

## Nested Functions

Syntax: func1 ( arg1, func2 (arg1, arg2...) )

{arg2}

nesting a function inside another function is called as Nested Function.

In nested function the innermost function is executed first and result is utilized by the outer function w.r.t corresponding argument.

## Character Function

1) CASE conversion function:-

→ ~~Select~~ 1) upper function is used to convert the lower case character to uppercase characters.

Ex: → select upper ('isidhar') from puct;

upper → ISIDHAR

→ select upper ('isidhar') from puct;  
upper → ISIDHAR

→ Select upper ('isidhar') from emp;

select upper ('ename') from emp; → 14 row selected

→ select upper ('sal') from emp; → 14 row selected out of 14 numbers

2) LOWER case function:- It is used to convert the uppercase characters to lowercase characters.

→ select lower ('ISIDHAR') from emp;  
lower → isidhar

→ select lower ('ENAME') from emp;  
lower ('ename') - 14 row selected

→ select lower ('HIREDATE') from emp;  
lower ('hiredate') - 14 row selected;

3) INITCAP function:- Is used to initial characters if converted to lower to appear (or) upper to lower case characters.

→ Select initcap('ISIDHAR') from puct;

INITCAP ('isidhar')

→ Select initcap('ename') from emp;  
initcap ('E.N

→ select

tip,

2) char

1) Length

(or) digits

→ Select

→ select

→ display

digit

→ select

where

2) Concat

the &

syntax!

→ Select

→ display

E b

→ selec

→ select 'Hi', 'Hello, (ename)' - you are teacher ARUN(S) bantop,  
Hi, Smith. You are a clever

ARUN(S)

PAGE NO.

DATE / /

## 2) char manipulation function

- 1) length function: length function return the no of characters  
in digits for char and column value respectively
- select length('password') from decel;  
length ('password') → 7
- select ename, length(ename) from emp; → 14 row selected
- display all the employee salary who getting 3 digit salary
- select \* from emp  
where length(sal)=3; → 2 row selected.

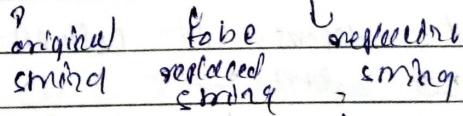
## 2) concat function: concat function is used to concatenate the data and character.

Syntax: concat (arg1, arg2)

- select concat('Hi', ' ',ename) from emp; → 14 row selected
- select concat('Hi', concat(ename, concat('.',ename,'.',job)))  
from emp; → 14 row selected
- select concat(concat(concat('Hi',ename),'.',ename,'.',job)) from emp;  
→ 14 row selected
- select concat('Hi',ename,'.',ename,'.') from emp → error

## 3) replace function: replace function returns a string in which the replaced string (arg2) will be replaced with the replacing string (arg3) from the original string (arg1).

Syntax: replace (arg1, arg2, arg3)



- select replace('apple&', 's', 'a') from decel;  
apple → applea

- select replace('apple&', 'a', 'e') from decel;  
apple → applee

- display all the name from the table by replacing e from the employee name with space
- select replace('.job', 'E', ' ') from emp; → 14 row selected

→ Select \* from emp  
where replace (ename, 'E', '') = ename; ~~displying doesn't change name~~

11/5/19

→ Select ename, replace (ename, 'L', '') from emp;

SQL> select resource\_name

from sys\_context\_info

ACCTN AEN → row selected

Substr

→ Select \* from emp

where length (ename) - length (replace (ename, 'I', ''))

not selected 2 row selected

replace function can also operate width 2 argument

arg1 & arg2 where in it eliminate & arguments

2 from eliminate argument I

Select Sub

JSPIDS

JSIPER

→ Select

→ Select replace ('SSPIERS', 'S') from Dept;

REPLAC

select \* from sale

→ DPER

where not replace(ename, 'E', '')

is not replaced only → ename; → display only affected

display all the jobs by replacing name width width women

→ Select replace (job, 'men', 'women') from emp;

Clerk

SALES WOMAN → 1 row selected

display

→ Select

where

→ Select

→ Select

where

→ Select replace (job, 'men', 1) from Emp;

→ 14 row selected

display department column by replace 20 with 30 and 30 with 20

→ select replace (replace (replace (deptno, 20, 'X'), 30, 20),  
'X', 30) from emp;

→ Select

→ Select

→ Select

→ Select

→ Select

#### 4) Substr()

substr (arg1, arg2, arg3)

arg1 → string beginning character  
arg2 → no of characters  
arg3 → ending character

J SPIDERS  
1 2 3 4 5 6 7 8

2 - 6 -> SPIDER

ARUN'S  
PAGE NO.  
DATE / /

Select str negative index

SPIDERC

-8 -7 -6 -5 -4 -3 -2 -1

→ Select Substr ('J SPIDERS', 2, 6) from emp;

