Introduction

What is Front end application?

The application where user will interacts is called frontend application.

These are developed by using frontend tools like .Net, Java, PHP, Etc..

EG:- Facebook. Gmail…

What is Back end application?

The application where exactly user data will be maintained is called Backend application.

Backend application will be developed by using Backend tools like MS SQL Server, MY SQL, Excel, D2K,.Etc..

MS SQL Server is a product of -Microsoft.

What is Database?

A Collection of meaningful/ processed / organized data is called as Database.A database is a place to store data.
The main purpose of database is to operate large amount of information by storing, retrieving and managing.

There are many databases available like MySQL, Sybase, Oracle, Mango DB, Informix, SQL Server etc.

Eg:-

1) 111 Rama 25000 27

The above data is meaning less data

2) Rama Salary was 25000

3) Rama age is 27

The above data was meaningful / Processed data

In any database we will work on Data, it will occupy less memory.

What is SQL

- SQL stands for **Structured Query Language**.
- SQL is just a query language, it is not a database. To perform SQL queries, you need to install any database.
- It is designed for managing data in a relational database management system (RDBMS).
- It is pronounced as S-Q-L or sometime See-Qwell.

Why SQL is required

- 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

What is RDBMS

RDBMS stands for Relational Database Management Systems.

All modern database management systems like MS SQL Server, IBM DB2, ORACLE, My-SQL and Microsoft Access are based on RDBMS.

Difference between DBMS and RDBMS

DBMS	RDBMS
DBMS applications store ***data as file***.	RDBMS applications store ***data in a tabular form***.
DBMS does not apply any security with regards to data manipulation.	RDBMS defines the integrityconstraint for the purpose of ACID (Atomocity, Consistency, Isolation and Durability) property.
DBMS uses file system to store data, so there will be ***no relation between the tables***.	In RDBMS, data values are stored in the form of tables, so a relationship between these data values will be stored in the form of a table as well.
DBMS ***does not support distributed database***.	RDBMS ***supports distributed database***.
DBMS is meant to be for small organization and ***deal with small data***. it supports ***single user***.	RDBMS is designed to handle large amount of data. it supports multiple users.
Examples of DBMS are file systems, ***xml*** etc.	Example of RDBMS are ***mysql***, ***postgre***, ***sql server***, ***oracle*** etc.

Degrees of Relations ships in RDBMS

1) One to one relation
2) One to many relation
3) Many to many relation

One to one relation:-

A row of a table is associated with a row in another table called one to one relation.

One to many relation:-

A row of a table is associated with multiple rows in another table is called one to many relation.

Many to many relation:-

Many rows in a table is associated with many rows in another table is called as Many to many relation.

What is table

The RDBMS database uses tables(Object) 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 of data storage in RDBMS.

What is field

Field is a smaller entity of the table which contains specific information about every record in the table.

What is row or record

A row of a table is also called record. It contains the specific information of each individual entry in the table. It is a horizontal entity in the table.

What is column

A column is a vertical entity in the table which contains all information associated with a specific field in a table. For example: "name" is a column in that column which contains all information about name column.

NULL Values

The NULL value of the table specifies that the field has been left blank during record creation. It is totally different from the value filled with zero or a field that contains space. By default every column will accept the NULL values.

SQL Syntax

SQL is not case sensitive. Generally SQL keywords are written in uppercase.

SQL statements are dependent on text lines. We can place a single SQL statement on one or multiple text lines.

SQL statements are started with any of the SQL commands/keywords like SELECT, INSERT, UPDATE, DELETE, ALTER, DROP etc. and the statement ends with a semicolon (;).

Example of SQL statement:

SELECT "column_name" **FROM** "table_name";

Why semicolon is used after SQL statements

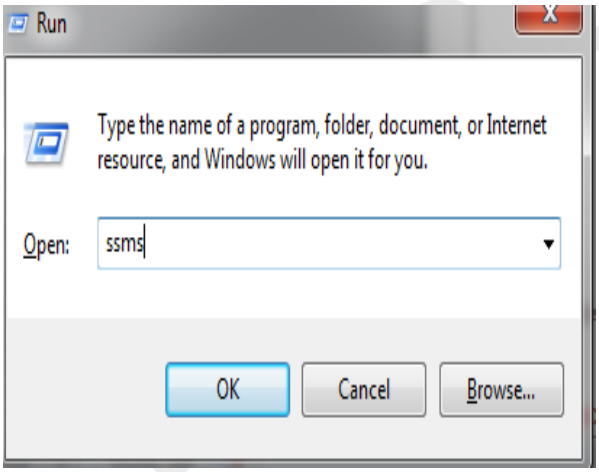
Semicolon is used to separate SQL statements. It is a standard way to separate SQL statements in a database system in which more than one SQL statements are used in the same call.

