TABLE: EMP

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| EMPNO | ENAME | JOB | MGR | HIREDATE | SAL | COMM | DEPTNO |
| 7369 | SMITH | CLERK | 7902 | 17-DEC-80 | 800 |  | 20 |
| 7499 | ALLEN | SALESMAN | 7698 | 20-FEB-81 | 1600 | 300 | 30 |
| 7521 | WARD | SALESMAN | 7698 | 22-FEB-81 | 1250 | 500 | 30 |
| 7566 | JONES | MANAGER | 7839 | 02-APR-81 | 2975 |  | 20 |
| 7654 | MARTIN | SALESMAN | 7698 | 28-SEP-81 | 1250 | 1400 | 30 |
| 7698 | BLAKE | MANAGER | 7839 | 01-MAY-81 | 2850 |  | 30 |
| 7782 | CLARK | MANAGER | 7839 | 09-JUN-81 | 2450 |  | 10 |
| 7788 | SCOTT | ANALYST | 7566 | 19-APR-87 | 3000 |  | 20 |
| 7839 | KING | PRESIDENT |  | 17-NOV-81 | 5000 |  | 10 |
| 7844 | TURNER | SALESMAN | 7698 | 08-SEP-81 | 1500 | 0 | 30 |
| 7876 | ADAMS | CLERK | 7788 | 23-MAY-87 | 1100 |  | 20 |
| 7900 | JAMES | CLERK | 7698 | 03-DEC-81 | 950 |  | 30 |
| 7902 | FORD | ANALYST | 7566 | 03-DEC-81 | 3000 |  | 20 |
| 7934 | MILLER | CLERK | 7782 | 23-JAN-82 | 1300 |  | 10 |

**RELATIONAL OPERATORS:**

**EQUAL (=):** SELECT \* FROM EMP WHERE SAL=1600;

**GREATER THAN (>):** SELECT \* FROM EMP WHERE SAL > 2100;

**LESS THAN (<):** SELECT \* FROM EMP WHERE SAL < 3200;

**GREATER THAN OR EQUAL TO (>=):**  SELECT \* FROM EMP WHERE SAL>= 2000;

**LESS THAN OR EQUAL TO (<=):** SELECT \* FROM EMP WHERE SAL <=3000;

**NOT EQUAL TO (<> OR !):**  SELECT \* FROM EMP WHERE SAL <>2000;

**LOGICAL OPERATORS**

1. AND
2. OR
3. NOT
4. IN
5. BETWEEN
6. LIKE
7. IS

**AND:**

Whenever we use AND operator, rows get selected if both the conditions are satisfied.

Ex: SELECT \* FROM EMP

WHERE ENAME='SMITH' AND JOB='CLERK';

**O/P:** EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO

------ ---------- --------- ---------- --------- ---------- ---------- ----------

7369 SMITH CLERK 7902 17-DEC-80 800 20

EX2: SELECT \* FROM EMP WHERE ENAME='MILLER' AND DEPTNO=10;

**O/P:**

EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO

------ ---------- --------- ---------- --------- ---------- ---------- ----------

7934 MILLER CLERK 7782 23-JAN-82 1300 10

**OR:**

whenever we use OR operator, rows get selected if any one of the conditions is satisfied.

Ex: SELECT \* FROM EMP

WHERE ENAME='SMITH' OR SAL=1600 OR JOB='CLERK' OR DEPTNO=20;

**O/P:**

EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO

----- ---------- --------- ---------- --------- ---------- ---------- ----------

7369 SMITH CLERK 7902 17-DEC-80 800 20

7499 ALLEN SALESMAN 7698 20-FEB-81 1600 300 30

7566 JONES MANAGER 7839 02-APR-81 2975 20

7788 SCOTT ANALYST 7566 19-APR-87 3000 20

\*\* **Write a query to display all the details of the manager who is earning salary greater than 2000**

Ans: SELECT \* FROM EMP WHERE JOB='MANAGER' AND SAL >2000;

EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO

------ ---------- --------- ---------- --------- ---------- ---------- ----------

7566 JONES MANAGER 7839 02-APR-81 2975 20

7698 BLAKE MANAGER 7839 01-MAY-81 2850 30

7782 CLARK MANAGER 7839 09-JUN-81 2450 10

**Write a query to display all the details of the clerk who belongs to the 20th dept.**

Ans: SELECT \* FROM EMP WHERE JOB='CLERK' AND DEPTNO=20;

EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO

------ ---------- --------- ---------- --------- ---------- ---------- ----------

7369 SMITH CLERK 7902 17-DEC-80 800 20

7876 ADAMS CLERK 7788 23-MAY-87 1100 20

**Write a query to display all the emp details who belongs to the 10th and 30th deptNo.**

Ans: SELECT \* FROM EMP WHERE DEPTNO=10 OR DEPTNO=30;

EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO

------ ---------- --------- ---------- --------- ---------- ---------- ----------

7499 ALLEN SALESMAN 7698 20-FEB-81 1600 300 30

7521 WARD SALESMAN 7698 22-FEB-81 1250 500 30

7654 MARTIN SALESMAN 7698 28-SEP-81 1250 1400 30

7698 BLAKE MANAGER 7839 01-MAY-81 2850 30

**Write a query to display all the emp details of Manager and Analyst.**

Ans: SELECT \* FROM EMP WHERE JOB='MANAGER' OR JOB='CLERK';

EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO

------ ---------- --------- ---------- --------- ---------- ---------- ----------

7369 SMITH CLERK 7902 17-DEC-80 800 20

7566 JONES MANAGER 7839 02-APR-81 2975 20

7698 BLAKE MANAGER 7839 01-MAY-81 2850 30

**IN:**

Whenever we use multiple OR operators, then we are going to replace it with a special operator called “IN”.

**Ex:** multiple OR

SELECT \* FROM EMP WHERE DEPTNO=10 OR DEPTNO=20 OR DEPTNO=30 OR DEPTNO=40;

**In**

SELECT \* FROM EMP WHERE DEPTNO IN(10,20,30,40);

**Write a query to display all the emp details of the clerk who belongs to the 10th and 30th department**

Ans: SELECT \* FROM EMP WHERE JOB=’CLERK’ AND DEPTNO IN (10,30);

EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO

----- ---------- --------- ---------- --------- ---------- ---------- ----------

7900 JAMES CLERK 7698 03-DEC-81 950 30

7934 MILLER CLERK 7782 23-JAN-82 1300 10

**Write a query to display all the details of a salesman and manager who belongs to 30th deptNo.**

ANS: SELECT \* FROM EMP WHERE JOB IN(‘SALESMAN’,’MANAGER’) AND DEPTNO=30;

**Write a query to display all the emp details who joined in the year 1981.**

Ans: SELECT \* FROM EMP WHERE HIREDATE >= ’10-JAN-81’ AND HIREDATE <=’31-DEC-81’;

SELECT \* FROM EMP WHERE HIREDATE >= '10-JAN-81' AND HIREDATE <='31-DEC-81';

EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO

------ ---------- --------- ---------- --------- ---------- ---------- ----------

7499 ALLEN SALESMAN 7698 20-FEB-81 1600 300 30

7521 WARD SALESMAN 7698 22-FEB-81 1250 500 30

7566 JONES MANAGER 7839 02-APR-81 2975 20

7654 MARTIN SALESMAN 7698 28-SEP-81 1250 1400 30

**BETWEEN:**

Whenever the values are present In the range, then we are going to replace it with a special operator called “BETWEEN”.

\*\*The results of the BETWEEN operator include begin and end values of the given range.

