Functions can be categorized as follows

Built-in Functions

These are provided by the SQL language

- Single row functions
- Group functions

SINGLE ROW FUNCTIONS

Single row functions can be categorized into five. These will be applied for each row and produces individual output for each row.

- Numeric functions
- String functions
- Date functions
- Miscellaneous functions
- Conversion functions

Group functions

Group functions operate on **sets of rows** to return **a single summary value**.

Function	Description
COUNT()	Counts rows (or non-null values)
SUM()	Total sum of numeric values
AVG()	Average value
MAX()	Maximum value
MIN()	Minimum value

NUMERIC FUNCTIONS

- Abs
- Sign
- Sqrt
- Mod
- Nvl
- Power
- Exp
- Ln
- Log
- Ceil
- Floor
- Round
- Trunk
- Bitand
- Greatest
- Least
- Coalesce

NUMERIC FUNCTIONS

ABS

Absolute value is the measure of the magnitude of value.

Absolute value is always a positive number.

Syntax: abs (value)

Ex:

```
SQL> select abs(5), abs(-5), abs(0), abs(null) from dual;
```

NUMERIC FUNCTIONS

SIGN

Sign gives the sign of a value.

Syntax: sign (value)

Ex:

SQL> select sign(5), sign(-5), sign(0), sign(null) from dual;

NUMERIC FUNCTIONS

This will give the square root of the given value.

Syntax: sqrt (value) -- here value must be positive.

Ex:

```
SQL> select sqrt(4), sqrt(0), sqrt(null), sqrt(1) from dual
```

NUMERIC FUNCTIONS

MOD

This will give the remainder.

Syntax: mod (value, divisor)

Ex:

SQL> select mod(7,4), mod(1,5), mod(null,null), mod(0,0), mod(-7,4) from dual;

NUMERIC FUNCTIONS

NVL

This will substitutes the specified value in the place of null values.

Syntax: nvl (null_col, replacement_value)

Ex:

SQL> select * from student;

SQL> select nvl(1,2), nvl(2,3), nvl(4,3), nvl(5,4) from dual;

SQL> select nvl(0,0), nvl(1,1), nvl(null,null), nvl(4,4) from dual;

NUMERIC FUNCTIONS

POWER

Power is the ability to raise a value to a given exponent.

Syntax: power (value, exponent)

Ex:

SQL> select power(2,5), power(0,0), power(1,1), power(null,null), power(2,-5) from

dual;

NUMERIC FUNCTIONS

EXP

This will raise e value to the give power.

34

Syntax: exp (value)

Ex:

SQL> select exp(1), exp(2), exp(0), exp(null), exp(-2) from dual;

NUMERIC FUNCTIONS

LN

This is based on natural or base e logarithm.

Syntax: ln (value) -- here value must be greater than zero which is positive only.

Ex:

SQL> select ln(1), ln(2), ln(null) from dual;

NUMERIC FUNCTIONS

LOG

This is based on 10 based logarithm.

Syntax: log (10, value) -- here value must be greater than zero which is positive only.

Ex:

SQL> select log(10,100), log(10,2), log(10,1), log(10,null) from dual;

SQL> select ln(3), log(exp(1),3) from dual;

NUMERIC FUNCTIONS

CEIL

This will produce a whole number that is greater than or equal to the specified value.

Syntax: ceil (value)

Ex:

SQL> select ceil(5), ceil(5.1), ceil(-5), ceil(-5.1), ceil(0), ceil(null) from dual;

NUMERIC FUNCTIONS

FLOOR

This will produce a whole number that is less than or equal to the specified value.

Syntax: floor (value)

Ex:

SQL> select floor(5), floor(5.1), floor(-5), floor(-5.1), floor(0), floor(null) from dual;

NUMERIC FUNCTIONS

ROUND

This will rounds numbers to a given number of digits of precision.

Syntax: round (value, precision)

Ex:

SQL> select round(123.2345), round(123.2345,2), round(123.2354,2) from dual;

SQL> select round(123.2345,-1), round(123.2345,-2), round(123.2345,-3),

round(123.2345,-4) from dual;

SQL> select round(123,0), round(123,1), round(123,2) from dual;

select round(123,-1), round(123,-2), round(123,-3), round(-123,-1), round(-123,-

2), round(-123,-3) from dual;

NUMERIC FUNCTIONS

TRUNC

This will truncates or chops off digits of precision from a number.

