**Operators**

**Operators:**

* Operators are **nothing but symbols** which is used to **perform** some specific **operation**.

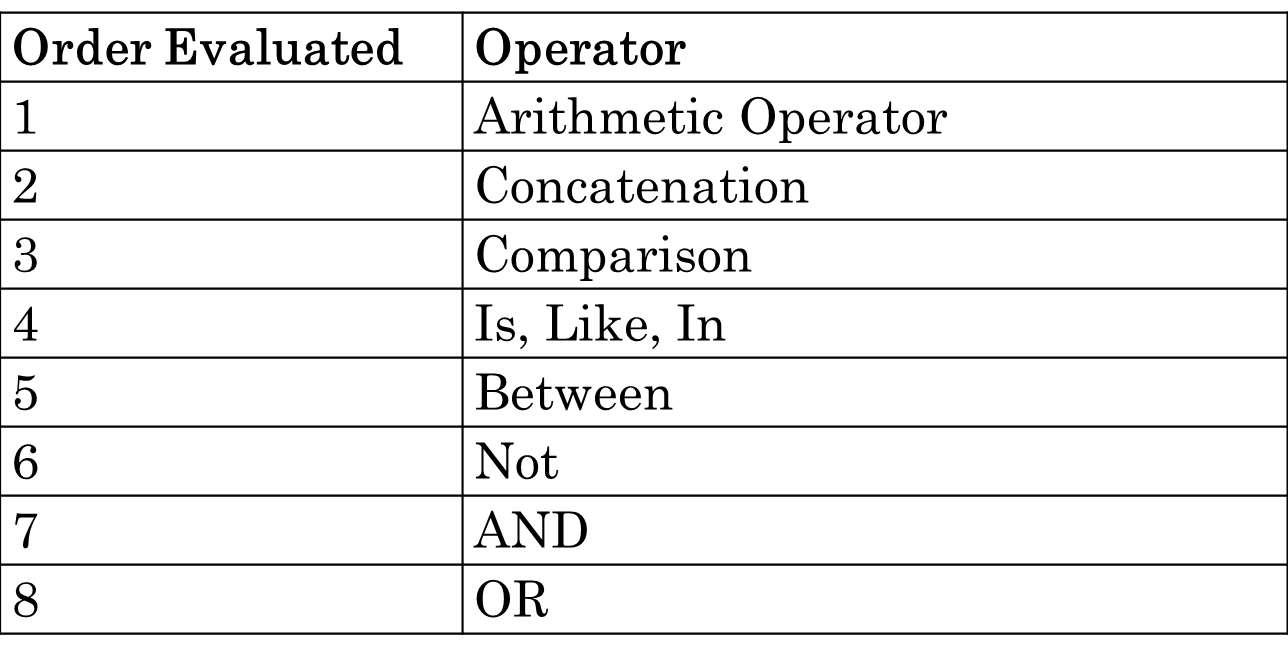
**Operands:**

* Operands are the **inputs required for the operation.**

Operators are classified into,

1. **Arithmetic Operators** ( +, - , \* , / )
2. **Relational Operators** ( > , < , >= , <= , = , < > **or** != - not equals to )
3. **Logical Operators** ( NOT, AND, OR )
4. **Special Operators** ( IN , LIKE , BETWEEN , IS )

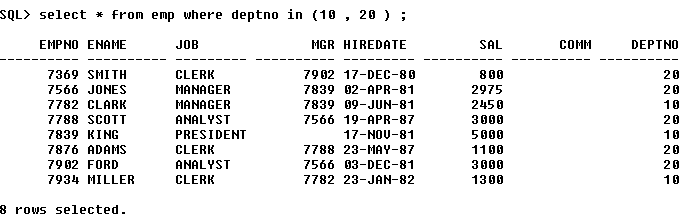
**Rules for precedence**



**SPECIAL OPERATORS**

**1) IN** – it is used for **evaluating multiple values.**

**Ex – 1)** List the employees in dept 10 & 20

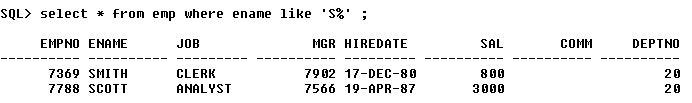


**2) LIKE** – used for pattern matching

**% (percentage) -** matches 0 or ‘n’ characters

**\_ (underscore)** - matches exactly one character

**Ex – 1) List all the employees whose name starts with ‘S’**



Whenever we use % or \_ , always ensure that it is preceded by the word **‘like’**

**BETWEEN:**

1. It is used to **evaluate the range of value**

**Syntax:**

Select \* from table\_name where columnname between value1 and value2;

**IS:**

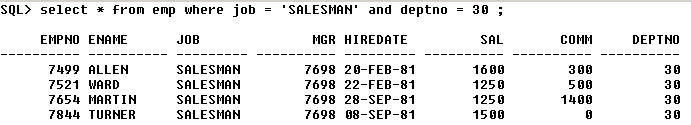
1. It is used to **compare null value.**

