CSE2007 : Database Management Systems

Name : Gudi Varaprasad

Reg. No. : 19BCE7048

School of Computer Science and Engineering

Lab Slot : L45 + L46

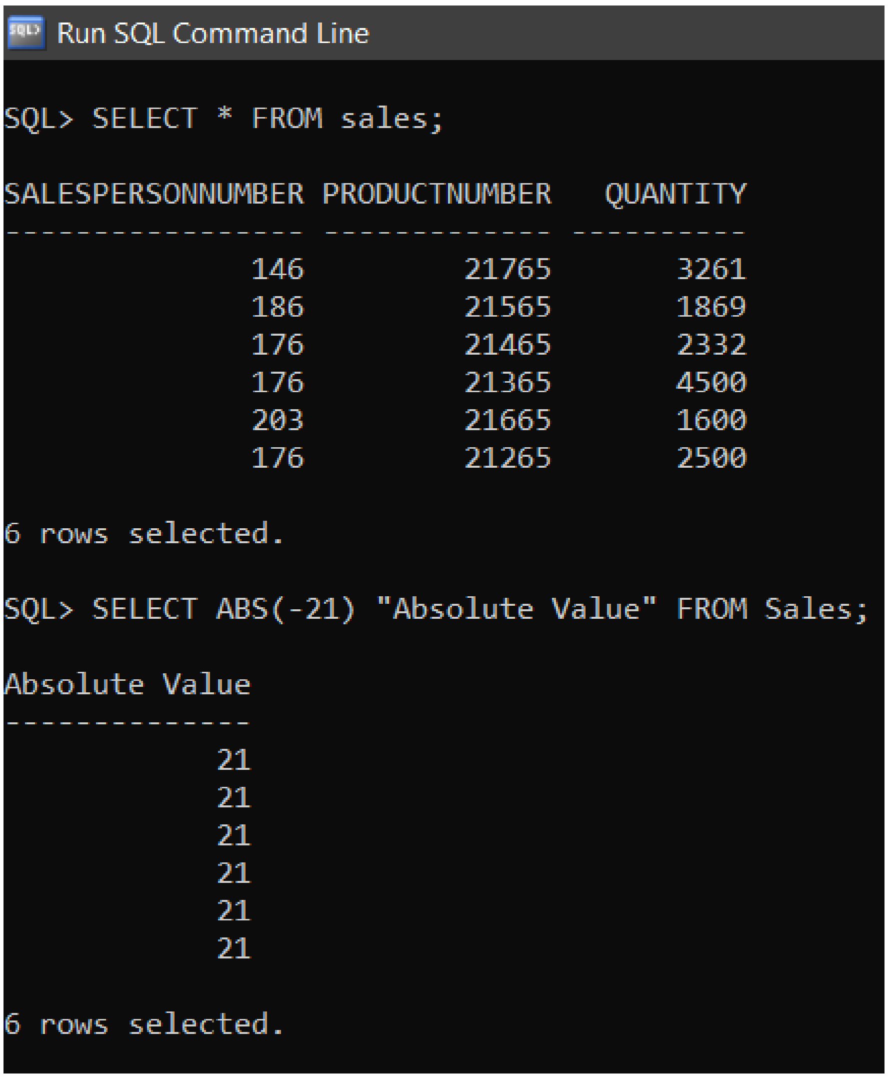
Date : 04 – 03 – 2021

Submitted to : Ms. Dhanavanthini madam

Single row and Group Function Queries

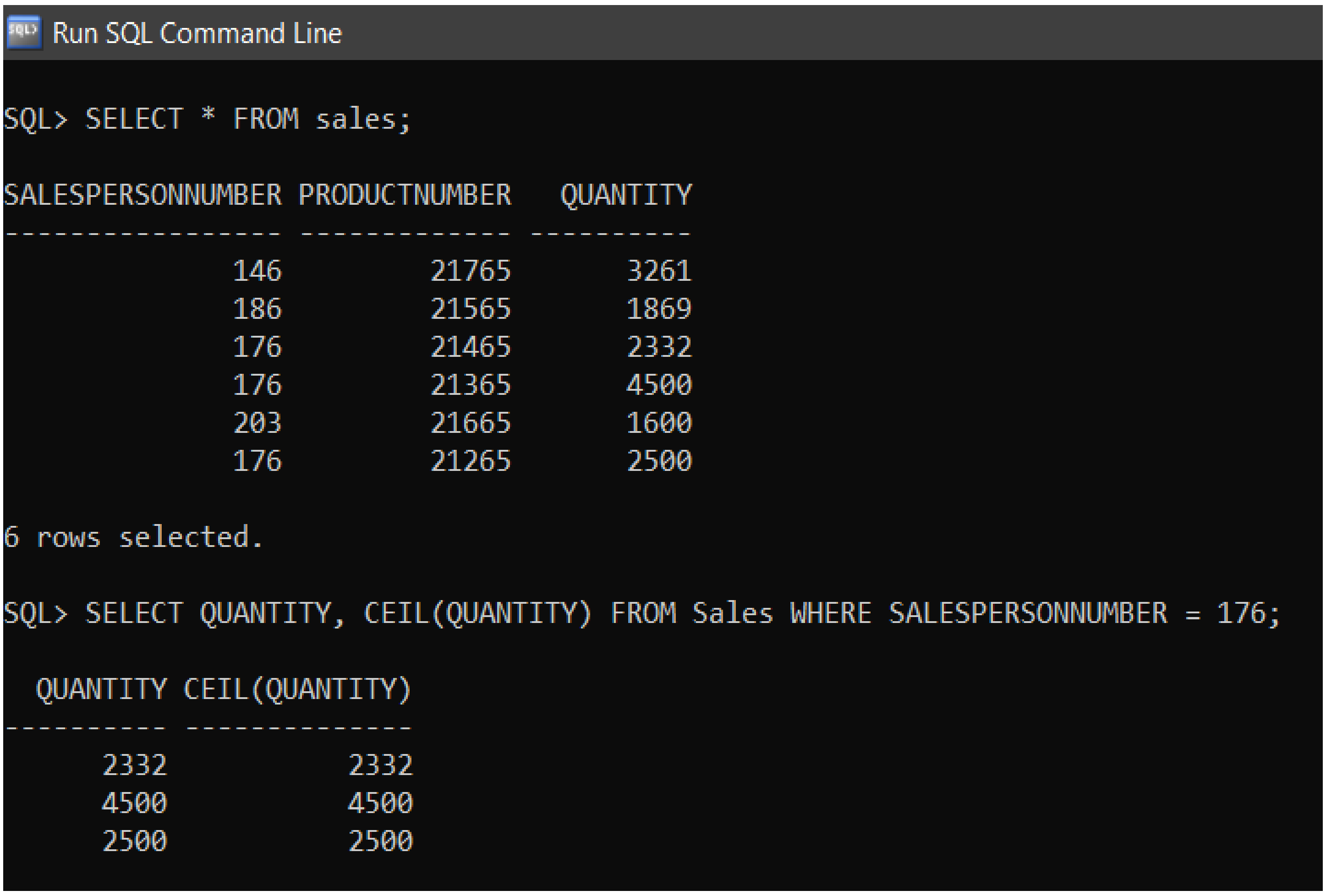
Numeric Functions

1. abs(n)



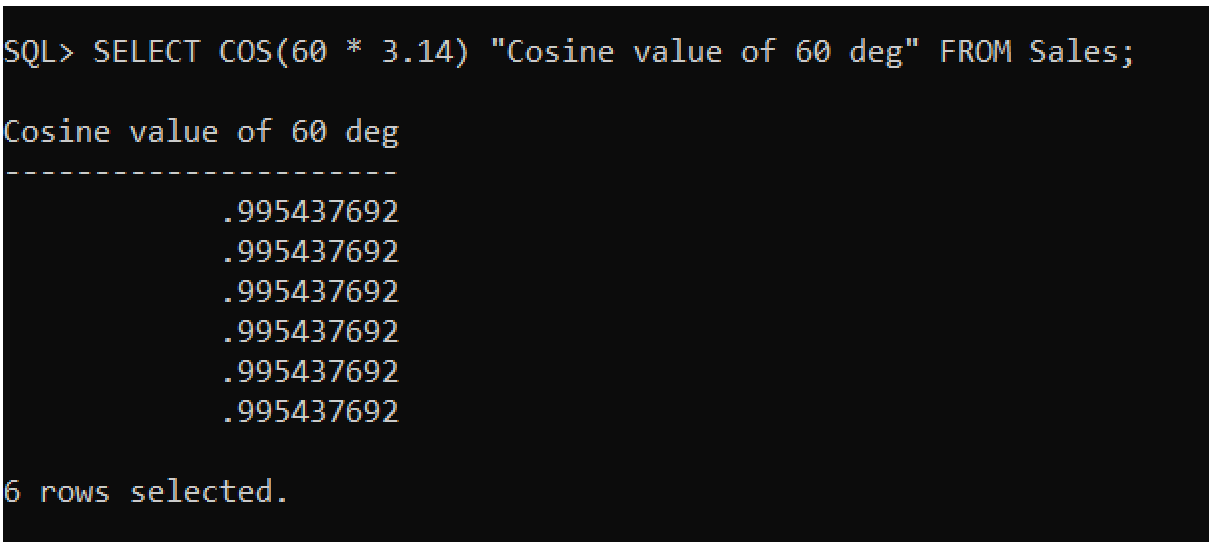
```
SQL> SELECT * FROM sales;

SALESPERSONNUMBER PRODUCTNUMBER   QUANTITY
----------------- ------------- ----------
              146         21765       3261
              186         21565       1869
              176         21465       2332
              176         21365       4500
              203         21665       1600
              176         21265       2500

6 rows selected.

SQL> SELECT ABS(-21) "Absolute Value" FROM Sales;

Absolute Value
--------------
            21
            21
            21
            21
            21
            21

6 rows selected.
```

2. ceil(n)



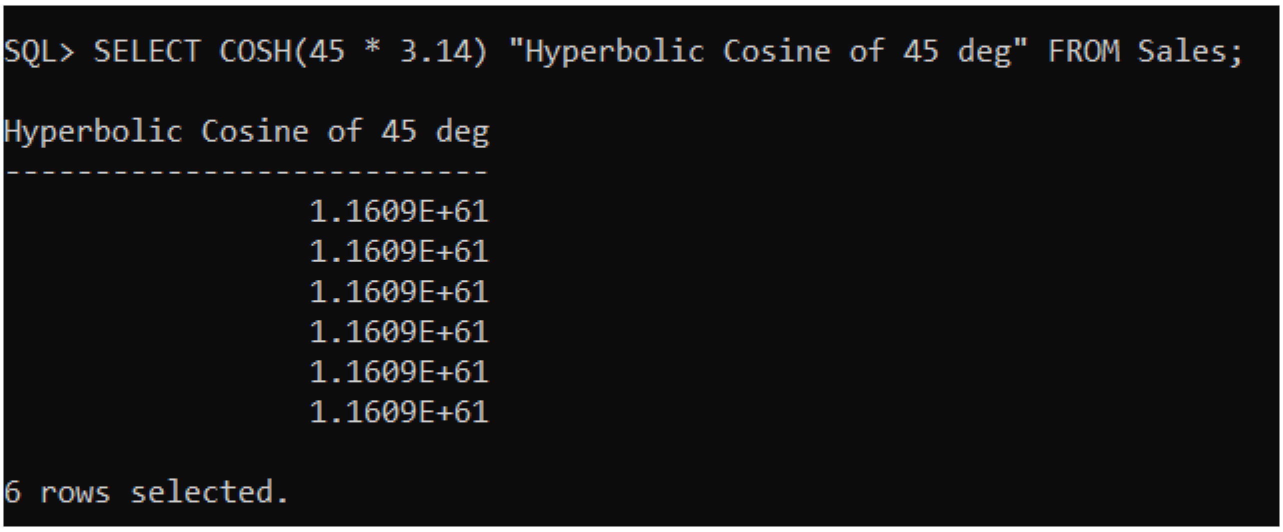
```
SQL> SELECT * FROM sales;

SALESPERSONNUMBER PRODUCTNUMBER   QUANTITY
----------------- ------------- ----------
              146         21765       3261
              186         21565       1869
              176         21465       2332
              176         21365       4500
              203         21665       1600
              176         21265       2500

6 rows selected.

SQL> SELECT QUANTITY, CEIL(QUANTITY) FROM Sales WHERE SALESPERSONNUMBER = 176;

  QUANTITY CEIL(QUANTITY)
---------- --------------
      2332           2332
      4500           4500
      2500           2500
```

