🔰 STEPS OF NORMALIZATION

Step	Description
1	Begin with a table (UNF) – with all data combined
2	Apply 1NF: Eliminate repeating groups / multivalued columns
3	Apply 2NF: Eliminate partial dependencies (if composite PK)
4	Apply 3NF: Eliminate transitive dependencies
5	Optionally, apply BCNF or higher (if determinant is not a candidate key)

🔄 NORMAL FORMS (NF)

✅ 1NF (First Normal Form)

- Atomic values only (no arrays, comma-separated values)
- Remove repeating groups
- Ensure each row is uniquely identified (usually with a primary key)

✅ 2NF (Second Normal Form)

- Table should be in 1NF
- No partial dependency on composite primary key
- All non-key columns depend on the entire primary key

✅ 3NF (Third Normal Form)

- Table should be in 2NF
- No transitive dependency
- Every non-key attribute must depend only on the primary key

✅ BCNF (Boyce-Codd Normal Form)

- Stricter version of 3NF
- Every determinant must be a candidate key

✅ FULL WORKING EXAMPLE – COVERS ALL NFs UPTO 3NF

🎯 Scenario: College Course Registration

A university maintains a single table like this:

StudentID	StudentName	Course	Instructor	InstructorPhone
S101	Alice	Java	Smith	9876543210
S101	Alice	DBMS	John	9123456789
S102	Bob	DBMS	John	9123456789

🔴 UNF (Unnormalized Form)

- Multivalued: One student has multiple courses in same row OR multiple rows repeating full details
- Repetition of student name, instructor details

🔵 1NF (First Normal Form)

Make sure each field is atomic and single valued.

StudentID	StudentName	Course	Instructor	InstructorPhone
S101	Alice	Java	Smith	9876543210
S101	Alice	DBMS	John	9123456789
S102	Bob	DBMS	John	9123456789

Now: ✅ Atomic values. But: ❌ Still data repetition → Instructor details are repeated.

🔵 2NF (Second Normal Form)

Condition: Remove **partial dependency**

Let's assume **composite key = (StudentID, Course)**

- StudentName depends only on StudentID → violates 2NF
- InstructorPhone depends only on Instructor → violates both PKs

Split into 3 tables:

👨‍🎓 **Table: Student**

StudentID	StudentName
S101	Alice
S102	Bob

📚 **Table: CourseRegistration**

StudentID	Course
S101	Java
S101	DBMS
S102	DBMS

👩‍🏫 **Table: Instructor**

Course	Instructor	InstructorPhone
Java	Smith	9876543210
DBMS	John	9123456789

✅ All non-key attributes now fully depend on **entire PK** of their respective tables.

🔵 **3NF (Third Normal Form)**

Check for **transitive dependencies**

In Instructor table:

- Course → Instructor
- Instructor → InstructorPhone → ❌ Transitive Dependency

Fix it by separating Instructor details

📘 **Table: Course**

Course	Instructor
Java	Smith

Course Instructor

DBMS John

📞 **Table: Instructor**

Instructor InstructorPhone

Smith 9876543210

John 9123456789

✅ Now everything is in **3NF**

🎓 **FINAL STRUCTURE**

1. **Student(StudentID, StudentName)**
2. **CourseRegistration(StudentID, Course)**
3. **Course(Course, Instructor)**
4. **Instructor(Instructor, InstructorPhone)**

BCNF (Boyce-Codd Normal Form) Example

🔍 First, What's the Problem?

Even if a table is in **3NF**, there can still be **anomalies** if:

A **non-primary attribute (determinant)** determines another attribute
AND
That **determinant is not a candidate key**

This is where **BCNF** is needed.

