DATABASE MANAGEMENT SYSTEMS

LAB CYCLE SHEET - 3

REG NO : 17BIT0028

NAME : SIDDHI SINGH

ALTERNATE QUERIES

7. 1. Write a simple PL/SQL block to. (High Level)

1. Print the Fibonacci series.

```
create or replace procedure fib(n number) is
f1 number;
f2 number;
c number;
i number;
begin
f1:=0;
f2:=1;
dbms_output.put_line(f1);
dbms_output.put_line(f2);
for i in 2..n
loop
    c:=f1+f2;
f1:=f2;
f2:=c;
dbms_output.put_line(c);
end loop;
end;
/
```

Output

Procedure FIB compiled

Execution

```
exec fib(5);
0
1
1
```

2

3

5

PL/SQL procedure successfully completed.

2. Print the Factorial of a given number.

```
create or replace procedure fac(n number) is
fac number:=1;
begin
while n>0 loop
fac:=n*fac;
n:=n-1;
end loop;
dbms_output.put_line(fac);
end;
/
```

Output:

Procedure FAC compiled

Execution

```
exec  fac(5);
```

120

PL/SQL procedure successfully completed.

3. Print 'NOT confirmed' based on the reservation status, of a particular passenger.

```
declare
r passenger.reservation_status%type;
a passenger.pnrno%type;
begin
a:=&a;
select reservation_status into r
```

```
from passenger where pnrno=a;
if r!='CONFIRM' then
    dbms_output.put_line('NOT CONFIRMED');
end if;
end;
/
```

Output:

NOT CONFIRMED

PL/SQL procedure successfully completed.

4. Print the total seats available for a particular train and for a particular class.

```
declare
a train.train_number%type;
b train.class%type;
seats number;
total number;
i number;
begin
a:=&a;
seats:=&seats;
select class into b from train where train_number=a;
total:=b.count;
for i in 1..total loop
dbms_output.put_line('train:' || a || ' class:' || b(i) || ' seats:'|| seats);
end loop;
end;
/
```

Output:

train:12015 class:AC CHAIR CAR seats:30

PL/SQL procedure successfully completed.

2. Write a cursor for the following.

1. Retrieve the passenger details for “x” train number and given journey date.

```
declare
d0 train.train_number%type;
d01 ticket.train_number%type;
d1 ticket.date_of_journey%type;
d10 ticket.date_of_journey%type;
d2 ticket.pnrno%type;
d3 passenger.pnrno%type;
d4 passenger.name%type;
d5 passenger.age%type;
d6 passenger.reservation_status%type;
cursor c1 is select pnrno,train_number,date_of_journey from ticket;
cursor c2 is select pnrno,name,age,reservation_status from passenger;
begin
d0:=&train_number;
d1:=&doj;
open c1;
loop
    fetch c1 into d2,d01,d10;
    exit when c1%notfound;
    if d01=d0 and d1=d10 then
        open c2;
        loop
            fetch c2 into d3,d4,d5,d6;
            exit when c2%notfound;
            if d3=d2 then
                dbms_output.put_line(d3||' '||d4||' '||d5||' '|| d6);
            end if;
        end loop;
```

```
        close c2;
    end if;
end loop;
close c1;
end;
/
```

Output

```
old:declare
d0 train.train_number%type;
d01 ticket.train_number%type;
d1 ticket.date_of_journey%type;
d10 ticket.date_of_journey%type;
d2 ticket.pnrno%type;
d3 passenger.pnrno%type;
d4 passenger.name%type;
d5 passenger.age%type;
d6 passenger.reservation_status%type;
cursor c1 is select pnrno,train_number,date_of_journey from ticket;
cursor c2 is select pnrno,name,age,reservation_status from passenger;
begin
d0:=&train_number;
d1:=&doj;
open c1;
loop
    fetch c1 into d2,d01,d10;
    exit when c1%notfound;
    if d01=d0 and d1=d10 then
        open c2;
        loop
            fetch c2 into d3,d4,d5,d6;
```

```
            exit when c2%notfound;
            if d3=d2 then
                dbms_output.put_line(d3||' '||d4||' '||d5||' '|| d6);
            end if;
        end loop;
        close c2;
    end if;
end loop;
close c1;
end;
new:declare
d0 train.train_number%type;
d01 ticket.train_number%type;
d1 ticket.date_of_journey%type;
d10 ticket.date_of_journey%type;
d2 ticket.pnrno%type;
d3 passenger.pnrno%type;
d4 passenger.name%type;
d5 passenger.age%type;
d6 passenger.reservation_status%type;
cursor c1 is select pnrno,train_number,date_of_journey from ticket;
cursor c2 is select pnrno,name,age,reservation_status from passenger;
begin
d0:=12621;
d1:=to_date('10-08-18','dd-mm-yy');
open c1;
loop
    fetch c1 into d2,d01,d10;
    exit when c1%notfound;
    if d01=d0 and d1=d10 then
```

```
        open c2;
        loop
            fetch c2 into d3,d4,d5,d6;
            exit when c2%notfound;
            if d3=d2 then
                dbms_output.put_line(d3||' '||d4||' '||d5||' '|| d6);
            end if;
        end loop;
        close c2;
    end if;
end loop;
close c1;
end;
2334567894 SANYA MALHOTRA 20 CANCELLED
PL/SQL procedure successfully completed.
```

