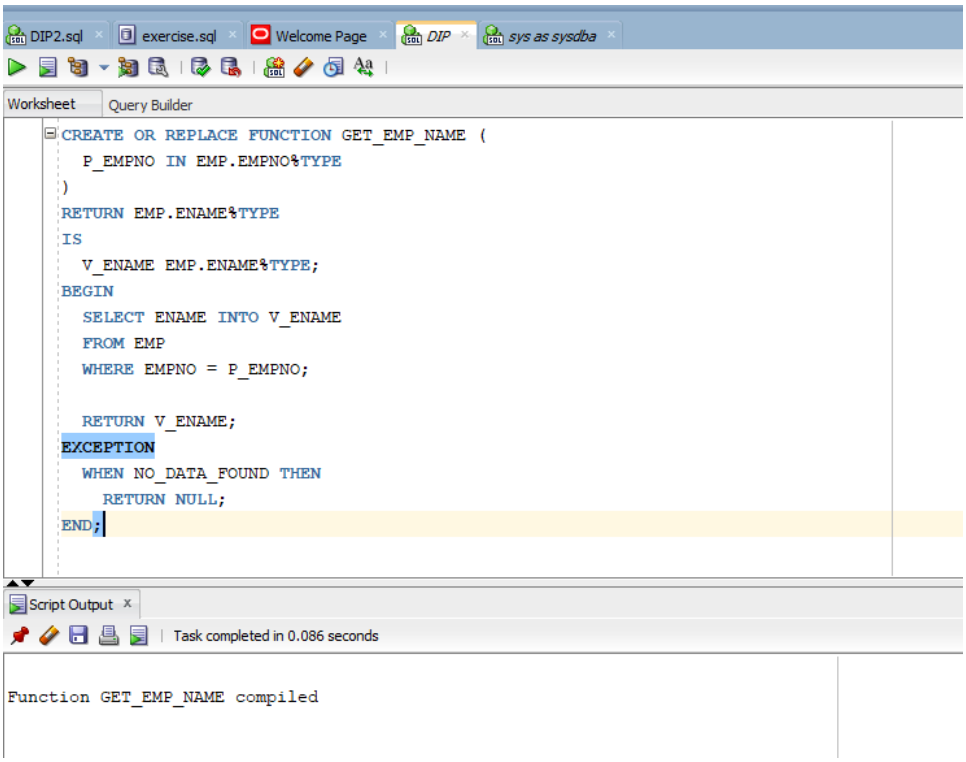
Advanced Database - TP 2 - Part 2

PL/SQL

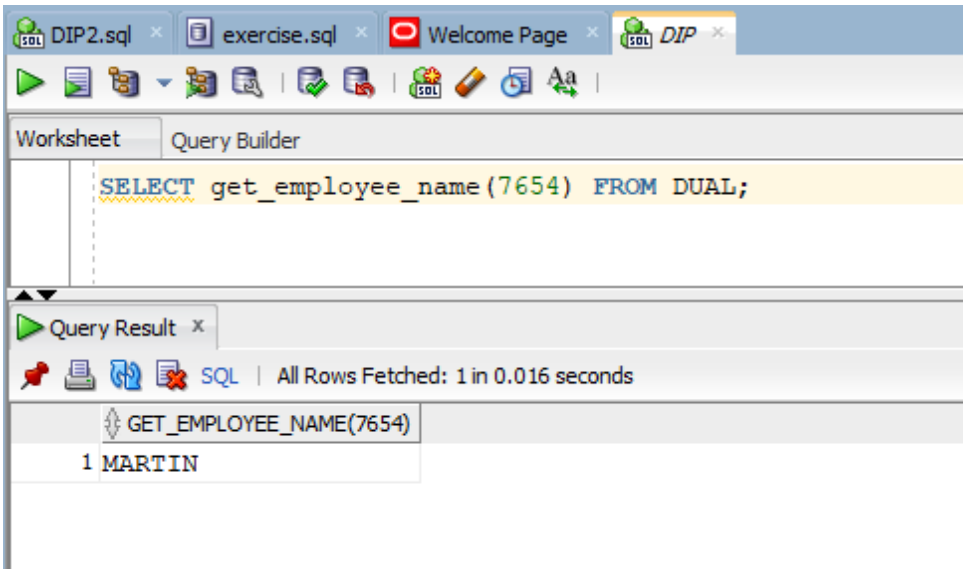
Name: Sayed Mujtaba Lastname: Ahmady STDNO: 62707

