



Day 1 Introduction, SQL basics: DDL, DML, DRL

Java

It is mainly used to develop a business application.



Business Application

Business organizations: the main objective for any business organization is profit.

1. Small-scale business organizations (grocery shops, petrol pumps, they are running their services in a limited area.)



2. Larger scale business organizations(Banks, railways, eCommerce, travel, insurance): these types of Business organizations are also called enterprises.



These Business Organizations provide their services to the **client/customers**, and to **computerize those services** whatever application we develop is known as a **Business application**.

Common and general things in any business application:

1. Storing and maintaining business data in a secure and easily retrievable manner.
2. Processing the business data according to the business rules.

3. Presenting the data to the user, in a user-understandable format.

Note: in real-time business applications we keep/maintain business data inside **RDMBS** s/w.

Data and information:

It is a collection of raw and isolated facts.

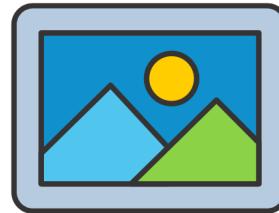
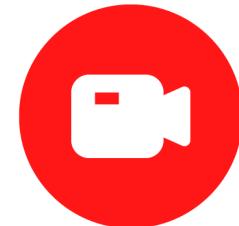
Information: when we process the data, then we will get a meaningful result, this result is called **information**.

Data	Information
<ul style="list-style-type: none">• 5• Cities• Top• John• Mangoes• Born• (1980)	<ul style="list-style-type: none">• 5 Mangoes• Top Cities• John Born (1980)

Technically **Data Format:**

Hello
0 1 2 3 4
-5 -4 -3 -2 -1

1 2 3



To store such kind of data we have a **Data Store**

Datastore

It is a store where we can store or keep our business-related data.

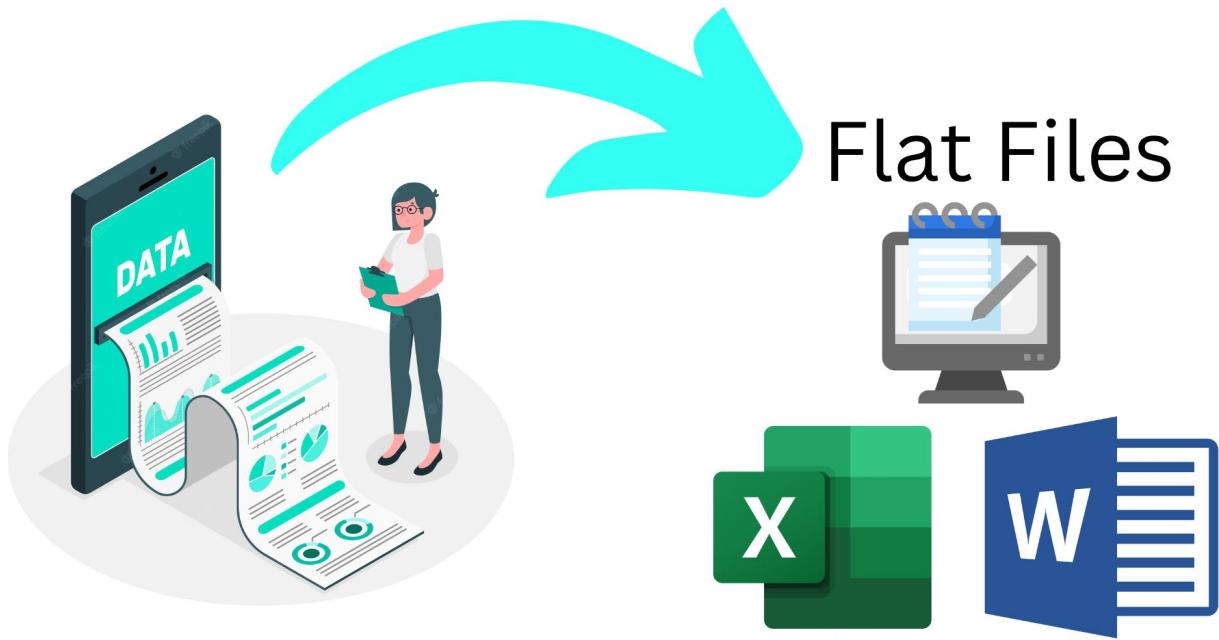
Real Life Example:



In the above example, the cupboard is a datastore whereas various spices and pulses are data.

