Database Normalization Report

What is Normalization?

Normalization is the process of organizing data in a database to reduce redundancy and improve data integrity. It involves dividing large tables into smaller, related tables and defining relationships between them using primary and foreign keys.

Types of Normal Forms

1. First Normal Form (1NF)

Definition: A relation is in 1NF if:

- It contains only atomic (indivisible) values.
- There are no repeating groups or arrays.

Use Case: Apply when a table contains multivalued or composite fields.

Example – Before 1NF:

StudentID	Name	Courses
S001	Yahya	Math, English
S002	Salim	Science, History

After Applying 1NF:

StudentID	Name	Course
S001	Yahya	Math
S001	Yahya	English

S002	Salim	Science
S002	Salim	History

2. Second Normal Form (2NF)

Definition: A relation is in 2NF if:

- It is in 1NF.
- There is no partial dependency (i.e., non-key attributes are fully dependent on the entire primary key).

Use Case: When a table has a composite key, and some fields depend only on part of it.

Example – Before 2NF:

StudentID	CourseID	StudentName	CourseName
S001	C01	Mohammed	Math
S001	C02	Mohammed	English

After Applying 2NF: Student Table:

StudentID	StudentName
S001	Mohammed

Course Table:

CourseID	CourseName
C01	Math
C02	English

Enrollment Table:

StudentID	CourseID
S001	C01
S001	C02

3. Third Normal Form (3NF)

Definition: A relation is in 3NF if:

- It is in 2NF.
- There are no transitive dependencies.

Use Case: When non-key fields are indirectly dependent on the primary key.

Example – Before 3NF:

EmployeeID	Name	DeptID	DeptName
E001	Sami	D01	HR
E002	Ali	D02	Finance

After Applying 3NF: Employee Table:

EmployeeID	Name	DeptID
E001	Sami	D01
E002	Ali	D02

Department Table:

DeptID	DeptName
D01	HR
D02	Finance

4. Boyce-Codd Normal Form (BCNF)

Definition: A stricter version of 3NF. A table is in BCNF if:

- Every determinant is a candidate key.

Use Case: When a table has multiple candidate keys, and one determines another non-key attribute.

Example – Before BCNF:

Course	Instructor	Room
DBMS	Ahmed	R101
DBMS	Ahmed	R102

After BCNF: Course Table:

Course	Instructor
DBMS	Ahmed

RoomAssignment Table:

Room	Course
R101	DBMS
R102	DBMS

5. Fourth Normal Form (4NF)

Definition: A table is in 4NF if:

- It is in BCNF.
- It has no multivalued dependencies.

Use Case: When a record contains independent multivalued facts.

Example – Before 4NF:

Student	Skill	Hobby
Ali	Python	Chess
Ali	Python	Music
Ali	Java	Chess
Ali	Java	Music

After 4NF: StudentSkill Table:

Student	Skill
Ali	Python
Ali	Java

StudentHobby Table:

Student	Hobby
Ali	Chess
Ali	Music

6. Fifth Normal Form (5NF)

Definition: A table is in 5NF if:

- It is in 4NF.
- Every join dependency is implied by candidate keys.

Use Case: Very rare. Used when complex joins create redundancy.

De-Normalization

What is De-Normalization? De-normalization is the process of combining tables that were split during normalization to improve performance.

Why Apply De-Normalization?

- To reduce complex joins that affect performance.
- To improve query speed for read-heavy applications.
- To simplify report generation.

When to Apply De-Normalization?

- When performance optimization is critical.
- In data warehousing and reporting systems.
- For read-heavy applications with complex joins.

Example: Instead of having:

- Orders
- OrderDetails
- Customers

We can have a single table:

OrderID	CustomerName	ProductName	Quantity
O001	Mohammed	Pen	10
O001	Mohammed	Book	2

This reduces joins but increases redundancy.

Summary Table

Normal Form	Eliminates	Focus
1NF	Repeating groups	Atomicity

2NF	Partial dependencies	Composite keys
3NF	Transitive dependencies	Non-key dependencies
BCNF	Non-candidate key determinants	Stronger form of 3NF
4NF	Multivalued dependencies	Independent multivalued facts
5NF	Join dependencies	Complex relationships
De-Normalization	Too many joins (performance)	Speed over redundancy