**Zhumagali Kanat IT2-2003**

**29485**

**Practice 3.1 (Interacting with the Oracle Server)**

1. Create an anonymous block to output the average salary for a particular department.

DECLARE

dep\_id NUMBER := 100;

avg\_salary NUMBER;

BEGIN

SELECT AVG(salary)

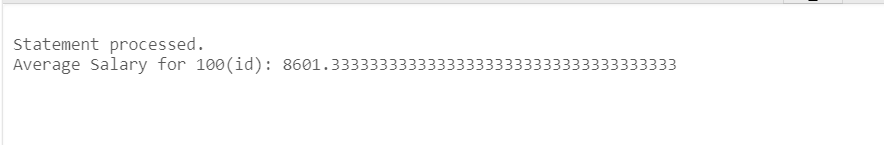
INTO avg\_salary

FROM employees

WHERE department\_id = dep\_id;

DBMS\_OUTPUT.PUT\_LINE('Average Salary for ' ||dep\_id || '(id): ' || avg\_salary);

END;



2. Create an anonymous block to count and output the number of employees of a particular position.

DECLARE

my\_job\_id VARCHAR2(50) := 'IT\_PROG';

employee\_count NUMBER;

BEGIN

SELECT COUNT(\*)

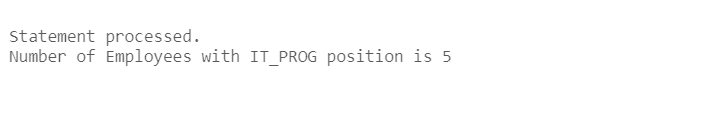
INTO employee\_count

FROM employees

WHERE JOB\_ID = my\_job\_id;

DBMS\_OUTPUT.PUT\_LINE('Number of Employees with ' || my\_job\_id || ' position is ' || employee\_count);

END;



3. Create an anonymous block to count and output the number of letters in the last name and the first name of a particular employee.

DECLARE

emplo\_id NUMBER := 100 ;

my\_first\_name VARCHAR2(50);

my\_last\_name VARCHAR2(50);

first\_l NUMBER;

last\_l NUMBER;

BEGIN

SELECT first\_name, last\_name

INTO my\_first\_name, my\_last\_name

FROM employees

WHERE employee\_id = emplo\_id;

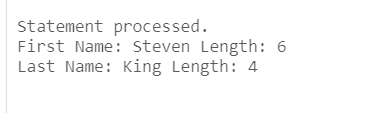
first\_l := LENGTH(my\_first\_name);

last\_l := LENGTH(my\_last\_name);

DBMS\_OUTPUT.PUT\_LINE('First Name: ' || my\_first\_name || ' Length: '|| first\_l );

DBMS\_OUTPUT.PUT\_LINE('Last Name: ' || my\_last\_name || ' Length: ' || last\_l);

END;



4. Create a PL/SQL block that selects the maximum department ID in the departments table and stores it in the max\_deptno variable. Display the maximum department ID.

DECLARE

max\_dep NUMBER;

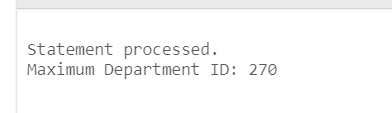
BEGIN

SELECT MAX(department\_id) INTO max\_dep

FROM departments;

DBMS\_OUTPUT.PUT\_LINE('Maximum Department ID: ' || max\_dep);

END;



5. Modify the PL/SQL block you created in exercise 4 and insert a new department with name ‘Education’ into the departments table. Use the SQL attribute SQL%ROWCOUNT to display the number of rows that are affected. Execute a select statement to check if the new department is inserted.

DECLARE

max\_dep NUMBER;

new\_dep VARCHAR2(50) := 'Education';

BEGIN

SELECT MAX(department\_id) INTO max\_dep

FROM departments;

max\_dep := max\_dep + 1;

INSERT INTO departments (department\_id, department\_name)

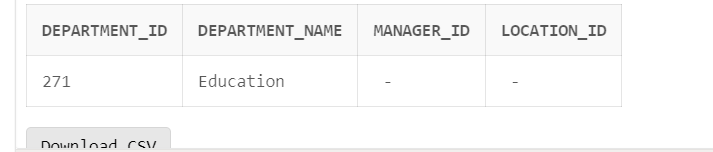
VALUES (max\_dep, new\_dep);

DBMS\_OUTPUT.PUT\_LINE('Affected: ' || SQL%ROWCOUNT);

COMMIT;

END;

SELECT \* FROM departments WHERE department\_name = 'Education';



6. In exercise 5, you have set location\_id to null. Create a PL/SQL block that updates the location\_id to 3000 for the new department. Use a SELECT statement to display the department that you updated. Finally, include a DELETE statement to delete the department that you added.

