VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Information Technology and Engineering

Lab Cycle Sheet, JUNE 2020

B.Tech, Winter-2019-2020

NAME	PRIYAL BHARDWAJ
REG. NO.	18BIT0272
COURSE CODE	ITE1003
COURSE NAME	DATABASE MANAGEMENT SYSTEMS
SLOT	L-33+L-34
FACULTY	Prof. BIMAL KUMAR RAY

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

1. Print the Fibonacci series.

```
set serveroutput on
declare
a number:=-1;
b number:=1;
c number:=0;
n number;
i number;
begin
n:=&n;
for i in 1..n loop
c:=a+b;
a:=b;
b:=c;
dbms_output.put_line(c);
end loop;
end;
/
```

```
SQL> set serveroutput on
SQL> declare
  2  a number:=-1;
  3  b number:=1;
  4  c number:=0;
  5  n number;
  6  i number;
  7  begin
  8  n:=&n;
  9  for i in 1..n loop
 10  c:=a+b;
 11  a:=b;
 12  b:=c;
 13  dbms_output.put_line(c);
 14  end loop;
 15  end;
 16  /
Enter value for n: 10
old   8: n:=&n;
new   8: n:=10;
0
1
1
2
3
5
8
13
21
34

PL/SQL procedure successfully completed.
```

2. Print the Factorial of a given number.

```
set serveroutput on
declare
f number:=1;
i number;
n number;
begin
dbms_output.put_line('Enter the number to find Factorial:');
n:=&n;
for i in 1..n loop
f:=f*i;
end loop;
dbms_output.put_line('the factorial of '||n' number is:'||f);
end;
/
```

```
SQL> set serveroutput on
SQL> declare f
  2  number:=1;
  3  i number;
  4  n number;
  5  begin
  6  dbms_output.put_line('Enter the number to find Factorial: ');
  7  n:=&n;
  8  for i in 1..n loop
  9  f:=f*i;
 10  end loop;
 11  dbms_output.put_line('Factorial of '||n' number is: '||f);
 12  end;
 13  /
Enter value for n: 5
old   7: n:=&n;
new   7: n:=5;
Enter the number to find Factorial:
Factorial of  number is: 120

PL/SQL procedure successfully completed.
```

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

```
set serveroutput on
declare
Passenger_pnr_no passenger.pnr_no%type:=&pnr_no;
Passenger_status passenger.reservation_status%type;
begin
select reservation_status into Passenger_status from passenger where
pnr_no = Passenger_pnr_no;
if(Passenger_status!='CNF') then dbms_output.put_line('Not
Confirmed');
else dbms_output.put_line('Confirmed');
end if;
end;
/
```

```
SQL> set serveroutput on
SQL> declare
  2  passenger_pnr_no passenger.pnr_no%type:=&pnr_no;
  3  passenger_status passenger.reservation_status%type;
  4  begin
  5  select reservation_status into passenger_status from passenger where pnr_no = passenger_pnr_no;
  6  if(passenger_status!='CNF') then
  7  dbms_output.put_line('Not Confirmed');
  8  else
  9  dbms_output.put_line('Confirmed');
 10  end if;
 11  end;
 12  /
Enter value for pnr_no: 3615847290
old   2: passenger_pnr_no passenger.pnr_no%type:=&pnr_no;
new   2: passenger_pnr_no passenger.pnr_no%type:=3615847290;
Not Confirmed

PL/SQL procedure successfully completed.
```

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

```
alter table train_ticket_fare add total_seats number(10);
update train_ticket_fare set total_seats = 150 where train_number =
12138;
set serveroutput on
declare
tn train_ticket_fare.train_number%type;
tclass train_ticket_fare.t_class%type;
total_seat train_ticket_fare.total_seats%type;
begin
tn:=&tn;
tclass:='&tclass';
select total_seats into total_seat from train_ticket_fare where
train_number = tn and t_class=tclass;
dbms_output.put_line('Total Seats:'||total_seat);
end;
/
```

```
SQL> alter table train_ticket_fare add total_seats number(10);

Table altered.

SQL> update train_ticket_fare set total_seats = 150 where train_number = 12138;

1 row updated.

SQL> set serveroutput on
SQL> declare
  2  tn train_ticket_fare.train_number%type;
  3  tclass train_ticket_fare.ticket_class%type;
  4  total_seat train_ticket_fare.total_seats%type;
  5  begin
  6  tn:=&tn;
  7  tclass:='&tclass';
  8  select total_seats into total_seat from train_ticket_fare where train_number=tn and ticket_class=tclass;
  9  dbms_output.put_line('Total Seats: '||total_seat);
 10  end;
 11  /
Enter value for tn: 12138
old   6: tn:=&tn;
new   6: tn:=12138;
Enter value for tclass: 2A
old   7: tclass:='&tclass';
new   7: tclass:='2A';
Total Seats: 150

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.

