

# **Basics of Database Management System**

**Prince Goswami**

## Table of Contents

<b>Overview of DBMS</b>	<b>2</b>
<b>MySQL</b>	<b>2</b>
<b>Overview of Workbench</b>	<b>2</b>
<b>Database Design</b>	<b>3</b>
Importance of Database Design	4
Database Development Life Cycle	4
<b>SQL Basics</b>	<b>5</b>
<b>Data Sorting</b>	<b>5</b>
ORDER BY	5
ASC	5
DESC	5
<b>Null Value &amp; Keyword</b>	<b>6</b>
Keywords	6
<b>Auto-Increment</b>	<b>8</b>
<b>SQL Commands ( DDL, DML, DCL, TCL, DQL ) :</b>	<b>9</b>
DDL ( Data Definition Language )	9
DML ( Data Manipulation Language )	10
<b>DCL ( Data Control Language )</b>	<b>10</b>
TCL ( Transaction Control Language )	10
DQL ( Data Query Language )	11
<b>Limit</b>	<b>11</b>
<b>Aggregate Function</b>	<b>11</b>
COUNT FUNCTION	12
SUM Function	12
AVG function	12
MAX Function	12
MIN Function	12
<b>SubQueries</b>	<b>12</b>
<b>Joins</b>	<b>13</b>
INNER JOIN	13
LEFT JOIN	13
RIGHT JOIN	14
FULL JOIN	14
Natural join (?)	15
<b>UNIONS</b>	<b>15</b>

## Overview of DBMS

A Database Management System (DBMS) is a software system that is designed to manage and organize data in a structured manner. It allows users to create, modify, and query a database, as well as manage the security and access controls for that database.

### Key Features of DBMS

1. **Data modeling:** A DBMS provides tools for creating and modifying data models, which define the structure and relationships of the data in a database.
2. **Data storage and retrieval:** A DBMS is responsible for storing and retrieving data from the database, and can provide various methods for searching and querying the data.
3. **Concurrency control:** A DBMS provides mechanisms for controlling concurrent access to the database, to ensure that multiple users can access the data without conflicting with each other.
4. **Data integrity and security:** A DBMS provides tools for enforcing data integrity and security constraints, such as constraints on the values of data and access controls that restrict who can access the data.
5. **Backup and recovery:** A DBMS provides mechanisms for backing up and recovering the data in the event of a system failure.
6. **DBMS can be classified into two types:** Relational Database Management System (RDBMS) and Non-Relational Database Management System (NoSQL or Non-SQL)
7. **RDBMS:** Data is organized in the form of tables and each table has a set of rows and columns. The data are related to each other through primary and foreign keys.
8. **NoSQL:** Data is organized in the form of key-value pairs, documents, graphs, or column-based. These are designed to handle large-scale, high-performance scenarios.

## MySQL

MySQL is a very popular open-source relational database management system (RDBMS).

### What is MySQL :

- MySQL is a relational database management system
- MySQL is open-source
- MySQL is free
- MySQL is ideal for both small and large applications
- MySQL is very fast, reliable, scalable, and easy to use
- MySQL is cross-platform
- MySQL is compliant with the ANSI SQL standard
- MySQL was first released in 1995
- MySQL is developed, distributed, and supported by Oracle Corporation

## Overview of Workbench

MySQL Workbench is a unified visual tool for database architects, developers, and DBAs. MySQL Workbench provides data modeling, SQL development, and comprehensive administration tools for server configuration, user administration, backup, and much more

- **Design**
  - MySQL Workbench enables a DBA, developer, or data architect to visually design, model, generate, and manage databases. It includes everything a data modeler needs for creating complex ER models, forward and reverse engineering, and also delivers key features for performing difficult change management and documentation tasks that normally require much time and effort.
- **Develop**
  - MySQL Workbench delivers visual tools for creating, executing, and optimizing SQL queries. The SQL Editor provides color syntax highlighting, auto-complete, reuse of SQL snippets, and execution history of SQL. The Database Connections Panel enables developers to easily manage standard database connections, including MySQL Fabric. The Object Browser provides instant access to database schema and objects.
- **Administer**
  - MySQL Workbench provides a visual console to easily administer MySQL environments and gain better visibility into databases. Developers and DBAs can use the visual tools for configuring servers, administering users, performing backup and recovery, inspecting audit data, and viewing database health.

