LABORATORY SESSION 4

Title: Functions in SQL

Objective: To understand how to work with functions.

Theory:

Dual

The **DUAL table** is a special one-row, one-column **table** present by default in **Oracle** and other database installations. In **Oracle**, the **table** has a single VARCHAR2(1) column called DUMMY that has a value of 'X'. It is suitable for use in selecting a pseudo column such as SYSDATE, USER, etc.

Example**:**

```
desc dual;
select sysdate from dual;
```

Aggregate Functions:

Syntax:

MIN – `SELECT MIN(<aggregate_expression>)FROM <table_name> [WHERE <conditions>];`

MAX – `SELECT MAX(<aggregate_expression>)FROM <table_name> [WHERE <conditions>];`

SUM – `SELECT SUM(<aggregate_expression>)FROM <table_name> [WHERE <conditions>];`

AVG – `SELECT AVG(<aggregate_expression>)FROM <table_name> [WHERE <conditions>];`

COUNT – `SELECT COUNT(<field / *>)FROM <table_name> [WHERE <conditions>];`

String Functions:

Syntax:

LOWER – `SELECT LOWER(<string1>) FROM <table_name>;`

UPPER – `SELECT UPPER(<string1>) FROM <table_name>;`

TRIM – `SELECT TRIM (<string1> FROM <table_name>;`

LTRIM – `SELECT LTRIM (<string1> [, <trim_string>]) FROM <table_name>;`

`LTRIM` removes from the left end of *`char`* all of the characters contained in *`set`*. If you do not specify *`set`*, then it defaults to a single blank. Oracle Database begins scanning *`char`* from its first character and removes all characters that appear in *`set`* until reaching a character not in *`set`* and then returns the result.

Both *`char`* and *`set`* can be any of the data types `CHAR`, `VARCHAR2`, `NCHAR`, `NVARCHAR2`, `CLOB`, or `NCLOB`. The string returned is of `VARCHAR2` data type if *`char`* is a character data type, `NVARCHAR2` if *`char`* is a national character data type, and a LOB if *`char`* is a LOB data type.

RTRIM – `SELECT RTRIM (<string1> [, <trim_string>]) FROM <table_name>;`

Date Functions:

Syntax:

ADD_MONTHS– `SELECT ADD_MONTHS( <date1>, <number_months> ) FROM <table_name>;`

MONTHS_BETWEEN - `SELECT MONTHS_BETWEEN(<date1>, <date2>) FROM <table_name>;`

TO_DATE – `SELECT TO_DATE (<string1>,<format>, [<nls_language>]) FROM <table_name>;`

```
select to_date('23rd February, 2022',
               'dd"rd" month, yyyy')
from dual;
```

Assignments:

Perform the following queries using DUAL:

SQL> select * from dual;

D (Dummy)
-
X (reference to all system library)

1. Display the current DATE and TIME.
   select sysdate,current_timestamp from dual;

2. Multiply 2 by 2 .
   select 2*2 from dual;

3. Find the absolute value of -15
   select abs(-15) from dual;

4. Calculate the square root of 5.
   select sqrt(5) from dual;

5. Round off 15.19 to one decimal point:
   select round(15.19, 1) from dual;

6. Display the name "IVAN BAYROSS" in LOWERCASE.
   select lower('IVAN BAYROSS') from dual;

7. Display the name "IVAN BAYROSS" in UPPERCASE.
   select upper('IVAN BAYROSS') from dual;

8. Add 5 months to the present date and print the output.
select add_months(sysdate, 5) from dual;

9. Display the number of months between '02-JAN-01' and '02-JUL-01'

```
select months_between(to_date('02-jul-01'), to_date('02-jan-01')) from dual;
```

10. Print the system date in the particular format 'DD/MM/YYYY'

```
select to_CHAR(sysdate, 'dd/mm/yyyy') New_date from dual;
```

Create the following table and insert 10 rows in the table:

EMP (EMPNO , ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPT _NAME)

```
create table emp_by_39(
        empno number(4) primary key check(empno between 7000 and 8000),
        ename varchar2(10),
        job varchar2(15),
        mgr number(4) check(mgr between 7000 and 8000),
        hiredate date,
        sal number(10),
        comm number(4) default(0) check(comm<1500),
        dept_name varchar2(15)
);
```

11. EMPNO must be between 7000 and 8000
12. ENAME must not exceed 10 characters
13. MGR is managers EMPNO
14. COMM (commission) must be under 1500 and defaults to O. Only who works as salesman gets certain commission.
15. DEPT _NAME is the name of the department in which the employees works.

Write SQL queries for the following:

14. List names of employees who are not managers.

```
select ename from emp_by_39
where empno<>mgr;
```

15. List the names of employees whose names have " i " as the second character .

```
select ename from emp_by_39
   where ename like('_i%');
```

16. Find the total number of managers.

```
select count(ename) Total_Manager from emp_by_39
where empno=mgr;
```

17. Display the highest, lowest, Sum and average of all employees salary. Label the columns as "Maximum" "Minimum" "Total" and "Average".

```
select max(sal) Maximum, min(sal) Minimum, sum(sal) Total, avg(sal) Average
from emp_by_39;
```

18. Modify-the above query to display the highest, lowest, sum and average salary for each job type.

```
select job, max(sal) Maximum, min(sal) Minimum, sum(sal) Total, avg(sal)
Average from emp_by_39 group by job;
```

19. Determine the number of managers. Label the column "Number of managers"

```
select count(ename) "Number of Managers"
      from emp_by_39
      where empno=mgr;
```

20. Find the employees who were hired after '01-jan-1980'

```
select * from emp_by_39
where hiredate > '01-jan-1980';
```

21. Display the name of employee who earns maximum salary whose job is Salesman

```
select ename from emp_by_39
where sal=(select max(sal) from emp_by_39 where job like 'Salesman');
```

22. Display the name of employee who earns minimum salary and whose job is clerk.

```
select ename from emp_by_39
where sal=(select min(sal) from emp_by_39 where job like 'Clerk');
```

23. List the name of the employee whose salary is more than ‘Eshani’

```
select ename from emp_by_39
where sal>(select sal from emp_by_39 where ename= 'Eshani');
```

24. Display the name of the department in which 'Akash' works.

```
select dept_name from emp_by_39 where ename='Akash';
```

25. Display the name of the department whose salary is maximum.

```
select distinct dept_name from emp_by_39
where sal=(select max(sal) from emp_by_39);
```

Questionnaire:

1. What are aggregate functions? Explain how *GROUP BY* works.
2. What is the content of the table *DUAL*? Why is it required?