LAB #07

Name –   Muhammad Taha

Roll NO – 23P-0559

SECTION – BCS(4A)

Subject – Database Systems LABS

1

```sql
SET SERVEROUTPUT ON ;

DECLARE
   e_num EMPLOYEES.EMPLOYEE_ID%TYPE;
   e_sal EMPLOYEES.SALARY%TYPE;
   e_bonus number(8) := 0;
BEGIN
 e_num := '&Enter_Employee_ID';
 select salary into e_sal from EMPLOYEES where employee_id = e_num;

 IF e_sal IS NULL THEN
   e_bonus := 0;
ELSIF e_sal < 1000 THEN
   e_bonus := e_sal * 0.10;
ELSIF e_sal BETWEEN 1000 AND 1500 THEN
   e_bonus := e_sal * 0.15;
ELSIF e_sal > 1500 THEN
   e_bonus := e_sal * 0.20;
END IF;

 DBMS_OUTPUT.PUT_LINE('The Employee ID ' || e_num || ' has a bonus salary of ' || e_bonus);


 END;
```
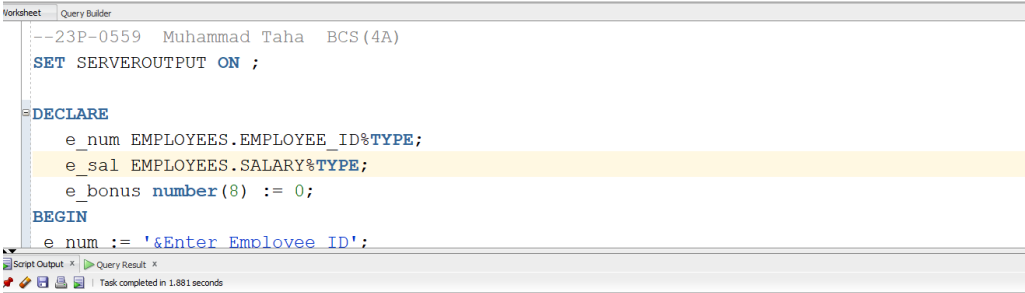
```
--23P-0559  Muhammad Taha  BCS(4A)
SET SERVEROUTPUT ON;

DECLARE
   e_num EMPLOYEES.EMPLOYEE_ID%TYPE;
   e_commission EMPLOYEES.COMMISSION_PCT%TYPE;
   e_salary EMPLOYEES.SALARY%TYPE;
BEGIN
   e_num := '&Enter_Employee_ID';

   BEGIN
      SELECT COMMISSION_PCT, SALARY INTO e_commission, e_salary
      FROM EMPLOYEES
      WHERE EMPLOYEE_ID = e_num;
   EXCEPTION
      WHEN NO_DATA_FOUND THEN
         DBMS_OUTPUT.PUT_LINE('Employee ID not found.');
         RETURN;
   END;

   -- If commission is NULL, update salary
   IF e_commission IS NULL THEN
      UPDATE EMPLOYEES
      SET SALARY = SALARY + 0  -- Adding 0 since commission is NULL
      WHERE EMPLOYEE_ID = e_num;