Similarly, To store **Technical Data**:

1. Normal books and papers.
2. Flat files in the computer system (notepad, excel sheet, word files)



Disadvantages of the flat files

1. Data maintenance.
 - Data redundancy
 - Data integrity /Data inconsistency
2. Security
3. Data retrieval

To overcome these problems we need to store the data inside the **DBMS s/w** (RDBMS s/w)

Database: It is an organized collection of interrelated data or a structured collection of data.

DBMS

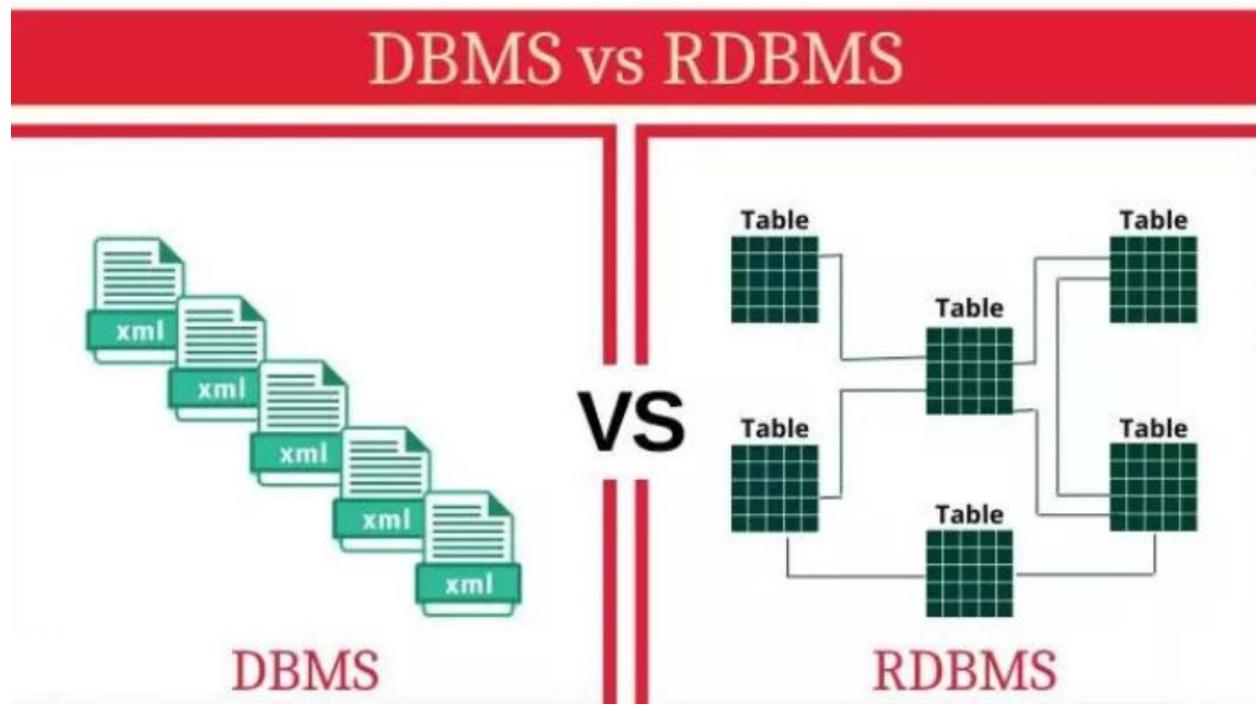
It is a type of s/w where we can manage multiple databases.

RDBMS

Relational Database Management System (RDBMS) is a more advanced version of a DBMS system that allows access to data in a more efficient way. It is used to store or

manage only the data that are in the form of tables.

It is an extension of DBMS.



