→ DBMS → Database management System.
→ RDBMS → Relational Database management System.

Both are used to store information in physical db.

DBMS	RDBMS
→ DBMS applications store data as files	Stores data in a tabular form
→ dbms can only be used by single users	rdbms can be used by multiple users
→ doesn't supports client-Server side interaction.	Supports client-Server side interaction.
→ dbms is lighter in its h/w and s/w requirements	Rdbms needs a powerful machine.
→ Normalization is not present in dbms.	Normalization is present in rdbms.
→ dbms use file s/m to store data. So there will be no relation b/w tables.	Rdbms stores data's as tables. So a relationship b/w these data's will be stored in the form of a table.

dbms	rdbms
→ dbms deals with small data.	rdbms deals with large amount of data.
→ dbms doesn't provide any security eg: xml, file s/m	rdbms is secure eg: mysql, Sql, Oracle etc.

Data

Data is a Collection of small unit of information.

Database.

→ Database is a Organised Collection of data, that can be easily accessed and managed.

→ The main purpose of db is to Operate a large amount of information by storing, retrieving and managing data.

→ There are many databases available like mysql, Sql, Oracle, mongo db etc.

→ Sql is used to perform Operation on data stored in a db.

→ A cylindrical structure is used to display the image of db.

SQL

→ SQL stands for Structured Query Language.

→ This db language is mainly designed for maintaining data in RDBMS.

→ It can easily create and manipulate db, access and modify the table, rows and columns etc.

→ It also helps to describe the structured data.

Why Sql is required

→ to create new db, tables
→ to insert record in a db
→ to update records in a db
→ to delete record from db
→ to retrieve record from db

What Sql does ?

→ We can query on db in a number of ways using english like statement.
→ With sql users can access data from rdbms.
→ It allows user to describe the data.
→ It allows user to create and drop db and table.
→ It allows user to define the data in db and manipulate it when needed.
→ It allows user to Create views, Stored Procedures in a db.
→ It allows user to set permission on table Procedures and views.

What is table ?

→ rdbms db uses table to store data.
→ A table is a Collection of related data entries and Contains rows and Columns to store data.
→ A table is the simplest example for data storage in rdbms.

What is field?

→ A field is a smallest entity of a table which contains specific information about every record in the table.

Record

rows in a table is called record. Row is the horizontal entity that contains specific information of each individual entry in the table.

→ Column is a vertical entity in a table which associates with specific field in a table.

→ The null value of the table specifies that the field has been left blank during record creation. It is totally different from the values filled with zeros or a field that contains space.

Data Integrity

It is used to maintain consistency and accuracy of data in a table

The following categories of data integrity exist with DBMS.

1. Entity Integrity
2. Domain Integrity

3. User defined Integrity
4. Referential Integrity

1. Entity Integrity

It specifies that there should be no duplicate rows in a column.

2. Domain Integrity

It contains valid entries for a given columns by restricting the type, format or range of values.

3. Referential Integrity

It specifies that rows cannot be deleted which are used by other records.

4. User Defined Integrity

It enforces some specific rules that are defined by the users.

Data Types

It is used to represent the nature of data that can be stored in a database table.

Data types can be mainly classified into three categories:-

1. String Data Type
   Char, Varchar, text, long text, medium text.
2. Numeric Data Type
   int, float, double, decimal
3. Date and Time
   Date, Time, year, timestamp

SQL Operators

SQL operators generally contains some reserved words or characters that are used to perform operations such as comparison, arithmetic etc. These reserved words are called operators.
There are three types of operators :-

1. Arithmetic operators
   +, -, *, /, %
2. Comparison operators
   =, >, <, >=, <=
3. Logical operators
   OR, NOT, AND, BETWEEN, IN, EXIST

SQL Constraints

→ It is used to specify rules for the data in a table.

→ They are used to limit the type of data that can go into a table. This ensures the accuracy of data in the table.

→ The following constraints are used in SQL:

* Not Null
* Unique
* Primary key
* Foreign key
* Check
* Default
* Create Index

* Not Null

Ensure that a column cannot have a null value.