```
set serveroutput on
declare
cursor pdetails is
select * from passenger
inner join ticket
on passenger.pnr_no=ticket.pnr_no;
pnr passenger.pnr_no%type;
tn ticket.train_number%type;
doj ticket.date_of_journey%type;
name passenger.name%type;
pdetailsr pdetails%rowtype;
begin
tn:=&tn;
doj:='&doj';
open pdetails;
loop
fetch pdetails INTO PDETAILSR;
exit when pdetails%notfound;
if(pdetailsr.train_number=tn and pdetailsr.date_of_journey=doj)
then dbms_output.put_line('Name: '||PDETAILSR.name||'
Age: '||PDETAILSR.age ||' Reservation Status:
'||PDETAILSR.reservation_status ||' Station:
'||PDETAILSR.from_station||' to '||PDETAILSR.to_station);
else
dbms_output.put_line('No data Found');
end if;
end loop;
close pdetails;
end;
/
```

```
SQL> set serveroutput on
SQL> declare
  2  cursor pdetails is
  3  select * from passenger
  4  inner join ticket
  5  on passenger.pnr_no=ticket.pnr_no;
  6  pnr passenger.pnr_no%type;
  7  tn ticket.train_number%type;
  8  doj ticket.date_of_journey%type;
  9  name passenger.name%type;
 10  pdetailsr pdetails%rowtype;
 11  begin
 12  tn:=&tn;
 13  doj:='&doj';
 14  open pdetails;
 15  loop
 16  fetch pdetails INTO PDETAILSR;
 17  exit when pdetails%notfound;
 18  if(pdetailsr.train_number=tn and pdetailsr.date_of_journey=doj)
 19  then dbms_output.put_line('Name: '||PDETAILSR.name||'
 20  Age: '||PDETAILSR.age ||' Reservation Status:
```

```
 21  '||PDETAILSR.reservation_status ||' Station:
 22  '||PDETAILSR.from_station||' to '||PDETAILSR.to_station);
 23  else
 24  dbms_output.put_line('No data Found');
 25  end if;
 26  end loop;
 27  close pdetails;
 28  end;
 29  /
Enter value for tn: 12138
old  12: tn:=&tn;
new  12: tn:=12138;
Enter value for doj: 30-JAN-20
old  13: doj:='&doj';
new  13: doj:='30-JAN-20';
No data Found
No data Found
No data Found
No data Found
No data Found
No data Found
No data Found
Name: Kushagra
Age: 83 Reservation Status:
Wait Station:
Mumbai to New Delhi

PL/SQL procedure successfully completed.
```

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

```
set serveroutput on
declare
cursor train is
select * from train inner join ticket on
train.train_number=ticket.train_number;
train_name train%rowtype;
begin
open train;
loop
fetch train into train_name;
exit when train%notfound;
dbms_output.put_line('Train name: '||train_name.name ||' Substations:
'||train_name.from_station);
dbms_output.put_line(train_name.to_station);
dbms_output.put_line(train_name.source||','||train_name.destination);
end loop;
close train;
end;
/
```

```
SQL> set serveroutput on
SQL> declare
  2  cursor train is
  3  select * from train inner join ticket on train.train_number=ticket.train_number;
  4  train_name train%rowtype;
  5  begin
  6  open train;
  7  loop
  8  fetch train into train_name;
  9  exit when train%notfound;
 10  dbms_output.put_line('Train name: '||train_name.name ||' Substations: '||train_name.from_station);
 11  dbms_output.put_line(train_name.to_station);
 12  dbms_output.put_line(train_name.source||','||train_name.destination);
 13  end loop;
 14  close train;
 15  end;
 16  /
Train name: Chennai Express Substations: Chennai Egmore
Mumbai Central
Chennai Egmore,Dadar
Train name: Mumbai Express Substations: Bangalore Cantt
Mumbai Cantt
Chennai Central,Mumbai Cst
Train name: Mumbai Express Substations: Katpadi
Manglore
Chennai Central,Mumbai Cst
Train name: Pune Express Substations: Mumbai Central
Bengalore Cantt
Delhi,Pune
Train name: Mumbai Mail Substations: Mumbai Central
Katpadi
Chennai Central,Mumbai Cst
Train name: Punjab Mail Substations: Mumbai
New Delhi
Mumbai,New Delhi
Train name: Chennai Express Substations: Chennai Egmore
Mumbai Central
Chennai Egmore,Dadar
Train name: Chennai Express Substations: Chennai Egmore
Mumbai Central
Chennai Egmore,Dadar

PL/SQL procedure successfully completed.
```

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

