OPERATORS IN SQLSERVER: (18-10-2024)
=======================

- to perform some operation on the given operand values.
- sqlserver supports the following operators are,

i) Assignment operator	=>	=	
ii) Arithmetic operators	=>	+ , - , * , /	
iii) Relational operators	=>	< , > , <= , >= , != (or)<>	
iv) Logical operators	=>	AND,OR,NOT	
v) Set operators	=>	UNION,UNION ALL,INTERSECT,EXCEPT.	
vi) Special operators	=>	(+ve)	(-ve)
		====	====
		IN	NOT IN
		BETWEEN	NOT BETWEEN
		IS NULL	IS NOT NULL
		LIKE	NOT LIKE

i) Assignment operator:
====================

- to assign a value to a variable / to an attribute.

syntax:
=======

<column name> <assignment operator> <value>

Ex:
```
UPDATE EMP SET SAL=34000;
UPDATE EMP SET LOC='HYD' WHERE DEPTNO=10;
```

ii) Arithmetic operators:
====================

- to perform addition,subtraction,multiple and division.

syntax:
======

<column name> <arithmetic operator> <value>

Ex:
waq to display all employees salaries after adding 2500/-?
```
SELECT SAL AS OLD_SALARY,SAL+2500 AS NEW_SALARY FROM EMP;
```

Ex:
waq to display EMPNO,ENAME,JOB,BASIC SALARY and ANNUAL SALARY of the employees who are working as a "MANAGER"?
```
SELECT EMPNO,ENAME,JOB,SAL AS BASIC_SALARY,SAL*12 AS ANNUAL_SALARY
FROM EMP WHERE JOB='MANAGER';
```

Ex:
waq to display all employees salaries after increment of 10%?
SELECT ENAME,SAL AS BEFORE_INCREMENT,SAL+SAL*10/100 AS AFTER_INCREMENT FROM EMP;

Ex:
waq to display ENAME,DEPTNO,BASIC_SALARY,INCREMENT OF 5% AMOUNT and TOTAL SALARY
of the employees who are working under deptno 20?
SELECT ENAME,DEPTNO,SAL AS BASIC_SALARY,SAL*0.05 AS INCREMENT_AMOUNT,
SAL+SAL*0.05 AS TOTAL_SALARY FROM EMP WHERE DEPTNO=20;

Ex:
waq to display EMPNO,ENAME,JOB,BASIC SALARY, 10% of HRA,20% of DA,5% of PF and also
findout GROSS SALARY of the employees who are working as "SALESMAN"?
SELECT EMPNO,ENAME,JOB,SAL AS BASIC_SALARY,SAL*0.1 AS HRA,
SAL*0.2 AS DA,SAL*0.05 AS PF,SAL+SAL*0.1+SAL*0.2+SAL*0.05 AS GROSS_SALARY
FROM EMP WHERE JOB='SALESMAN';

Ex:
waq to display all employees salaries after decrement of 5%?
SELECT ENAME,SAL AS BEFORE_DECREMENT,SAL-SAL*5/100 AS AFTER_DECREMENT FROM EMP;

19-10-2024:
==========
iii) Relational operators:
=====================

- comparing a specific column values with user defined condition in the query.

syntax:
======

where <column name> <relational operators> <value>;

EX:
waq to display list of employees who are joined before 1981?
SELECT * FROM EMP WHERE HIREDATE<'1981-01-01';

EX:
waq to display list of employees who are joined after 1981?

SELECT * FROM EMP WHERE HIREDATE>'1981-12-31';

iv) Logical operators:
==================

- to check more than one condition in the query.
- AND,OR,NOT.

AND operator:
=============

- it return a value when both conditions are true in the query.

Cond1	Cond2	
T	T	===> T
T	F	===> F
F	T	===> F
F	F	===> F

syntax:
=======

where <condition1> and <condition2>

EX:
waq to display employees details who are working as a "CLERK" and whose name is "YUVIN"?
SELECT * FROM EMP WHERE JOB='CLERK' AND ENAME='YUVIN';

OR operator:
=============

- it return a value if any one condition is true in the query.

Cond1	Cond2	
T	T	===> T
T	F	===> T
F	T	===> T
F	F	===> F

syntax:
=======

where <condition1> or <condition2>

EX:
waq to display employees who are working as a "MANAGER","ANALYST","PRESIDENT"?

