

NOTE: SQL language is not case sensitive.



* SELECT

Syntax:

Select < column1, col2, ..>

From <table name>

Eg.:

i) select Job
From emp;

Job
Manager
Clerk
Manager

ii) Select Job, ename:
from emp;

Job	ename
Manager	King

iii) select * from emp;

empno	ename	Job	MGR
7369	KING	Manager	Y

iv) Select distinct Job
from emp;

Job
Manager
Clerk
Salesman

NOTE: When it comes to comparing data in the db, the data should be exact. eg: if 'Manager' is stored in db, then 'manager' will not do.

* Where

Select * from emp
where job = 'Manager'

* AND

Select * from emp
where job = 'Salesman'
AND sal = 1600
AND comm = 300

Empro	Ename	Job	sal	Comm
7499	Allen	Salesman	1600	300

NOTE: (i) Select, From, Where/And, is the basis of the SQL queries.

(ii) eg: Select * from emp,
where Job = 'Artist'

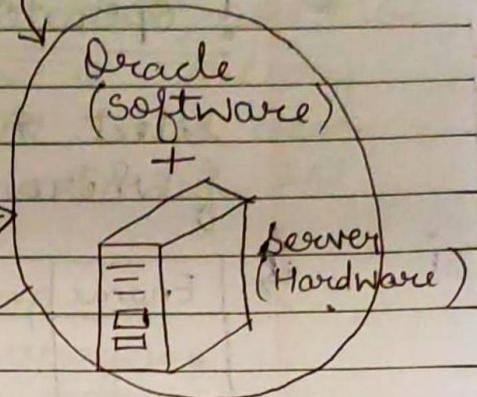
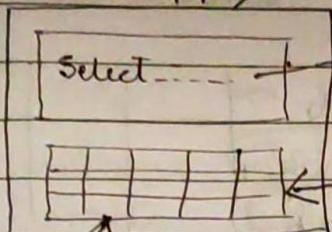
⇒ No data found

(This is not an error in syntax)



Oracle Database

Apex tool
(Webapp)



Select ... ③
 From Emp ①
 where sal = 3000 ②

emp ①

A	C

A	Z	C	Z

③

Execution of a statement,

①, ②, ③

* '!' operator (Not operator)

select * from emp
 where job != 'SALESMAN'

Ename	Job
—	Manager
—	Clerk

* Operators:

< → Less than	= → equal to
> → More than	!= → not
<= → Less than equal to	equal to
>= → More than equal to	

- Q) Write a query that returns those employees who have a commission greater than their salary

→ Select * from emp
 from comm > sal

Q) get all employees that are not managers & have a salary greater than 2500 and also work in department no. 20

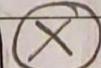


Select * from emp
where Job != 'Manager'
AND Sal > 2500
AND DeptNo = 20

Q) get all employees that are either Clerk or Manager.



Select * from emp
where job = 'Clerk'
and job = 'Manager'



NOTE : The above query will not give any result because an employee cannot be both Clerk as well as Manager ('AND' operator is making compulsion to satisfy both statements simultaneously)

Select * from emp
where job = 'Clerk'
OR job = 'Manager'



'OR'

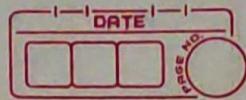
- Q) The name of those employees that are not managers nor salesmen & have a salary greater than or equal to 2000

Ename

→ select * from emp
 where Job != 'Manager'
(AND) Job != 'SALESMAN'
 AND SAL >= 2000

- Q) Write a query to return the names & hire dates of those employees that work in Dallas or Chicago.

→ select ename, hiredate
 from emp
 where deptno = 20
 OR deptno = 30



ALTERNATIVE: (2nd method)

```
select ename, hiredate  
from emp  
where deptno IN (20,30);
```

* 'IN' clause :

```
select * from emp  
where ename = 'Allen'  
or ename = 'Ward'  
or ename = 'Jones'
```

↑
is same as

```
select * from emp  
where ename IN ('Allen', 'Ward',  
                 'Jones')
```

* 'NOT IN' clause :

```
select * from emp  
where ename NOT IN ('King', 'Blake')
```

⇒ The above query will give
all the result except ename
of King and Blake.

* BETWEEN:

Syntax:

select * from tablename
where column-name between

(Val1) and (Val2);

NOTE: (Val1 and Val2 are both inclusive)

Eg:

DATE \Rightarrow i) select * from emp
where hiredate between '05/01/1981'
and '12/09/1982';

Number \Rightarrow ii) select * from emp
where sal between 1000
and 2000;

Character \Rightarrow iii) select * from emp
where ename between 'Allen' and
'Jones';

* NOT BETWEEN:

i) select * from emp
where sal NOT BETWEEN
950 and 1600;



* NULL:

This will give all the records in which the particular column is empty (Not filled).

Eg:

Ename	deptno	Sal	Comm
Allen	20	1000	-

i) Select * from emp
where COMM is NULL;

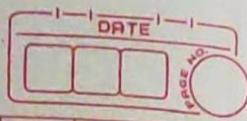
→ NOTE: 'Zero(0)' is considered as a filled data.