3. cos(n)



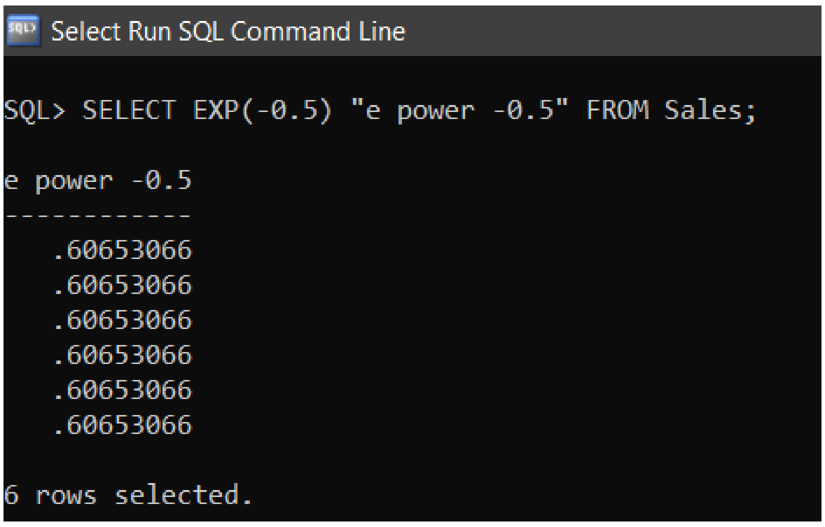
```
SQL> SELECT COS(60 * 3.14) "Cosine value of 60 deg" FROM Sales;

Cosine value of 60 deg
----------------------
            .995437692
            .995437692
            .995437692
            .995437692
            .995437692
            .995437692

6 rows selected.
```

4. cosh(n)



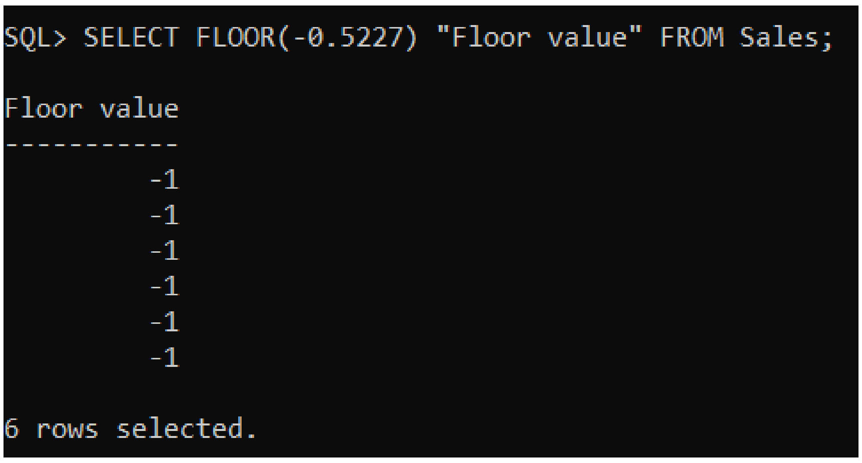
```
SQL> SELECT COSH(45 * 3.14) "Hyperbolic Cosine of 45 deg" FROM Sales;

Hyperbolic Cosine of 45 deg
---------------------------
                 1.1609E+61
                 1.1609E+61
                 1.1609E+61
                 1.1609E+61
                 1.1609E+61
                 1.1609E+61

6 rows selected.
```

5. exp(n)



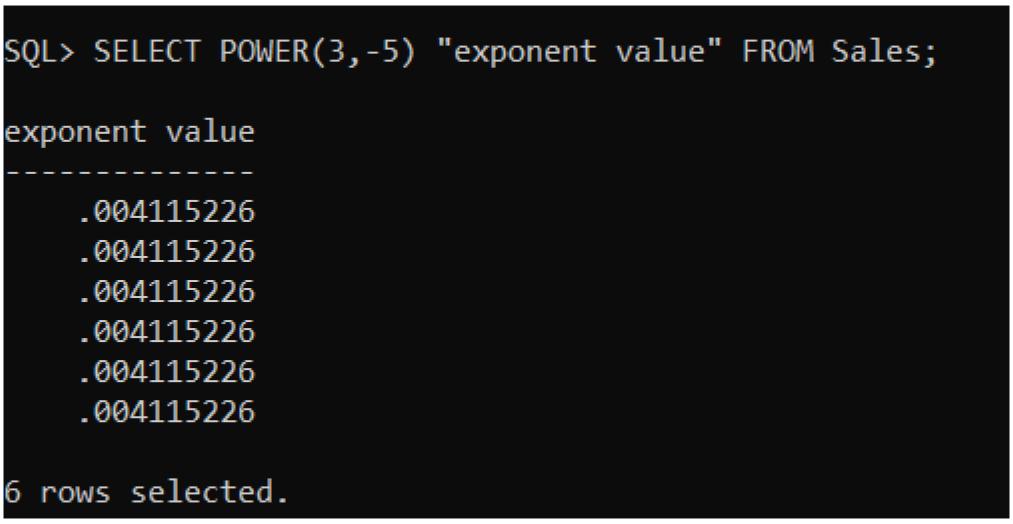
```
SQL> SELECT EXP(-0.5) "e power -0.5" FROM Sales;

e power -0.5
------------
  .60653066
  .60653066
  .60653066
  .60653066
  .60653066
  .60653066

6 rows selected.
```

6. floor(n)



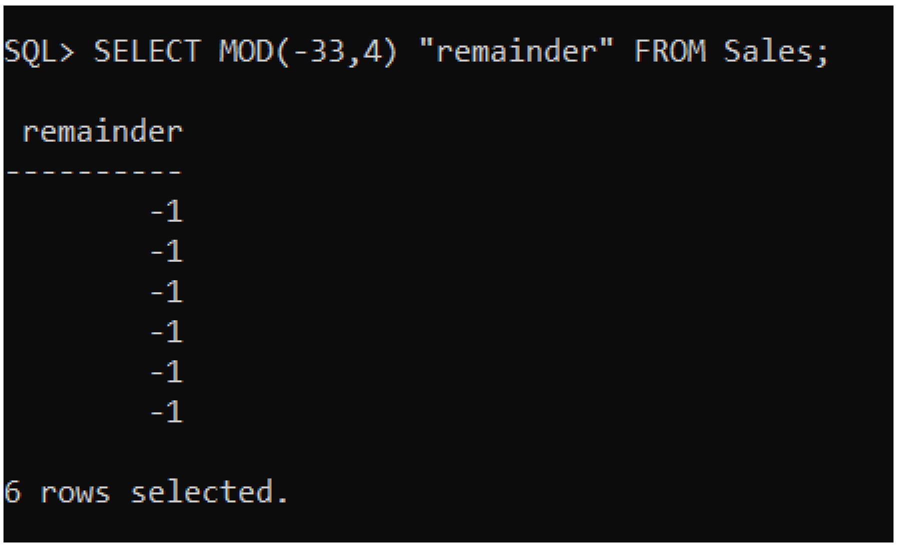
```
SQL> SELECT FLOOR(-0.5227) "Floor value" FROM Sales;

Floor value
-----------
         -1
         -1
         -1
         -1
         -1
         -1

6 rows selected.
```

7. power(m,n)



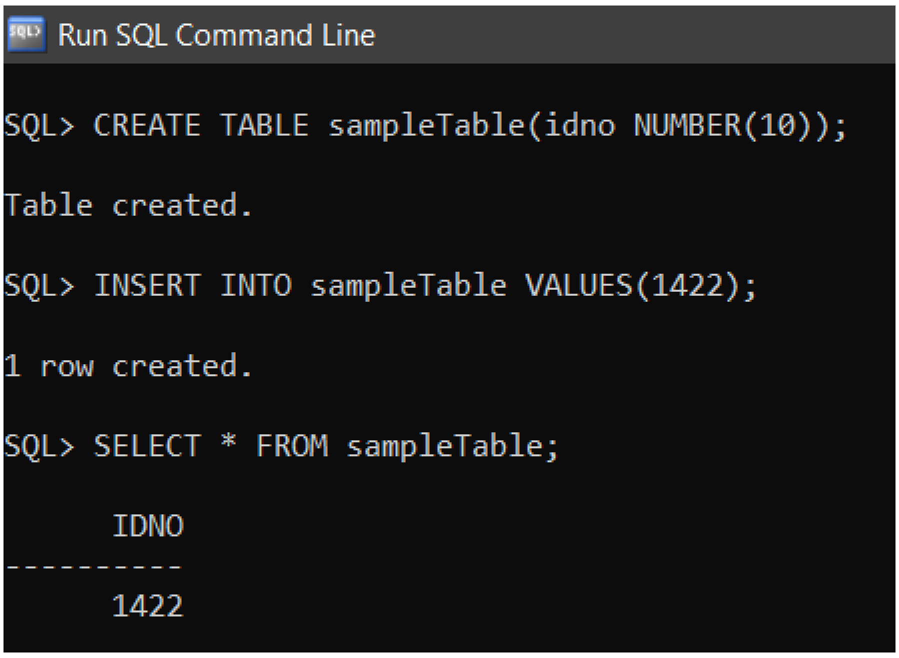
```
SQL> SELECT POWER(3,-5) "exponent value" FROM Sales;

exponent value
--------------
    .004115226
    .004115226
    .004115226
    .004115226
    .004115226
    .004115226

6 rows selected.
```

8. mod(m,n)



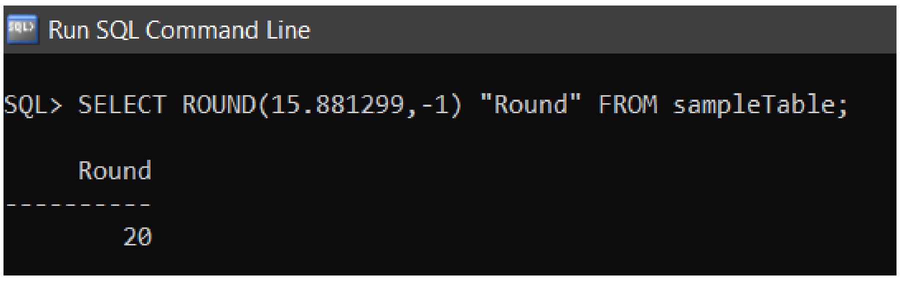
```
SQL> SELECT MOD(-33,4) "remainder" FROM Sales;

 remainder
----------
        -1
        -1
        -1
        -1
        -1
        -1

6 rows selected.
```