SPIDERS

SPIDERS → 14 rows,

→ Select substr ('SPIDERS', 2, 6) from Dceel;

SPIDERS

→ Display the 6th character of all employee name

→ select substr (ename, 1, 1) from emp

Display the <sup>last</sup> character of all employee name

→ select substr (ename, length (ename), 1) from emp

- 14 row selected

Display the employee name starting with A

→ select \* from emp

where substr (ename, 1, 1) = 'A';

& now selected

→ select substr (ename, length (ename)-1, 1), ename from emp;

→ select \* from emp

where substr (ename, length (ename)-1, 1) = 'E'

5 row selected

Display employee job ending with 'man'

→ select job, substr (job, length (job)-2, 3) from emp

- 3, 3 ) - ↳ 14 row selected

→ select \* from emp;

→ select job, substr (job, length (job)-2, 3) = 'man'

& now selected.

SQL > Negate all where  
select substr('JSIiders', -1, 1) from race

ARUN'S  
PAGE NO.  
DATE 2/5/19

S

SQL > select substr('JSPIders', -3, 3) from race;

Subs

JS

JSPIDERS  
8 7 6 5 4 3 2 1

Sub str  
can use negative indexing as argument too  
where the indexing is reversing order but the selection  
of character follows order

SQL > select substr('Name', -length('name'), 1) from emp;

S

1 row selected

A

J

H

SQL > select substr('JSIiders', 0, 3) from race;

Subs

JSI

SQL > select substr('JSPIders', 1, 3) from race;

Subs

JSPI

SQL > select substr('JSPIders', 1, 5, 3) from race;

Subs

JSPI

SQL > select substr('JSPIders', 9, 2) from race;

→ blank

SQL > select substr('JSPIders', -9, 2) from race;

→ blank

Pt due order is not in reverse then Substr  
returns nothing

SQL > select substr('JSPIders', 1, -2) from race

→ blank

SQL > Select Se

S

SQL > Select Se

Sa

JS

SQL > Select Se

substr C

CPIDE

Note:- It c

hanges, th

according

Note:- substr (

-8 cargo

SQL > Select

char

SQL > Select

JS

SQL > Select

display

SQL > Select

smith

Allen

SQL > Select

Smith

Allie

(SQL) select substr('SSPIDEr', 1, 6) from pceel;

S

ARUN'S

PAGE NO.

DATE / /

(SQL) select substr('SSPIDEr', 1, 2, 5) from pceel

Sa

JS

(SQL) select substr('SSPIDEr', 2, 10) from pceel;

Select C

SPIDER

Note:- If arg3 is out of range of given original

string, then substr returns all the characters available from the given beginning index

Note:- substr() function can also operate with 3 arguments

-> (arg1, arg2, arg3) where it return all the available

(SQL) select substr character from the given beginning  
character.

(SQL) select substr('SSPIDEr', 2) from pceel;

& SPIDER

(SQL) select substr('SSPIDEr', 2) from pceel;

20008

cap

display like upper function  $\rightarrow$  Smith  $\rightarrow$  lower

(SQL) select substr replace (ename, substr (ename, 1), lower (substr  
(ename, 1))) from emp;

Smith

, & now selected.

Allen

$\rightarrow$  upper convert operator display  $\rightarrow$  Allen  $\rightarrow$  lower

(SQL) select concat (substr (ename, 1, 1), lower (substr (ename, 2)))  
from emp;

Sonita

, & now selected

Allen

SMITH  $\rightarrow$  Sonita

Smith

5) INSTR (arg1, arg2, arg3, arg4)

↓      ↓      ↓      ↓  
String    Search string    beginning index    no of occurrence

ARUN'S  
PAGE NO.  
DATE / /

INSTR () function returns the index of the search string, if it is present in the string.

(current)  
1 2 3 4 5 6 7 8  
S P T V G R S

SQL> select INSTR ('JSRiders', 's', 1, 1) from dual;

INSTR

2

SQL> select INSTR ('JSRiders', 's', 1, 2) from dual;

8

it returns the index of the search string if it is in the original string for the specified number of occurrence (argument 4).

if the search string is not found in original string for the specified number of occurrence then

INSTR returns zero (0).

NOTE:- If the search string is not found in the original string then for the specified number of occurrence then INSTR returns zero (0).

SQL> select INSTR ('JSRiders', 's', 3, 2) from dual;

0

SQL> select \* from emp

where INSTR (ename, 'A', 1, 1)=1;

ALLAN

ADAM

SQL> select \* from emp

where INSTR (ename, 'A', 1, 1)!=0

ALLEN

WARD

ADAM

SQL> select INSTR (ename, 'A', 1, 1), ename from emp

① display

SQL> select result

② lower (su

, 1), lower

SMITH

SCOTT

ADAMS

③

SQL> select conc

cat str (en

lower (su

to

④ select re

② display cu

+ select re

length (en

replace (ca

+ 1))

SQL> select Inis

INSTR (

SQL> Select fr

INSTR (

① INSTR C

function

Search

② INSTR

string m

① Display the SMITH ~~name~~

ARUN'S

PAGE NO.

