2022

SUBJECT:

DATABASE MANAGEMENT SYSTEM (MINI PROJECT)

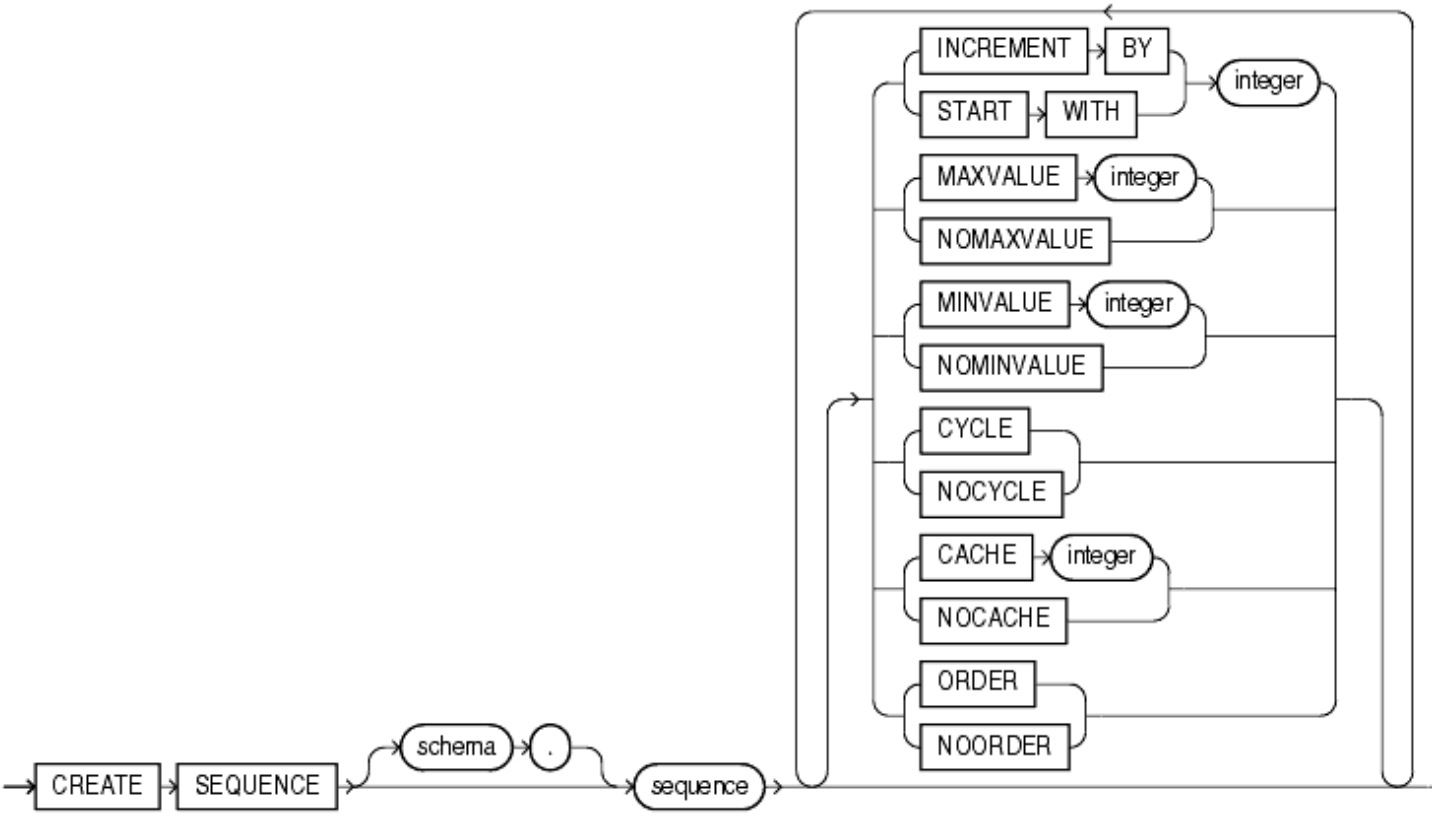
TOPIC:

TRANSPORT COMPANY + QUERIES ON SEQUENCES,PROCEDURE, FUNCTION, TRIGGER & CURSOR.

SEQUENCES (AUTONUMBER) :

A sequence is an Oracle object that is used to generate a number sequence. This can be useful when you need to create a unique number asa primary key.

create_sequence::=



COMPONENTS :

INCREMENT BY

Specify the interval between sequence numbers. This integer value can be any positive or negative integer, but it cannot be 0. This value canhave 28 or fewer digits.

START WITH

Specify the first sequence number to be generated. Use this clause tostart an ascending sequence at a value greater than its minimum or to start a descending sequence at a value less than its maximum.

MAXVALUE

Specify the maximum value the sequence can generate. This integervalue can have 28 or fewer digits. MAXVALUE must be equal to or greater than START WITH and must be greater than MINVALUE.

MINVALUE

Specify the minimum value of the sequence. This integer value canhave 28 or fewer digits. MINVALUE must be less than or equal
to START WITH and must be less than MAXVALUE.

CYCLE

Specify CYCLE to indicate that the sequence continues to generate values after reaching either its maximum or minimum value. After an ascending sequence reaches its maximum value, it generates its minimumvalue. After a descending sequence reaches its minimum, it generates its maximum value.

CACHE

Specify how many values of the sequence the database preallocates and keeps in memory for faster access. This integer value can have 28 or fewer digits. The minimum value for this parameter is 2. For sequences thatcycle, this value must be less than the number of values in the cycle. You cannot cache more values than will fit in a given cycle of sequence numbers.

