1NF (First Normal Form)

Definition:
A table is in First Normal Form (1NF) if:

- **All the data is stored in atomic values (indivisible, single value per cell).**
- **There are no repeating groups or arrays (no multiple values in a single cell).**
- **Each record must be unique (no duplicate rows).**

Example: Table NOT in 1NF

StudentID	StudentName	Courses
1	John	Math, Science
2	Alice	English
3	Bob	Math, History, Art

- **The Courses column has multiple values in a single cell.**
  => This is NOT in 1NF!

How to convert to 1NF?

Make sure one value per cell:

StudentID	StudentName	Course
1	John	Math
1	John	Scienc e
2	Alice	English
3	Bob	Math

3	Bob	History
3	Bob	Art

- Now, every cell has only one value.
- No multi-valued attributes.
- 1NF is achieved!

👉 Tell me when to continue to 2NF explanation! (hit Enter)
(Or just say "continue" if you want.) 🚀

2NF (Second Normal Form)

Definition:
A table is in Second Normal Form (2NF) if:

- It is already in First Normal Form (1NF).
- AND every non-prime attribute (non-primary key column) is fully dependent on the entire primary key, not just a part of it.
  (This mainly applies when we have a composite primary key — a primary key made of two or more columns.)

Key Concept:

- Partial dependency must be removed.
- Partial Dependency means: a non-key column depends only on part of the composite primary key.

Example: Table in 1NF but NOT in 2NF

StudentID	CourseName	InstructorName
1	Math	Mr. Smith

1	Science	Mrs. Taylor
2	English	Mr. Brown
3	History	Mr. Clark

Suppose:

- **Primary Key = (StudentID, CourseName)**

Problem:

- **InstructorName depends only on CourseName, not on StudentID.**
- **So InstructorName is partially dependent on the primary key.**

How to fix and move to 2NF?

Decompose into two tables:

1. Student-Course Table:

StudentID	CourseName
1	Math
1	Science
2	English
3	History

2. Course-Instructor Table:

CourseName	InstructorName
Math	Mr. Smith
Science	Mrs. Taylor
English	Mr. Brown
History	Mr. Clark

- Now, each non-prime attribute (like `InstructorName`) fully depends on the entire primary key of its table.

✅ Now the tables are in 2NF!

👉 Hit Enter when ready for 3NF explanation! 🚀
(Or just say "continue" if you prefer!)

3NF (Third Normal Form)

Definition:
A table is in Third Normal Form (3NF) if:

- It is already in Second Normal Form (2NF).
- AND no non-prime attribute depends transitively on the primary key.

Key Concept:

- Transitive Dependency must be removed.
- Transitive Dependency means:
  Primary Key → Non-key Attribute A → Non-key Attribute B

  (i.e., a non-prime attribute depends on another non-prime attribute, which depends on the primary key.)

Example: Table in 2NF but NOT in 3NF

StudentID	StudentName	DepartmentID	DepartmentName
1	John	10	Computer Science
2	Alice	20	Mathematics
3	Bob	10	Computer Science

Suppose:

- Primary Key = StudentID

Problems:

- `DepartmentID` depends directly on `StudentID` (OK ✅).
- But `DepartmentName` depends on `DepartmentID`, not directly on `StudentID`.
  (StudentID → DepartmentID → DepartmentName)
  => This is a transitive dependency.

How to fix and move to 3NF?

Decompose into two tables:

1. Student Table:

StudentID	StudentName	DepartmentID
1	John	10
2	Alice	20
3	Bob	10

2. Department Table:

DepartmentID	DepartmentName
10	Computer Science
20	Mathematics

- Now, `DepartmentName` depends only on `DepartmentID`.
- No non-prime attribute depends indirectly on the primary key anymore.

✅ Now the tables are in 3NF!

Summary of 1NF, 2NF, 3NF

Normal Form	Rule
1NF	No multi-valued attributes. Only atomic values.
2NF	No partial dependencies on a composite primary key.
3NF	No transitive dependencies on primary key.

✅ Finished explaining 1NF ➔ 2NF ➔ 3NF step-by-step!

Would you also like me to explain BCNF (Boyce-Codd Normal Form)?
(It's a stronger version of 3NF.)
Let me know! 🚀