**Лабораторная работа №12**

**Пакет 1: Работа с отделами (dept\_pkg)**

**Спецификация пакета**  
CREATE OR REPLACE PACKAGE dept\_pkg AS

-- Исключения

e\_dept\_exists EXCEPTION;

e\_dept\_not\_found EXCEPTION;

-- Процедуры и функции

PROCEDURE new\_dept(p\_name IN DEPT.d\_name%TYPE, p\_loc IN DEPT.location%TYPE);

FUNCTION count\_emp\_in\_dept(p\_name IN DEPT.d\_name%TYPE) RETURN NUMBER;

PROCEDURE count\_all\_emp\_per\_dept;

PROCEDURE rename\_dept(p\_old\_name IN DEPT.d\_name%TYPE, p\_new\_name IN DEPT.d\_name%TYPE);

PROCEDURE relocate\_dept(p\_name IN DEPT.d\_name%TYPE, p\_new\_loc IN DEPT.location%TYPE);

END dept\_pkg;

**Тело пакета**  
  
CREATE OR REPLACE PACKAGE BODY dept\_pkg AS

PROCEDURE new\_dept(p\_name IN DEPT.d\_name%TYPE, p\_loc 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 e\_dept\_exists;

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\_loc);

EXCEPTION

WHEN e\_dept\_exists THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Такой отдел уже существует');

END;

FUNCTION count\_emp\_in\_dept(p\_name IN DEPT.d\_name%TYPE) RETURN NUMBER IS

v\_count NUMBER;

BEGIN

SELECT COUNT(\*) INTO v\_count

FROM EMP

WHERE d\_no = (SELECT d\_no FROM DEPT WHERE d\_name = p\_name);

RETURN v\_count;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR(-20002, 'Отдел не найден');

END;

PROCEDURE count\_all\_emp\_per\_dept IS

CURSOR c\_depts IS SELECT d\_name FROM DEPT;

v\_name DEPT.d\_name%TYPE;

v\_count NUMBER;

BEGIN

FOR rec IN c\_depts LOOP

v\_name := rec.d\_name;

SELECT COUNT(\*) INTO v\_count

FROM EMP

WHERE d\_no = (SELECT d\_no FROM DEPT WHERE d\_name = v\_name);

DBMS\_OUTPUT.PUT\_LINE('Отдел: ' || v\_name || ' — сотрудников: ' || v\_count);

END LOOP;

END;

PROCEDURE rename\_dept(p\_old\_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\_old\_name;

IF v\_count = 0 THEN

RAISE e\_dept\_not\_found;

END IF;

UPDATE DEPT SET d\_name = p\_new\_name WHERE d\_name = p\_old\_name;

EXCEPTION

WHEN e\_dept\_not\_found THEN

RAISE\_APPLICATION\_ERROR(-20003, 'Указанный отдел не найден');

END;

PROCEDURE relocate\_dept(p\_name IN DEPT.d\_name%TYPE, p\_new\_loc 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 e\_dept\_not\_found;

END IF;

UPDATE DEPT SET location = p\_new\_loc WHERE d\_name = p\_name;

EXCEPTION

WHEN e\_dept\_not\_found THEN

RAISE\_APPLICATION\_ERROR(-20004, 'Указанный отдел не найден');

END;

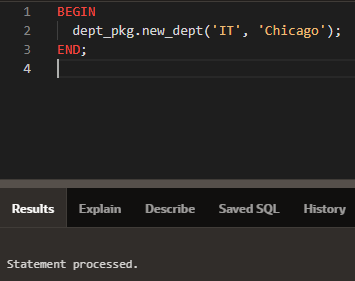
END dept\_pkg;  
  