In DBMS, the data is stored as a file, whereas in RDBMS, data is stored in the form of tables.

RDBMS	DBMS
Data stored is in table format	Data stored is in the file format
Multiple data elements are accessible together	Individual access of data elements
Data in the form of a table are linked together	No connection between data
Normalisation is not achievable	There is normalisation
Support distributed database	No support for distributed database
Data is stored in a large amount	Data stored is a small quantity
Here, redundancy of data is reduced with the help of key and indexes in RDBMS	Data redundancy is common
RDBMS supports multiple users	DBMS supports a single user

We have multiple RDBMS s/w



Note: Every RDBMS s/w is essentially a DBMS s/w but the reverse is not true.

Example: MongoDB is a DBMS s/w but it is not a RDBMS.

To Download MySQL s/w: <https://dev.mysql.com/downloads/installer/>

Video:<https://drive.google.com/file/d/1vokU2khVj4OSU97Hn2vHEoqEBPLi-wvS/view?usp=sharing>

SQL

In order to work with RDBMS s/w we need to use SQL (**Structured Query language**)
It is an interface using which we can work with any kind of RDBMS s/w.

Note: Whenever we install any RDBMS s/w then at a physical level (in hard disk) some databases will be created automatically.

- Each RDBMS s/w has one main component is there which is called as "**Database engine**", which will execute the SQL commands(statements).
1. To execute any SQL commands with DB-engine we need a client application.
 2. Along with the MySQL installation, we get the "MySQL command line client", from where we can execute any type of SQL statement with the database engine.
 3. We can install some GUI clients also for Mysql for example: "Mysql workbench".

SQL Query:

SQL is a case-insensitive language. (but data kept inside the tables are not case insensitive)

The SQL language is a collection of some predefined commands:

these commands are categorized into the following categories:

1. **DDL** (Data definition language)
(create, alter, drop, truncate, rename)
2. **DML** (Data manipulation language)
(insert, update, delete)
3. **DRL** (Data retrieval language)
(select)
4. **TCL** (Transaction control language)
(commit, rollback, savepoint)

5. DCL (Decision control language) (grant, revoke)

Note: In real-time application development, Java developers are only allowed to perform DML, DRL, TCL, and other commands used by the DBA(database administrator)

```
show databases;
```

It will list all the databases in our application.

```
create database sb101db;
```

It will create the database name **sb101db**

```
drop database sb101db;
```

It will delete the database from the MySQL

Note: After creating the database we need to move inside that database to perform some other operations.

Before inserting anything inside the database, Need to use the database using the following command.

```
use sb101db;
```

```
show tables;
```

To check created a table inside the sb101db database.

DDL commands

(create, alter, drop, truncate, rename)

1. create

```
create table student (roll int, name varchar(12), marks int);

or

create table student
(
roll int,
name varchar(12),
marks int
);

show tables; // TO CHECK TABLES

desc student; // TO CHECK THE Description of table
```

Datatypes in MySQL

1. numeric types:
 2. string types
 3. date and time types
 4. numeric types:
-

tinyint: 1byte

smallint: 2 byte

mediumint: 3 byte

int: 4 byte

bigint: 8 byte

floating point:

float(6, 2): the column can store 6 digits with 2 decimal places.

2. String type:

1. char: fixed length of string range bt 0 to 255 char.
2. varchar: variable length of string bt 1 to 65500, here we must define the max length.

char(4) vs varchar(4)

value char(4) Storage_required

'a' -----> 4 bytes

'ab' -----> 4 bytes

'abcdef' -----> error, data is too long

value varchar(4) Storage_required

'a' -----> 1 bytes

'ab' -----> 2 bytes

'abcdef' -----> error, data is too long

Note: In the term of efficiency, if we are storing with a variable length of character then we should use varchar, and if the length is always fixed then we should use char, here char is slightly faster than varchar.