* Unique

Ensures that all values in the column are different.

* Primary key

A combination of Not Null and Unique. It uniquely identifies each row in a table.

* foreign key

It is a field in one table that refers to the primary key in the another table.

It is used to prevent actions that would destroy links between the table.

* Check

Ensures that the value in a column satisfies a specific condition.

* Default

Set a default value for the column if no value is specified.

* Create Index

Used to create and retrieve data from the db very quickly.

SQL Commands

SQL Commands are instructions. It is used to communicate with db. It is used to perform specific task and queries of data.

SQL can perform various task like Create a table, add data to table, drop data in a table, modify table.

Types of SQL Commands

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

1. DDL (Data Definition language)

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

→ all Commands of ddl are autocommit that means permanently save all the changes in the db.

eg: Create, alter, drop, truncate.

2. DML (Data Manipulation language)

DML Commands are used to modify the db. It is responsible for all forms of change in the db.

Commands of dml are not auto committed, that means it cannot permanently save all the changes in the db, they can be rollback.

eg: insert, update, delete.

3. DCL (Data Control language)

→ used to grant and takeback authority from any db user.

eg: grant, revoke

grant → used to give user access privilege to a db

GRant Select, update on tb_name to some_user, another_user;

revoke → It is used to take back permission from the user

Revoke Select, update on tb_name from user_name;

4. 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 db. That's why they cannot be used while creating tables or dropping them.

Commit, rollback, Savepoint

Commit

used to save all transactions to the db.

Commit;

rollback - used to undo transactions.

Savepoint - used to roll the transaction back to a certain point with

rolling back the entire transaction
Savepoint Savepoint_name;

5. DQL (Data Query Language)
used to fetch data from db.
eg: Select:

Data Normalization

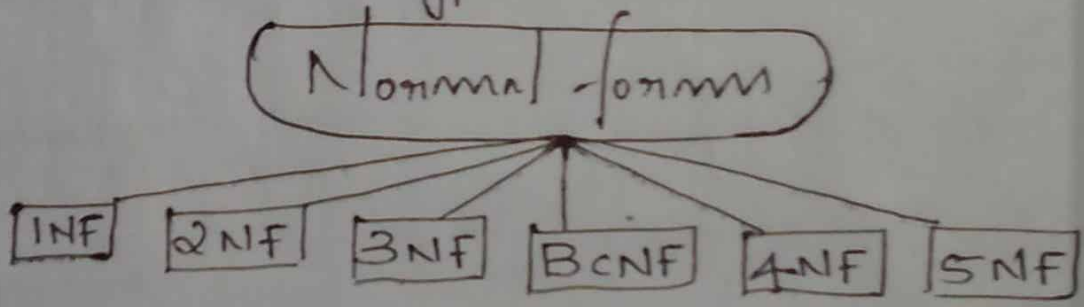
→ It is the process of Organising the data in db.

→ normalization divides larger tables into smaller tables and link them using relationship.

→ normal forms is used to reduce redundancy from db table.

Types of Normal forms

There are six types of normal forms.



First Normal form (1NF)

→ A relation will be 1NF, if it contains an atomic value (must be single valued attribute)

2NF

→ In 2NF, relation must be in 1NF.
→ In 2NF, all nonkey attributes are fully dependent on Primary key.

3NF

→ relation must be in 2NF
→ There should be no transitive dependency for non prime attributes
→ 3NF is used to reduce data duplication.

$a \to b$
$b \to c$
$\therefore a \to c$

BCNF (Boyce Codd Normal form)

* Advanced version of 3NF
* also Called 3.5 NF

→ relation must be in 3NF
→ for every functional, LHS is a Superkey.

Superkey → group of Single or multiple keys which identifies rows in a table.

4NF

→ relation must be in BCNF.
→ has no multivalued dependency

multivalued – for a dependency A → B
dependency for a single value of A,
multiple value of B exist

eg: telephone directory
name and address values are
dependent on phone number.

