PL/ SQL

Topics :-

1. Aggregate Function Aggregate Funtions
2. DDL
3. Function
4. Procedure
5. Package
6. Cursor
7. Triggers
8. Primary key
9. Foreign key
10. Dynamic SQL

1. Aggregate Function
   TOPICES COVERED IN THIS SESSION

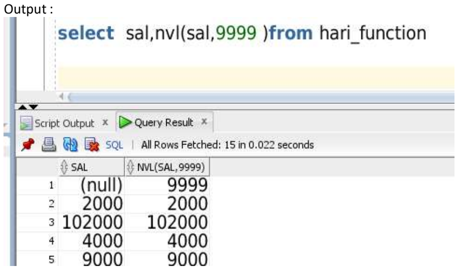
- NVL
- NULLIF
- NTH VALUE
- NUMTODSINTERVAL
- NVL2
- NUMTOYMINTERVAL
- POWER
- RANK
- RAWTOHEX
- ROWNUM
- RPAD
- RTRIM
- REPLACE
- REGEXP_REPLACE
- REGEXP_COUNT
- REMAINDER

____

NVL :

Query :
 select nvl(sal,9999)from hari_function

Output :



```
select  sal,nvl(sal,9999 )from hari_function
```

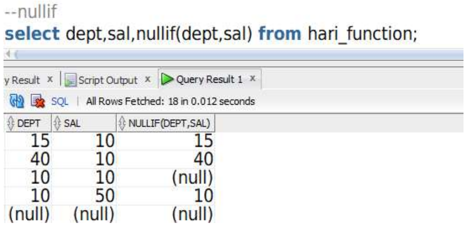
```
     SAL      NVL(SAL,9999)
1   (null)         9999
2     2000         2000
3   102000       102000
4     4000         4000
5     9000         9000
```

____

NULLIF:

Query:
select dept,sal,nullif(dept,sal) from hari_function;

Output:



```
--nullif
select dept,sal,nullif(dept,sal) from hari_function;
```

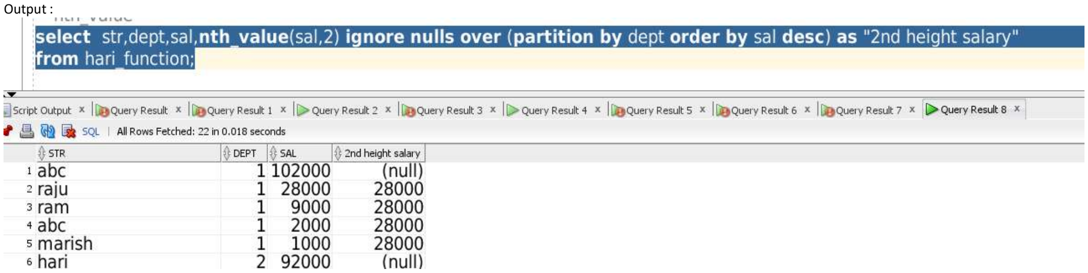
```
  DEPT    SAL    NULLIF(DEPT,SAL)
   15      10        15
   40      10        40
   10      10    (null)
   10      50        10
(null) (null)    (null)
```

____

NTH VALUE :

Query :
select str,dept,sal,nth_value(sal,2) ignore nulls over (partition by dept order by sal desc) as "2nd height salary" from hari_function;

Output :



```
select  str,dept,sal,nth_value(sal,2) ignore nulls over (partition by dept order by sal desc) as "2nd height salary"
from hari_function;
```

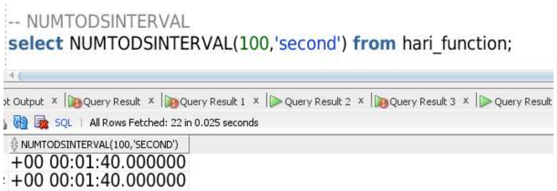
```
   STR       DEPT   SAL     2nd height salary
1  abc         1  102000        (null)
2  raju        1   28000         28000
3  ram         1    9000         28000
4  abc         1    2000         28000
5  marish      1    1000         28000
6  hari        2   92000        (null)
```

____

NUMTODSINTERVAL :

Query :
select NUMTODSINTERVAL(100,'second') from hari_function;

Output:



```
-- NUMTODSINTERVAL
select NUMTODSINTERVAL(100,'second') from hari_function;
```

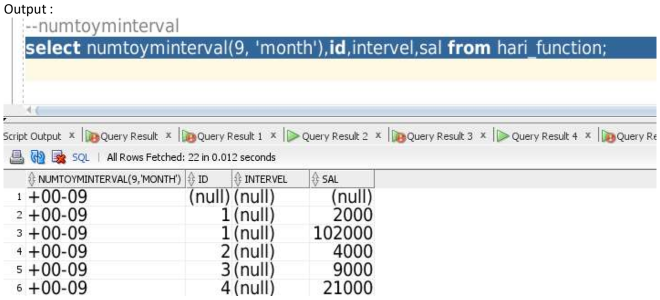
All Rows Fetched: 22 in 0.025 seconds

```
NUMTODSINTERVAL(100,'SECOND')
+00 00:01:40.000000
+00 00:01:40.000000
```

NUMTOYMINTERVAL :

Query :
select numtoyminterval(9, 'month') from hari_function;

Output :



```
--numtoyminterval
select numtoyminterval(9, 'month'),id,intervel,sal from hari_function;
```

All Rows Fetched: 22 in 0.012 seconds

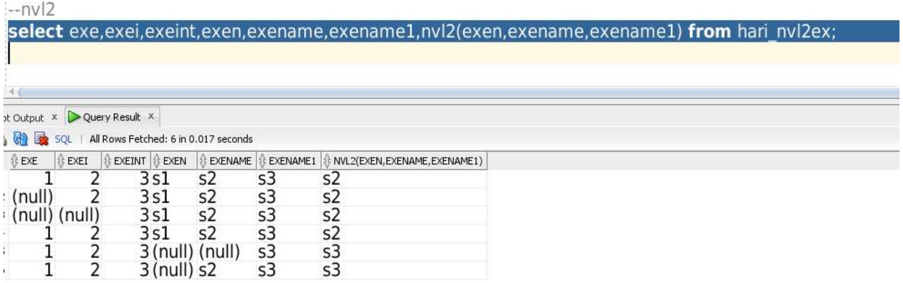
```
    NUMTOYMINTERVAL(9,'MONTH')  ID      INTERVEL   SAL
  1 +00-09                      (null)  (null)     (null)
  2 +00-09                           1  (null)       2000
  3 +00-09                           1  (null)     102000
  4 +00-09                           2  (null)       4000
  5 +00-09                           3  (null)       9000
  6 +00-09                           4  (null)      21000
```

