

**Part A****Aim:**SQL commands:

- i) To perform SQL Numeric Function, String, Date and conversion functions.

**Prerequisite:** Oracle.**Outcome:** Understanding and use of various Oracle functions.

Theory:

Numeric Functions

ABS: returns the absolute value of n

abs(n)

POWER: returns m raised to nth power

power(m,n)

ROUND: returns n rounded to m places to the right of the decimal point

round(n,m)

SQRT: returns square root of n

sqrt(n)

EXP: returns e raised to nth power

exp(n)

FLOOR: returns the largest integer value that is equal to or less than a number

floor(n)

CEILING: returns the smallest integer value that is equal to or greater than a number

ceiling(n)

MOD(m,n): returns the remainder of m divided by n. Returns m if n is 0.

RAND: returns a random number or a random number within a range.

Rand()

Rand()\*(upper value-lower value) e.g. rand()\*(10-1)

String Functions:

SUBSTR:

return a portion of string, beginning at character position, substring\_length characters long.

- If position is 0, then it is treated as 1.
- If position is positive, then Oracle Database counts from the beginning of string to find the first character.
- If position is negative, then Oracle counts backward from the end of string.
- If substring\_length is omitted, then Oracle returns all characters to the end of string. If substring\_length is less than 1, then Oracle returns null.

The following example returns several specified substrings of "ABCDEFGH":

```
SELECT SUBSTR('ABCDEFGH',3,4) "Substring"
FROM DUAL;
```

```
Substring
-----
CDEF
```

```
SELECT SUBSTR('ABCDEFGH',-5,4) "Substring"
FROM DUAL;
```

```
Substring
-----
CDEF
```

LEN: returns the length of the specified string.

LEN( string )

CONCAT: allows you to concatenate strings together.

CONCAT( string1, string2, ... string\_n )

LOWER: converts all letters in the specified string to lowercase (same for Upper).

LOWER( string )

**Date Functions:**

**CURRENT\_TIMESTAMP:** returns the current date and time (GETDATE function can also be used).

**CURRENT\_DATE** returns the current date in the session time zone, in a value in the Gregorian calendar of datatype DATE.

**DATEADD:** returns a date after which a certain time/date interval has been added.

DATEADD( date, format, days, months, years )

**Syntax**

DateAdd (Date, Format, Days, Months, Years)

Parameter	Description
Date	Enter a date string. The system assumes your entry to be in the format specified in the Format parameter. The default is the current date.
Format	Enter a date format string that describes the contents of the Date parameter. The default is date format 1 (MM/DD/YY).
Days	Enter the number of days. The default is zero (0).
Months	Enter the number of months. The default is zero (0).
Years	Enter the number of years. The default is zero (0).

**DIFFDATE:** returns the difference between two date values, based on the interval specified.

DIFFDATE ( interval, date1, date2 )

Example

```
SELECT DATEDIFF(day,'2016-06-05','2016-08-05') AS DiffDate
```

Result:

DiffDate

61

**Sysdate:**

```
SQL>SELECT SYSDATE FROM DUAL; 29-DEC-08
```

**next\_day:**

```
SQL>SELECT NEXT_DAY(SYSDATE,'WED') FROM DUAL;
```

05-JAN-09

**add\_months:**

```
SQL>SELECT ADD_MONTHS(SYSDATE,2)FROM DUAL;
```

28-FEB-09

**last\_day:**

```
SQL>SELECT LAST_DAY(SYSDATE)FROM DUAL;
```

31-DEC-08

**months\_between:**

```
SQL>SELECT MONTHS_BETWEEN(SYSDATE,HIREDATE)FROM EMP;
```

4

**Least:**

```
SQL>SELECT LEAST('10-JAN-07','12-OCT-07') FROM DUAL;
```

10-JAN-07

**Greatest:**

```
SQL>SELECT GREATEST('10-JAN-07','12-OCT-07')FROM DUAL;
```

10-JAN-07

**to\_char:**

```
SQL> select to_char(sysdate, "dd\mm\yy") from dual;
```

24-mar-05.