DATE 3/15/19

SQl> select replace(replace(ename, substr(ename,1,1)),

② lower(substr(ename,1,1)), substr(ename,length(ename),  
1), lower(substr(ename, length(ename), 1)) from emp;

SMITH

Ex: SCOTT

SCOTT

ADAMS

(8)

SQl> select concat(concat(lower(substr(ename,1,1)),  
substr(ename,2,length(ename)-2)),  
lower(substr(ename,length(ename))))  
from emp;

or select replace

② Display all the employee names in the format SMITH  
+ Select replace(ename, ename, lower(substr(ename,1,  
length(ename)/2))) || substr(ename, length(ename)/2+1,1)||  
replace(ename, ename, lower(substr(ename, length(ename)  
+1))) from emp

J S R D G R S

- P - 7 - 6 - 5 - 4 - 3 - 2 - 1

SQl> select instr('ISRAEL', 'S', 1, 1) from dual;

INSTR ('ISRAEL', 'S', 1, 1)

2

SQl> select instr('ISRAEL', 'S', -1, 1) from dual;

INSTR ('ISRAEL')

8

x INSTR can operate with negative indexing where negative  
 bedeutet in the string ist dervery order will for the  
 search in searchstr in reverse order.

y INSTR returns corresponding + de code of the return  
 string in form of

SQL> select instr('SSPWORK','S',1,2) from dual;

2

ARUN'S  
PAGE NO.  
DATE

Note 2: INSTR

where n

only consider

Date Arithmetic

\* calc allowed

+ substraction

\* Addition +

ix allowed

previous de

→ only in sub

i) Date function

ii) sysdate :-

on which

\* SQL always

of d

SQL select by

06 MAY

SQL select syste

16 - MAY

SQL select syste

26 - APR

SQL select syste

02 - MAR

-1 display a

SQL select syste

format

SQL select syste

SQL select syste

SQL> select instr('SSPWORK','S',1,2) from dual;

2

SQL> select instr('SSPWORK','S',1,3) from dual;

2

SQL> select instr('SSPWORK','S',1,-3,2) from dual;

0

INSTR ('SSPWORK','S',1,-3,2)

SQL> select instr('SSPWORK','S',0,1) from dual;

0

SQL> select instr('SSPWORK','S',0,1,1) from dual;

0

SQL> select instr('SSPWORK','S',1,-1) from dual;

Error

SQL> select instr('SSPWORK','S',1,0) from dual;

Error

SQL> select instr('SSPWORK','S',1,10) from dual;

0

SQL> select instr('SSPWORK','S',1,1)

2

default occurrence n get to 1

SQL> select instr('SSPWORK','S')

2

by default take both arguments

NOTE!: INSTR function can also operate with 3 arguments

arg1, arg2 & arg3. where n arg1 & m arg2

→ by 1 but if INSTR by default look for 1<sup>st</sup> occurrence at the given string

Note 2: Instr can also operate certain & curcumstances  
where in args & args by default, I first of all  
will remember given a search arg1 start from beginning  
then

ARUN'S  
PAGE NO. 60  
DATE / /

6/5/19

C and double L

7/5/19

instr(cname, 'L', 1, 1) != 0 } for single L  
AND instr(cname, 'L', 1, 2) = 0 }

instr(cname, 'C', 1, 2) != 0 } for double C  
AND instr(cname, 'C', 1, 3) != 0 } in diff place

SQL) Select \* from emp

where instr(cname, 'C', 1, 1) != 0 and  
instr(cname, 'C', 1, 2) = 0;

OR Block

Link

SQL) select \* from emp

where instr(cname, 'L', 1, 2) != 0 and

SQL) select \* from emp  
where instr(ename, 'L', 1, 2) != 0 AND instr(ename, 'U', 1, 3) = 0;

level      select -> L  
Allen  
miller

SQL) select \* from emp  
where instr(ename, 'L', 1, 2) != 0;

level  
COLLITA  
ALLAN  
miller

SQL) select \* from emp  
where instr(ename, 'L', 1, 1) != 0;

COLLITA  
-ACLEN  
miller

Select num

(last in given word)

SQL) select \* from emp

where job like '%MAN%'

ACGAL  
WAHP  
MAYRON  
TURMER

SQL) select \* from emp

where instr(job, 'MAN') = length(job)-2;

