
MySQL
mongoDB
amazon
REDSHIFT
SQL
IBM
Google BigQuery
druid
ORACLE
cassandra
snowflake

What is a Database?

A database is an organized collection of structured information.

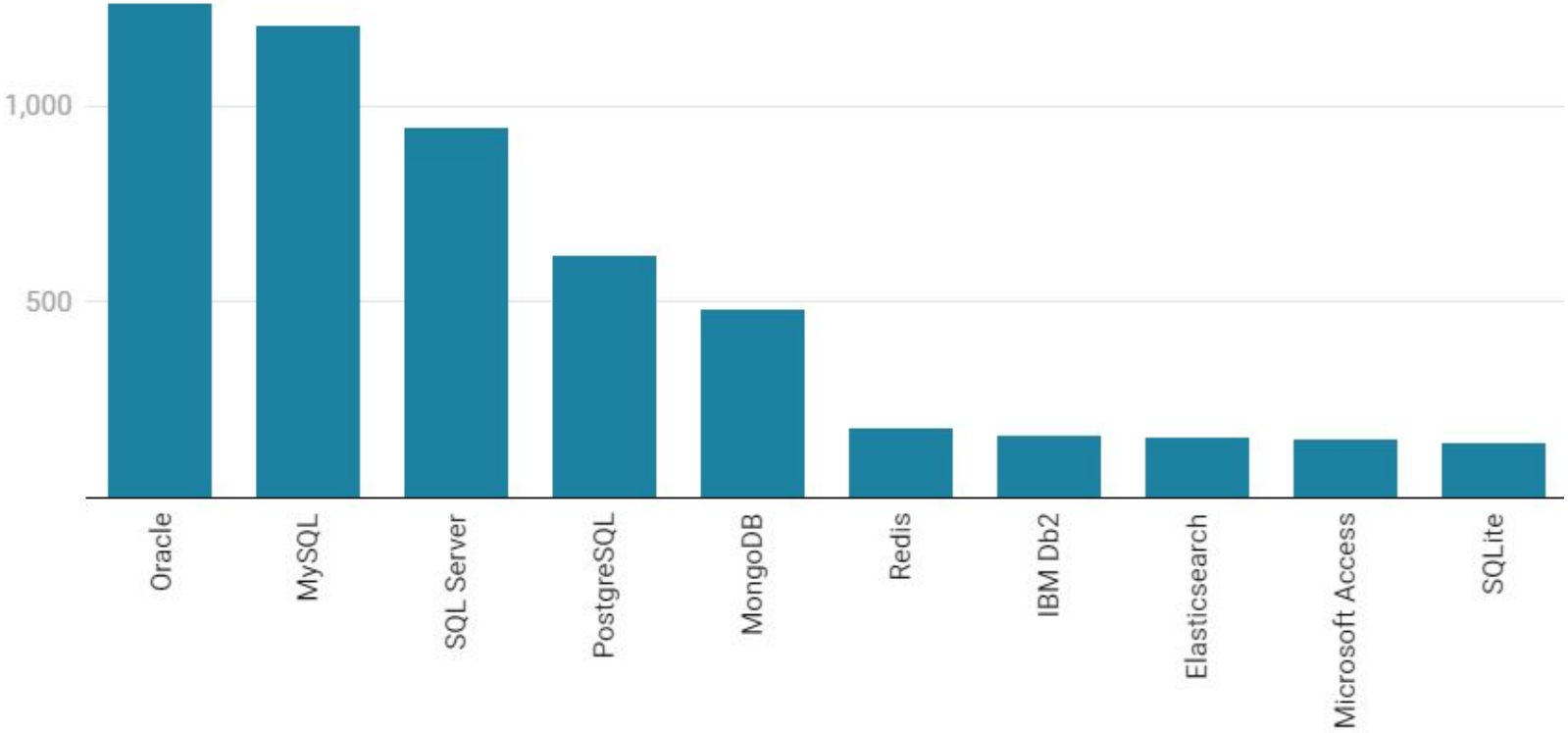
or

Data, typically stored electronically in a computer system.