      DBMS_OUTPUT.PUT_LINE('Salary updated for Employee ID: ' || e_num);
   ELSE
      DBMS_OUTPUT.PUT_LINE('Commission is not NULL, so salary is not updated.');
   END IF;

END;
```
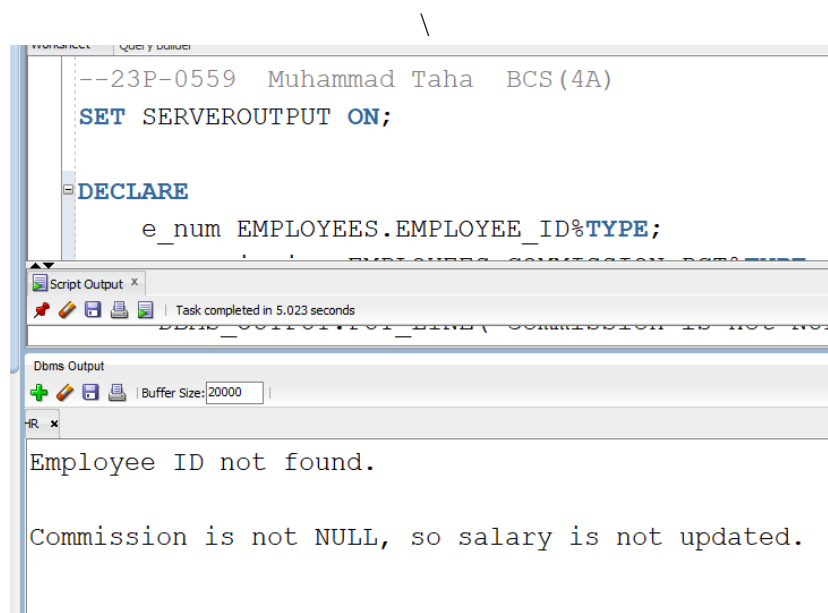
\

```
--23P-0559  Muhammad Taha  BCS(4A)
SET SERVEROUTPUT ON;

DECLARE
 dp_id number := 30;
 dp_name DEPARTMENTS.DEPARTMENT_NAME%TYPE;

BEGIN
select department_name into dp_name from DEPARTMENTS where DEPARTMENT_ID = dp_id;

DBMS_OUTPUT.PUT_LINE('Department Name for Dept No 30: ' || dp_name);
EXCEPTION
   WHEN NO_DATA_FOUND THEN
     DBMS_OUTPUT.PUT_LINE('No department found with Dept No 30.');


END;
```
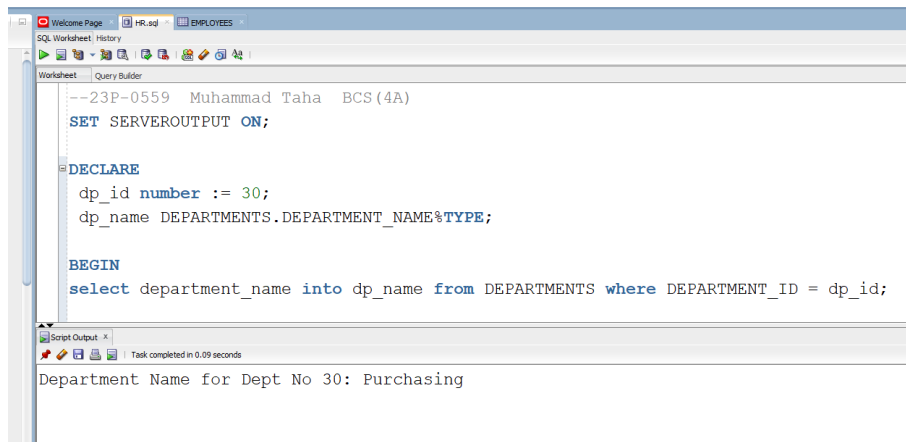
```
--23P-0559  Muhammad Taha  BCS(4A)
SET SERVEROUTPUT ON;

CREATE OR REPLACE FUNCTION get_job_title(p_dept_id NUMBER)
RETURN VARCHAR2 IS
   e_num EMPLOYEES.JOB_ID%TYPE;
BEGIN
   SELECT JOB_ID
   INTO e_num
   FROM EMPLOYEES
   WHERE DEPARTMENT_ID = p_dept_id
   AND ROWNUM = 1; -- Fetching only one job title

   RETURN e_num;
EXCEPTION
   WHEN NO_DATA_FOUND THEN
     RETURN 'No employee found in this department';
```
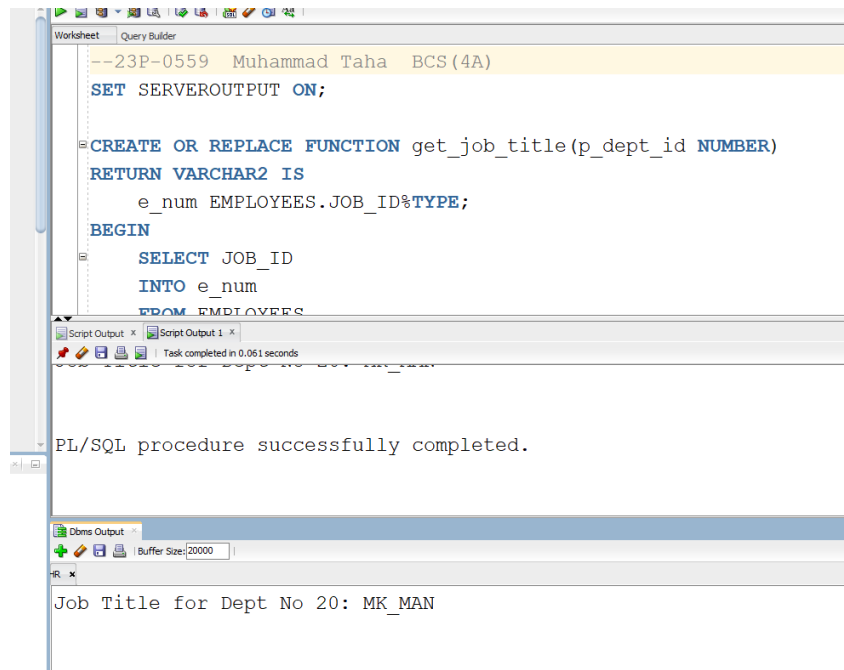
```plsql
END get_job_title;
/

DECLARE
    dept_id NUMBER := 20;
    job_title VARCHAR2(50);
BEGIN
    job_title := get_job_title(dept_id);
    DBMS_OUTPUT.PUT_LINE('Job Title for Dept No ' || dept_id || ': ' || job_title);
END;
/
```


