Here are the different types of the JOINs in SQL:

* **(INNER) JOIN**: Returns records that have matching values in both tables

SQL INNER JOIN also known as simple join is the most common type of join.

**SELECT** Staff\_ID, Staff\_NAME, Staff\_AGE, AMOUNT

**FROM** STAFF s, PAYMENT p  //upto this it is Cross join (whithout ‘where’)

**WHERE** s.ID =p.STAFF\_ID;  //use where clouase

* **LEFT (OUTER) JOIN**: Return all records from the left table, and the matched records from the right table, **if no record matches in right table then it return NULL value.**

**SELECT** table1.column1, table2.column2....

**FROM** table1

**LEFT JOIN**table2

**ON** table1.column\_field = table2.column\_field;

* **RIGHT (OUTER) JOIN**: Return all records from the right table, and the matched records from the left table.

**SELECT** table1.column1, table2.column2.....

**FROM** table1

RIGHT JOIN table2

**ON** table1.column\_field = table2.column\_field;

* **FULL (OUTER) JOIN**: Return all records when there is a match in either left or right table. full outer join=full join
* You don't have FULL JOINS on MySQL so use union between left and right join as belowss

SELECT \* FROM t1

LEFT JOIN t2 ON t1.id = t2.id

UNION

SELECT \* FROM t1

RIGHT JOIN t2 ON t1.id = t2.id

# **SQL Cross Join**

When each row of first table is combined with each row from the second table, known as Cartesian join or cross join.

**We can specify a CROSS JOIN in two ways:**

**1. SELECT** \* **FROM** [TABLE1] CROSS JOIN [TABLE2]//by specifying CROSS JOIN

**2.** **SELECT** \* **FROM** [ TABLE1] , [TABLE2] //OR without where clause

## What is SQL?

SQL is a language which is used to operate your database. SQL is the basic language used for all the databases.

 Its design for managing the data in an RDBMS, such as MYSQL.

## What is MYSQL?

Developed in the mid-90s., MySQL was one of the first open-source database available in the market. Today there are many alternatives variants of MySQL,

MySQL uses the SQL language to query the database.

* Conclusion:- SQL is a query language while MYSQL is database software, It used "SQL" language to query the database.

1.If there’s any alternative to replace MySQL, it’s **MariaDB,** which was created by the original developers of MySQL. Specifically, MariaDB is a database server with drop-in replacement for MySQL. It’s used by popular tech destinations like Google, Facebook, and even Wikipedia.

2. **Postgre SQL** is another open source MySQL alternative. Postgre SQL users can also expect additional features like **table inheritance**, a rules system (also called the query rewrite system), and database events. PostgreSQL runs stored procedures in a multitude of programming languages, including Java, Perl, Python, Ruby, Tcl, C/C++, and its own PL/pgSQL (similar to Oracle’s PL/SQL).

3. **SQLite** is suggested as a great database for the internet of things (IoT) and is a popular choice for cellphones, PDAs, and even MP3 players. This solution is also a great option for those looking for software that makes efficient use of space as SQLite has a small code footprint and doesn’t require maintenance from a database administrator.

**SQL** vs **NoSQL**: High-Level Differences. SQL databases are primarily called as Relational Databases (RDBMS); whereas NoSQL database are primarily called as non-relational or distributed database. ... SQL databases have predefined schema

That means data stored in table (rows and columns),whereas NoSQL databases have dynamic schema for unstructured data that means data stored in individual document.

 Example-SQL databases **include** MySQL, Oracle, PostgreSQL, and Microsoft SQL Server. NoSQL database examples **include** MongoDB, BigTable, Redis, RavenDB Cassandra, HBase, Neo4j and CouchDB.

Difference between RDMS and DBMS

* Relational Database Management System (RDBMS) is an advanced version of a DBMS system.
* RDBMS uses a **tabular structure** where the headers are the column names, and the rows contain corresponding values DBMS system, stores data in either a navigational or **hierarchical form.**
* RDBMS supports the **Normalization** and **ACID** Property where DBMS not.