POPULAR DATABASES

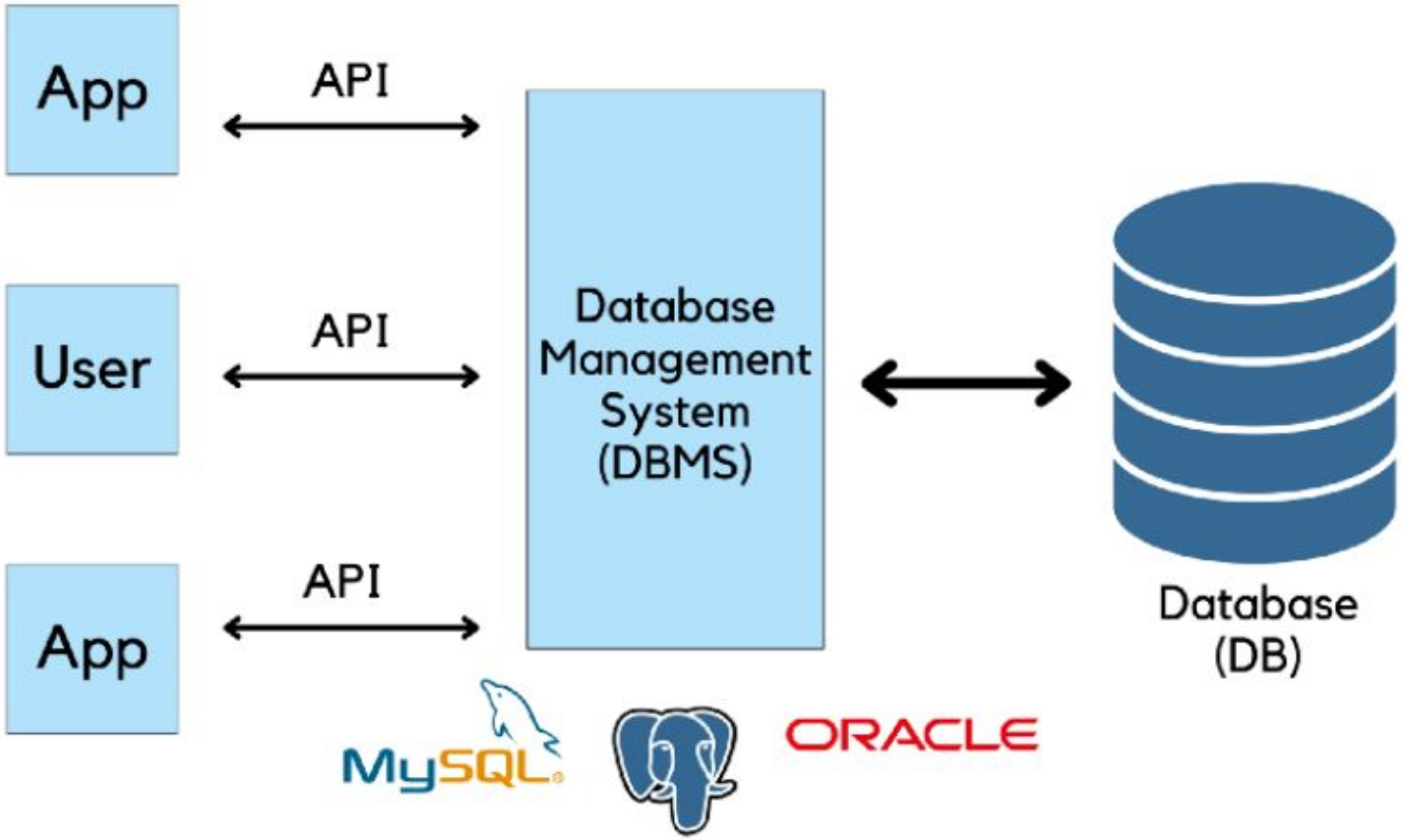
Oracle	MySQL	MS SQL
PostgreSQL	MangoDB	DB2
SQLite	Sybase	MariaDB