Created a Table for performing operations :



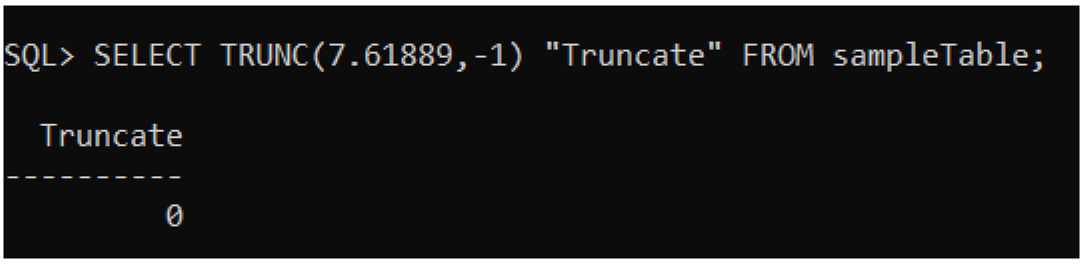
```
SQL> CREATE TABLE sampleTable(idno NUMBER(10));

Table created.

SQL> INSERT INTO sampleTable VALUES(1422);

1 row created.

SQL> SELECT * FROM sampleTable;

      IDNO
----------
      1422
```

9. round(m,n)



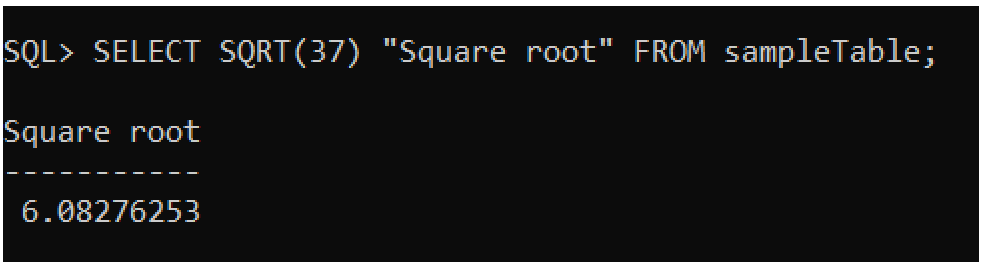
```
SQL> SELECT ROUND(15.881299,-1) "Round" FROM sampleTable;

    Round
---------
       20
```

10. trunc (m,n)



```
SQL> SELECT TRUNC(7.61889,-1) "Truncate" FROM sampleTable;

  Truncate
----------
         0
```

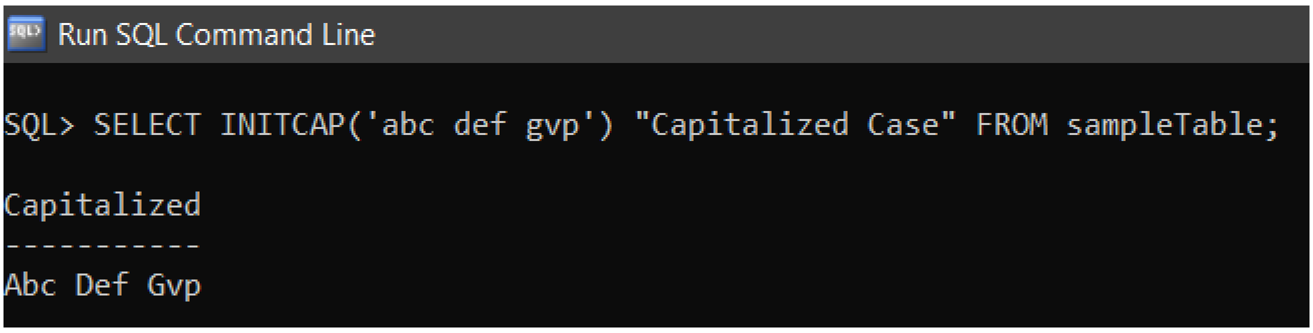
11. sqrt (n)



```
SQL> SELECT SQRT(37) "Square root" FROM sampleTable;

Square root
-----------
 6.08276253
```

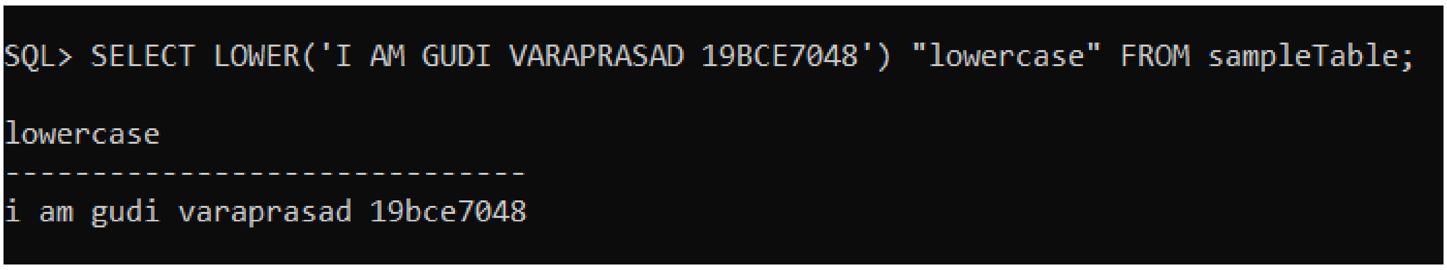
Character Functions

1. initcap(char)



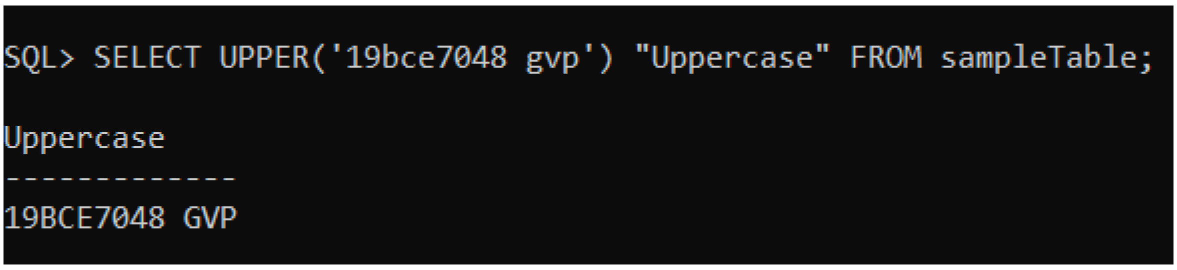
```
SQL> SELECT INITCAP('abc def gvp') "Capitalized Case" FROM sampleTable;

Capitalized
-----------
Abc Def Gvp
```

2. lower (char)



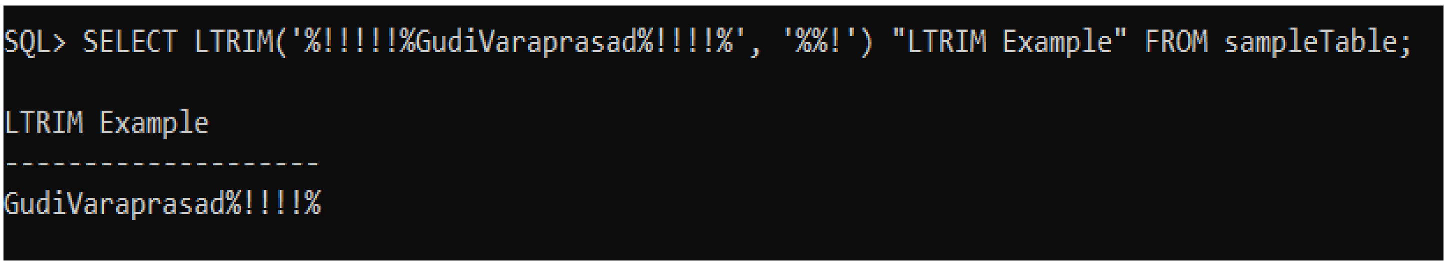
```
SQL> SELECT LOWER('I AM GUDI VARAPRASAD 19BCE7048') "lowercase" FROM sampleTable;

lowercase
------------------------------
i am gudi varaprasad 19bce7048
```

3. upper (char)



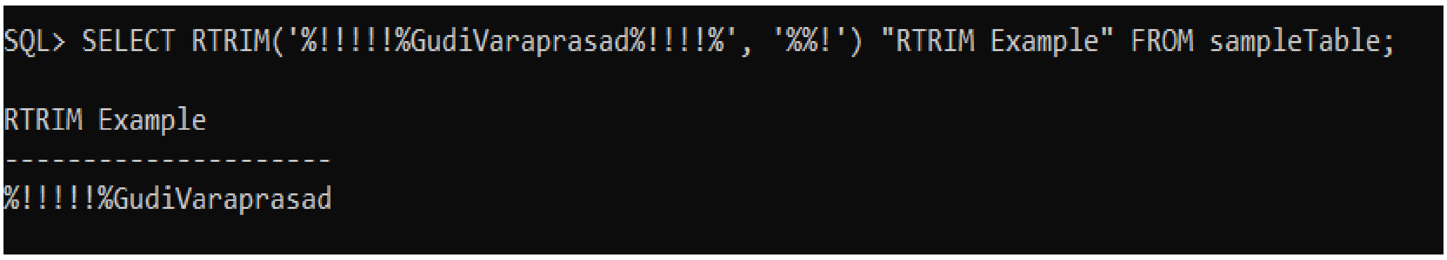
```
SQL> SELECT UPPER('19bce7048 gvp') "Uppercase" FROM sampleTable;

Uppercase
-------------
19BCE7048 GVP
```

4. ltrim (char, set)



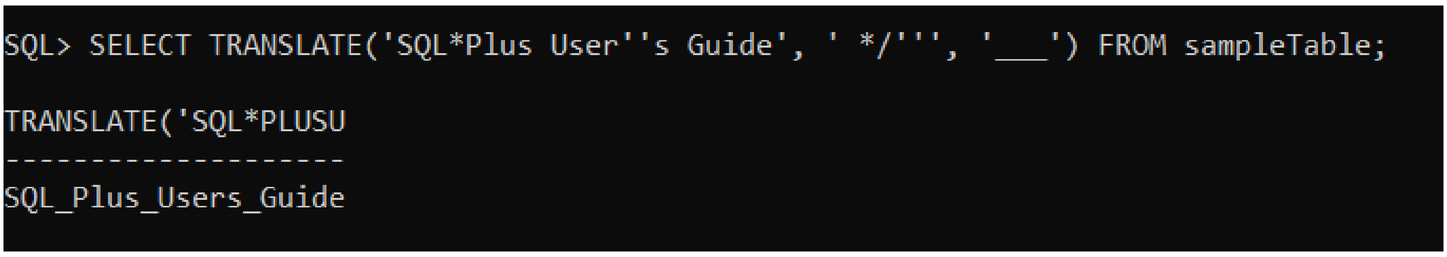
```
SQL> SELECT LTRIM('%!!!!!%GudiVaraprasad%!!!!%', '%%!') "LTRIM Example" FROM sampleTable;

LTRIM Example
--------------------
GudiVaraprasad%!!!!%
```

5. rtrim (char, set)



