ORACLE LAB ASSIGNMENT – 9

Cursor Programs

1. Write a PL/SQL program that increase salary of each employee to make the salary equivalent to average of all employee's salary.

- Employee table

```
SQL> select *from employees;

    EMP_ID     SALARY
---------- ----------
         1      50000
         2      30000
         3      10000
```

- PL/SQL block

```
SQL>
SQL> set serveroutput on
SQL> DECLARE
  2      CURSOR emp_cursor IS
  3          SELECT emp_id, salary FROM employees;
  4
  5      avg_salary NUMBER;
  6      current_emp_id employees.emp_id%TYPE;
  7      current_salary employees.salary%TYPE;
  8
  9  BEGIN
 10      SELECT AVG(salary) INTO avg_salary FROM employees;
 11      OPEN emp_cursor;
 12      LOOP
 13          FETCH emp_cursor INTO current_emp_id, current_salary;
 14
 15          EXIT WHEN emp_cursor%NOTFOUND;
 16
 17     UPDATE employees
 18          SET salary = avg_salary
 19          WHERE emp_id = current_emp_id;
 20
 21      END LOOP;
 22
 23      CLOSE emp_cursor;
 24
 25      COMMIT;
 26
 27
 28      DBMS_OUTPUT.PUT_LINE('Salaries updated to the average salary of: ' || avg_salary);
 29  END;
 30  /
Salaries updated to the average salary of: 30000

PL/SQL procedure successfully completed.
```

- PL/SQL OUPUT

```
SQL> select *from employees;

    EMP_ID     SALARY
---------- ----------
         1      30000
         2      30000
         3      30000
```

2. Write a PL/SQL program that insert record in Product_Master_New table with product and its sales information for each product which is sold or whose description is ‘1.44 floppies’, ‘monitor’ or ‘mouse’

- Product master table

```
SQL> select *from  Product_Master;

PRODUCT_ID DESCRIPTION                                        SALES_INFO
---------- -------------------------------------------------- ----------
         1 1.44 floppies                                             100
         2 keyboard                                                  150
         3 monitor                                                   200
         4 mouse                                                     250
```

- PL/SQL BLOCK

```
SQL> SET SERVEROUTPUT ON
SQL> DECLARE
  2      CURSOR product_cursor IS
  3          SELECT product_id, description, sales_info
  4          FROM Product_Master
  5          WHERE description IN ('1.44 floppies', 'monitor', 'mouse');
  6
  7      current_product_id Product_Master.product_id%TYPE;
  8      current_description Product_Master.description%TYPE;
  9      current_sales_info Product_Master.sales_info%TYPE;
 10
 11  BEGIN
 12      OPEN product_cursor;
 13
 14      LOOP
 15          FETCH product_cursor INTO current_product_id, current_description, current_sales_info;
 16
 17          EXIT WHEN product_cursor%NOTFOUND;
 18
 19          INSERT INTO Product_Master_New
 20          VALUES (current_product_id, current_description, current_sales_info);
 21
 22      END LOOP;
 23
 24      CLOSE product_cursor;
 25
 26      COMMIT;
 27
 28      DBMS_OUTPUT.PUT_LINE('Records successfully inserted into Product_Master_New table.');
 29  END;
 30  /
Records successfully inserted into Product_Master_New table.

PL/SQL procedure successfully completed.
```

- PL/SQL OUPUT

```
SQL>
SQL> SELECT *FROM  Product_Master_New;

PRODUCT_ID DESCRIPTION                                        SALES_INFO
---------- -------------------------------------------------- ----------
         1 1.44 floppies                                             100
         3 monitor                                                   200
         4 mouse                                                     250
```

3. Write a PL/SQL program that will give increment of 10% to all employees. If salary increased then display massage as records are updated otherwise display appropriate message.

- EMPLOYEES TABLE

```
SQL> SELECT *FROM  EMPLOYEES;

    EMP_ID NAME                                               SALARY
---------- -------------------------------------------------- ----------
         1 AJAY                                                    20000
         2 VIJAY                                                   40000
         3 JAY                                                     60000
```

- PL/SQL BLOCK

```
SQL> SET SERVEROUTPUT ON
SQL> DECLARE
  2
  3            CURSOR emp_cursor IS
  4         SELECT emp_id, salary FROM employees;
  5
  6            current_emp_id employees.emp_id%TYPE;
  7            current_salary employees.salary%TYPE;
  8
  9     records_updated BOOLEAN := FALSE;
 10
 11  BEGIN
 12
 13            OPEN emp_cursor;
 14
 15
 16            LOOP
 17         FETCH emp_cursor INTO current_emp_id, current_salary;
 18
 19         EXIT WHEN emp_cursor%NOTFOUND;
 20
 21         UPDATE employees
 22         SET salary = salary * 1.10
 23         WHERE emp_id = current_emp_id;
 24
 25         IF SQL%ROWCOUNT > 0 THEN
 26         records_updated := TRUE;
 27         END IF;
 28
 29      END LOOP;
 30
 31      CLOSE emp_cursor;
 32
 33     COMMIT;
 34
 35      IF records_updated THEN
 36          DBMS_OUTPUT.PUT_LINE('Records are updated. All employees received a 10% increment.');
 37      ELSE
 38          DBMS_OUTPUT.PUT_LINE('No records were updated.');
 39      END IF;
 40
 41  END;
 42  /
Records are updated. All employees received a 10% increment.

PL/SQL procedure successfully completed.
```