Top 10 Databases in 2022

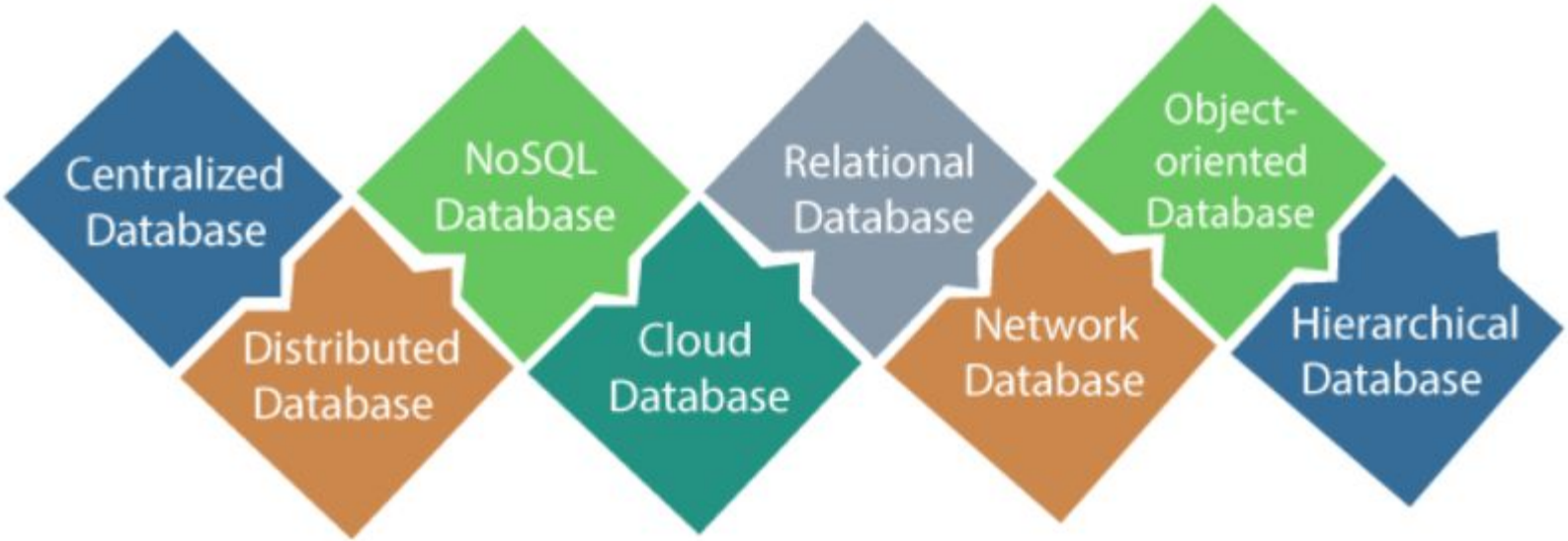


What is Database Management System?

- **Database management system is a software which is used to manage the database.** For example: MySQL, Oracle, etc are a very popular commercial database which is used in different applications.
- DBMS provides an interface to perform various operations like database creation, storing data in it, updating data, creating a table in the database and a lot more


App
API
User
API
App
API
Database
Management
System
(DBMS)
Database
(DB)
MySQL
ORACLE

Types of Database



What is SQL?

SQL

- SQL stands for **Structured Query Language**. It is used for storing and managing data in **relational database management system (RDMS)**.
- It is a **standard language for Relational Database System.** It enables a user to **create, read, update and delete** relational databases and tables.
- All the RDBMS like **MySQL, Informix, Oracle, MS Access and SQL Server use SQL** as their standard database language.
- SQL allows users to query the database in a number of ways, using **English-like** statements.

History of SQL

SQL was first brought into origin by IBM Researcher's – **Raymond F. Boyce**, and **Donald D. Chamberlin** in the **1970**'s and the initial version created by them was called **SEQUEL** or **Structured English Query Language** which worked on **manipulation and retrieving data from IBM databases.**

After commercial testing, IBM released various versions like System/38, SQL/DS, and DB2 in 1979, 1981, and 1983, respectively.

In 1986 making a breakthrough, ANSI and ISO adopted the Standard "Database Language SQL".

