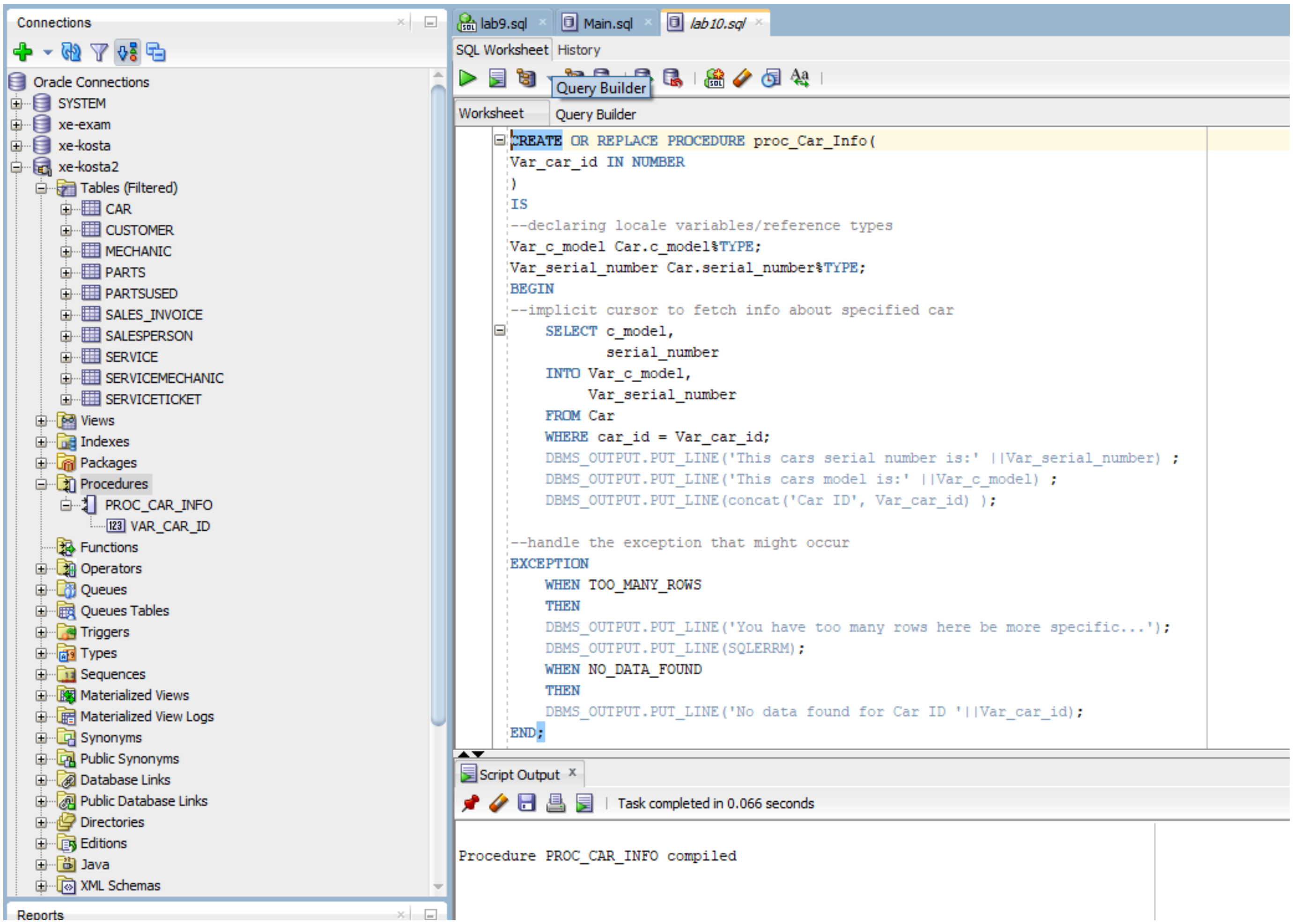
Kosta Nikopoulos
March 30th, 2021

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



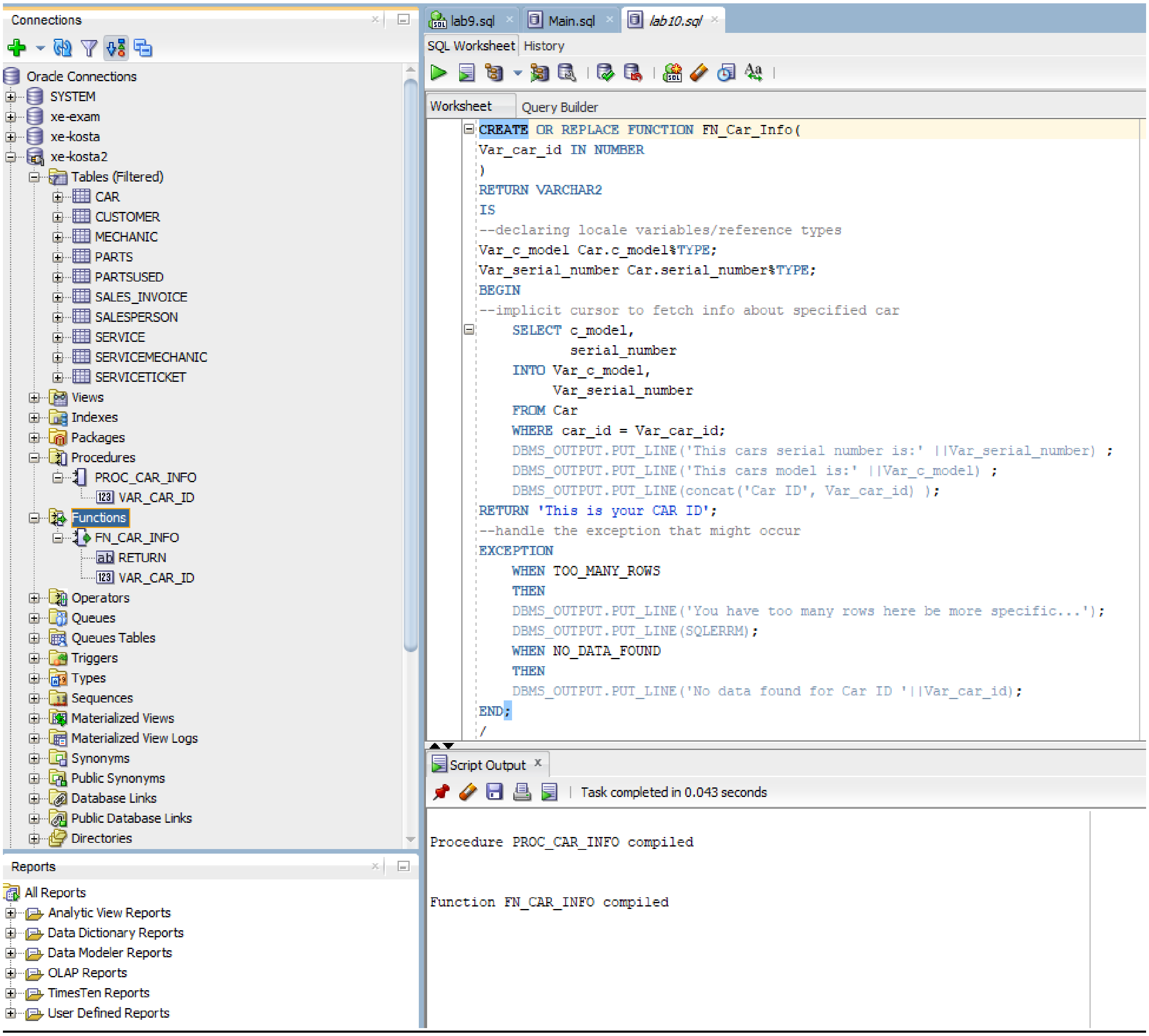
```
CREATE OR REPLACE PROCEDURE proc_Car_Info(
Var_car_id IN NUMBER
)
IS
--declaring locale variables/reference types
Var_c_model Car.c_model%TYPE;
Var_serial_number Car.serial_number%TYPE;
BEGIN
--implicit cursor to fetch info about specified car
    SELECT c_model,
           serial_number
    INTO 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;
```

Task completed in 0.066 seconds

```
Procedure PROC_CAR_INFO compiled
```

The Function: carries out actions and returns a value



```
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;
/
```