- PL/SQL OUTPUT

```
SQL> SELECT *FROM  EMPLOYEES;

    EMP_ID NAME                                               SALARY
---------- -------------------------------------------------- ----------
         1 AJAY                                                22000
         2 VIJAY                                               44000
         3 JAY                                                 66000

SQL>
```

PROCEDURE

1. Write a procedure that will accept employee number and display employee details for that number.

- EMPLOYEES TABLE

```
SQL> select *from employees;

    EMP_ID NAME                                               SALARY DEPARTMENT
---------- -------------------------------------------------- ---------- --------------------------------------------------
         1 ajay                                                50000 HR
         2 vijay                                               60000 Finance
         3 jay                                                 55000 IT
```

- PL/SQL BLOCK

```
SQL>
SQL> SET SERVEROUTPUT ON
SQL> CREATE OR REPLACE PROCEDURE get_employee_details (p_emp_id IN NUMBER) IS
  2      v_name employees.name%TYPE;
  3      v_salary employees.salary%TYPE;
  4      v_department employees.department%TYPE;
  5  BEGIN
  6      SELECT name, salary, department
  7      INTO v_name, v_salary, v_department
  8      FROM employees
  9      WHERE emp_id = p_emp_id;
 10
 11      DBMS_OUTPUT.PUT_LINE('Employee Details:');
 12      DBMS_OUTPUT.PUT_LINE('Name: ' || v_name);
 13      DBMS_OUTPUT.PUT_LINE('Salary: ' || v_salary);
 14      DBMS_OUTPUT.PUT_LINE('Department: ' || v_department);
 15
 16  END;
 17  /

Procedure created.
```

- RUN PROCEDURE

```
SQL> BEGIN
  2      get_employee_details(1);
  3  END;
  4  /
Employee Details:
Name: ajay
Salary: 50000
Department: HR

PL/SQL procedure successfully completed.
```

2. Write a procedure that will find out the minimum, maximum, average and sum of salaries from employee table. If total salary is more then 1, 00,000 then do not give any increment in salary. If total salary is >50,000 but <=50,000 then give increment of 10%.

- EMPLOYEES TABLE

```
SQL> select *from employees;

    EMP_ID NAME                                               SALARY
---------- -------------------------------------------------- ----------
         1 jay                                                 30000
         2 ajay                                                25000
         3 vijay                                               35000
```

- PL/SQL BLOCK

```
SQL> SET SERVEROUTPUT ON
SQL> CREATE OR REPLACE PROCEDURE manage_salaries IS
  2      v_min_salary employees.salary%TYPE;
  3      v_max_salary employees.salary%TYPE;
  4      v_avg_salary employees.salary%TYPE;
  5      v_sum_salary employees.salary%TYPE;
  6
  7  BEGIN
  8      SELECT MIN(salary), MAX(salary), AVG(salary), SUM(salary)
  9      INTO v_min_salary, v_max_salary, v_avg_salary, v_sum_salary
 10      FROM employees;
 11
 12      DBMS_OUTPUT.PUT_LINE('Minimum Salary: ' || v_min_salary);
 13      DBMS_OUTPUT.PUT_LINE('Maximum Salary: ' || v_max_salary);
 14      DBMS_OUTPUT.PUT_LINE('Average Salary: ' || v_avg_salary);
 15      DBMS_OUTPUT.PUT_LINE('Total Salary: ' || v_sum_salary);
 16
 17      IF v_sum_salary > 100000 THEN
 18          DBMS_OUTPUT.PUT_LINE('No increment applied.');
 19
 20      ELSIF v_sum_salary > 50000 AND v_sum_salary <= 100000 THEN
 21          DBMS_OUTPUT.PUT_LINE('Applying 10% increment...');
 22
 23          UPDATE employees
 24          SET salary = salary * 1.10;
 25
 26          COMMIT;
 27          DBMS_OUTPUT.PUT_LINE('Salaries updated with 10% increment.');
 28      ELSE
 29          DBMS_OUTPUT.PUT_LINE('No specific action defined.');
 30      END IF;
 31
 32  END;
 33  /

Procedure created.
```

- RUN PROCEDURE

```
SQL> BEGIN
  2      manage_salaries;
  3  END;
  4  /
Minimum Salary: 25000
Maximum Salary: 35000
Average Salary: 30000
Total Salary: 90000
Applying 10% increment...
Salaries updated with 10% increment.
```