1. Create a function that gets an employee's name from it's empno.

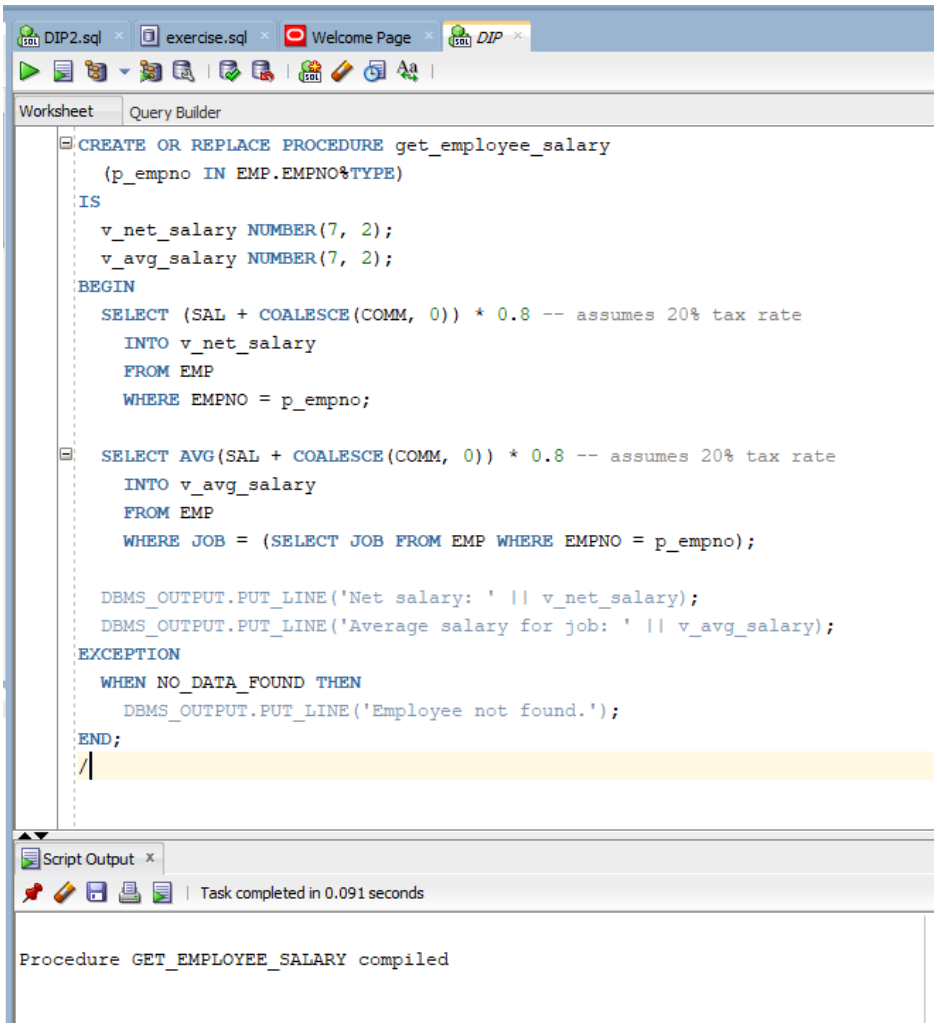


2. Test the query with

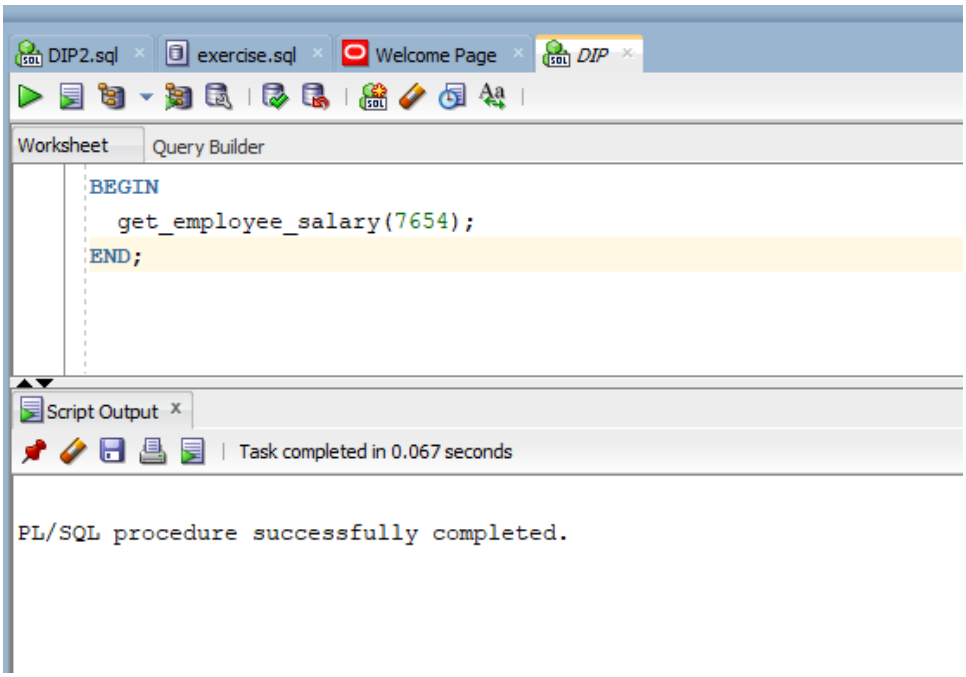
```
SELECT FunctionName(7654) FROM DUAL;
```



