

Лабораторная работа №12. Работа с пакетами.

Задание:

Создать 2 пакета:

Первый пакет – обработка данных для отделов:

- 1) Создание нового отдела
- 2) Подсчет количества сотрудников в указанном отделе
- 3) Подсчет количества сотрудников в каждом отделе
- 4) Изменение названия указанного отдела на новое
- 5) Перевод указанного отдела в новый город.

Второй пакет – обработка данных для сотрудников фирмы:

- 1) Зачисление нового сотрудника
- 2) Изменение зарплаты указанного сотрудника на заданную величину
- 3) Расчет ежегодного объема зарплаты для указанного сотрудника
- 4) Расчет ежегодного объема зарплаты сотрудников в указанном отделе.
- 5) Перевод сотрудника в указанный отдел.

Ход работы:

Первый пакет:

Спецификация:

```
CREATE OR REPLACE PACKAGE dept_package AS
```

```
    PROCEDURE add_dept(p_name IN DEPT.d_name%TYPE, p_location IN  
        DEPT.location%TYPE);
```

```
    FUNCTION count_employees_in(p_name IN DEPT.d_name%TYPE)  
        RETURN NUMBER;
```

```
    FUNCTION count_employees_all RETURN SYS_REFCURSOR;
```

```
PROCEDURE rename_dept(p_name IN DEPT.d_name%TYPE, p_new_name  
    IN DEPT.d_name%TYPE);
```

```
PROCEDURE move_dept(p_name IN DEPT.d_name%TYPE, p_new_city IN  
    DEPT.location%TYPE);
```

```
END dept_package;
```

Тело пакета:

```
CREATE OR REPLACE PACKAGE BODY dept_package AS
```

```
    PROCEDURE add_dept(p_name IN DEPT.d_name%TYPE, p_location IN  
        DEPT.location%TYPE) IS
```

```
        v_count NUMBER;
```

```
        v_new_id DEPT.d_no%TYPE;
```

```
BEGIN
```

```
    SELECT COUNT(*) INTO v_count FROM DEPT WHERE d_name =  
p_name;
```

```
    IF v_count > 0 THEN
```

```
        RAISE_APPLICATION_ERROR(-20001, 'Отдел уже существует');
```

```
    END IF;
```

```
    SELECT NVL(MAX(d_no), 0) + 1 INTO v_new_id FROM DEPT;
```

```
    INSERT INTO DEPT(d_no, d_name, location) VALUES (v_new_id,  
p_name, p_location);
```

```
END;
```

```
FUNCTION count_employees_in(p_name IN DEPT.d_name%TYPE)  
RETURN NUMBER IS
```

```
    v_count NUMBER;
```

```

v_dno DEPT.d_no%TYPE;

BEGIN

SELECT d_no INTO v_dno FROM DEPT WHERE d_name = p_name;

SELECT COUNT(*) INTO v_count FROM EMP WHERE d_no = v_dno;

RETURN v_count;

EXCEPTION

WHEN NO_DATA_FOUND THEN

    RAISE_APPLICATION_ERROR(-20002, 'Такой отдел не найден');

END;

```

```

FUNCTION count_employees_all RETURN SYS_REFCURSOR IS

c SYS_REFCURSOR;

BEGIN

OPEN c FOR

    SELECT D.d_name, COUNT(E.e_no) AS emp_count

    FROM DEPT D LEFT JOIN EMP E ON D.d_no = E.d_no

    GROUP BY D.d_name;

RETURN c;

END;

```

```

PROCEDURE rename_dept(p_name IN DEPT.d_name%TYPE, p_new_name
IN DEPT.d_name%TYPE) IS

v_count NUMBER;

```

```

BEGIN

    SELECT COUNT(*) INTO v_count FROM DEPT WHERE d_name =
p_name;

    IF v_count = 0 THEN

        RAISE_APPLICATION_ERROR(-20003, 'Отдел не найден');

    END IF;

    UPDATE DEPT SET d_name = p_new_name WHERE d_name = p_name;

END;

PROCEDURE move_dept(p_name IN DEPT.d_name%TYPE, p_new_city IN
DEPT.location%TYPE) IS

    v_count NUMBER;

BEGIN

    SELECT COUNT(*) INTO v_count FROM DEPT WHERE d_name =
p_name;

    IF v_count = 0 THEN

        RAISE_APPLICATION_ERROR(-20004, 'Отдел не найден');

    END IF;

    UPDATE DEPT SET location = p_new_city WHERE d_name = p_name;

END;

END dept_package;