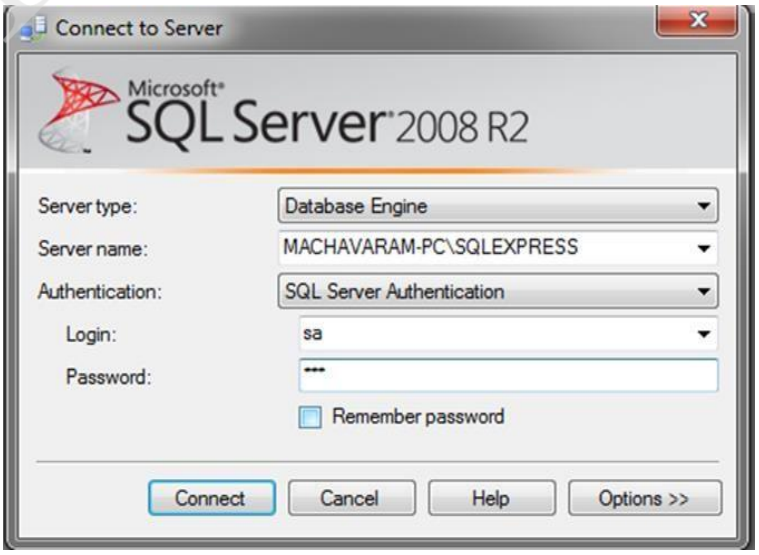
Steps to open MS SQL Server

Start → Programs → MS SQL server 2008 / 12/ 14 → SQL Server Management Studio

(OR)

Go to Run (Windows +R) type -ssms then OK it will open SQL Server home page





Server Type:-

Server type= Database Engine

This will use to store the large amount of data processing the data and providing security. Here data will store I 2D format (Rows& columns).

By using SQL Server we can develop

- SQL Server Reporting Services [SSRS]
- SQL Server Analysis Services [SSAS]
- SQL Server Integration Services[SSIS]

Server Name:-

It is used to set the server name where SQL server was installed.

Authentication:-

It is the process of checking credentials means Username & Password

We can connect to SQL Server database in 2 ways

→ By using Windows Authentication

→ By using SQL Server Authentication

When we connect to SQL Server through Windows authentication no need to give Credentials.

Default user name is -**sa** (System Administrator)

SQL Commands

The standard SQL commands to interact with relational databases are CREATE, SELECT, INSERT, UPDATE, DELETE and DROP. These commands can be classified into groups based on their nature:

- Data Definition Language (DDL)
- Data Manipulation Language (DML)
- Data Retrieval Language (DRL)
- Transaction Control Language (TCL)
- Data Control Language (DCL)

Command	Command	Description
DDL	CREATE	Creates a new table, a view of a table, or other object in database
	ALTER	Modifies an existing database object, such as a table.
	DROP	Deletes an entire table, a view of a table or other object in the database.
DML	INSERT	Creates a record
	UPDATE	Modifies records
	DELETE	Deletes records in the table
DRL	SELECT	Retrieves certain records from one or more tables
TCL	commit	It commits the transaction
	rollback	Cancelling the transaction to particular point (Save point)
	savepoint	Save the transaction gives a name
DCL	GRANT	Gives a privilege to user
	REVOKE	Takes back privileges granted from user

SQL Data Types

The SQL data type defines a kind of value that a column can contain.

In a database table, every column is required to have a name and a data type.

Data-type	Syntax	Explanation
Integer	INTEGER	The integer data type is used to specify an integer value.
Character	Char	To store the single (or) group of characters, Size of Char is fixed. Max size:8000 characters
Varchar	Varchar	To store the single (or) group of characters, Size of Char is not fixed.
Text	Text	Similar to varchar(max)
Nchar	nchar	The size of nchar is fixed like char. By using this we can store National Charater data i.e if we want to store Chinese, Japanese, French are any other different language then we go for nchar Max size:8000 characters
Real	REAL	The real integer is used to specify a single precision floating point number.
Decimal	DECIMAL(P,S)	It specifies a decimal value. Here 'p' is precision value and 's' is scale value.
Double precision	DOUBLE PRECISION	It specifies double precision floating point number.
Float	FLOAT(P)	It specifies floating-point value e.g. 12.3, 4.5 etc. Here, 'p' is precision value.
Date	DATE	It stores year, month and days values.
Time	TIME	It stores hour, minute and second values
Money	MONEY	It stores the money related values, like salary.