**EX:** **write a query to display all the employ details who is earning salary 2000 to 3000.**

SELECT \* FROM EMP WHERE SAL BETWEEN 2000 AND 3000;

EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO

----- ---------- --------- ---------- --------- ---------- ---------- ----------

7566 JONES MANAGER 7839 02-APR-81 2975 20

7698 BLAKE MANAGER 7839 01-MAY-81 2850 30

7782 CLARK MANAGER 7839 09-JUN-81 2450 10

7788 SCOTT ANALYST 7566 19-APR-87 3000 20

7902 FORD ANALYST 7566 03-DEC-81 3000 20

**NOT:**

NOT operator in SQL shows those records from the table where the criteria is not met. NOT operator is used with where clause in a SELECT query.

Ex: SELECT \* FROM EMP WHERE SAL NOT BETWEEN 2000 AND 3000;

EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO

----- ---------- --------- ---------- --------- ---------- ---------- ----------

7369 SMITH CLERK 7902 17-DEC-80 800 20

7499 ALLEN SALESMAN 7698 20-FEB-81 1600 300 30

**NULL:**

Null is an empty space(or) blank space.

Null means nothing.

Null will not occupy any space in the memory.

Null is not equal to 0.

\*\*To achieve NULL values, we have to use special operator called “Is”.

\*\*SQL provides the IS operator to check whether a value is NULL.

EX: SELECT \* FROM EMP WHERE COMM IS NULL;

EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO

----- ---------- --------- ---------- --------- ---------- ---------- ----------

7369 SMITH CLERK 7902 17-DEC-80 800 20

7566 JONES MANAGER 7839 02-APR-81 2975 20

7698 BLAKE MANAGER 7839 01-MAY-81 2850 30

7782 CLARK MANAGER 7839 09-JUN-81 2450 10

7788 SCOTT ANALYST 7566 19-APR-87 3000 20

**Write a query to display total salary of each employ by adding salary with commission.**

SELECT ENAME, SAL+COMM FROM EMP;

ENAME SAL+COMM

---------- ----------

SMITH

ALLEN 1900

WARD 1750

JONES

MARTIN 2650

Note: whenever we perform arithmetic operations with NULL, then the result is always NULL.

**LIKE:**

LIKE Operator in SQL displays only those data from the table which matches the pattern specified in the query.

% – It is used for zero or more than one character.

(\_) – It is used for only one character means fixed length.

**EX:** SELECT ENAME FROM EMP WHERE ENAME LIKE ‘S%’;

EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO

----- ---------- --------- ---------- --------- ---------- ---------- ----------

7369 SMITH CLERK 7902 17-DEC-80 800 20

7788 SCOTT ANALYST 7566 19-APR-87 3000 20

**FUNCTIONS IN SQL:**

**SINGLE ROW FUNCTIONS:**

Single row functions will take multiple inputs and gives you the corresponding output for each input.

Single row functions are 1) UPPER ( )

2) LOWER ( )

3) INITCAP ( )

4) CONCAT ( )

5) REPLACE ( )

6) LENGTH ()

7) SUBSTR ( )

**UPPER ():** UPPER function converts a string to upper case.

EX: SELECT UPPER(‘hyderabad’) FROM DUAL;

OP: HYDERABAD

**LOWER():** LOWER function converts a string to lower case.

Ex: SELECT LOWER(‘HYDERABAD’) FROM DUAL; O/P: hyderabad

**INITCAP ();** INITCAP function converts only the initial alphabets of a string to upper case.

EX: SELECT INITCAP(‘ANDHRA’) FROM DUAL; Andhra

\*\* SELECT UPPER(‘nishanth’), LOWER(‘NISHANTH’), INITCAP(‘NISHANTH’) FROM DUAL;

O/P: NISHANTH, Nishanth, Nishanth

**CONCAT ():**

It is similar to concatenation

Joining or merging or adding two or more column values (or) literals is known as concat().

Concat function will take maximum of two arguments.