2. Display the train name(once) and the substation names.

```
declare
t1 train%rowtype;
t2 train_route%rowtype;
cursor t3 is select *from train;
n number(10);
cursor t4 is select *from train_route where train_number=n;
begin
for t1 in t3
loop
n:=t1.train_number;
dbms_output.put_line(t1.name);
for t2 in t4
loop
```

```
dbms_output.put_line(t2.name);
end loop;
end loop;
end;
/
AJMER SHTBDI
MANDOR EXPRESS
AGRACANTT
KATPADI
BHOPAL
JHANSI
PONDI
BANGALORE
SURAT
FAKHABAD
LAKHBAD
ONDABAD
MNDABAD
G T EXPRESS
TAMIL NADU EXPRESS
METTUPALAM
KATPADI
PL/SQL procedure successfully completed.
```

3. Display the fare details of a particular train(use basic exceptions)

```
declare
train number(5) :=&train;
cursor c1 is
select base_fare from train_ticket_fare where train_number=train;
pas_rec c1%rowtype;
```

```
begin
open c1;
loop
fetch c1 into pas_rec;
exit when c1%notfound;
dbms_output.put_line(' '||train||' '||pas_rec.base_fare);
end loop;
end;
/
output
old:declare
train number(5) :=&train;
cursor c1 is
select base_fare from train_ticket_fare where train_number=train;
pas_rec c1%rowtype;
begin
open c1;
loop
fetch c1 into pas_rec;
exit when c1%notfound;
dbms_output.put_line(' '||train||' '||pas_rec.base_fare);
end loop;
end;
new:declare
train number(5) :=12621;
cursor c1 is
select base_fare from train_ticket_fare where train_number=train;
pas_rec c1%rowtype;
begin
open c1;
```

```
loop
fetch c1 into pas_rec;
exit when c1%notfound;
dbms_output.put_line(' '||train||' '||pas_rec.base_fare);
end loop;
end;
12621 730
PL/SQL procedure successfully completed.
```

8. 1. Write a PL/SQL procedure to. (High Level)

1. List the details of passengers who has reserved next to “Mr. X”.

```
declare
pnr number(10):=&pnr;
pnr2 number(10):=pnr+1;
cursor c1 is
select name from passenger where pnrno=pnr2;
pas_rec c1%rowtype;
begin
open c1;
loop
fetch c1 into pas_rec;
exit when c1%notfound;
dbms_output.put_line(pas_rec.name);
end loop;
close c1;
end;
/
```

Ouput

```
old:declare
pnr number(10):=&pnr;
```

```
pnr2 number(10):=pnr+1;
cursor c1 is
select name from passenger where pnrno=pnr2;
pas_rec c1%rowtype;
begin
open c1;
loop
fetch c1 into pas_rec;
exit when c1%notfound;
dbms_output.put_line(pas_rec.name);
end loop;
close c1;
end;
new:declare
pnr number(10):=2334567893;
pnr2 number(10):=pnr+1;
cursor c1 is
select name from passenger where pnrno=pnr2;
pas_rec c1%rowtype;
begin
open c1;
loop
fetch c1 into pas_rec;
exit when c1%notfound;
dbms_output.put_line(pas_rec.name);
end loop;
close c1;
end;
SANYA MALHOTRA
PL/SQL procedure successfully completed.
```

2. PNR No. of a passengers for a given source and a destination.

```
declare
source varchar(20);
dest varchar(20);
cursor pas_cur is
select pnrno from ticket WHERE from_station='&source' and to_station='&dest';
pas_rec pas_cur%rowtype;
begin
open pas_cur;
loop
fetch pas_cur into pas_rec;
exit when pas_cur%notfound;
dbms_output.put_line(pas_rec.pnrno);
end loop;
close pas_cur;
end;
/
```

Output:

```
old:declare
source varchar(20);
dest varchar(20);
cursor pas_cur is
select pnrno from ticket WHERE from_station='&source' and to_station='&dest';
pas_rec pas_cur%rowtype;
begin
open pas_cur;
loop
fetch pas_cur into pas_rec;
exit when pas_cur%notfound;
dbms_output.put_line(pas_rec.pnrno);
```

```
end loop;
close pas_cur;
end;
new:declare
source varchar(20);
dest varchar(20);
cursor pas_cur is
select pnrno from ticket WHERE from_station='CHENNAI' and to_station='DAURAI';
pas_rec pas_cur%rowtype;
begin
open pas_cur;
loop
fetch pas_cur into pas_rec;
exit when pas_cur%notfound;
dbms_output.put_line(pas_rec.pnrno);
end loop;
close pas_cur;
end;
2334567893
PL/SQL procedure successfully completed.
```

2. Write a PL/SQL function to.

1. Get the PNRNo and return the total ticket fare.

```
declare
n number(10);
pd ticket%rowtype;
begin
n:=&n;
select *into pd from ticket where pnrno=n;
dbms_output.put_line(pd.total_ticket_fare);
```

```
end;
/
Output
old:declare
n number(10);
pd ticket%rowtype;
begin
n:=&n;
select *into pd from ticket where pnrno=n;
dbms_output.put_line(pd.total_ticket_fare);
end;
new:declare
n number(10);
pd ticket%rowtype;
begin
n:=2334567881;
select *into pd from ticket where pnrno=n;
dbms_output.put_line(pd.total_ticket_fare);
end;
800
PL/SQL procedure successfully completed.
```