MySQL

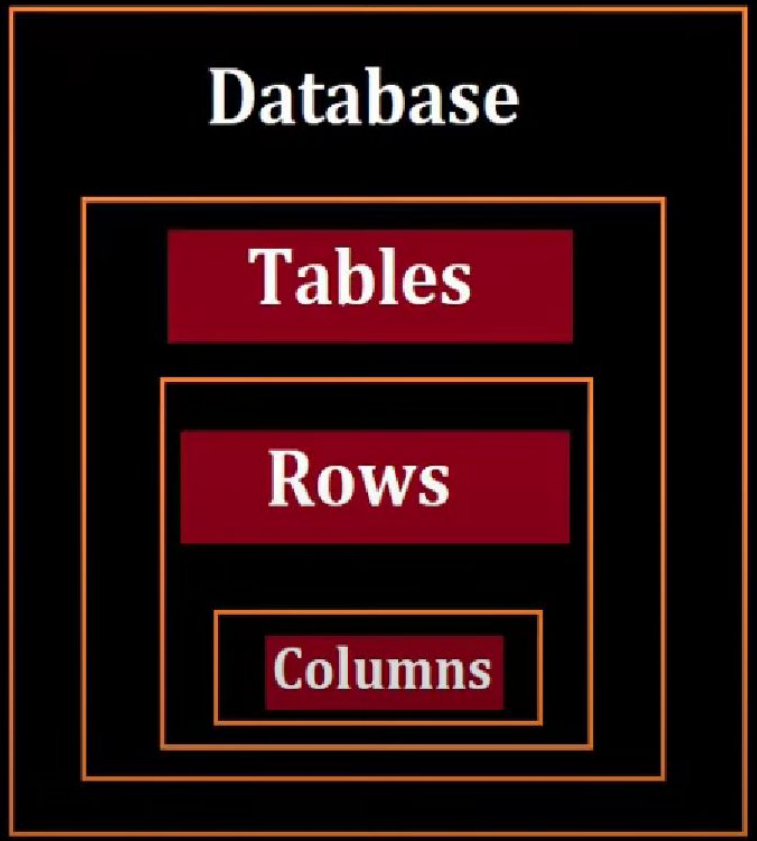
- **MySQL** is an **open-source relational database management system (RDBMS).** Its name is a combination of "My", the name of co-founder Michael Widenius daughter My and "SQL", the abbreviation for Structured Query Language.
- MySQL is currently the **most popular database management system software** used for managing the relational database.
- MySQL is an open-source relational database management system that uses SQL commands to perform specific functions/operations in a database.
- Swedish company MySQL AB →Sun microsystem→Oracle

What SQL can do?

We can use SQL for various operations. Some of them are stated below

- We can use SQL to run queries on a database.
- SQL is used to perform CRUD(Create, Retrieve, Update, Delete) operations on a database.
- SQL is required to create and manage databases.
- We use SQL to update and manipulate the existing data in the database.
- SQL can be used to create views over an already existing database table.
- SQL comes in handy when we need to divide permissions among different users of a database.
- We are able to perform transactions easily on databases using SQL as it is compatible with almost all programming languages like – C++, Java, Python, etc.

MySQL



Example

EmpID	FirstName	LastName	EmpAGE	EmpZONE
1	Jack	Sparrow	25	North
2	Tom	Hanks	25	South
3	Brad	Pitt	30	West
4	Emma	Stone	27	East

Data Types in MYSQL

- Numeric
- String
- Data and time
- Spatial
- Json

Numeric Data Types

Integer types(Exact value):

- INTEGER
- INT
- SMALLINT
- TINYINT
- MEDIUMINT
- BIGINT

Fixed-point type:

- DECIMAL(TOTAL_LENGTH,PRECISION)

Floating point types(Approximate values):

- FLOAT(p)
- DOUBLE

Type	Storage (Bytes)	Minimum Value Signed	Minimum Value Unsigned	Maximum Value Signed	Maximum Value Unsigned
TINYINT	1	-128	0	127	255
SMALLINT	2	-32768	0	32767	65535
MEDIUMINT	3	-8388608	0	8388607	16777215
INT	4	-2147483648	0	2147483647	4294967295
BIGINT	8	-2^{63}	0	$2^{63}-1$	$2^{64}-1$

String Data Types

- **CHAR(size)** →size is 1,if size not mentioned,max size 30 characters
- **VARCHAR(size)** →size is mandatory for varchar max size 65,535 characters
- **BINARY**
- **VARBINARY**
- **BLOB(Binary large object)** →TINYBLOB,BLOB,MEDIUMBLOB,LONGBLOB
- **TEXT** →TINYTEXT,TEXT,MEDIUMTEXT,LONGTEXT
- **ENUM('chc-1','chc-2',...) →list of permitted values for a column**
- **SET**

