# Database Normalization Techniques in Relational Database Systems

**Hareesh Edupuganti**

*Engineering Manager, Cerner Corporation, Kansas City, MO, USA*

**ABSTRACT**

Relational Database Management Systems (RDBMS) rely on SQL as the standard language for defining, manipulating, and controlling data. Popular systems such as MySQL, Oracle, SQL Server, MS Access, Sybase, and Postgres all implement SQL to ensure structured data management. One of the most critical aspects of relational database design is normalization, a systematic approach to organizing data in order to reduce redundancy, enforce data integrity, and improve efficiency. This paper provides an in-depth study of SQL concepts in the context of RDBMS and focuses on the principles and techniques of database normalization. It highlights the various normal forms, their role in eliminating anomalies, and their importance in maintaining consistency within relational systems.

**Index Terms**: RDBMS, SQL, database normalization, data integrity

## I. OVERVIEW OF SQL

SQL is a language to operate databases; it includes database creation, deletion, fetching rows, modifying rows, etc. SQL is an **ANSI** (American National Standards Institute) standard language, but there are many different versions of the SQL language.

### What is SQL?

SQL is Structured Query Language, which is a computer language for storing, manipulating and retrieving data stored in a relational database.

Also, they are using different dialects, such as:

- MS SQL Server using T-SQL,
- Oracle using PL/SQL,
- MS Access version of SQL is called JET SQL (native format) etc.

### Why SQL?

SQL is widely popular because it offers the following advantages:

- Allows users to access data in the relational database management systems.

- Allows users to describe the data.
- Allows users to define the data in a database and manipulate that data.
- Allows to embed within other languages using SQL modules, libraries & pre-compilers.
- Allows users to create and drop databases and tables.
- Allows users to create view, stored procedure, functions in a database.
- Allows users to set permissions on tables, procedures and views.

**A Brief History of SQL**

- **1970 –** Dr. Edgar F. "Ted" Codd of IBM is known as the father of relational databases. He described a relational model for databases.
- **1974 –** Structured Query Language appeared.
- **1978 –** IBM worked to develop Codd's ideas and released a product named System/R.
- **1986 –** IBM developed the first prototype of relational database and standardized by ANSI. The first relational database was released by Relational Software which later came to be known as Oracle.

## II. SQL PROCESS

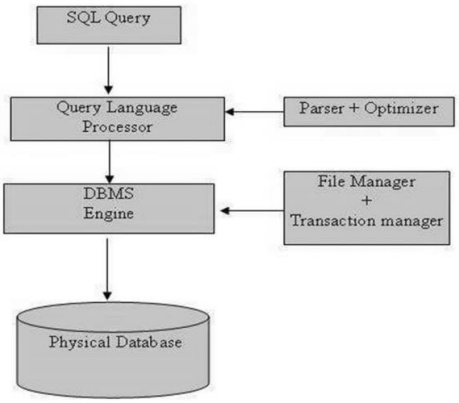
When you are executing an SQL command for any RDBMS, the system determines the best way to carry out your request and SQL engine figures out how to interpret the task. There are various components included in this process.

These components are –

- Query Dispatcher
- Optimization Engines
- Classic Query Engine
- SQL Query Engine, etc.

A classic query engine handles all the non-SQL queries, but a SQL query engine won't handle logical files.

Following is a simple diagram showing the SQL Architecture:



## III. SQL COMMANDS

The standard SQL commands to interact with relational databases are CREATE, SELECT, INSERT, UPDATE, DELETE and DROP. These commands can be classified into the following groups based on their nature:

DDL - Data Definition Language

| Command | Description |
|---|---|
| CREATE | Creates a new table, a view of a table, or other object in the database. |
| ALTER | Modifies an existing database object, such as a table. |
| DROP | Deletes an entire table, a view of a table or other objects in the database. |

**DML - Data Manipulation Language**

| Command | Description |
|---|---|
| SELECT | Retrieves certain records from one or more tables. |
| INSERT | Creates a record. |
| UPDATE | Modifies records. |
| DELETE | Deletes records. |

**DCL - Data Control Language**

| Command | Description |
|---|---|
| GRANT | Gives a privilege to user. |
| REVOKE | Takes back privileges granted from user. |

## IV.SQL - RDBMS CONCEPTS

**What is a table?**

The data in an RDBMS is stored in database objects which are called as **tables**. This table is basically a collection of related data entries and it consists of numerous columns and rows.