**Syntax:**

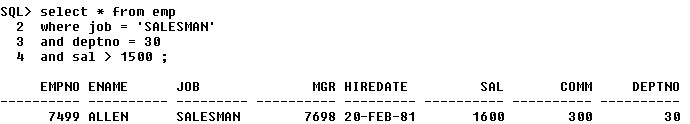
Select \* from table\_name where columnname is null;

**LOGICAL OPERATORS**

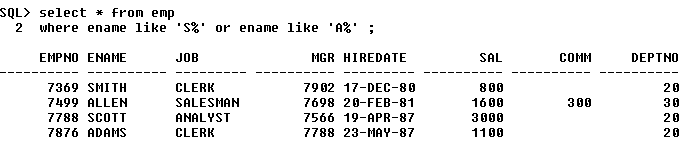
**1) List all the salesmen in dept 30**

****

**2) List all the salesmen in dept number 30 and having salary greater than 1500**

****

**3) List all the employees whose name starts with ‘s’ or ‘a’**

****

**Order by:**

1. Order by class is used to **sort the data ascending or descending order.**
2. Order by class should be **last statement in query.**
3. If we use **asc keyword** in order by class then data will be sorted in ascending order.
4. If we use **desc keyword** in order by class then data will be sorted in descending order.
5. If we **won’t provide asc/desc** keyword in order by class the by default data will be sorted in **ascending order.**

**Syntax:**

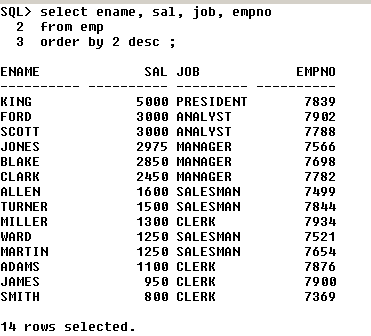
Select column\_list

From table\_name

Order by column\_name asc/desc;

**Ex:**

1. **Arrange ename, sal, job, empno and sort by descending order of salary**

****

In the above query we have – **order by 2** – thus it arranges only the 2nd column ‘salary’ in the descending order.

Thus to arrange the specific columns in order – we must have to specify the column number.

SQL | ALL and ANY

ALL & ANY are logical operators in SQL. They return boolean value as a result.

ALL

ALL operator is used to select all tuples of SELECT STATEMENT. It is also used to compare a value to every value in another value set or result from a subquery.

The ALL operator returns TRUE iff all of the subqueries values meet the condition. The ALL must be preceded by comparison operators and evaluates true if all of the subqueries values meet the condition.

ALL is used with SELECT, WHERE, HAVING statement.

ALL with SELECT Statement:

Syntax:

SELECT ALL field\_name

FROM table\_name

WHERE condition(s);

ALL with WHERE or HAVING Statement:

Syntax:

SELECT column\_name(s)

FROM table\_name

WHERE column\_name comparison\_operator ALL

(SELECT column\_name

FROM table\_name

WHERE condition(s));

Example:

Consider the following Products Table and OrderDetails Table,

Products Table

OrderDetails Table

Queries

Find the name of the all the product.

SELECT ALL ProductName

FROM Products

WHERE TRUE;

Output:

Find the name of the product if all the records in the OrderDetails has Quantity either equal to 6 or 2.

SELECT ProductName

FROM Products

WHERE ProductID = ALL (SELECT ProductId

FROM OrderDetails

WHERE Quantity = 6 OR Quantity = 2);

Output:

Find the OrderID whose maximum Quantity among all product of that OrderID is greater than average quantity of all OrderID.

SELECT OrderID

FROM OrderDetails

GROUP BY OrderID

HAVING max(Quantity) > ALL (SELECT avg(Quantity)

FROM OrderDetails

GROUP BY OrderID);

Output:

ANY

ANY compares a value to each value in a list or results from a query and evaluates to true if the result of an inner query contains at least one row.

ANY return true if any of the subqueries values meet the condition.

ANY must be preceded by comparison operators.

Syntax:

SELECT column\_name(s)

FROM table\_name

WHERE column\_name comparison\_operator ANY

(SELECT column\_name

FROM table\_name

WHERE condition(s));

Queries

Find the Distinct CategoryID of the products which have any record in OrderDetails Table.

SELECT DISTINCT CategoryID

FROM Products

WHERE ProductID = ANY (SELECT ProductID

FROM OrderDetails);

Output:

Finds any records in the OrderDetails table that Quantity = 9.

SELECT ProductName

FROM Products

WHERE ProductID = ANY (SELECT ProductID

FROM OrderDetails

WHERE Quantity = 9);

This article is contributed by Anuj Chauhan. If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.