NVL2:

Query :

select exe,exei,exeint,exen,exename,exename1,nvl2(exen,exename,exename1) from hari_nvl2ex;

Output :



```
--nvl2
select exe,exei,exeint,exen,exename,exename1,nvl2(exen,exename,exename1) from hari_nvl2ex;
```

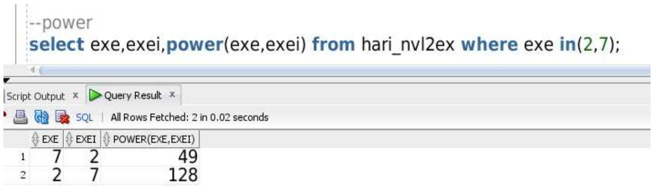
All Rows Fetched: 6 in 0.017 seconds

```
EXE     EXEI    EXEINT  EXEN    EXENAME  EXENAME1  NVL2(EXEN,EXENAME,EXENAME1)
     1       2       3  s1      s2       s3        s2
(null)       2       3  s1      s2       s3        s2
(null)  (null)       3  s1      s2       s3        s2
     1       2       3  s1      s2       s3        s2
     1       2       3  (null)  (null)   s3        s3
     1       2       3  (null)  s2       s3        s3
```

Power:

Query:
select exe,exei,power(exe,exei) from hari_nvl2ex where exe in(2,7);

Output:



```
--power
select exe,exei,power(exe,exei) from hari_nvl2ex where exe in(2,7);
```

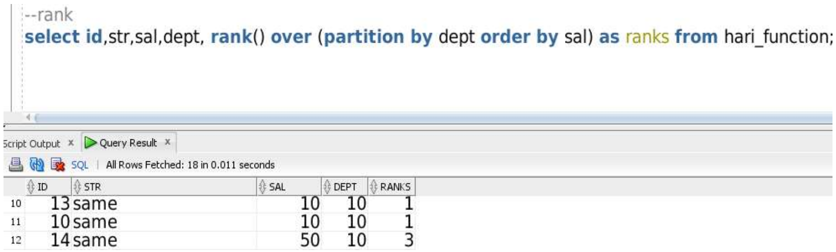
All Rows Fetched: 2 in 0.02 seconds

```
    EXE  EXEI  POWER(EXE,EXEI)
  1   7     2               49
  2   2     7              128
```

Rank:

Query:
select id,str,sal,dept, rank() over (partition by dept order by sal) as ranks from hari_function;

Output:-



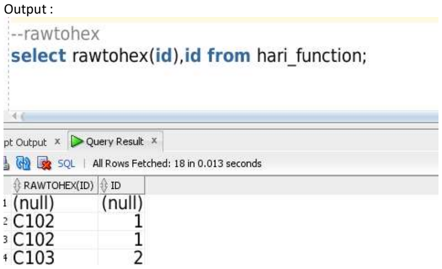
```
--rank
select id,str,sal,dept, rank() over (partition by dept order by sal) as ranks from hari_function;
```

```
     ID  STR     SAL  DEPT  RANKS
10   13  same     10    10      1
11   10  same     10    10      1
12   14  same     50    10      3
```

Rawtohex:
Query:-

select rawtohex(id),id from hari_function;

Output :



```
--rawtohex
select rawtohex(id),id from hari_function;
```

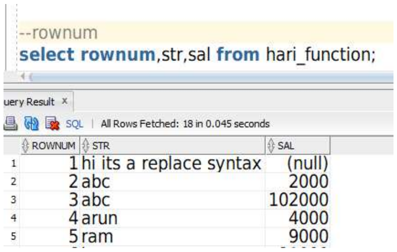
```
  RAWTOHEX(ID)  ID
1 (null)        (null)
2 C102               1
3 C102               1
4 C103               2
```

Rownum:

Query:-

select rownum,str,sal from hari_function;

Output:-



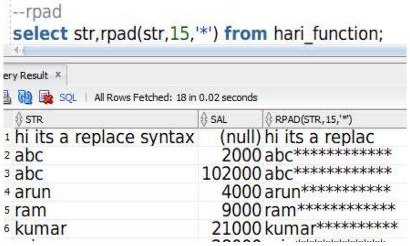
```
--rownum
select rownum,str,sal from hari_function;
```

```
   ROWNUM  STR                      SAL
1       1  hi its a replace syntax  (null)
2       2  abc                      2000
3       3  abc                      102000
4       4  arun                     4000
5       5  ram                      9000
```

RPAD :-
Query :-

select str,rpad(str,15,'*') from hari_function;

Output :-



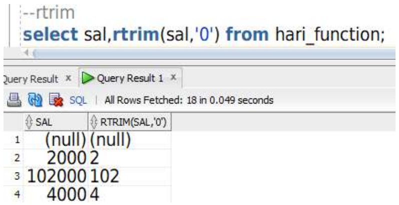
```
--rpad
select str,rpad(str,15,'*') from hari_function;
```

```
  STR                      SAL     RPAD(STR,15,'*')
1 hi its a replace syntax  (null)  hi its a replac
2 abc                      2000    abc************
3 abc                      102000  abc************
4 arun                     4000    arun***********
5 ram                      9000    ram************
6 kumar                    21000   kumar**********
```

RTRIM :-
Query:-

select rtrim('teck 123 teck is 123123123','123') from dual;