```

Второе задание:

Спецификация:

```

CREATE OR REPLACE PACKAGE emp_package AS

```

```
PROCEDURE add_employee(p_name IN EMP.e_name%TYPE, p_date IN  
EMP.date_hire%TYPE, p_salary IN EMP.salary%TYPE, p_address IN  
EMP.address%TYPE, p_dept IN DEPT.d_name%TYPE);
```

```
PROCEDURE change_salary(p_name IN EMP.e_name%TYPE, p_amount IN  
NUMBER);
```

```
FUNCTION yearly_salary(p_name IN EMP.e_name%TYPE) RETURN  
NUMBER;
```

```
FUNCTION yearly_salary_dept(p_dept IN DEPT.d_name%TYPE) RETURN  
NUMBER;
```

```
PROCEDURE move_employee(p_name IN EMP.e_name%TYPE,  
p_new_dept IN DEPT.d_name%TYPE);
```

```
END emp_package;
```

Тело пакета:

```
CREATE OR REPLACE PACKAGE BODY emp_package AS
```

```
PROCEDURE add_employee(p_name IN EMP.e_name%TYPE, p_date IN  
EMP.date_hire%TYPE, p_salary IN EMP.salary%TYPE, p_address IN  
EMP.address%TYPE, p_dept IN DEPT.d_name%TYPE) IS
```

```
    v_dept_no DEPT.d_no%TYPE;
```

```
    v_new_id EMP.e_no%TYPE;
```

```
BEGIN
```

```
    SELECT d_no INTO v_dept_no FROM DEPT WHERE d_name = p_dept;
```

```
    SELECT NVL(MAX(e_no), 100000) + 1 INTO v_new_id FROM EMP;
```

```
    INSERT INTO EMP(e_no, e_name, date_hire, salary, address, d_no)
```

```
    VALUES (v_new_id, p_name, p_date, p_salary, p_address, v_dept_no);
```

```
EXCEPTION
```

```
    WHEN NO_DATA_FOUND THEN
```

```
        RAISE_APPLICATION_ERROR(-20005, 'Отдел не найден');

END;


PROCEDURE change_salary(p_name IN EMP.e_name%TYPE, p_amount IN
NUMBER) IS

BEGIN

    UPDATE EMP SET salary = salary + p_amount WHERE e_name = p_name;

    IF SQL%ROWCOUNT = 0 THEN

        RAISE_APPLICATION_ERROR(-20006, 'Сотрудник не найден');

    END IF;

END;
```

```
FUNCTION yearly_salary(p_name IN EMP.e_name%TYPE) RETURN
NUMBER IS

    v_salary NUMBER;

BEGIN

    SELECT salary * 12 INTO v_salary FROM EMP WHERE e_name =
p_name;

    RETURN v_salary;

EXCEPTION

    WHEN NO_DATA_FOUND THEN

        RAISE_APPLICATION_ERROR(-20007, 'Сотрудник не найден');

END;
```

FUNCTION yearly_salary_dept(p_dept IN DEPT.d_name%TYPE) RETURN
NUMBER IS

 v_total NUMBER;

 v_dno DEPT.d_no%TYPE;

BEGIN

 SELECT d_no INTO v_dno FROM DEPT WHERE d_name = p_dept;

 SELECT NVL(SUM(salary), 0) * 12 INTO v_total FROM EMP WHERE
d_no = v_dno;

 RETURN v_total;

EXCEPTION

 WHEN NO_DATA_FOUND THEN

 RAISE_APPLICATION_ERROR(-20008, 'Отдел не найден');

END;

PROCEDURE move_employee(p_name IN EMP.e_name%TYPE,
p_new_dept IN DEPT.d_name%TYPE) IS

 v_new_dno DEPT.d_no%TYPE;

BEGIN

 SELECT d_no INTO v_new_dno FROM DEPT WHERE d_name =
p_new_dept;

 UPDATE EMP SET d_no = v_new_dno WHERE e_name = p_name;

 IF SQL%ROWCOUNT = 0 THEN

 RAISE_APPLICATION_ERROR(-20009, 'Сотрудник не найден');

 END IF;

```
EXCEPTION
```

```
WHEN NO_DATA_FOUND THEN
```

```
    RAISE_APPLICATION_ERROR(-20010, 'Отдел не найден');
```

```
END;
```

```
END emp_package;
```

Также, я произвел проверки на ошибки, проверил работают ли исключения и заполнил/поменял существующие данные таблиц.