5

--23P-0559 Muhammad Taha BCS(4A)

```plsql
SET SERVEROUTPUT ON;

CREATE OR REPLACE FUNCTION get_emp_salary(dept_num NUMBER)
RETURN NUMBER IS
    emp_salary EMPLOYEES.SALARY%TYPE;
BEGIN
    SELECT SALARY
    INTO emp_salary
    FROM EMPLOYEES
    WHERE DEPARTMENT_ID = dept_num
    AND ROWNUM = 1;

    RETURN emp_salary;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RETURN NULL;
END get_emp_salary;
/
```
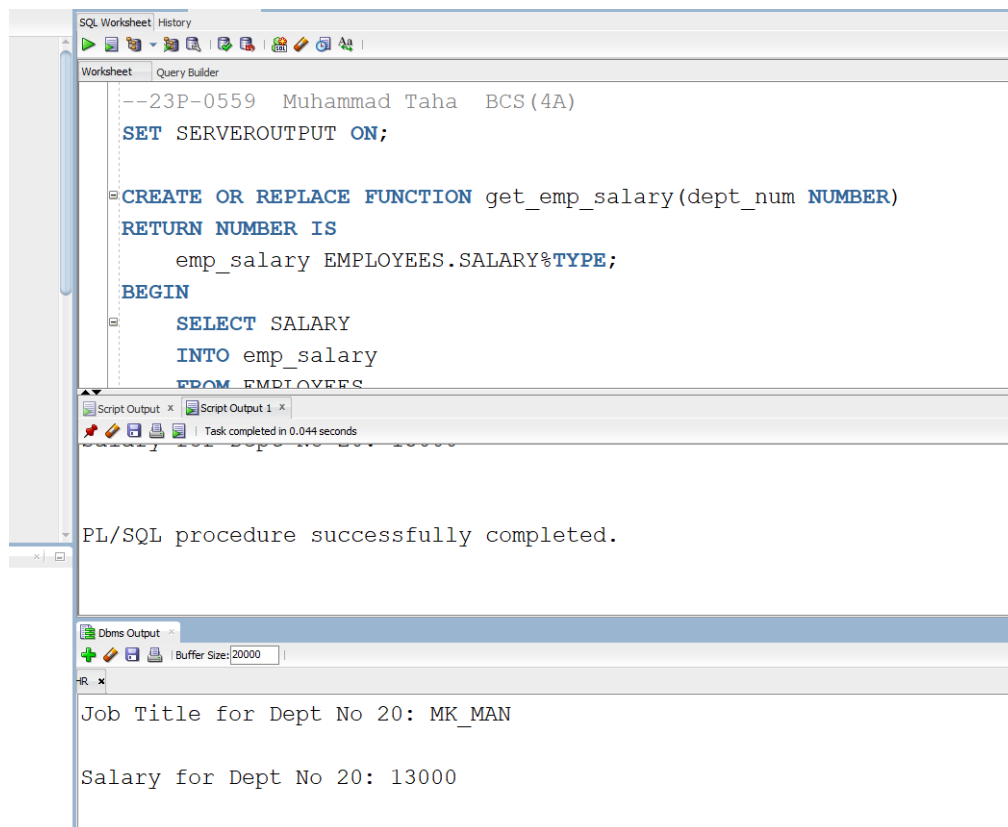
```
DECLARE
    dept_id NUMBER := 20;
    salary NUMBER;
BEGIN
    salary := get_emp_salary(dept_id);

    IF salary IS NULL THEN
        DBMS_OUTPUT.PUT_LINE('No employee found in department ' || dept_id);
    ELSE
        DBMS_OUTPUT.PUT_LINE('Salary for Dept No ' || dept_id || ': ' || salary);
    END IF;
END;
/
```