🧪 EXAMPLE: College Room Allocation

🎓 Table: RoomAllocation

StudentID	RoomNo	Warden
S1	R1	W1
S2	R1	W1
S3	R2	W2
S4	R3	W3

🔍 Functional Dependencies

Let's identify the **Functional Dependencies (FDs)**:

1. StudentID → RoomNo ✅ (Each student is assigned one room)
2. RoomNo → Warden ✅ (Each room has one warden)

🔎 Primary Key = StudentID

So all **non-key attributes** should depend **only** on StudentID

→ Let's check:

- RoomNo depends on StudentID ✅
- But Warden depends on RoomNo, **not StudentID** ❌

✅ **Is it in 3NF?**

Check Transitive Dependency:

StudentID → RoomNo → Warden
So StudentID → Warden is **transitive** → We can move RoomNo → Warden to a new table.

Split Tables for 3NF:

🔹 **Table 1: StudentRoom**

StudentID	RoomNo
S1	R1
S2	R1
S3	R2
S4	R3

🔹 **Table 2: RoomWarden**

RoomNo	Warden
R1	W1
R2	W2
R3	W3

✅ Now all non-key attributes depend **only** on the PK.
So this is in **3NF**.

❌ **But... Not Yet in BCNF**

Look at RoomNo → Warden

- RoomNo is a **determinant**
- But is RoomNo a **candidate key** in original table?

NO – because in the original RoomAllocation, the **PK was StudentID**, and RoomNo was not unique

➡️ So 3NF allowed this, but **BCNF does not**

✅ **To Fix for BCNF**

Split the original table into:

📘 **Table 1: StudentRoom (StudentID → RoomNo)**

StudentID	RoomNo
S1	R1
S2	R1
S3	R2
S4	R3

📘 **Table 2: RoomWarden (RoomNo → Warden)**

RoomNo	Warden
R1	W1
R2	W2
R3	W3

Now, both tables are in **BCNF**:

- Every determinant is a candidate key
- No anomalies left

📌 **Key Takeaways for BCNF:**

Concept	Rule
Determinant	Attribute that determines others (e.g., A → B)
Candidate Key	A column or combination that can uniquely identify each row
When to Apply BCNF	If a **non-candidate key** is a **determinant**
3NF ≠ BCNF if	There is a non-key determinant
Goal of BCNF	Eliminate all anomalies even from non-key dependencies

✅ TRICKY CONCEPTS TO WATCH FOR:

Tricky Concept	Why It's Confusing
Functional vs Transitive Dependency	FD is direct (A → B), transitive is indirect (A → B → C, so A → C)
2NF applies only to composite keys	If no composite key → 2NF automatically satisfied
BCNF stricter than 3NF	All determinants must be candidate keys, not just PK
Denormalization vs 1NF	Denormalization adds redundancy, 1NF removes multi-valued fields
Multivalued dependency vs Repeating Group	Multivalued is more advanced → applies in 4NF

🎯 TRICKY MCQs

🧠 Q1. Which of the following tables is automatically in 2NF?

A. A table with only atomic columns
B. A table with a single-column primary key
C. A table with transitive dependency
D. A table with composite primary key and partial dependency

✅ **Answer: B**
📌 Because 2NF issues arise **only in composite primary keys**.

🧠 Q2. A → B and B → C. What does this imply?

A. Transitive dependency exists
B. Functional dependency is violated
C. 2NF is maintained
D. Candidate key is B

✅ **Answer: A**
📌 This is a classic **transitive dependency** → must be removed in 3NF.

🧠 **Q3. Which of the following violates 3NF?**

A. Every attribute is fully dependent on PK
B. There is a column dependent on a non-key attribute
C. There is no repeating group
D. All columns are atomic

✅ **Answer: B**
📌 If a column depends on a non-key column, it breaks 3NF → transitive dependency.

🧠 **Q4. Which NF ensures there are no partial and no transitive dependencies?**

A. 2NF
B. 3NF
C. 1NF
D. BCNF

✅ **Answer: B**
📌 2NF removes partial, 3NF removes **both partial and transitive**.

🧠 **Q5. A determinant must be a candidate key is a rule of:**

A. 3NF
B. 2NF
C. BCNF
D. 4NF

✅ **Answer: C**
📌 This is the **strict rule** of BCNF.