2. Get the Passenger name , train no and return the total journey time in hours and minutes.

```
declare
n number(5);
pd train%rowtype;
begin
n:=&n;
select *into pd from train where train_number=n;
dbms_output.put_line(pd.traveltime_hours);
```

```
end;
/
```

Output:

```
old:declare
n number(5);
pd train%rowtype;
begin
n:=&n;
select *into pd from train where train_number=n;
dbms_output.put_line(pd.traveltime_hours);
end;
new:declare
n number(5);
pd train%rowtype;
begin
n:=12015;
select *into pd from train where train_number=n;
dbms_output.put_line(pd.traveltime_hours);
end;
5
PL/SQL procedure successfully completed.
```

9. Write a Trigger for the following: (High Level)

1. When a passenger cancels a ticket, do the necessary process and update the cancellation history table.

```
create or replace trigger update_cancel_history
after delete on ticket
for each row
declare
pnr ticket.pnrno%type;
source ticket.from_station%type;
```

```
destination ticket.to_station%type;
journey_date ticket.date_of_journey%type;
trainno ticket.train_number%type;
pass_name passenger.name%type;
reservation_status varchar(30);
begin pnr:=:old.pnrno;
source:=:old.from_station;
destination :=:old.to_station;
journey_date :=:old.date_of_journey;
trainno:=:old.train_number;
select name into pass_name from passenger where pnrno=pnr;
reservation_status:='Booked but Cancelled';
dbms_output.put_line('Deleting person:'||pass_name);
insert into cancel_history
values(pnr,source,destination,journey_date,trainno,pass_name,reservation_status);
dbms_output.put_line('Cancel history table successfully updated');
end;
/

Trigger UPDATE_CANCEL_HISTORY compiled
```

2. When train number is changed, update it in referencing tables.

```
create or replace trigger update_trainno_refertab
after update on train for each row
declare
old_trainno train.train_number%type;
new_trainno train.train_number%type;
begin
old_trainno:=:old.train_number;
new_trainno:=:new.train_number;
update ticket set train_number=new_trainno where train_number=old_trainno;
```

```
update train_ticket_fare set train_number=new_trainno where train_number=old_trainno;

update train_route set train_number=new_trainno where train_number=old_trainno;

dbms_output.put_line('Successfully train_number updated from '||:old.train_number||' to
'||:new.train_number ||' for all the tables.');

end;

/

Trigger UPDATE_TRAINNO_REFERTAB compiled
```

3. When a passenger record is inserted reservation status should be automatically updated.

```
create or replace trigger update_reservation_status

after insert on ticket

for each row

declare

pnr ticket.pnrno%type;

reservation_sta passenger.reservation_status%type;

pass_name passenger.name%type;

begin reservation_sta:='Confirmed';

pnr:=:new.pnrno;

select name into pass_name from passenger where pnrno=pnr;

update passenger set reservation_status=reservation_sta where pnrno=pnr;

dbms_output.put_line('After insert of '||pnr||' & '||pass_name||' the reservation status has
been updated');

end;

/

Trigger UPDATE_RESERVATION_STATUS compiled
```

10. 1. Use TCL commands for your transactions. (commit,rollback,savepoint) (Middle Level)

```
commit;

Commit complete.

rollback;

Rollback complete.
```

```
SAVEPOINT s;

Savepoint created.
```

2. Create a role named 'clerk', and give permisson for him to select only the trains starting from 'Katpadi' along with fare details.

```
create role clerk;

grant select total_ticket_fare,from_station from ticket where from_station='katpadi' to clerk;
```

3. Create a nested table containing trainno,name,source,destination and passengers who have booked for it (PNR no,sno, name,age). Find the passengers whose name start with 'S' and train starts from 'Katpadi'

```
create type passenger_t as object (pnr_no varchar(10),

sno number(5),

name varchar(100),

age number(2));

/

Type PASSENGER_T compiled

create type passengers_t is table of passenger_t;

Type PASSENGERS_T compiled

CREATE TABLE train_1(

Trainno number(5),

Name varchar(50),

Source varchar(50),

Destination varchar(50),

Passengers passengers_t

);

SQL> SELECT p.* FROM train t , TABLE (t.passengers) p WHERE t.source = ‘Katpadi’ AND
p.name LIKE ‘S%’
```

7.

***1**. Write a simple PL/SQL block to :*

***1.** Print the fibonacci series.*

```
SQL> SET SERVEROUTPUT ON;
SQL> DECLARE
  2   a NUMBER := 0;
  3   b NUMBER := 1;
  4   c NUMBER := 0;
  5   i NUMBER := 0;
  6   BEGIN
  7   dbms_output.put_line(a);
  8   dbms_output.put_line(b);
  9   WHILE i<20 LOOP
 10   c:=a+b;
 11   dbms_output.put_line(c);
 12   b:=a;
 13   a:=c;
 14   i:=i+1;
 15   END LOOP;
 16   END;
 17   /
0
1
1
1
2
3
5
8
13
21
34
55
89
144
233
377
610
987
1597
2584
4181
6765

PL/SQL procedure successfully completed.
```

