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)
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 "ABCDEFG":

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

Substring
-----
CDEF

SELECT SUBSTR('ABCDEFG',-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:

SEC: CSE 4

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:

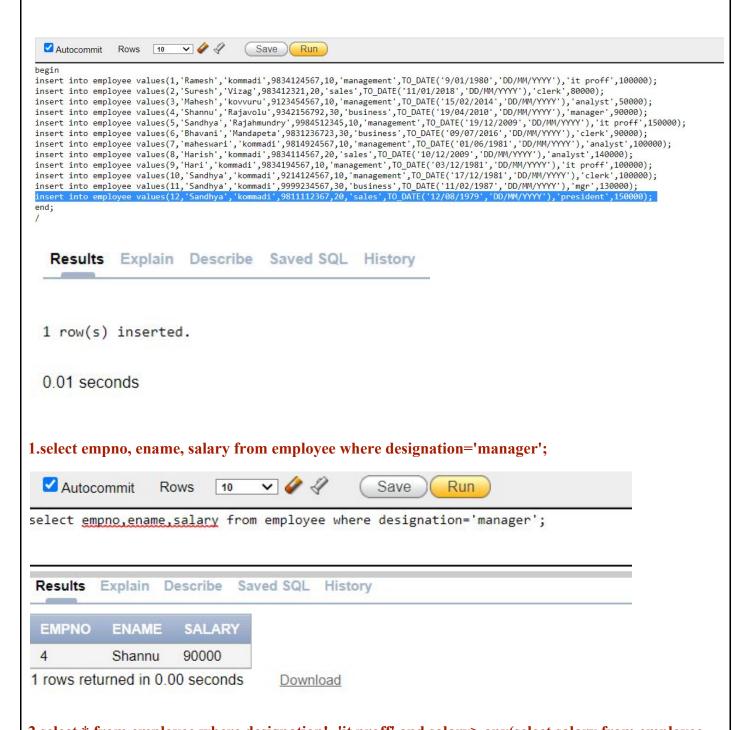
- 1. Write and execute the guery in Oracle SQL server.
- 2. Paste the snapshot of the output in input & output section.

Part B

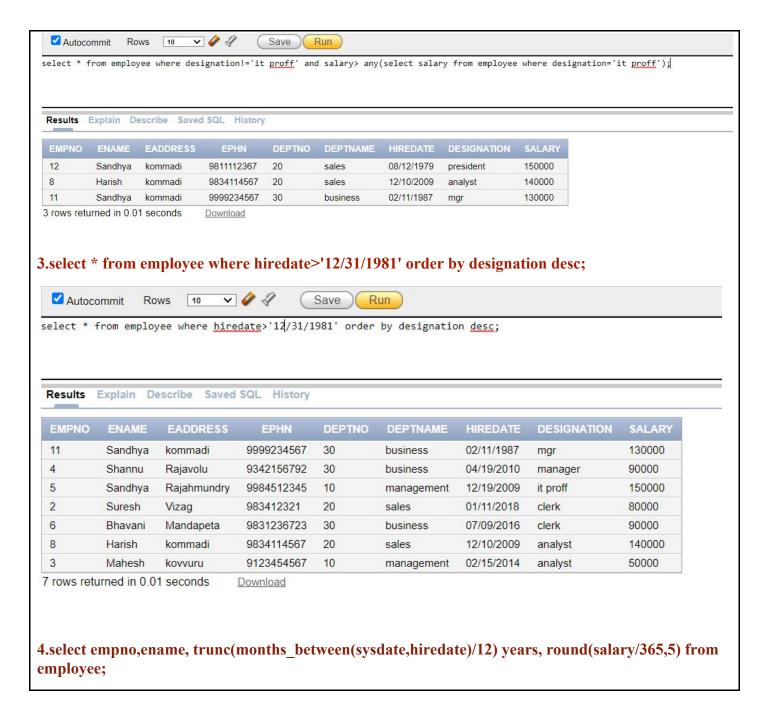
Code and Output:

