SQL developer tool to connect oracle.

SQL is not case sensitive #we can write query in both small or capital letters

We can call query as sql statement

A query should have two important keywords

1.SELECT 2. FROM

SELECT/FROM/WHERE #these are clauses

Syntax:

Select column\_name from table\_name;

\*astersick is a oracle operator which denotes all\_columns;

Eg;

To take a single column from a table

Use: select names from Employee; here names is one of the column in Employee

table

To show more than one column in a table use , to divide

Use: select names,salary from Employee;

Syntax:

To select a particular row in a table use where with condition

Select column\_name from table\_name where condition;

Use: select names,salary from Employee where salary>10000;

SQL sometimes act a case sensitive to fetch the data

Eg:

Use has a table named Employee with two columns named names and salary

Here names are characters which may be capital letter or small letter

But the values cannot have any case sensitive case.

You have to fetch data from Naveen which is in the table

Select \* from Employee where names=Naveen;

Here Naveen first letter is capital so it will fetch correctly

When you give naveen instead of Naveen it will not fetch data because sql act as case sensitive while accessing the data.

Need to mention in a single quote for case sensitive

Eg: select \* from employee where names=’Naveen’;

CHARACTER/DATE/SPCL CHAR – these consider under case sensitive

OPERATOR:

We can write a multiple condition to a query for that we need logical operator.

Logical operator:

1.AND Operator- it must satisfy all the given conditions

Eg:

Select names,salary from Employee where first\_name=’naveen’ and salary>1100;

Here AND operator acts like both the conditions should satisfy

2.OR Operator- neither any condition given

Select names,salary from Employee where first\_name=’naveen’ or salary=300;

RELATIONAL OPERATOR:

= Equal too

> greater than

<smaller than

>=greater than equal too

<=smaller than equal too

!= (or) <> not equal too

in #operator which selects multiple data in table

not in #operator which deselects multiple data in table

between #show the range or limit

not between #does not show the range or limit

like #used to display first or last letter to fetch data

not like #not used to display first or last letter to fetch data

% shows the remaining letters

\_ #underscore matches only one letter

= equal too which matches only one value

is null #display the value which is null # undefined/unassigned/unknown value

is not null #display the value which is not null # undefined/unassigned/unknown value

Eg: we cannot give like this

Select \* from Employee where names=’naveen’,’praveen’,’satvik’;

Here = is relational operator which matches only one value.

We can use

Select \* from Employee where names in (’naveen’,’praveen’,’satvik’);

To show the data that is without the names given, we can use

Select \* from Employee where names not in (’naveen’,’praveen’,’satvik’);

Eg for between/not between:

Select \* from Employee where salary between 1000 to 2000;

This displays the data from salaried names with salary between 1000 to 2000.

Select \* from Employee where salary not between 1000 to 2000;

This displays the data from salaries names without salary between 1000 to 2000.

Like:

Select \* from Employee where first\_name like ‘a%’;

This would display the names that are starting with character ‘a’

% shows the remaining letters

To show the data of last letters

Use this:

Select \* from Employee where first\_name like ‘%n’;

This will display the names that are ending with character ‘n’

To display the exact third word starting letter

Use this:

Select \* from Employee where first\_name like ‘\_\_n%’; #underscore shows the position of the character.

Null: #display the value which is null # undefined/unassigned/unknown value

null<>null

null<>-

null<>0

null<>blank

to find a value which is null we use ‘is null’

Use this:

Select first\_name,details from Employee where details is null;

CONCATENATED OPERATOR:

Used to join two columns

For Eg:

You need to joint first\_name and last\_name in a single column

Use this:

Select first\_name||last\_name from Employee;

#it will merge both the columns but no space in between

Use this:

Select first\_name||’ ’||last\_name from Employee;

Single quoted in between shows the space so it will give the space

What and all you give in between the single quoted it will display

ALIAS ‘as’

#to change the column name we use as

Use this:

Select first\_name,last\_name as names from Employee;

Here names will be displayes instead of first\_name and last\_name.