SQL Create Database

Databases are basically divided into 2 types.

1. System defined database
2. User defined database

System Defined Database:-

These databases are installed automatically when we install SQL Server Management Studio.

Different types of system defined databases are

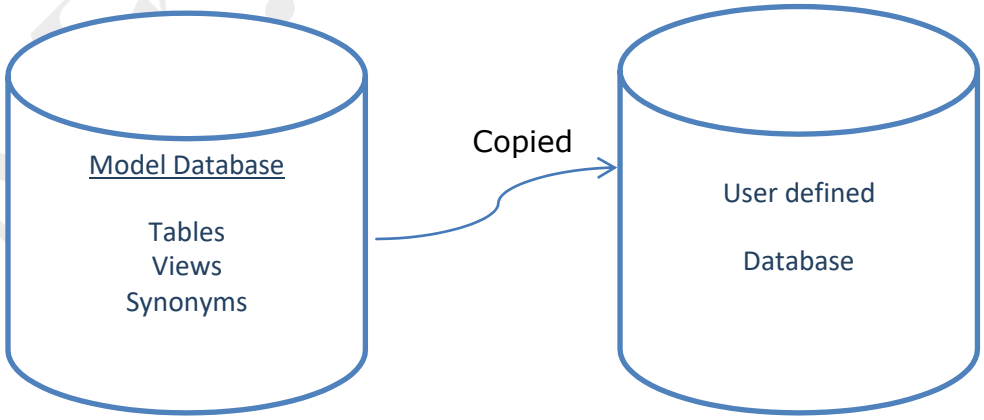
1. Master
2. Model
3. Temp db
4. MSdb

Master:-

- It consists of configuration settings that are required to run the SQL Server Management studio.
- It is used to maintain the user credentials & Authentication

Model:-

- Model database is a Template for all the other databases
- Whenever we are creating user defined database then the contents of model database are copied into user defined database.



Temp db:-

- It is used to perform temporary calculations and operations.

MS db:-

- It is used to schedule jobs & alerts

->To view all database details in system

```
Select * from sysdatabases;
```

-> To view all system user details

```
Select * from sysusers;
```

-> To view all login details..

```
Select * from syslogins;
```

-> To view all object details in the system

```
Select * from sysobjects;
```

User defined database:-

The database that was created depending on the user requirement is called user defined database.

Eg: - College, Employee, School…

Note:-

Once we create a new database in is SQL server then the system will create 2 types of datafiles for each and every database in sql server.

Those are

1) Primary Data file(.mdf file):-

It will store all database tables data with an extension of **.mdf** (Master Data File).

2) Log data file (.ldf file):-

It will store transaction or overall query information which was executed by the user on database objects (Tables, Synonyms, Views, etc...) will save with an extension of **.ldf**(log data file)

The above two files are used to transfer the database information one system other system / location.

DDL Commands

- These commands are used for create database(or) database objects like Table, Views, Synonyms, stored procedures , etc…
- By using DDL we can modify the database objects like adding a new column, removing a column, change the data type name etc..
- By using DDL commands we can remove database (or) Tables.

Creating Database (User defined)

// Syntax for creating a database

Syntax

CREATE DATABASE database_name;

Eg:- create database Companydb;

Select query and press ‗F5' to execute.

- If you want to add tables in that database, you can use CREATE TABLE statement.
- Always database name should be unique within the RDBMS.

// Write a query to view the information of the database.

Syntax

SP_helpdb databasename;

*Eg:-*sp_helpdb Companydb;

- When we execute the above query it will display entire information of database like database name, owner name, size, creation time etc,…..
- SP_helpdb is predefined stored procedure which was written by Microsoft.

// Write a query to use database.

Syntax

Use databasename;

*Eg:-*use Companydb;

// Write a query to rename the database.