## Database Design

Database design is a collection of steps that help create, implement, and maintain a business's data management systems. The primary purpose of designing a database is to produce physical and logical models of designs for the proposed database system.

Specific rules govern a good database design process.

- The first rule in creating a database design is to avoid data redundancy. It wastes space and increases the probability of faults and discrepancies within the database.
- The second rule is that the accuracy and comprehensiveness of information are imperative.
- A database containing erroneous information will lead to inaccurate analysis and reporting. Consequently, it can mislead decision-makers and adversely affect a company's performance.

### **A well-designed database is one that:**

- Distributes your data into tables based on specific subject areas to decrease data redundancy
- Delivers the database the information needed to link the data in the tables
- Provides support and guarantees the precision and reliability of data
- Caters to your information processing and reporting requirements
- Functions interactively with the database operators

## Importance of Database Design

Database design defines the database structure used for planning, storing, and managing information. To ensure data accuracy, you must design a database that only stores relevant and valuable information.

A well-designed database is essential to guarantee information consistency, eliminate redundant data, efficiently execute queries, and improve the database's performance. A methodological approach toward designing a database will save you time in the database development phase.

## Database Development Life Cycle

### 1- Requirement analysis

Requirement analysis requires two steps:

- Planning: In this stage of database development, the entire Database Development Life Cycle plan is decided. It also requires an analysis of the organization's information systems strategy.
- Defining the system: This stage explains the proposed database system's scope.

### 2- Database designing

The actual database design takes into account two fundamental data models:

- Logical model: It uses the given requirements to create a database model. The complete structure is laid out on paper at this stage, without considering any specific database management system (DBMS) requirement or physically implementing it.
- Physical model: This stage comes after the logical model and therefore involves physically implementing the logical model. It takes the DBMS and other physical implementation factors into consideration.

### 3- Implementation

The implementation phase of the database development life cycle is concerned with:

- Data conversion and loading involve importing and converting data from the old system into the new database.
- Testing: Finally, this stage identifies errors in the new system and meets all the database requirement specifications.

## SQL Basics

SQL is a standard programming language used to operate Relational Databases and carry out various operations such as inserting, manipulating, updating, and retrieving data from relational databases.

### SQL is required for the following tasks

- To create new databases, tables, and views
- To insert records in a database
- To update records in a database
- To delete records from a database
- To retrieve data from a database

### Applications of SQL

- SQL is used to create a database, define its structure, implement it, and perform various functions on the database.
- SQL is also used for accessing, maintaining, and manipulating already created databases.
- SQL is a well built language for entering data, modifying data, and extracting data in a database.

### Features of SQL

- SQL is used to access data within the relational database.
- SQL is very fast in extracting large amounts of data very efficiently.
- SQL is flexible as it works with multiple database systems from Oracle, IBM, Microsoft, etc.
- SQL helps you manage databases without knowing a lot of coding.

## Data Sorting

### ORDER BY

The ORDER BY statement in SQL is used to sort the fetched data in either ascending or descending according to one or more columns. In this article, we will discuss different ways of using Order By in SQL.

Here are some basic rules of Order By statement in SQL.

- By default ORDER BY sorts the data in ascending order.
- We can use the keyword DESC to sort the data in descending order and the keyword ASC to sort in ascending order.

### ASC

The ASC command is used to sort the data returned in ascending order.

### DESC

The DESC command is used to sort the data returned in descending order.

## Null Value & Keyword

### Keywords

SQL keywords are a crucial component of the SQL language and are used to perform specific tasks such as retrieving, filtering, and managing data in a database. These reserved words have special meanings and are used to define data structures, perform operations, and specify conditions.