**to\_date:**

```
SQL> select to date (sysdate, "dd\mm\yy") from dual;
```

24-mar-05

**CONVERSION FUNCTIONS:**

To\_char: TO\_CHAR (number) converts n to a value of VARCHAR2 data type, using the optional number format fmt. The value n can be of type NUMBER, BINARY\_FLOAT, or BINARY\_DOUBLE.

The following are number examples for the TO\_CHAR function.

```
TO_CHAR(1210.73, '9999.9')
```

*Result:* ' 1210.7'

```
TO_CHAR(-1210.73, '9999.9')
```

*Result:* '-1210.7'

```
TO_CHAR(1210.73, '9,999.99')
```

*Result:* ' 1,210.73'

TO\_CHAR(1210.73, '\$9,999.00')

Result: ' \$1,210.73'

TO\_CHAR(21, '000099')

Result: ' 000021'

To\_number : TO\_NUMBER converts expr to a value of NUMBER data type.

SQL>Select to\_number ('1234.64') from Dual; 1234.64

To\_date:TO\_DATE converts char of CHAR, VARCHAR2, NCHAR, or NVARCHAR2 data type to a value of DATE data type.

SQL>SELECT TO\_DATE('January 15, 1989, 11:00 A.M.')FROM DUAL; TO\_DATE ----- 15-JAN-89

#### Procedure:

1. Formulate the query for given problem.
2. Write the SQL query with proper input.
3. Execute the query.

#### Practice Exercise:

Create a table EMPLOYEE with following schema:

(Emp\_no, E\_name, E\_address, E\_ph\_no, Dept\_no, Dept\_name, Job\_id, Designation , Salary)

1. List the E\_no, E\_name, Salary of all employees working for MANAGER.
2. Display all the details of the employee whose salary is more than the Sal of any IT PROFF..
3. List the employees in the ascending order of Designations of those joined after 1981.
4. List the employees along with their Experience and Daily Salary.
5. List the employees who are either 'CLERK' or 'ANALYST' .
6. List the employees who joined on 1-MAY-81, 3-DEC-81, 17-DEC-81, 19-JAN-80 .
7. List the employees who are working for the Deptno 10 or 20.
8. List the Enames that are starting with 'S' .
9. Display the name as well as the first five characters of name(s) starting with 'H'
10. List all the emps except 'PRESIDENT' & 'MGR' in asc order of Salaries.

#### Instructions:



1. Write and execute the query in Oracle SQL server.
2. Paste the snapshot of the output in input & output section.

#### Part B

**Code and Output:**

Perform the operation and paste the running code here.

**create table employee(empno number(8) not null unique, ename varchar(20) not null, eaddress varchar(12), ephn number(10) not null unique, deptno number(5) not null, deptname varchar(12) not null, hiredate date, designation varchar(12) not null, salary number(10) not null);**

☒ Autocommit Rows 10   Save Run



```
begin
insert into employee values(1,'Ramesh','kommadi',9834124567,10,'management',TO_DATE('9/01/1980','DD/MM/YYYY'),'it proff',100000);
insert into employee values(2,'Suresh','Vizag',983412321,20,'sales',TO_DATE('11/01/2018','DD/MM/YYYY'),'clerk',80000);
insert into employee values(3,'Mahesh','kovvuru',9123454567,10,'management',TO_DATE('15/02/2014','DD/MM/YYYY'),'analyst',50000);
insert into employee values(4,'Shannu','Rajavolu',9342156792,30,'business',TO_DATE('19/04/2010','DD/MM/YYYY'),'manager',90000);
insert into employee values(5,'Sandhya','Rajahmundry',9984512345,10,'management',TO_DATE('19/12/2009','DD/MM/YYYY'),'it proff',150000);
insert into employee values(6,'Bhavani','Mandapeta',9831236723,30,'business',TO_DATE('09/07/2016','DD/MM/YYYY'),'clerk',90000);
insert into employee values(7,'maheswari','kommadi',9814924567,10,'management',TO_DATE('01/06/1981','DD/MM/YYYY'),'analyst',100000);
insert into employee values(8,'Harish','kommadi',9834114567,20,'sales',TO_DATE('10/12/2009','DD/MM/YYYY'),'analyst',140000);
insert into employee values(9,'Hari','kommadi',9834194567,10,'management',TO_DATE('03/12/1981','DD/MM/YYYY'),'it proff',100000);
insert into employee values(10,'Sandhya','kommadi',9214124567,10,'management',TO_DATE('17/12/1981','DD/MM/YYYY'),'clerk',100000);
insert into employee values(11,'Sandhya','kommadi',9999234567,30,'business',TO_DATE('11/02/1987','DD/MM/YYYY'),'mgr',130000);
insert into employee values(12,'Sandhya','kommadi',9811112367,20,'sales',TO_DATE('12/08/1979','DD/MM/YYYY'),'president',150000);
end;
/
```