5NF

→ relation must be in 4NF.
→ Contains no join
→ also known as project join normal form.

Db Creation

```
Create database db_name ;
```

drop db

```
drop database db_name;
```

Select db

```
use db_name;
```

Table Creation

```
Create table table_name (
        Col1  datatype,
        Col2  datatype,
        Col3  datatype,
        ---------
    );
```

eg: Create table Stud (Stud_id int Primary key, Stud_name varchar(20), Stud_age int);

desc table_name;

Insert

(col1, col2,--)

```
Insert into table_name values (value1,
value2,--);
```

eg:

insert into stud (stud_id, stud_name,
age) values (1, 'shilpa', 22);

insert into stud (stud_id, stud_name
age) values (2, 'sruthi', 21);

Insert multiple values

insert into stud (stud_id, stud_name
age) values (1, 'shilpa', 22), (2, 'sruthi',
21);

Select Statement

```
Select * from table_name;
```

```
Show tables;
```

Select * from table_name where Cond;

eg: Select * from Stud where id = 2;

drop table table_name;

delete from table_name where Cond; (Remove a specific row from the table)

delete from table_name; (Removes all rows from the table)

Truncate – used to del all the rows from the table and free the containing space.

Truncate table table_name;

Drop

→ when we drop a table, the table structure, relationship will be deleted.

Truncate

→ when we truncate a table, the table structure remains.

update Statement

used to modify existing records in the table.

```
update table_name set Col1=value1,
Col2=value2,... where Condition.
```

eg: update Stud set Studname="Sru
where Stud_id = 1;

distinct Statement

used to return only different values

```
Select distinct Col1, Col2,... from
table_name;
```

eg: Select distinct Stud_name
from Stud;

Aliases

used to give a table, or a Column in a table, a temporary name.

→ alias is Created with 'As' keyword.

```
Select Col_name as alias_name from table_name;
```

eg: Select age as Stud_age from Stud;

min value

returns smallest value

```
Select min(Col_name) from table_name;
```

max value

returns highest value.

```
Select max(Col_name) from table_name;
```

Sum

returns total sum

```
Select Sum(Col_name) from table_name;
```

Average

returns average value.

```
Select avg(Col_name) from table_name;
```

Count

```
Select Count(Col_name) from table_name;
```

length

```
Select length(Col_name) from table_nm
```

limit

```
Select * from table_name limit 2;
```

And, or, not

And → displays a record if all the conditions seperated by And are true.

or → if any of the cond seperated by or are true.

not → displays a record if the cond is not true.

```
Select Col1, Col2 from tablename
where Cond1 And Cond2;
```

And or not

eg: Select * from Stud where name='Shilpa' and id='1';

Select * from Stud where name = 'Sruthi' or id = 1;

Select * from Stud where name = 'Shilpa' and not name = 'Sruthi';

In

→ Specifies multiple values in a where clause.

```
Select * from table_name where
Col_name in (val1, val2);
```

not in

eg:

Select * from Stud where name in ('Shilpa', 'Sruthi');

Select * from Stud where name not in ('Shilpa', 'Sruthi');

Order By

→ used to sort the result set in ascending or descending order.

→ ascending order by default.

```
Select * from table_name order by
Column_name asc/desc ;
```

eg:

```
Select * from Stud order by Stud_nam
desc;
Select * from Stud Order by Stud_nam
asc;
```

Trim

→ used to remove white spaces.

```
Select ltrim (Col_name) from table_name;
Select rtrim (Col_name) from table_nam
```

Select curdate();

Select Now();

Between

→ Selects values within a given range.
→ begin and end values are included.

Select * from table_name where Col_name between val1 and val2;

eg:
Select * from Stud where Stud-id between 1 and 7;

not Between → displays values outside the range.

Wild card characters

→ used to substitute one or more characters in a String.
→ wild Card characters are used with like Operator.

The like Operator is used in a where clause to search for a specified pattern in a Column.