NOCACHE

Specify NOCACHE to indicate that values of the sequence are not preallocated. If you omit both CACHE and NOCACHE, the databasecaches 20 sequence numbers by default.

- Creating table and putting numbers using

sequences.CREATE TABLE transport1(
sr_no number(5)
NOT NULL,name
varchar2(20),
mobile
number(11),
pick_up
varchar2(20),
total_tickets
number(3),

amount_paid
number(8,2)

CREATE SEQUENCE sr_no

start with 101

increment by 1

maxvalue 107

min
val
ue
10
1
noc
ach
e;

CREATE
SEQUENCE mob_no
start with
9221523422
increment by 1

maxvalue 9221523428

minvalue 9221523422;

insert into transport1
values(sr_no.nextval,'Abhishek',mob_no.nextval,'Mulund',
2,400.00);

insert into transport1
values(sr_no.nextval,'Rubina',mob_no.nextval,'Sion',
4,800.00);insert into transport1

values(sr_no.nextval,'Rahul',mob_no.nextval,'Dadar',
1,200.00);insert into transport1
values(sr_no.nextval,'Yashika',mob_no.nextval,'King Circle',2,400.00);

insert into transport1
values(sr_no.nextval,'Bhawarth',mob_no.nextval,'Khar',
4,800.00);

insert into transport1
values(sr_no.nextval,'Raju',mob_no.nextval,'Royal
Circus',1,200.00);insert into transport1

values(sr_no.nextval,'Shyam',mob_no.nextval,'Royal Circus',3,600.00);

```
SQL> set tab off;
SQL> set trimout on;
SQL> set lines 256;
SQL> select * from transport1;

     SR_NO NAME                    MOBILE PICK_UP             TOTAL_TICKETS AMOUNT_PA
---------- -------------------- ---------- -------------------- ------------- ---------
       101 Abhishek             9221523422 Mulund                        2         4
       102 Rubina               9221523423 Sion                          4         8
       103 Rahul                9221523424 Dadar                         1         2
       104 Yashika              9221523425 King Circle                   2         4
       105 Bhawarth             9221523426 Khar                          4         8
       106 Raju                 9221523427 Royal Circus                  1         2
       107 Shyam                9221523428 Royal Circus                  3         6

7 rows selected.
```

- Alter the sr_no sequence to make an

increment by 2.alter sequence sr_no
incre
ment
by 2
cycl
e;

```
SQL> alter sequence sr_no
  2  increment by 2
  3  cycle;

Sequence altered.
```

```
SQL> select * from transport2;

    SR_NO NAME                 MOBILE PICK_UP             TOTAL_TICKETS AMOUNT_PAI
--------- -------------------- ---------- -------------------- ------------ ----------
      103 Abhishek             9221523422 Mulund                        2         40
      105 Rubina               9221523423 Sion                          4         80
      107 Rahul                9221523424 Dadar                         1         20
      101 Yashika              9221523425 King Circle                   2         40
      103 Bhawarth             9221523426 Khar                          4         80
      105 Raju                 9221523427 Royal Circus                  1         20
      107 Shyam                9221523428 Royal Circus                  3         60

7 rows selected.
```

- Drop the mobile no sequence and re create a new one.

drop sequence mob_no;

CREATE
SEQUENCE mob_no
start with
8108123456
increment by 1

maxvalue 8108123462

minvalue 8108123456;

```
SQL> CREATE SEQUENCE mob_no
  2  start with 8108123456
  3  increment by 1
  4  maxvalue 8108123462
  5  minvalue 8108123456;

Sequence created.
```

```
SQL> drop sequence mob_no;

Sequence dropped.
```

```
SQL> select * from transport2;

     SR_NO NAME                    MOBILE PICK_UP             TOTAL_TICKETS AMOUNT_PAI
---------- -------------------- ---------- -------------------- ------------- ----------
       101 Abhishek             8108123456 Mulund                        2         40
       103 Rubina               8108123457 Sion                          4         80
       105 Rahul                8108123458 Dadar                         1         20
       107 Yashika              8108123459 King Circle                   2         40
       101 Bhawarth             8108123460 Khar                          4         80
       103 Raju                 8108123461 Royal Circus                  1         20
       105 Shyam                8108123462 Royal Circus                  3         60

7 rows selected.
```

- Alter the sequence sr_no to make increment by 3alter sequence sr_no increment by 3;

```
SQL> alter sequence sr_no
  2  increment by 3;

Sequence altered.
```

```
SQL> select * from transport2;

     SR_NO NAME                    MOBILE PICK_UP             TOTAL_TICKETS AMOUNT_PA
---------- -------------------- ---------- -------------------- ------------ ---------
       104 Abhishek             8108123456 Mulund                          2         4
       107 Rubina               8108123457 Sion                            4         8
       101 Rahul                8108123458 Dadar                           1         2
       104 Yashika              8108123459 King Circle                     2         4
       107 Bhawarth             8108123460 Khar                            4         8
       101 Raju                 8108123461 Royal Circus                    1         2
       104 Shyam                8108123462 Royal Circus                    3         6

7 rows selected.
```

- Alter sequence mob_no to increase maximum value and makeincrement by 3.

alter
sequence
mob_no
increment by
3
maxvalue 8108123480;

```
SQL> alter sequence mob_no
  2  increment by 3
  3  maxvalue 8108123480;

Sequence altered.

SQL> select * from transport2;

     SR_NO NAME                    MOBILE PICK_UP             TOTAL_TICKETS AMOUNT_PAID
---------- -------------------- ---------- -------------------- ------------ -----------
       107 Abhishek             8108123465 Mulund                          2         400
       101 Rubina               8108123468 Sion                            4         800
       104 Rahul                8108123471 Dadar                           1         200
       107 Yashika              8108123474 King Circle                     2         400
       101 Bhawarth             8108123477 Khar                            4         800
       104 Raju                 8108123480 Royal Circus                    1         200

6 rows selected.
```

PROCEDURE :

A PL/SQL procedure is a named block that does a specific task. PL/SQL procedure allows you to encapsulate complex business logic andreSyntax:

CREATE [OR REPLACE] **PROCEDURE**
procedure_name[(parameter [,parameter])]
IS

[declaration_section]

BEGIN

executab
le_section
[EXCEPTIO
N
exception_section]

END [procedure_name];

use it in both database layer and application layer.