Normalization divides the larger table into the smaller table and links them using relationship, hence RDMS support relationship(client-server).

* Examples of DBMS are a file system, XML, Windows Registry, etc.
* Example of RDBMS is MySQL, Oracle, SQL Server, etc.

### **What are the advantages of MySQL in comparison to Oracle?**

1. MySQL is a free, fast, reliable, open source relational database while Oracle is expensive, although they have provided Oracle free edition to attract MySQL users.

Oracle provide tools for enterprises while MySQL not.

**MySQL default port number is 3306.**

**ACID**

**Atomicity**

All changes to data are performed as if they are a single operation. That is, all the changes are performed, or none of them are.

For example, in an application that transfers funds from one account to another, the atomicity property ensures that, if a debit is made successfully from one account, the corresponding credit is made to the other account.

**Consistency**

Data is in a consistent state when a transaction starts and when it ends.

For example, in an application that transfers funds from one account to another, the consistency property ensures that the **total value of funds in both the accounts is the same at the start and end of each transaction.**

**Isolation**

The intermediate state of a transaction is invisible to other transactions. As a result, transactions that run concurrently appear to be serialized.

For example, in an application that transfers funds from one account to another, the isolation property ensures that another transaction sees the transferred funds in one account or the other, but not in both, nor in neither.

**Durability**

Once a transaction completes successfully, the changes it has made into the database should be permanent even if there is a system failure.

For example, in an application that transfers funds from one account to another, the durability property ensures that the changes made to each account **will not be reversed**.

# **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.
* Normalization divides the larger table into the smaller table and links them using relationship.
* The **normal form** is used to reduce redundancy from the database table.

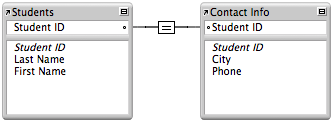
# **First Normal Form (1NF)**

It states that an attribute of a table cannot hold multiple values. It must hold only single-valued attribute. Example: EMP\_PHONE.

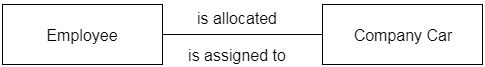
For more: <https://www.javatpoint.com/dbms-second-normal-form>

**Types Of Relationship(also many-one)**

In a **one-to-one relationship**, one [record](https://fmhelp.filemaker.com/help/17/fmp/en/FMP_Help/glossary.html#ww1028030) in a [table](https://fmhelp.filemaker.com/help/17/fmp/en/FMP_Help/glossary.html#ww1045562) is associated with one and only one record in another table. For example, in a school database, each student has only one student ID, and each student ID is assigned to only one person.

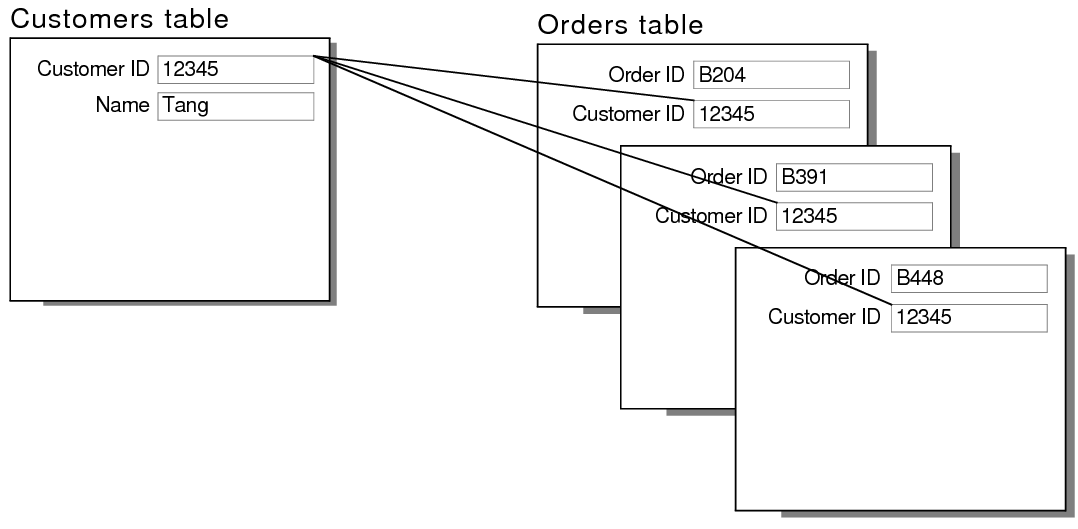


* **For example:** if an employee is allocated a company car then that car can only be driven by that employee.
* Therefore, employee and company car have a one-to-one relationship.

