

[Home](#) [DBMS](#) [SQL](#) [PL/SQL](#) [SQLite](#) [MongoDB](#) [Cassandra](#) [MySQL](#) [Oracle](#) [CouchDB](#) [Neo4j](#) [DB2](#) [C](#) [Java](#) [Projects](#)

|

↑ SCROLL TO TOP

Normalization

A large database defined as a single relation may result in data duplication. This repetition of data may result in:

- Making relations very large.
- It isn't easy to maintain and update data as it would involve searching many records in relation.
- Wastage and poor utilization of disk space and resources.
- The likelihood of errors and inconsistencies increases.

So to handle these problems, we should analyze and decompose the relations with redundant data into smaller, simpler, and well-structured relations that satisfy desirable properties. Normalization is a process of decomposing the relations into relations with fewer attributes.

What is Normalization?

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate undesirable characteristics like Insertion, Update, and Deletion Anomalies.
- Normalization divides the larger table into smaller and links them using relationships.
- The normal form is used to reduce redundancy from the database table.



Why do we need Normalization?

The main reason for normalizing the relations is removing these anomalies. Failure to eliminate anomalies leads to data redundancy and can cause data integrity and other problems as the database grows. Normalization consists of a series of guidelines that helps to guide you in creating a good database structure.



Data modification anomalies can be categorized into three types:

- **Insertion Anomaly:** Insertion Anomaly refers to when one cannot insert a new tuple into a relationship due to lack of data.
- **Deletion Anomaly:** The delete anomaly refers to the situation where the deletion of data results in the unintended loss of some other important data.
- **Updation Anomaly:** The update anomaly is when an update of a single data value requires multiple rows of data to be updated.



Types of Normal Forms:

Normalization works through a series of stages called Normal forms. The normal forms apply to individual relations. The relation is said to be in particular normal form if it satisfies constraints.

Following are the various types of Normal forms:

	1NF	2NF	3NF	4NF	5NF
Decomposition of Relation	R	R ₁₁ R ₁₂	R ₂₁ R ₂₂ R ₂₃	R ₃₁ R ₃₂ R ₃₃ R ₃₄	R ₄₁ R ₄₂ R ₄₃ R ₄₄ R ₄₅
Conditions	Eliminate Repeating Groups	Eliminate Partial Functional Dependency	Eliminate Transitive Dependency	Eliminate Multi-values Dependency	Eliminate Join Dependency

Normal Form	Description
1NF	A relation is in 1NF if it contains an atomic value.
2NF	A relation will be in 2NF if it is in 1NF and all non-key attributes are fully functional dependent on the primary key.

3NF	A relation will be in 3NF if it is in 2NF and no transition dependency exists.
BCNF	A stronger definition of 3NF is known as Boyce Codd's normal form.
4NF	A relation will be in 4NF if it is in Boyce Codd's normal form and has no multi-valued dependency.
5NF	A relation is in 5NF. If it is in 4NF and does not contain any join dependency, joining should be lossless.

Advantages of Normalization

- Normalization helps to minimize data redundancy.
- Greater overall database organization.
- Data consistency within the database.
- Much more flexible database design.
- Enforces the concept of relational integrity.

Disadvantages of Normalization

- You cannot start building the database before knowing what the user needs.
- The performance degrades when normalizing the relations to higher normal forms, i.e., 4NF, 5NF.

- It is very time-consuming and difficult to normalize relations of a higher degree.
- Careless decomposition may lead to a bad database design, leading to serious problems.

[< Prev](#)[Next >](#)

 For Videos Join Our Youtube Channel: [Join Now](#)

Feedback

- Send your Feedback to feedback@javatpoint.com

Help Others, Please Share



Learn Latest Tutorials



[Python Pillow](#)[Python Turtle](#)[Keras](#)

Preparation

[Aptitude](#)[Aptitude](#)[Logical Reasoning](#)[Reasoning](#)[Verbal Ability](#)[Verbal Ability](#)[Interview Questions](#)[Interview Questions](#)[Company Interview Questions](#)[Company Questions](#)

Trending Technologies

[Artificial Intelligence Tutorial](#)[Artificial Intelligence](#)[AWS Tutorial](#)[AWS](#)[Selenium tutorial](#)[Selenium](#)[Cloud Computing tutorial](#)[Hadoop tutorial](#)[Hadoop](#)[ReactJS Tutorial](#)[ReactJS](#)

Cloud Computing

 Data Science
Tutorial

Data Science


 Angular 7
Tutorial

Angular 7

 Blockchain
Tutorial


Blockchain


 Git Tutorial
Git

 Machine
Learning Tutorial
Machine Learning


 DevOps
Tutorial
DevOps


B.Tech / MCA

 DBMS tutorial Data Structures
tutorial DAA tutorial Operating
System tutorial


Operating System Computer
Network tutorial

Computer Network Compiler
Design tutorial


Compiler Design Computer
Organization and
Architecture

Computer
Organization Discrete
Mathematics
Tutorial


Discrete
Mathematics Ethical Hacking
Tutorial


Ethical Hacking Computer
Graphics Tutorial


Computer Graphics Software
Engineering
Tutorial


Software
Engineering html tutorial




Web Technology Cyber Security
tutorial

Cyber Security Automata
Tutorial

Automata C Language
tutorial

C Programming C++ tutorial Java tutorial

 C++ Java .Net
Framework
tutorial

.Net Python tutorial
Python List of
Programs
Programs Control
Systems tutorial
Control System Data Mining
Tutorial
Data Mining Data
Warehouse
Tutorial
Data Warehouse