Output:-



```
--rtrim
select sal,rtrim(sal,'0') from hari_function;
```

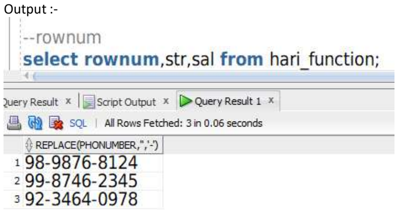
```
  SAL     RTRIM(SAL,'0')
1 (null)  (null)
2 2000    2
3 102000  102
4 4000    4
```

REPLACE :-

Query :-

select replace(phonumber,' ','-') from hari_number;

Output :-



```
--rownum
select rownum,str,sal from hari_function;
```

```
All Rows Fetched: 3 in 0.06 seconds

  REPLACE(PHONUMBER,'','-')
1 98-9876-8124
2 99-8746-2345
3 92-3464-0978
```

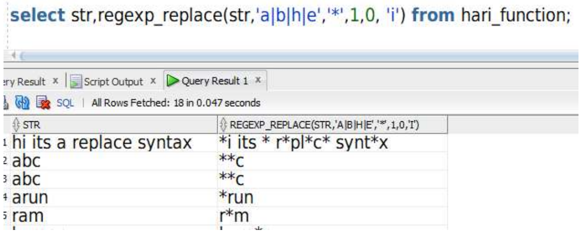
____

REGEXP_REPLACE:-

Query:-

select str,regexp_replace(str,'a|b|h|e','*',1,0, 'i') from hari_function;

Output:-



```
select str,regexp_replace(str,'a|b|h|e','*',1,0, 'i') from hari_function;
```

```
All Rows Fetched: 18 in 0.047 seconds

  STR                      REGEXP_REPLACE(STR,'A|B|H|E','*',1,0,'I')
1 hi its a replace syntax  *i its * r*pl*c* synt*x
2 abc                      **c
3 abc                      **c
4 arun                     *run
5 ram                      r*m
```

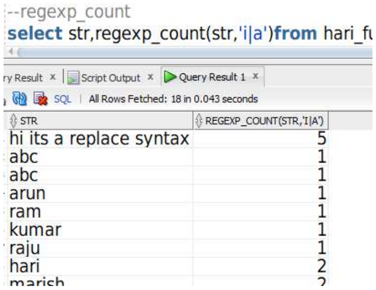
____

Regexp_count:-

Query :-

select str,regexp_count(str,'i|a')from hari_function;

Output:-



```
--regexp_count
select str,regexp_count(str,'i|a')from hari_fu
```

```
All Rows Fetched: 18 in 0.043 seconds

STR                      REGEXP_COUNT(STR,'I|A')
hi its a replace syntax                        5
abc                                            1
abc                                            1
arun                                           1
ram                                            1
kumar                                          1
raju                                           1
hari                                           2
marish                                         2
```

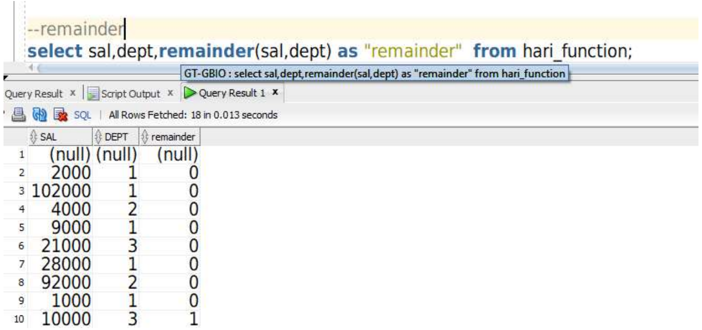
____

REMAINDER:-

Query:-

select remainder('7','3') from dual;
select sal,dept,remainder(sal,dept) as "remainder" from hari_function;

Output:-



```
--remainder
select sal,dept,remainder(sal,dept) as "remainder"  from hari_function;
```

```
GT-GBIO : select sal,dept,remainder(sal,dept) as "remainder" from hari_function
```

```
All Rows Fetched: 18 in 0.013 seconds

      SAL   DEPT remainder
 1 (null) (null)    (null)
 2   2000      1         0
 3 102000      1         0
 4   4000      2         0
 5   9000      1         0
 6  21000      3         0
 7  28000      1         0
 8  92000      2         0
 9   1000      1         0
10  10000      3         1
```

2. DDL

```
--creat table
CREATE TABLE hari_first (
                                        sid NUMBER (10),
                                        sname VARCHAR(20),
                                        mobile NUMBER(10),
                                        dob DATE
                                    );
--Add column
ALTER TABLE hari_first ADD blood_group number(5);

--Modify datatype
ALTER TABLE hari_first modify blood_group varchar(5);

-- drop column
```

```
ALTER TABLE hari_first drop column blood_group;

--rename
ALTER TABLE hari_first rename column mobile to phone_no;

--insert
insert into hari_first values ('01','hari','123456789','20-01-20');

--insert all
insert  all
     into hari_first values ('02','hari','2222222222','20-02-2000')
     into hari_first values ('03','ram','3333333','21-12-2003')
     into hari_first values ('04','vikay','4444444','29-09-2009')
     into hari_first values ('06','nivas','556677788','27-08-2007')
     into hari_first values ('07','arun','0009998887','26-07-2008')
     into hari_first values ('08','kumar','0912573897','28-05-2011')
     into hari_first values ('09','praven','908783657','19-03-2022')
select * from hari_first;
commit;

--insert all
update hari_first set sname = 'hhmm' where sid='2';
commit;

--delete
delete from hari_first where sid='08';
commit;

select * from hari_first;

TRUNCATE TABLE hari_first;
```

3. Function

```
--function creation
drop function hari_fun_areaofcircle;
create function hari_fun_areaofcircle(n_radius in number)
return number
is
pye constant number(7,3) := 3.14;
n_output number(7.3):=0.0;
begin
    n_output := pye * n_radius * n_radius;
    return n_output;
end;

--function execution or calling
begin
    dbms_output.put_line('area of the circle is
'||hari_fun_areaofcircle(5));
end;
                -- output is :- area of the circle is 79. | it
will display in dbms output.
select hari_fun_areaofcircle(10) from dual;
                --output is :- 314. | it will display in script
output.
```