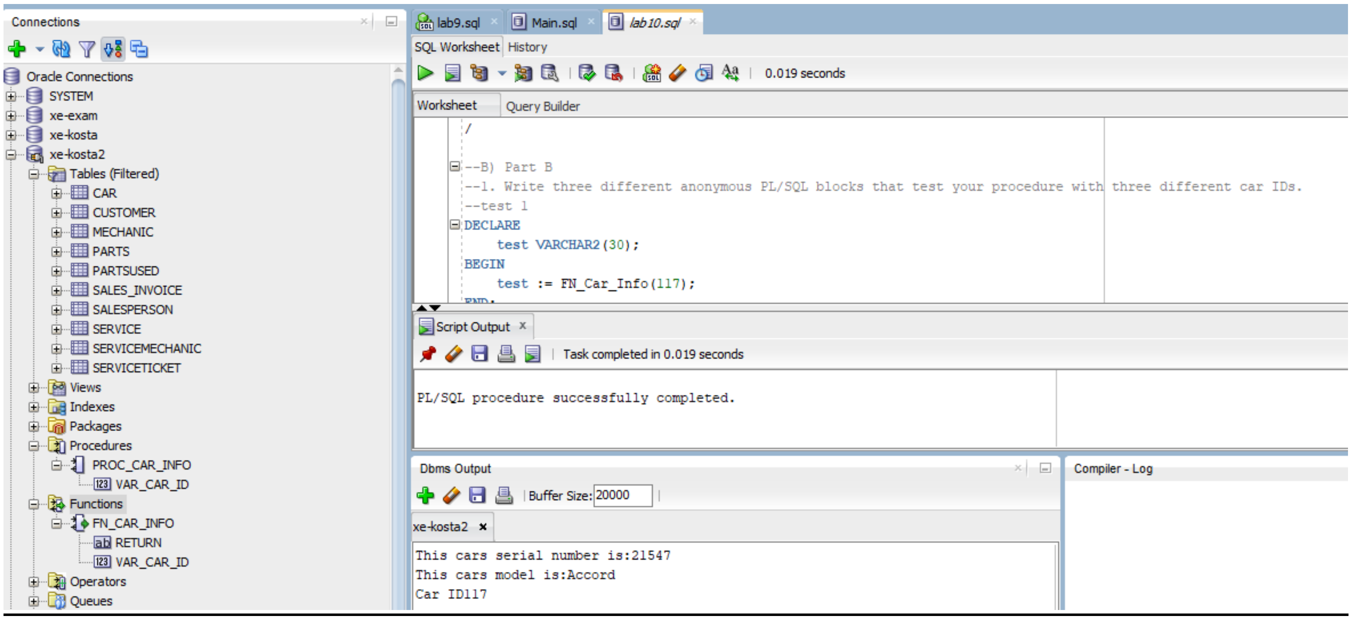
Task completed in 0.043 seconds

```
Procedure PROC_CAR_INFO compiled


Function FN_CAR_INFO compiled
```

Part B)

TEST 1:



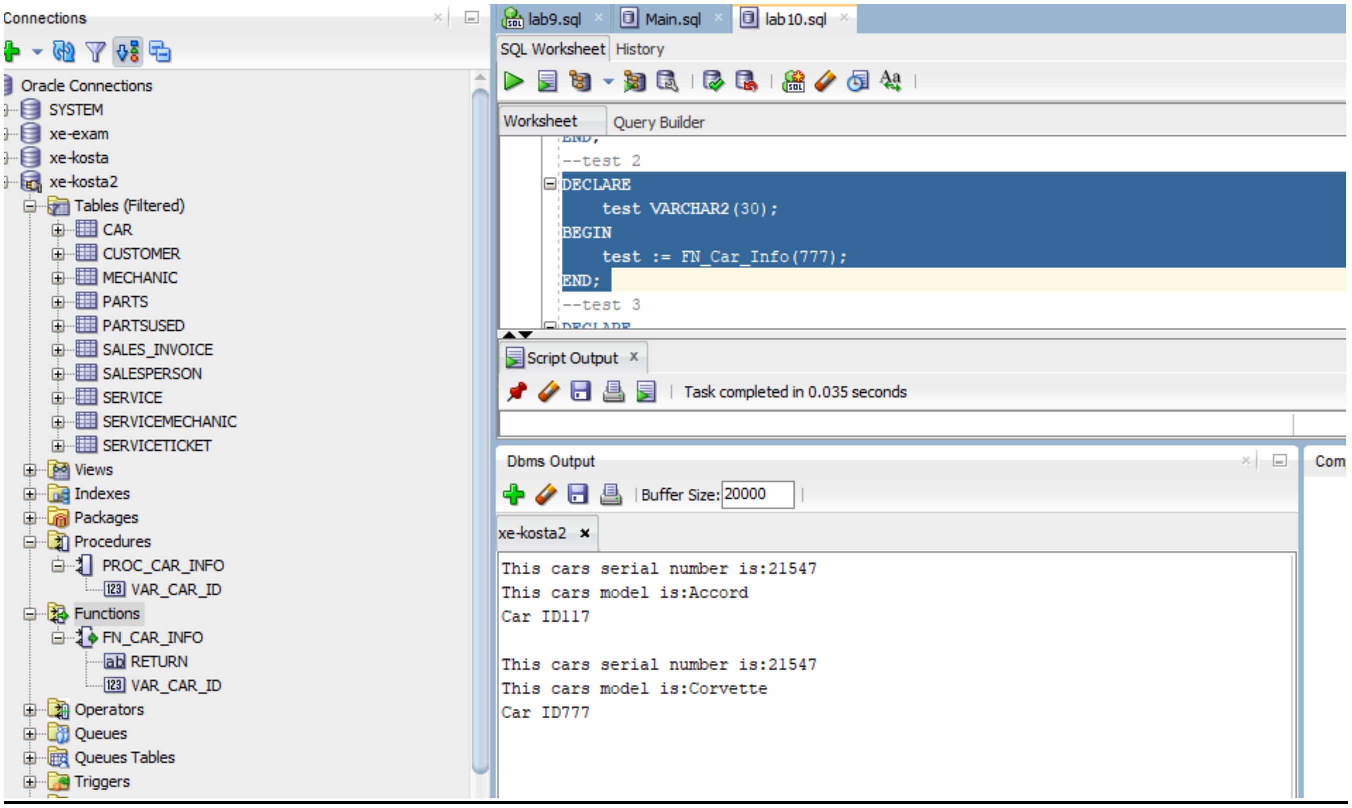
```
/

--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;
```

```
PL/SQL procedure successfully completed.
```

```
This cars serial number is:21547
This cars model is:Accord
Car ID117
```

