Class List#:

Pc#:

Tshwane University of Technology
We empower people

FACULTY Of INFORMATION, COMMUNICATION AND TECHNOLOGY

DEPARTMENT COMPUTER SCIENCE
SOFTWARE ENGINEERING SECTION

SUMMATIVE November 2020

Subject Code : DSO23BT/SFW20BT

Subject Name : Development Software 2B/ Software Skills 2B

Examiners : Mr. S.K Mogapi

Moderator : Mr F Kgoete

Marks : 75 Time : 180 Minutes

Total Pages(Including this one) : 08

Rules

1. Do not tear any page from this test paper. Read and understand the question before you answer. Answer all the questions.
2. Your answers must produce the same output as the ones the examiner(s) has given as examples.
3. Complete the information on this cover page before you start answering questions.
4. The question paper is based on the **AMW database**. Install the database on your computer.
5. Use a notepad to answer and save it as **StudentNo_SUMMATIVE_2020_1.txt** and upload your work into the given slot alphabet.
6. It's your responsibility to make it a point that you follow the rules. Empty notepad means 0% mark.
7. Copy the following to your Sql editor: Set linesize 250
8. Go to Properties of SQL editor and click Layout, increase the window size to 200

Questions	Ques 1	Ques 2	Ques 3	Ques 4	Full Marks
Marks	10	18	21	25	75
Allocation					

Total: 75

%

Question 1 **[10]**

Write a PL/SQL block that prompts a user to enter the car registration and the block must retrieve the number of times the car was successfully serviced. The value the user must enter must be in any alphabetic case and it must display old and new values. Format your output as the one below.

```
Enter value for car_registration: prt344gp
SS MBELE has taken a GREEN MAHINDRA BOLERO 2.5 TD for maintenance 1 time(s).
Enter value for car_registration: dzz433fs
CME MARAIS has taken a CANDY FERRARI 360 SPIDER F1 for maintenance 2 time(s).
```

```
DECLARE
 v_surname       client.surname%TYPE;
 v_initials      client.initials%TYPE;
 v_regno         vehicle.regno%TYPE:=UPPER('&car_registration')✓; No Upper ½ a mark
 v_make          vehicle.make%TYPE; ✓
 v_model             vehicle.model%TYPE; ✓
 v_colour            vehicle.colour%TYPE; ✓
 v_tot_service   NUMBER(3); ✓
BEGIN
 SELECT surname,initials,make,model,colour,COUNT(jc.regno)✓ tot_service ✓
 INTO v_surname,v_initials,v_make,v_model,v_colour,v_tot_service ✓
 FROM client c,vehicle v,jobcard jc ✓
 WHERE v.clcode=c.clcode
 AND v.regno=v_regno✓
 AND jc.regno=v.regno✓
 GROUP BY surname,initials,make,model,colour; ✓
 DBMS_OUTPUT.PUT_LINE(v_initials||' '||v_surname||' has taken a '||UPPER ✓ (v_colour||'
'||v_make||' '||v_model)||' for maintenance '||v_tot_service||' time(s).'); ✓✓
END;
/
```

Question 2 **[18]**

Create a PL/SQL program declares a record **client_vehicle_rec** that retrieves and display the number of vehicles that a client brought in for service. Your program must prompt the user to enter the client code. If the client did not bring in the vehicle for service, display a message **"Client hasn't brought in vehicle for service"** by making use of a NON-PREDIFENED exception. Your output must resemble the one below.

```
Enter value for code: MAT001
Client Mathonsi ( 7402280778082)brought 2 vehicle for service

PL/SQL procedure successfully completed.

SQL> /
Enter value for code: BOT001
Client BOTHA ( 7802130344074)brought 1 vehicle for service

PL/SQL procedure successfully completed.

SQL> /
Enter value for code: MBE001
Client MBELE ( 8012310576081)brought 1 vehicle for service

PL/SQL procedure successfully completed.

SQL> /
Enter value for code: MAT002
Client hasnt brought in vehicle for service

PL/SQL procedure successfully completed.
```

```
DECLARE

type client_vehicle_rec is record ✓
(
v_surname client.surname%type,
v_idno client.idno%type,                    ✓✓
v_count number(5)
);

v_clcode client.clcode%type; ✓
client_vehicle_record client_vehicle_rec; ✓

NO_DATA EXCEPTION; ✓
PRAGMA EXCEPTION_INIT(NO_DATA,+100); ✓

begin

select surname,idno,count(regno) ✓✓
into client_vehicle_record✓
from client c,vehicle v✓
where c.clcode=v.clcode✓
and C.clcode='&Code'✓
```

```
group by c.clcode,surname,idno; ✓

dbms_output.put_line('Client '||client_vehicle_record.v_surname||' (
'||client_vehicle_record.v_idno||')' ||'brought '||client_vehicle_record.v_count||' vehicle for service');
✓✓

EXCEPTION

WHEN NO_DATA THEN ✓

dbms_output.put_line('Client hasnt brought in vehicle for service'); ✓

end;
```

Question 3 **[21]**

Run the table script to create a table named client_vehicle_older_coetzee.

```
SQL> create table client_vehicle_older_coetzee
  (surname varchar2(15),
  idno varchar2(15),
  make varchar2(10),
  year varchar2(25)
  );

Table created.
```