***2**. Print the factorial of a given number.*

```
SQL> SET SERVEROUTPUT ON;
SQL> DECLARE
  2  c NUMBER(5) := 1;
  3  i NUMBER(5);
  4  BEGIN
  5  i := &i;
  6  WHILE i > 0 LOOP
  7  c := c*i;
  8  i := i - 1;
  9  END LOOP;
 10  dbms_output.put_line('Factorial is :'||c);
 11  END;
 12  /
Enter value for i: 5
old   5: i := &i;
new   5: i := 5;
Factorial is :120

PL/SQL procedure successfully completed.
```

3. *Print 'NOT confirmed' based on the reservation status, of a particular passenger.*

```
SQL> SET SERVEROUTPUT ON;
SQL> DECLARE
  2  CURSOR pass_cur is
  3  SELECT Name,Reservation_Status FROM PASSENGER_DETAILS WHERE Reservation_Sta
tus!='CNF';
  4  pass_rec pass_cur%rowtype;
  5  BEGIN
  6  open pass_cur;
  7  LOOP
  8  fetch pass_cur into pass_rec;
  9  exit when pass_cur%notfound;
 10  dbms_output.put_line('Passenger Name: '||pass_rec.Name||' '||'Reservation S
tatus : '|| pass_rec.Reservation_Status);
 11  END LOOP;
 12  END;
 13  /
Passenger Name: Rehan Reservation Status : Waiting
Passenger Name: Nidhi Reservation Status : RAC
Passenger Name: Armaan Reservation Status : Waiting

PL/SQL procedure successfully completed.
```

4*. Print the total seats available for a particular train and for a particular class.*

```
SQL> ALTER TABLE TRAIN ADD Seat NUMBER(5);

Table altered.

SQL> UPDATE TRAIN SET Seat=100 WHERE Name='Intercity';

1 row updated.

SQL> UPDATE TRAIN SET Seat=200 WHERE Name='BGKT MQ';

1 row updated.

SQL> UPDATE TRAIN SET Seat=50 WHERE Name='Pune Duronto';

1 row updated.

SQL> UPDATE TRAIN SET Seat=20 WHERE Name='Double Decker';

1 row updated.

SQL> UPDATE TRAIN SET Seat=150 WHERE Name='Brindavan';

1 row updated.

SQL> UPDATE TRAIN SET Seat=30 WHERE Name='Lalbagh';

1 row updated.
```

```
SQL> SET SERVEROUTPUT ON;
SQL> DECLARE
  2  CURSOR Seat is
  3  SELECT Name,Class,Seat FROM TRAIN WHERE Class='2A';
  4  Seat_rec Seat%rowtype;
  5  BEGIN
  6  OPEN Seat;
  7  LOOP
  8  FETCH Seat INTO Seat_rec;
  9  EXIT WHEN Seat%notfound;
 10  dbms_output.put_line('Train Name:'||seat_rec.Name||''||'Class:'||Seat_r
lass||' Total Seats Available:'||Seat_rec.Seat);
 11  END LOOP;
 12  END;
 13  /
Train Name:BGKT MQClass:2A Total Seats Available:200

PL/SQL procedure successfully completed.
```

2. *Write a cursor for the following.*

1. *Retrieve the passenger details for "x" train number and given journey date.*

```
SQL> SET SERVEROUTPUT ON;
SQL> DECLARE
  2  Train NUMBER(5);
  3  Journey_Date DATE;
  4  CURSOR pas_cur is
  5  SELECT Name,Age FROM PASSENGER_DETAILS WHERE PNRNo in(SELECT PNRNo FROM TIC
KET WHERE Train_Number=&Train and Date_Of_Journey='&Journey_Date');
  6  pas_rec pas_cur%rowtype;
  7  BEGIN
  8  OPEN pas_cur;
  9  LOOP
 10  FETCH pas_cur into pas_rec;
 11  EXIT WHEN pas_cur%notfound;
 12  dbms_output.put_line(pas_rec.Name||' '||pas_rec.Age);
 13  END LOOP;
 14  END;
 15  /
Enter value for train: 12640
Enter value for journey_date: 18-AUG-2017
old   5: SELECT Name,Age FROM PASSENGER_DETAILS WHERE PNRNo in(SELECT PNRNo FROM
 TICKET WHERE Train_Number=&Train and Date_Of_Journey='&Journey_Date');
new   5: SELECT Name,Age FROM PASSENGER_DETAILS WHERE PNRNo in(SELECT PNRNo FROM
 TICKET WHERE Train_Number=12640 and Date_Of_Journey='18-AUG-2017');
Shivan 20

PL/SQL procedure successfully completed.
```

2. *Display the train name(once) and the substation names.*

```
SQL> SET SERVEROUTPUT ON;
SQL> DECLARE
  2  T TRAIN%rowtype;
  3  TR TRAIN_ROUTE%rowtype;
  4  CURSOR Train1 is SELECT * FROM TRAIN;
  5  N NUMBER(10);
  6  CURSOR T_R IS SELECT * FROM TRAIN_ROUTE WHERE Train_Number=n;
  7  BEGIN
  8  FOR T IN TRAIN1
  9  LOOP
 10  N := T.Train_Number;
 11  dbms_output.put_line(T.Name);
 12  FOR TR in T_R
 13  LOOP
 14  dbms_output.put_line(TR.Name);
 15  END LOOP;
 16  END LOOP;
 17  END;
 18  /
Intercity
Jodhpur
BGKT MQ
Jodhpur
Pune Duronto
PUNE
Double Decker
Brindavan
Bengaluru
Lalbagh
Katpadi

PL/SQL procedure successfully completed.
```