Exercice 2. Procedure & Display & Cursors

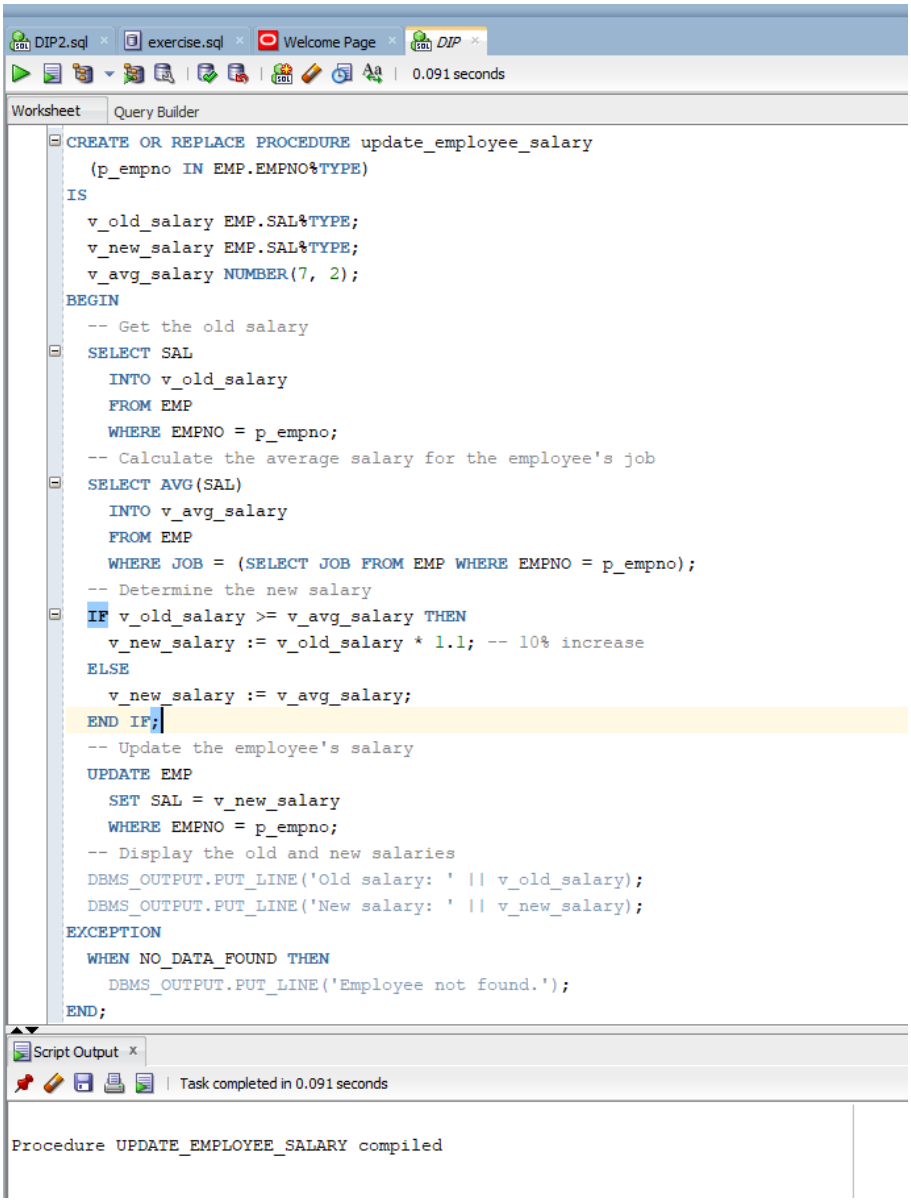


This will display the net salary and average salary for the employee with empno 7654 (assuming that employee exists in the EMP table). The DBMS_OUTPUT.PUT_LINE statements are used to display the results in the console.