Keyword	Description
ADD	Adds a column in an existing table
ADD CONSTRAINT	Adds a constraint after a table is already created
ALL	Returns true if all of the subquery values meet the condition
ALTER	Adds, deletes, or modifies columns in a table, or changes the data type of a column in a table
ALTER COLUMN	Changes the data type of a column in a table
ALTER TABLE	Adds, deletes, or modifies columns in a table
AND	Only includes rows where both conditions is true
ANY	Returns true if any of the subquery values meet the condition
AS	Renames a column or table with an alias
ASC	Sorts the result set in ascending order
BACKUP DATABASE	Creates a back up of an existing database
BETWEEN	Selects values within a given range
CASE	Creates different outputs based on conditions
CHECK	A constraint that limits the value that can be placed in a column
COLUMN	Changes the data type of a column or deletes a column in a table
CONSTRAINT	Adds or deletes a constraint
CREATE	Creates a database, index, view, table, or procedure
CREATE DATABASE	Creates a new SQL database
CREATE INDEX	Creates an index on a table (allows duplicate values)
CREATE OR REPLACE VIEW	Updates a view
CREATE TABLE	Creates a new table in the database
CREATE PROCEDURE	Creates a stored procedure
CREATE UNIQUE INDEX	Creates a unique index on a table (no duplicate values)
CREATE VIEW	Creates a view based on the result set of a SELECT statement

DATABASE	Creates or deletes an SQL database
DEFAULT	A constraint that provides a default value for a column
DELETE	Deletes rows from a table
DESC	Sorts the result set in descending order
DISTINCT	Selects only distinct (different) values
DROP	Deletes a column, constraint, database, index, table, or view
DROP COLUMN	Deletes a column in a table
DROP CONSTRAINT	Deletes a UNIQUE, PRIMARY KEY, FOREIGN KEY, or CHECK constraint
DROP DATABASE	Deletes an existing SQL database
DROP DEFAULT	Deletes a DEFAULT constraint
DROP INDEX	Deletes an index in a table
DROP TABLE	Deletes an existing table in the database
DROP VIEW	Deletes a view
EXEC	Executes a stored procedure
EXISTS	Tests for the existence of any record in a subquery
FOREIGN KEY	A constraint that is a key used to link two tables together
FROM	Specifies which table to select or delete data from
FULL OUTER JOIN	Returns all rows when there is a match in either left table or right table
GROUP BY	Groups the result set (used with aggregate functions: COUNT, MAX, MIN, SUM, AVG)
HAVING	Used instead of WHERE with aggregate functions
IN	Allows you to specify multiple values in a WHERE clause
INDEX	Creates or deletes an index in a table
INNER JOIN	Returns rows that have matching values in both tables
INSERT INTO	Inserts new rows in a table
INSERT INTO SELECT	Copies data from one table into another table
IS NULL	Tests for empty values
IS NOT NULL	Tests for non-empty values
JOIN	Joins tables
LEFT JOIN	Returns all rows from the left table, and the matching rows from the right table
LIKE	Searches for a specified pattern in a column
LIMIT	Specifies the number of records to return in the result set
NOT	Only includes rows where a condition is not true
NOT NULL	A constraint that enforces a column to not accept NULL values
OR	Includes rows where either condition is true
ORDER BY	Sorts the result set in ascending or descending order



OUTER JOIN	Returns all rows when there is a match in either left table or right table
PRIMARY KEY	A constraint that uniquely identifies each record in a database table
PROCEDURE	A stored procedure
RIGHT JOIN	Returns all rows from the right table, and the matching rows from the left table
ROWNUM	Specifies the number of records to return in the result set
SELECT	Selects data from a database
SELECT DISTINCT	Selects only distinct (different) values
SELECT INTO	Copies data from one table into a new table
SELECT TOP	Specifies the number of records to return in the result set
SET	Specifies which columns and values that should be updated in a table
TABLE	Creates a table, or adds, deletes, or modifies columns in a table, or deletes a table or data inside a table
TOP	Specifies the number of records to return in the result set
TRUNCATE TABLE	Deletes the data inside a table, but not the table itself
UNION	Combines the result set of two or more SELECT statements (only distinct values)
UNION ALL	Combines the result set of two or more SELECT statements (allows duplicate values)
UNIQUE	A constraint that ensures that all values in a column are unique
UPDATE	Updates existing rows in a table
VALUES	Specifies the values of an INSERT INTO statement
VIEW	Creates, updates, or deletes a view
WHERE	Filters a result set to include only records that fulfill a specified condition

## Auto-Increment

Auto-increment allows a unique number to be generated automatically when a new record is inserted into a table.

Often this is the primary key field that we would like to be created automatically every time a new record is inserted.

