

Experiment No 5

Objectives: Write SQL Queries using Character, Number, Date and group Functions.

Order by: The order by clause is used to display the results in sorted order.

Group by: The attribute or attributes given in the clauses are used to form groups. Tuples with the same value on all attributes in the group by clause are placed in one group.

Having: SQL applies predicates (conditions) in the having clause after groups have been formed, so aggregate function be used.

1. List the emps in the asc order of their Salaries?

A) select * from emp order by sal asc;

2. List the details of the emps in asc order of the Dptnos and desc of Jobs?

A) select * from emp order by deptno asc, job desc;

3. Display all the unique job groups in the descending order?

A) select distinct job from emp order by job desc;

4. List the emps in the asc order of Designations of those joined after the second half of 1981.

A) select * from emp where hiredate > ('30-jun-81') and to_char(hiredate, 'YYYY') = 1981 order by job asc;

5. List all the emps except 'PRESIDENT' & 'MGR' in asc order of Salaries.

A) Select * from emp where job not in ('PRESIDENT', 'MANAGER') order by sal asc;

B) Select * from emp where job not like 'PRESIDENT' and job not like 'MANAGER' order by sal asc;

C) Select * from emp where job != 'PRESIDENT' and job <> 'MANAGER' order by sal asc;

6. List the Enames those are having five characters in their Names.

A) select ename from emp where length (ename) = 5;

7. List the Enames those are starting with 'S' and with five characters.

A) select ename from emp where ename like 'S%' and length (ename) = 5;

8. List the Five-character names starting with 'S' and ending with 'H'.

A) select * from emp where length(ename) = 5 and ename like 'S%H';

9. List the emps who joined in January.

A) select * from emp where to_char (hiredate, 'mon') = 'jan';

10. List the emps who joined in the month of which second character is 'a'.

A) select * from emp where to_char(hiredate, 'mon') like '_a_'; (OR)

B) select * from emp where to_char(hiredate, 'mon') like '_a%';

11. List the emps whose Sal is four digit number ending with Zero.

A) select * from emp where length (sal) = 4 and sal like '%0';

12. List the emps those who joined in 80's.

A) select * from emp where to_char(hiredate, 'yy') like '8%';

13. List all the emps who joined before or after 1981.

A) select * from emp where to_char (hiredate, 'YYYY') not in ('1981'); (OR)

B) select * from emp where to_char (hiredate, 'YYYY') != '1981'; (OR)

C) select * from emp where to_char(hiredate,'YYYY') <> '1981'; (OR)

D) select * from emp where to_char(hiredate,'YYYY') not like '1981';

14. List the emps who are working under 'MGR'.

A) select e.ename || ' works for ' || m.ename from emp e ,emp m where e.mgr = m.empno ; (OR)

B) select e.ename || ' has an employee ' || m.ename from emp e , emp m where e.empno = m.mgr;

15. List the emps who joined in any year but not belongs to the month of March.

A) select * from emp where to_char(hiredate,'MON') not in ('MAR'); (OR)

B) select * from emp where to_char(hiredate,'MON') != 'MAR'; (OR)

C) select * from emp where to_char(hiredate,'MONTH') not like 'MAR%'; (OR)

D) select * from emp where to_char(hiredate,'MON') <> 'MAR';

16. Give a count of how many employees are working in each department

A) select count(empid),deptid from emp group by deptid;

17. Display total salary spent for each job category

A) select job, sum (sal) from emp group by job;

The Helpful Dual

Oracle Database provides a single-row, single-column table called DUAL that is useful for many purposes, not the least of which is learning about Oracle functions. DUAL is an Oracle system table owned by the SYS user, not the SQL_101 schema. Many Oracle system tables are made available to all users via public synonyms. Synonyms will be discussed in subsequent articles in this series.

The DUAL table contains no data that's useful in and of itself. (It has one row with one column—called the DUMMY column—that contains the value X.) You can use DUAL to try out functions that work on string literals and, as you will see in subsequent articles in this series, on number literals and even on today's date.

The following demonstrates the single-row, single-column output of a SELECT statement executed against the DUAL table:

```
SQL> select * from dual;
```

```
D - X
```

```
1 row selected.
```

To display the current date, you can query the DUAL table as follows:

```
SQL> select sysdate from dual;
```

```
SYSDATE
```

```
18-APR-12
```

```
1 row selected.
```

7: Query that demonstrates the LENGTH function

```
SQL> select first_name, LENGTH(first_name) length from employee order by length desc, first_name;
```

FIRST_NAME	LENGTH
Frances	7
Matthew	7
michael	7

Donald	6
Betsy	5
Emily	5
Roger	5
mark	4

8 rows selected.

8: Query that demonstrates a function in a WHERE clause

SQL> select first_name||' '||last_name employee_name from employee where LENGTH(first_name||' '||last_name) > 15 order by employee_name;

EMPLOYEE_NAME

Matthew Michaels

michael peterson

2 rows selected.

When used with the DISTINCT option, the COUNT function shows how many distinct values are in a data set. The two queries in Example 4-38 show the total number of departments and the number of departments that have employees.

Example 7: Counting the Number of Distinct Values in a Set

SELECT COUNT(*) FROM DEPARTMENTS;

Result:

COUNT(*)

27

SELECT COUNT(DISTINCT DEPARTMENT_ID) "Number of Departments" FROM EMPLOYEES;

Result:

Number of Departments

11