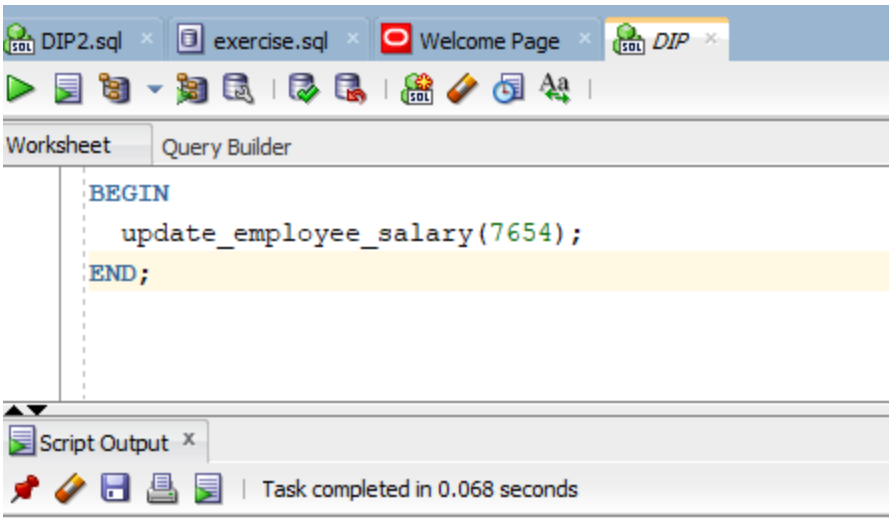


Exercice 3. Procedure & Update

This procedure takes an employee number (empno) as input and updates the employee's salary based on whether their current salary is greater than or equal to the average salary for their job. If their salary is higher than or equal to the average, it increases their salary by 10%. If it's lower than the average, their new salary becomes the average.



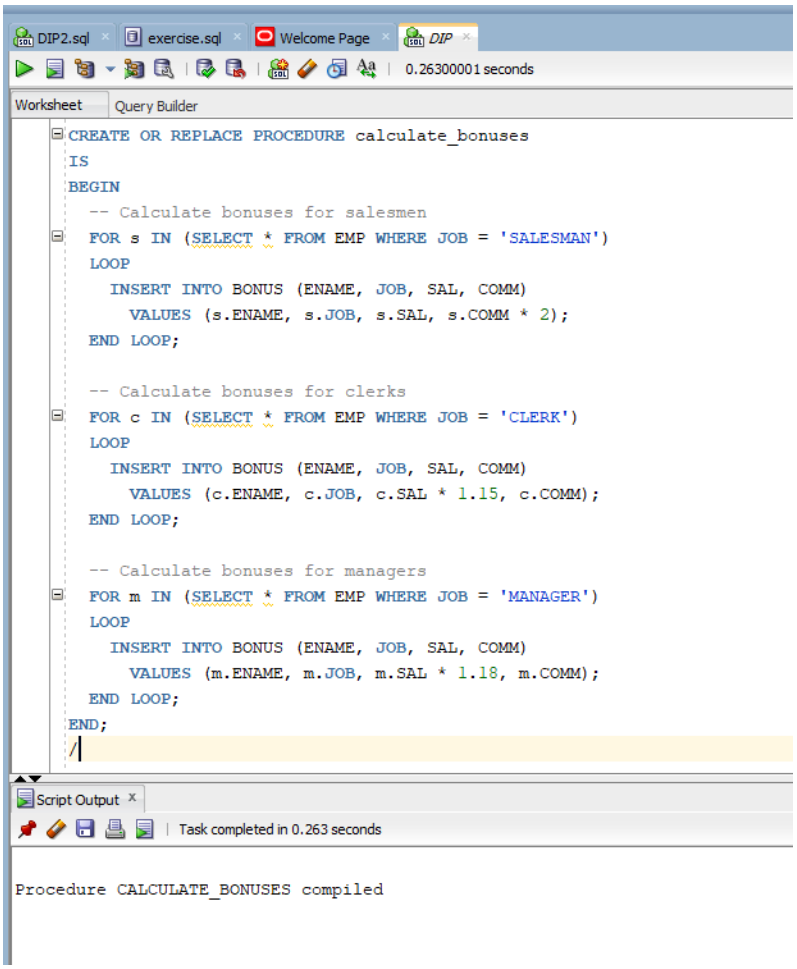
This will update the salary for the employee with empno 7654 and display their old and new salaries.