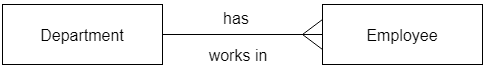


In a **one-to-many relationship**, one [record](https://fmhelp.filemaker.com/help/17/fmp/en/FMP_Help/glossary.html#ww1028030) in a [table](https://fmhelp.filemaker.com/help/17/fmp/en/FMP_Help/glossary.html#ww1045562) can be associated with one or more records in another table. For example, each customer can have many sales orders.

A one-to-many relationship looks like this in the [relationships graph](https://fmhelp.filemaker.com/help/17/fmp/en/FMP_Help/glossary.html#ww1028051):

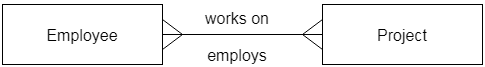


* **For example:** An employee works in one department, but a department has many employees.
* Therefore, department and employee have a one-to-many relationship.



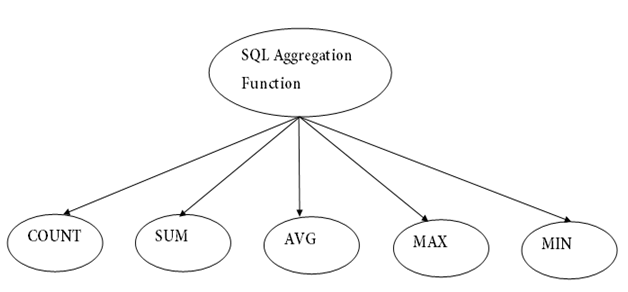
A **many-to-many relationship**occurs when multiple [records](https://fmhelp.filemaker.com/help/17/fmp/en/FMP_Help/glossary.html#ww1028030) in a [table](https://fmhelp.filemaker.com/help/17/fmp/en/FMP_Help/glossary.html#ww1045562) are associated with multiple records in another table. For example, a many-to-many relationship exists between customers and products: customers can purchase various products, and products can be purchased by many customers.

* **For example:** At the same time, an employee can work on several projects, and a project has a team of many employees.
* Therefore, employee and project have a many-to-many relationship.



# **SQL Aggregate Functions**

SQL aggregation function is used to perform the calculations on **multiple** **rows** **of** a **single** **column**.



**COUNT** function is used to Count the number of rows in a database table.(return int value).

**COUNT(\*)** counts null rows also.

**Sum** function is used to calculate the sum of selected columns.

**AVG** function returns the average of all non-Null values (not consider NULL values).

**MAX** function is used to find the maximum value of a certain column.

**MIN** function is used to find the minimum value of a certain column.

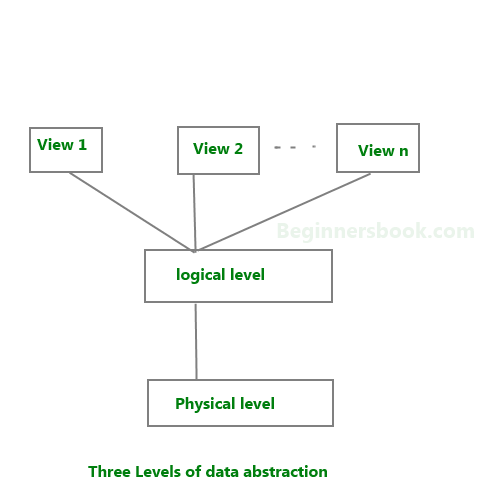
### **What is the difference between NOW() and CURRENT\_DATE()?**

NOW() command is used to show current year, month, date with hours, minutes and seconds while CURRENT\_DATE() shows the current year with month and date only.