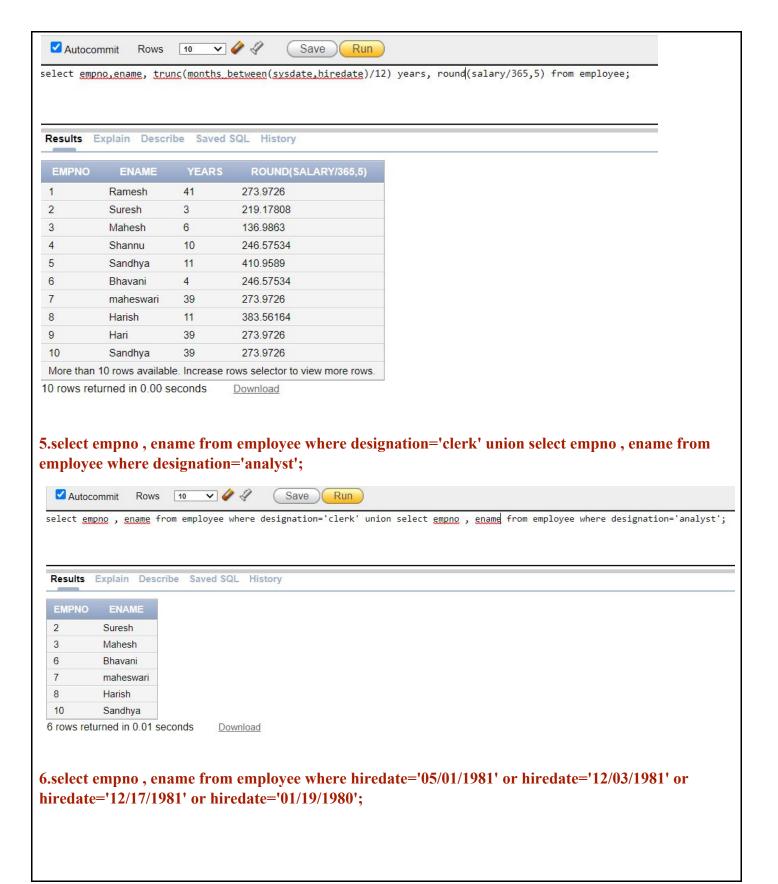
Perform the operation and paste the running code here.

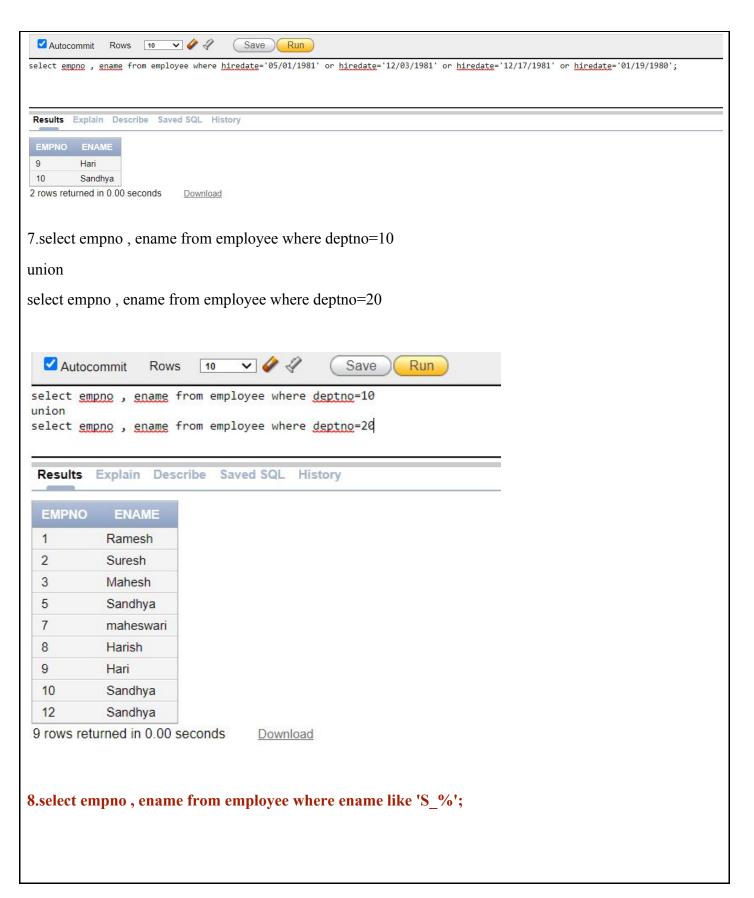
create table employee(empno number(8) not null unique, ename varchar(20) not null, eaddress varchar(12), ephn number(10) not null unique, deptno number(5) not null, deptname varchar(12) not null, hiredate date, designation varchar(12) not null, salary number(10) not null);