Syntax: trunc (value, precision)

Ex:

SQL> select trunc(123.2345), trunc(123.2345,2), trunc(123.2354,2) from dual;

SQL> select trunc(123.2345,-1), trunc(123.2345,-2), trunc(123.2345,-3),

trunc(123.2345,-4) from dual;

SQL> select trunc(123,0), trunc(123,1), trunc(123,2) from dual;

NUMERIC FUNCTIONS

BITAND

This will perform bitwise and operation.

Syntax: bitand (value1, value2)

Ex:

SQL> select bitand(2,3), bitand(0,0), bitand(1,1), bitand(null,null), bitand(-2,-3) from

dual;

NUMERIC FUNCTIONS

GREATEST

This will give the greatest number.

Syntax: greatest (value1, value2, value3 … valuen)

Ex:

SQL> select greatest(1, 2, 3), greatest(-1, -2, -3) from dual;

NUMERIC FUNCTIONS

LEAST

This will give the least number.

Syntax: least (value1, value2, value3 … valuen)

Ex:

SQL> select least(1, 2, 3), least(-1, -2, -3) from dual;

NUMERIC FUNCTIONS

COALESCE

This will return first non-null value.

Syntax: coalesce (value1, value2, value3 … valuen)

Ex:

SQL> select coalesce(1,2,3), coalesce(null,2,null,5) from dual;

STRING FUNCTIONS

- STRING FUNCTIONS
- Initcap
- Upper
- Lower
- Length
- Rpad
- Lpad
- Ltrim
- Rtrim
- Trim
- Translate
- Replace
- Soundex
- Concat (‘ || ‘ Concatenation operator)
- Ascii
- Chr
- Substr
- Instr
- Decode
- Greatest
- Least
- Coalesce

STRING FUNCTIONS

INITCAP

This will capitalize the initial letter of the string.

Syntax: initcap (string)

Ex:

SQL> select initcap('computer') from dual;

STRING FUNCTIONS

UPPER

This will convert the string into uppercase.

Syntax: upper (string)

Ex:

SQL> select upper('computer') from dual;

STRING FUNCTIONS

LOWER

This will convert the string into lowercase.

Syntax: lower (string)

Ex:

SQL> select lower('COMPUTER') from dual;

STRING FUNCTIONS

LENGTH

This will give length of the string.

Syntax: length (string)

Ex:

SQL> select length('computer') from dual;

STRING FUNCTIONS

RPAD

This will allows you to pad the right side of a column with any set of characters.

Syntax: rpad (string, length [, padding_char])

Ex:

SQL> select rpad('computer',15,'*'), rpad('computer',15,'*#') from dual;

STRING FUNCTIONS

LPAD

This will allows you to pad the left side of a column with any set of characters.

43

Syntax: lpad (string, length [, padding_char])

Ex:

SQL> select lpad('computer',15,'*'), lpad('computer',15,'*#') from dual;

STRING FUNCTIONS

LTRIM

This will trim off unwanted characters from the left end of string.

Syntax: ltrim (string [,unwanted_chars])

Ex:

SQL> select ltrim('computer','co'), ltrim('computer','com') from dual;

STRING FUNCTIONS

RTRIM

This will trim off unwanted characters from the right end of string.

44

Syntax: rtrim (string [, unwanted_chars])

Ex:

SQL> select rtrim('computer','er'), rtrim('computer','ter') from dual;

STRING FUNCTIONS

TRIM

This will trim off unwanted characters from the both sides of string.

Syntax: trim (unwanted_chars from string)

Ex:

SQL> select trim('i' from 'indiani') from dual;

select trim(leading'i' from 'indiani') from dual; -- this will work as LTRIM

SQL> select trim(trailing'i' from 'indiani') from dual; -- this will work as RTRIM

STRING FUNCTIONS

TRANSLATE

This will replace the set of characters, character by character.

Syntax: translate (string, old_chars, new_chars)

Ex:

SQL> select translate('india','in','xy') from dual;

STRING FUNCTIONS

REPLACE

This will replace the set of characters, string by string.

Syntax: replace (string, old_chars [, new_chars])

Ex:

SQL> select replace('india','in','xy'), replace(‘india’,’in’) from dual;

STRING FUNCTIONS

SOUNDEX

This will be used to find words that sound like other words, exclusively used in where clause.

Syntax: soundex (string)

Ex:

SQL> select * from emp where soundex(ename) = soundex('SMIT');

STRING FUNCTIONS

CONCAT

This will be used to combine two strings only.