SYNTAX: select concat(arg1, arg2) from table name;

Where arg1-🡪column/literals

Where arg2-🡪column/literals

EX: SELECT CONCAT(ENAME, JOB) FROM EMP;

CONCAT(ENAME, JOB)

-----------------

SMITHCLERK

ALLENSALESMAN

WARDSALESMAN

JONESMANAGER

MARTINSALESMAN

BLAKEMANAGER

EX2: **Write a query to display in below format SMITH IS EARNING 800**

SQL> SELECT CONCAT(CONCAT(ENAME, 'IS EARNING'),SAL) FROM EMP

CONCAT(CONCAT(ENAME,'ISEARNING'),SAL)

------------------------------------------------------------

SMITHIS EARNING800

ALLENIS EARNING1600

WARDIS EARNING1250

JONESIS EARNING2975

MARTINIS EARNING1250

BLAKEIS EARNING2850

EX3: **Write a query to display in the below format: smith is a clerk joined on 17-dec-80 and his salary is 800.**

SELECT CONCAT (CONCAT(CONCAT(CONCAT(CONCAT(CONCAT(ENAME,'IS A'),JOB),'JOINED ON'),HIREDATE), 'HIS SALARY IS'),SAL) FROM EMP;

SMITHIS ACLERKJOINED ON17-DEC-80HIS SALARY IS800

ALLENIS ASALESMANJOINED ON20-FEB-81HIS SALARY IS1600

WARDIS ASALESMANJOINED ON22-FEB-81HIS SALARY IS1250

**REPLACE ():**

Replace function will replace your old text data to new text data.

* Your old text data can be column name/literals.

SYNTAX: select replace (arg1, arg2, arg3) from table Name;

Arg1-🡪column name/literals

Arg2-🡪which character/literal we are going to replacing

Arg3--🡪replacing character/literals

Ex: SELECT REPLACE (‘JAVA’, ‘J’, ‘B’) FROM DUAL; O/P: BAVA

SELECT REPALCE (‘TESTENGINEER’, ‘TEST’, ‘SOFTWARE’) FROM DUAL; SOFTWARE ENGINEER

SELECT REPLACE (‘TEST’, ‘E’) FROM DUAL; O/P: TST

Note: Replace function will take maximum of 3 arguments and minimum of 2 arguments. If we fail to give the 3rd argument then, it will remove the 2nd argument from the given input.

**EX:** **write a query to display all the emp details whose designation contains man.**

SELECT \* FROM EMP WHERE JOB=REPLACE(JOB,’MAN’);

EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO

----- ---------- --------- ---------- --------- ---------- ---------- ----------

7369 SMITH CLERK 7902 17-DEC-80 800 20

7788 SCOTT ANALYST 7566 19-APR-87 3000 20

7839 KING PRESIDENT 17-NOV-81 5000 10

7876 ADAMS CLERK 7788 23-MAY-87 1100 20

SELECT \* FROM EMP WHERE JOB! =REPLACE(JOB,’MAN’);

EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO

----- ---------- --------- ---------- --------- ---------- ---------- ----------

7499 ALLEN SALESMAN 7698 20-FEB-81 1600 300 30

7521 WARD SALESMAN 7698 22-FEB-81 1250 500 30

7566 JONES MANAGER 7839 02-APR-81 2975 20

7654 MARTIN SALESMAN 7698 28-SEP-81 1250 1400 30

7698 BLAKE MANAGER 7839 01-MAY-81 2850 30

**Ex:** **write a query to display all the emp details whose name contains the letter e.**

SELECT \* FROM EMP WHERE ENAME <> REPLACE(ENAME,’E’);

EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO

------ ---------- --------- ---------- --------- ---------- ---------- ----------

7499 ALLEN SALESMAN 7698 20-FEB-81 1600 300 30

7566 JONES MANAGER 7839 02-APR-81 2975 20

7698 BLAKE MANAGER 7839 01-MAY-81 2850 30

7844 TURNER SALESMAN 7698 08-SEP-81 1500 0 30

7900 JAMES CLERK 7698 03-DEC-81 950 30

7934 MILLER CLERK 7782 23-JAN-82 1300 10

**LENGTH():**

It gives the no of characters present in the given input.