Syntax

SP_renamedb old_databasename, new_databsename;

*Eg:-*sp_renamedb Companydb,Company;

// Write a query to remove the database

- If you want to delete or drop an existing database in a SQL schema, you can use SQL DROP DATABASE

Syntax

Drop database databasename;

*Eg:-*drop database Company;

Creating Table (User defined)

// Syntax for creating a table

Syntax

```
Create table "tablename"("column1" "data type",
"column2" "data type", -column3" "data type",..
"columnN" "data type");
```

Eg:-

Create table emp (empid int,empname varchar(10), designation varchar(10),salary money, doj date);

// Write a query to view the information of the table.

Syntax

SP_help tablename;

*Eg:-*sp_help emp;

Points to remember at the time of Table creation:-

- A table name should be unique within the same database.
- A column name should be unique in the same table definition.
- A table name should not start with numbers, special characters except (_) underscore symbol.
- Don't provide space in the table name, if required can be used underscore only.
- Don't uses the reserved keywords like Insert, Delete, Update and Select as a table name.
- A table name should contain minimum of 1 character and maximum 128 characters.
- A table should contain min 1 column and max 1024 columns.
- A table contains unlimited rows.

Alter Command:-

- This command is used to do modifications for database objects like tables, views, synonyms, stored procedures, triggers…..
- By using alter command we can add a new column for existing table as well as we can remove a column from the table.
- By using alter command we change the data type name of column.
- We can change the size of thedata type for column(s).

// Write a query to add a new column in emp table with name Gender.

Syntax

Alter table tablename add Columnnamedatatype;

*Eg:-*Alter table emp add Gender char(1);

Note: - we can add a new column even if the table consists of data.

// Write a query to change datatype name of empid column from int to bigint.

Syntax

Alter table tablename alter column Columnname new datatype;

*Eg:-*Alter table emp alter column empid bigint;

// Write a query to change the size of datatype from varchar(10) to varchar(50) for empname .

Syntax

Alter table tablename alter column Columnname new datatype size;

*Eg:-*Alter table emp alter column empname varchar(50);

Truncate Command:-

- This command is used to remove the data from the table

Syntax

Truncate table tablename;

Eg:-Truncate table emp;

- Before we are using this command table should have some records to perform this command.

Drop Command:-

- This command is used to remove the data from the table along with table structure.

Syntax

Drop table tablename;

Eg:-Drop table emp;

// Write a query to remove doj column from emp table.

Syntax

Alter table tablename drop column coumnname;

Eg:- Alter table emp drop column doj;

// Write a query to rename the table.

Syntax

Sp_rename <tablename>,<New_table_name>;

Eg:- sp_rename emp , emp_tbl;

// Write a query to rename the column name in table.

Syntax

Sp_rename 'tablename.oldcolumnname', 'Newcolumnname;

Eg:- sp_rename 'emp-tbl.designation','cader';

// Write a query to copy the table in new table.

Syntax

*Select * into <destination_table> FROM <source_table>*
Eg:-select * Into emp_tbl_new From emp_tbl;

- It will copy the entire table with data in another new table.

DML Commands

- These commands are used to Insert (or) Delete (or) Update the records in the Table.

Insert:-

It is used to insert the record into the table.

Syntax (Implicit method)

Insert into <tablename> values (Val1,Val2, Val3…);

Eg:-

Insert into emp_tbl values (111,'Rama','CEO',25000,'M');

Note:-

The above insert query is used to insert all the records in the table. If you want to insert specific records in the table we have to use

Syntax (Explicit method)

Insert into <tablename> (Colname)values(value of column);

Eg:-

Insert into emp_tbl(empname,salary)values ('Basha',12000);

Inserting multiple records into the table:-

Syntax

Insert into <table Name>
(col1,col2, col3) values (Col1value, Col2value, Col3value),;

Eg:-

Insert into emp_tbl(empid,empname,cader,salary,Gender) values (123,'lakshman','MD',18000,'M'), (121,'sita','HR',2000,'F'), (124,'Ghouse','GM',20000,'M');

-- with the aboue statement we inserted 3 records at a time.

Identity (Seed, Increment)

- This is a predefined function which will use to generate numeric values to particular column in a table.
- This function contains 2 arguments
  - Seed :- will represent starting / initial id value
  - Increment: - will represent incremental value between id's.
- The default value of seed & increment is (1,1)

Example:-

Create table Tbl_Emp (Eid int identity , ename varchar(20), age int)

Insert into Tbl_Emp values (1,'sai',29);// not allowed
Insert into Tbl_Emp values ('sai',29);// allowed

Note:-

If the user want to insert the values into identity column by explicitly following syntax for that

Syntax:-

Set identity_insert <Tbl_Name> on/off;

E.g:-

Set identity_insert Tbl_Emp off;

Delete:-

It is used to delete specific records (or) group of records from table.