Results Explain Describe Saved SQL History

1 row(s) inserted.

0.01 seconds

**1.select empno, ename, salary from employee where designation='manager';**

☒ Autocommit Rows 10   Save Run



```
select empno,ename,salary from employee where designation='manager';
```

Results Explain Describe Saved SQL History

EMPNO	ENAME	SALARY
4	Shannu	90000

1 rows returned in 0.00 seconds [Download](#)

**2.select \* from employee where designation!='it proff' and salary> any(select salary from employee where designation='it proff');**

☒ Autocommit Rows   



```
select * from employee where designation!='it proff' and salary> any(select salary from employee where designation='it proff');
```

**Results** Explain Describe Saved SQL History

EMPNO	ENAME	EADDRESS	EPHN	DEPTNO	DEPTNAME	HIREDATE	DESIGNATION	SALARY
12	Sandhya	kommadi	9811112367	20	sales	08/12/1979	president	150000
8	Harish	kommadi	9834114567	20	sales	12/10/2009	analyst	140000
11	Sandhya	kommadi	9999234567	30	business	02/11/1987	mgr	130000

3 rows returned in 0.01 seconds [Download](#)

**3.select \* from employee where hiredate>'12/31/1981' order by designation desc;**

☒ Autocommit Rows   

```
select * from employee where hiredate>'12/31/1981' order by designation desc;
```



**Results** Explain Describe Saved SQL History

EMPNO	ENAME	EADDRESS	EPHN	DEPTNO	DEPTNAME	HIREDATE	DESIGNATION	SALARY
11	Sandhya	kommadi	9999234567	30	business	02/11/1987	mgr	130000
4	Shannu	Rajavolu	9342156792	30	business	04/19/2010	manager	90000
5	Sandhya	Rajahmundry	9984512345	10	management	12/19/2009	it proff	150000
2	Suresh	Vizag	983412321	20	sales	01/11/2018	clerk	80000
6	Bhavani	Mandapeta	9831236723	30	business	07/09/2016	clerk	90000
8	Harish	kommadi	9834114567	20	sales	12/10/2009	analyst	140000
3	Mahesh	kovvuru	9123454567	10	management	02/15/2014	analyst	50000

7 rows returned in 0.01 seconds [Download](#)

**4.select empno,ename, trunc(months\_between(sysdate,hiredate)/12) years, round(salary/365,5) from employee;**



☒ Autocommit Rows    Save Run

```
select empno,ename, trunc(months_between(sysdate,hiredate)/12) years, round(salary/365,5) from employee;
```



Results Explain Describe Saved SQL History

EMPNO	ENAME	YEARS	ROUND(SALARY/365,5)
1	Ramesh	41	273.9726
2	Suresh	3	219.17808
3	Mahesh	6	136.9863
4	Shannu	10	246.57534
5	Sandhya	11	410.9589
6	Bhavani	4	246.57534
7	maheswari	39	273.9726
8	Harish	11	383.56164
9	Hari	39	273.9726
10	Sandhya	39	273.9726

More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.00 seconds [Download](#)

5.select empno , ename from employee where designation='clerk' union select empno , ename from employee where designation='analyst';

☒ Autocommit Rows    Save Run

```
select empno , ename from employee where designation='clerk' union select empno , ename from employee where designation='analyst';
```