Write program that declares an implicit cursor **client_vehicle_older_coetzee** to retrieve all vehicles that are older than the vehicle owned by Coetzee. The cursor must retrieve vehicle owner's surname, id number, vehicle make and year. The program must pass the cursor's information into a record **vehicle_rec** by making use of **%rowtype**, and it must then insert all records retrieved by a cursor and insert them into a table **client_vehicle_older_coetzee**. Ensure that the id number is right padded showing the last 5 digits padded with stars. The vehicle year must be displayed in words as shown in the output below.

```
PL/SQL procedure successfully completed.

SQL> SELECT * FROM client_vehicle_older_coetzee;

SURNAME         IDNO          MAKE       YEAR
--------------- --------------- ---------- ------------------------
Nkosi           93087*****    VW         Two Thousand Fourteen
Zwane           34084*****    VW         Two Thousand Fifteen
Zwane           34084*****    VW         Two Thousand Fifteen
Zwane           34084*****    Maserati   Two Thousand Eleven
Marais          56086*****    Porsche    Two Thousand Fifteen
```

```
declare

cursor client_vehicle_older_coetzee is ✓
select surname,idno,v.make,year  ✓
from client c, vehicle v ✓
where c.clcode=v.clcode ✓
and year >all (select year  ✓✓
              from vehicle v,client c  ✓
            where c.clcode=v.clcode ✓
            and c.surname='COETZEE'); ✓

vehicle_rec client_vehicle_older_coetzee%rowtype; ✓✓

begin
open client_vehicle_older_coetzee; ✓
loop
fetch client_vehicle_older_coetzee ✓
into vehicle_rec; ✓
exit when client_vehicle_older_coetzee%notfound; ✓
INSERT INTO client_vehicle_older_coetzee✓
values (initcap(vehicle_rec.surname),rpad(substr(vehicle_rec.idno,-5),10,'*')
✓✓,vehicle_rec.make,to_char(to_date(vehicle_rec.year,'j'),'Jsp') ✓✓);

end loop;
close client_vehicle_older_coetzee; ✓

end;
```

Question 4 **[25]**

A program is needed that will produce the following output about vehicle's service together with the rate and the number of hours spent to service the vehicle.

```
SQL> /
Enter value for regno: RTF893GP
Mathonsi RTF893GP Ford
Vehicle kilometers: 44788
Service rate 2915
Service hours :2.5
==================================

PL/SQL procedure successfully completed.

SQL> /
Enter value for regno: BOT002
No such record

PL/SQL procedure successfully completed.

SQL> /
Enter value for regno: FRD998NW
ZWANE FRD998NW VW
There are too many rows retrieved

PL/SQL procedure successfully completed.
```

4.1 Create a pl/sql program that declares a procedure that will accept vehicle's registration number and later retrieves and displays the vehicle's owner together with the vehicle's make. This procedure must display the below out. (7)

```
SQL> /
Enter value for regno: RTF893GP
Mathonsi RTF893GP Ford
```

```
Create or replace procedure client_proc ✓
(p_regno IN OUT vehicle.regno%type) ✓
is
v_Clcode client.clcode%type;
v_surname client.surname%type; ✓
v_make vehicle.make%type;
begin
select c.clcode,surname,regno,make ✓
into v_Clcode,v_surname,p_regno,v_make ✓
from client c,vehicle v
where c.clcode=v.clcode ✓
and v.regno=p_regno;

dbms_output.put_line(v_surname||' '||p_regno||' '||v_make); ✓
end client_proc;
```

4.2. Create a pl/sql function that declares a function to determine the new service price which is 10% more from the normal service rate. This function must accept vehicle's registration number and later display the new service rate and the number of hours spent to service the vehicle. This function must display the below out. (13)

```
Vehicle kilometers: 44788
Service rate 2915
Service hours :2.5
```

```
Create or replace function client_service ✓
(p_regno IN OUT vehicle.regno%type) ✓
return number ✓
is
v_regno vehicle.regno%type;
v_km jobcard.kmreading%type;
v_srate service.srate%type;                ✓
v_shrs service.servicehours%type;

begin
select v.regno,kmreading,srate,servicehours ✓
into v_regno,v_km,v_srate,v_shrs ✓
from client c,vehicle v,jobcard jc,jobservice js,service s ✓✓
where c.clcode=v.clcode
and v.regno=jc.regno
and jc.jobcardno=js.jobcardno              ✓✓
and s.scode=js.scode
and v.regno=p_regno;

v_srate:= v_srate*1.10; ✓

dbms_output.put_line('Vehicle kilometers: '||v_km);
dbms_output.put_line('Service rate '||v_srate);          ✓✓
dbms_output.put_line(' Service hours :'||v_shrs);

return v_srate;


end client_service;
```

4.3 Create an anonymous block that will call the above procedure in 5.1 together with the above function in 5.2. This block must prompt the user to enter the vehicle's registration number. If there are too many rows returned, allow the program to trap the error and display "There are too many rows retrieved". If no records found the program must trap the error and display "No such record". (5)

```
==================================

PL/SQL procedure successfully completed.

SQL> /
```

```
Enter value for regno: BOT002
No such record

PL/SQL procedure successfully completed.

SQL> /
Enter value for regno: FRD998NW
ZWANE FRD998NW VW
There are too many rows retrieved

PL/SQL procedure successfully completed.
```

```
declare

v_Clcode client.clcode%type;
v_surname client.surname%type;
v_make vehicle.make%type;
v_regno vehicle.regno%type:='&regno';
v_km jobcard.kmreading%type;
v_srate service.srate%type;
v_shrs number(7,2);

begin
client_proc(V_regno);
v_shrs:=client_service(V_regno);



dbms_output.put_line('==================================');

exception

when too_many_rows then

dbms_output.put_line(' There are too many rows retrieved');

when no_data_found then

dbms_output.put_line(' No such record');


end;
```