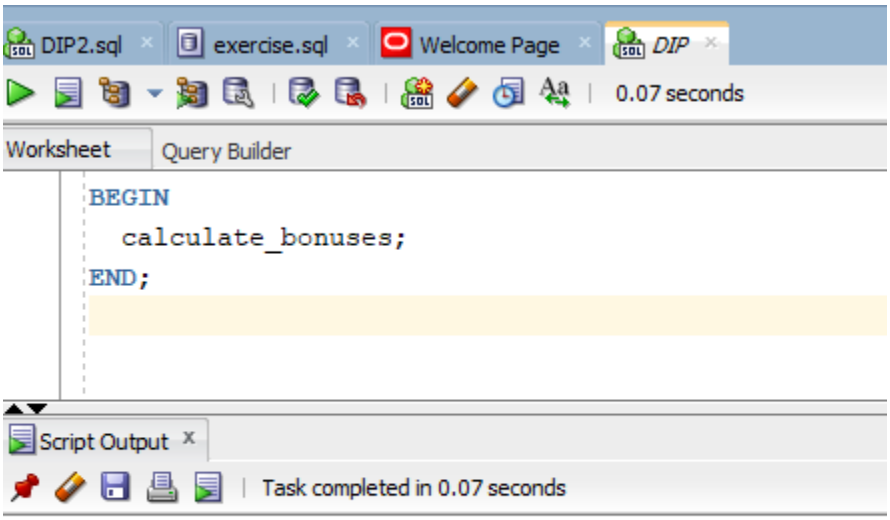

```
PL/SQL procedure successfully completed.
```

Exercice 4. Procedure

This procedure loops through the EMP table and calculates bonuses for employees based on their job. For salesmen, it doubles their commission (COMM * 2). For clerks, it increases their salary by 15% (SAL * 1.15). For managers, it increases their salary by 18% (SAL * 1.18). The calculated bonuses are then inserted into the BONUS table.



This will calculate and insert bonuses for all employees in the EMP table.

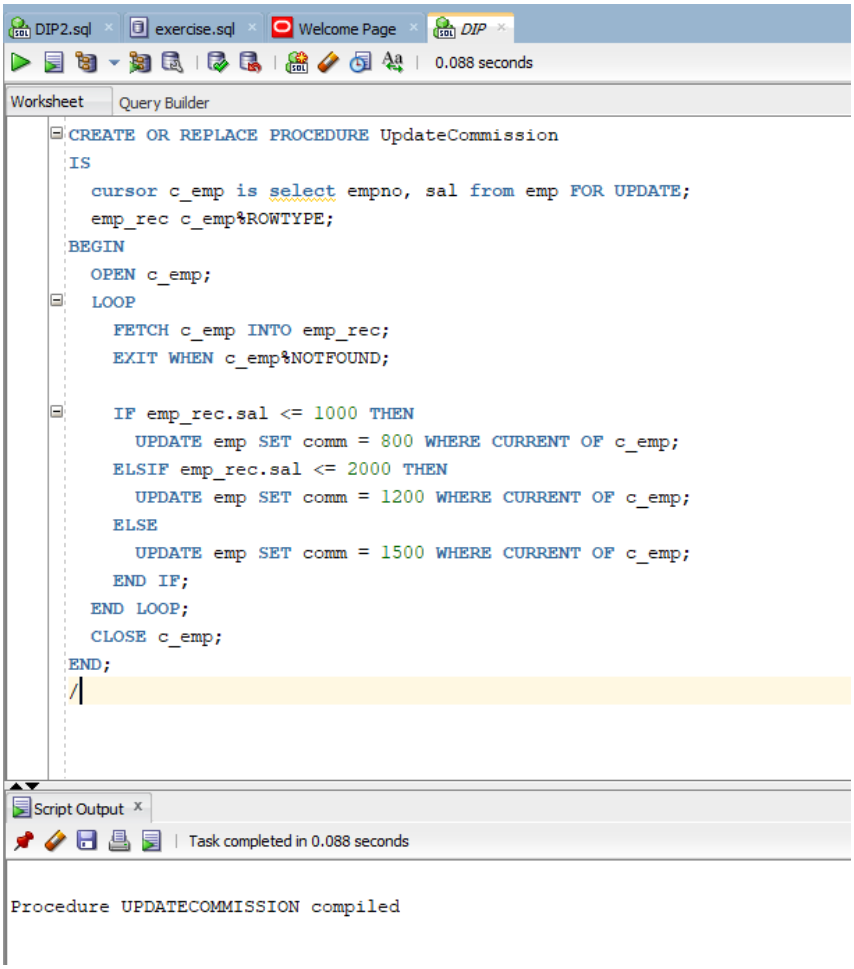



```
PL/SQL procedure successfully completed.
```