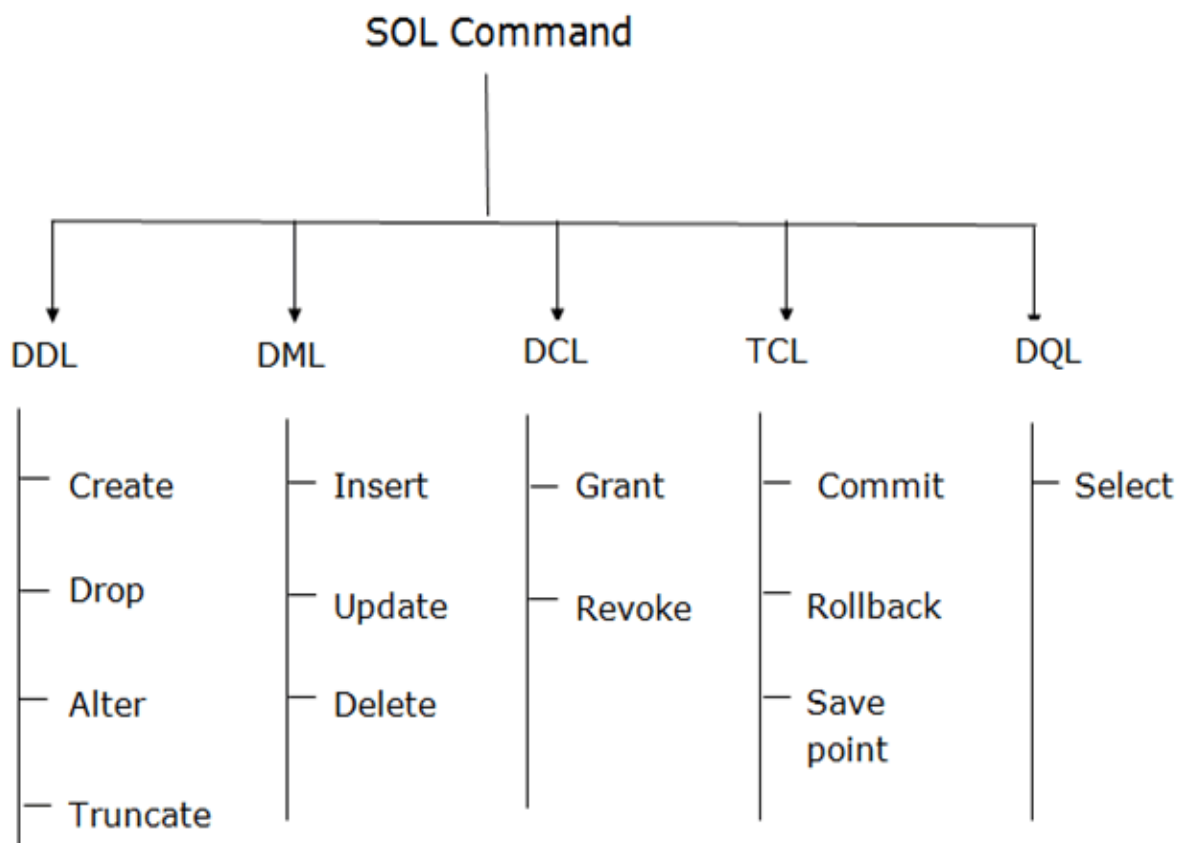
```
CREATE TABLE Persons (
  Personid int IDENTITY(1,1) PRIMARY KEY,
  LastName varchar(255) NOT NULL,
  FirstName varchar(255),
  Age int
);
```

## SQL Commands ( DDL, DML, DCL, TCL, DQL ) :

SQL commands are instructions. It is used to communicate with the database. It is also used to perform specific tasks, functions, and queries of data. SQL can perform various tasks like create a table, add data to tables, drop the table, modify the table, set permission for users.

There are 5 types of SQL commands :

1. DDL
2. DML
3. DCL
4. TCL
5. DQL



### DDL ( Data Definition Language )

DDL changes the structure of the table like creating a table, deleting a table, altering a table, etc.

All the command of DDL are auto-committed that means it permanently save all the changes in the database.

Here are some commands that come under DDL:

**CREATE** - It is used to create a new table in the database.

- **CREATE TABLE TABLE\_NAME (COLUMN\_NAME DATATYPES[,...]);**

**ALTER** - It is used to alter the structure of the database.

- **DROP TABLE table\_name;**

**DROP** - It is used to delete both the structure and record stored in the table.

- ***ALTER TABLE table\_name ADD column\_name COLUMN-definition;***

**TRUNCATE** - It is used to delete all the rows from the table.

- ***TRUNCATE TABLE table\_name;***

## **DML ( Data Manipulation Language )**

DML commands are used to modify the database. It is responsible for all form of changes in the database.