6

```
--23P-0559  Muhammad Taha  BCS(4A)
SET SERVEROUTPUT ON;

CREATE OR REPLACE PROCEDURE update_salary(emp_id NUMBER) IS
BEGIN
    UPDATE EMPLOYEES
    SET SALARY = SALARY * 1.10
    WHERE EMPLOYEE_ID = emp_id;

    IF SQL%ROWCOUNT = 0 THEN
        DBMS_OUTPUT.PUT_LINE('No employee found with ID ' || emp_id);
    ELSE
        DBMS_OUTPUT.PUT_LINE('Salary updated successfully for Employee ID: ' || emp_id);
    END IF;
```

```
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
END update_salary;
/


DECLARE
    emp_num NUMBER := 101;  -- Change this ID to test with a different employee
BEGIN
    update_salary(emp_num);
END;
/
```

```
--23P-0559  Muhammad Taha  BCS(4A)
SET SERVEROUTPUT ON;

CREATE OR REPLACE PROCEDURE increase_salary(p_dept_id NUMBER) IS
BEGIN
    UPDATE EMPLOYEES
    SET SALARY = SALARY + 1000
    WHERE DEPARTMENT_ID = p_dept_id AND SALARY > 5000;

    IF SQL%ROWCOUNT = 0 THEN
        DBMS_OUTPUT.PUT_LINE('No employees found with salary > 5000 in department ' || p_dept_id);
    ELSE
```
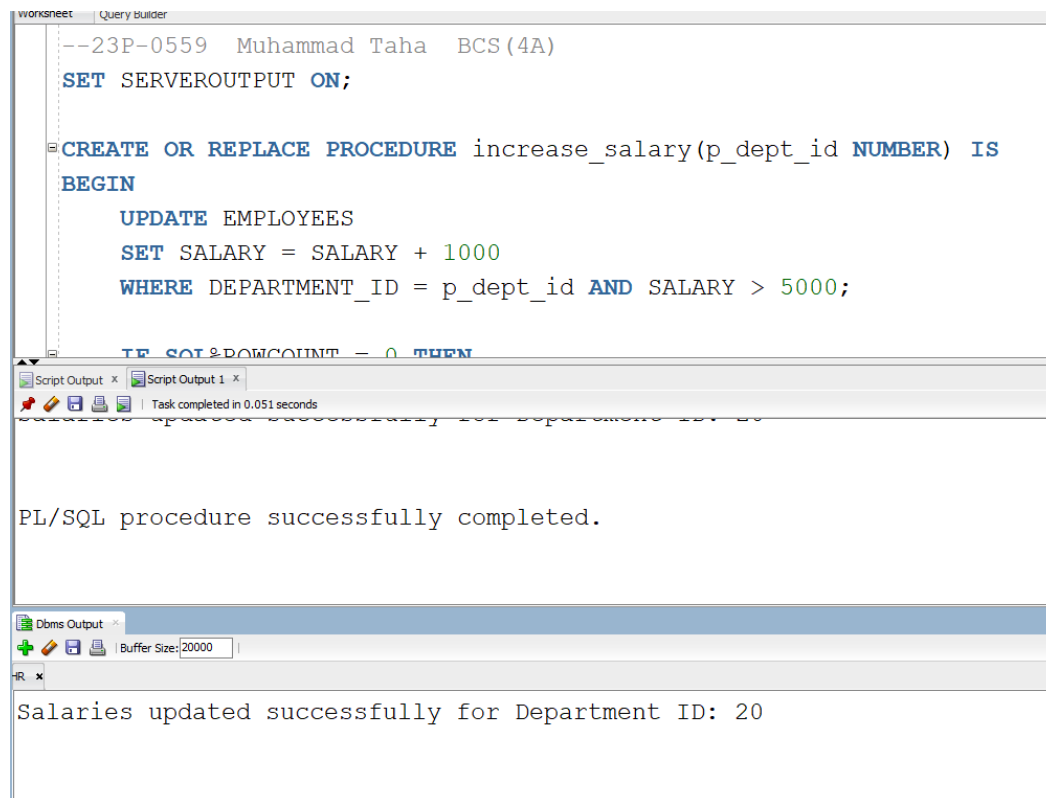
```
    DBMS_OUTPUT.PUT_LINE('Salaries updated successfully for Department ID: ' || p_dept_id);
  END IF;
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
END increase_salary;
/


DECLARE
  dept_num NUMBER := 20;  -- Change this to test with a different department
BEGIN
  increase_salary(dept_num);
END;
/
```