Use the WHERE clause to Delete only specific records.

Differences between Truncate and Delete

Truncate	Delete
Its DLL Command	Its DML Command
Using this we can't delete specific record	We can delete specific record from table
By using this we can delete all the records from table	We can delete all the records from the table
We can't roll back the deleted data	We can roll back the deleted data (restoring to table)
Where **condition** will not work	Where **condition** will work
Truncate will delete all the rows at a time	Delete will delete the rows on by one
It will work faster	It will work slowly

// Write a query to delete the table data (all records).

Syntax

Delete from <table-name>;

Eg:-Delete from emp_tbl;

// Write a Delete query on condition.

Syntax

Delete from <table-name> where <condition>;

Eg:-

Delete from emp_tbl where empid=123;

Delete from emp_tbl where empname='Basha';

Delete from emp_tbl where salary>=10000;

Update:-

It is used to update the data which is available in the table.

- UPDATE can update one or more records in a table.
- Use the WHERE clause to UPDATE only specific records.

Syntax

Update <table_name> set <col1> = value1 where<condition>;

Eg:-

Update emp_tbl set salary=20000 where empid=121;

Update emp_tbl set name=Hari where empid=161;

DRL Commands

This command is used to display a specific records (or) group of records from Table.

- The SELECT statement retrieves data from a database.
- The data is returned in a table-like structure called a result-set.
- Select is the most frequently used action on a database.

Syntax

Select * *from <table_name>;*

Eg:-

Select * from emp_tbl;

Select empid , empname, salary from emp_tbl;

Select * from emp_tbl where empid=121;

Operators

- Operators are used to perform operation on two (or) more operands.
- Operators are classified into 5 types
  - Arithmetical Operators
  - Logical operators
  - Comparison operators
  - Range operators
  - String Operators

Arithmetical Operators:-

These operators are used to perform arithmetical operation`s like Addition, Div, Sub, Mul, Division…

Operators	Descriptions
+	It is used to add containing values of both operands
-	It subtracts right hand operand from left hand operand
*	It multiply both operand`s values
/	It divides left hand operand by right hand operand and returns Quotient
%	It divides left hand operand by right hand operand and returns Reminder

What is an Expression?

- An expression is the combination of 2 (or) more operators.
- Whenever we are evaluating an expression operation we have to evaluate based on the priority of the operation.

Eg:- 23+4-5*2+7*2-4/2

1st Priority → *,/,%
2nd Priority → +,-
3rd Priority → =

Note:- if numerator < zero the quotient =0 & numerator = reminder.

Eg 1:- Create a table with name student with columns stno, stname, sub1, sub2, sub3.

```
Create table student
        (
        stno int, stname varchar(15),sub1 int, sub2 int,sub3 int
        );
Insert into student values (002,'Aashrith',80,96,70);
```

// WAQ to display student details.

```
Select * from student;
```

// WAQ to display student details along with total marks.

```
Select *, (sub1+sub2+sub3) from  student ;
```

o/p

stno	stname	sub1	sub2	sub3	(No column name)
1	Aashrith	80	96	70	246

```
Select *,(sub1+sub2+sub3) as'Total' from  student  ;
```

O/P (create a column as Total name)

stno	stname	sub1	sub2	sub3	Total
1	Aashrith	80	96	70	246

// WAQ to display details along with Total marks , % of marks.

```
Select *,(sub1+sub2+sub3) as 'Total',(sub1+sub2+sub3)/3 as 'Percentage'
from student ;
```

O/P

Stno	stname	sub1	sub2	sub3	Total	Percentage
1	Aashrith	80	96	70	246	82

Eg 2:- Create a table with name Employee with columns EID, Ename, basicsal.

Create table Employee(EID int, Ename varchar(10), BasicSal money);

// WAQ to display Employee details along with DA, HRA, Gross.

DA= 0.4*BasicSal.

HRA=0.6*BasicSal.

Gross=BasicSal+DA+HRA.

Select*,(0.4*Basicsal) as 'DA',(0.6*BasicSal) as 'HRA' , (BasicSal+(0.4*BasicSal) + (0.6*BasicSal)) as 'Gross' from Employee;

O/P

EID	Ename	BasicSal	DA	HRA	Gross
189	Sekhar	20000.	8000	12000.	40000.

