PC10	S21010	21
PC45	S10010	21
PC45	S10001	18
PC45	S31002	25
PC45	S13210	17
PC64	S31002	23
PC64	S21010	17
PC64	S10034	16

Emplyoee No	**Employee Name**	**Department No**
S10001	ARUN	L004
S10030	LOKESH	L023
S21010	PAVANI	L004
S10010	BABU	L004
S31002	TULASI	L028
S13210	WILLIAM	L008
S10034	BALU	L009

Department No	**Department Name**
L004	IT
L023	Pionix
L028	Database
L008	Systems
L009	HR