

Kosta Nikopoulos
March 31st, 2021

Lab 11

Part 1:

The Car Procedure

The screenshot shows the Oracle SQL Developer interface. On the left, the 'Connections' pane displays a tree view of the database schema, including tables like CAR, CUSTOMER, MECHANIC, PARTS, and PROC_CAR_INFO. The 'PROC_CAR_INFO' procedure is selected. The main window shows the 'Worksheet' tab with the following SQL code:

```
--lab 11
--PL/SQL Stored Procedures and Functions
--Kosta Nikopoulos

--Part 1
CREATE OR REPLACE PROCEDURE proc_Car_Info(
  in_car_id IN NUMBER
)
IS
  --declaring locale variables/reference types
  r_car Car%ROWTYPE;
BEGIN
  --implicit cursor to fetch info about specified car
  SELECT *
  INTO r_car
  FROM Car
  WHERE car_id = in_car_id;
  DBMS_OUTPUT.PUT_LINE('This cars serial number is: '||r_car.serial_number);
  DBMS_OUTPUT.PUT_LINE('This cars model is: '||r_car.c_model);
  DBMS_OUTPUT.PUT_LINE(concat('Car ID: ', in_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...');
  WHEN NO_DATA_FOUND
  THEN
    DBMS_OUTPUT.PUT_LINE('No data found for Car ID: '||in_car_id);
END;
```

The 'Script Output' pane at the bottom shows the message: 'Procedure PROC_CAR_INFO compiled'.

--Testing the Car Procedure

The screenshot shows the Oracle SQL Developer interface with the 'PROC_CAR_INFO' procedure being tested. The 'Worksheet' tab contains the following SQL code:

```
--Block Testing
--test 1
BEGIN
  proc_car_info(117);
END;
/
--test 2
BEGIN
  proc_car_info(777);
END;
/
--test 3
BEGIN
  proc_car_info(138);
END;
```

The 'Script Output' pane shows the message: 'Procedure PROC_CAR_INFO compiled' and 'PL/SQL procedure successfully completed.'.

The 'DBMS Output' pane shows the results of the tests:

```
xe-kosta2 *
This cars serial number is: 21547
This cars model is:Accord
Car ID: 117

This cars serial number is: 21547
This cars model is:Corvette
Car ID: 777

This cars serial number is: 67347
This cars model is:Camrie
Car ID: 138
```

Part 2:

The Customer Function

The screenshot displays the Oracle SQL Developer interface. On the left, the 'Connections' pane shows a tree view of the database schema, with 'FN_CUSTOMER_INFO' selected under the 'Functions' folder. The main window is titled 'lab11.sql' and contains the following SQL code:

```
--Part 2
CREATE OR REPLACE FUNCTION FN_customer_info(
  in_phone_number IN CHAR
)
RETURN VARCHAR2
IS
  r_cus_info customer%ROWTYPE;
BEGIN
  --fetch info about specified customer
  SELECT
    *
  INTO r_cus_info
  FROM customer
  WHERE phone_number = in_phone_number;
  DBMS_OUTPUT.PUT_LINE(
    'Name: ' || r_cus_info.first_name || ' ' || r_cus_info.last_name || ' ' ||
    'Address: ' || r_cus_info.address || ' ' ||
    'City: ' || r_cus_info.city || ' ' ||
    'State: ' || r_cus_info.cus_state || ' ' ||
    'Country: ' || r_cus_info.country || ' ' ||
    'Postal Code: ' || r_cus_info.postal_code || ' ' ||
    ' ID: ' || r_cus_info.customer_id);
  RETURN 'customer info';
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 customer phone number: ' || in_phone_number);
END;
```

Below the code editor, the 'Script Output' pane shows the message: 'Function FN_CUSTOMER_INFO compiled'. The status bar at the bottom indicates 'Task completed in 0.198 seconds'.

--Testing the Customer Function

The screenshot displays the Oracle SQL Developer environment. On the left, the 'Connections' pane shows a tree view of the database schema, including tables, views, indexes, packages, procedures, functions, and operators. The 'xe-kosta2' connection is selected. The main window shows a SQL worksheet with a PL/SQL script being executed. The script consists of three test blocks, each declaring a VARCHAR2 variable and calling the FN_customer_info function with a specific phone number. The 'Script Output' pane shows three messages: 'PL/SQL procedure successfully completed.' for each test. The 'Dbms Output' pane shows the results of the FN_customer_info function for three different phone numbers.

```
DECLARE
  test VARCHAR2(50);
BEGIN
  test := FN_customer_info('666-666-6666');
END;
/
--test 2
DECLARE
  test VARCHAR2(50);
BEGIN
  test := FN_customer_info('111-222-3333');
END;
/
--test 3
DECLARE
  test VARCHAR2(50);
BEGIN
  test := FN_customer_info('555-888-6666');
END;
/
```

Script Output x

Task completed in 0.064 seconds

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

Dbms Output

xe-kosta2 x

Name: Lucifer Morningstar	Address: 666th street	City: LosAngeles	State: California	Country: USA	Postal Code: A6L6G6	ID: 666
Name: Jack Newman	Address: 1st street	City: Victoria	State: BC	Country: Canada	Postal Code: E4T6D6	ID: 333
Name: Jamal Jackson	Address: 3rd street	City: Las Vegas	State: Nevada	Country: USA	Postal Code: C0T9F3	ID: 444