

The SQL plus (+) operator is used to add two or more expressions or numbers. SQL plus (+) operator is nothing but add (+) operator.

Example: ①

```
Select 15+10 from dual;
```

o/p:- 25

Example: ②

Sample table: Student

Student-name	marks1	marks2	marks3	marks4
Ram	90	91	92	93
Ravi	80	81	82	83
Raju	70	71	72	73
Revanth	91	92	93	94

The following can be used to get the total marks of every student.

```
Select student-name, marks1, marks2, marks3, marks4, (marks1 + marks2 + marks3 + marks4) as total  
from student;
```

Output:-

Student-name	marks1	marks2	marks3	marks4	total
Ram	90	91	92	93	366
Ravi	80	81	82	83	326
Raju	70	71	72	73	286
Revent	91	92	93	94	370

4 rows selected.

The SQL Minus (-) operator is used to subtract one expression or number from another expression or number. SQL Minus (-) operator is nothing but subtraction (-) operator.

Example ①

```
select 15-10 from dual;
```

Output: 5

Example ②

Sample table: Bank_account

Account-holder	Total-Amount	withdraw-amount
Ram	10,000	5,000
Ravi	15,000	10,000
Raju	20,000	15,000
Rajesh	25,000	20,000

The following command can be used to get the remaining amount of each account holder.

```
select Account-holder, Total-Amount, withdraw-Amount, (TotalAmount - withdraw-amount) as remaining-amount from Bank-account;
```

Output: -

Account-holder	Total-Amount	withdraw-Amount	remaining-amount
Ram	10,000	5,000	5,000
Ravi	15,000	10,000	5,000
Raju	20,000	15,000	5,000
Rajesh	25,000	20,000	5,000

The SQL multiply (*) operator is used to multiply two or more expressions or numbers.

Example ①

```
select 15*10 from dual;
```

o/p:- 150

Example ②

Sample-table : ~~Business-profit~~ student-percentage.

<u>student-name</u>	<u>percentage</u>	<u>Totalmarks</u>
Ravi	91.6	600
Ram	92.6	600
Raju	93.6	600
Rajesh	94.6	600

To get the total marks attained by every student, we are using the following command.

```
select student-name, percentage, totalmarks, (percentage * totalmarks) as attained
marks
from student-percentage;
```

output:-

<u>student-name</u>	<u>percentage</u>	<u>Total marks</u>	<u>attained - marks, = $\frac{\text{Percent} \times \text{Total}}{100}$</u>
Ravi	91.6	600	550
Ram	92.6	600	490 556
Raju	93.6	600	562
Rajesh	94.6	600	568

4 rows selected.

4. SQL divide (/) operator :-

The SQL divide (/) operator is used to divide one expression or number by another expression or number.

Example ①

```
select 15/10 from dual;
```

o/p :- 1.5

Example ②

Sample table: student

Student-name	marks1	marks2	marks3	total
Ram	90	91	92	273
Ravi	80	81	82	243
Raju	70	71	72	213
Rajesh	90	92	93	275

To get the average marks we are using the following command.

```
select student-name, marks1, marks2, marks3, total, (total/3) as average  
from data student;
```

Output :-

student-name	marks1	marks2	marks3	total	average
Ram	90	91	92	273	91
Ravi	80	81	82	243	81
Raju	70	71	72	213	71
Rajesh	90	92	93	275	91.666

4 rows selected.

Logical 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. There are three logical operators. They are -

1. OR
2. AND
3. NOT

Syntax:-
 select column from table-name where
 <condition1> [logical operator] <condition2>;

Logical operator	Description
OR	for the row to be selected at least one of conditions must be true.
AND	for a row to be selected all the specified conditions must be true.
NOT	for a row to be selected the specified condition must be false.

1. "OR" logical Operator:-

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

Example

Sample table: student

<u>student_name</u>	<u>Percentage</u>	<u>Result</u>
Ram	92.2	Pass
Ravi	82.6	Pass
Raju	73.6	Fail
Rajesh	66.6	Pass
Revanth	70.7	Fail

The SQL divide (/) operator is used to divide one expression or number by another expression or number.

Example ①

```
select 15/10 from dual;
```

OP :- 1.5

Example ②

Sample-table: student

student-name	marks1	marks2	marks3	total
Ram	90	91	92	273
Ravi	80	81	82	243
Raju	70	71	72	213
Rajesh	90	92	93	275

To get the average marks we are using the following command.

```
select student-name, marks1, marks2, marks3, total, (total/3) as average  
from data student;
```

output :-

student-name	marks1	marks2	marks3	total	average
Ram	90	91	92	273	91
Ravi	80	81	82	243	81
Raju	70	71	72	213	71
Rajesh	90	92	93	275	91.666

4 rows selected.

result is pass, the following command is used.

```
Select student_name
from student
where percentage > 73 or Result = 'pass';
```

Output:-

student_name

Ram

Ravi

Raju

Rajsh

4 rows selected.

The following table describes how logical "OR" operator selects a row.

column1 satisfied?	column2 satisfied?	Row selected?
yes	Yes	yes
yes	NO	yes
NO	yes	yes
NO	NO	NO

2. "AND" logical Operator:-

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

Example:-

sample table: student.

Student Name	Percentage	Result
Ram	92.2	Pass
Ravi	82.6	Pass
Raju	73.6	Fail
Rajsh	66.6	Pass
Revanth	70.6	Fail

To get the student-name whose percentage is greater than 73 and whose result is pass, the following command is used.

```

select student-name
from student
where percentage > 73 and result = 'pass';

```

Output:

student-name

Ram

Ravi

2 rows selected.

The following table describes how logical "AND" operator selects a row.

Column1 Satisfied?	Column2 Satisfied?	Row selected
Yes	Yes	Yes
Yes	No	No
No	Yes	No
No	No	No

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 row is not returned.

Example:

Sample table: student

<u>student-name</u>	<u>percentage</u>	<u>Result</u>
Ram	92.2	pass
Ravi	82.6	pass
Raju	73.6	fail
Rajesh	66.6	pass
Revant	70.6	fail

To get the student-name who did not fail, the following command is used.

```

select student-name
from student
where not Result = 'fail';
    
```

Output:-

student-name

Ram

Ravi

Rajesh.

3 rows selected.

The following table describes how logical "NOT" operator selects a row.

column1 satisfied?	NOT Column1 satisfied?	Row selected
Yes	NO	NO
NO	yes	Yes.

Date and Time Functions:- These are the functions that take values that are of

datatype DATE as input and return values of datatype DATE, except for Months-between function which returns a number.

1. Sysdate():-

Sysdate() is one of the date-function which is used to display the systemdate.

The function sysdate() returns a 7 byte binary data element whose bytes represents

- (i) century
- (ii) year
- (iii) month
- (iv) day
- (v) hour
- (vi) minute
- (vii) second.

ie select sysdate from dual;

sysdate

October, 2 2017 13:28:42+0000

Note:-

The select statement must have a from clause. That's why oracle has a dual table. It is a special table with a column called DUMMY that has a value of 'X' used in selecting a pseudo column such as SYS DATE.

Examples:-

① select sysdate from dual;

SYS DATE

26-09-17