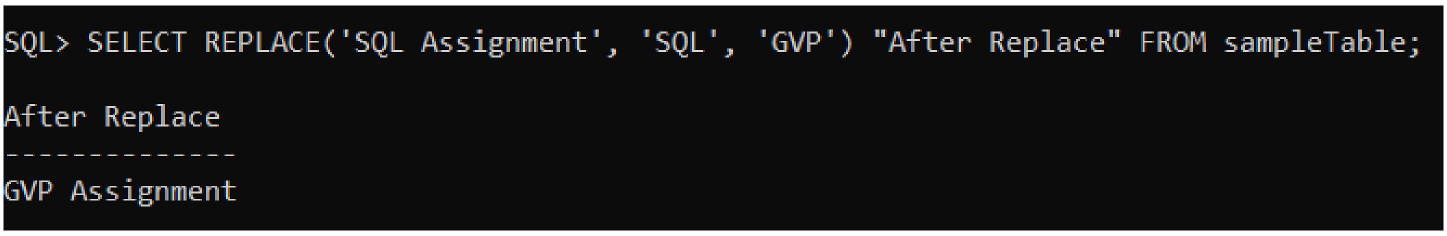
```
SQL> SELECT RTRIM('%!!!!!%GudiVaraprasad%!!!!%', '%%!') "RTRIM Example" FROM sampleTable;

RTRIM Example
---------------------
%!!!!!%GudiVaraprasad
```

6. translate(char, from, to)



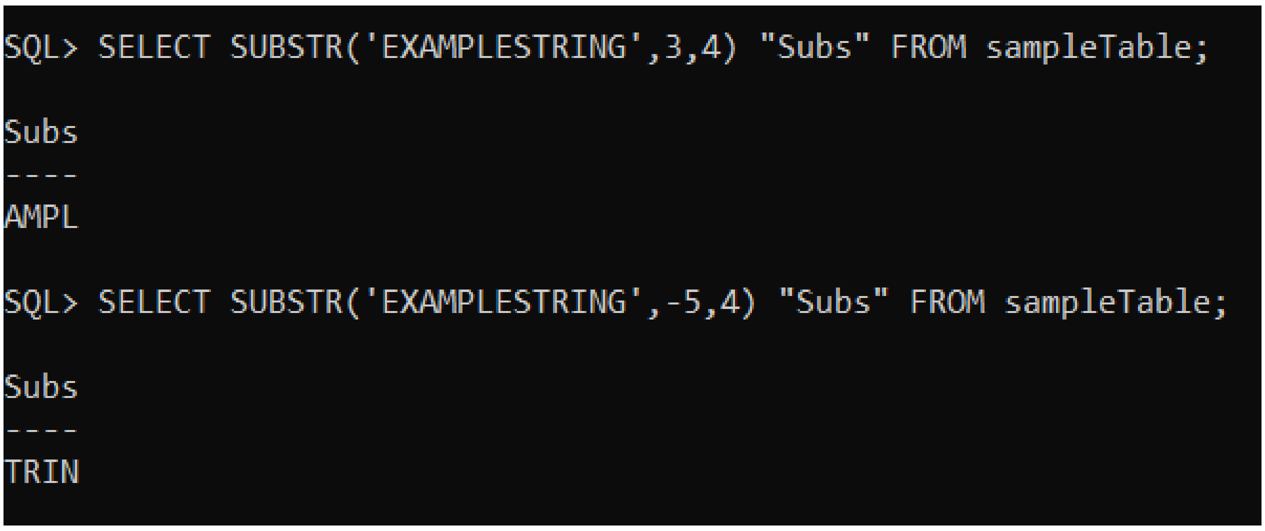
```
SQL> SELECT TRANSLATE('SQL*Plus User''s Guide', ' */''', '___') FROM sampleTable;

TRANSLATE('SQL*PLUSU
--------------------
SQL_Plus_Users_Guide
```

7. replace(char, search string, replace string)



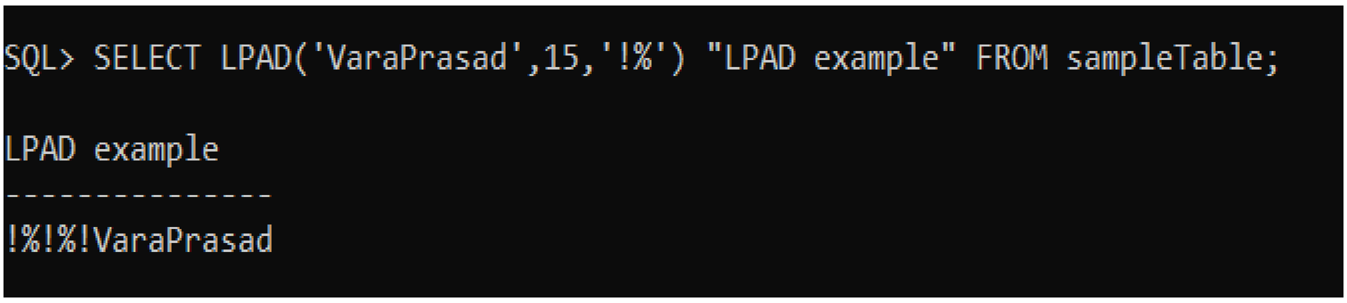
```
SQL> SELECT REPLACE('SQL Assignment', 'SQL', 'GVP') "After Replace" FROM sampleTable;

After Replace
--------------
GVP Assignment
```

8. substr(char, m , n)



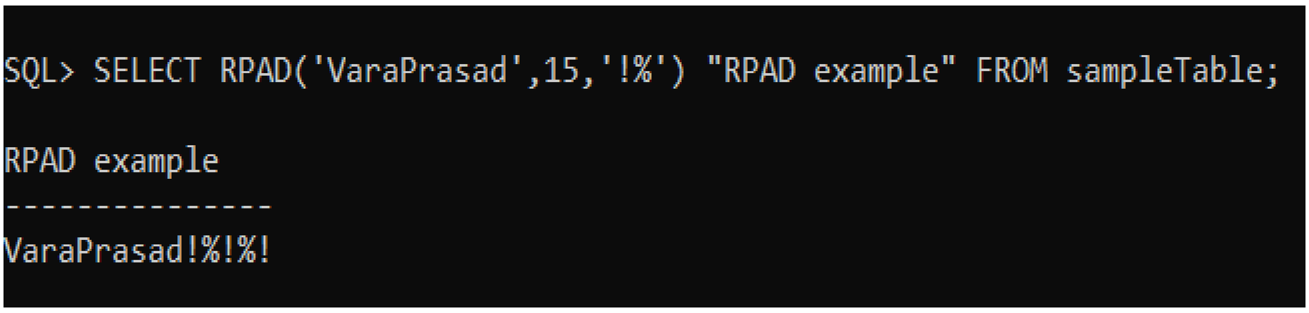
```
SQL> SELECT SUBSTR('EXAMPLESTRING',3,4) "Subs" FROM sampleTable;

Subs
----
AMPL

SQL> SELECT SUBSTR('EXAMPLESTRING',-5,4) "Subs" FROM sampleTable;

Subs
----
TRIN
```

9. lpad(char, length, special char)



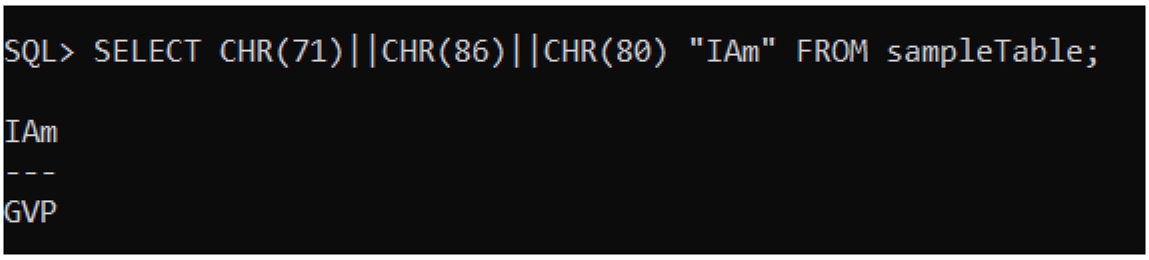
```
SQL> SELECT LPAD('VaraPrasad',15,'!%') "LPAD example" FROM sampleTable;

LPAD example
---------------
!%!%!VaraPrasad
```

10. rpad(char, length, special char)



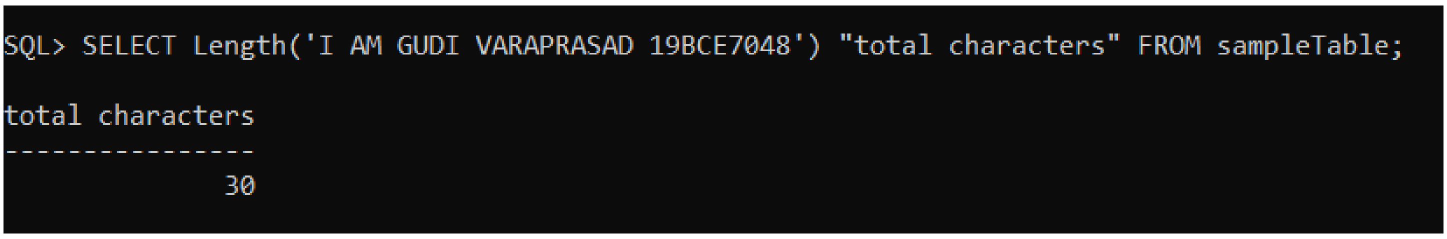
```
SQL> SELECT RPAD('VaraPrasad',15,'!%') "RPAD example" FROM sampleTable;

RPAD example
---------------
VaraPrasad!%!%!
```

11. chr(number)



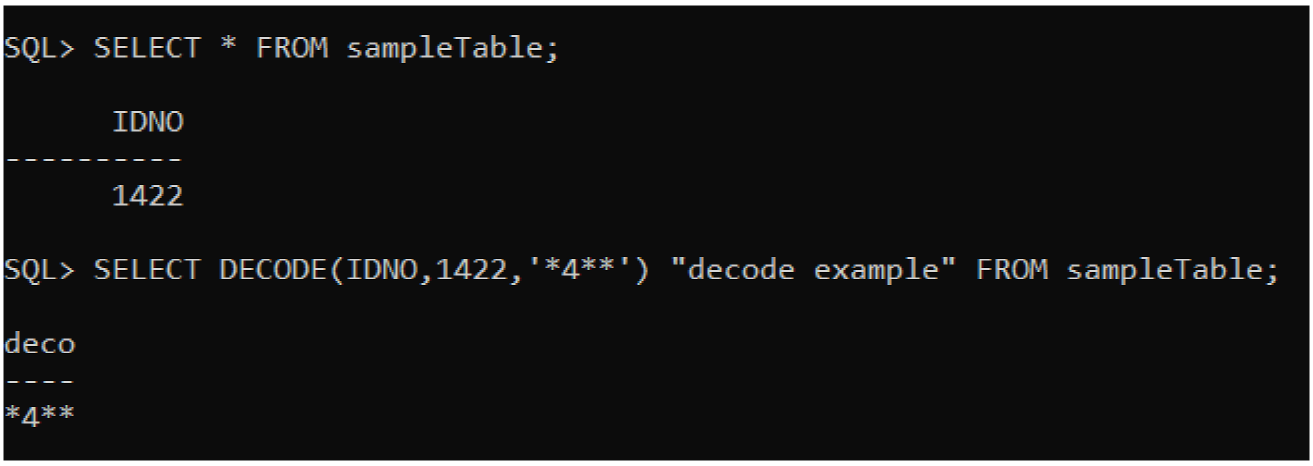
```
SQL> SELECT CHR(71)||CHR(86)||CHR(80) "IAm" FROM sampleTable;

IAm
---
GVP
```