```
set serveroutput on
declare
cursor train_fare is
select * from train_ticket_fare;
fare train_fare%rowtype;
train_number number;
begin
open train_fare;
loop
fetch train_fare into fare;
exit when train_fare%notfound;
train_number:=&train_number;
if(fare.train_number=train_number) then
dbms_output.put_line(' Base Fare:'||fare.base_fare);
dbms_output.put_line(' Reservation
Charge:'||fare.reservation_charge);
dbms_output.put_line(' Superfast
Charge:'||fare.superfast_charge);
dbms_output.put_line(' Other Charge:'||fare.other_charge);
dbms_output.put_line(' Tatkal Charge:'||fare.tatkal_charge);
else
```

```
dbms_output.put_line('Train number not found');
end if;
end loop;
end;
/
```

```
SQL> set serveroutput on
SQL> declare
  2  cursor train_fare is
  3  select * from train_ticket_fare;
  4  fare train_fare%rowtype;
  5  train_number number;
  6  begin
  7  open train_fare;
  8  loop
  9  fetch train_fare into fare;
 10  exit when train_fare%notfound;
 11  train_number:=&train_number;
 12  if(fare.train_number=train_number) then
 13  dbms_output.put_line(' Base Fare:'||fare.base_fare);
 14  dbms_output.put_line(' Reservation
 15  Charge:'||fare.reservation_charge);
 16  dbms_output.put_line(' Superfast
 17  Charge:'||fare.superfast_charge);
 18  dbms_output.put_line(' Other Charge:'||fare.other_charge);
 19  dbms_output.put_line(' Tatkal Charge:'||fare.tatkal_charge);
 20  else
 21  dbms_output.put_line('Train number not found');
 22  end if;
 23  end loop;
 24  end;
 25  /
Enter value for train_number: 12138
old  11: train_number:=&train_number;
new  11: train_number:=12138;
Train number not found
Train number not found
Train number not found
Train number not found
Train number not found
Base Fare:920
Reservation
Charge:30
Superfast
Charge:50
Other Charge:13
Tatkal Charge:20

PL/SQL procedure successfully completed.
```

4. Write a cursor to update the reservation status of the passengers (generate seat number, if seats have reached maximum, put waiting list number (30% of total seats), if waiting list number reaches maximum, put PQWL (10%of total seats), RAC-20%)

```
set serveroutput on
declare
pnr ticket.pnr_no%type;
seat seats.no_of_seats%type;
rs passenger.reservation_status%type;
msg varchar(2);
wl number(2);
pqwl number(2);
begin
wl:=0;
pqwl:=0;
select pnr_no into pnr from ticket where train_number=12138;
select count(no_of_seats) into seat from seats where train_number=12138;
select reservation_status into rs from passenger where pnr_no=(select
pnr_no from ticket where train_number = 12138);
if seat > 100 then
dbms_output.put_line('Seats have reached Maximum… You are now added in
waiting list');
wl:=wl+1;
if wl>30 then
dbms_output.put_line('Waiting list reached maximum. You are now added in
PQWL');
pqwl:=pqwl+1;
end if;
else
dbms_output.put_line('Seats allocated');
end if;
end;
/
```

```
SQL> set serveroutput on
SQL> declare
  2  pnr ticket.pnr_no%type;
  3  seat seats.no_of_seats%type;
  4  rs passenger.reservation_status%type;
  5  msg varchar(2);
  6  wl number(2);
  7  pqwl number(2);
  8  begin
  9  wl:=0;
 10  pqwl:=0;
 11  select pnr_no into pnr from ticket where train_number=12138;
 12  select count(no_of_seats) into seat from seats where
 13  train_number=12138;
 14  select reservation_status into rs from passenger where
 15  pnr_no=(select pnr_no from ticket where train_number =
 16  12138);
 17  if seat > 100 then
 18  dbms_output.put_line('Seats have reached Maximum… You
 19  are now added in waiting list');
 20  wl:=wl+1;
 21  if wl>30 then
 22  dbms_output.put_line('Waiting list reached maximum. You
 23  are now added in PQWL');
 24  pqwl:=pqwl+1;
 25  end if;
 26  else
 27  dbms_output.put_line('Seats allocated');
 28  end if;
 29  end;
 30  /
```

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

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

```
create or replace procedure getnextp(pass out passenger%rowtype,pn out
passenger.pnr_no%type) is
begin
select pnr_no+0101010101 into pn from passenger where name ='&name';
select * into pass from passenger here pnr=pn;
end;
declare
pass passenger%rowtpye;
pn passenger.pnrno%type;
begin
getnextp(pass,pn);
dbms_output.put_line('PNR NO: '||pass.pnrno||'SERIAL
NO:'||pass.serial_no||'Name:'||pass.name||'Age:'||pass.age||'Resv
Status:'||pass.reservation_status);
end;
/
```