3. *Display the fare details of a particular train(use basic exceptions)*

```
SQL> SET SERVEROUTPUT ON;
SQL> DECLARE
  2  Train NUMBER(5) := &Train;
  3  CURSOR pas_cur is
  4  SELECT Base_Fare FROM TRAIN_TICKET_FARE WHERE Train_Number=Train;
  5  pas_rec pas_cur%rowtype;
  6  BEGIN
  7  OPEN pas_cur;
  8  LOOP
  9  FETCH pas_cur into pas_rec;
 10  EXIT WHEN pas_cur%notfound;
 11  dbms_output.put_line(' '||Train ||' '|| pas_rec.base_fare);
 12  END LOOP;
 13  END;
 14  /
Enter value for train: 12640
old   2: Train NUMBER(5) := &Train;
new   2: Train NUMBER(5) := 12640;
12640 70

PL/SQL procedure successfully completed.
```

8.

1. *Write a PL/SQL procedure to.*

1. List the details of passengers who has reserved next to "Mr. X".

```
SQL> SET SERVEROUTPUT ON;
SQL> DECLARE
  2  PNR NUMBER(12) := &PNR;
  3  PNR2 NUMBER(12) := PNR+1;
  4  CURSOR pas_cur is
  5  SELECT Name FROM PASSENGER_DETAILS WHERE PNRNo=PNR2;
  6  pas_rec pas_cur%rowtype;
  7  BEGIN
  8  OPEN pas_cur;
  9  LOOP
 10  FETCH pas_cur into pas_rec;
 11  EXIT WHEN pas_cur%notfound;
 12  dbms_output.put_line(pas_rec.Name);
 13  END LOOP;
 14  CLOSE pas_cur;
 15  END;
 16  /
Enter value for pnr: 4647577549
old   2: PNR NUMBER(12) := &PNR;
new   2: PNR NUMBER(12) := 4647577549;
Ansh

PL/SQL procedure successfully completed.
```

2. PNR No. of a passengers for a given source and a destination.

```
SQL> SET SERVEROUTPUT ON;
SQL> DECLARE
  2  Source VARCHAR(20);
  3  Dest VARCHAR(20);
  4  CURSOR pas_cur is
  5  SELECT PNRNo FROM TICKET WHERE FROMSSTATION = '&Source' AND TO_STATION = '&
Dest';
  6  pas_rec pas_cur%rowtype;
  7  BEGIN
  8  OPEN pas_cur;
  9  LOOP
 10  FETCH pas_cur into pas_rec;
 11  EXIT WHEN pas_cur%notfound;
 12  dbms_output.put_line(pas_rec.PNRNo);
 13  END LOOP;
 14  CLOSE pas_cur;
 15  END;
 16  /
Enter value for source: Katpadi
Enter value for dest: Bengaluru
old   5: SELECT PNRNo FROM TICKET WHERE FROMSSTATION = '&Source' AND TO_STATION
= '&Dest';
new   5: SELECT PNRNo FROM TICKET WHERE FROMSSTATION = 'Katpadi' AND TO_STATION
= 'Bengaluru';
2346457520

PL/SQL procedure successfully completed.
```

2. *Write a PL/SQL function to.*

1. Get the PNRNo and return the total ticket fare.

```
SQL> SET SERVEROUTPUT ON;
SQL> DECLARE
  2  N NUMBER(10);
  3  PD TICKET%rowtype;
  4  BEGIN
  5  N := &N;
  6  SELECT * INTO PD FROM TICKET WHERE PNRNo=N;
  7  dbms_output.put_line(pd.TOTAL_TICKET_FARE);
  8  END;
  9  /
Enter value for n: 4647577549
old   5: N := &N;
new   5: N := 4647577549;
130

PL/SQL procedure successfully completed.
```

2. Get the Passenger Name, Train_Number and return the total journey time in hours and minutes.

```
SQL> SET SERVEROUTPUT ON;
SQL> DECLARE
  2  N NUMBER(5);
  3  PD TRAIN%ROWTYPE;
  4  BEGIN
  5  N := &N;
  6  SELECT * INTO PD FROM TRAIN WHERE Train_Number=N;
  7  dbms_output.put_line(to_char(PD.Travel_Time));
  8  END;
  9  /
Enter value for n: 12640
old   5: N := &N;
new   5: N := 12640;
04:16:00

PL/SQL procedure successfully completed.
```

9.