  
**Тесты для dept\_pkg**

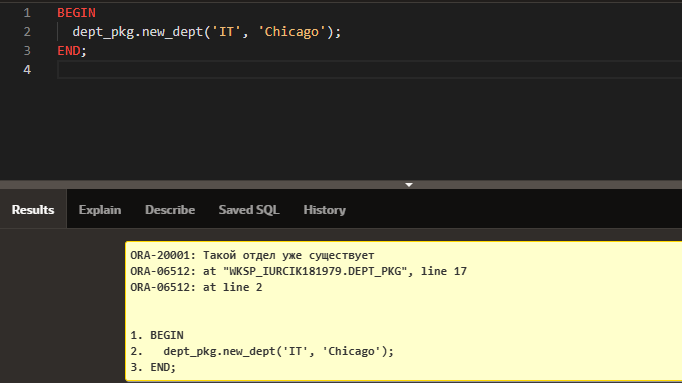
**1. Создание нового отдела new\_dept (p\_name, p\_loc)**

**Пример:** Добавим отдел IT в Chicago

BEGIN

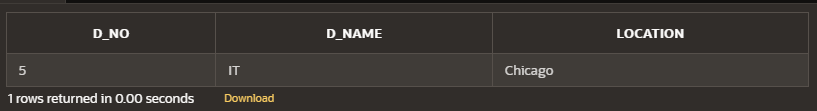
dept\_pkg.new\_dept('IT', 'Chicago');

END;

  
 - Повторив вызов — будет ошибка ORA-20001: Такой отдел уже существует.

**Проверка:**

SELECT \* FROM DEPT WHERE d\_name = 'IT' AND location = 'Chicago';



**2. Подсчет количества сотрудников в указанном отделе count\_emp\_in\_dept (p\_name)**

**Пример:** Узнать сколько сотрудников в отделе 'Soft'

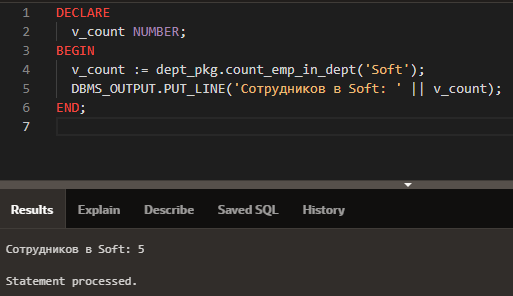
DECLARE

v\_count NUMBER;

BEGIN

v\_count := dept\_pkg.count\_emp\_in\_dept('Soft');

DBMS\_OUTPUT.PUT\_LINE('Сотрудников в Soft: ' || v\_count);

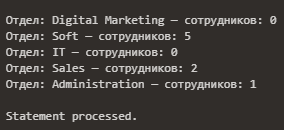
END;

**3. Подсчет количества сотрудников в каждом отделе (count\_all\_emp\_per\_dept)**

BEGIN

dept\_pkg.count\_all\_emp\_per\_dept;

END;



**4. Изменение названия указанного отдела на новое rename\_dept (p\_old\_name, p\_new\_name)**

**Пример:** Переименуем 'Marketing' в 'Digital Marketing'

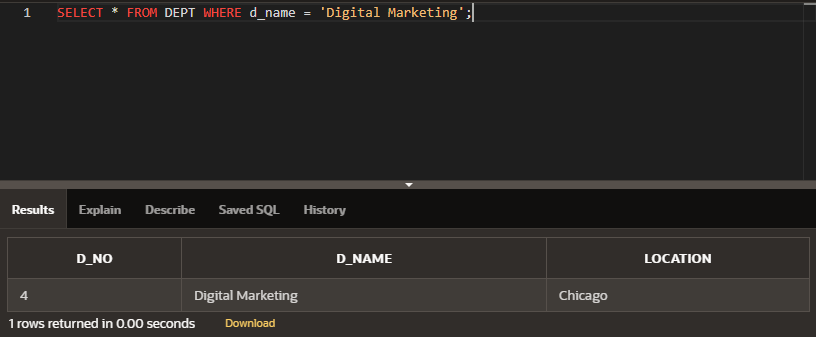
BEGIN

dept\_pkg.rename\_dept('Marketing', 'Digital Marketing');

END;

Затем:

SELECT \* FROM DEPT WHERE d\_name = 'Digital Marketing';