By using this Transport1 table we will perform all the queries.

```
SQL> select * from transport1;

    SR_NO NAME                 MOBILE PICK_UP            TOTAL_TICKETS AMOUNT_PAI
--------- -------------------- ---------- -------------------- ------------ ----------
        1 Abhishek             8108123456 Mulund                        2         40
        2 Rubina               8108123457 Sion                          4         80
        3 Rahul                8108123458 Dadar                         1         20
        4 Yashika              8108123459 King Circle                   2         40
        5 Bhawarth             8108123460 Khar                          4         80
        6 Raju                 8108123461 Royal Circus                  1         20
        7 Shyam                8108123462 Royal Circus                  3         60

7 rows selected.
```

- Create a procedure for insertion of new data.

```
CREATE OR REPLACE PROCEDURE insertuser(sr_no in
number,namein varchar2,mob_no in number,pick_up in
varchar2,total_tickets in number,amount_paid in number) IS
BEGIN
insert into transport1
values(sr_no,name,mob_no,pick_up,total_tickets,amo
unt_paid);
end insertuser;
/

B
E
G
I
N
insertuser(8,'Tipendra',8108123481,'Goregoan',
3,600.00);dbms_output.put_line('record
inserted');
END;
/
```

```
SQL> CREATE OR REPLACE PROCEDURE insertuser(sr_no in number,name in varchar2,mob_no in number,pick_up in varchar2,total_tickets in number,amou
  2  BEGIN
  3  insert into transport1 values(sr_no,name,mob_no,pick_up,total_tickets,amount_paid);
  4  end insertuser;
  5  /

Procedure created.

SQL> BEGIN
  2  insertuser(8,'Tipendra',8108123481,'Goregoan',3,600.00);
  3  dbms_output.put_line('record inserted');
  4  END;
  5  /
record inserted

PL/SQL procedure successfully completed.
```

```
SQL> select * from transport1;

     SR_NO NAME                     MOBILE PICK_UP              TOTAL_TICKETS AMOUNT_PA
---------- -------------------- ---------- -------------------- ------------- ---------
         1 Abhishek             8108123456 Mulund                           2         4
         2 Rubina               8108123457 Sion                             4         8
         3 Rahul                8108123458 Dadar                            1         2
         4 Yashika              8108123459 King Circle                      2         4
         5 Bhawarth             8108123460 Khar                             4         8
         6 Raju                 8108123461 Royal Circus                     1         2
         7 Shyam                8108123462 Royal Circus                     3         6
         8 Tipendra             8108123481 Goregoan                         3         6

8 rows selected.
```

- Create a procedure to update pick up point to ‘Andheri’ by taking sr_noas parameter.

CREATE OR REPLACE PROCEDURE updatedata(sr_no in number)ISv_num number(3);
BEGIN

v_num :=
sr_no;
UPDATE
transport1
SET pick_up
= 'Andheri'
WHERE sr_no
= v_num;end
updatedata;
/

B
E
G
I
N

updatedata(8);
dbms_output.put_line('record
updated!');end;
/

```
SQL> CREATE OR REPLACE PROCEDURE updatedata(sr_no in number)IS
  2  v_num number(3);
  3  BEGIN
  4  v_num := sr_no;
  5  UPDATE transport1
  6  SET pick_up = 'Andheri'
  7  WHERE sr_no =  v_num;
  8  end updatedata;
  9  /

Procedure created.

SQL> BEGIN
  2  updatedata(8);
  3  dbms_output.put_line('record updated!');
  4  end;
  5  /
record updated!
```

```
SQL> select * from transport1;

    SR_NO NAME                    MOBILE PICK_UP             TOTAL_TICKETS AMOUNT_PAID
--------- -------------------- ---------- -------------------- ------------- -----------
        1 Abhishek             8108123456 Mulund                        2         400
        2 Rubina               8108123457 Sion                          4         800
        3 Rahul                8108123458 Dadar                         1         200
        4 Yashika              8108123459 King Circle                   2         400
        5 Bhawarth             8108123460 Khar                          4         800
        6 Raju                 8108123461 Royal Circus                  1         200
        7 Shyam                8108123462 Royal Circus                  3         600
        8 Tipendra             8108123481 Andheri                       3         600

8 rows selected.
```

- Create a procedure to update the number of tickets by taking pickuppoint as input from users.