12. length(char)



```
SQL> SELECT Length('I AM GUDI VARAPRASAD 19BCE7048') "total characters" FROM sampleTable;

total characters
----------------
              30
```

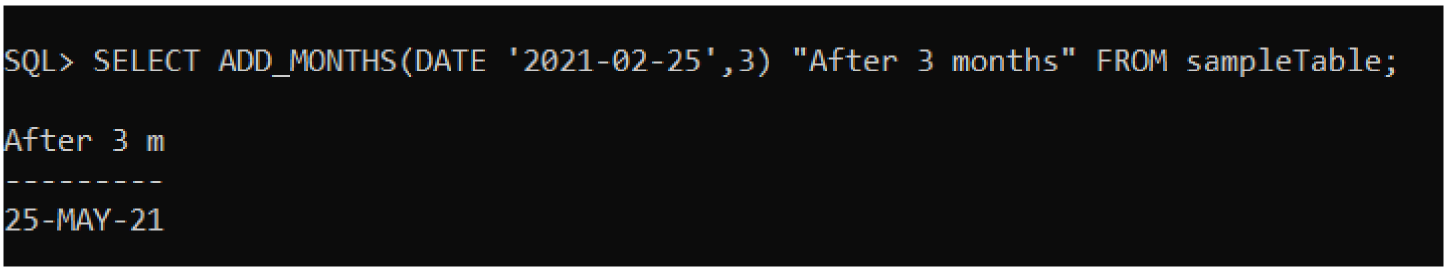
13. decode(column name, col value, replace value)



```
SQL> SELECT * FROM sampleTable;

      IDNO
----------
      1422

SQL> SELECT DECODE(IDNO,1422,'*4**') "decode example" FROM sampleTable;

deco
----
*4**
```

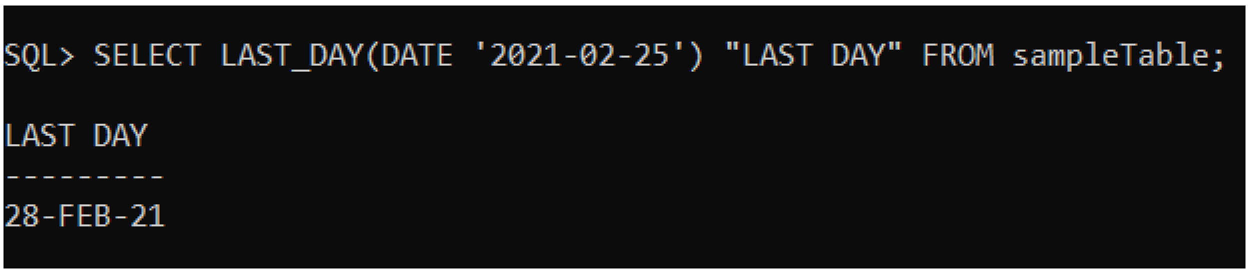
Date functions

1. add_months(date,no. of months)



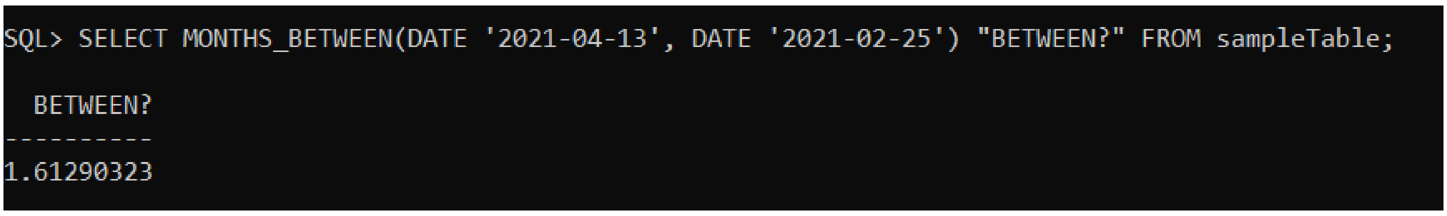
```
SQL> SELECT ADD_MONTHS(DATE '2021-02-25',3) "After 3 months" FROM sampleTable;

After 3 m
---------
25-MAY-21
```

2. last_day(date)



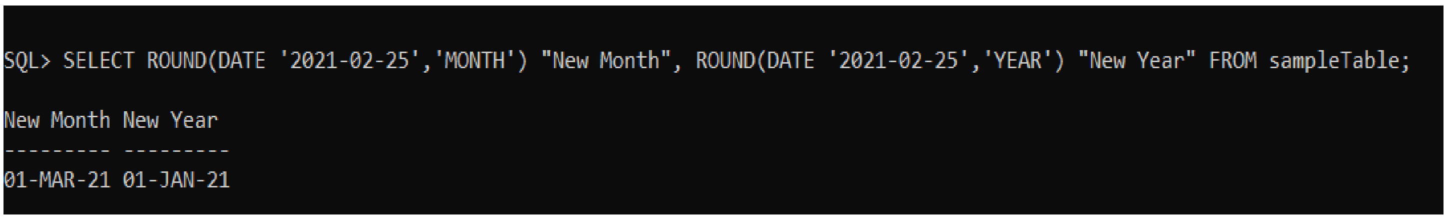
```
SQL> SELECT LAST_DAY(DATE '2021-02-25') "LAST DAY" FROM sampleTable;

LAST DAY
---------
28-FEB-21
```

3. months_between(date1,date2)



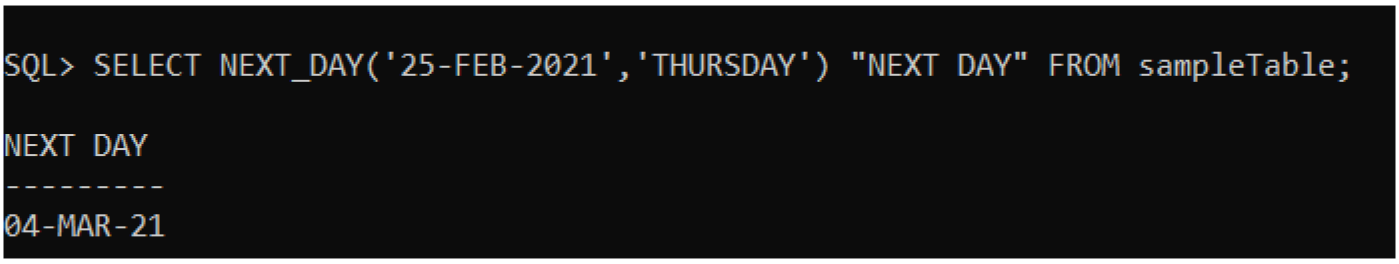
```
SQL> SELECT MONTHS_BETWEEN(DATE '2021-04-13', DATE '2021-02-25') "BETWEEN?" FROM sampleTable;

  BETWEEN?
----------
1.61290323
```

4. round(date)



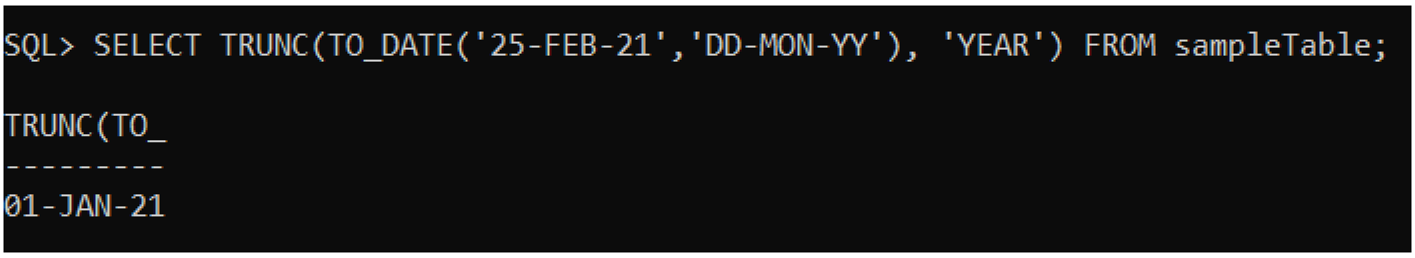
```
SQL> SELECT ROUND(DATE '2021-02-25','MONTH') "New Month", ROUND(DATE '2021-02-25','YEAR') "New Year" FROM sampleTable;

New Month New Year
--------- ---------
01-MAR-21 01-JAN-21
```

5. next_day(date, day)



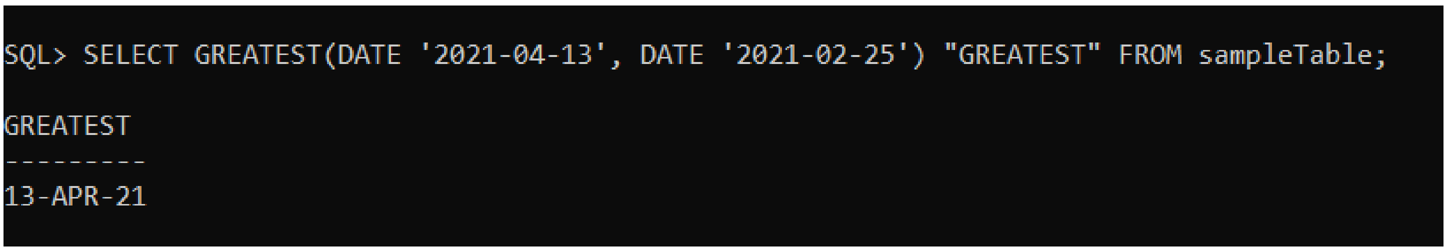
```
SQL> SELECT NEXT_DAY('25-FEB-2021','THURSDAY') "NEXT DAY" FROM sampleTable;

NEXT DAY
---------
04-MAR-21
```

6. trunc(date, [format])



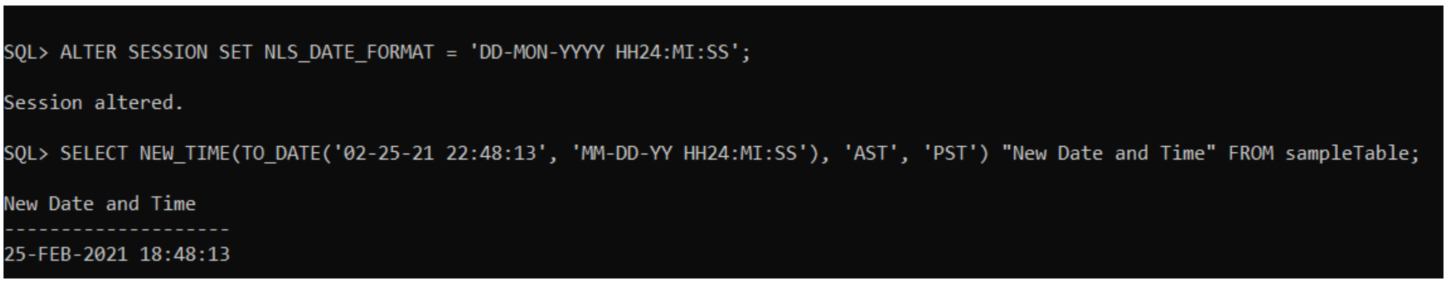
```
SQL> SELECT TRUNC(TO_DATE('25-FEB-21','DD-MON-YY'), 'YEAR') FROM sampleTable;

TRUNC(TO_
---------
01-JAN-21
```

