SINGLE ROW FUNCTIONS

1. LENGTH( )
2. CONCAT( )
3. UPPER( )
4. LOWER( )
5. INITCAP( )
6. REVERSE( )
7. SUSBTR( )
8. INSTR( )
9. REPLACE( )
10. MOD( )
11. TRUNC( )
12. ROUND( )
13. MONTHS_BETWEEN( )
14. LAST_DAY( )
15. TO_CHAR( )
16. NVL( )

1. LENGTH : "It is used to count the number of characters present In the given string ".

SYNTAX: **LENGTH ('string')**

Example :

- WAQT count number of characters present in 'SMITH' .

```
SELECT LENGTH ( ENAME )
FROM EMP
WHERE ENAME ='SMITH' ;
```

LENGTH(ENAME)
5

```
SELECT LENGTH( 'SMITH' )
FROM DUAL ;

SELECT LENGTH( 'HELLO WORLD' ) ⟶11
FROM DUAL;
```

NOTE : DUAL TABLE

It is a DUMMY table which has 1 col and 1 row . Which is used to output the result .

- DESC DUAL ;
- SELECT *
  FROM DUAL ;

2. **CONCAT( ) :** "It is used to join the given two strings '

SYNTAX : CONCAT ('string1' , 'String2')

Example :

Input : Smith
Output : Mr. Smith

```
SELECT CONCAT( 'Mr. ' , ENAME )
FROM EMP
WHERE ENAME ='SMITH' ;
```

3. **UPPER( )** : "It is used to convert a given string to upper case "

SYNTAX: UPPER ('string')

4. **LOWER( ) :**"It is used to convert a given string to lower case "

SYNTAX: LOWER('string')

5. **INITCAP( ):**"It is used to convert a given string to initial capital letter case ".

SYNTAX: INITCAP('string')

6. **REVERSE( ):** "It is used to reverse a given string ".

SYNTAX: REVERSE('string')

Example :

```
SELECT REVERSE( 'SMITH' ).
FROM DUAL ;
```

REVERSE('SMITH')
HTIMS

```
SELECT UPPER( 'smith' ).
FROM DUAL ;
```

UPPER('smith')
SMITH

```
SELECT LOWER( 'SMITH' ).
FROM DUAL  ;
```

LOWER('SMITH')
smith

```
SELECT INITCAP( 'SMITH' ).
FROM DUAL ;
```

INITCAP('SMITH')
Smith

7. **SUBSTR :** It is used to extract a part of string from the given Original string.

SYNTAX: **SUBSTR** ('Original_String' , Position [, Length])

Example:

-ve

-7	-6	-5	-4	-3	-2	-1
Q	**S**	**P**	**I**	**D**	**E**	**R**
1	2	3	4	5	6	7

+ve

Example :	SUBSTR('QSPIDER' , 2 , 3)	SPI
Example :	SUBSTR('QSPIDER' , 3 , 3)	PID
Example :	SUBSTR('QSPIDER' , 2)	SPIDER
Example :	SUBSTR('QSPIDER' , 1 , 6)	QSPIDE
Example :	SUBSTR('QSPIDER' , 4, 1)	I
Example :	SUBSTR('QSPIDER' , 1 , 1)	Q
Example :	SUBSTR('QSPIDER' , 7 , 1)	R
Example :	SUBSTR('QSPIDER' , 6)	ER
Example :	SUBSTR('QSPIDER' , 0 , 3)	QSP
Example :	SUBSTR('QSPIDER' , 6 , 6)	ER
Example :	SUBSTR('QSPIDER' , -2 , 1)	E
Example :	SUBSTR('QSPIDER' , -5 , 3)	PID
Example :	SUBSTR('QSPIDER' , -7 , 2)	QS
Example :	SUBSTR('QSPIDER' , -1)	R

- WAQT extract first 3 characters of the emp names .

```
SELECT SUBSTR(ENAME, 1,3)
FROM EMP;
```

- WAQT extract last 3 characters of the employee names .

```
SELECT SUBSTR(ENAME, -3 )
FROM EMP;
```

- WAQT to display **first half of employee names .**

ENAME	OUTPUT
SMITH	SM
MILLER	MIL
JONES	JO
WARD	WA

```
SELECT SUBSTR( ENAME , 1 , LENGTH( ENAME ) / 2 )
FROM EMP ;
```

EXAMPLE:

SMITH	SUBSTR(ENAME , 1 , **LENGTH(ENAME) / 2**)