ACGAL  
WAHP  
MAYRON  
TURMER

SQL) select ~~\*~~ from emp concat(concat(substr(ename, -1),  
substr(ename, 2, length(ename)-2)), substr  
(ename, 1, 1) from sql);

smith → HMTS like all

init cap

1) select initcap(firstname) from emp → shows selected

replace

1) select sal, job from emp

where not replace(job, 'man', 'woman') = job;

if display's which ~~not~~ obj. will be replaced. ⚡

1st evaluate true and

2) select sal, job from emp

the. not operation take

where replace(job, 'man', 'woman') = job; ~~will~~

if display's which not replaced.

substr → assignment

~~SMITH~~ SMITH

1) select replace(replace(ename, substr(ename, 1, 1), lower(substr(ename, 1, 1))), substr(ename, length(ename), 1), lower(substr(ename, length(ename), 1))) from emp

SMITH →

Q) select ename, replace(replace(ename, substr(ename,  
1, length(ename)/2), reverse(substr(ename, 1, length(ename)  
/2))), substr(ename, length(ename)/2+1, 1), upper(  
substr(ename, length(ename)/2+1, 1)), substr(ename,  
length(ename)/2+2), lower(substr(ename, length(ename)/2+2))  
from emp;

INSTO

1) select ename from emp

where insto(ename, 'A', 1, 1)=1;

ALLEN and ADAM selected because it's searching

a A in index position at i=1 only in First

occurrence

2) select ename from emp

where insto(ename, 'A', 1, 1)=3;

BLAKE and CLERK will select But ADAM is not

Select Because it searching Astrik in the position

3 of First occurrence in ADAM ename

First occurrence already has \* in 1st position

SMITH

Smith

Centre enume letter upper case

WARP BRUNSWICK  
PAGE NO. 1  
DATE 1/1

(2) lower (substr (ename, (length (ename) / 2)) ) ^ 3 (en) wa

or lower || substr (ename, (length (ename) / 2) + 1, 1 ) @ 3 (2) r  
lower || substr (ename, ((length (ename) / 2) + 2)) @ 4 (2) s

selected → 14 now selected

selected SMITH @ 4 WARP

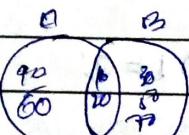
12 3 4 5 6 7 8 9 10

1 2 8 6 7 9 10

Set theory

18/5/19

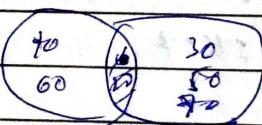
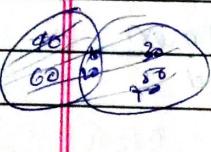
Formation functions



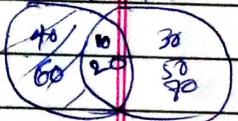
A = {10, 20, 40, 60}

B = {10, 20, 30, 50, 70}

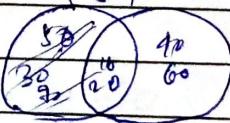
A ∩ B = {10, 20, 40, 60}



A - B = {40, 60}

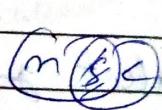


B - A = {30, 50, 70}

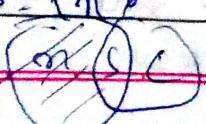


$\{S, m\}^A$	$S \times f \text{ emp}$	Set theory	set op's
$\{S, m\}^B$	$w, job \text{ in } (S, m)$	Set op's	union
$\{S, m\}^C$	$S \times f \text{ emp}$	Set op's	Union All
$\{S, m\}^D$	$w, job \text{ in } (S, m)$	DQL	intersect
$\{S, m\}^E$	$S \times f \text{ emp}$	Set op's	minus

A ∪ B = {S, m, C}    A ∩ B = {m, S, C}    A ⊕ B = S



A - B = {1}



B - A = {2}



~~ABC~~

~~m \$X~~

Half name

ARUN'S

PAGE NO.

DATE / /

SQL) select substr(ename, 1, length(ename)/2) from emp  
 SAn  
 AL  
 ?

SQL) select ename, substr(ename, 1, length(ename)/2) from emp;  
 'smith' em  
 'ALLEN' AL  
 ? :

SQL) select concat(substr(ename, (length(ename)/2)+1))  
 from emp;  
 smith itu  
 ALLEN len  
 ? :

### Conversion function

\* Conversion is converting one data type to another data type  
 \* depending on whether conversion is automatic (by default)  
 categorized as implicit or explicit respectively.

SQL) select '10' + '0' from dual;  
 '10' + 10 -> 20

SQL) Select 'a' + 10. from dual;  
 Error

SQL) select \* from emp  
 where hiredate > '10-feb-81';  
 syntax error select

SQL) select substrdate - '10-feb-81' from dual  
 error

NOTE:-  
 \* implicit conversion may (not) be done in all the given scenario

- \* to ensure the conversion some time has to be done explicitly using conversion concert conversion function