4. Procedure

```
--procedure creation
create or replace procedure hari_pro_add(n_num1 in number,n_num2 in
number)
is
n_sum number := 0;
begin
    n_sum := n_num1 + n_num2;
    dbms_output.put_line('output '|| n_sum);
end;

--execution of a procedure
execute hari_pro_add(10,20);

begin
    hari_pro_add(10,30);
end;
```

5. Package

```
create or replace package niranjan_pkg as
function niranjan_fun(a in number,b in number)RETURN number;
PROCEDURE niranjan_prc(a in number,b in number,c out number);
end niranjan_pkg;


create or replace package body niranjan_pkg as
procedure niranjan_prc(a in number,b in number,c out number)
IS
begin
c:=a+b;
end niranjan_prc;
```

6. Cursor

Cursor is a pointer to a memory area called context area. It is used to hold the information about select or dml statement.

Context area:-

It is a memory region inside the PGA which helps oracle server in processing an sql statement. It will also hold the important information about that statement which is stored

Information like -

- Rows returned by a query.
- Number of rows processed by a query.
- A pointer to the parsed query in the shared pool.

Types of cursors:-

1. Implicit
2. Explicit

1. **Implicit**:-
   - Automatically created by the oracle server when a sql dml statement is executed.
   - User cannot control the behavior of these cursors.
   - Oracle server creates an implicit cursor for any pl/sql block which executes an SQL statement as long as an explicit cursor does not exists for that SQL statement.
2. Explicit:-
   - Explicit cursor are user defined cursor .

- User has full control of explicit cursor.

Steps for creating an explicit cursor:-

- Declare - declaring a cursor means initializing a cursor into memory.
  we define explicit cursor in declaration section of your PL/SQL block.
- Open - in order to put that cursor to work we have to open it first.
  when you open a cursor the memory will be allotted to it.
- Fetch - the process of retrieving the data from the cursor.
- Close - the cursor will relinquish all the resources associated with it .

Syntax :-

- Declare
  ```
  CURSOR cursor_name IS select_statement;
  ```
- Open
  ```
  OPEN cursor_name;
  ```
- Fetch
  ```
  FETCH cursor_name INTO PL/SQL variable;
                           Or
  FETCH cursor_name INTO PL/SQL record;
  ```
- Close
  ```
  CLOSE cursor_name;
  ```

EXAMPLE:-

```
declare
   v_name varchar2(30);
   --curser declaration
   cursor cur_first is
   select str from hari_function where sal<=10000;
begin
   --open cursor
   open cur_first;
   loop
      --fetch cursor
      fetch cur_first into v_name;
      DBMS_OUTPUT.PUT_LINE(v_name);
      exit when cur_first%notfound;
   end loop;
   close cur_first;
end;
```

CURSOR ATTRIBUTE :-

- %rowcount
- %found
- %notfound
- %isopen

RECORD TYPE:-

There r 3 types

- Table based record
  - Variable name table name%rowtype
- Cursor based record
  - Variable name cursor name%rowtype
- User defined record

--execution

```
declare
CURSOR cursor_1  is select * from niranjan_customers
where id=11;
CURSOR cursor_2 is select * from hari_function where id=2;
record_1 niranjan_customers%rowtype;
record_2 hari_function%rowtype;
begin
open cursor_1;
fetch cursor_1 into record_1;
close cursor_1;
dbms_output.put_line('The name = '||record_1.name||' The salary = '||record_1.salary);
open cursor_2;
fetch cursor_2 into record_2;
close cursor_2;
dbms_output.put_line('The id = '||record_2.id||' The salary = '||record_2.SAL);
end;
```

7. **Triggers**

We can usen triggers in the following events

- DML statements
- DDL statements
- A system event - shut down/startup events
- A user event - log off/log on of a DB

Types :-

- DML triggers
- DDL triggers
- System/ databse event triggers - log in / log off
- Instead-of - stop and redirect the performance of DML Triggers
- Compound - used

8. **Primary key**

Primary key:-

Primary key constraint is the combination of NOT NULL and UNIQUE constraints. I,e, if a column is said to be primary key then that column will not accept NULL and duplicate values .

There are two types of primary keys:-

1. Simple primary key.
2. Composite primary key.

1. Simple primary key:-
   If a table contains only one primary key column, then it is said to be simple primary key.

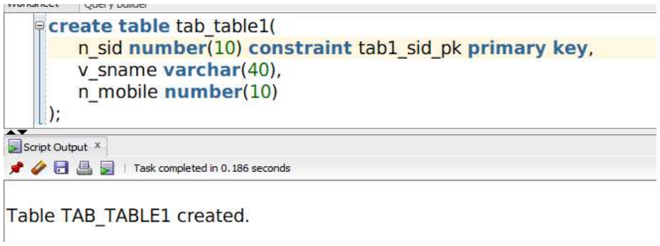
2. Composite primary key:-
   If a table contains more than one primary key column, then it is said to be composite primary key.

1.simple primary key:-

Ways to create primary key:-

It can be created in two ways.

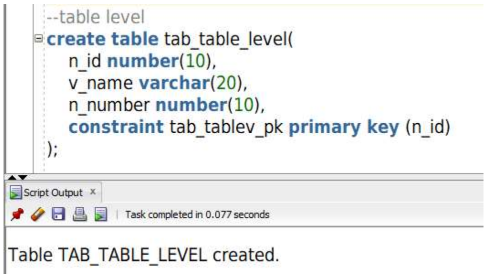
1. Using create table - 2 types. Column and table level
   1. Column level :-



```
create table tab_table1(
    n_sid number(10) constraint tab1_sid_pk primary key,
    v_sname varchar(40),
    n_mobile number(10)
);
```

```
Table TAB_TABLE1 created.
```