```
SQL> create or replace procedure getnextp(pass out passenger%rowtype,pn out passenger.pnr_no%type) is
  2  begin
  3  select pnr_no+0101010101 into pn from passenger where name ='&name';
  4  select * into pass from passenger here pnr=pn;
  5  end;
  6  declare
  7  pass passenger%rowtpye;
  8  pn passenger.pnr_no%type;
  9  begin
 10  getnextp(pass,pn);
 11  dbms_output.put_line('PNR NO: '||pass.pnrno||'SERIAL NO:'||pass.serial_no||'Name:'||pass.name||'Age:'||pass.age||'Resv Status:'||pass.reservation_status);
 12  end;
 13  /
Enter value for name: Priyal
old   3: select pnr_no+0101010101 into pn from passenger where name ='&name';
new   3: select pnr_no+0101010101 into pn from passenger where name ='Priyal';
```

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

```
set serveroutput on;
declare
sour ticket.from_station%type;
dest ticket.to_station%type;
pnr_number ticket.pnr_no%type;
begin
sour:='&sour';
dest:='&dest';
select pnr_no into pnr_number from ticket where from_station = sour and to_station =
dest;
dbms_output.put_line('Passenger Pnr:' ||pnr_number);
end;
/
```

```
SQL> set serveroutput on;
SQL> declare
  2  sour ticket.from_station%type;
  3  dest ticket.to_station%type;
  4  pnr_number ticket.pnr_no%type;
  5  begin
  6  sour:='&sour';
  7  dest:='&dest';
  8  select pnr_no into pnr_number from ticket where from_station = sour and to_station = dest;
  9  dbms_output.put_line('Passenger Pnr:' ||pnr_number);
 10  end;
 11  /
Enter value for sour: Mumbai
old   6: sour:='&sour';
new   6: sour:='Mumbai';
Enter value for dest: New Delhi
old   7: dest:='&dest';
new   7: dest:='New Delhi';
Passenger Pnr:3615847290

PL/SQL procedure successfully completed.
```

2. Write a PL/SQL function to.

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

```
set serveroutput on;
declare
p ticket.pnr_no%type;
tf ticket.ticket_fare%type;
function get_totalfare(pnr out ticket.pnr_no%type, tfare out ticket.ticket_fare%type)
return number is
begin
select ticket_fare into tfare from ticket where pnr_no=&pnr;
return tfare;
end;
begin
dbms_output.put_line('Train ticket fare:'||get_totalfare(p,tf));
end;
/
```

```
SQL> set serveroutput on;
SQL> declare
  2  p ticket.pnr_no%type;
  3  tf ticket.ticket_fare%type;
  4  function get_totalfare(pnr out ticket.pnr_no%type, tfare out ticket.ticket_fare%type) return number is
  5  begin
  6  select ticket_fare into tfare from ticket where pnr_no=&pnr;
  7  return tfare;
  8  end;
  9  begin
 10  dbms_output.put_line('Train ticket fare:'||get_totalfare(p,tf));
 11  end;
 12  /
Enter value for pnr: 1928091842
old   6: select ticket_fare into tfare from ticket where pnr_no=&pnr;
new   6: select ticket_fare into tfare from ticket where pnr_no=1928091842;
Train ticket fare:2100
```

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

```
set serveroutput on;
declare
na passenger.name%type;
pnr passenger.pnr_no%type;
tn ticket.train_number%type;
jt train.traveltime%type;
function get_journey_time(na out passenger.name%type,pnr out
passenger.pnr_no%type,tn out ticket.train_number%type,jt out
train.traveltime%type) return interval day to second is
begin
select name into na from passenger where name ='&na';
select pnr_no into pnr from passenger where name = na;
select train_number into tn from ticket where pnr_no = pnr;
select traveltime into jt from train where train_number=tn;
return jt;
end;
begin
dbms_output.put_line('Total journey time:'||
get_journey_time(na,pnr,tn,jt));
end;
/
```

```
SQL> set serveroutput on;
SQL> declare
  2  na passenger.name%type;
  3  pnr passenger.pnr_no%type;
  4  tn ticket.train_number%type;
  5  jt train.traveltime%type;
  6  function get_journey_time(na out passenger.name%type,pnr out passenger.pnr_no%type,tn out ticket.train_number%type,jt out train.traveltime%type) return interval day to second is
  7  begin
  8  select name into na from passenger where name ='&na';
  9  select pnr_no into pnr from passenger where name = na;
 10  select train_number into tn from ticket where pnr_no = pnr;
 11  select traveltime into jt from train where train_number=tn;
 12  return jt;
 13  end;
 14  begin
 15  dbms_output.put_line('Total journey time:'||
 16  get_journey_time(na,pnr,tn,jt));
 17  end;
 18  /
Enter value for na: Priyal
old   8: select name into na from passenger where name ='&na';
new   8: select name into na from passenger where name ='Priyal';
Total journey time:+01 04:55:00.000000

PL/SQL procedure successfully completed.
```

******