Normalisation

What is it?

Normalisation is a process used in relational database design to minimise data redundancy and ensure data integrity.

It involves organising database tables in a way that reduces duplication and improves efficiency.

The process is guided by a series of rules known as normal forms (NFs).

Normalisation

Why do we need it?

1. Eliminates Redundant Data – Reduces storage space and avoids duplication.
2. Improves Data Integrity – Ensures data consistency by reducing anomalies in insertion, deletion, and updates.
3. Enhances Query Performance – Optimises searches and improves efficiency.
4. Enforces Relationships Between Tables – Ensures that data dependencies are logical.

Normalisation

Functional Dependency

A functional dependency occurs when the value of one attribute (column) uniquely determines the value of another attribute.

Definition:

If A → B, then knowing A allows us to determine B uniquely.

Normalisation

Partial Dependency

A partial dependency occurs when a non-key attribute depends on part of a composite primary key instead of the entire key.
This happens in 1NF and is removed in 2NF.

Definition -

If a primary key consists of (A, B) but an attribute C depends only on A, then C is partially dependent on the key.

Normalisation

Transitive Dependency

A transitive dependency occurs when a non-key attribute depends on another non-key attribute, rather than depending directly on the primary key.
This is found in 2NF and removed in 3NF.

Definition:
If A → B and B → C, then A → C is a transitive dependency.

The Normalisation Process

First Normal Form (1NF) – Eliminating Repeating Groups

A table is in 1NF if:

1. **Each column contains no multiple values in a single column (atomic values).**

2. **Each row is unique (a primary key should be identified).**

First Normal Form (1NF) - Unnormalised VS Normalised

StudentID	StudentName	Subjects
1	Alice	Math, Science
2	Bob	Science
3	Charlie	Math, English

First Normal Form (1NF) - Unnormalised VS Normalised

StudentID	StudentName	Subjects
1	Alice	Math
1	Alice	Science
2	Bob	Science
3	Charlie	Math
3	Charlie	English

Second Normal Form (2NF) – Removing Partial Dependencies

A table is in 2NF if:

1. **It is already in 1NF.**
2. **All non-key attributes must depend on the entire primary key, not just part of it.**

Second Normal Form (2NF)

Problem: Student_Name depends on Student_ID, but not on Course_ID.

StudentID	CourseID	CourseName	StudentName
1	C001	Database Design	Alice
2	COO2	Web Development	Bob

Second Normal Form (2NF)

Solution (2NF): Separate the tables.

StudentID	StudentName
1	Alice
2	Bob

CourseID	CourseName
C001	Database Design
COO2	Web Development

StudentID	CourseID
1	C001
2	COO2

Third Normal Form (3NF) – Removing Transitive Dependencies

A table is in 3NF if:

1. **It is already in 2NF.**
2. **No non-key column should depend on another non-key column (no transitive dependency).**

Third Normal Form (3NF)

Problem: City depends on Postal_Code, not on Student_ID.

StudentID	StudentName	City	PostalCode
1	Alice	Durban	4001
2	Bob	Cape Town	8000

Third Normal Form (3NF)

Solution (3NF): Create a separate City Table.

StudentID	StudentName	PostalCode
1	Alice	4001
2	Bob	8000

City	PostalCode
Durban	4001
Cape Town	8000

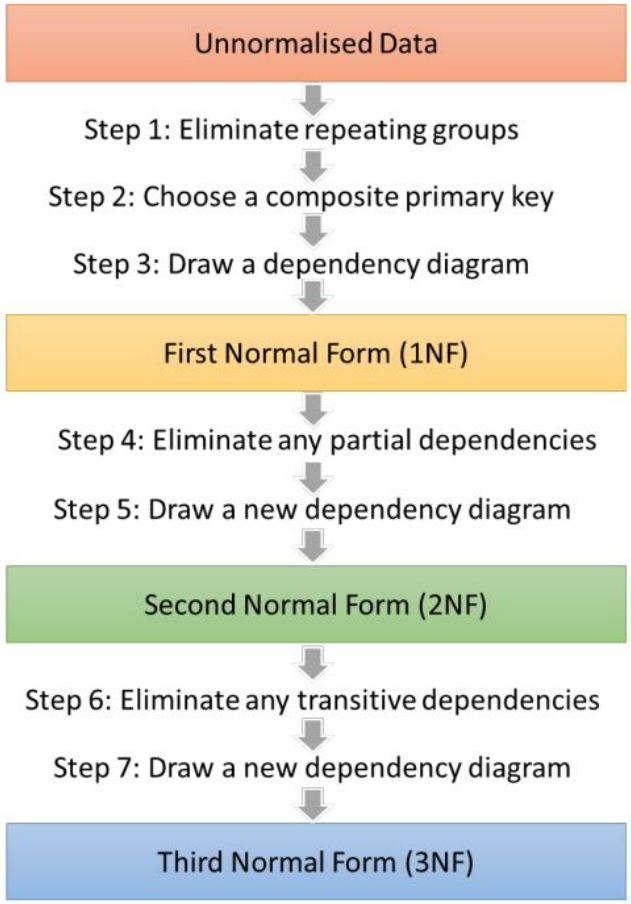

Figure 41. Normalisation Process, based on the process as described by (Coronel, et al., 2013)