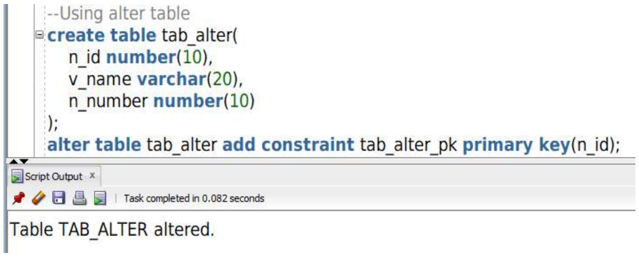
2. Table level



```
--table level
create table tab_table_level(
    n_id number(10),
    v_name varchar(20),
    n_number number(10),
    constraint tab_tablev_pk primary key (n_id)
);
```

```
Table TAB_TABLE_LEVEL created.
```

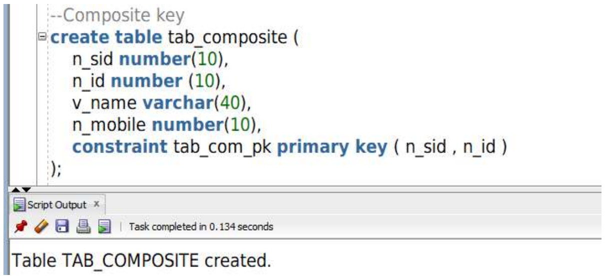
2. Using alter table:-



```
--Using alter table
create table tab_alter(
    n_id number(10),
    v_name varchar(20),
    n_number number(10)
);
alter table tab_alter add constraint tab_alter_pk primary key(n_id);
```

```
Table TAB_ALTER altered.
```

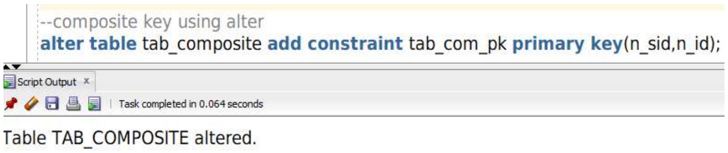
1. **Composite primary key:-**



```
--Composite key
create table tab_composite (
    n_sid number(10),
    n_id number (10),
    v_name varchar(40),
    n_mobile number(10),
    constraint tab_com_pk primary key ( n_sid , n_id )
);
```

```
Table TAB_COMPOSITE created.
```

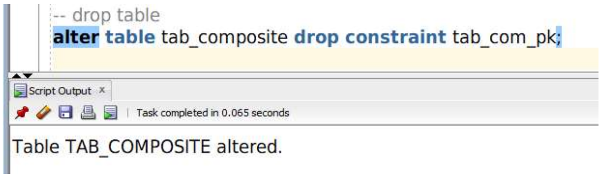
Using after :-



```
--composite key using alter
alter table tab_composite add constraint tab_com_pk primary key(n_sid,n_id);
```

```
Table TAB_COMPOSITE altered.
```

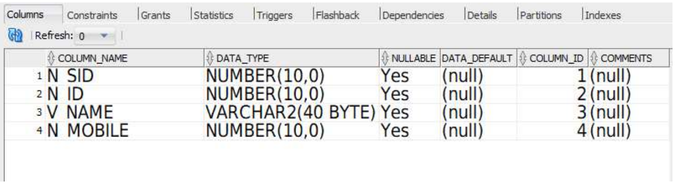
Drop constraint :-



```
-- drop table
alter table tab_composite drop constraint tab_com_pk;
```

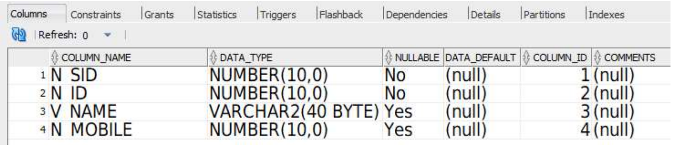
```
Table TAB_COMPOSITE altered.
```

Before using primary key:-



	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	N SID	NUMBER(10,0)	Yes	(null)	1	(null)
2	N ID	NUMBER(10,0)	Yes	(null)	2	(null)
3	V NAME	VARCHAR2(40 BYTE)	Yes	(null)	3	(null)
4	N MOBILE	NUMBER(10,0)	Yes	(null)	4	(null)

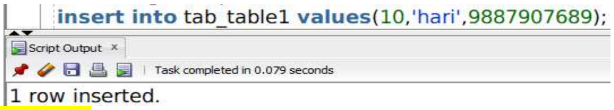
After using primary key:-



	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	N SID	NUMBER(10,0)	No	(null)	1	(null)
2	N ID	NUMBER(10,0)	No	(null)	2	(null)
3	V NAME	VARCHAR2(40 BYTE)	Yes	(null)	3	(null)
4	N MOBILE	NUMBER(10,0)	Yes	(null)	4	(null)

Inserting :-

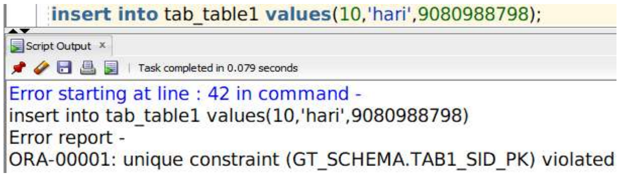
1. Inserting a record in single primary key:-



```
insert into tab_table1 values(10,'hari',9887907689);
```

```
1 row inserted.
```

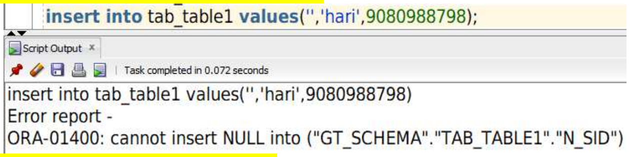
NO ERROR INSERTED .



```
insert into tab_table1 values(10,'hari',9080988798);
```

```
Error starting at line : 42 in command -
insert into tab_table1 values(10,'hari',9080988798)
Error report -
ORA-00001: unique constraint (GT_SCHEMA.TAB1_SID_PK) violated
```

PRIMARY KEY COLUMN DOES NOT ALLOW DUPLICATIVE VALUES.