The command of DML is not auto-committed that means it can't permanently save all the changes in the database. They can be rollback.

**INSERT** - It is used to insert data into the row of a table.

- ***INSERT INTO TABLE\_NAME  
(col1, col2, col3,... col N)  
VALUES (value1, value2, value3, .... valueN);***

**UPDATE** - This command is used to update or modify the value of a column in the table.

- ***UPDATE table\_name SET [column\_name1= value1,...column\_nameN = valueN]  
[WHERE CONDITION]***

**DELETE** - It is used to remove one or more row from a table.

- ***DELETE FROM table\_name [WHERE condition];***

## **DCL ( Data Control Language )**

DCL commands are used to grant and take back authority from any database user.

Here are some commands that come under DCL:

- **Grant** - It is used to give user access privileges to a database.
  - ***GRANT SELECT, UPDATE ON MY\_TABLE TO SOME\_USER, ANOTHER\_USER;***
- **Revoke** - It is used to take back permissions from the user.
  - ***REVOKE SELECT, UPDATE ON MY\_TABLE FROM USER1, USER2;***

## **TCL ( Transaction Control Language )**

TCL commands can only use with DML commands like INSERT, DELETE and UPDATE only.

These operations are automatically committed in the database that's why they cannot be used while creating tables or dropping them.

Here are some commands that come under TCL:

- **COMMIT** - Commit command is used to save all the transactions to the database.
  - ***COMMIT***
- **ROLLBACK** - Rollback command is used to undo transactions that have not already been saved to the database.
  - ***ROLLBACK***
- **SAVEPOINT** - It is used to roll the transaction back to a certain point without rolling back the entire transaction.

- **SAVEPOINT SAVEPOINT\_NAME**

## DQL ( Data Query Language )

DQL is used to fetch the data from the database.

It uses only one command:

- SELECT - It is used to select the attribute based on the condition described by WHERE clause.
  - **SELECT expressions**
  - **FROM TABLES**
  - **WHERE conditions;**

## Limit

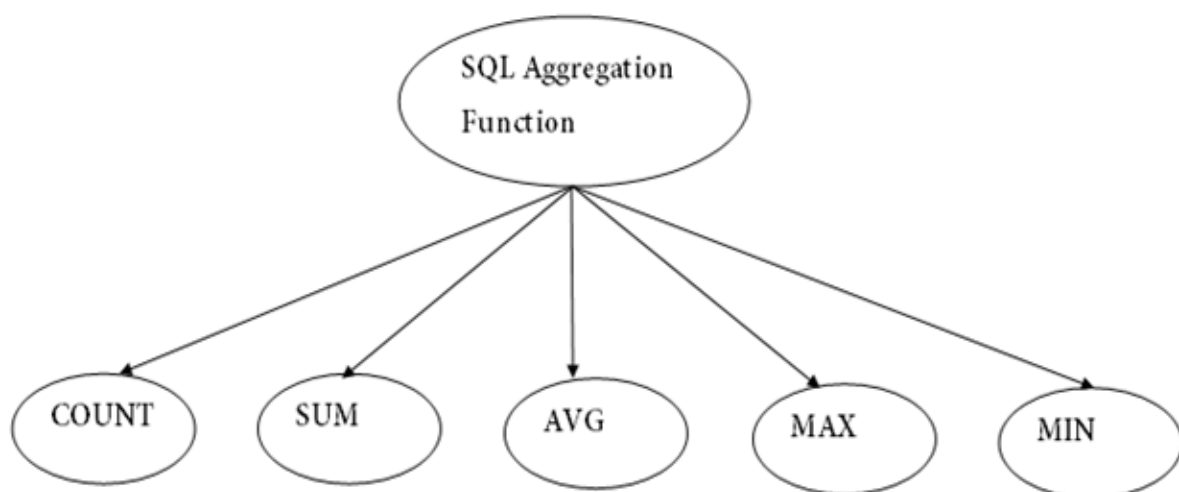
The LIMIT clause is used to specify the number of records to return.