SELECT * FROM EMP WHERE JOB='MANAGER' OR JOB='ANALYST' OR JOB='PRESIDENT';

NOT operator:
============
- it return all values except the given conditional values in the query.

syntax:
======
where not <condition1> and not <condition2>

Ex:
waq to display employees whose EMPNO is not 7566,7788?
SELECT * FROM EMP WHERE NOT EMPNO=7566 AND NOT EMPNO=7788;

v) Set operators:
===============
- are used to combined the results of two select statements and show as a single set of values.
syntax:
======
<select query1> <set operator> <select query2>;

EX:
A={10,20,30}   B={30,40,50}

UNION:
======
- to combined two sets values without duplicates.
A U B = {10,20,30,40,50}

UNION ALL:
==========
- to combined two sets values with duplicates.
A UL B = { 10,20,30,30,40,50}

INTERSECT:
===========
- it return common values from both sets.
A I B = { 30 }

EXCEPT:
========

- it return un-common values from the left set but not the right set.
  A - B = { 10,20 }
  B - A = { 40,50 }

DEMO_TABLES:
============
CREATE TABLE EMP_HYD(EID INT,ENAME VARCHAR(20),SALARY MONEY)
INSERT INTO EMP_HYD
VALUES(1021,'SMITH',85000),(1022,'WARD',46000),(1023,'JONES',73000)

CREATE TABLE EMP_MUMBAI(EID INT,ENAME VARCHAR(20),SALARY MONEY)
INSERT INTO EMP_MUMBAI VALUES(1021,'SMITH',85000),(1024,'MILLER',55000)

SELECT * FROM EMP_HYD
SELECT * FROM EMP_MUMBAI

EX:
waq to display employees who are working in hyderabad but not in mumbai branch?
SELECT * FROM EMP_HYD EXCEPT SELECT * FROM EMP_MUMBAI;

EX:
waq to display employees who are working in both branches?
SELECT * FROM EMP_HYD INTERSECT SELECT * FROM EMP_MUMBAI;

EX:
waq to display all employees who are working in the organization?
SELECT * FROM EMP_HYD UNION ALL SELECT * FROM EMP_MUMBAI;(with duplicate rows)
SELECT * FROM EMP_HYD UNION SELECT * FROM EMP_MUMBAI;(without duplicate rows)

Basic rules:
==========
1. no.of columns and order of columns should be same in both select queries.
2. their corresponding datatypes are also match.

21-10-2024:
===========
vi) Special operators:
==================
IN operator:
==========
- comparing the list of values based on a single condition.

syntax:

=======

    where <column name> IN(<list of values>);
    where <column name> NOT IN(<list of values>);

Ex:
waq to display employees whose EMPNO is 7369,7566,7788,7900?
SELECT * FROM EMP WHERE EMPNO IN(7369,7566,7788,7900);

EX:
waq to display the list of employees who are not working as a "CLERK","SALEMAN","MANAGER"?
SELECT * FROM EMP WHERE JOB NOT IN('CLERK','SALESMAN','MANAGER');

BETWEEN operator:
=================

    - comparing the particular range value.

syntax:
=======

    where <column name> between <low value> and <high value>;
    where <column name> not between <low value> and <high value>;

EX:
waq to display employees who are joined in 1982?
SELECT * FROM EMP WHERE HIREDATE BETWEEN '1982-01-01' AND '1982-12-31';

EX:
waq to display employees who are not joined in 1982?
SELECT * FROM EMP WHERE HIREDATE NOT BETWEEN '1982-01-01' AND '1982-12-31';

IS NULL operator:
===============

    - comparing NULLs in a table.

syntax:
=======

    where <column name> is null;
    where <column name> is not null;

EX:
waq to fetch employees whose commission is undefined / unknown / null?
SELECT * FROM EMP WHERE COMM IS NULL;

EX:
waq to fetch employees whose commission is defined / known / not null?
SELECT * FROM EMP WHERE COMM IS NOT NULL;

What is NULL?
============

- it is an empty / undefined value / unknown value in database.
- NULL != 0 & NULL != space.
- when we perform arithematic operations with NULL then it again return NULL only.

Ex: If x=500;

i) x+null ===> 500+null ===> null
ii) x-null ===> 500-null ===> null
iii) x*null ==> 500*null ===> null
iv) x/null ===> 500/null ====> null

Ex:
waq to display EMPNO,ENAME,SALARY,COMM and SAL+COMM from emp table whose employee name is "YUVIN"?
SELECT EMPNO,ENAME,SAL,COMM,SAL+COMM AS TOTAL_AMOUNT FROM EMP WHERE ENAME='YUVIN';

OUTPUT:
========
```
EMPNO    ENAME     SAL     COMM TOTAL_AMOUNT
======   ======    ======       ===== =============
7369  YUVIN 5000.00        NULL   NULL
```

- In the above example then employee "YUVIN" salary is 5000 and there is no commission so that salary+commission is 5000 only but it return NULL.
- To overcome the above problem SQLSERVER will provide a pre-defined function is known as "ISNULL()".

What is ISNULL():
==============

- to replace a user-defined value inplace of NULL.

syntax:
======

ISNULL(exp1,exp2)

- If exp1 is NULL then it return exp2 value(user defined value)
- If exp1 is NOT NULL then it return exp1 value only.