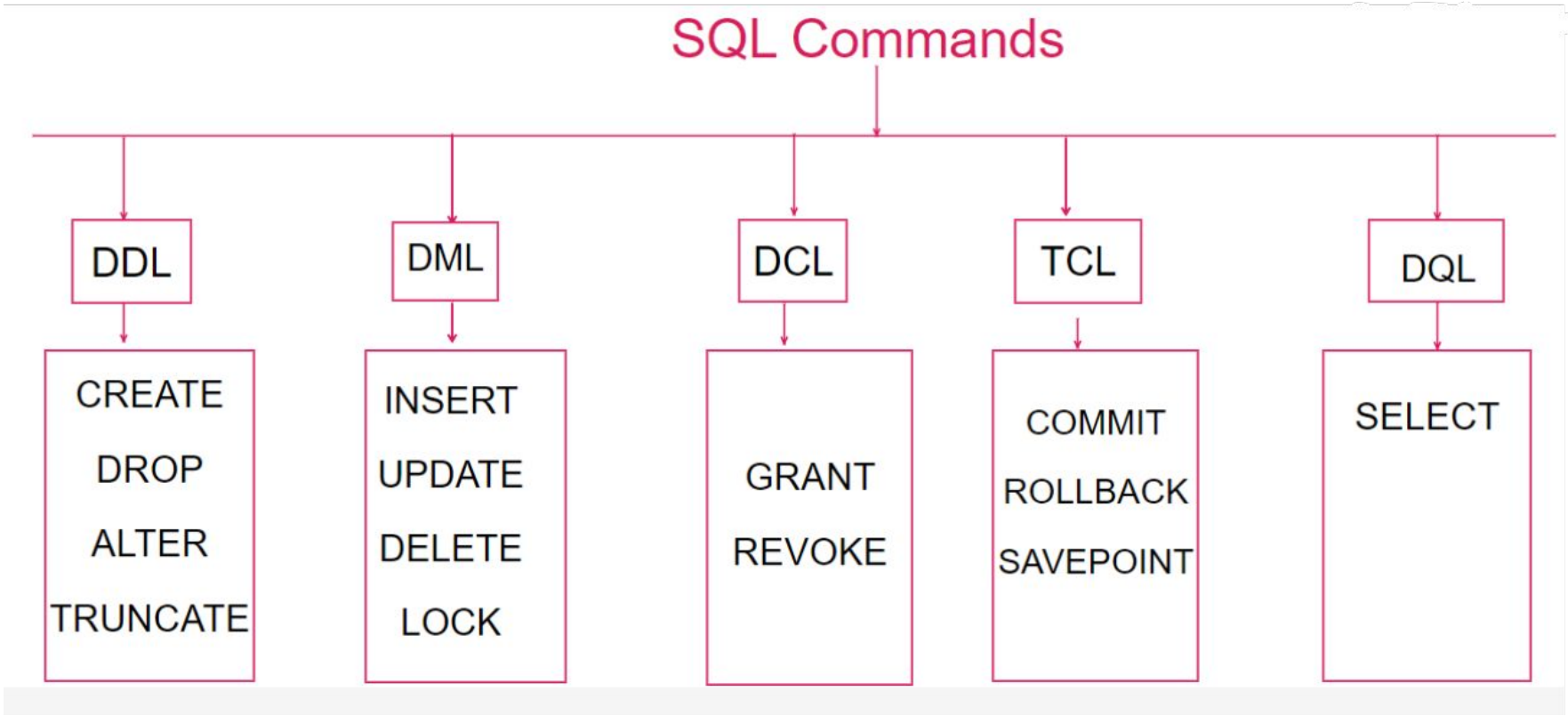
Date and Time Data types

- DATE
- TIME[(fsp)]
- DATETIME[(fsp)]
- TIMESTAMP[(fsp)]
- YEAR

SQL COMMANDS

SQL commands are mainly categorized into four types-

1. DDL – Data Definition Language
2. DQl – Data Query Language
3. DML – Data Manipulation Language
4. DCL – Data Control Language


SQL Commands
DDL
DML
DCL
TCL
DQL
CREATE
DROP
ALTER
TRUNCATE
INSERT
UPDATE
DELETE
LOCK
GRANT
REVOKE
COMMIT
ROLLBACK
SAVEPOINT
SELECT

CREATING A DATABASE

```
mysql> CREATE DATABASE CODEGNAN;
Query OK, 1 row affected (0.01 sec)
```

USING A DATABASE

```
mysql> USE CODEGNAN;
Database changed
```

DDL Commands

CREATE :

Syntax : *CREATE TABLE TABLE_NAME(COLUMN_1 DTYPE,*
COLUMN_2 DTYPE,
COLUMN_3 DTYPE,

COLUMN_n DTYPE);