<div align="center">

8

</div>

```
--23P-0559  Muhammad Taha  BCS(4A)
CREATE OR REPLACE VIEW Designation_Count AS
SELECT JOB_ID AS Designation, COUNT(*) AS Employee_Count
FROM EMPLOYEES
GROUP BY JOB_ID;


CREATE OR REPLACE VIEW Employee_Details_Without_King AS
SELECT E.EMPLOYEE_ID, E.First_NAME, E.DEPARTMENT_ID, D.DEPARTMENT_NAME
FROM EMPLOYEES E
JOIN DEPARTMENTS D ON E.DEPARTMENT_ID = D.DEPARTMENT_ID
WHERE E.FIRST_NAME <> 'KING';
```
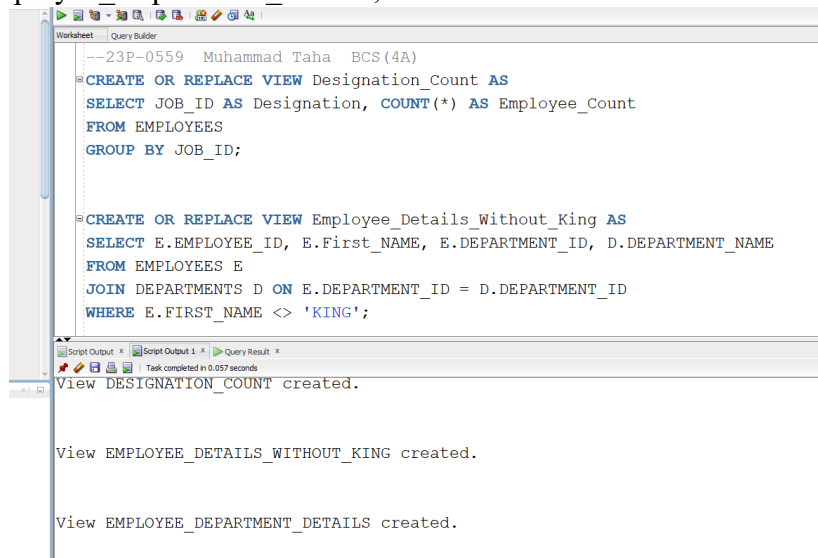
CREATE OR REPLACE VIEW Employee_Department_Details AS
SELECT E.EMPLOYEE_ID, E.First_NAME, E.DEPARTMENT_ID, D.DEPARTMENT_NAME
FROM EMPLOYEES E
JOIN DEPARTMENTS D ON E.DEPARTMENT_ID = D.DEPARTMENT_ID;

SELECT * FROM Designation_Count;
SELECT * FROM Employee_Details_Without_King;
SELECT * FROM Employee_Department_Details;

9

--23P-0559 Muhammad Taha BCS(4A)
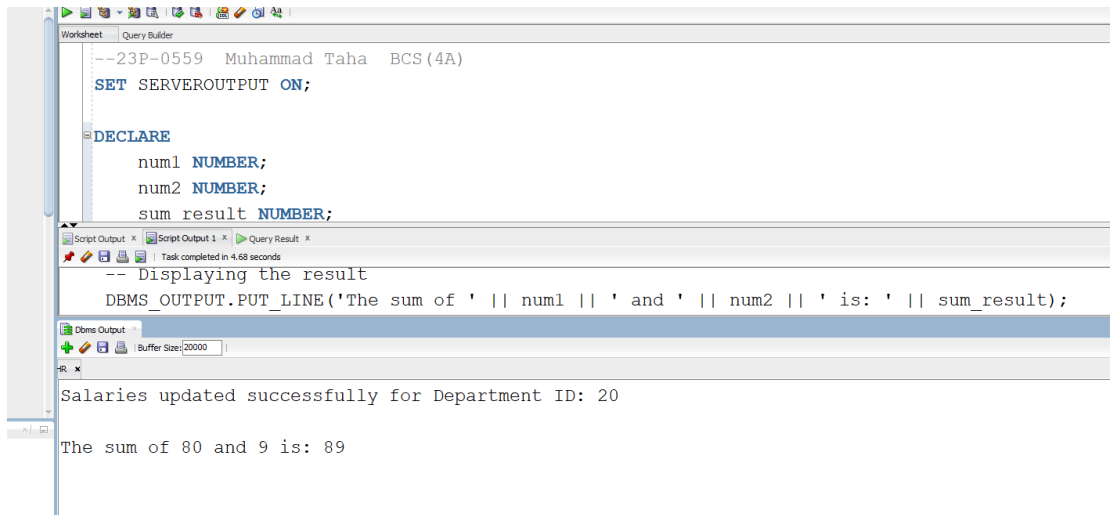SET SERVEROUTPUT ON;

```
DECLARE
    num1 NUMBER;
    num2 NUMBER;
    sum_result NUMBER;
BEGIN
    -- Taking input from the user
    num1 := '&Enter_First_Number';
    num2 := '&Enter_Second_Number';

    -- Adding the numbers
    sum_result := num1 + num2;

    -- Displaying the result
    DBMS_OUTPUT.PUT_LINE('The sum of ' || num1 || ' and ' || num2 || ' is: ' || sum_result);
END;
/
```