### 5. Перевод указанного отдела в новый город relocate\_dept (p\_name, p\_new\_loc)

**Пример:** Переносим 'Sales' в 'Chicago'

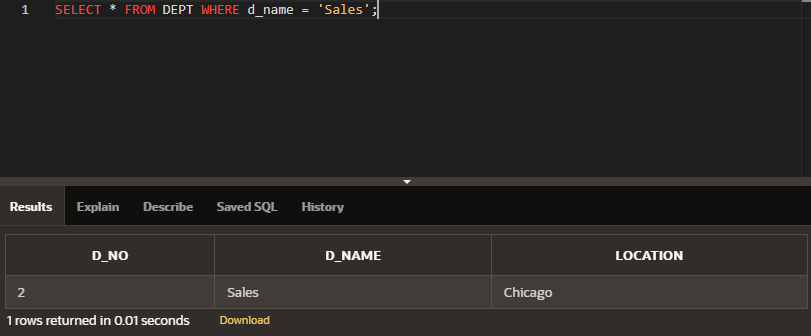
BEGIN

dept\_pkg.relocate\_dept('Sales', 'Chicago');

END;

**Проверка:**

SELECT \* FROM DEPT WHERE d\_name = 'Sales';



**Пакет 2: Работа с сотрудниками (emp\_pkg)**

**Спецификация пакета**

CREATE OR REPLACE PACKAGE emp\_pkg AS

-- Исключения

e\_emp\_exists EXCEPTION;

e\_emp\_not\_found EXCEPTION;

e\_dept\_not\_found EXCEPTION;

-- Подпрограммы

PROCEDURE add\_emp(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\_name IN DEPT.d\_name%TYPE);

PROCEDURE update\_salary(p\_name IN EMP.e\_name%TYPE, p\_new\_salary IN EMP.salary%TYPE);

FUNCTION yearly\_salary(p\_name IN EMP.e\_name%TYPE) RETURN NUMBER;

FUNCTION dept\_salary\_total(p\_dept\_name IN DEPT.d\_name%TYPE) RETURN NUMBER;

PROCEDURE move\_emp(p\_name IN EMP.e\_name%TYPE, p\_new\_dept IN DEPT.d\_name%TYPE);

END emp\_pkg;

**Тело пакета**

CREATE OR REPLACE PACKAGE BODY emp\_pkg AS

PROCEDURE add\_emp(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\_name IN DEPT.d\_name%TYPE) IS

v\_exists NUMBER;

v\_dept\_id DEPT.d\_no%TYPE;

v\_new\_id EMP.e\_no%TYPE;

BEGIN

SELECT COUNT(\*) INTO v\_exists FROM EMP WHERE e\_name = p\_name;

IF v\_exists > 0 THEN

RAISE e\_emp\_exists;

END IF;

SELECT d\_no INTO v\_dept\_id FROM DEPT WHERE d\_name = p\_dept\_name;

SELECT NVL(MAX(e\_no), 100099) + 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\_id);

EXCEPTION

WHEN e\_emp\_exists THEN

RAISE\_APPLICATION\_ERROR(-20010, 'Сотрудник уже существует');

WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR(-20011, 'Отдел не найден');

END;

PROCEDURE update\_salary(p\_name IN EMP.e\_name%TYPE, p\_new\_salary IN EMP.salary%TYPE) IS

v\_count NUMBER;

BEGIN

SELECT COUNT(\*) INTO v\_count FROM EMP WHERE e\_name = p\_name;

IF v\_count = 0 THEN

RAISE e\_emp\_not\_found;

END IF;

UPDATE EMP SET salary = p\_new\_salary WHERE e\_name = p\_name;

EXCEPTION

WHEN e\_emp\_not\_found THEN

RAISE\_APPLICATION\_ERROR(-20012, 'Сотрудник не найден');

END;

FUNCTION yearly\_salary(p\_name IN EMP.e\_name%TYPE) RETURN NUMBER IS

v\_salary EMP.salary%TYPE;

BEGIN

SELECT salary INTO v\_salary FROM EMP WHERE e\_name = p\_name;

RETURN v\_salary \* 12;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR(-20013, 'Сотрудник не найден');