% → represents zero, one or multiple characters

_ → represents one, single character.

a%- find any value starts with a.

%a — find any value Ends with a.

%ab% - finds any value that have 'ab' in any position.

_a% - find any value that have a in second position

__a% - find any value that have a in third position.

a__% — find any value that starting with a and have atleast 3 characters in length

a%b — starts with a and ends with b

Select Col_name from table_name where Col_name like "pattern";

eg:
Select name from Stud where name like "r%";

Alter table

→ used to add, modify or delete Columns in an existing table.

add Column — to add Columns

Alter table table_name Add Col_name datatype;

modify Column — to modify existing Col in the table.

Alter table table_name modify Col_name datatype;

drop Column → to drop a Column

Alter table table_name drop Col_name;

Null values

→ field with no value.

Is null

```
Select Col_name from table_name
where Col_name is null;
```

Is not null

```
Select Col_name from table_name
where Col_name is not null;
```

Exists

→ used to test for the existence of any record.

```
Select Col_name from table_name
where exists (Select Col_name from
table_name where Condition);
```

Case

→ Case Statement works like if Statement, uses the keyword WHEN

→ evaluated from top to bottom

```
Select Col_name, Case when Cond1
then result1 when Cond2 then result2
- - - - -
when Condn then resultn
else result
end from table_name;
```

eg:

Studid	name	marks
1	abc	472
2	efg	430
3	hij	320
4	klm	400
5	nop	222
6	qrs	220

```
Select Stud_id, name, marks, Case when
marks > 250 then 'PASSED' else
'FAILED' end as result from table_name;
```

Group by

→ used to group the data on the basis of certain Criteria.

→ aggregate functions are Calculated for each group.

eg:

emp (eid, name, year of joining, dep, Salary)

determine the max Sal of emp in each department.

```
Select * from table_name group by
Col_name;
```

Select dept, max(Salary) from emp group by dept;

Having

→ used to place Condition on group by

→ using having clause we Can Perform filteration after group by.

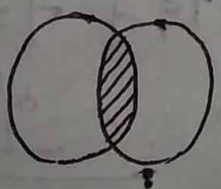
Select * from table_name group by Col_name having Condition;

Joins

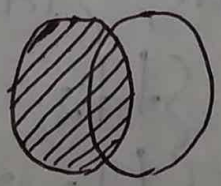
used to Combine rows from two or more table based on a related Column.

Four types of joins

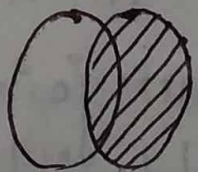
1. Inner join
2. left join
3. right join
4. Outer join.



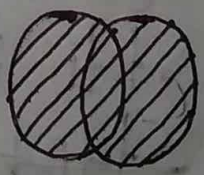
Inner join



left join



right join



full outer join

Inner join

Selects records that have matching values in both tables.

```
Select Col_name from tablename1
inner join tablename2 on relationship;
```

eg:

$std1

id	name	grade
1	abc	A
2	def	B
3	ghi	C
4	jkl	D
5	mno	E

$std2

$stid	amount	id
600	1500	2
601	200	3

```
Select $std1.name, $std2.$stid from
$std1 inner join $std2 on
$std1.id = $std2.id;
```

o/p

name	$stdid
def	600
ghi	601

Left Join

returns all records from the left table and matching record from the right table.

Select Col_name from table1 left join table2 on relationship;

Select std1.name, std2.sid from std1 leftjoin std2 on std1.id = std2.id;

o/p

name	sid
def	600
ghi	601
jkl	null
mno	null
abc	null

Right join

returns all records from the right table and matching records from left table.

Select Col_name from table1 right join table2 on relationship;

Select std1.name, std2.stid from std1 right join std2 on std1.id = std2.id;

Outer join

return all records when there is a match in either left or right table..

Select * from table1 join table2 on relationship.

Select * from std1 join std2 on std1.id = std2.id;

o/p

id	name	grade	stid	amount	id
2	def	B	600	1500	2
3	ghi	C	601	200	3