7. greatest(date1, date2,…)



```
SQL> SELECT GREATEST(DATE '2021-04-13', DATE '2021-02-25') "GREATEST" FROM sampleTable;

GREATEST
---------
13-APR-21
```

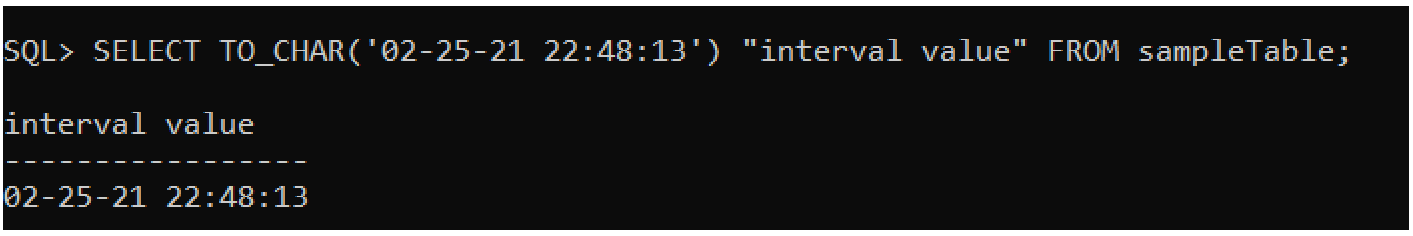
8. new_time(date, ‘this’, ‘other’)



```
SQL> ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MON-YYYY HH24:MI:SS';

Session altered.

SQL> SELECT NEW_TIME(TO_DATE('02-25-21 22:48:13', 'MM-DD-YY HH24:MI:SS'), 'AST', 'PST') "New Date and Time" FROM sampleTable;

New Date and Time
--------------------
25-FEB-2021 18:48:13
```

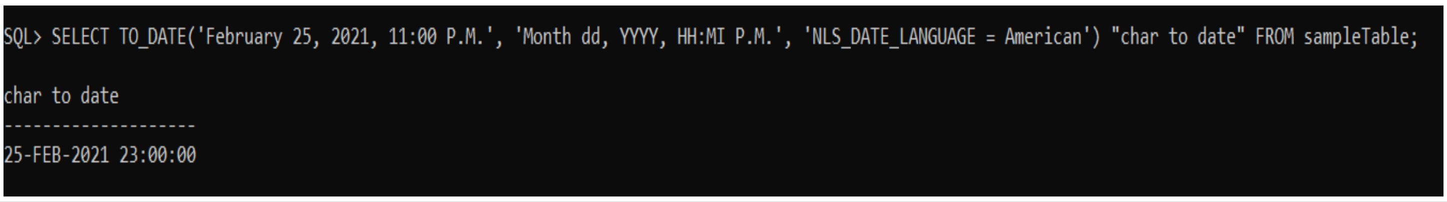
Conversion functions

1. to_char(date, format)



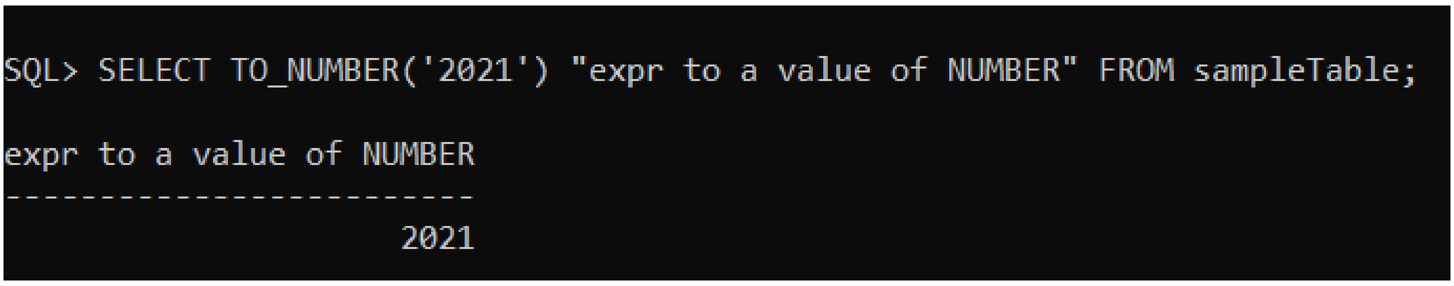
```
SQL> SELECT TO_CHAR('02-25-21 22:48:13') "interval value" FROM sampleTable;

interval value
-----------------
02-25-21 22:48:13
```

2. to_date(char,format)



```
SQL> SELECT TO_DATE('February 25, 2021, 11:00 P.M.', 'Month dd, YYYY, HH:MI P.M.', 'NLS_DATE_LANGUAGE = American') "char to date" FROM sampleTable;

char to date
------------------
25-FEB-2021 23:00:00
```

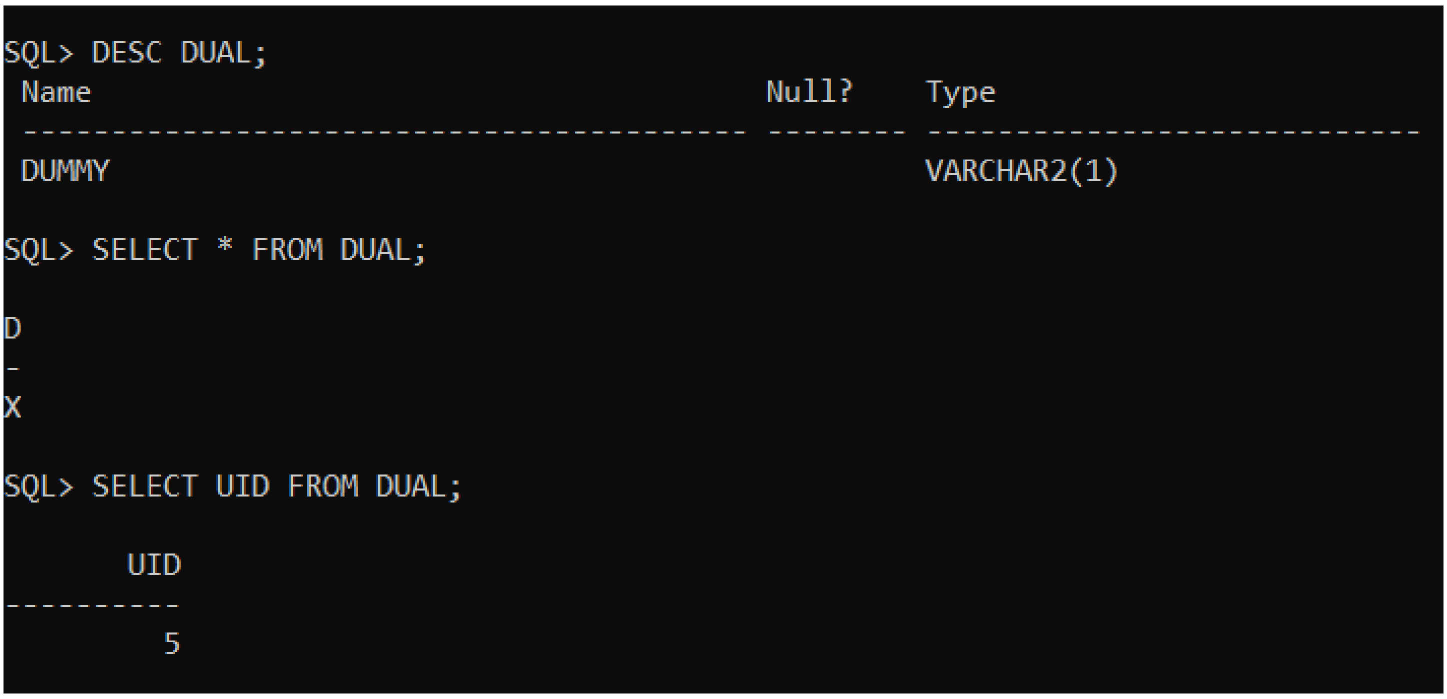
3. to_number(char)



```
SQL> SELECT TO_NUMBER('2021') "expr to a value of NUMBER" FROM sampleTable;

expr to a value of NUMBER
-------------------------
                     2021
```

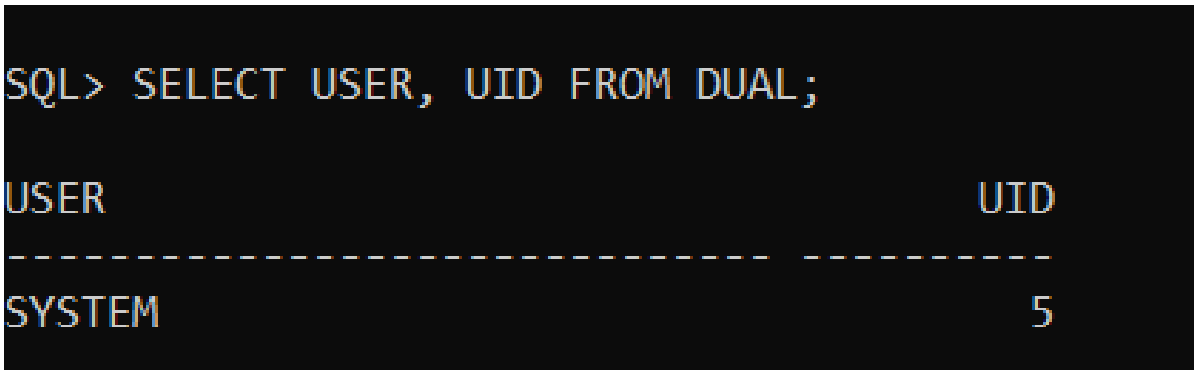
V. Miscellaneous functions :

1. uid



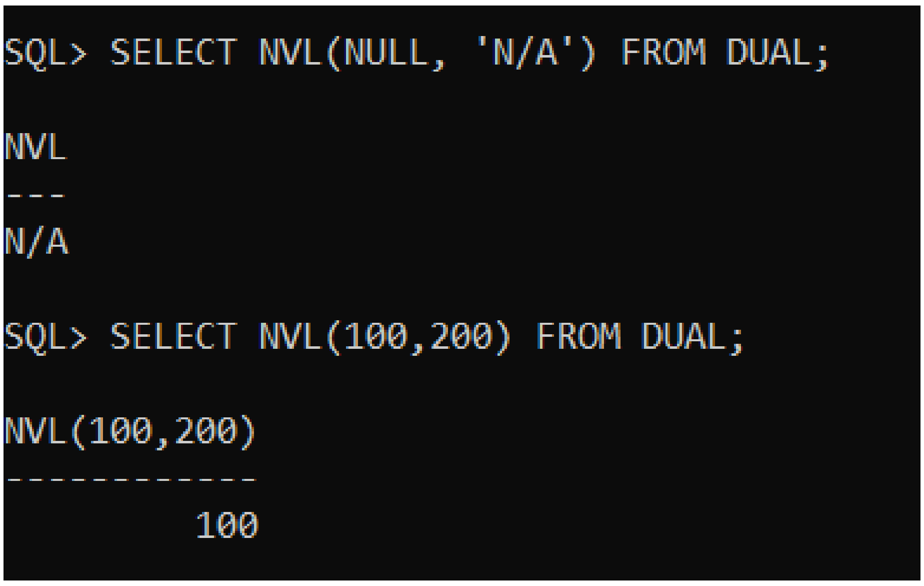
```
SQL> DESC DUAL;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 DUMMY                                              VARCHAR2(1)

SQL> SELECT * FROM DUAL;

D
-
X

SQL> SELECT UID FROM DUAL;

       UID
----------
         5
```