	SUBSTR('SMITH' , 1 , LENGTH ('SMITH') / 2)
	SUBSTR('SMITH' , 1 , 5 / 2)
	SUBSTR('SMITH' , 1 , 2)
	OUTPUT: SM

WARD	SUBSTR(ENAME , 1 , LENGTH(ENAME) / 2)
	SUBSTR('WARD' , 1 , LENGTH ('WARD') / 2)
	SUBSTR('WARD' , 1 , 4 / 2)
	SUBSTR('WARD' , 1 , 2)
	OUTPUT: WA

- WAQT to display **second half of employee names .**

ENAME	OUTPUT
SMITH	ITH
MILLER	LER
JONES	NES
WARD	RD

SELECT SUBSTR(ENAME , LENGTH(ENAME) / 2 + 1)
FROM EMP ;

SMITH	SUBSTR(ENAME , LENGTH(ENAME) / 2 +1)
	SUBSTR('SMITH' , LENGTH ('SMITH') / 2 +1)
	SUBSTR('SMITH' , 5 / 2 +1)
	SUBSTR('SMITH' , 3)
	ITH

WARD	SUBSTR(ENAME , LENGTH(ENAME) / 2+1)
	SUBSTR('WARD' , LENGTH ('WARD') / 2+1)
	SUBSTR('WARD' , 4 / 2 +1)
	SUBSTR('WARD' , 3)
	RD

8. **REPLACE ( ) :** It is used to replace a string with another string in The original string.

Null

SYNTAX:**REPLACE** ('Original_String' , 'string' [, 'new_String'])

Example :	REPLACE ('BANANA' , 'A' , 'C')	BCNCNC
Example :	REPLACE ('BANANA' , 'N' , 'ABC')	BAABCAABCA
Example :	REPLACE ('OPPO' , 'O' , 'J')	JPPJ
Example :	REPLACE ('BANANA' , 'A')	BNN
Example :	REPLACE ('ENGINEERING' , 'E')	NGINRING
Example :	REPLACE ('ENGINEERING' , 'E' , '123')	123N123123GINRING

NOTE : if the third argument is not mentioned the default Value of it is Null .

1. WAQTD the number of times char 'A' is present in BANANA.

```
SELECT LENGTH('BANANA') - LENGTH ( REPLACE( 'BANANA','A' )
FROM DUAL ;
```

Length ('BANANA') - LENGTH(REPLACE('BANANA','A'))

Length ('BANANA') - LENGTH ('BNN')

=6 - 3

= 3 times 'A' is present in BANANA

2. WAQTD to count number of time 'A' is present in 'MALAYALAM'

```
SELECT LENGTH('MALAYALAM') - LENGTH( REPLACE( 'MALAYALAM','A' )
FROM DUAL ;
```

9. **INSTR ( ) :** "*it is used to obtain the position in which the string is present in the Original string* ". It is used to search for a string in the Original string if present it returns the POSITION Else it returns **0** ".

Syntax: **INSTR('Original_String' , 'String' , Position [, Occurrence])**

Note : *if occurrence is not Mentioned then , the default value of Occurrence is 1* .

B	A	N	A	N	A
1	2	3	4	5	6

Example : INSTR('BANANA' , 'A' , 1 , 1)	POS: 2
Example : INSTR('BANANA' , 'A' , 2 , 1)	POS: 2
Example : INSTR('BANANA' , 'A' , 1 , 2)	POS: 4
Example : INSTR('BANANA' , 'A' , 1 , 3)	POS: 6
Example : INSTR('BANANA' , 'A' , 1 , 4)	POS: 0
Example : INSTR('BANANA' , 'A' , 4 , 2)	POS: 6

Example : INSTR('BANANA' , 'A' , 2)	POS: 2
Example : INSTR('BANANA' , 'N' , 2 , 1)	POS: 3
Example : INSTR('BANANA' , 'O' , 1 ,1)	POS: 0
Example : INSTR('BANANA' , 'NA' , 2 , 2)	POS: 5
Example : INSTR('BANANA' , 'A' , 3 , 3)	POS: 0
Example : INSTR('BANANA' , 'ANA' , 1 , 2)	POS: 4

1. WAQTD NAMES OF THE EMPLOYEES IF THEY HAVE CHAR 'A' PRESENT IN THEIR NAMES

```
SELECT ENAME
FROM EMP
WHERE INSTR( ENAME , 'A' , 1 ,1 ) > 0 ;
```