The LIMIT clause is useful on large tables with thousands of records. Returning a large number of records can impact performance.

```
SELECT column_name(s)  
FROM table_name  
WHERE condition  
LIMIT number;
```

## Aggregate Function

- SQL aggregation function is used to perform the calculations on multiple rows of a single column of a table. It returns a single value.
- It is also used to summarize the data.



## COUNT FUNCTION

- COUNT function is used to Count the number of rows in a database table. It can work on both numeric and non-numeric data types.
- COUNT function uses the COUNT(\*) that returns the count of all the rows in a specified table. COUNT(\*) considers duplicate and Null.

**COUNT(\*)**  
**or**  
**COUNT( [ALL | DISTINCT] expression )**

## SUM Function

- Sum function is used to calculate the sum of all selected columns. It works on numeric fields only.

**SUM()**  
**or**  
**SUM( [ALL | DISTINCT] expression )**

## AVG function

The AVG function is used to calculate the average value of the numeric type. AVG function returns the average of all non-Null values.

**AVG()**  
**or**  
**AVG( [ALL | DISTINCT] expression )**

## MAX Function

MAX function is used to find the maximum value of a certain column. This function determines the largest value of all selected values of a column.

**MAX()**  
**or**  
**MAX( [ALL | DISTINCT] expression )**

## MIN Function

MIN function is used to find the minimum value of a certain column. This function determines the smallest value of all selected values of a column.

**MIN()**  
**or**  
**MIN( [ALL | DISTINCT] expression )**

## SubQueries

An SQL subquery is nothing but a query inside another query. We use a subquery to fetch data from two tables. A subquery is often also referred to as an inner query, while the statement containing a subquery is also called an outer query or outer select.

### General Rules of Subqueries

- A subquery must be enclosed in parentheses.
- Subqueries that return over one row can only be used with multiple value operators such as the IN operator.
- SQL Server allows you to nest subqueries up to 32 levels.

```
SELECT *  
FROM products  
WHERE product_id IN (SELECT product_id  
FROM products  
WHERE quantity_in_stock > 45) ;
```

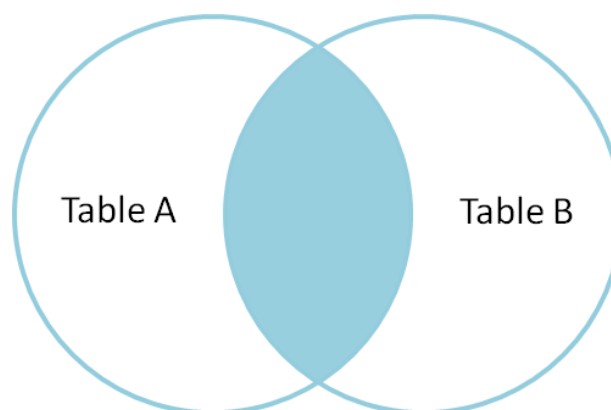
### Joins

SQL Join statement is used to combine data or rows from two or more tables based on a common field between them. Different types of Joins are as follows:

1. INNER JOIN
2. LEFT JOIN
3. RIGHT JOIN
4. FULL JOIN
5. NATURAL JOIN

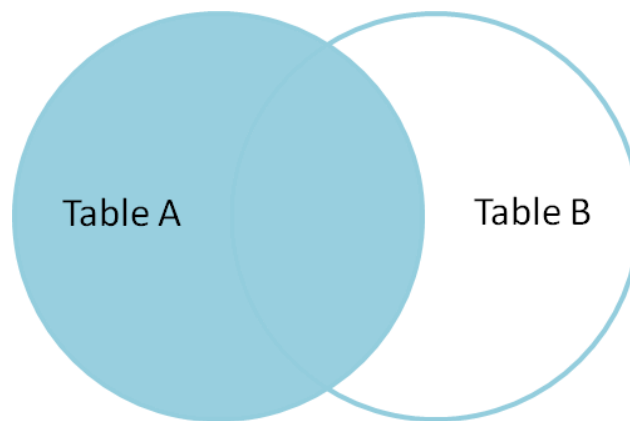
#### INNER JOIN

The INNER JOIN keyword selects all rows from both the tables as long as the condition is satisfied. This keyword will create the result-set by combining all rows from both the tables where the condition satisfies i.e value of the common field will be the same.