TEST 2:

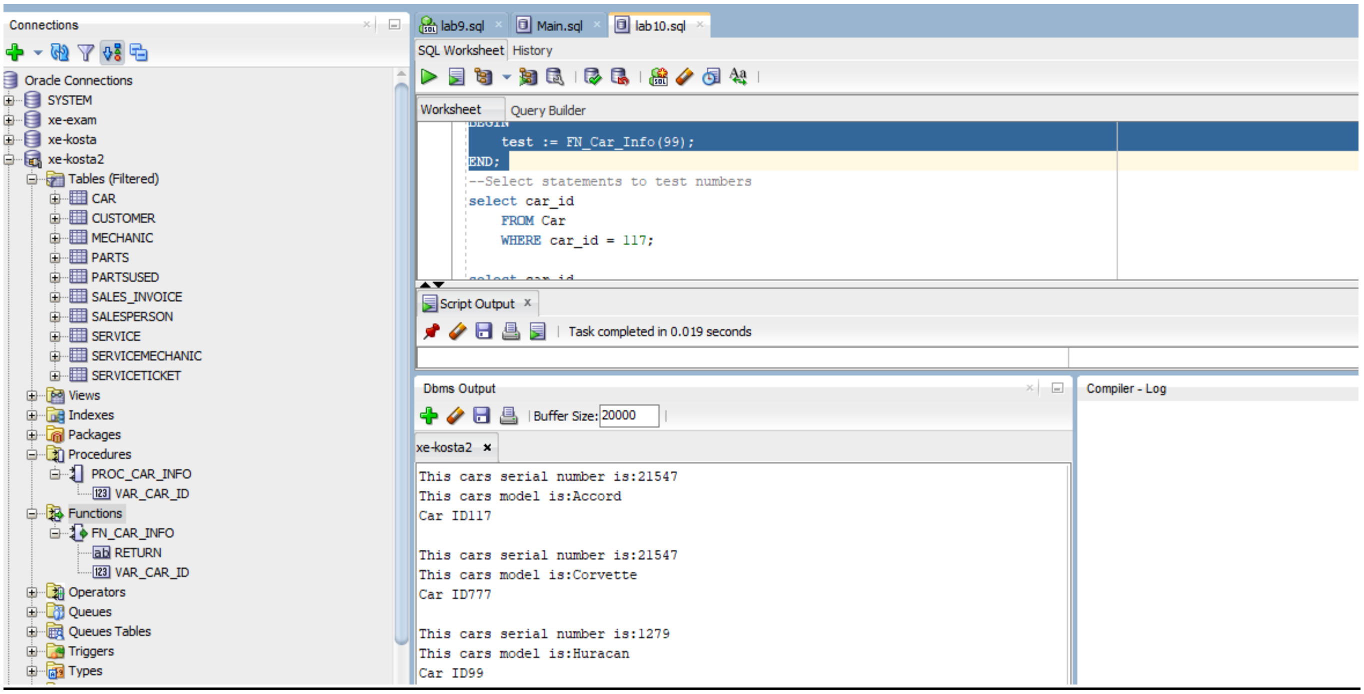


```
END;
--test 2
DECLARE
    test VARCHAR2(30);
BEGIN
    test := FN_Car_Info(777);
END;
--test 3
DECLARE
```

```
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:



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

select car_id
```

Task completed in 0.019 seconds

```
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

This cars serial number is:1279
This cars model is:Huracan
Car ID99
```