2. user



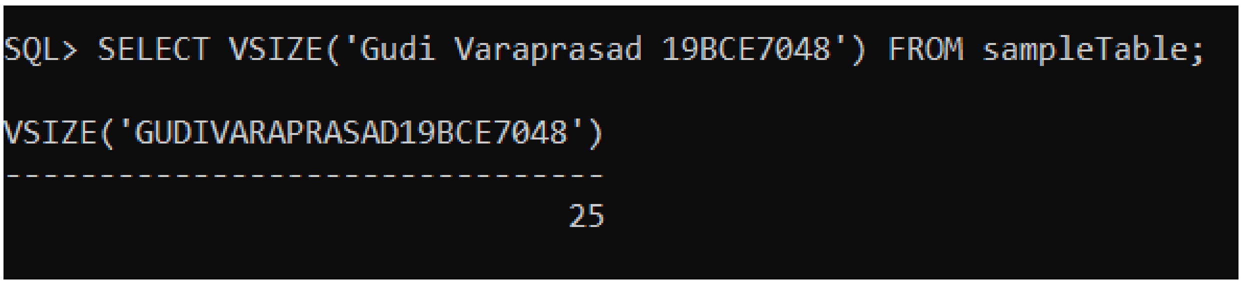
```
SQL> SELECT USER, UID FROM DUAL;

USER                                  UID
------------------------------ ----------
SYSTEM                                  5
```

3. nvl (column name, new value)



```
SQL> SELECT NVL(NULL, 'N/A') FROM DUAL;

NVL
---
N/A

SQL> SELECT NVL(100,200) FROM DUAL;

NVL(100,200)
------------
         100
```

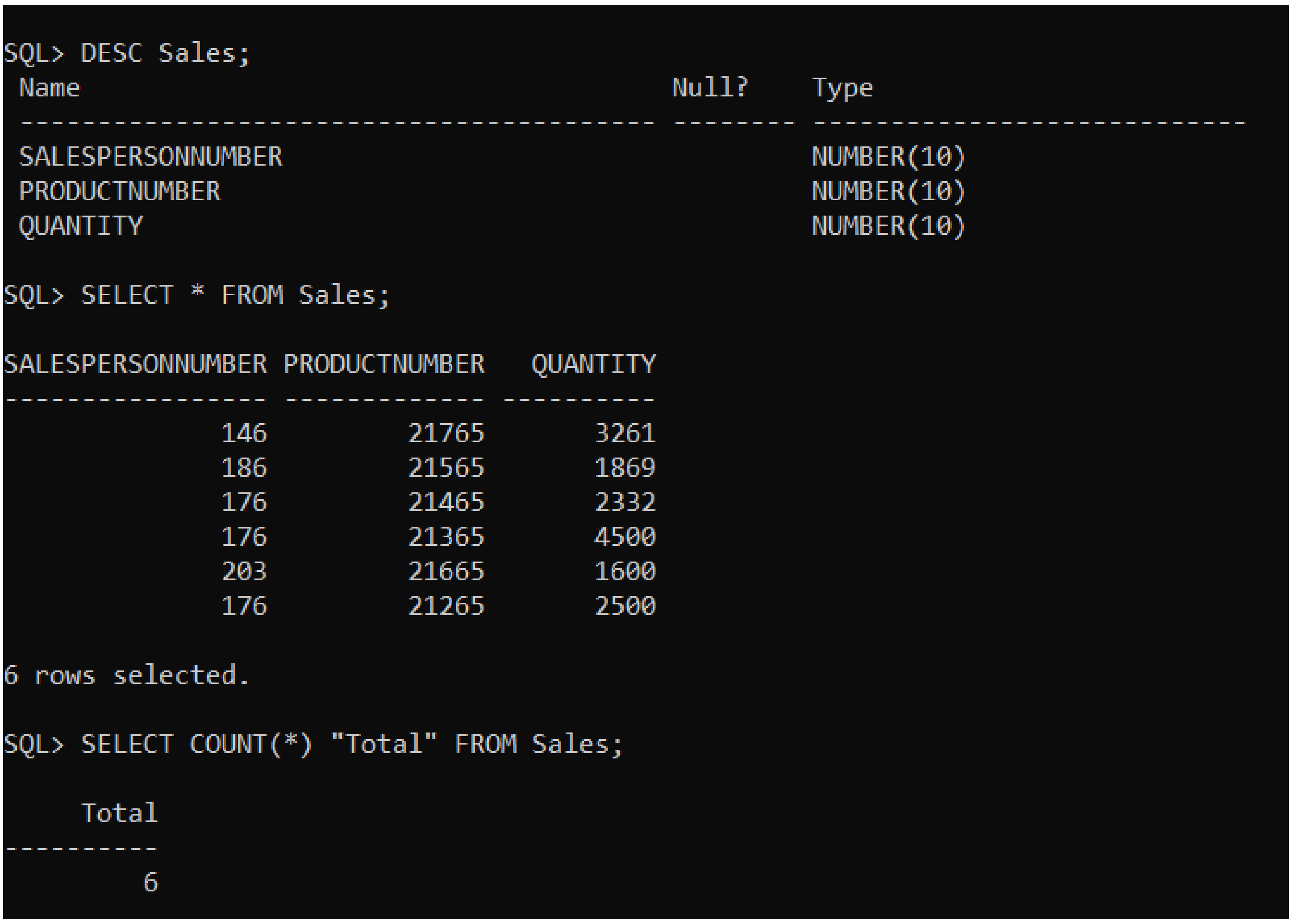
4. vsize(value)



```
SQL> SELECT VSIZE('Gudi Varaprasad 19BCE7048') FROM sampleTable;

VSIZE('GUDIVARAPRASAD19BCE7048')
--------------------------------
                              25
```

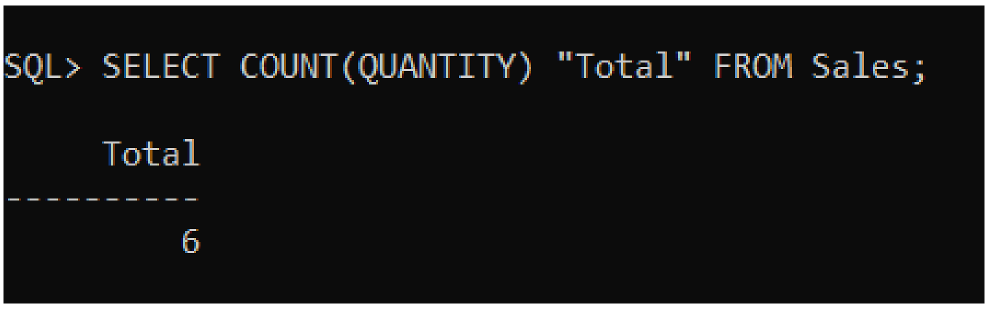
Group functions

1. count (*)



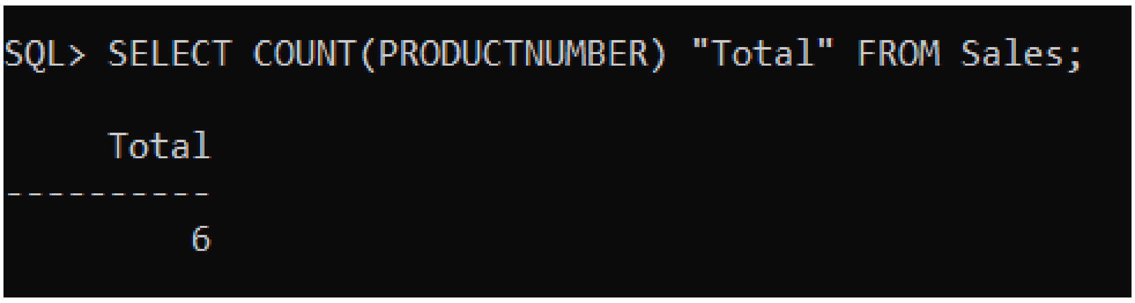
```
SQL> DESC Sales;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SALESPERSONNUMBER                                  NUMBER(10)
 PRODUCTNUMBER                                      NUMBER(10)
 QUANTITY                                           NUMBER(10)

SQL> SELECT * FROM Sales;

SALESPERSONNUMBER PRODUCTNUMBER   QUANTITY
----------------- ------------- ----------
              146         21765       3261
              186         21565       1869
              176         21465       2332
              176         21365       4500
              203         21665       1600
              176         21265       2500

6 rows selected.

SQL> SELECT COUNT(*) "Total" FROM Sales;

     Total
----------
         6
```

2.count (column name)



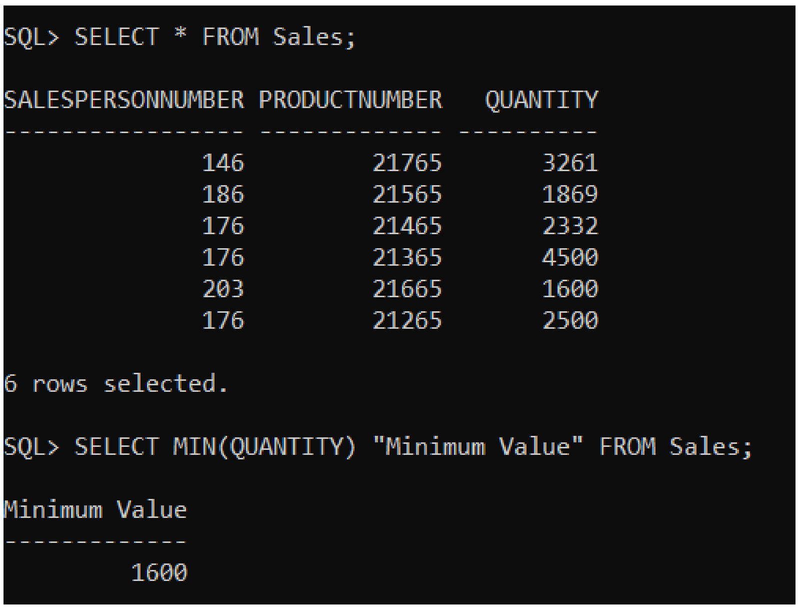
```
SQL> SELECT COUNT(QUANTITY) "Total" FROM Sales;

     Total
----------
         6
```

3. count (distinct column name)



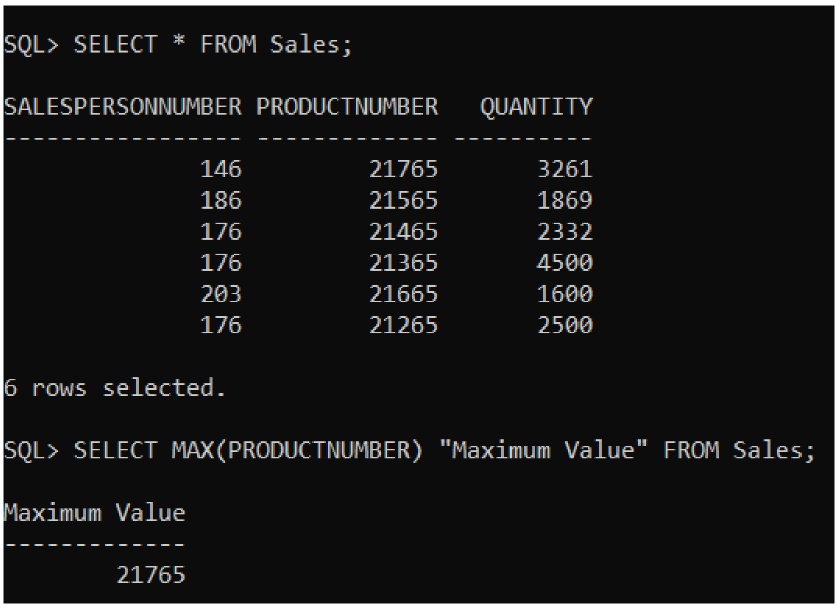
```
SQL> SELECT COUNT(PRODUCTNUMBER) "Total" FROM Sales;

     Total
----------
         6
```