Write a Trigger for the following:

1. *When a passenger cancels a ticket, do the necessary process and update the cancellation history table.*
2. *When train number is changed, update it in referencing tables.*
3. *When a passenger record is inserted reservation status should be automatically updated.*

```
1. create or replace trigger update_cancel_history after delete
on ticket for each row declare pnr
 ticket.pnrno%type;
 source ticket.from_station%type;
 destination ticket.to_station%type;
 journey_date ticket.date_of_journey%type;
 trainno ticket.train_number%type;
 pass_name passenger.name%type;
 reservation_sta varchar(30);
begin pnr:=:old.pnrno;
 source:=:old.from_station;
 destination :=:old.to_station;
 journey_date
```

```
  :=:old.date_of_journey;
  trainno:=:old.train_number;
  select name into pass_name from passenger where
  pnr_no=pnr; reservation_sta:='Booked but Cancelled';
dbms_output.put_line('Deleting person:'||pass_name); insert into
cancel_history
values(pnr,source,destination,journey_date,trainno,pass_n
ame,reservation_sta);
  dbms_output.put_line('Cancel history table successfully updated');
end;
/
```

EXPLANATION :

SQL> desc cancel_history;

Name	Null?	Type
PNRNO		NUMBER(10)
FROM_STATION		VARCHAR2(30)
TO_STATION		VARCHAR2(30)
DATE_OF_JOURNEY		DATE
TRAIN_NUMBER		NUMBER(6)
NAME		VARCHAR2(30)
RESERVATION_STATUS		VARCHAR2(30)

SQL> select * from cancel_history;

no rows selected

Ticket details:

```
SQL> desc ticket;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------

 PNRNO                                     NOT NULL NUMBER(10)
 TRANSACTIONID                                      VARCHAR2(20)
 FROM_STATION                                       VARCHAR2(30)
 TO_STATION                                         VARCHAR2(30)
 DATE_OF_JOURNEY                                    DATE
 CLASS_DATE_OF_BOOKING                              DATE
 TOTAL_TICKET_FARE                                  NUMBER(5)
 TRAIN_NUMBER                                       NUMBER(6)
 SEAT_NO                                            NUMBER(3)

SQL> select * from ticket;

     PNRNO TRANSACTIONID         FROM_STATION
---------- -------------------- ------------------------------
TO_STATION                     DATE_OF_J CLASS_DAT TOTAL_TICKET_FARE
------------------------------ --------- --------- ------------------
TRAIN_NUMBER    SEAT_NO
------------ ----------
       111 100                 Chennai
Mumbai                         30-AUG-16 15-AUG-16              1000
       10000          1

       222 200                 Mumbai
Chennai                        30-SEP-16 10-AUG-16              1500
       10001          2

     PNRNO TRANSACTIONID         FROM_STATION
---------- -------------------- ------------------------------
TO_STATION                     DATE_OF_J CLASS_DAT TOTAL_TICKET_FARE
------------------------------ --------- --------- ------------------
TRAIN_NUMBER    SEAT_NO
------------ ----------

       333 300                 Chennai
Delhi                          15-OCT-16 16-AUG-16              1200
       10002          3

       444 400                 Mumbai
Delhi                          25-DEC-16 01-SEP-16              1300

     PNRNO TRANSACTIONID         FROM_STATION
---------- -------------------- ------------------------------
TO_STATION                     DATE_OF_J CLASS_DAT TOTAL_TICKET_FARE
------------------------------ --------- --------- ------------------
TRAIN_NUMBER    SEAT_NO
------------ ----------
       10003          4

       555 500                 Delhi
Chennai                        30-OCT-16 30-SEP-16              4000
       10004          5
```

```
SQL> set serveroutput on;
SQL> @ E:\PLSQL\Triggers\9.1.txt

Trigger created.

SQL>
```

SQL> delete from ticket where pnrno=222; Deleting person: Raj Cancel history table successfully updated 1 row deleted.

```
SQL> delete from ticket where pnrno=222;
Deleting person:Raj
Cancel history table successfully updated

1 row deleted.

SQL>
```

SQL> select * from cancel_history;

PNRNO FROM_STATION TO_STATION ----------
---------- ------------------------------ ------------------------------
DATE_OF_J TRAIN_NUMBER NAME --------- ------------
------------------------------ RESERVATION_STATUS

----
222 Mumbai
Chennai
10001
30-SEP-16 Raj
Booked but
Cancelled

<u>Screenshot:</u>

```
SQL> delete from ticket where pnrno=222;
Deleting person:Raj
Cancel history table successfully updated

1 row deleted.

SQL> select * from cancel_history;

     PNRNO FROM_STATION                   TO_STATION
---------- ------------------------------ ------------------------------
DATE_OF_J TRAIN_NUMBER NAME
--------- ------------ ------------------------------
RESERVATION_STATUS
--------------------------
       222 Mumbai                         Chennai
30-SEP-16        10001 Raj
Booked but Cancelled
```

2. *When train number is changed, update it in referencing tables*

<u>OVERALL CODE:</u>

create or replace trigger ***update_trainno_refertab after update on train*** *for each row*

```
 declare
   old_trainno train.train_number%type;
  new_trainno
  train.train_number%type; begin
old_trainno:=:old.train_number;
new_trainno:=:new.train_number;
  update ticket set train_number=new_trainno where
train_number=old_trainno;
  update train_ticket_fare set train_no=new_trainno
where train_no=old_trainno;
  update train_route set train_no=new_trainno
where train_no=old_trainno;
dbms_output.put_line('Successfully train_number updated
from '||:old.train_number||' to '||:new.train_number ||' for all
the tables.');
end;
/
```

EXPLANATION:

Ticket table,Train_ticket_fare table and train_route tables are refer the values of train_number from train table.