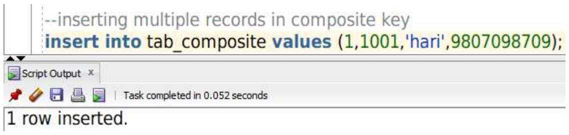


```
insert into tab_table1 values('','hari',9080988798);
```

```
insert into tab_table1 values('','hari',9080988798)
Error report -
ORA-01400: cannot insert NULL into ("GT_SCHEMA"."TAB_TABLE1"."N_SID")
```

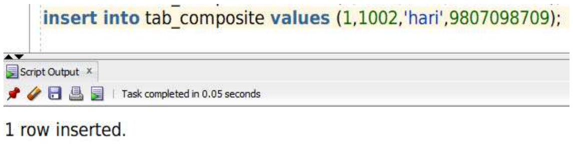
PRIMARY KEY COLUMM DOES NOT ALLOW NULL VALUES

2. Inserting a record in composite primary key:-



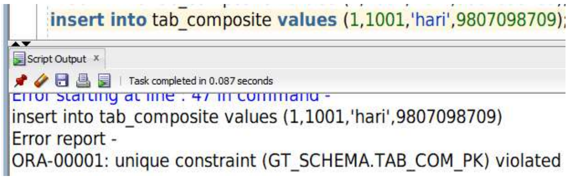
```
--inserting multiple records in composite key
insert into tab_composite values (1,1001,'hari',9807098709);
```

```
1 row inserted.
```



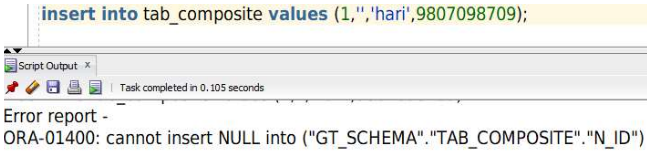
```
insert into tab_composite values (1,1002,'hari',9807098709);
```

```
1 row inserted.
```



```
insert into tab_composite values (1,1001,'hari',9807098709);
```

```
Error starting at line : 47 in command -
insert into tab_composite values (1,1001,'hari',9807098709)
Error report -
ORA-00001: unique constraint (GT_SCHEMA.TAB_COM_PK) violated
```

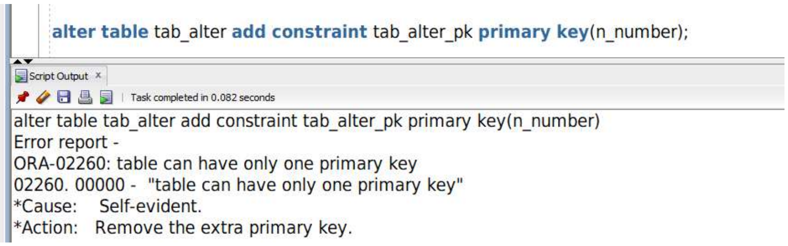


```
insert into tab_composite values (1,'','hari',9807098709);
```

```
Error report -
ORA-01400: cannot insert NULL into ("GT_SCHEMA"."TAB_COMPOSITE"."N_ID")
```

NOTES :-

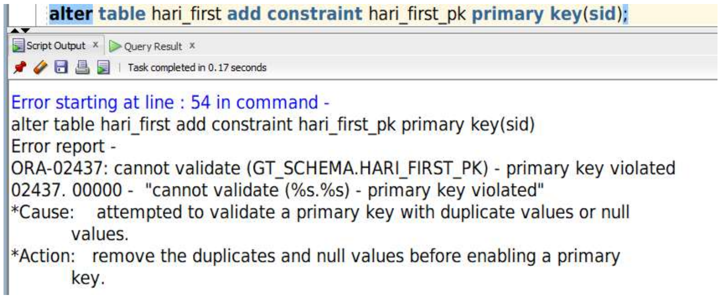
- If a table has a primary key and if we try to add one more primary key. Then it will show the below error.



```
alter table tab_alter add constraint tab_alter_pk primary key(n_number);
```

```
alter table tab_alter add constraint tab_alter_pk primary key(n_number)
Error report -
ORA-02260: table can have only one primary key
02260. 00000 -  "table can have only one primary key"
*Cause:    Self-evident.
*Action:   Remove the extra primary key.
```

- If a column of a table has a duplicate value and if we try to add primary key to that column. Then is will show the below error.

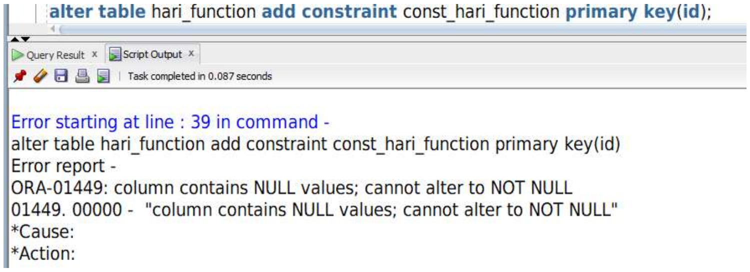


```
alter table hari_first add constraint hari_first_pk primary key(sid);
```

Task completed in 0.17 seconds

```
Error starting at line : 54 in command -
alter table hari_first add constraint hari_first_pk primary key(sid)
Error report -
ORA-02437: cannot validate (GT_SCHEMA.HARI_FIRST_PK) - primary key violated
02437. 00000 -  "cannot validate (%s.%s) - primary key violated"
*Cause:    attempted to validate a primary key with duplicate values or null
           values.
*Action:   remove the duplicates and null values before enabling a primary
           key.
```

- If a column of a table has a NULL value and if we try to add primary key to that column. Then is will show the below error.