// WAQ to display total Salary of Sekhar.

Select *,(BasicSal+(0.4*BasicSal) + (0.6*BasicSal)) as 'Total' from Employee where Ename='Raja';

To Rename a Colun in the Table:-

Syntax:-

Sp_Rename ˹Table_Name.Column_Name‘,‘New_Column‘, ˹Column‘;

Logical operators

There are three Logical Operators namely, AND, OR, and NOT. These operators compare two conditions at a time to determine whether a row can be selected for the output. When retrieving data using a SELECT statement, you can use logical operators in the WHERE clause, which allows you to combine more than one condition.

S.No	Logical Operators	Description
1	OR	For the row to be selected at least one of the conditions must be true.
2	AND	For a row to be selected all the specified conditions must be true.
3	NOT	For a row to be selected the specified condition must be false.

Note:-Create a table-"Student_Tbl" to perform logicaloperators on that table.

1. "OR"

If you want to select rows that satisfy at least one of the given conditions, you can use the logical operator, ‗OR'.

Eg: - If you want to find the names of students who are studying either Maths or Science from the table Student_tbl, the query would be like,

Query:-

```
SELECT first_name, last_name, subject FROM student_tbl
WHERE subject = 'Maths' OR subject = 'Science' ;
```

Column1 Satisfied?	Column2 Satisfied?	Row Selected
YES	YES	YES
YES	NO	YES
NO	YES	YES
NO	NO	NO

2."AND"

If you want to select rows that must satisfy all the given conditions, you can use the logical operator, 'AND'.

Eg:- To find the names of the students between the age 10 to 15 yearsfrom the table Student_tbl, the query would be like:

Query:-

```
Select first_name, last_name, age from Student_tbl
where age >= 10 AND age <= 15;
```

Column1 Satisfied?	Column2 Satisfied?	Row Selected
YES	YES	YES
YES	NO	NO
NO	YES	NO
NO	NO	NO

3."NOT"

If you want to find rows that do not satisfy a condition, you can use the logical operator 'NOT'. NOT results in the reverse of a condition. That is, if a condition is satisfied, then the row is not returned.

EG: - If you want to find out the names of the students who do not play football, the query would be like:

Query:-

```
Select first_name, last_name, games from Student_Tbl
where NOT games = 'Football';
```

Comparison Operators:

Comparison operators are used to compare the column data with specific values in a condition. And also used along with the SELECT statement to filter data based on specific conditions.

Comparison Operators	Description
=	equal to
<>, !=	is not equal to
<	less than
>	greater than
>=	greater than or equal to
<=	less than or equal to

Queries:-

1) Select * from *Student_Tbl* where Fee>15000;
2) Select * from *Student_Tbl* where Fee<8000;
3) Select * from *Student_Tbl* where Fee=10000;
4) Select * from *Student_Tbl* where Fee != 18000;
5) Select * from *Student_Tbl* where Fee <> 18000;

Range Operators:

These operators which selects data according to particular range.

2types of Range operators are there,

1) Between
2) Not between

Between:-

This will select the data between any ranges.The values can be numbers, text, or dates.

Example Query:-

*1) Select * from Student_Tbl where Fee between 10000 and 12000;*

With this query it will select data between 10000 and 12000.

Not between:-

To display the data outside the range of the previous example, use not between.

Example Query:-

*Select * from Student_Tbl where Fee not between 10000 and 12000;*

String operators:-

These operators are used in where clause to search the data by using a specific pattern. These are two types,

1. Like (Like keyword allows you to select records that match with pattern)
2. Not like (Notkeyword allows you to select records that do Not match the pattern.)

Eg:-

S% → String starts with'S'.
%S → String ends with 'S'.
%S% → in between 'S'.

Example Queries:-

*1) Select * from Student_Tbl where sname like 'S%';*

It will display all the names starts with 'S' letter.

*2) Select * from Student_Tbl where sname like 'S%' and Fee>12000;*

It will display all the names starts with ‗S' and Who's Fee is greater than12000.

*3) Select * from Student_Tbl where sname like 'S%S';*

It will display all the names starts with ‗S' and end's with'S'.

*4) SELECT * FROM Student_Tbl WHERE Country NOT LIKE '%Ind%';*

The above SQL statement selects all students with Country NOT containing the pattern "Ind".