EXP 4

INBUILT FUNCTIONS IN SQL

AIM: To execute different inbuilt functions in SQL.

Character Functions

It calculates the ASCII equivalent of the first character of the given input string. ASCII(<Character>)

CHR(<Character>)

Returns the character equivalent of the given integer.

Example:

SELECT CHR(65), CHR(97) FROM dual;

```
SQL> SELECT CHR(65),CHR(97) FROM DUAL;

C C
- -
A a
```

CONCAT(<string1>,<string2>)

This function returns String2 appended to String1.

Example:

SELECT CONCAT('Fname', 'Lname') Emp_name FROM emp;

```
SQL> SELECT CONCAT('CHITRA',' LEKHA') FROM DUAL;

CONCAT('CHIT
------------
CHITRA LEKHA
```

INITCAP(<String>)

This function returns String with the first character of each word in upper case and rest of all in lower case.

Example:

SELECT INITCAP('oracle aws')

FROM Dual;

O/p Oracle Aws

```
SQL> SELECT INITCAP('oracle aws') FROM DUAL;

INITCAP('O
----------
Oracle Aws
```

INSTR

instr(string1, string2 [, start_position [, nth_Appearance]]):

where,

```
SQL> SELECT INSTR('CHARACTER','R') FROM DUAL;

INSTR('CHARACTER','R')
----------------------
                     4

SQL> SELECT INSTR('CHARACTER','R',1,2) FROM DUAL;

INSTR('CHARACTER','R',1,2)
--------------------------
                         9
```

LENGTH(<Str>)

Returns length of a

string

select length('Sql Tutorial') as len

from dual;

O/p len 12

```
SQL> SELECT LENGTH('CHITRALEKHA') FROM DUAL;

LENGTH('CHITRALEKHA')
---------------------
                   11
```

LOWER(<Str>)

This function returns a character string with all characters in lower case.

```
SQL> SELECT LOWER('CHITRALEKHA') FROM DUAL;

LOWER('CHIT
-----------
chitralekha
```

UPPER(<Str>)

This function returns a character string with all characters in upper case.

```
SQL> SELECT UPPER ('chitralekha') FROM DUAL;

UPPER('CHIT
-----------
CHITRALEKHA
```

LPAD(<Str1>,<i>[,<Str2>])

This function returns the character string Str1 expanded in length to i characters, using Str2 to fill in space as needed on the left side of Str1. Example

SELECT LPAD('Oracle',10,'.') lapd_doted from Dual, would return Oracle

SELECT LPAD('RAM', 7) lapd_exa from Dual would return ' RAM'

```
SQL> SELECT lpad('HEYYOU',3,'*') FROM DUAL;

LPA
---
HEY
```

RPAD(<Str1>,<i>[,<Str2>])

RPAD is same as LPAD but Str2 is padded at the right side

```
SQL> SELECT RPAD('HEYYOU',3,'*') FROM DUAL;

RPA
---
HEY
```

LTRIM(<Str1>[,<Str2>])

The LTRIM function removes characters from the left side of the character String, with all the leftmost characters that appear in another text expression removed. This function returns Str1 without any leading character that appears in Str2.If Str2 characters are leading character in Str1, then Str1 is returned unchanged. Str2 defaults to a single space.

Example

Select LTRIM('datawarehousing','ing') trim1 ,
LTRIM('datawarehousing ')

```
SQL> SELECT LTRIM('DATAWAREHOUSING','DATA') FROM DUAL;

LTRIM('DATA
-----------
WAREHOUSING
```

RTRIM(<Str1>[,<Str2>])

Same as LTRIM but the characters are trimmed from the right side

```
SQL> SELECT RTRIM('DATAWAREHOUSING','HOUSING') FROM DUAL;

RTRIM('D
--------
DATAWARE
```

TRIM([[<Str1>]<Str2> FROM]<Str3>)

If present Str1 can be one of the following literal: LEADING, TRAILING, BOTH. This function returns Str3 with all C1(leading trailing or both) occurrences of characters in Str2 removed. If any of Str1,Str2 or Str3 is Null, this function returns a Null. Str1 defaults to BOTH, and Str2 defaults to a space character.

