

Kosta Nikopoulos  
March 30th, 2021

## Lab 10

I wasn't sure if procedure was needed because the function is what will carry out the actions and return a value in this case a Car ID. Function is also needed for testing the PL/SQL Blocks since it returns values.

### Part A)

The Procedure: carries out actions

The screenshot displays the Oracle SQL Developer interface. On the left, the 'Connections' pane shows a tree view of the database schema, including tables like CAR, CUSTOMER, MECHANIC, PARTS, PARTSUSED, SALES\_INVOICE, SALESPERSON, SERVICE, SERVICE MECHANIC, and SERVICE TICKET. The 'Procedures' folder is expanded, showing 'PROC\_CAR\_INFO' and 'VAR\_CAR\_ID'. The main window is the 'SQL Worksheet' for 'lab10.sql', which contains the following PL/SQL code:

```
CREATE OR REPLACE PROCEDURE proc_Car_Info(  
  Var_car_id IN NUMBER  
)  
IS  
  --declaring locale variables/reference types  
  Var_c_model Car.c_model%TYPE;  
  Var_serial_number Car.serial_number%TYPE;  
BEGIN  
  --implicit cursor to fetch info about specified car  
  SELECT c_model,  
         serial_number  
    INTO Var_c_model,  
         Var_serial_number  
  FROM Car  
 WHERE car_id = Var_car_id;  
  DBMS_OUTPUT.PUT_LINE('This cars serial number is:' || Var_serial_number) ;  
  DBMS_OUTPUT.PUT_LINE('This cars model is:' || Var_c_model) ;  
  DBMS_OUTPUT.PUT_LINE(concat('Car ID', Var_car_id) );  
  
  --handle the exception that might occur  
EXCEPTION  
  WHEN TOO_MANY_ROWS  
  THEN  
    DBMS_OUTPUT.PUT_LINE('You have too many rows here be more specific...');  
    DBMS_OUTPUT.PUT_LINE(SQLERRM);  
  WHEN NO_DATA_FOUND  
  THEN  
    DBMS_OUTPUT.PUT_LINE('No data found for Car ID ' || Var_car_id);  
END;
```

At the bottom, the 'Script Output' pane shows the message: 'Task completed in 0.066 seconds' and 'Procedure PROC\_CAR\_INFO compiled'.

**The Function: carries out actions and returns a value**

The screenshot displays the Oracle SQL Developer interface. On the left, the 'Connections' pane shows a tree view of the database schema, including tables, views, indexes, packages, procedures, and functions. The 'Functions' folder is expanded, showing the function 'FN\_CAR\_INFO' with its return type 'RETURN' and parameter 'VAR\_CAR\_ID'. The main window shows the 'SQL Worksheet' with the following SQL code:

```
CREATE OR REPLACE FUNCTION FN_Car_Info(  
  Var_car_id IN NUMBER  
)  
RETURN VARCHAR2  
IS  
  --declaring locale variables/reference types  
  Var_c_model Car.c_model%TYPE;  
  Var_serial_number Car.serial_number%TYPE;  
BEGIN  
  --implicit cursor to fetch info about specified car  
  SELECT c_model,  
         serial_number  
    INTO Var_c_model,  
         Var_serial_number  
  FROM Car  
 WHERE car_id = Var_car_id;  
  DBMS_OUTPUT.PUT_LINE('This cars serial number is:' ||Var_serial_number) ;  
  DBMS_OUTPUT.PUT_LINE('This cars model is:' ||Var_c_model) ;  
  DBMS_OUTPUT.PUT_LINE(concat('Car ID', Var_car_id) );  
RETURN 'This is your CAR ID';  
--handle the exception that might occur  
EXCEPTION  
  WHEN TOO_MANY_ROWS  
  THEN  
    DBMS_OUTPUT.PUT_LINE('You have too many rows here be more specific...');  
    DBMS_OUTPUT.PUT_LINE(SQLERRM);  
  WHEN NO_DATA_FOUND  
  THEN  
    DBMS_OUTPUT.PUT_LINE('No data found for Car ID '||Var_car_id);  
END;
```

The 'Script Output' pane at the bottom shows the execution results:

```
Task completed in 0.043 seconds  
  
Procedure PROC_CAR_INFO compiled  
  
Function FN_CAR_INFO compiled
```

## Part B)

### TEST 1:

The screenshot shows the Oracle SQL Developer interface. On the left, the 'Connections' pane lists the 'xe-kosta2' connection. The main window displays a SQL Worksheet with the following code:

```
--B) Part B
--1. Write three different anonymous PL/SQL blocks that test your procedure with three different car IDs.
--test 1
DECLARE
    test VARCHAR2(30);
BEGIN
    test := FN_Car_Info(117);
END;
```

The 'Script Output' pane shows the message: 'PL/SQL procedure successfully completed.' The 'Dbms Output' pane shows the results of the procedure call:

```
xe-kosta2 x
This cars serial number is:21547
This cars model is:Accord
Car ID117
```

### TEST 2:

The screenshot shows the Oracle SQL Developer interface. On the left, the 'Connections' pane lists the 'xe-kosta2' connection. The main window displays a SQL Worksheet with the following code:

```
END;
--test 2
DECLARE
    test VARCHAR2(30);
BEGIN
    test := FN_Car_Info(777);
END;
--test 3
DECLARE
```

The 'Script Output' pane shows the message: 'Task completed in 0.035 seconds'. The 'Dbms Output' pane shows the results of the procedure call:

```
xe-kosta2 x
This cars serial number is:21547
This cars model is:Accord
Car ID117

This cars serial number is:21547
This cars model is:Corvette
Car ID777
```

### TEST 3:

The screenshot displays the SQL Developer environment. On the left, the 'Connections' pane shows a tree view of the database schema for 'xe-kosta2', including tables like CAR, CUSTOMER, MECHANIC, PARTS, and various views and procedures. The main workspace is divided into several panes. The top pane shows the 'SQL Worksheet' with a query: 

```
test := FN_Car_Info(99);  
END;  
  
--Select statements to test numbers  
select car_id  
FROM Car  
WHERE car_id = 117;
```

 Below this, the 'Script Output' pane indicates 'Task completed in 0.019 seconds'. The 'Dbms Output' pane shows the results of the query execution for the 'xe-kosta2' connection. The output is as follows:

Output Line	Text
1	This cars serial number is:21547
2	This cars model is:Accord
3	Car ID117
4	 
5	This cars serial number is:21547
6	This cars model is:Corvette
7	Car ID777
8	 
9	This cars serial number is:1279
10	This cars model is:Huracan
11	Car ID99