Structure Query Language (SQL)

Example to Select all Records from Table

A special character **asterisk** * is used to address all the data(belonging to all columns) in a query. *SELECT* statement uses * character to retrieve all records from a table.

```
SELECT * from student;
```

The above query will show all the records of Student table, that means it will show complete Student table as result.

S_id	S_Name	Age	address
101	Adam	15	Noida
102	Alex	18	Delhi
103	Abhi	17	Rohtak
104	Ankit	22	Panipat

Example to Select particular Record based on Condition

```
SELECT * from Student WHERE s_name = 'Abhi';
```

103	Abhi	17	Rohtak

Example to Perform Simple Calculations using Select Query

Conside the following **Employee** table.

Eid	Name	Age	Salary
101	Adam	26	5000

102	Ricky	42	8000
103	Abhi	22	10000
104	Rohan	35	5000

```
SELECT eid, name, salary+3000  from Employee;
```

The above command will display a new column in the result, showing 3000 added into existing salaries of the employees.

Eid	Name	salary+3000
101	Adam	8000
102	Ricky	11000
103	Abhi	13000
104	Rohan	8000

6.9 Like clause

Like clause is used as condition in SQL query. **Like** clause compares data with an expression using wildcard operators. It is used to find similar data from the table.

Wildcard operators

There are two wildcard operators that are used in like clause.

- **Percent sign** % : represents zero, one or more than one character.
- **Underscore sign** _ : represents only one character.

Example of LIKE clause

Consider the following **Student** table.

s_id	s_Name	Age
101	Adam	15
102	Alex	18
103	Abhi	17

```
SELECT * from Student where s_name like 'A%';
```

The above query will return all records where **s_name** starts with character 'A'.

s_id	s_Name	Age
101	Adam	15
102	Alex	18
103	Abhi	17

Example:

```
SELECT * from Student where s_name like '_d%';
```

The above query will return all records from **Student** table where **s_name** contain 'd' as second character.

s_id	s_Name	Age
101	Adam	15

Example:

```
SELECT * from Student where s_name like '%x';
```

The above query will return all records from **Student** table where **s_name** contain 'x' as last character.

s_id	s_Name	Age

102	Alex	18

6.9 Order By Clause

Order by clause is used with **Select** statement for arranging retrieved data in sorted order. The **Order by**clause by default sort data in ascending order. To sort data in descending order **DESC** keyword is used with**Order by** clause.

Syntax of Order By

```
SELECT column-list|* from table-name order by asc|desc;
```

Example using Order by

Consider the following **Emp** table,

Eid	Name	Age	salary
401	Anu	22	9000
402	Shane	29	8000
403	Rohan	34	6000
404	Scott	44	10000
405	Tiger	35	8000

```
SELECT * from Emp order by salary;
```

The above query will return result in ascending order of the **salary**.

Eid	Name	Age	salary
403	Rohan	34	6000
402	Shane	29	8000

405	Tiger	35	8000
401	Anu	22	9000
404	Scott	44	10000

Example of Order by DESC

Consider the **Emp** table described above,

```
SELECT * from Emp order by salary DESC;
```

The above query will return result in descending order of the **salary**.

Eid	Name	age	Salary
404	Scott	44	10000
401	Anu	22	9000
405	Tiger	35	8000
402	Shane	29	8000
403	Rohan	34	6000

6.10 HAVING Clause

Having clause is use with SQL Queries to give more precise condition for a statement. It is used to mention condition in Group based SQL functions, just like WHERE clause.

Syntax for having will be,

```
select column_name, function(column_name)

FROM table_name

WHERE column_name condition

GROUP BY column_name

HAVING function(column_name) condition
```

Example of HAVING Statement

Consider the following **Sale** table.

oid	order_name	previous_balance	customer
11	ord1	2000	Alex
12	ord2	1000	Adam
13	ord3	2000	Abhi
14	ord4	1000	Adam
15	ord5	2000	Alex

Suppose we want to find the customer whose previous_balance sum is more than 3000.

We will use the below SQL query,

```
SELECT *

from sale group customer

having sum(previous_balance) > 3000
```

Result will be,

oid	order_name	previous_balance	customer
11	ord1	2000	Alex

6.11 Distinct keyword

The **distinct** keyword is used with **Select** statement to retrieve unique values from the table. **Distinct** removes all the duplicate records while retrieving from database.

Syntax for DISTINCT Keyword

```
SELECT distinct column-name from table-name;
```

Example

Consider the following **Emp** table.

Eid	Name	Age	Salary
401	Anu	22	5000
402	Shane	29	8000
403	Rohan	34	10000
404	Scott	44	10000
405	Tiger	35	8000

```
select distinct salary from Emp;
```

The above query will return only the unique salary from **Emp** table

salary
5000
8000
10000

6.12 AND & OR operator

AND and **OR** operators are used with **Where** clause to make more precise conditions for fetching data from database by combining more than one condition together.

6.12.1 AND operator

AND operator is used to set multiple conditions with *Where* clause.

Example of AND

Consider the following **Emp** table

Eid	Name	Age	Salary
401	Anu	22	5000
402	Shane	29	8000
403	Rohan	34	12000
404	Scott	44	10000
405	Tiger	35	9000

```
SELECT * from Emp WHERE salary < 10000 AND age > 25
```

The above query will return records where salary is less than 10000 and age greater than 25.

Eid	Name	Age	Salary
402	Shane	29	8000
405	Tiger	35	9000