Results Explain Describe Saved SQL History

EMPNO	ENAME
2	Suresh
3	Mahesh
6	Bhavani
7	maheswari
8	Harish
10	Sandhya

6 rows returned in 0.01 seconds [Download](#)

6.select empno , ename from employee where hiredate='05/01/1981' or hiredate='12/03/1981' or hiredate='12/17/1981' or hiredate='01/19/1980';



☒ Autocommit Rows   

```
select empno , ename from employee where hiredate='05/01/1981' or hiredate='12/03/1981' or hiredate='12/17/1981' or hiredate='01/19/1980';
```

---

**Results** Explain Describe Saved SQL History



EMPNO	ENAME
9	Hari
10	Sandhya

2 rows returned in 0.00 seconds [Download](#)

7. select empno , ename from employee where deptno=10

union

select empno , ename from employee where deptno=20

☒ Autocommit Rows   

```
select empno , ename from employee where deptno=10  
union  
select empno , ename from employee where deptno=20
```



---

**Results** Explain Describe Saved SQL History

EMPNO	ENAME
1	Ramesh
2	Suresh
3	Mahesh
5	Sandhya
7	maheswari
8	Harish
9	Hari
10	Sandhya
12	Sandhya

9 rows returned in 0.00 seconds [Download](#)

8. select empno , ename from employee where ename like 'S\_%';

☒ Autocommit Rows    Save Run



```
select empno , ename from employee where ename like 'S_%';
```

Results Explain Describe Saved SQL History

EMPNO	ENAME
2	Suresh
4	Shannu
5	Sandhya
10	Sandhya
11	Sandhya
12	Sandhya

6 rows returned in 0.00 seconds [Download](#)

9.select ename,substr(ename,1,5) from employee where ename like 'H\_%';

☒ Autocommit Rows    Save Run

```
select ename,substr(ename,1,5) from employee where ename like 'H_%';
```

Results Explain Describe Saved SQL History

ENAME	SUBSTR(ENAME,1,5)
Harish	Haris
Hari	Hari

2 rows returned in 0.01 seconds [Download](#)

10.select \* from employee where designation!='president' and designation!='mgr' order by salary asc;

☒ Autocommit
 Rows 15
Save Run

```
select * from employee where designation!='president' and designation!='mgn' order by salary asc;
```

**Results** Explain Describe Saved SQL History

EMPNO	ENAME	EADDRESS	EPHN	DEPTNO	DEPTNAME	HIREDATE	DESIGNATION	SALARY
3	Mahesh	kovvuru	9123454567	10	management	02/15/2014	analyst	50000
2	Suresh	Vizag	983412321	20	sales	01/11/2018	clerk	80000
6	Bhavani	Mandapeta	9831236723	30	business	07/09/2016	clerk	90000
4	Shannu	Rajavolu	9342156792	30	business	04/19/2010	manager	90000
7	maheswari	kommadi	9814924567	10	management	06/01/1981	analyst	100000
9	Hari	kommadi	9834194567	10	management	12/03/1981	it proff	100000
1	Ramesh	kommadi	9834124567	10	management	01/09/1980	it proff	100000
10	Sandhya	kommadi	9214124567	10	management	12/17/1981	clerk	100000
8	Harish	kommadi	9834114567	20	sales	12/10/2009	analyst	140000
5	Sandhya	Rajahmundry	9984512345	10	management	12/19/2009	it proff	150000

10 rows returned in 0.00 seconds [Download](#)

### Observation & Learning:

Practiced SQL Numeric Function, String, Date and conversion functions queries .

### Conclusion:

Learned NUMERIC functions , date functions ,string functions and their uses.

### Questions:

1. What is the use of aggregate function?
2. How different number of rows can be counted?
3. What is the difference between having and where clause?
4. Dose WHERE clause work with aggregate functions?

### Answers

1. An aggregate function performs a calculation one or more values and returns a single value.
2. Using count(\*) aggregate function,distinct rows can be counted.
3. The WHERE clause is used in the selection of rows according to given conditions whereas the HAVING Clause enables you to specify conditions that filter which group results
4. Aggregate functions cannot be used with a WHERE clause.