**What is a field?**

Every table is broken up into smaller entities called fields. The fields in the CUSTOMERS table consist of ID, NAME, AGE, ADDRESS and SALARY.

A field is a column in a table that is designed to maintain specific information about every record in the table.

**What is a column?**

A column is a vertical entity in a table that contains all information associated with a specific field in a table.

**What is a NULL value?**

A NULL value in a table is a value in a field that appears to be blank, which means a field with a NULL value is a field with no value.

It is very important to understand that a NULL value is different than a zero value or a field that contains spaces. A field with a NULL value is the one that has been left blank during a record creation.

## V. SQL CONSTRAINTS

Constraints are the rules enforced on data columns on a table. These are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the database.

Constraints can either be column level or table level. Column level constraints are applied only to one column whereas, table level constraints are applied to the entire table.

Following are some of the most commonly used constraints available in SQL:

- NOT NULL Constraint: Ensures that a column cannot have a NULL value.
- DEFAULT Constraint: Provides a default value for a column when none is specified.
- UNIQUE Constraint: Ensures that all the values in a column are different.
- PRIMARY Key: Uniquely identifies each row/record in a database table.
- FOREIGN Key: Uniquely identifies a row/record in any another database table.
- CHECK Constraint: The CHECK constraint ensures that all values in a column satisfy certain conditions.
- INDEX: Used to create and retrieve data from the database very quickly.

## VI. DATA INTEGRITY

The following categories of data integrity exist with each RDBMS:

- **Entity Integrity:** There are no duplicate rows in a table.
- **Domain Integrity:** Enforces valid entries for a given column by restricting the type, the format, or the range of values.

- **Referential integrity:** Rows cannot be deleted, which are used by other records.
- **User-Defined Integrity:** Enforces some specific business rules that do not fall into entity, domain or referential integrity.

## VII. DATABASE NORMALIZATION

Database normalization is the process of efficiently organizing data in a database. There are two reasons of this normalization process:

- Eliminating redundant data. For example, storing the same data in more than one table.
- Ensuring data dependencies make sense.
-

Both these reasons are worthy goals as they reduce the amount of space a database consumes and ensures that data is logically stored. Normalization consists of a series of guidelines that help guide you in creating a good database structure.

Normalization guidelines are divided into normal forms; think of a form as the format or the way a database structure is laid out. The aim of normal forms is to organize the database structure, so that it complies with the rules of first normal form, then second normal form and finally the third normal form.

It is your choice to take it further and go to the fourth normal form, fifth normal form and so on, but in general, the third normal form is more than enough.

- First Normal Form (1NF)
- Second Normal Form (2NF)
- Third Normal Form (3NF)

The advantages of removing transitive dependencies are mainly two-fold. First, the amount of data duplication is reduced and therefore your database becomes smaller.

The second advantage is data integrity. When duplicated data changes, there is a big risk of updating only some of the data, especially if it is spread out in many different places in the database.

For example, if the address and the zip code data were stored in three or four different tables, then any changes in the zip codes would need to ripple out to every record in those three or four tables.

## VIII. CONCLUSION

RDBMS stands for **R**elational **D**atabase **M**anagement **S**ystem. RDBMS is the basis for SQL, and for all modern database systems like MS SQL Server, IBM DB2, Oracle, MySQL, and Microsoft Access. A Relational database management system (RDBMS) is a database management system (DBMS) that is based on the relational model as introduced by E. F. Codd.

## REFERENCES

[1] Blaha, Michael R. *A Manager's Guide to Database Technology.* Upper Saddle River, NJ: Prentice Hall (2001). ISBN 0-13-030418-2.

[2] Connolly, Thomas, and Carolyn Begg. *Database Systems: A Practical Approach to Design, Implementation, and Management.* 4th ed. Har- low, England: Addison-Wesley (2004). ISBN 978-0-321-29401-2.