Example

```
SQL> SELECT TRIM(' ORACLE ') TRIM1, TRIM('ORACLE ') TRIM2 FROM DUAL;

TRIM1  TRIM2
------ ------
ORACLE ORACLE
```

REPLACE(<Str1>,<Str2>,<Str 3>)

This function returns Str1 with all occurrence of Str2 replaced with Str3

Example

SELECT REPLACE (“Oracle”, “Ora“, “Arti“) replace_exa

FROM Dual;

O/p replace_exa

Atricle

```
SQL> SELECT REPLACE('ORACLE','ORA','MIRA') FROM DUAL;

REPLACE
-------
MIRACLE
```

ESSENTIAL NUMERIC FUNCTIONS

ABS()

Select Absolute value

SELECT ABS(-25) "Abs" FROM DUAL;

Abs

15

```
SQL> SELECT ABS(-25) FROM DUAL;

  ABS(-25)
----------
        25
```

ACOS ()

Select cos value

SELECT ACOS(.28)"Arc_Cosine" FROM DUAL;

```
SQL> SELECT ACOS(.28)"Arc_Cosine" FROM DUAL;

Arc_Cosine
----------
1.28700222
```

ASIN ()

Select sin value

SELECT ASIN(.6)"Arc_Cosine" FROM DUAL;

```
SQL> SELECT ASIN(.6)"Arc_Cosine" FROM DUAL;

Arc_Cosine
----------
.643501109
```

ATAN()

Select tan value

SELECT ATAN(.6)"Arc_Cosine" FROM DUAL;

```
SQL> SELECT ATAN(.6)"Arc_Cosine" FROM DUAL;

Arc_Cosine
----------
  .5404195
```

CEIL()

Returns the smallest integer greater than or equal to the order total of a specified SELECT CEIL(239.8) FROM Dual would return 240

```
SQL> SELECT CEIL(239.1) FROM DUAL;

CEIL(239.1)
-----------
        240
```

FLOOR()

Returns the largest integer equal to or less than value. SELECT FLOOR(15.65) "Floor" FROM DUAL;

```
SQL> SELECT FLOOR(15.65) "FLOOR" FROM DUAL;

     FLOOR
----------
        15
```

MOD()

Return modulus value

SELECT MOD(11,3) "Mod" FROM DUAL;

Modulus

```
SQL> SELECT MOD(11,3) "MOD" FROM DUAL;

       MOD
----------
         2
```

POWER()

SELECT POWER(3,2) "Power" FROM DUAL;

```
SQL> SELECT POWER(3,2) "POWER" FROM DUAL;

     POWER
----------
         9
```

ROUND (number)

SELECT ROUND(43.698,1) "Round" FROM DUAL;

Round

```
SQL> SELECT ROUND(43.698,1) "ROUND" FROM DUAL;

     ROUND
----------
      43.7
```

TRUNC (number)

The TRUNC (number) function returns n1 truncated to n2 decimal places. If n2 is omitted, then n1 is truncated to 0 places. n2 can be negative to truncate (make zero) n2 digits left of the decimal point.

SELECT TRUNC(12.75,1) "Trunc" FROM DUAL;

Trunc

12.75

SELECT TRUNC(12.75,-1) "Trunc" FROM DUAL;

Trunc

10

```
SQL> SELECT TRUNC(12.75,1) "TRUNC" FROM DUAL;

     TRUNC
----------
      12.7

SQL> SELECT TRUNC(12.75,-1) "TRUNC" FROM DUAL;

     TRUNC
----------
        10
```

Date And Time Function

ADD_MONTHS(date,number_of_ month)