SQL> select train_number,name from train;

```
TRAIN_NUMBER NAME
------------ --------------------
10000 Chennaiexp
10001 Mumbaiexp
10002 Chennaico
10003 Mumbaico
10004 Delhico
10006 Vellore

6 rows selected
```

SQL> select train_number,pnrno from ticket;

TRAIN_NUMBER	*PNRNO*
10000	*111*
10002	*333*
10003	*444*
10004	*555*

SQL> select train_no from train_ticket_Fare;

TRAIN_NO
10000
10001
10002
10003
10004

SQL> select train_no,name from train_route;

TRAIN_NO NAME
10000 Maduras_1
10001 Mumbai_1
10002 Maduras_2
10003 Mumbai_2
10004 Delhi_1
10001 chennaiexp
10006 chennaiexp

7 *rows selected.*

```
SQL> select train_number,name from train;

TRAIN_NUMBER NAME
------------ --------------------
       10000 Chennaiexp
       10001 Mumbaiexp
       10002 Chennaico
       10003 Mumbaico
       10004 Delhico
       10006 Vellore

6 rows selected.

SQL> select train_number,pnrno from ticket;

TRAIN_NUMBER      PNRNO
------------ ----------
       10000        111
       10002        333
       10003        444
       10004        555

SQL> select train_no from train_ticket_Fare;

  TRAIN_NO
----------
     10000
     10001
     10002
     10003
     10004

SQL> select train_no,name from train_route;

  TRAIN_NO NAME
---------- --------------------
     10000 Maduras_1
     10001 Mumbai_1
     10002 Maduras_2
     10003 Mumbai_2
     10004 Delhi_1
     10001 chennaiexp
     10006 chennaiexp

7 rows selected.

SQL>
```

SQL> @ E:\PLSQL\Triggers\9.2.txt Trigger created.

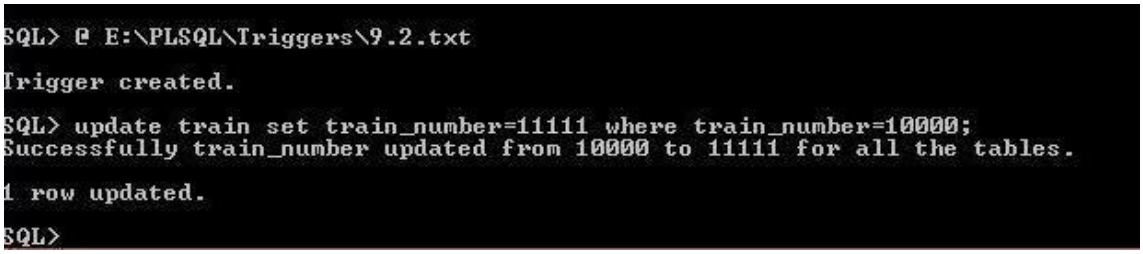
```
SQL> @ E:\PLSQL\Triggers\9.2.txt

Trigger created.
```

SQL> update train set train_number=11111 where train_number=10000;

Successfully train_number updated from 10000 to 11111 for all the tables.

1 row updated.



SQL> select train_number,name from train;

TRAIN_NUMBER NAME

---------- ----------

11111 Chennaiexp

10001 Mumbaiexp

10002 Chennaico

10003 Mumbaico

10004 Delhico

10006 Vellore

6 rows selected.

SQL> select train_number,pnrno from ticket;

TRAIN_NUMBER	*PNRNO*
11111	*111*
10002	*333*
10003	*444*
10004	*555*

SQL> select train_no from train_ticket_Fare;

TRAIN_NO

10001
10002
10003
10004
11111

SQL> select train_no,name from train_route;

TRAIN_NO NAME
--------- ----------
11111 Maduras_1
10001 Mumbai_1
10002 Maduras_2
10003 Mumbai_2
10004 Delhi_1
10001 chennaiexp
10006 chennaiexp

7 *rows selected.*

```
SQL> select train_number,name from train;

TRAIN_NUMBER NAME
------------ --------------------
       11111 Chennaiexp
       10001 Mumbaiexp
       10002 Chennaico
       10003 Mumbaico
       10004 Delhico
       10006 Vellore

6 rows selected.

SQL> select train_number,pnrno from ticket;

TRAIN_NUMBER      PNRNO
------------ ----------
       11111        111
       10002        333
       10003        444
       10004        555

SQL> select train_no from train_ticket_Fare;

  TRAIN_NO
----------
     10001
     10002
     10003
     10004
     11111

SQL> select train_no,name from train_route;

  TRAIN_NO NAME
---------- --------------------
     11111 Maduras_1
     10001 Mumbai_1
     10002 Maduras_2
     10003 Mumbai_2
     10004 Delhi_1
     10001 chennaiexp
     10006 chennaiexp

7 rows selected.

SQL>
```

3. *When a passenger record is inserted reservation status should be automatically updated.*

OVERALL CODE:

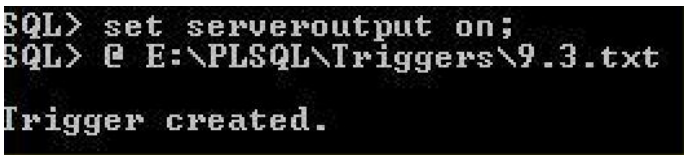
```
create or replace trigger update_reservation_status after insert on
ticket for each row declare pnr
ticket.pnrno%type;
reservation_sta passenger.reservation_status%type;
pass_name passenger.name%type;
begin reservation_sta:='Confirmed';
pnr:=:new.pnrno;
  select name into pass_name from passenger where pnr_no=pnr;
  update passenger set reservation_status=reservation_sta where
pnr_no=pnr;
  dbms_output.put_line('After insert of '||pnr||' & '||pass_name||' the
reservation status has been updated');
end;
```

EXPLANATION:

When passenger records insert on ticket table it will automaticallly update the reservation_station as ticket 'Confirmed' in passenger table.