2. WAQTD NAMES OF THE EMPLOYEES IF THEY HAVE CHAR 'A' PRESENT ATLEAST TWICE INTHEIR NAMES

```
SELECT ENAME
FROM EMP
WHERE INSTR( ENAME , 'A' , 1 , 2 ) > 0 ;
```

3. WAQTD NAMES OF THE EMPLOYEES IF THEY HAVE CHAR 'A' PRESENT ATLEAST THRICE INTHEIR NAMES

```
SELECT ENAME
FROM EMP
WHERE INSTR( ENAME , 'A' , 1 , 3 ) > 0 ;
```

4. WAQTD NAMES OF THE EMPLOYEES IF THEY HAVE CHAR 'A' ***EXACTLY TWICE***

```
SELECT ENAME
FROM EMP
WHERE INSTR( ENAME , 'A' , 1 , 2 ) > 0 AND  INSTR( ENAME , 'A' , 1 , 3 ) = 0  ;

OR

SELECT ENAME
FROM EMP
WHERE (  LENGTH( ENAME ) - LENGTH( REPLACE( ENAME ,'A') )  ) = 2;
```

ALLEN	INSTR('ALLEN','A',1,2)	Pos:0	INSTR('ALLEN','A',1,3)	Pos:0
ADAMS	INSTR('ADAMS','A',1,2)	Pos:3	INSTR('ADAMS','A',1,3)	Pos:0
AATISH	INSTR('AATISH','A',1,2)	Pos:2	INSTR('AATISH','A',1,3)	Pos:0
AAA	INSTR('AAA' ,'A',1 ,2)	Pos:2	INSTR('AAA' ,'A',1 ,3)	Pos:3
MALAYALAM	INSTR('MALAYALAM' , 'A' , 1, 2)	Pos:4	INSTR('MALAYALAM' , 'A' , 1, 3)	Pos:6

ALLEN	LENGTH('ALLEN') - LENGTH(REPLACE('ALLEN' ,A'))	= 2

	5 - LENGTH('LLEN')	
	5 - 4	
	1	!= 2
ADAMS	**5 - LENGTH('DMS')**	
	5 - 3	
	2	= 2
AAAAO	**5 - LENGTH('O')**	
	5 - 1	
	4	!= 2

SINGLE ROW FUNCTIONS

10. MOD( )
11. TRUNC( )
12. ROUND( )
13. MONTHS_BETWEEN( )
14. LAST_DAY( )
15. TO_CHAR( )
16. NVL( )
17.

10. MOD( ) : "It is used to obtain modulus/remainder of the given number "

Syntax: **MOD** (m , n) → n) m (

Example : SELECT MOD(5 , 2)
FROM DUAL ; → 1

1 . WAQTD ENAMES OF THE EMPLOYEES WHO EARN SALARY IN MULTIPLES OF 3

```
SELECT ENAME
FROM EMP
WHERE MOD( SAL , 3 ) = 0 ;
```

2. WAQTD DETAILS OF THE EMPLOYEE WHO HAVE ODD EID

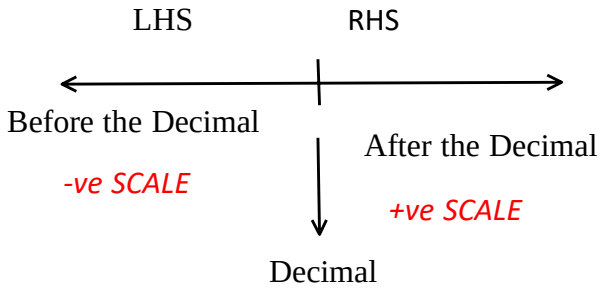
```
SELECT *
FROM EMP
WHERE MOD( EID , 2 ) = 1 ;
```

11. ROUND( ) : " It is used to Round-off the given number based on the scale value "

Syntax: **ROUND (Number [, Scale])**

The default value of scale is 0

Example : ROUND (5.6)	6
Example : ROUND (5.5)	6
Example : ROUND (5.4)	5
Example : ROUND (9.9)	10
Example : ROUND (9.4)	9
Example : ROUND (8.6 , 0)	9



When the scale is -ve it indicated the digits before the decimal
And the digit count begins from 1 .