- PL/SQL OUTPUT

```
SQL> select *from employees;

    EMP_ID NAME                                               SALARY
---------- -------------------------------------------------- ----------
         1 jay                                                 33000
         2 ajay                                                27500
         3 vijay                                               38500
```

3. Write a procedure that will display all employee details.

- EMPLOYEES TABLE

```
SQL>
SQL> select *from employees;

    EMP_ID NAME                                               SALARY DEPARTMENT
---------- -------------------------------------------------- ---------- --------------------------------------------------
         1 jay                                                 50000 HR
         2 ajay                                                60000 Finance
         3 vijay                                               55000 IT
```

- PL/SQL BLOCK

```
SQL> SET SERVEROUTPUT ON
SQL> CREATE OR REPLACE PROCEDURE display_all_employees IS
  2
  3      v_emp_id employees.emp_id%TYPE;
  4      v_name employees.name%TYPE;
  5      v_salary employees.salary%TYPE;
  6      v_department employees.department%TYPE;
  7
  8      CURSOR emp_cursor IS
  9          SELECT emp_id, name, salary, department
 10          FROM employees;
 11
 12  BEGIN
 13
 14      OPEN emp_cursor;
 15
 16      LOOP
 17          FETCH emp_cursor INTO v_emp_id, v_name, v_salary, v_department;
 18
 19          EXIT WHEN emp_cursor%NOTFOUND;
 20
 21          DBMS_OUTPUT.PUT_LINE('Employee ID: ' || v_emp_id);
 22          DBMS_OUTPUT.PUT_LINE('Name: ' || v_name);
 23          DBMS_OUTPUT.PUT_LINE('Salary: ' || v_salary);
 24          DBMS_OUTPUT.PUT_LINE('Department: ' || v_department);
 25          DBMS_OUTPUT.PUT_LINE('--------------------------');
 26      END LOOP;
 27
 28      CLOSE emp_cursor;
 29
 30  END;
 31  /

Procedure created.
```

- RUN PROCEDURE

```
SQL> BEGIN
  2      display_all_employees;
  3  END;
  4  /
Employee ID: 1
Name: jay
Salary: 50000
Department: HR
-------------------------
Employee ID: 2
Name: ajay
Salary: 60000
Department: Finance
-------------------------
Employee ID: 3
Name: vijay
Salary: 55000
Department: IT
-------------------------

PL/SQL procedure successfully completed.
```

4. Write a PL/SQL program that will accept Product_No as input and insert product details with their profit & loss in Product_Profit_Loss table.

- PRODUCTS TABLE

```
SQL> select *from products;

PRODUCT_NO PRODUCT_NAME                                                                                          COST_PRICE SELLING_PRICE
---------- ---------------------------------------------------------------------------------------------------- ---------- -------------
         1 Laptop                                                                                                    50000         55000
         2 Monitor                                                                                                   10000          9500
         3 Mouse                                                                                                       500           600
```

- PL/SQL BLOCK

```
SQL> SET SERVEROUTPUT ON
SQL> DECLARE
  2      v_product_no products.product_no%TYPE;
  3      v_product_name products.product_name%TYPE;
  4      v_cost_price products.cost_price%TYPE;
  5      v_selling_price products.selling_price%TYPE;
  6      v_profit_or_loss NUMBER;
  7      v_status VARCHAR2(10);
  8
  9  BEGIN
 10      v_product_no := &Product_No;
 11
 12      SELECT product_name, cost_price, selling_price
 13      INTO v_product_name, v_cost_price, v_selling_price
 14      FROM products
 15      WHERE product_no = v_product_no;
 16
 17      v_profit_or_loss := v_selling_price - v_cost_price;
 18
 19      IF v_profit_or_loss > 0 THEN
 20          v_status := 'Profit';
 21      ELSE
 22          v_status := 'Loss';
 23      END IF;
 24
 25      INSERT INTO Product_Profit_Loss
 26      VALUES (v_product_no, v_product_name, v_cost_price, v_selling_price, v_profit_or_loss, v_status);
 27
 28      COMMIT;
 29
 30      DBMS_OUTPUT.PUT_LINE('Product details with Profit/Loss have been inserted successfully.');
 31
 32  END;
 33  /
Enter value for product_no: 1
old  10:      v_product_no := &Product_No;
new  10:      v_product_no := 1;
Product details with Profit/Loss have been inserted successfully.

PL/SQL procedure successfully completed.
```

- PL/SQL OUTPUT

```
SQL> SELECT * FROM Product_Profit_Loss;

PRODUCT_NO PRODUCT_NAME                                                                                          COST_PRICE SELLING_PRICE PROFIT_OR_LOSS STATUS
---------- ---------------------------------------------------------------------------------------------------- ---------- ------------- --------------- ----------
         1 Laptop                                                                                                    50000         55000            5000 Profit
```