10

```
--23P-0559  Muhammad Taha  BCS(4A)
SET SERVEROUTPUT ON;

DECLARE
    lower_bound NUMBER;
    upper_bound NUMBER;
    total_sum NUMBER := 0;
BEGIN
    lower_bound := '&Enter_Lower_Boundary';
    upper_bound := '&Enter_Upper_Boundary';

    FOR i IN lower_bound .. upper_bound LOOP
        total_sum := total_sum + i;
    END LOOP;

    DBMS_OUTPUT.PUT_LINE('Sum of numbers from ' || lower_bound || ' to ' || upper_bound || ' is: ' ||
total_sum);
END;
/
```
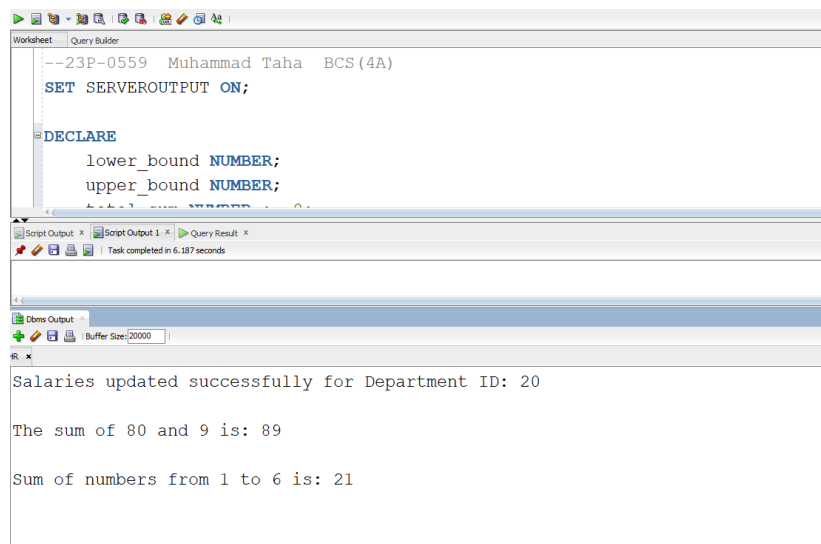
```
--23P-0559  Muhammad Taha  BCS(4A)
SET SERVEROUTPUT ON;

DECLARE
   emp_id EMPLOYEES.EMPLOYEE_ID%TYPE;
   emp_name EMPLOYEES.FIRST_NAME%TYPE;
   hire_date EMPLOYEES.HIRE_DATE%TYPE;
   dept_name DEPARTMENTS.DEPARTMENT_NAME%TYPE;
BEGIN
   emp_id := '&Enter_Employee_ID';

   SELECT e.FIRST_NAME, e.HIRE_DATE, d.DEPARTMENT_NAME
   INTO emp_name, hire_date, dept_name
   FROM EMPLOYEES e
   JOIN DEPARTMENTS d ON e.DEPARTMENT_ID = d.DEPARTMENT_ID
   WHERE e.EMPLOYEE_ID = emp_id;

   DBMS_OUTPUT.PUT_LINE('Employee Name: ' || emp_name);
   DBMS_OUTPUT.PUT_LINE('Hire Date: ' || hire_date);
   DBMS_OUTPUT.PUT_LINE('Department Name: ' || dept_name);
EXCEPTION
   WHEN NO_DATA_FOUND THEN
      DBMS_OUTPUT.PUT_LINE('No employee found with this ID.');
END;
/
```
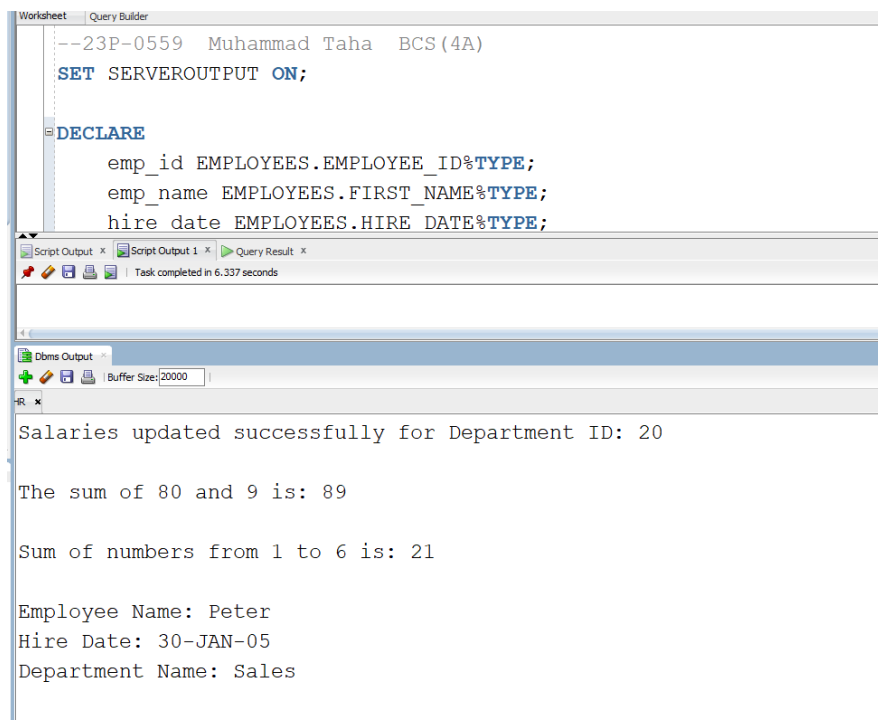
```
--23P-0559  Muhammad Taha  BCS(4A)
SET SERVEROUTPUT ON;

DECLARE
    num NUMBER;
    rev_num NUMBER := 0;
    temp NUMBER;
    digit NUMBER;
BEGIN
    num := '&Enter_Number';
    temp := num;

    WHILE temp > 0 LOOP
        digit := MOD(temp, 10);
        rev_num := (rev_num * 10) + digit;
        temp := TRUNC(temp / 10);
    END LOOP;

    IF num = rev_num THEN
        DBMS_OUTPUT.PUT_LINE(num || ' is a Palindrome.');
    ELSE
        DBMS_OUTPUT.PUT_LINE(num || ' is NOT a Palindrome.');
    END IF;
END;
/
```
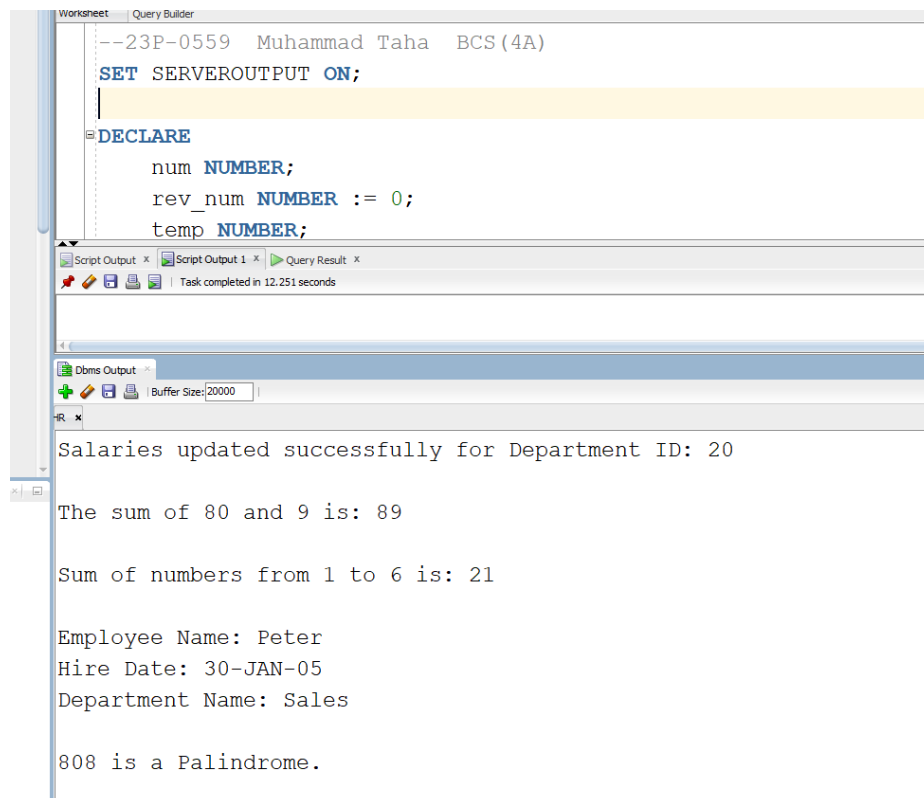
```
--23P-0559  Muhammad Taha  BCS(4A)
select * from EMPLOYEES where EMPLOYEE_ID = 1;