ROUND (8421 .12 , -1)

5.6	6	Rounding off to units place

Example : ROUND (8421 .12 , -1)	8420
Example : ROUND (8426.12 , -1)	8430
Example : ROUND (154264.12 , -2)	154300
Example : ROUND (338222 , -4)	340000
Example : ROUND (2514 , -3)	3000

When the scale is +ve it indicated the digits **after the decimal**

And the digit count begins from 0 .

And the digit count begins from 0 .

Example : ROUND (124.23541 , 0)	124
Example : ROUND (124. 23541 , 1)	124.2
Example : ROUND (124. 23541 , 2)	124.24
Example : ROUND (124. 2354391 , 5)	124.23544

(124.23541 , 0) = 124

(124.23541 , 1) = 124.2

(123.671263 8723 , 6) = 123.671264

12. **TRUNC( ): "It is similar to ROUND() but it always rounds-off the given number to the lower value "**

Syntax: TRUNC(Number [, Scale])

Example : TRUNC (5.6)	5
Example : TRUNC (5.5)	5
Example : TRUNC (5.4)	5
Example : TRUNC (9.9)	9
Example : TRUNC (9.4)	9
Example : TRUNC (8.6 , 0)	8
Example: TRUNC(451258.32541 , -5)	400000

NOTE :

DATE COMMANDS :

i. **SYSDATE** : " it is used to obtain Todays Date "
ii. **CURRERNT_DATE** : " it is also used to obtain todays date "
iii. **SYSTIMESTAMP** : "It is used to obtain date , time and time zone "

```
SQL> SELECT SYSDATE
  2 FROM DUAL ;

SYSDATE

17-MAY-20

SQL> SELECT CURRENT_DATE
  2 FROM DUAL ;

CURRENT_D

17-MAY-20

SQL> SELECT SYSTIMESTAMP
  2 FROM DUAL ;

SYSTIMESTAMP

17-MAY-20 05.05.52.356000 PM +05:30
```

13. **MONTHS_BETWEEN( ) :"It is used to Obtain the number of months present between the Given two dates "**

Syntax: **MONTHS_BETWEEN** (DATE1 , DATE2)

```
SELECT TRUNC( MONTHS_BETWEEN( SYSDATE , HIREDATE ) ) || ' Months'
FROM EMP

TRUNC(MONTHS_BETWEEN(SYSDATE,HIREDATE))||'MONTH
----------------------------------------------------------
473 Months
470 Months
```

14. **LAST_DAY( ): " it is used to Obtain the last day in the particular of the given date" .**

Syntax: LAST_DAY(DATE) ;

SQL> SELECT LAST_DAY(SYSDATE) **SYSDATE = 08-JUL-2020**

2 FROM DUAL ;

LAST_DAY

31-JUL-20

15. TO_CHAR( ) :"It is used to convert the given date into String format based on the Model given "

Syntax: **TO_CHAR(DATE , 'Format _ Models')**

Format Models :

i. YEAR : TWENTY TWENTY
ii. YYYY : 2020
iii. YY : 20
iv. MONTH : JULY
v. MON : JUL
vi. MM : 07
vii. DAY : WEDNESDAY
viii. DY : WED
ix. DD : 08
x. D : 4 (day of the week)
xi. HH24 : 17 hours
xii. HH12 : 5 hours
xiii. MI : 22 minutes
xiv. SS : 53 seconds
xv. 'HH12:MI:SS' : 5 : 22 : 53
xvi. 'DD-MM-YY' : 17 - 05 - 20
xvii. 'MM-DD-YYYY' : 05 - 17 – 2020

1. WAQTD DETAILS OF THE EMPLOYEE WHO WAS HIRED ON A SUNDAY .

```
SELECT *
FROM EMP
WHERE TO_CHAR( HIREDATE , 'DAY' ) = 'SUNDAY' ;
```

2. WAQTD DETAILS OF AN EMPLOYEE HIRED ON MONDAY AT 10AM

```
SELECT *
FROM EMP
WHERE TO_CHAR( HIREDATE , 'D' ) = 2 AND TO_CHAR( HIREDATE , 'HH24' ) = 10 ;
```

16. NVL( ) : [NULL VALUE LOGIC] " It is used to eliminate the side effects of using null in arithmeticoperations " .

ENAME	SAL	COMM
A	500	100
B	1000	NULL
C	2000	200
D	2000	NULL

WAQTD NAME AND TOTAL SALALRY OF ALL THE EMPLOYEES?