Позитивные тесты для проверки работы:

-- Добавление отдела

```
BEGIN
```

```
    dept_package.add_dept('Marketing', 'Boston');
```

```
END;
```

-- Подсчёт сотрудников в отделе

```
DECLARE v NUMBER;
```

```
BEGIN
```

```
    v := dept_package.count_employees_in('Sales');
```

```
    DBMS_OUTPUT.PUT_LINE(v);
```

```
END;
```

-- Все отделы с количеством

```
DECLARE c SYS_REFCURSOR; vname VARCHAR2(30); vcount NUMBER;
```

```
BEGIN
```

```
    c := dept_package.count_employees_all;
```

```
    LOOP
```



```
    FETCH c INTO vname, vcount;

    EXIT WHEN c%NOTFOUND;

    DBMS_OUTPUT.PUT_LINE(vname || ' - ' || vcount);

END LOOP;

CLOSE c;

END;
```

-- Переименование отдела

```
BEGIN

    dept_package.rename_dept('Marketing', 'PR');

END;
```

-- Перевод отдела

```
BEGIN

    dept_package.move_dept('PR', 'Seattle');

END;
```

-- Добавление сотрудника

```
BEGIN

    emp_package.add_employee('Ivanov',
    TO_DATE('01/01/2022','MM/DD/YYYY'), 2000, 'XY', 'Sales');

END;
```

-- Изменение зарплаты

```
BEGIN

    emp_package.change_salary('Ivanov', 300);

END;
```

-- Годовая зарплата сотрудника

```
DECLARE v NUMBER;
```

```
BEGIN
```

```
    v := emp_package.yearly_salary('Petrov');
```

```
    DBMS_OUTPUT.PUT_LINE(v);
```

```
END;
```

-- Фонд отдела

```
DECLARE v NUMBER;
```

```
BEGIN
```

```
    v := emp_package.yearly_salary_dept('Soft');
```

```
    DBMS_OUTPUT.PUT_LINE(v);
```

```
END;
```

-- Перевод сотрудника

```
BEGIN
```

```
    emp_package.move_employee('Ivanov', 'Soft');
```

```
END;
```

Вызовы для проверки ошибок (исключений):

-- Пытается добавить отдел с названием 'Soft', который уже существует в таблице DEPT.

```
BEGIN
```

```
    dept_package.add_dept('Soft', 'Chicago');
```

```
END;
```

-- Пытается получить количество сотрудников в несуществующем отделе 'NoSuchDept'.

DECLARE v NUMBER;

BEGIN

v := dept_package.count_employees_in('NoSuchDept');

END;

-- Пытается получить количество сотрудников в несуществующем отделе 'NoSuchDept'.

BEGIN

dept_package.rename_dept('GhostDept', 'Anything');

END;

-- Пытается перевести несуществующий отдел 'GhostDept' в другой город 'London'.

BEGIN

dept_package.move_dept('GhostDept', 'London');

END;

-- Пытается добавить сотрудника 'Ghost' в несуществующий отдел 'NoSuchDept'.

BEGIN

emp_package.add_employee('Ghost', SYSDATE, 1000, 'ZZ', 'NoSuchDept');

END;

-- Пытается изменить зарплату несуществующего сотрудника 'NoNameGuy'.

BEGIN

emp_package.change_salary('NoNameGuy', 500);

END;

-- Пытается получить годовую зарплату несуществующего сотрудника 'NoNameGuy'.

DECLARE v NUMBER;

BEGIN

v := emp_package.yearly_salary('NoNameGuy');

END;

-- Пытается получить фонд зарплаты для несуществующего отдела 'NoSuchDept'.

DECLARE v NUMBER;

BEGIN

v := emp_package.yearly_salary_dept('NoSuchDept');

END;

-- Пытается перевести несуществующего сотрудника 'NoNameGuy' в существующий отдел 'Soft'.

BEGIN

emp_package.move_employee('NoNameGuy', 'Soft');

END;

-- Пытается перевести сотрудника 'Petrov' в несуществующий отдел 'NoSuchDept'.

BEGIN

emp_package.move_employee('Petrov', 'NoSuchDept');

END;

Все проверки и заполнения выполняются без проблем и все работает как и нужно.