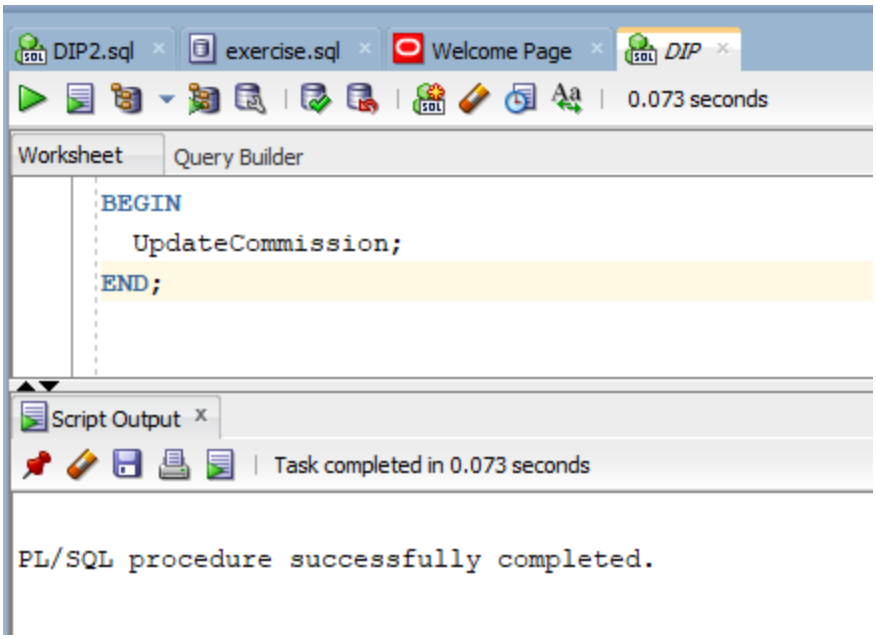
Exercice 5. SELECT for UPDATE

This procedure first opens a cursor that selects the employee's empno and sal columns from the emp table with the FOR UPDATE clause. This puts a lock on the selected rows, ensuring that other transactions cannot modify them at the same time.

Then, it loops through the cursor and checks the employee's salary. Based on their salary, it updates their commission using the UPDATE statement and the WHERE CURRENT OF cursor clause to update the current row being processed by the cursor.



This will update the commission of all employees in the EMP table based on their salary.