END;

FUNCTION dept\_salary\_total(p\_dept\_name IN DEPT.d\_name%TYPE) RETURN NUMBER IS

v\_total NUMBER;

BEGIN

SELECT SUM(salary \* 12) INTO v\_total

FROM EMP

WHERE d\_no = (SELECT d\_no FROM DEPT WHERE d\_name = p\_dept\_name);

RETURN NVL(v\_total, 0);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR(-20014, 'Отдел не найден');

END;

PROCEDURE move\_emp(p\_name IN EMP.e\_name%TYPE, p\_new\_dept IN DEPT.d\_name%TYPE) IS

v\_emp\_count NUMBER;

v\_dept\_id DEPT.d\_no%TYPE;

BEGIN

SELECT COUNT(\*) INTO v\_emp\_count FROM EMP WHERE e\_name = p\_name;

IF v\_emp\_count = 0 THEN

RAISE e\_emp\_not\_found;

END IF;

SELECT d\_no INTO v\_dept\_id FROM DEPT WHERE d\_name = p\_new\_dept;

UPDATE EMP SET d\_no = v\_dept\_id WHERE e\_name = p\_name;

EXCEPTION

WHEN e\_emp\_not\_found THEN

RAISE\_APPLICATION\_ERROR(-20015, 'Сотрудник не найден');

WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR(-20016, 'Отдел не найден');

END;

END emp\_pkg;

**Тесты для emp\_pkg**

1. **Зачисление нового сотрудника add\_emp (p\_name, p\_date, p\_salary, p\_address, p\_dept\_name)**

**Пример:** Добавим нового сотрудника 'Ivanov' в отдел 'Soft'

BEGIN

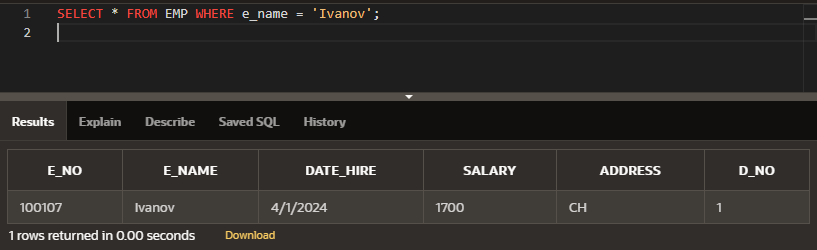
emp\_pkg.add\_emp('Ivanov', TO\_DATE('01.04.2024', 'DD.MM.YYYY'), 1700, 'CH', 'Soft');

END;



**Проверка:**

SELECT \* FROM EMP WHERE e\_name = 'Ivanov';



Если повторно — выдаст ошибку:

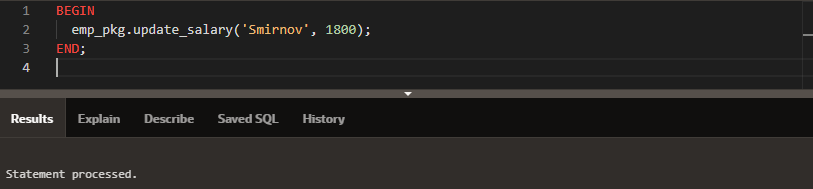
ORA-20010: Сотрудник уже существует

1. **Изменение зарплаты указанного сотрудника на заданную величину update\_salary (p\_name, p\_new\_salary)**

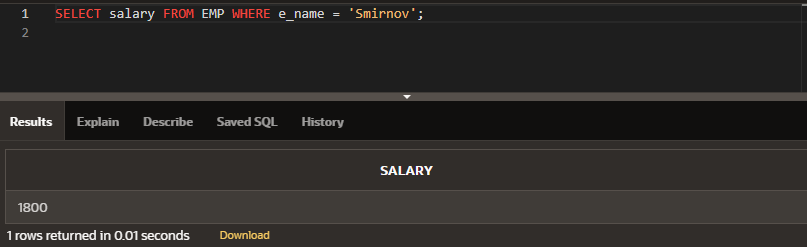
**Пример:** Изменим зарплату Smirnov на 1800