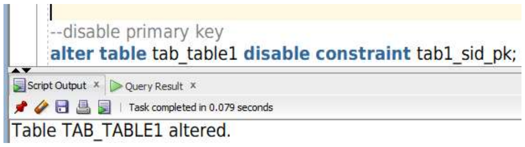


```
alter table hari_function add constraint const_hari_function primary key(id);
```

Task completed in 0.087 seconds

```
Error starting at line : 39 in command -
alter table hari_function add constraint const_hari_function primary key(id)
Error report -
ORA-01449: column contains NULL values; cannot alter to NOT NULL
01449. 00000 -  "column contains NULL values; cannot alter to NOT NULL"
*Cause:
*Action:
```

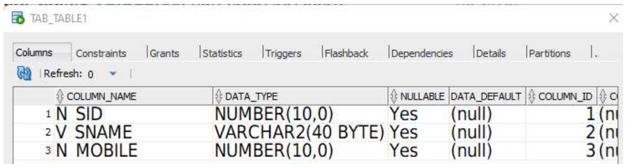
DISABLE PRIMARY KEY:-



```
--disable primary key
alter table tab_table1 disable constraint tab1_sid_pk;
```

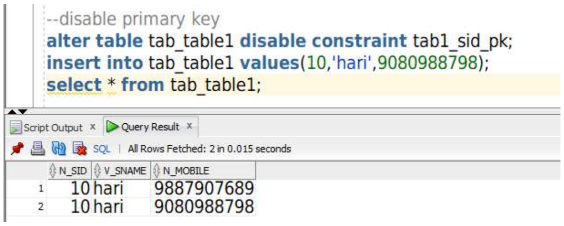
Task completed in 0.079 seconds

```
Table TAB_TABLE1 altered.
```



```
    COLUMN_NAME   DATA_TYPE          NULLABLE DATA_DEFAULT COLUMN_ID C
  1 N_SID         NUMBER(10,0)       Yes      (null)               1 (nu
  2 V_SNAME       VARCHAR2(40 BYTE)  Yes      (null)               2 (nu
  3 N_MOBILE      NUMBER(10,0)       Yes      (null)               3 (nu
```

AFTER DIPLABLE primary key



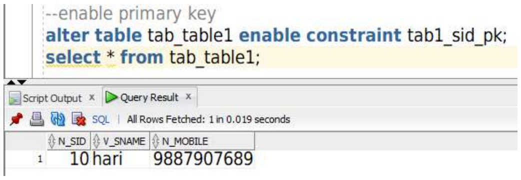
```
--disable primary key
alter table tab_table1 disable constraint tab1_sid_pk;
insert into tab_table1 values(10,'hari',9080988798);
select * from tab_table1;
```

All Rows Fetched: 2 in 0.015 seconds

```
    N_SID V_SNAME N_MOBILE
  1    10 hari    9887907689
  2    10 hari    9080988798
```

Once primary key column of a table is disable then we can insert duplicate an null values.

Enable primary key:-



```
--enable primary key
alter table tab_table1 enable constraint tab1_sid_pk;
select * from tab_table1;
```

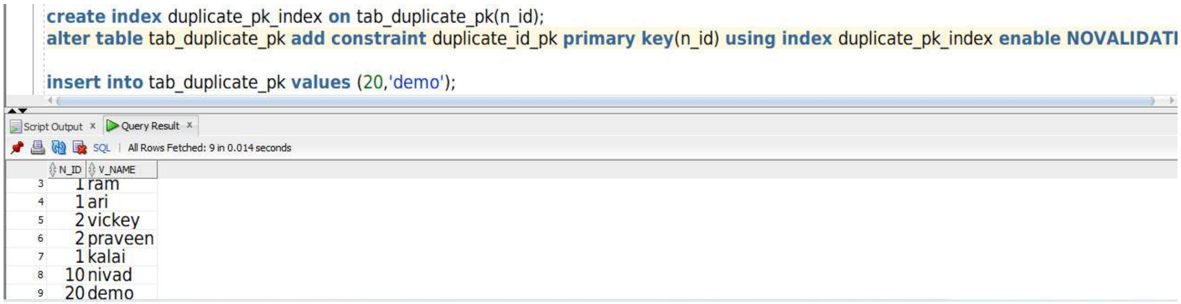
All Rows Fetched: 1 in 0.019 seconds

```
    N_SID V_SNAME N_MOBILE
  1    10 hari    9887907689
```

To add primary key to a exiting value even it has duplicative values:-

create index duplicate_pk_index on tab_duplicate_pk(n_id);

alter table tab_duplicate_pk add constraint duplicate_id_pk primary key(n_id) using index duplicate_pk_index enable NOVALIDATE;

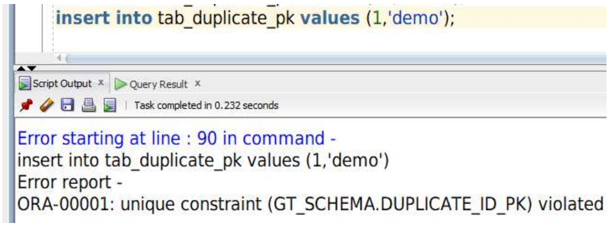


```sql
create index duplicate_pk_index on tab_duplicate_pk(n_id);
alter table tab_duplicate_pk add constraint duplicate_id_pk primary key(n_id) using index duplicate_pk_index enable NOVALIDATI

insert into tab_duplicate_pk values (20,'demo');
```

All Rows Fetched: 9 in 0.014 seconds

	N_ID	V_NAME
3	1	ram
4	1	ari
5	2	vickey
6	2	praveen
7	1	kalai
8	10	nivad
9	20	demo

Screen clipping taken: 03-08-2022 03:21 PM



```sql
insert into tab_duplicate_pk values (1,'demo');
```

Task completed in 0.232 seconds

```
Error starting at line : 90 in command -
insert into tab_duplicate_pk values (1,'demo')
Error report -
ORA-00001: unique constraint (GT_SCHEMA.DUPLICATE_ID_PK) violated
```

9. **Foreign Key**

Foreign key:-

Foreign key constraint is known as referential integrity constraint.
We need two table, a parent table and child table.
Primary key 0r unique key Columns of a parent table can be only in foreign key constraint. I,e, apart for primary key or unique column in a parents, other column can't be made as foreign key in the child table.
The datatype and size of a primary key column should be same as foreign.
There are two types :-