4. min (column name)



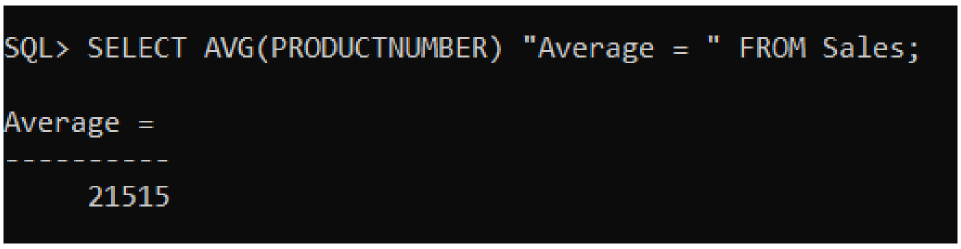
```
SQL> SELECT * FROM Sales;

SALESPERSONNUMBER PRODUCTNUMBER   QUANTITY
----------------- ------------- ----------
              146         21765       3261
              186         21565       1869
              176         21465       2332
              176         21365       4500
              203         21665       1600
              176         21265       2500

6 rows selected.

SQL> SELECT MIN(QUANTITY) "Minimum Value" FROM Sales;

Minimum Value
-------------
         1600
```

5. max (column name)



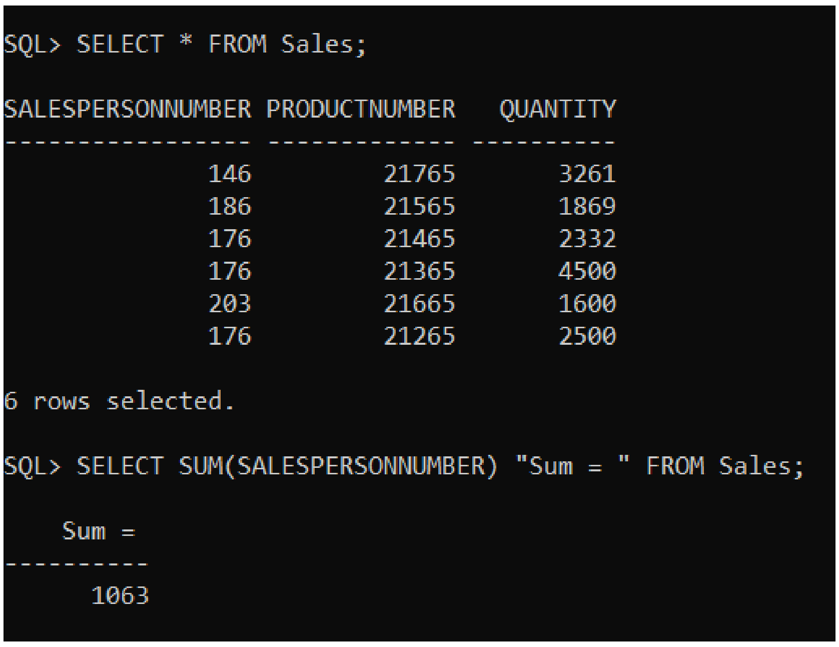
```
SQL> SELECT * FROM Sales;

SALESPERSONNUMBER PRODUCTNUMBER   QUANTITY
----------------- ------------- ----------
              146         21765       3261
              186         21565       1869
              176         21465       2332
              176         21365       4500
              203         21665       1600
              176         21265       2500

6 rows selected.

SQL> SELECT MAX(PRODUCTNUMBER) "Maximum Value" FROM Sales;

Maximum Value
-------------
        21765
```

6. avg (column name)



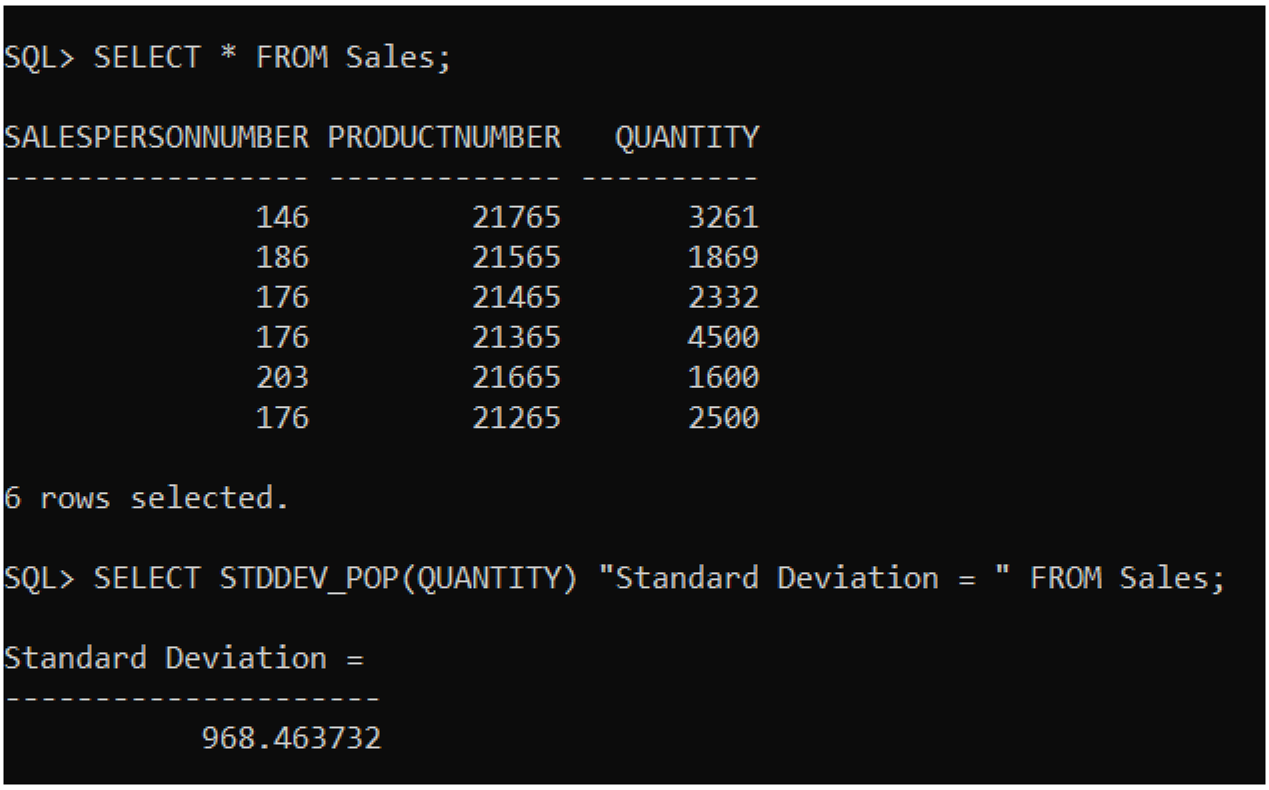
```
SQL> SELECT AVG(PRODUCTNUMBER) "Average = " FROM Sales;

Average =
----------
     21515
```

7. sum (column name)



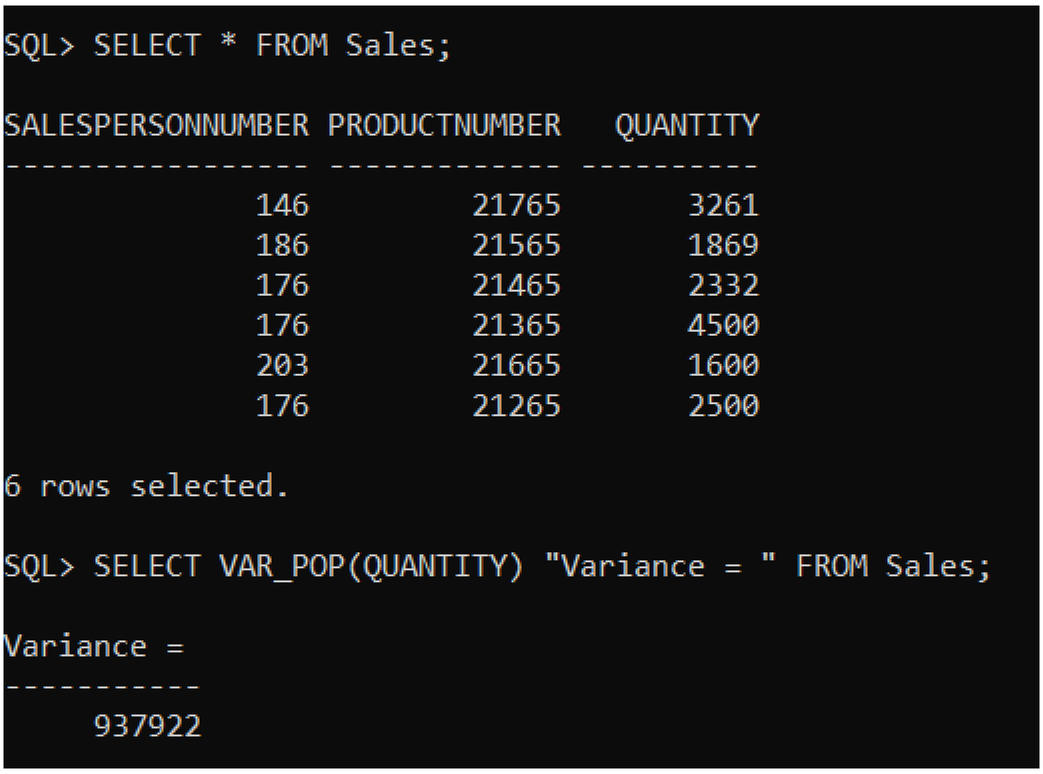
```
SQL> SELECT * FROM Sales;

SALESPERSONNUMBER PRODUCTNUMBER   QUANTITY
----------------- ------------- ----------
              146         21765       3261
              186         21565       1869
              176         21465       2332
              176         21365       4500
              203         21665       1600
              176         21265       2500

6 rows selected.

SQL> SELECT SUM(SALESPERSONNUMBER) "Sum = " FROM Sales;

    Sum =
----------
      1063
```

8. stddev_pop(column name)



```
SQL> SELECT * FROM Sales;

SALESPERSONNUMBER PRODUCTNUMBER   QUANTITY
----------------- ------------- ----------
              146         21765       3261
              186         21565       1869
              176         21465       2332
              176         21365       4500
              203         21665       1600
              176         21265       2500

6 rows selected.

SQL> SELECT STDDEV_POP(QUANTITY) "Standard Deviation = " FROM Sales;

Standard Deviation =
---------------------
           968.463732
```

9. var_pop(column name)



```
SQL> SELECT * FROM Sales;

SALESPERSONNUMBER PRODUCTNUMBER   QUANTITY
----------------- ------------- ----------
              146         21765       3261
              186         21565       1869
              176         21465       2332
              176         21365       4500
              203         21665       1600
              176         21265       2500

6 rows selected.

SQL> SELECT VAR_POP(QUANTITY) "Variance = " FROM Sales;

Variance =
-----------
     937922
```