SET SERVEROUTPUT ON;
DECLARE
   emp_id EMPLOYEES.EMPLOYEE_ID%TYPE;
   emp_first_name EMPLOYEES.FIRST_NAME%TYPE;
   emp_last_name EMPLOYEES.LAST_NAME%TYPE;
   emp_email EMPLOYEES.EMAIL%TYPE;
   emp_phone EMPLOYEES.PHONE_NUMBER%TYPE;
   emp_job_id EMPLOYEES.JOB_ID%TYPE;
   emp_salary EMPLOYEES.SALARY%TYPE;
   emp_hire_date EMPLOYEES.HIRE_DATE%TYPE;
   dept_id DEPARTMENTS.DEPARTMENT_ID%TYPE;
   dept_name DEPARTMENTS.DEPARTMENT_NAME%TYPE;
   v_count NUMBER;
BEGIN
   emp_id := '&Enter_Employee_ID';
   emp_first_name := '&Enter_Employee_First_Name';
   emp_last_name := '&Enter_Employee_Last_Name';
   emp_email := '&Enter_Employee_Email';
   emp_phone := '&Enter_Employee_Phone_Number';
   emp_job_id := '&Enter_Employee_Job_ID';  -- Added Job ID input
   emp_salary := '&Enter_Employee_Salary';
   emp_hire_date := SYSDATE;
   dept_id := '&Enter_Department_ID';
   dept_name := '&Enter_Department_Name';

   -- Check if department exists
   SELECT COUNT(*) INTO v_count FROM DEPARTMENTS WHERE DEPARTMENT_ID = dept_id;
   IF v_count = 0 THEN
      INSERT INTO DEPARTMENTS (DEPARTMENT_ID, DEPARTMENT_NAME)
      VALUES (dept_id, dept_name);
   END IF;

   -- Insert into EMPLOYEES table
   INSERT INTO EMPLOYEES (EMPLOYEE_ID, FIRST_NAME, LAST_NAME, EMAIL,
PHONE_NUMBER, JOB_ID, HIRE_DATE, SALARY, DEPARTMENT_ID)
   VALUES (emp_id, emp_first_name, emp_last_name, emp_email, emp_phone, emp_job_id,
emp_hire_date, emp_salary, dept_id);

   COMMIT;
   DBMS_OUTPUT.PUT_LINE('Employee and Department added successfully.');
EXCEPTION
   WHEN DUP_VAL_ON_INDEX THEN
      DBMS_OUTPUT.PUT_LINE('Error: Employee ID already exists.');
   WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE('Error occurred: ' || SQLERRM);
END;
```
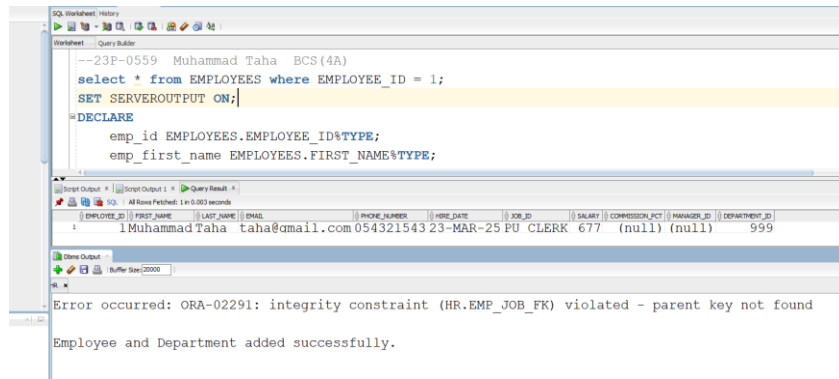
```
--23P-0559  Muhammad Taha  BCS(4A)
select * from EMPLOYEES where EMPLOYEE_ID = 1;
SET SERVEROUTPUT ON;
DECLARE
    emp_id EMPLOYEES.EMPLOYEE_ID%TYPE;
    emp_first_name EMPLOYEES.FIRST_NAME%TYPE;
```