* NOT NULL:

i) select * from emp
where COMM is NOT NULL;

→

Ename	deptno	Sal	Comm
Ward	20	1600	500
Martin	30	3000	1400



Q) write a query that returns those employees that don't make any commission and have a sal. greater than 1100 but less than 5000. Exclude those employees that have a salary equal to 3000.

→

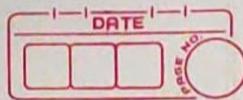
Q) Return those employees that are salesman & that make either 300 dollars in commission or greater than 1000 dollars in commission .

→

select *
from emp

where job = 'Salesman'

AND (COMM = 3000 OR COMM > 1000)



* 'Like' operator :

To search for records that don't match exactly but kind of match (similar match)

Eg : i) select * from emp
where job like 'S%'

Job
Salesman
Salesman

ii) select Job from emp
where job like '% ger'

Job
Manager
Manager

iii) select Job from emp
where Job like '% MANAGER %';

Job
Manager
Manager

* Aliases :

It is used to rename column names in the result.

e.g.: (i) Without Aliases:

```
select Ename, sal, comm
from emp;
```

Ename	Sal	Comm
" " "	" "	" "
" " "	" "	" "

(ii) With Aliases:

```
select Ename AS Employee-Name,
       sal AS "Salary", comm AS Commission
  from emp;
```

Employee-Name	Salary	Commission

→ NOTE: (i) To use space in alias name, we use "double quote".
 Eg: Ename as "Employee Name"



* Concatenation : (||)

Combining of two attributes together into a sentence.

Eg: i) select 'Hello my name is'
 || ename from emp;

'Hello my name is' || ename

hello my name is Blake

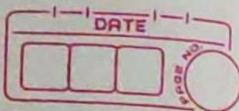
Clarke

ii) & Using Aliases :

Select 'Hello my name is' || ename
As "Concatenated value" from emp;

Concatenated Value

Hello my name is Blake
 Clarke



(iii) select 'ename' || 'makes \$' ||
sal || 'per month' AS
"employee income" from Emp;

Employee income

Blake makes \$ 2450 per month

* Ordering By :

The data is being sorted
ascending from smallest
to largest by default.

Eg: i) Select sal from emp
Order by Sal ;

Sal

900

1250

1400

↓ Ascending

ii) Select sal from emp
Order by Sal DESC;



Sal
5000
3000 ↓
3000 Descending

iii) select DeptNo, Sal, Ename
from emp
order by Dept No, Sal ;

	DeptNo	Sal	Ename
ASC	1	500 Asc	Nitin
	1	725 ↓	Raj
	2	600	Dylan
	3	980 Asc	Rahul
	3	1000 ↓	Dimple

iv) select DeptNo, Sal , Ename
from Emp
order by Dept No DSC , Sal DSC ;

	DeptNo	Sal	Ename
DSC	3	1000 DSC	Dimple
	3	980 ↓	Rahul
	2	600	Dylan
	1	725 DSC	Raj
	1	500 ↓	Nitin



* Single Row Functions (SRFs)

(1) concat

select concat ('my name is',
ename) from emp;

concat ('My name is', ename)

my name is King
" " Clark

select concat ('Hello', 'ename')
as sentence from emp

sentence

Hello King
" " "

(2) upper

select upper ('Hello') from emp

Upper ('Hello')

HELLO

HELLO
" " "



NOTE select 'hello' from dept;

'Hello'

hello

hello
" "

NOTE: There is a special table in oracle called Dual Table.

This table is used to check functionality of a function.

select upper ('hello')
from Dual;

Upper ('Hello')

HELLO

select * from Dual;

Dummy

X

select 'pizza' as Food,
'fanta' as Drink from dual

Food

pizza

Drink

fanta

NOTE: A function can be used
in another function.

Eg: select concat(lower(ename),
' is the name')
from emp;

Q) jones IS THE NAME and their job
is: MANAGER

scott IS THE NAME and
their job is: ANALYST

→ select concat(concat(lower(ename),
upper(' is the name')),
concat(' and their job is: ',
Job)) as function call
from emp;



- * Using Functions in WHERE and character based SRFs :

NOTE :

select * from emp
where Job = lower ('MANAGER');

→ NO DATA FOUND

(Because all the data in the employee table are in Capital Letters ie 'MANAGER' and not 'manager')

- * INITCAP Function

Capitalizes the initial letter of every word in the string.

EXAMPLE i) select initcap ('hello there')
as sentence
from ems dual;

Sentence

Hello There

Length Function

This function returns the number of characters in a character sequence (string).

- i) select length ('hello there') from emp; dual;

Length ('hellothere')

- ii) select ename, length (ename) from emp;

Ename	length (Ename)
-------	----------------

Clark	5
Nitin	5
Raj	3

- iii) select ename, length (ename) as length from emp where length (ename) = 5 ;

Ename	Length
Clark	5
Nitin	5