QUERY:

```
mysql> CREATE TABLE EMPLOYEES(EMP_ID CHAR(5),
    -> FNAME VARCHAR(50),
    -> LNAME VARCHAR(50),
    -> AGE INT,DOJ DATE,
    -> ADDRESS TINYTEXT,
    -> DEPT VARCHAR(20));
Query OK, 0 rows affected (0.02 sec)

mysql> DESC EMPLOYEES;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| EMP_ID  | char(5)     | YES  |     | NULL    |       |
| FNAME   | varchar(50) | YES  |     | NULL    |       |
| LNAME   | varchar(50) | YES  |     | NULL    |       |
| AGE     | int         | YES  |     | NULL    |       |
| DOJ     | date        | YES  |     | NULL    |       |
| ADDRESS | tinytext    | YES  |     | NULL    |       |
| DEPT    | varchar(20) | YES  |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
7 rows in set (0.00 sec)
```

ALTER:

Syntax:

1. ALTER -MODIFY:

ALTER TABLE TABEL_NAME

MODIFY COLUMN_NAME1 DTYPE ,

MODIFY COLUMN_NAME2 DTYPE,

----------;

1. ALTER - ADD:

ALTER TABLE TABLE_NAME

ADD COLUMN_NAME1 DTYPE [FIRST|AFTER COLUMN_NAME],

ADD COLUMN_NAME2DTYPE [FIRST|AFTER COLUMN_NAME],

----------;

QUERY:

```
mysql> ALTER TABLE EMPLOYEES
    -> MODIFY FNAME VARCHAR(30);
Query OK, 4 rows affected (0.06 sec)
```

```
mysql> ALTER TABLE EMPLOYEES
    -> ADD LOCATION TINYTEXT;
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> DESC EMPLOYEES;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| EMPLOYEE_ID | char(5)     | YES  |     | NULL    |       |
| FNAME       | varchar(30) | YES  |     | NULL    |       |
| LNAME       | varchar(50) | YES  |     | NULL    |       |
| AGE         | int         | YES  |     | NULL    |       |
| DOJ         | date        | YES  |     | NULL    |       |
| ADDRESS     | tinytext    | YES  |     | NULL    |       |
| DPT         | varchar(20) | YES  |     | NULL    |       |
| LOCATION    | tinytext    | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
```

QUERY

```
mysql> ALTER TABLE EMPLOYEES
    -> ADD PFID INT AFTER ADDRESS;
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> DESC EMPLOYEES;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| EMPLOYEE_ID | char(5)     | YES  |     | NULL    |       |
| FNAME       | varchar(30) | YES  |     | NULL    |       |
| LNAME       | varchar(50) | YES  |     | NULL    |       |
| AGE         | int         | YES  |     | NULL    |       |
| DOJ         | date        | YES  |     | NULL    |       |
| ADDRESS     | tinytext    | YES  |     | NULL    |       |
| DPT         | varchar(20) | YES  |     | NULL    |       |
| LOCATION    | tinytext    | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
8 rows in set (0.00 sec)
```

ALTER -RENAME COLUMN

METHOD 1-

RENAME COLUMN:

ALTER TABLE TABLE_NAME

RENAME COLUMN OLD_COLUMN1 TO NEW_COLUMNNAME1,

RENAME COLUMN OLD_COLUMN2 TO NEW_COLUMNNAME2,

----------;

METHOD 2-

CHANGE COLUMN:

ALTER TABLE TABLE_NAME

CHANGE COLUMN OLD_COLUMN1 NEW_COLUMN_NAME1 DTYPE,

CHANGE COLUMN OLD_COLUMN2 NEW_COLUMN_NAME2 DTYPE,

----------;

QUERY

```
mysql> ALTER TABLE EMPLOYEES
    -> RENAME COLUMN FNAME TO FIRSTNAME;
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> ALTER TABLE EMPLOYEES
    -> CHANGE COLUMN LNAME LASTNAME VARCHAR(60);
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> DESC EMPLOYEES;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| EMPLOYEE_ID | char(5)     | YES  |     | NULL    |       |
| FIRSTNAME   | varchar(30) | YES  |     | NULL    |       |
| LASTNAME    | varchar(60) | YES  |     | NULL    |       |
| AGE         | int         | YES  |     | NULL    |       |
| DOJ         | date        | YES  |     | NULL    |       |
| ADDRESS     | tinytext    | YES  |     | NULL    |       |
| PFID        | int         | YES  |     | NULL    |       |
| DPT         | varchar(20) | YES  |     | NULL    |       |
| LOCATION    | tinytext    | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
9 rows in set (0.01 sec)
```