In passenger table, pnr_no 222 has reservation status as 'Not Confirmed'.Because that record was not inserted in ticket table.So when I insert that pnrno in ticket table it will automatically update the reservation_status as ***'Confirmed'.***

```
SQL> set serveroutput on;
SQL> @ E:\PLSQL\Triggers\9.3.txt Trigger created.
```



SQL> insert into ticket values (222,200,'Mumbai','Chennai','10-Nov-16','30-Oct-16 ',1500,10006,6);
After insert of 222 & Raj the reservation status has been updated

1 row created.

```
SQL> insert into ticket values(222,200,'Mumbai','Chennai','10-Nov-16','30-Oct-16
',1500,10006,6);
After insert of 222 & Raj the reservation status has been updated

1 row created.
```

SQL> select pnrno,train_number,date_of_journey from ticket;

PNRNO TRAIN_NUMBER DATE_OF_J
--------- ----------- ---------

111	*11111 30-AUG-16*
333	*10002 15-OCT-16*
444	*10003 25-DEC-16*
555	*10004 30-OCT-16*
222	*10006 10-NOV-16*

*SQL> select * from passenger; PNR_NO SERIAL_NO NAME AGE RESERVATION_STA*

111	*10*	*Gulsath*	*18*	*Confirmed*
222	*20*	*Raj*	*25*	*Confirmed*
333	*30*	*Rassul*	*20*	*Confirmed*
444	*40*	*Mohammed*	*30*	*Confirmed*
555	*50*	*Asgar*	*16*	*Confirmed*

10

1. Use TCL commands for your transactions. (commit,rollback,savepoint)

2. *Create a role named 'clerk', and give permisson for him to select only the trains starting from 'Katpadi' along with fare details.*

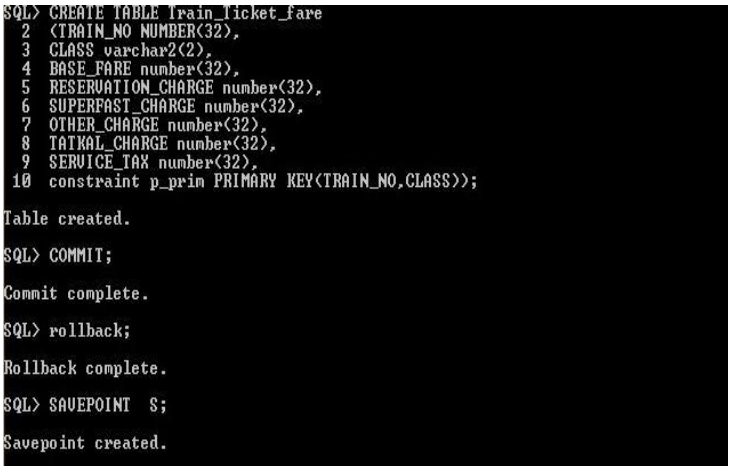
3. *Create a nested table containing trainno,name,source,destination and passengers who have booked for it (PNR no,sno, name,age). Find the passengers whose name start with 'S' and train starts from 'Katpadi'*

1. ***Use TCL commands for your transactions. (commit,rollback,savepoint)***

SQL> COMMIT;
Commit complete. SQL> rollback; Rollback complete.

SQL>SAVEPOINT S;

Savepoint created.



2. ***Create a role named 'clerk', and give permission for him to select only the trains starting from 'Katpadi' along with fare details.***

SQL> create role clerk;
SQL> GRANT select total_ticket_fare, from_station from ticket where from_station ='katpadi' TO clerk;

3. ***Create a nested table containing train.no,name,source,destination and passengers who have booked for it (PNR no,sno, name,age). Find the passengers whose name start with 'S' and train starts from 'Katpadi'.***

CREATE TYPE passenger_t AS OBJECT (PNR_NO varchar(10),

sno Number(5),
name varchar(100),
age Number(2)
);
/
CREATE TYPE passengers_t IS TABLE OF passenger_t; CREATE

TABLE train (

Trainno number(5), Name varchar(50), Source varchar(50),

Destination varchar(50), Passengers passengers_t

);

```
SQL> CREATE TYPE passenger_t AS OBJECT (
  2  PNR_NO     varchar(10),
  3  sno              Number(5),
  4  name         varchar(100),
  5  age              Number(2)
  6  );
  7
  8
  9
 10  /

Type created.

SQL> CREATE TYPE passengers_t IS TABLE OF passenger_t;
  2  CREATE TABLE train (
  3     Trainno number(5),
  4     Name    varchar(50),
  5     Source  varchar(50),
  6     Destination     varchar(50),
  7     Passengers      passengers_t
  8  );
  9  /
```

SQL> SELECT p. FROM train t , TABLE (t.passengers) p WHERE t.source = 'Katpadi' AND p.name LIKE 'S%'*

```
SQL> SELECT p.* FROM train t , TABLE (t.passengers) p WHERE t.source = 'Katpadi'
 AND p.name LIKE 'S%'
  2  -
```