**SYNTAX:** select length(colName/’literal) from table\_name:

**EX:** **write a query to display all the employee details whose name contains exactly 5 characters.**

SELECT \* FROM EMP WHERE LENGTH(ENAME)=5;

EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO

----- ---------- --------- ---------- --------- ---------- ---------- ----------

7369 SMITH CLERK 7902 17-DEC-80 800 20

7499 ALLEN SALESMAN 7698 20-FEB-81 1600 300 30

7566 JONES MANAGER 7839 02-APR-81 2975 20

7698 BLAKE MANAGER 7839 01-MAY-81 2850 30

**EX2:** **write a query to display no of ‘L’s ‘ present in each employe name.**

SQL> SELECT ENAME, LENGTH(ENAME)-LENGTH(REPLACE(ENAME,'L')) FROM EMP;

ENAME LENGTH(ENAME)-LENGTH(REPLACE(ENAME,'L'))

---------- ----------------------------------------

SMITH 0

ALLEN 2

WARD 0

JONES 0

MARTIN 0

**Ex3:** **write a query to display no of 5s present in each employe name.**

SELECT SAL, LENGTH(SAL)-LENGTH(REPLACE(SAL,5)) FROM EMP;

SAL LENGTH(SAL)-LENGTH(REPLACE(SAL,5))

---- ----------------------------------

800 0

1600 0

1250 1

2975 1

1250 1

2850 1

**SUBSTR():**

It returns a part of a string which is present in the given input.

\*\* the input can be columnName/literal.

SYNTAX: select substr(arg1, arg2, arg3) from tableName;

Arg1-🡪 columnName/literal

Arg2--🡪 starting position of the character

Arg3---🡪 No of characters to pick.

EX: SELECT SUBSTR(‘NISHANTH’, 2,6) FROM DUAL; O/P: ISHANT

SELECT SUBSTR(‘NISHANTH’, 1,3) FROM DUAL; O/P: NIS

SELECT SUBSTR(‘ENGINEER’,-5,2) FROM DUAL; O/P: IN

\*\* Note\*\*: Substring will take maximum of 3 arguments and minimum of 2 arguments, if 3rd argument is not given , then it will execute till the end of the String.

**EX: WAQTD last 3 characters of emp names.**

SELECT ENAME,SUBSTR(ENAME,-3,3) FROM EMP;

ENAME SUB

---------- ---

SMITH ITH

ALLEN LEN

**MULTIROW FUNCTIONS:**

It will take multiple functions and gives the single output.

Multirow functions are: **1) MIN ()**

SELCT MIN(SAL) FROM EMP;

**2)MAX ():** SELECT MAX(SAL) FROM EMP;

**3)SUM ():** SELECT SUM(SAL) FROM EMP;

**4)AVG ():** SELECT AVG(SAL) FROM EMP;

**5)COUNT ():** SELECT COUNT (\*) FROM EMP; O/P: 14

SELECT COUNT(ENAME) FROM EMP; O/P:14

NOTE\*\* we can have a combination of a single row functions and a column name in select statement but, we cannot have a combination of multi-row functions and a column name in the select statement.

\*\* If we write multi-row functions and column name in select statement then we are going to get the errors.

**Ex:** SELECT MIN(SAL),SAL FROM EMP; O/P: ERROR

**DISTINCT ():**

Distinct Keyword is used to get the unique values present in a column.

* Distinct can be used in the select statement.
* We can have a combination of distinct, columnName.

i.e. (distinct(colName),columnName)

**EX:** SELECT DISTINCT(DEPTNO),JOB FROM EMP;

DEPTNO JOB

------ ---------

20 CLERK

30 SALESMAN

20 MANAGER

30 CLERK

10 PRESIDENT

\*\*Note\*\* we cannot write more than one distinct () in select statement and also we cannot write more than one column in distinct function.

EX: SELECT DISTINCT (DEPTNO, JOB) FROM EMP; // ERROR

SELECT DISTINCT (DEPTNO), DISTINCT(JOB) FROM EMP; // ERROR