SELECT ENAME , SAL + COMM
FROM EMP ;

ENAME	SAL+COMM
A	600
B	NULL
C	2200
D	NULL

Null value logic :

Syntax : **NVL (Argument1 , Argument2)**

Argument 1 : Here write any column / exp which can result In null .
Argument 2 : Here we write a numeric value which will be substituted if argument 1 results in Null ,
If argument 1 is NOT NULL then the same value will be considered .

SELECT ENAME , SAL + **NVL (COMM , 0)**

FROM EMP ;

A	500 + NVL (100 , 0)	500 + 100	600
B	1000 + NVL (null , 0)	1000 + 0	1000
C	2000 + NVL (200 , 0)	2000+200	2200
D	2000 + NVL(null , 0)	2000 + 0	2000

After using NVL

ENAME	SAL+nvl(COMM ,0)
A	600
B	1000
C	2200
D	2000

1. List employees whose name having 4 characters

```
SELECT *
FROM EMP
WHERE LENGTH(ENAME)=4 ;
```

2. List employees whose job is having 7 characters

```
SELECT *
FROM EMP
WHERE LENGTH(JOB)=4;
```

3. Find out how many times letter 'S' occurs in 'qspiders'

```
SELECT LENGTH('QSPIDERS') - LENGTH( REPLACE( 'QSPIDERS' , 'S' ) )
FROM DUAL ;
```

4. List the employees whose job is having last 3 characters as 'man'

```
SELECT *
FROM EMP
WHERE SUBSTR( JOB , -3 ) = 'MAN' ;
```

5. List employees whose job is having first 3 characters as 'man'.

```
SELECT *
FROM EMP
WHERE SUBSTR( JOB , 1 , 3 ) = 'MAN' ;
```

6. Display all the names whose name is having exactly 1 'L'

```
SELECT ENAME
FROM EMP
WHERE INSTR( ENAME , 'L' , 1,1 ) != 0 AND INSTR( ENAME , 'L' , 1, 2 ) = 0 ;
```

OR

```
SELECT ENAME
FROM EMP
WHERE LENGTH( ENAME ) - LENGTH( REPLACE( ENAME , 'L' ) ) = 1 ;
```

7. Display dept names which are having letter 'O'

```
SELECT DNAME
FROM DEPT
WHERE INSTR(DNAME,'O',1,1 ) !=0 ;
```

10. Calculate number of L in string 'HELLLLL'

```
SELECT LENGTH('HELLLLL') - LENGTH( REPLACE( 'HELLLLL' , 'L' ) )

FROM DUAL ;
```

11. Display all the employees whose job has a string 'MAN'

```
SELECT *

FROM EMP

WHERE INSTR(JOB,'MAN',1,1 ) !=0 ;
```

12. Display all the employees whose job starts with string 'MAN'

```
SELECT *
FROM EMP

WHERE INSTR(JOB,'MAN',1,1 ) =1 ;


OR


 SELECT *

FROM EMP

WHERE SUBSTR( JOB ,1,3) = 'MAN' ;
```

13. Display all the employees whose job ends with string 'MAN'

```
SELECT *
FROM EMP

WHERE SUBSTR( JOB , -3 ) = 'MAN' ;
```

14. Display first 3 characters of ename in lower case and rest everything in upper case.
If ename is 'QSPIDERS' then display this as ‘qspIDERS’

```
SELECT LOWER(SUBSTR('QSPIDERS',1,3)) || UPPER( SUSBTR('QSPIDERS' , 4) )

FROM DUAL ;
```

15. Display the result from emp table as below.SMITH is a CLERK and gets salary 2000
Here SMITH is ename column, CLERK is JOB and 2000 is SAL column and rest everything is literal strings.

```
SELECT ENAME || ' IS A '|| JOB || ' AND GETS SALARY ' || SAL

FROM EMP

WHERE ENAME = 'SMITH' ;
```

16.list the employees hired on a Wednesday

```
SELECT *
FROM EMP

WHERE TO_CHAR( HIREDATE , 'DY' ) = WED ;
```

17.list the employees hired on a leap year

```
SELECT *
FROM EMP
WHERE MOD(TO_CHAR( HIREDATE , 'YY' ) , 4 ) = 0 ;
```

18.List the employees hired on a Sunday in the month of may

```
SELECT *
FROM EMP
WHERE TO_CHAR( HIREDATE , 'DY' ) = 'SUN'  AND  TO_CHAR( HIREDATE , 'MON' ) = 'MAY' ;
```