BEGIN

emp\_pkg.update\_salary ('Smirnov', 1800);

END;  
  


**Проверка:**

SELECT salary FROM EMP WHERE e\_name = 'Smirnov';   
  


1. **Расчет ежегодного объема зарплаты для указанного сотрудника yearly\_salary (p\_name) RETURN NUMBER**

**Пример:** Получим годовую зарплату Clark

DECLARE

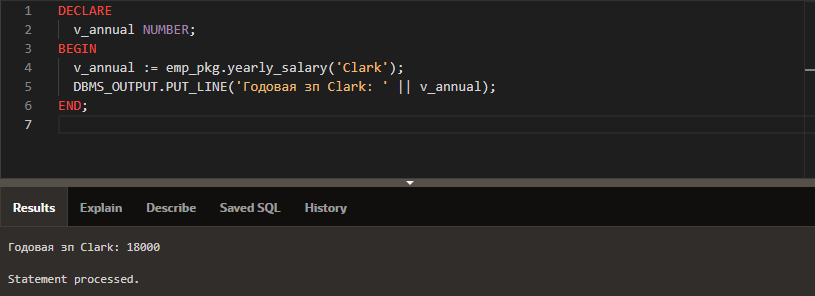
v\_annual NUMBER;

BEGIN

v\_annual := emp\_pkg.yearly\_salary('Clark');

DBMS\_OUTPUT.PUT\_LINE('Годовая зп Clark: ' || v\_annual);

END



1. **Расчет ежегодного объема зарплаты сотрудников в указанном отделе. dept\_salary\_total (p\_dept\_name) RETURN NUMBER**

**Пример:** Годовая зарплата всех сотрудников в 'Sales'

DECLARE

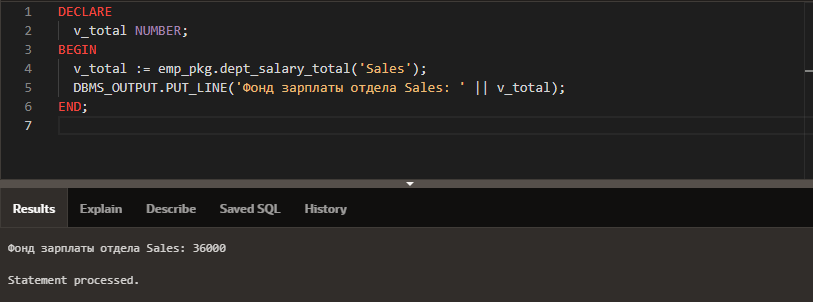
v\_total NUMBER;

BEGIN

v\_total := emp\_pkg.dept\_salary\_total('Sales');

DBMS\_OUTPUT.PUT\_LINE('Фонд зарплаты отдела Sales: ' || v\_total);

END;



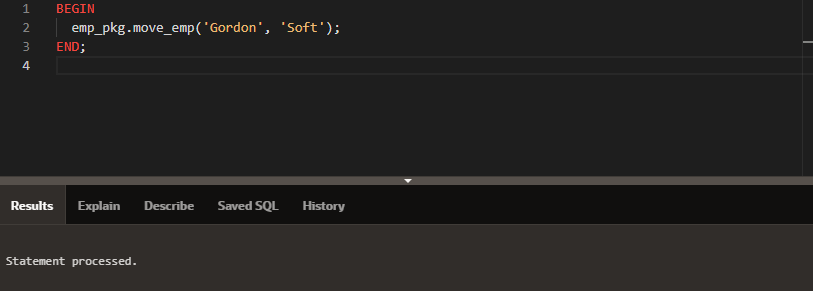
1. **Перевод сотрудника в указанный отдел. move\_emp (p\_name, p\_new\_dept)**

**Пример:** Переводим Gordon из 'Administration' в 'Soft'

BEGIN

emp\_pkg.move\_emp('Gordon', 'Soft');

END;



**Проверка:**

SELECT e\_name, d\_no FROM EMP WHERE e\_name = 'Gordon';