ALTER -DROP COLUMN

SYNTAX

ALTER TABLE TABLE_NAME

DROP COLUMN COLUMN_NAME1,

DROP COLUMN COLUMN_NAME2,

----------,

DROP COLUMN COLUMN_NAMEN;

QUERY

```
mysql> ALTER TABLE EMPLOYEES
    -> DROP COLUMN LOCATION;
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE EMPLOYEES
    -> DROP COLUMN PFID;
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DESC EMPLOYEES;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| EMPLOYEE_ID | char(5)     | YES  |     | NULL    |       |
| FIRSTNAME   | varchar(30) | YES  |     | NULL    |       |
| LASTNAME    | varchar(60) | YES  |     | NULL    |       |
| AGE         | int         | YES  |     | NULL    |       |
| DOJ         | date        | YES  |     | NULL    |       |
| ADDRESS     | tinytext    | YES  |     | NULL    |       |
| DPT         | varchar(20) | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
```

ALTER TABLE RENAME:

SYNTAX

ALTER TABLE TABLE_NAME RENAME NEW_TABLENAME

```
mysql> ALTER TABLE EMPLOYEES RENAME CODEGNAN_EMP;
Query OK, 0 rows affected (0.02 sec)

mysql> DESC CODEGNAN_EMP;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| EMPLOYEE_ID | char(5)     | YES  |     | NULL    |       |
| FIRSTNAME   | varchar(30) | YES  |     | NULL    |       |
| LASTNAME    | varchar(60) | YES  |     | NULL    |       |
| AGE         | int         | YES  |     | NULL    |       |
| DOJ         | date        | YES  |     | NULL    |       |
| ADDRESS     | tinytext    | YES  |     | NULL    |       |
| DPT         | varchar(20) | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
```

DROP

SYNTAX

DROP TABLE TABLE_NAME

QUERY

```
mysql> DROP TABLE ESWAR;
```

TRUNCATE

SYNTAX

TRUNCATE TABLE TABLE_NAME;

QUERY

```
mysql> TRUNCATE TABLE CODEGNAN_EMP;
Query OK, 0 rows affected (0.03 sec)
```

DML

```
INSERT--
INSERT INTO TABLE_NAME(COLUMN1,COLUMN2,COLUMN3,....)
VALUES(COLUMN1_VALUE1,COLUMN_VALUE2,......)

                                OR

INSERT INTO TABLE_NAME
VALUES(COLUMN1_VALUE1,COLUMN_VALUE2,......)
```

Multiple rows at a time-

```
INSERT INTO TABLE_NAME VALUES
(VALUE1,VALUE2...),(VALUE1,VALUE2...),(VALUE1,VALUE2...)
```

OR

```
INSERT INTO TABLE_NAME(column1,column2,...) VALUES
(VALUE1,VALUE2...),(VALUE1,VALUE2...),(VALUE1,VALUE2...)
```

QUERY

```
mysql> INSERT INTO CODEGNAN_EMP(EMPLOYEE_ID,FIRSTNAME,LASTNAME,AGE,DOJ,ADDRESS,DPT)
    -> VALUES('19720','Eswar','Nandivada',21,'2022-01-10','VIJAYAWADA','IT');
Query OK, 1 row affected (0.01 sec)
```

```
mysql> INSERT INTO CODEGNAN_EMP
    -> VALUES('19716','VARA','PRASAD',22,'2022-08-01','KERALA','HR'),
    -> ('19711','RAVI','KUMAR',23,'2021-09-11','KHAMMAM','IT'),
    -> ('19715','YESHWANT','RAVILA',25,'2019-04-23','HYDERABAD','IT');
Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

MYSQL WARNINGS

In MYSQL,warnings are diagnostic statements that displays information about the conditions (errors, warnings, and notes) resulting from executing a statement in the current session.

Warnings are generated for DML statements such as `INSERT`, `UPDATE` as well as DDL statements such as `CREATE TABLE` and `ALTER TABLE`.

We can display warnings of a query by using *SHOW WARNINGS* STATEMENT
We can enable warnings by using *warings* or *\W*
We can disable warnings by using *nowarning* or *\w*

QUERY

```
mysql> nowarning
Show warnings disabled.
mysql> select 1/0;
+------+
| 1/0  |
+------+
| NULL |
+------+
1 row in set, 1 warning (0.00 sec)
```

```
mysql> warnings
Show warnings enabled.
mysql> select 1/0;
+------+
| 1/0  |
+------+
| NULL |
+------+
1 row in set, 1 warning (0.00 sec)