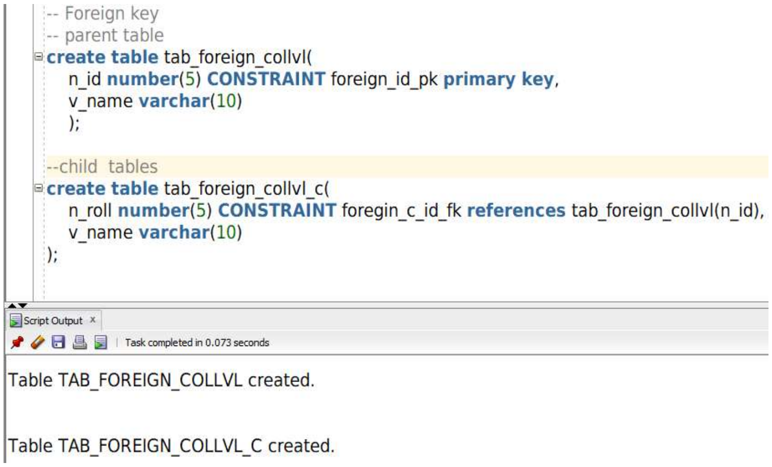
A. Simple foreign key
B. Composite foreign key

Types of statement :-

a. Create table statement
   i. At column level
   ii. At table level

b. Alter table statement

A. Simple foreign key:-
   a. Create table:-
   b. At column level-

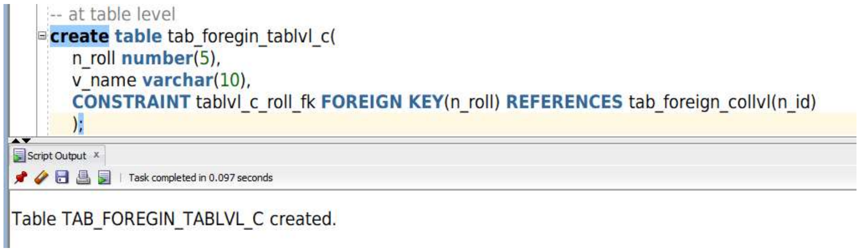


```sql
-- Foreign key
-- parent table
create table tab_foreign_collvl(
    n_id number(5) CONSTRAINT foreign_id_pk primary key,
    v_name varchar(10)
    );

--child  tables
create table tab_foreign_collvl_c(
    n_roll number(5) CONSTRAINT foregin_c_id_fk references tab_foreign_collvl(n_id),
    v_name varchar(10)
);
```

Task completed in 0.073 seconds

```
Table TAB_FOREIGN_COLLVL created.


Table TAB_FOREIGN_COLLVL_C created.
```

i. At table level:-

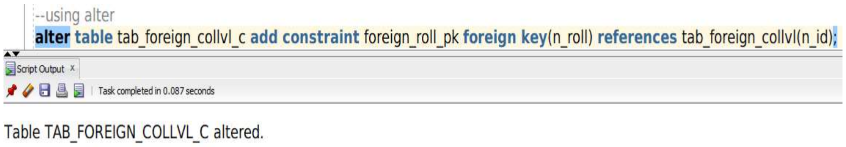


```sql
-- at table level
create table tab_foregin_tablvl_c(
    n_roll number(5),
    v_name varchar(10),
    CONSTRAINT tablvl_c_roll_fk FOREIGN KEY(n_roll) REFERENCES tab_foreign_collvl(n_id)
    );
```

Task completed in 0.097 seconds

```
Table TAB_FOREGIN_TABLVL_C created.
```

a. Alter table:-



```
--using alter
alter table tab_foreign_collvl_c add constraint foreign_roll_pk foreign key(n_roll) references tab_foreign_collvl(n_id);
```

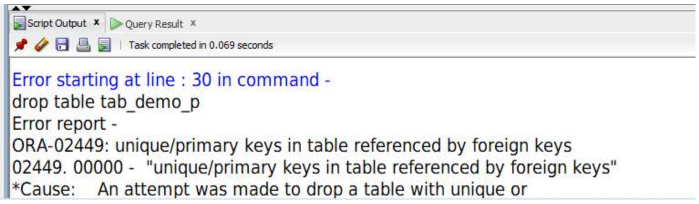
```
Table TAB_FOREIGN_COLLVL_C altered.
```

B. Compositive key:-

ERROR:-

Parent table	Child table	Error	output	desc
Primary key - id number(10)	Foreign key - roll number(10)	-	created	Datatype & size same as primary key
Unique -id number(10)	Foreign key - roll number(10)	-	Created	Datatype & size same as unique key
Primary key id number(10)	Foreign key - roll number(5)	Error while inserting	value larger	Difference in size.
Primary key id number(10) Screen clipping taken: 03-08-2022 02:40 PM	Foreign key - roll varchar(10)	Incompatible error	column type incompatible with referenced column type	Difference in datatype
Primary key id number(10)	Foreign key - roll float(10)	integrity constrain	integrity constrain violated - parent key not found	Difference in datatype

- If we try to drop a table which have been referred(foreign key reference) by another table then the base table can't be deleted.



```
Error starting at line : 30 in command -
drop table tab_demo_p
Error report -
ORA-02449: unique/primary keys in table referenced by foreign keys
02449. 00000 -  "unique/primary keys in table referenced by foreign keys"
*Cause:    An attempt was made to drop a table with unique or
```

10. Dynamic SQL

```
create table hari_emp (emp_id number,emp_name
varchar(20),emp_number number(10));
insert into hari_emp values(1003,'ram',9080809080);

--native dynamic sql
declare
    v_sql_query varchar2(200);
    n_number number(10);
begin
    v_sql_query := 'select count(*) from hari_emp';
    EXECUTE IMMEDIATE v_sql_query into n_number;
    -- or replace 8 & 9 to     ->   execute immediate 'select
count(*) from hari_emp' into hari_emp;    -- both are same
    DBMS_OUTPUT.PUT_LINE('NUMBER OF EMPLOYEES '||n_number);
end;



-- trying to create a table in plsql directly
begin
    create table tn(id number(5));
end;
    -- error -> PLS-00103: Encountered the symbol "CREATE" when
expecting one of the following:

-- Then how to create a table in plsql
declare
    v_query varchar(100);
begin
    v_query := 'create table tn( id number(10))';
    EXECUTE IMMEDIATE v_query;
end;
desc tn;
```