Query Result:
EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
1	Muhammad	Taha	taha@gmail.com	054321543	23-MAR-25	PU CLERK	677	(null)	(null)	999

```
Error occurred: ORA-02291: integrity constraint (HR.EMP_JOB_FK) violated - parent key not found

Employee and Department added successfully.
```

14

```
--23P-0559  Muhammad Taha  BCS(4A)
SET SERVEROUTPUT ON;

DECLARE
    emp_id EMPLOYEES.EMPLOYEE_ID%TYPE := 90;
    emp_salary EMPLOYEES.SALARY%TYPE;
    mgr_id EMPLOYEES.MANAGER_ID%TYPE;
BEGIN
    LOOP
        SELECT SALARY, MANAGER_ID
        INTO emp_salary, mgr_id
        FROM EMPLOYEES
        WHERE EMPLOYEE_ID = emp_id;

        IF emp_salary > 2500 THEN
            DBMS_OUTPUT.PUT_LINE('First higher-level employee with salary > 2500 is Employee ID: ' ||
emp_id);
            EXIT;
        END IF;

        emp_id := mgr_id;
        EXIT WHEN emp_id IS NULL;
    END LOOP;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('No such employee found.');
END;
/
```

15

--23P-0559 Muhammad Taha BCS(4A)
SET SERVEROUTPUT ON;

DECLARE
    total_sum NUMBER := 0;
BEGIN
    FOR i IN 1 .. 100 LOOP
        total_sum := total_sum + i;
    END LOOP;

    DBMS_OUTPUT.PUT_LINE('Sum of first 100 numbers is: ' || total_sum);
END;
/