CREATE OR REPLACE PROCEDURE updatetickets(pick_up in

```
varchar2)IS
v_name
varchar2(25
);BEGIN
v_name  :=
pick_up;
UPDATE
transport1
SET
total_ticket
s = 3

Where pick_up
= v_name;end
updatetickets;
/

B
E
G
I
N
updatetickets('Royal Circus');
dbms_output.put_line('record
updated!');end;
/
```

```
SQL> CREATE OR REPLACE PROCEDURE updatetickets(pick_up in varchar2) IS
  2  v_name varchar2(25);
  3  BEGIN
  4  v_name := pick_up;
  5  UPDATE transport1
  6  SET total_tickets = 3
  7  Where pick_up = v_name;
  8  end updatetickets;
  9  /

Procedure created.

SQL> BEGIN
  2  updatetickets('Royal Circus');
  3  dbms_output.put_line('record updated!');
  4  end;
  5  /
record updated!

PL/SQL procedure successfully completed.

SQL> select * from transport1;

    SR_NO NAME                 MOBILE PICK_UP             TOTAL_TICKETS AMOUNT_PAID
--------- -------------------- ---------- -------------------- ------------ -----------
        1 Abhishek             8108123456 Mulund                        2         400
        2 Rubina               8108123457 Sion                          4         800
        3 Rahul                8108123458 Dadar                         1         200
        4 Yashika              8108123459 King Circle                   2         400
        5 Bhawarth             8108123460 Khar                          4         800
        6 Raju                 8108123461 Royal Circus                  3         200
        7 Shyam                8108123462 Royal Circus                  3         600
        8 Tipendra             8108123481 Andheri                       3         600

8 rows selected.
```

- Create a procedure to update the amount paid for updated number oftickets.

CREATE OR REPLACE PROCEDURE updateprice(total_tickets innumber) IS

v_num
number(
3);

```
BEGIN
v_num :=
total_tickets;
UPDATE
transport1
SET amount_paid =
v_num*200Where
total_tickets = v_num;
end updateprice;
/

B
E
G
I
N
updateprice(3);
dbms_output.put_line('Record
updated!');end;
/
```

```
SQL> CREATE OR REPLACE PROCEDURE updateprice(total_tickets in number) IS
  2  v_num number(3);
  3  BEGIN
  4  v_num := total_tickets;
  5  UPDATE transport1
  6  SET amount_paid = v_num*200
  7  Where total_tickets = v_num;
  8  end updateprice;
  9  /

Procedure created.

SQL> BEGIN
  2  updateprice(3);
  3  dbms_output.put_line('Record updated!');
  4  end;
  5  /
Record updated!

PL/SQL procedure successfully completed.
```

```
SQL> select * from transport1;

    SR_NO NAME                    MOBILE PICK_UP             TOTAL_TICKETS AMOUNT_PA
--------- -------------------- ---------- -------------------- ------------- ---------
        1 Abhishek             8108123456 Mulund                        2         4
        2 Rubina               8108123457 Sion                          4         8
        3 Rahul                8108123458 Dadar                         1         2
        4 Yashika              8108123459 King Circle                   2         4
        5 Bhawarth             8108123460 Khar                          4         8
        6 Raju                 8108123461 Royal Circus                  3         6
        7 Shyam                8108123462 Royal Circus                  3         6
        8 Tipendra             8108123481 Andheri                       3         6

8 rows selected.
```

- Create a procedure to delete the data by taking sr no as input fromusers.

CREATE OR REPLACE PROCEDURE deletedata(sr_no in number) ISv_num number(3);
BEGIN

```
v_num :=
sr_no;
DELETE from
transport1
Where sr_no =
v_num;

end deletedata;
/

B
E
G
I
N
deletedata(8);
dbms_output.put_line('Record
deleted!');end;
/
```

```
SQL> CREATE OR REPLACE PROCEDURE deletedata(sr_no in number) IS
  2  v_num number(3);
  3  BEGIN
  4  v_num := sr_no;
  5  DELETE from transport1
  6  Where sr_no = v_num;
  7  end deletedata;
  8  /

Procedure created.

SQL> BEGIN
  2  deletedata(8);
  3  dbms_output.put_line('Record deleted!');
  4  end;
  5  /
Record deleted!

PL/SQL procedure successfully completed.

SQL> select * from transport1;

     SR_NO NAME                 MOBILE PICK_UP             TOTAL_TICKETS AMOUNT_PAI
---------- -------------------- ---------- -------------------- ------------ ----------
         1 Abhishek             8108123456 Mulund                        2         40
         2 Rubina               8108123457 Sion                          4         80
         3 Rahul                8108123458 Dadar                         1         20
         4 Yashika              8108123459 King Circle                   2         40
         5 Bhawarth             8108123460 Khar                          4         80
         6 Raju                 8108123461 Royal Circus                  3         60
         7 Shyam                8108123462 Royal Circus                  3         60

7 rows selected.
```

Function :

The PL/SQL Function is very similar to PL/SQL Procedure. The maindifference between procedure and a function is, a function must always return a value, and on the other hand a procedure may or may not return avalue. Except this, all the other things of PL/SQL procedure are true for PL/SQL function too.

Syntax:

```
CREATE [OR REPLACE] FUNCTION function_name
[ (parameter[,parameter]) ]

RETURN
return_datatype IS
| AS
[declaration_secti
on]BEGIN
executab
le_sectio
n
[EXCEP
TION
 exception_
section]
END
[function_na
me];
/
```

- Create a pl/sql function to return the count of total number of data. CREATE OR REPLACE FUNCTION countfunc RETURN number IStotal_count number(2) := 0;
BEGIN

SELECT count(*) into total_count FROM
transport1;RETURN total_count;
END countfunc;

/
D
E
C
L
A

R
E

num
numb
er(2);
BEGI
N
num := countfunc();

dbms_output.put_line('Total no. of registrations:
' || num);END;
/

```
SQL> CREATE OR REPLACE FUNCTION countfunc RETURN number IS
  2  total_count number(2) := 0;
  3  BEGIN
  4  SELECT count(*) into total_count FROM transport1;
  5  RETURN total_count;
  6  END countfunc;
  7  /

Function created.

SQL> DECLARE
  2  num number(2);
  3  BEGIN
  4  num := countfunc();
  5  dbms_output.put_line('Total no. of registrations: ' || num)
  6  END;
  7  /
Total no. of registrations: 7

PL/SQL procedure successfully completed.
```