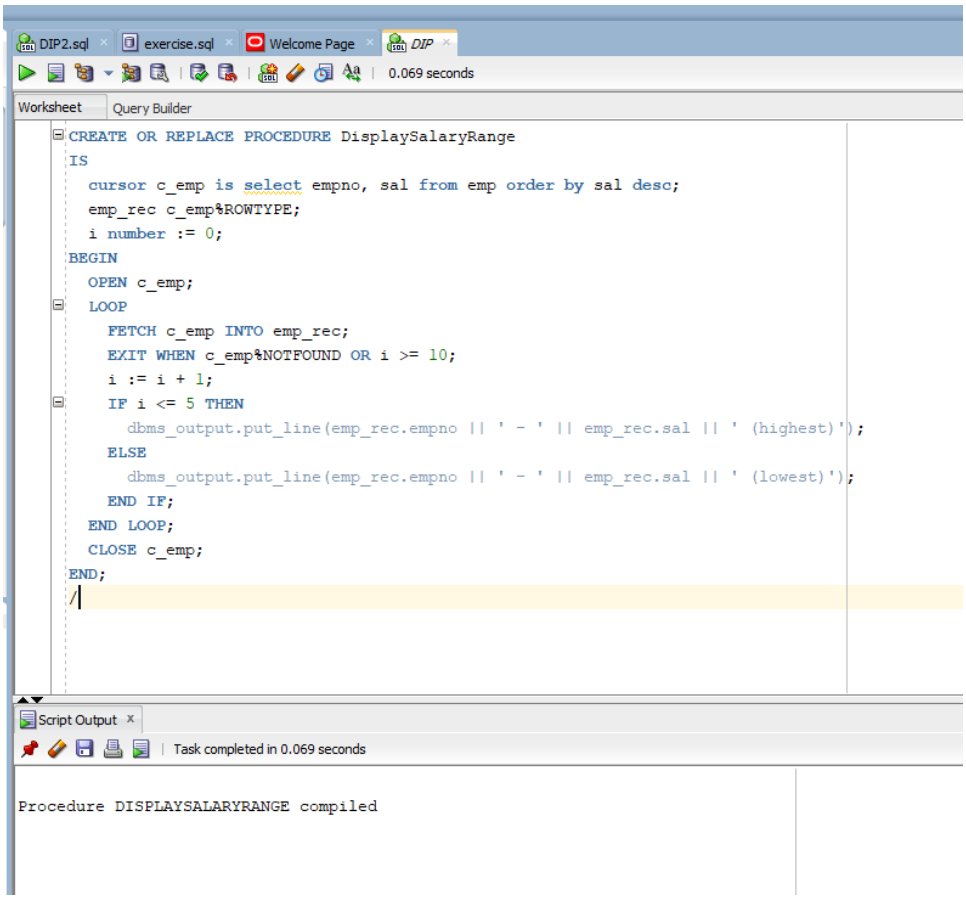


Exercice 6. Condition on EXIT loop

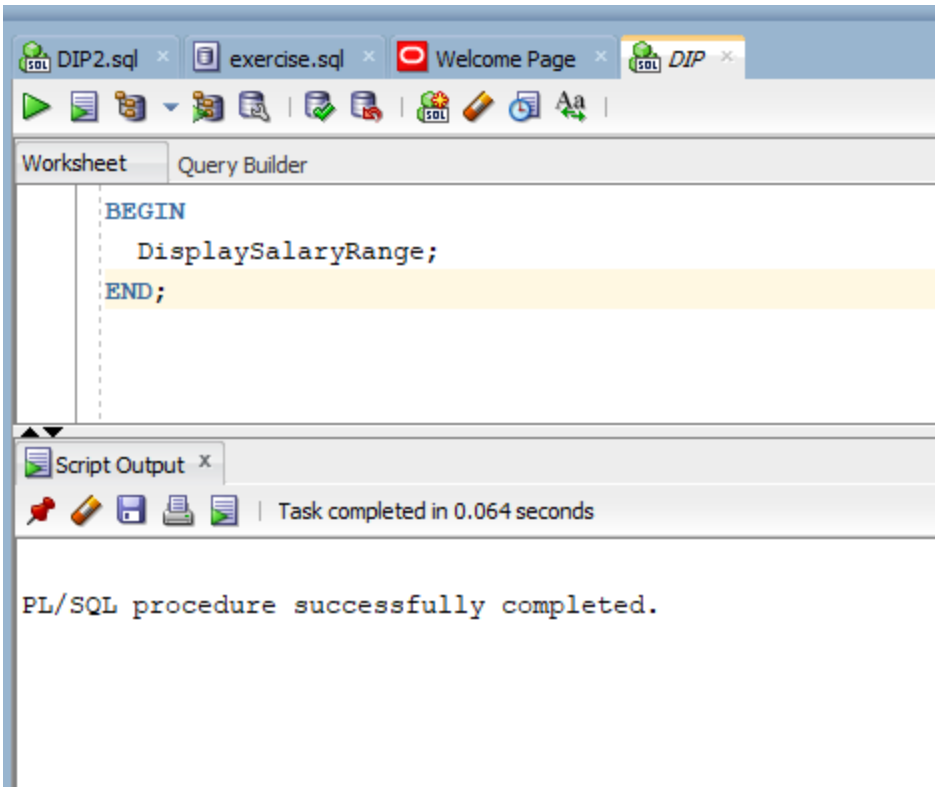
This procedure first opens a cursor that selects the employee's empno and sal columns from the emp table and orders them by salary in descending order.

Then, it loops through the cursor using FETCH to retrieve each row. The loop also checks two conditions: whether the end of the cursor has been reached using the NOTFOUND attribute of the cursor, and whether the loop has already processed the 5 highest and 5 lowest salaries using a counter variable i.

Inside the loop, the procedure checks whether the current employee is one of the 5 highest or 5 lowest salaries, and displays their empno and sal accordingly using the dbms_output.put_line procedure.



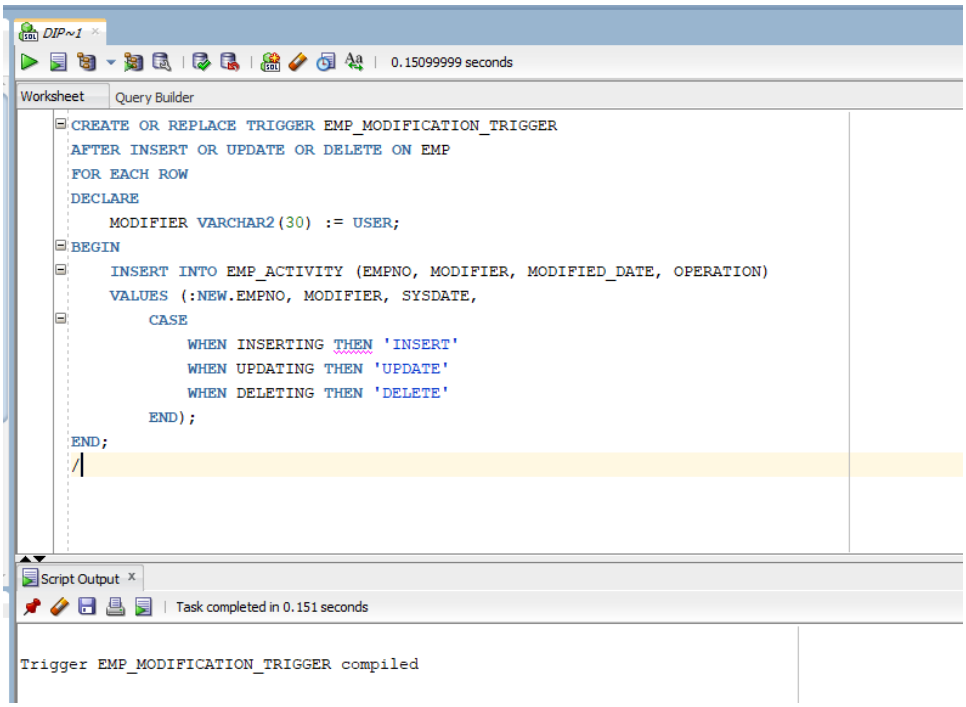
This will display the empno and sal of the 5 employees with the highest salaries, followed by the 5 employees with the lowest salaries.