Syntax: concat (string1, string2)

Ex:

SQL> select concat('computer',' operator') from dual;

If you want to combine more than two strings you have to use concatenation operator (||).

SQL> select 'how' || ' are' || ' you' from dual;

STRING FUNCTIONS

ASCII

This will return the decimal representation in the database character set of the first character of the string.

Syntax: ascii (string)

Ex:

SQL> select ascii('a'), ascii('apple') from dual;

STRING FUNCTIONS

CHR

This will return the character having the binary equivalent to the string in either the database character set or the national character set.

Syntax: chr (number)

Ex:

SQL> select chr(97) from dual;

STRING FUNCTIONS

SUBSTR

This will be used to extract substrings.

Syntax: substr (string, start_chr_count [, no_of_chars])

Ex:

SQL> select substr('computer',2), substr('computer',2,5), substr('computer',3,7) from

dual;

SUBSTR(SUBST

STRING FUNCTIONS

INSTR

This will allows you for searching through a string for set of characters.

Syntax: instr (string, search_str [, start_chr_count [, occurrence]])

Ex:

SQL> select instr('information','o',4,1), instr('information','o',4,2) from dual;

STRING FUNCTIONS

DECODE

Decode will act as value by value substitution.

For every value of field, it will checks for a match in a series of if/then tests.

Syntax: decode (value, if1, then1, if2, then2, ……. else);

Ex:

SQL> select sal, decode(sal,500,'Low',5000,'High','Medium') from emp;

STRING FUNCTIONS

GREATEST

This will give the greatest string.

Syntax: greatest (strng1, string2, string3 … stringn)

Ex:

SQL> select greatest('a', 'b', 'c'), greatest('satish','srinu','saketh') from dual;

STRING FUNCTIONS

LEAST

This will give the least string.

Syntax: greatest (strng1, string2, string3 … stringn)

Ex:

SQL> select least('a', 'b', 'c'), least('satish','srinu','saketh') from dual;

STRING FUNCTIONS

COALESCE

This will gives the first non-null string.

Syntax: coalesce (strng1, string2, string3 … stringn)

Ex:

SQL> select coalesce('a','b','c'), coalesce(null,'a',null,'b') from dual;

DATE FUNCTIONS

- Sysdate
- Current_date
- Current_timestamp
- Systimestamp
- Localtimestamp
- Dbtimezone
- Sessiontimezone
- To_char
- To_date
- Add_months
- Months_between
- Next_day
- Last_day
- Extract

- Greatest
- Least
- Round
- Trunc
- New_time
- Coalesce

DATE FUNCTIONS

Oracle default date format is DD-MON-YY.

We can change the default format to our desired format by using the following command.

SQL> alter session set nls_date_format = ‘DD-MONTH-YYYY’;

But this will expire once the session was closed.

DATE FUNCTIONS

SYSDATE

This will give the current date and time.

Ex:

SQL> select sysdate from dual;

CURRENT_DATE

This will returns the current date in the session’s timezone.

Ex:

SQL> select current_date from dual;

DATE FUNCTIONS

CURRENT_TIMESTAMP

This will returns the current timestamp with the active time zone information.

Ex:

SQL> select current_timestamp from dual;

SYSTIMESTAMP

This will returns the system date, including fractional seconds and time zone of the database.

Ex:

SQL> select systimestamp from dual;

DATE FUNCTIONS

TO_CHAR

This will be used to extract various date formats.

The available date formats as follows.

Syntax: to_char (date, format)

DATE FUNCTIONS

DATE FORMATS
D -- No of days in week
DD -- No of days in month
DDD -- No of days in year
MM -- No of month
MON -- Three letter abbreviation of month
MONTH -- Fully spelled out month
RM -- Roman numeral month
DY -- Three letter abbreviated day
DAY -- Fully spelled out day
Y -- Last one digit of the year
YY -- Last two digits of the year
IW -- No of weeks in year from ISO standard
HH -- Hours
MI -- Minutes

YYY -- Last three digits of the year
YYYY -- Full four digit year
SYYYY -- Signed year
I -- One digit year from ISO standard
IY -- Two digit year from ISO standard
IYY -- Three digit year from ISO standard
IYYY -- Four digit year from ISO standard
Y, YYY -- Year with comma
YEAR -- Fully spelled out year
CC -- Century
Q -- No of quarters
W -- No of weeks in month
WW -- No of weeks in year

DATE FUNCTIONS

Ex:

SQL> select to_char(sysdate,'dd month yyyy hh:mi:ss am dy') from dual;