- Create a pl/sql function to return the count of passengers from theirpickup point .

```
CREATE OR REPLACE FUNCTION count_people(pick_up in varchar2)
RETURN
number AS
v_count
number(5);
v_places
varchar2(50);
BEGIN

v_places := pick_up;

SELECT COUNT(sr_no) INTO v_count from transport1 where
pick_up =v_places;

return
v_count;
end
count_p
eople;
/

B
E
G
I
N
dbms_output.put_line('Total students is
'||count_people('&pick_up'));end;
/
```

- Create a plsql function to return the value of maximum amount paid.

```
CREATE OR REPLACE FUNCTION max_amt RETURN number AS

v_max
```

number(
5);
BEGIN
SELECT MAX(amount_paid) INTO v_max from
transport1;return v_max;
end max_amt;

/

B
E
G
I
N
dbms_output.put_line('Maximum amount paid is
'||max_amt());end;
/

```
PL/SQL procedure successfully completed.

SQL> CREATE OR REPLACE FUNCTION max_amt RETURN number AS
  2  v_max number(5);
  3  BEGIN
  4  SELECT MAX(amount_paid) INTO v_max from transport1;
  5  return v_max;
  6  end max_amt;
  7  /

Function created.

SQL>
SQL> BEGIN
  2  dbms_output.put_line('Maximum amount paid is '||max_amt());
  3  end;
  4  /
Maximum amount paid is 800

PL/SQL procedure successfully completed.
```

- Create a plsql function to return the value of minimum amount paid.

```
CREATE OR REPLACE FUNCTION min_amt RETURN number AS

v_min
number
(5);
BEGIN
SELECT MIN(amount_paid) INTO v_min from
transport1;return v_min;
end min_amt;

/


B
E
G
I
N
dbms_output.put_line('Minimum amount paid is
'||min_amt());end;
/
```

```
SQL> CREATE OR REPLACE FUNCTION min_amt RETURN number AS
  2  v_min number(5);
  3  BEGIN
  4  SELECT MIN(amount_paid) INTO v_min from transport1;
  5  return v_min;
  6  end min_amt;
  7  /

Function created.

SQL> BEGIN
  2  dbms_output.put_line('Minimum amount paid is '||min_amt());
  3  end;
  4  /
Minimum amount paid is 200

PL/SQL procedure successfully completed.
```

- Create a plsql function to return the value of sum of total amount paid byall passegers.

CREATE OR REPLACE FUNCTION sumdata RETURN number AS

v_sum
number(
7);
BEGIN

SELECT SUM(amount_paid) INTO v_sum from transport1;return v_sum;
end sumdata;

/

B
E
G
I

```
dbms_output.put_line('The sum of total amount paid is
'||sumdata());end;
/
```

```
SQL> CREATE OR REPLACE FUNCTION sumdata RETURN number AS
  2  v_sum number(7);
  3  BEGIN
  4  SELECT SUM(amount_paid) INTO v_sum from transport1;
  5  return v_sum;
  6  end sumdata;
  7  /

Function created.

SQL> BEGIN
  2  dbms_output.put_line('The sum of total amount paid is '||sumdata());
  3  end;
  4  /
The sum of total amount paid is 3800

PL/SQL procedure successfully completed.
```

TRIGGERS :

Triggers are stored programs, which are automatically executed or fired when some events occur. Triggers are, in fact, written to be executedin response to any of the following events –

A **database manipulation (DML)** statement (DELETE,

INSERT, orUPDATE)

A **database definition (DDL)** statement (CREATE, ALTER, or DROP).

A **database operation** (SERVERERROR, LOGON, LOGOFF, STARTUP,or SHUTDOWN).

Triggers can be defined on the table, view, schema, or database withwhich the event is associated.

CREATE [OR REPLACE] TRIGGER trigger_name

{BEFORE | AFTER | INSTEAD OF } {INSERT [OR] | UPDATE [OR] | DELETE} [OF col_name] ON
table_name[REFERENCING
OLD AS o NEW AS n] [FOR
EACH ROW]
WHEN (condition)

DECLARE
Declaration-statements
BEGIN
Executable-statements
END;

- Create a plsql trigger block to execute when data is inserted.

CREATE TRIGGER
trigg_namebefore
insert

on
tra
ns
por
t1
for
ea

ch
ro
w

WHEN(ne
w.name>0)
BEGIN
dbms_output.put_line('New row inserted
successfully!');end;
/

insert into transport1 values(8,'Jethalal',8108123490,'Goregoan',2,400.00);

```
SQL> CREATE OR REPLACE TRIGGER trigger_insert
  2  BEFORE
  3  INSERT on transport1
  4  for each row
  5  when(new.sr_no>0)
  6  begin
  7  dbms_output.put_line('New data inserted!');
  8  end;
  9  /

Trigger created.

SQL> insert into transport1
  2  values(8,'Jetha',8108123491,'Goregoan',4,800.00);
New data inserted!

1 row created.
```

```
SQL> select * from transport1;

    SR_NO NAME                 MOBILE PICK_UP             TOTAL_TICKETS AMOUNT_PAID
--------- -------------------- ---------- -------------------- ------------ -----------
        1 Abhishek             8108123456 Mulund                        2         400
        2 Rubina               8108123457 Sion                          4         800
        3 Rahul                8108123458 Dadar                         1         200
        4 Yashika              8108123459 King Circle                   2         400
        5 Bhawarth             8108123460 Khar                          4         800
        6 Raju                 8108123461 Royal Circus                  1         200
        7 Shyam                8108123462 Royal Circus                  3         600
        8 Jetha                8108123491 Goregoan                      4         800

8 rows selected.
```