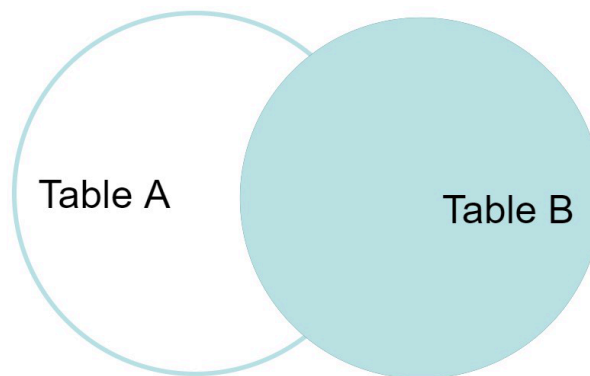
#### LEFT JOIN

This join returns all the rows of the table on the left side of the join and matches rows for the table on the right side of the join. For the rows for which there is no matching row on the right side, the result-set will contain null. LEFT JOIN is also known as LEFT OUTER JOIN.



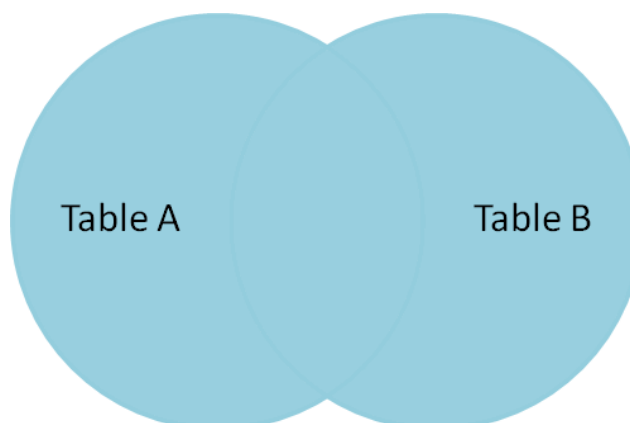
### RIGHT JOIN

RIGHT JOIN is similar to LEFT JOIN. This join returns all the rows of the table on the right side of the join and matching rows for the table on the left side of the join. For the rows for which there is no matching row on the left side, the result-set will contain null. RIGHT JOIN is also known as RIGHT OUTER JOIN.



### FULL JOIN

FULL JOIN creates the result-set by combining results of both LEFT JOIN and RIGHT JOIN. The result-set will contain all the rows from both tables. For the rows for which there is no matching, the result-set will contain NULL values.



## Natural join (?)

Natural join can join tables based on the common columns in the tables being joined. A natural join returns all rows by matching values in common columns having same name and data type of columns and that column should be present in both tables.

Both table must have at list one common column with same column name and same data type.

## UNIONS

The UNION operator is used to combine the result-set of two or more SELECT statements.

- Every SELECT statement within UNION must have the same number of columns
- The columns must also have similar data types
- The columns in every SELECT statement must also be in the same order

```
SELECT column_name(s) FROM table1  
UNION  
SELECT column_name(s) FROM table2;
```

## Index

Indexes are special lookup tables that need to be used by the database search engine to speed up data retrieval. An index is simply a reference to data in a table. A database index is similar to the index in the back of a journal. It cannot be viewed by the users and just used to speed up the database access.

Two primary methods to use the index in SQL are:

1. CREATE INDEX Command
2. DROP INDEX Command

### Advantages of Index in SQL

- Speed up select query
- Helps to make a row special or without duplicates (primary, unique)
- We can check against broad string values if the index is set to full-text index and find a word from a sentence.

### Disadvantages of Index in SQL

- Indexes take more disc space.
- INSERT, UPDATE, and DELETE are all slowed by indexes, but UPDATE is speed up if the WHERE condition has an indexed field. Since the indexes must be modified with each process, INSERT, UPDATE, and DELETE become slower.



## View

Views in SQL are kind of virtual tables. A view also has rows and columns as they are in a real table in the database. We can create a view by selecting fields from one or more tables present in the database. A View can either have all the rows of a table or specific rows based on certain condition. In this article we will learn about creating , deleting and updating Views.

```
CREATE VIEW view_name AS  
SELECT column1, column2.....  
FROM table_name  
WHERE condition;
```

## Backup & Restore, Explain Keyword