SELECT SYSDATE, ADD_MONTHS(SYSDATE,2), ADD_MONTHS(SYSDATE,-2) FROM

DUAL;

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

SYSDATE   ADD_MONTH ADD_MONTH
--------- --------- ---------
08-MAR-22 08-MAY-22 08-JAN-22
```

EXTRACT(<type> FROM <date>)

'Type' can be YEAR, MONTH, DAY, HOUR, MIN, SECOND, TIME_ZONE_HOUR, TIME_ZONE_MINUTE, TIME_ZONE_REGION

SELECT SYSDATE, EXTRACT(YEAR FROM SYSDATE)YEAR, EXTRACT(DAY FROM SYSDATE)DAY , EXTRACT(TIMEZONE_HOUR

FROM SYSTIMESTAMP) TZH

FROM DUAL;

```
SQL> SELECT SYSDATE, EXTRACT(YEAR FROM SYSDATE)YEAR, EXTRACT(DAY FROM SYSDATE)DAY , EXTRACT(TIMEZONE_HOUR FROM SYSTIMESTAMP) TZH FROM DUAL;

SYSDATE        YEAR       DAY        TZH
--------- ---------- ---------- ----------
08-MAR-22      2022          8          0
```

LAST_DAY(<date>)

Extract last day of month

Example:

SELECT SYSDATE, LAST_DAY(SYSDATE) END_OF_MONTH FROM DUAL;

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

SYSDATE   END_OF_MO
--------- ---------
08-MAR-22 31-MAR-22
```

NEXT_DAY(<date>,<day>)

SELECT NEXT_DAY('31-Aug-18','SUN') "FIRST MONDAY OF

SEPTEMBER" FROM DUAL;

O/P FIRST MONDAY OF SEPTEMBER

03-Sep-18

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

NEXT_DAY(
---------
13-MAR-22
```

ROUND (date[,<fmt>])

SELECT SYSDATE, ROUND(SYSDATE,'MM'),
ROUND(SYSDATE,'YYYY') FROM DUAL;

Result:

SYSDATE ROUND(SYS ROUND(SYS

10-FEB-18 01-MAR-18 01-JAN-18

```
SQL> SELECT SYSDATE, ROUND(SYSDATE,'MM'), ROUND(SYSDATE,'YYYY') FROM DUAL;

SYSDATE   ROUND(SYS ROUND(SYS
--------- --------- ---------
08-MAR-22 01-MAR-22 01-JAN-22
```

TRUNC(date[,<fmt>])

SELECT SYSDATE, TRUNC(SYSDATE,'MM'),
TRUNC(SYSDATE,'YYYY')

FROM DUAL;

```
SQL> SELECT SYSDATE, TRUNC(SYSDATE,'MM'), TRUNC(SYSDATE,'YYYY') FROM DUAL;

SYSDATE   TRUNC(SYS TRUNC(SYS
--------- --------- ---------
08-MAR-22 01-MAR-22 01-JAN-22
```

MONTHS_BETWEEN

function returns the number of months between

date1 and date2.

SYNTAX

The syntax for the Oracle/PLSQL MONTHS_BETWEEN function is:

MONTHS_BETWEEN(date1, date2)

Parameters or Arguments

date1 and date2 are the dates used to calculate the number of months. If a fractional month is calculated, the MONTHS_BETWEEN function calculates the frac- tion based on a 31-day month.

```
SQL> SELECT MONTHS_BETWEEN(SYSDATE, '8-MAR-22' ) FROM DUAL;

MONTHS_BETWEEN(SYSDATE,'8-MAR-22')
----------------------------------
                                 0

SQL> SELECT MONTHS_BETWEEN(SYSDATE, '20-MAY-22' ) FROM DUAL;

MONTHS_BETWEEN(SYSDATE,'20-MAY-22')
-----------------------------------
                         -2.3800792
```

RESULT:

Thus the inbuilt functions in SQL are successfully executed and outputs are shown.

CHITRALEKHA.CH

RA1911003010387