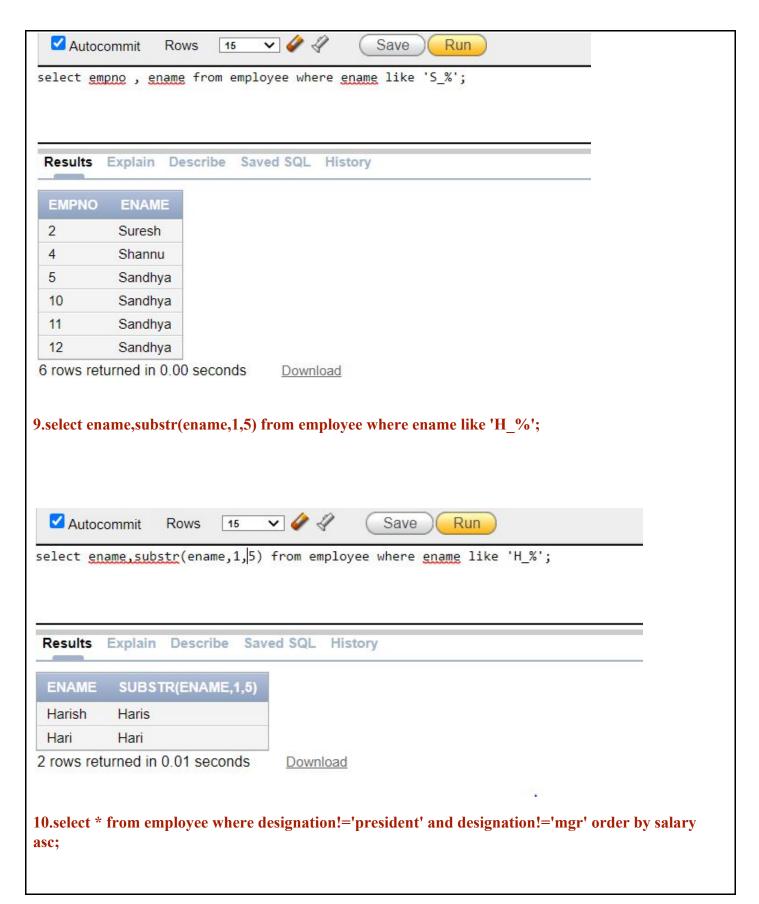


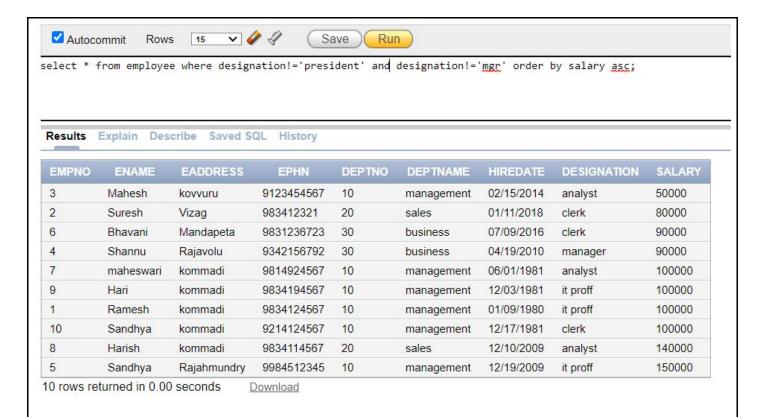
2.select * from employee where designation!='it proff' and salary> any(select salary from employee where designation='it proff');











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.
- 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.