Warning (Code 1365): Division by 0
```

Types of Relationships in DBMS

One to One relationship

One to many or many to one relationship

Many to many relationships

Foreign Keys

A FOREIGN KEY is a field (or collection of fields) in one table, that refers to the PRIMARY KEY in another table.

The table with the foreign key is called the child table, and the table with the primary key is called the referenced or parent table

MYSQL JOINS

What are JOINS?

Joins help retrieving data from two or more database tables.
The tables are mutually related using primary and foreign keys.

Types of JOINS in MYSQL

CROSS JOIN or cartesian join

INNER JOIN or Simple join

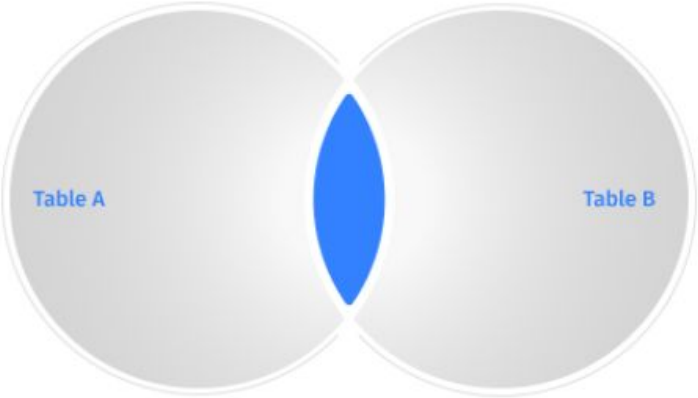
OUTER JOINS-

A. LEFT OUTER JOIN or LEFT JOIN
B. RIGHT OUTER JOIN or RIGHT JOIN

INNER JOIN

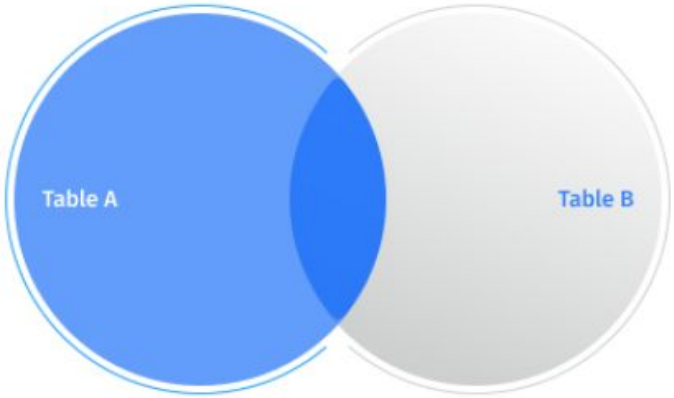
INNER JOINs are used to fetch only common matching records. The INNER JOIN clause allows retrieving only those records from Table A and Table B, that meet the join condition. It is the most widely used type of JOIN.

INNER JOIN



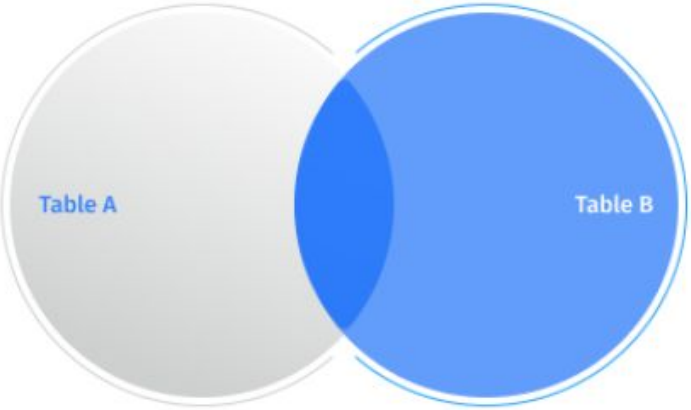
LEFT JOIN

LEFT JOINs allow retrieving all records from Table A, along with those records from Table B for which the join condition is met. For the records from Table A that do not match the condition, the NULL values are displayed.



RIGHT JOIN

RIGHT JOINs allow retrieving all records from Table B, along with those records from Table A for which the join condition is met. For the records from Table B that do not match the condition, the NULL values are displayed



CROSS JOIN

MySQL CROSS JOIN, also known as a cartesian join, retrieves all combinations of rows from each table. In this type of JOIN, the result set is returned by multiplying each row of table A with all rows in table B if no additional condition is introduced.