- Create a plsql trigger block to execute when data is updated.

```
CREATE OR REPLACE TRIGGER
trigger_updateBEFORE
UPDATE on
transport1for
each row
when(new.sr
_no>0) begin
dbms_output.put_line('Data
updated!');end;
/

B
E
G
I
N
UPDATE
transport1
```

SET pick_up
= 'Dharavi'
WHERE
sr_no = 7;
END;
/

```
SQL> CREATE OR REPLACE TRIGGER trigger_update
  2  BEFORE
  3  UPDATE on transport1
  4  for each row
  5  when(new.sr_no>0)
  6  begin
  7  dbms_output.put_line('Data updated!');
  8  end;
  9  /

Trigger created.
```

```
SQL> BEGIN
  2  UPDATE transport1
  3  SET pick_up = 'Dharavi'
  4  WHERE sr_no = 7;
  5  END;
  6  /
Data updated!
```

```
SQL> select * from transport1;

    SR_NO NAME                 MOBILE PICK_UP            TOTAL_TICKETS AMOUNT_P
--------- -------------------- ---------- -------------------- ------------ --------
        1 Abhishek             8108123456 Mulund                        2
        2 Rubina               8108123457 Sion                          4
        3 Rahul                8108123458 Dadar                         1
        4 Yashika              8108123459 King Circle                   2
        5 Bhawarth             8108123460 Khar                          4
        6 Raju                 8108123461 Royal Circus                  1
        7 Shyam                8108123462 Dharavi                       3
        8 Jetha                8108123491 Goregoan                      4

8 rows selected.
```

- Create a plsql trigger block to stop the execution when the total numberof tickets exceeds its limit.

```
CREATE TRIGGER
exceedticketBEFORE
INSERT or UPDATE on
transport1for each row
BEGIN

IF (select
sum(total_tickets))>22 THEN
RAISE
number_limit_exceed;
END;

/
```

```
SQL> CREATE TRIGGER exceedticket
  2  BEFORE
  3  INSERT or UPDATE on transport1
  4  for each row
  5  BEGIN
  6  IF (select sum(total_tickets))>22 THEN
  7  RAISE number_limit_exceed;
  8  END;
  9  /
```

```
SQL> insert into transport1
  2  values(9,'Tapu',8108123424,'Sion',2,400.00);
insert into transport1
            *
ERROR at line 1:
ORA-04098: trigger 'SYSTEM.EXCEEDTICKET' is invalid and failed re-validation
```

```
SQL> UPDATE transport1
  2  SET total_tickets = 3
  3  WHERE sr_no = 6;
UPDATE transport1
       *
ERROR at line 1:
ORA-04098: trigger 'SYSTEM.EXCEEDTICKET' is invalid and failed re-validation
```

- Create a plsql trigger block to stop execution of program wheninappropriate amount paid.

CREATE or REPLACE TRIGGER
check_paymentBEFORE
INSERT or UPDATE on
transport1for each row
BEGIN

IF (new.total_tickets = 1 and new.amount_paid!=
200) thenraise check_payment;
ELSIF (new.total_tickets = 2 and new.amount_paid!=
400) thenraise check_payment;
ELSIF (new.total_tickets = 3 and new.amount_paid!=
600) thenraise check_payment;
ELSIF (new.total_tickets = 4 and new.amount_paid!=
800) thenraise check_payment;

e
n
d

i
f
;
e

n
d
;

```
SQL> CREATE or REPLACE TRIGGER check_payment
  2  BEFORE
  3  INSERT or UPDATE on transport1
  4  for each row
  5  BEGIN
  6  IF (new.total_tickets = 1 and new.amount_paid!= 200) then
  7  raise check_payment;
  8  ELSIF (new.total_tickets = 2 and new.amount_paid!= 400) then
  9  raise check_payment;
 10  ELSIF (new.total_tickets = 3 and new.amount_paid!= 600) then
 11  raise check_payment;
 12  ELSIF (new.total_tickets = 4 and new.amount_paid!= 800) then
 13  raise check_payment;
 14  end if;
 15  end;
 16  /

Warning: Trigger created with compilation errors.
```

```
SQL> insert into transport1
  2  values(8,'Gitesh',8108123487,'Jogeshwari',2,100);
insert into transport1
            *
ERROR at line 1:
ORA-04098: trigger 'SYSTEM.CHECK_PAYMENT' is invalid and failed re-validati


SQL> UPDATE transport1
  2  SET total_tickets = 2 where sr_no = 6;
UPDATE transport1
       *
ERROR at line 1:
ORA-04098: trigger 'SYSTEM.CHECK_PAYMENT' is invalid and failed re-validatio
```

- Create plsql trigger block to be executes when data is deleted.CREATE OR REPLACE TRIGGER

data_deleted
BEFORE DELETE on
transport1for each row
BEGIN

dbms_output.put_line('Data
Deleted!');END;
/

```
SQL> CREATE OR REPLACE TRIGGER data_deleted
  2  BEFORE DELETE on transport1
  3  for each row
  4  BEGIN
  5  dbms_output.put_line('Data Deleted!');
  6  END;
  7  /

Trigger created.

SQL> DELETE from transport1
  2  Where sr_no = 8;
Data Deleted!

1 row deleted.
```

CURSOR :

A **cursor** is a pointer to this context area. PL/SQL controls the context area through a cursor. A cursor holds the rows (one or more) returned by a SQL statement. The set of rows the cursor holds is referredto as the **active set**.