EX:
SELECT ISNULL(NULL,0) AS RESULT;-----------------> 0
SELECT ISNULL(NULL,100) AS RESULT;-------------> 100
SELECT ISNULL(500,0) AS RESULT;---------------------> 500

SELECT ISNULL(0,700) AS RESULT;---------------------> 0

Solution:
=======
SELECT EMPNO,ENAME,SAL,COMM,SAL+ISNULL(COMM,0) AS TOTAL_AMOUNT FROM EMP
WHERE ENAME='YUVIN';

OUTPUT:
========
```
EMPNO         ENAME         SAL     COMM TOTAL_AMOUNT
======        ======        ======         ===== =============
7369    YUVIN 5000.00          NULL   5000
```

22-10-2024:
===========
LIKE operator:
=============
- comparing specific character pattern wise.
- when we use "LIKE" operator we must use the following wildcard operators are,
  - i) % - it represent the remaining group of characters after selected character/(s).
  - ii) _ - counting a single character in the expression.
  - iii) [ ] - it represent set of characters.

syntax:
======
where <column name> '<wildcard operator> <character pattern> <wildcrad operator>';

Ex:
waq to fetch employees whose name starts with "S" character?
SELECT * FROM EMP WHERE ENAME LIKE 'S%';

EX:
to fetch employees whose name starts with "M" and ends with "N" character?
SELECT * FROM EMP WHERE ENAME LIKE 'M%N';

EX:
to fetch employees whose name is having "I" character?
SELECT * FROM EMP WHERE ENAME LIKE '%I%'

EX:
to fetch employees whose name is having four characters?
SELECT * FROM EMP WHERE ENAME LIKE '____';

EX:
to fetch employees whose name is having the second character is " O " ?
SELECT * FROM EMP WHERE ENAME LIKE '_O%';

EX:
to fetch employees whose EMPNO starts with 7 and ends with 8?
SELECT * FROM EMP WHERE EMPNO LIKE '7%8';

EX:
to fetch list of employees who are joined 1981?
SELECT * FROM EMP WHERE HIREDATE LIKE '1981%';
(OR)
SELECT * FROM EMP WHERE HIREDATE LIKE '%1981%';

EX:
to fetch list of employees who are joined in the month of DECEMBER?
SELECT * FROM EMP WHERE HIREDATE LIKE '%-12-%';

EX:
to fetch list of employees who are joined in the month of DECEMBER in 1980?
SELECT * FROM EMP WHERE HIREDATE LIKE '1980-12-%'
(OR)
SELECT * FROM EMP WHERE HIREDATE LIKE '%-12-%' AND HIREDATE LIKE '1980%'

EX:
to fetch list of employees who are joined in the month of JUNE,DECEMBER?
SELECT * FROM EMP WHERE HIREDATE LIKE '%-06-%' OR HIREDATE LIKE '%-12-%';

Ex:
to fetch employees whose name starts with "A","J","K" characters?
SELECT * FROM EMP WHERE ENAME LIKE '[A,J,K]%';

EX:
to fetch employees whose name starts with A-Z characters?
SELECT * FROM EMP WHERE ENAME LIKE '[A-Z]%';

LIKE operator with special characters:
================================
DEMO_TABLE:

============
SELECT * FROM CUSTOMER;

```
CID     CNAME               CMBNO
===     ========     ==========
1       WAR@NER     9874563214
2       _JAMES              8523697415
3       MILL%ER             7412589636
4       JONE%S              6985231478
5       TUR#NER             8741253695
6       SCO_TT              9632587415
```

EX:
waq to fetch customer details whose name is having "#" symbol?
SELECT * FROM CUSTOMER WHERE CNAME LIKE '%#%';

EX:
waq to fetch customer details whose name is having "@" symbol?
SELECT * FROM CUSTOMER WHERE CNAME LIKE '%@%';

EX:
waq to fetch customer details whose name is having "_" symbol?
SELECT * FROM CUSTOMER WHERE CNAME LIKE '%_%'; ----------------------> it return wrong result

EX:
waq to fetch customer details whose name is having "%" symbol?
SELECT * FROM CUSTOMER WHERE CNAME LIKE '%%%'; ---------------------> it return wrong result

NOTE:
=====
- whenever we are fetching data from a table based on " _ , %" then sqlserver return wrong result because these "_,%" symbols are treating as wildcard operator but not as a special characters.
- To overcome the above problem we must use a pre-defined keyword is " ESCAPE '\' ".

Solution:
SELECT * FROM CUSTOMER WHERE CNAME LIKE '%_%'ESCAPE'\';
SELECT * FROM CUSTOMER WHERE CNAME LIKE '%\%%'ESCAPE'\';

Ex:
waq to fetch employees whose name not starts with "S" character?

```
SELECT * FROM EMP WHERE ENAME NOT LIKE 'S%';
=========================================================================
======
```