Exercice 7. Triggers

1. Create The triggers:

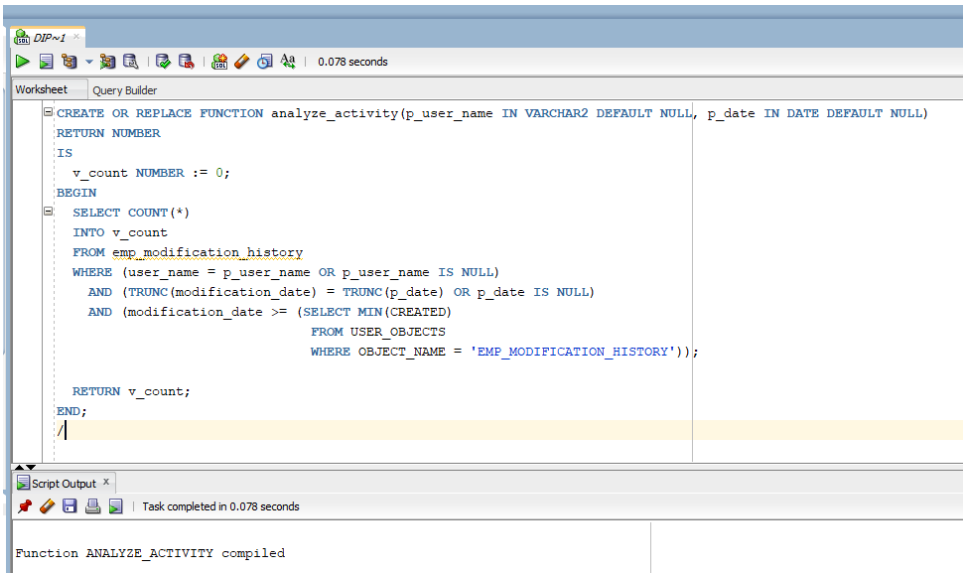
This trigger will insert a new row in the emp_modification_history table every time an insert, update, or delete operation is performed on the emp table. The row will contain the name of the user who performed the operation, the date and time of the operation, and the type of operation.



2. Function to analyze results:

This function takes two optional parameters: p_user_name and p_date. If both parameters are null, it will return the total number of modifications performed on the emp table since it was created. If p_user_name is not null, it will return the total number of modifications performed by that user. If p_date is not null, it will return the total number of modifications performed on that date.

The function uses the emp_modification_history table to count the number of modifications that match the specified criteria. It also uses the USER_OBJECTS view to get the creation date of the emp_modification_history table, so that it can filter out modifications that were performed before the table was created.


DIP~1
0.078 seconds
Worksheet
Query Builder
CREATE OR REPLACE FUNCTION analyze_activity(p_user_name IN VARCHAR2 DEFAULT NULL, p_date IN DATE DEFAULT NULL)
RETURN NUMBER
IS
v_count NUMBER := 0;
BEGIN
SELECT COUNT(*)
INTO v_count
FROM emp_modification_history
WHERE (user_name = p_user_name OR p_user_name IS NULL)
AND (TRUNC(modification_date) = TRUNC(p_date) OR p_date IS NULL)
AND (modification_date >= (SELECT MIN(CREATED)
FROM USER_OBJECTS
WHERE OBJECT_NAME = 'EMP_MODIFICATION_HISTORY'));
RETURN v_count;
END;
/
Script Output
Task completed in 0.078 seconds
Function ANALYZE_ACTIVITY compiled