You can name a cursor so that it could be referred to in a program tofetch and process the rows returned by the SQL statement, one at a time. There are two types of cursors –

Implicit cursors : Implicit cursors are automatically created by Oracle whenever an SQL statement is executed, when there is no explicit cursor for the statement. Programmers cannot control the implicit cursors and theinformation in it.

Explicit cursors : Explicit cursors are programmer-defined cursors for gaining more control over the **context area**. An explicit cursor should bedefined in the declaration section of the PL/SQL Block. It is created on aSELECT Statement which returns more than one row.

Syntax includes:

- **Declaring the Cursor**
- **Opening the Cursor**
- **Fetching the Cursor**
- **Closing the Cursor**

- Create a plsql cursor to display the sr_no, name and mobile number'scolumns.

DECLARE

cursor traveller is SELECT sr_no, name, mobile from transport1;t_sr_no transport1.sr_no%type;
t_name
transport1.name%type;
t_mobile
transport1.mobile%type;

```
BEGIN
OPE
N
travel
ler;
LOO
P
FETCH traveller into t_sr_no, t_name, t_mobile;

EXIT WHEN traveller%notfound;
dbms_output.put_line(t_sr_no || ' ' || t_name || ' ' ||
t_mobile);END LOOP;
CLOS
E
travell
er;
END;
/
```

```
SQL> select * from transport1;

     SR_NO NAME                 MOBILE PICK_UP             TOTAL_TICKETS AMOUNT_PAI
---------- -------------------- ---------- -------------------- ------------ ----------
         1 Abhishek             8108123456 Mulund                        2         40
         2 Rubina               8108123457 Sion                          4         80
         3 Rahul                8108123458 Dadar                         1         20
         4 Yashika              8108123459 King Circle                   2         40
         5 Bhawarth             8108123460 Khar                          4         80
         6 Raju                 8108123461 Royal Circus                  1         20
         7 Shyam                8108123462 Royal Circus                  3         60

7 rows selected.
```

```
SQL> DECLARE
  2  cursor traveller is SELECT sr_no, name, mobile from transport1;
  3  t_sr_no transport1.sr_no%type;
  4  t_name transport1.name%type;
  5  t_mobile transport1.mobile%type;
  6  BEGIN
  7  OPEN traveller;
  8  LOOP
  9  FETCH traveller into t_sr_no, t_name, t_mobile;
 10  EXIT WHEN traveller%notfound;
 11  dbms_output.put_line(t_sr_no || ' ' || t_name || ' ' || t_mobile
 12  END LOOP;
 13  CLOSE traveller;
 14  END;
 15  /
1 Abhishek 8108123456
2 Rubina 8108123457
3 Rahul 8108123458
4 Yashika 8108123459
5 Bhawarth 8108123460
6 Raju 8108123461
7 Shyam 8108123462

PL/SQL procedure successfully completed.
```

- Create a plsql cursor to display the columns of sr_no, name, mobilenumber and total tickets where pickup point is ‘Royal Circus’.

DECLARE

cursor traveller2 is SELECT sr_no, name, mobile, total_tickets fromtransport1 WHERE pick_up = 'Royal Circus';

v_sr_no
transport1.sr_no%type;
v_name
transport1.name%type;
v_mobile
transport1.mobile%type
;
v_total_tickets

transport1.total_tickets%type;
BEGIN
OPEN
travell
er2;
LOOP
FETCH traveller2 into v_sr_no, v_name, v_mobile,
v_total_tickets;EXIT WHEN traveller2%notfound;
dbms_output.put_line(v_sr_no || ' ' || v_name || ' ' ||
v_mobile || ' ' ||v_total_tickets);
END LOOP;

CLOSE
traveller
2;END;
/

```
SQL> DECLARE
  2  cursor traveller2 is SELECT sr_no, name, mobile, total_tickets from transport1 WHERE pick_up = 'Royal Circ
  3  v_sr_no transport1.sr_no%type;
  4  v_name transport1.name%type;
  5  v_mobile transport1.mobile%type;
  6  v_total_tickets transport1.total_tickets%type;
  7  BEGIN
  8  OPEN traveller2;
  9  LOOP
 10  FETCH traveller2 into v_sr_no, v_name, v_mobile, v_total_tickets;
 11  EXIT WHEN traveller2%notfound;
 12  dbms_output.put_line(v_sr_no || ' ' || v_name || ' ' || v_mobile || ' ' || v_total_tickets);
 13  END LOOP;
 14  CLOSE traveller2;
 15  END;
 16  /
6 Raju 8108123461 1
7 Shyam 8108123462 3

PL/SQL procedure successfully completed.
```