DECLARE

new\_loc NUMBER := 3000;

BEGIN

UPDATE departments

SET location\_id = new\_loc

WHERE department\_name = 'Education';

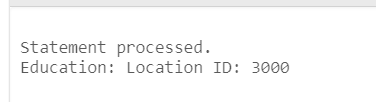
DBMS\_OUTPUT.PUT\_LINE('Education: ' || 'Location ID: ' || new\_loc);

DELETE FROM departments

WHERE department\_name = 'Education';

COMMIT;

END;



7. Create a block with a single SELECT statement, which retrieves the employee\_id of the employee working in the ‘Human Resources’ department.

DECLARE

emp\_id NUMBER;

BEGIN

SELECT e.employee\_id

INTO emp\_id

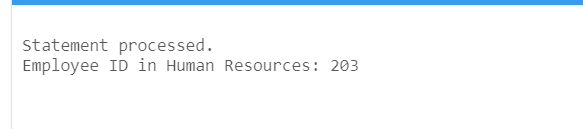
FROM employees e

JOIN departments d ON e.department\_id = d.department\_id

WHERE d.department\_name = 'Human Resources';

DBMS\_OUTPUT.PUT\_LINE('Employee ID in Human Resources: ' || emp\_id);

END;



**Practice 3.2 (Writing Executable Statements)**

1. Write a PL/SQL block to accept a year and check whether it is a leap year. For example, if the year entered is 1990, the output should be “1990 is not a leap year.” Hint: The year should be exactly divisible by 4 but not divisible by 100, or it should be divisible by 400.

Test your solution with the following years:



DECLARE

year NUMBER;

l\_year VARCHAR2(20);

BEGIN

year := 1990;

IF (year mod 4 = 0 AND year mod 100 != 0) OR (year mod 400 = 0) THEN

l\_year := 'a leap year.';

ELSE

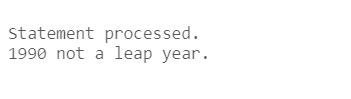
l\_year := 'not a leap year.';

END IF;

DBMS\_OUTPUT.PUT\_LINE(year || ' ' || l\_year);

END;

--1990 2000 1996 1886 1992 1824



2. Write a PL/SQL block to declare a variable called sal to store the salary of an employee. If the salary is less than 3,000, give the employee a raise of 500 and display the message “<Employee Name>’s salary updated” in the window. If the salary is more than 3,000, print the employee’s salary in the format, “<Employee Name> earns …...………”

Test the PL/SQL block for the following last names:



DECLARE

salary NUMBER := 1000;

name VARCHAR2(50) := 'Greenberg';

BEGIN

IF salary < 3000 THEN

salary := salary + 500;

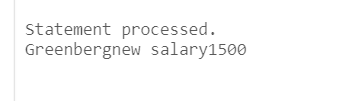
DBMS\_OUTPUT.PUT\_LINE(name || 'new salary ' || salary);

ELSE

DBMS\_OUTPUT.PUT\_LINE(name || ' earns ' || salary);

END IF;

END;

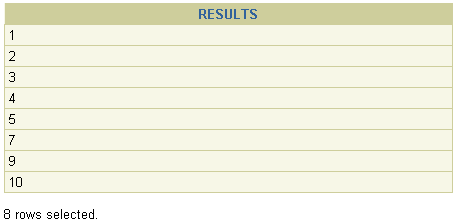


3. Create a table **messages** with one column Results of the type Number (3). Write a PL/SQL block to insert numbers into the messages table.

a. Insert the numbers 1 to 10, excluding 6 and 8.

b. Commit before the end of the block.

c. Execute a SELECT statement to verify that your PL/SQL block worked. You should see the following output. Use different types of loops.



CREATE TABLE messages (

Results NUMBER(3)

);

DECLARE

counter NUMBER := 1;

BEGIN

WHILE counter <= 10 LOOP

IF counter != 6 AND counter != 8 THEN

INSERT INTO messages (Results) VALUES (counter);

END IF;

counter := counter + 1;

END LOOP;

COMMIT;

END;



4. Create a PL/SQL block to identify a name of the month depending on its number. For example, if a user input number 3, the output must be March. Use CASE expression and CASE statement control structures.

DECLARE

number\_m NUMBER := 3;

m\_name VARCHAR2(20);

BEGIN

CASE number\_m

WHEN 1 THEN m\_name := 'January';

WHEN 2 THEN m\_name := 'February';

WHEN 3 THEN m\_name := 'March';

WHEN 4 THEN m\_name := 'April';

WHEN 5 THEN m\_name := 'May';

WHEN 6 THEN m\_name := 'June';

WHEN 7 THEN m\_name := 'July';

WHEN 8 THEN m\_name := 'August';

WHEN 9 THEN m\_name := 'September';

WHEN 10 THEN m\_name := 'October';

WHEN 11 THEN m\_name := 'November';

WHEN 12 THEN m\_name := 'December';

ELSE m\_name := 'Invalid';

END CASE;

DBMS\_OUTPUT.PUT\_LINE(m\_name);

END;