3. Date and Time:

a. date: yyyy-mm-dd

b. datetime: yyyy-mm-dd hh:MM: ss

```
select * from the student; // trying to get all the columns and all the rows.
```

2. alter

- It is used to change the structure of the existing table.

this command has 4 sub-commands:

a. add

b. modify

c. drop

d. change

a. add: it is used to add a new column to the existing table.

ex:

```
alter table student add address varchar(15);
```

b. modify: it is used to change the column data type or its size.

```
alter table student modify address varchar(20);
```

c. drop: to drop single or multiple columns from a table:

This command is also used to drop any constraint from a particular column.

```
alter table student drop column address;
```

d. change: to rename a column;

```
alter table student change name sname varchar(12);
```

3. drop:

- To drop/delete the entire table (structure)from the database.

```
drop table student;
```

4. truncate:

- It is used to truncate/clean all the rows/records from the table permanently.
here table structures will not be deleted.

Note: all the DDL commands can not be rollback.

```
rename table student to student1;
```

DML (Data Manipulation Language)

(insert, update, delete)

These commands work with the data inside a table.

1. insert:

It is used to insert the records in the table:

```
insert into student values(10, 'Ram', 850);
```

Inserting partial column value

```
insert into student values(14, 'Amit', null);
insert into student(roll, name) values(15, 'Ravi')
insert into student(name, roll) values('pawan', 18);
```

2. update:

It is used to update the data within the table.

Ex:- the following command will set the value (update) for all the students.

```
update student set marks = 500;
```

Note: to update marks for only one student, here we need to use the 'where' clause.

```
update student set marks = 500 where roll = 14;
```

Examples:

```
update student set marks = 500 where roll = 14 OR name='pawan';

update student set marks = marks+50 where name = 'Ramesh';

update student set marks = marks+50 where marks <= 700;

update student set marks = 600 where marks IS NULL;
```

```
update student set name='Ram Kumar', marks = marks+20 where roll = 10;
```

2. delete:

It is used to delete the records/rows from the table.

```
delete from student;
```

It will delete all the records from the table, like the truncate command.

Note: truncate is the DDL command whereas delete is a DML command, DDL commands we can not rollback whereas DML commands can be rolled back.

```
delete from student where roll = 18;
```

It will delete the student whose roll is 18

DRL (Data Retrieval Language)

(select)

This command is used to query a table

Example 1: all the columns and all the rows.

```
select * from student; // all the columns and all the rows.
```

Example 2: restricting the number of rows by using the 'where' condition.

```
select * from student where roll = 10;  
select * from student where marks > 600;
```

Example 3: projecting few/single columns:

```
select name from student;  
  
select name, marks from student;  
  
select marks,roll, name from student;
```

Example 4: Using order by clause to sort the records:

```
select * from student order by marks;  
  
select * from student order by marks desc;
```

Operators:

1. Arithmetic operators: (*, /, + , -, %)

Note: mostly arithmetic operators are used after the select statements (90%) and all other types of operators are used inside the where clause only.

1. relational operators : (= , > ,< ,>=, <=, [!= or <>])
2. logical operators : (AND, OR, NOT)
3. special operators :(IS NULL, LIKE, BETWEEN, etc..)

Examples:

1. Arithmetic operators: (*, /, + , -, %)

```
select name, marks, marks+100 from student;  
  
select name, marks, marks+100 UpdatedMarks from student;
```

- *** This temporary name of a column we can not use inside the where clause.

Getting unique data (DISTINCT)

```
select DISTINCT marks from student;
```

Special Operators:

IN ... NOT IN

IS NULL IS NOT NULL

LIKE ... NOT LIKE

BETWEEN NOT BETWEEN

```
select * from student where marks IN(700, 550,600);
select * from student where marks BETWEEN 500 AND 700;
```

Or

```
select * from student where marks >=500 AND marks <=700;
```

LIKE ... NOT LIKE:

It is used to retrieve data based on character patterns.

1. % --> it represents the string or group of characters.
2. _ --> it represents a single character.

Example:

```
select * from student where name LIKE 'r%'; // name should start with 'r'.
```

Example: In name, r can be any character.

```
select * from student where name LIKE '%r%';
```

Example: r should be the 3rd character:

```
select * from student where name LIKE '__r%';
```