- Create a plsql cursor to display the columns of sr_no, name, mobilenumber and total tickets where total tickets is 3.

```
DECLARE
cursor traveller3 is SELECT sr_no, name, mobile,
total_tickets,amount_paid from transport1 WHERE
total_tickets>=3;
v_sr_no
transport1.sr_no%type;
v_name
transport1.name%type;
v_mobile
transport1.mobile%type
;
v_total_tickets
transport1.total_tickets%type;
v_amount_paid
transport1.amount_paid%type;
BEGIN
OPEN
travell
er3;
LOOP
FETCH traveller3 into v_sr_no, v_name, v_mobile,
v_total_tickets,v_amount_paid;
EXIT WHEN traveller3%notfound;
dbms_output.put_line(v_sr_no || ' ' || v_name || ' ' ||
v_mobile || ' ' ||v_total_tickets || ' ' || v_amount_paid);
END LOOP;
CLOSE
traveller
3;END;
/
```

```
SQL> DECLARE
  2  cursor traveller3 is SELECT sr_no, name, mobile, total_tickets, amount_paid from transport1 WHERE total_tickets>
  3  v_sr_no transport1.sr_no%type;
  4  v_name transport1.name%type;
  5  v_mobile transport1.mobile%type;
  6  v_total_tickets transport1.total_tickets%type;
  7  v_amount_paid transport1.amount_paid%type;
  8  BEGIN
  9  OPEN traveller3;
 10  LOOP
 11  FETCH traveller3 into v_sr_no, v_name, v_mobile, v_total_tickets, v_amount_paid;
 12  EXIT WHEN traveller3%notfound;
 13  dbms_output.put_line(v_sr_no || ' ' || v_name || ' ' || v_mobile || ' ' || v_total_tickets || ' ' || v_amount_pa
 14  END LOOP;
 15  CLOSE traveller3;
 16  END;
 17  /
2 Rubina 8108123457 4 800
5 Bhawarth 8108123460 4 800
7 Shyam 8108123462 3 600

PL/SQL procedure successfully completed.
```

- Create a plsql cursor to display the columns of sr_no, name and mobilenumber where total tickets is 1.

DECLARE

cursor traveller4 is SELECT sr_no, name, mobile from transport1 WHEREtotal_tickets = 1;

v_sr_no
transport1.sr_no%type;
v_name
transport1.name%type;
v_mobile
transport1.mobile%type;
BEGIN
OPEN
travell
er4;
LOOP
FETCH traveller4 into v_sr_no, v_name, v_mobile;

EXIT WHEN traveller4%notfound;
dbms_output.put_line(v_sr_no || ' ' || v_name || ' ' ||
v_mobile);END LOOP;
CLOSE
traveller
4;END;
/

```
SQL> DECLARE
  2  cursor traveller4 is SELECT sr_no, name, mobile from transport1 WHERE total_tickets =
  3  v_sr_no transport1.sr_no%type;
  4  v_name transport1.name%type;
  5  v_mobile transport1.mobile%type;
  6  BEGIN
  7  OPEN traveller4;
  8  LOOP
  9  FETCH traveller4 into v_sr_no, v_name, v_mobile;
 10  EXIT WHEN traveller4%notfound;
 11  dbms_output.put_line(v_sr_no || ' ' || v_name || ' ' || v_mobile);
 12  END LOOP;
 13  CLOSE traveller4;
 14  END;
 15  /
3 Rahul 8108123458
6 Raju 8108123461

PL/SQL procedure successfully completed.
```

- Create a plsql cursor to display the columns of sr_no, name and mobilenumber where pickup point is ‘Virar’.

DECLARE

cursor traveller5 is SELECT sr_no, name, mobile from transport1 WHEREpick_up = ‘Virar';

v_sr_no
transport1.sr_no%type;
v_name
transport1.name%type;

v_mobile
transport1.mobile%type;
BEGIN
OPEN traveller5;

LOOP

FETCH traveller5 into v_sr_no, v_name,
v_mobile;EXIT WHEN
traveller5%notfound;
dbms_output.put_line(v_sr_no || ' ' || v_name || ' ' ||
v_mobile);END LOOP;
CLOSE
traveller
5;END;
/

```
SQL> DECLARE
  2  cursor traveller5 is SELECT sr_no, name, mobile from transport1 WHERE pick_up = 'Vira
  3  v_sr_no transport1.sr_no%type;
  4  v_name transport1.name%type;
  5  v_mobile transport1.mobile%type;
  6  BEGIN
  7  OPEN traveller5;
  8  LOOP
  9  FETCH traveller5 into v_sr_no, v_name, v_mobile;
 10  EXIT WHEN traveller5%notfound;
 11  dbms_output.put_line(v_sr_no || ' ' || v_name || ' ' || v_mobile);
 12  END LOOP;
 13  CLOSE traveller5;
 14  END;
 15  /

PL/SQL procedure successfully completed.
```

- Create a plsql cursor to display the columns of sr_no, name and mobilenumber where pickup point is ‘Dadar’ or ‘Sion’.

DECLARE

```
cursor traveller6 is SELECT sr_no, name, mobile from
transport1 WHEREpick_up = 'Dadar' or pick_up = 'Sion';

v_sr_no
transport1.sr_no%type;
v_name
transport1.name%type;
v_mobile
transport1.mobile%type
;

BEGIN

OPEN
travell
er6;
LOOP
FETCH traveller6 into v_sr_no, v_name,
v_mobile;EXIT WHEN
traveller6%notfound;
dbms_output.put_line(v_sr_no || ' ' || v_name || ' ' ||
v_mobile);END LOOP;
CLOSE
traveller
6;END;
/
```

```
SQL> DECLARE
  2  cursor traveller6 is SELECT sr_no, name, mobile from transport1 WHERE pick_up = 'Dadar' or pick_up = 'Sio
  3  v_sr_no transport1.sr_no%type;
  4  v_name transport1.name%type;
  5  v_mobile transport1.mobile%type;
  6  BEGIN
  7  OPEN traveller6;
  8  LOOP
  9  FETCH traveller6 into v_sr_no, v_name, v_mobile;
 10  EXIT WHEN traveller6%notfound;
 11  dbms_output.put_line(v_sr_no || ' ' || v_name || ' ' || v_mobile);
 12  END LOOP;
 13  CLOSE traveller6;
 14  END;
 15  /
2 Rubina 8108123457
3 Rahul 8108123458

PL/SQL procedure successfully completed.
```