SELECT NOW();

SELECT CURRENT\_DATE();

# Data Abstraction in DBMS



**We have three levels of abstraction**:

**Physical level**: This is the lowest level of data abstraction. It describes **how** data is actually stored in database. You can get the complex data structure details at this level.

**Logical level**: This is the middle level of 3-level data abstraction architecture. It describes **what** data is stored in database.

**View level**: Highest level of data abstraction. This level describes the user interaction with database system. At **view level**, user just interact with system with the help of GUI and enter the details at the screen, they are not aware of how the data is stored and what data is stored; such details are hidden from them.

## DBMS Schema

**Definition of schema**: **Design of a database is called the schema**. Schema is of three types: Physical schema, logical schema and view schema.

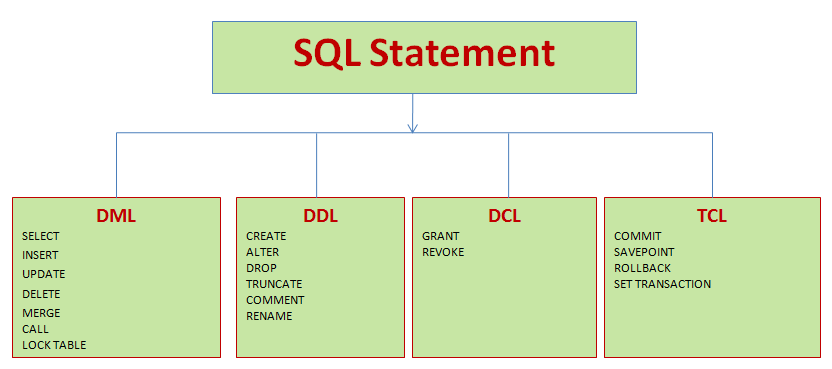
DBMS Language :

1.DDL: Related to the Structure of the Database.

2.DML: Related to the Data in the Table/Database.

3.DCL:- used to **control** access to data stored in a database (Authorization).

4.TCL:- Transaction Control Language(**TCL**) commands are used to **manage** transactions in the database.



Delete v/s Drop.(Table):-

1. Delete statement performs **conditional** based deletion, whereas Drop command deletes **entire records** in the table.
2. Delete statement removes only the rows in the table and it preserves the table structure as same, and Drop command removes all the data in the table and the table structure.
3. DROP and TRUNCATE are DDL commands, whereas DELETE is a DML command.

### **Using the --safe-updates option**

For beginners, a useful startup option is --safe-updates (or --i-am-a-dummy, which has the same effect). It is helpful for cases when you might have issued a DELETE FROM tbl\_name statement but forgotten the WHERE clause. Normally, such a statement deletes all rows from the table. With --safe-updates, you can delete rows only by specifying the key values that identify them. This helps prevent accidents.

When you use the --safe-updates option, mysql issues the following statement when it connects to the MariaDB server:

SET sql\_safe\_updates=1, sql\_select\_limit=1000, sql\_max\_join\_size=1000000;

The [SET](https://mariadb.com/kb/en/set/) statement has the following effects:

* You are not allowed to execute an [UPDATE](https://mariadb.com/kb/en/update/) or [DELETE](https://mariadb.com/kb/en/delete/) statement unless you specify a key constraint in the WHERE clause or provide a LIMIT clause (or both). For example:

**UPDATE** tbl\_name SET not\_key\_column=val **WHERE** key\_column=val;

**UPDATE** tbl\_name SET not\_key\_column=val **LIMIT** 1;

* The server limits all largeSELECT results to 1,000 rows unless the statement includes a LIMIT clause.
* The server aborts multiple-table SELECT statements that probably need to examine more than 1,000,000 row combinations.

To specify limits different from 1,000 and 1,000,000, you can override the defaults by using the --select\_limit and --max\_join\_size options:

mysql --safe-updates --select\_limit=500 --max\_join\_size=10000