

Лабораторная работа №11

Для Oracle APEX:

1. Написать функцию для определения количества сотрудников в указанном регионе.

```
CREATE OR REPLACE FUNCTION get_emp_count_by_region(
    p_region IN DEPT.location%TYPE
) RETURN NUMBER
IS
    v_count NUMBER;
BEGIN
    SELECT COUNT(*)
    INTO v_count
    FROM EMP E
    JOIN DEPT D ON E.d_no = D.d_no
    WHERE D.location = p_region;

    RETURN v_count;
END;
/
```

Проверка функции:

```
SELECT get_emp_count_by_region('New-York') FROM dual;
```

Создать подпрограммы, которые будут включать в себя следующие операции:

а) изменение зарплаты указанному сотруднику на заданную величину (2 параметра: фамилия сотрудника и величина изменения з/пл (если число положительное, то зарплата увеличивается, если отрицательное, то зарплата уменьшается)

исключение – нет сотрудника;

```

CREATE OR REPLACE PROCEDURE change_salary(
    p_name    IN EMP.e_name%TYPE,
    p_delta   IN NUMBER
)
IS
    v_count NUMBER;
BEGIN
    SELECT COUNT(*) INTO v_count FROM EMP WHERE e_name =
p_name;

    IF v_count = 0 THEN
        RAISE_APPLICATION_ERROR(-20001, 'Сотрудник не найден');
    ELSE
        UPDATE EMP
        SET salary = salary + p_delta
        WHERE e_name = p_name;
    END IF;
END;
/

```

Проверка:

```

BEGIN
    change_salary('Petrov', 300); -- Увеличит зарплату на 300
END;
/

```

```
BEGIN
```

```
change_salary('Smith', -200); -- Уменьшит зарплату на 200
```

```
END;
```

```
/
```

Затем просто пишем: `SELECT * FROM EMP;` для вывода таблицы с сотрудниками и проверки изменений.

б) Реструктуризация фирмы – отделы с названием Sales перевести в Dallas, остальные в New York (процедура без параметров) исключений нет.

```
CREATE OR REPLACE PROCEDURE restructure_company IS
```

```
BEGIN
```

```
UPDATE DEPT
```

```
SET location = CASE
```

```
    WHEN d_name = 'Sales' THEN 'Dallas'
```

```
    ELSE 'New-York'
```

```
END;
```

```
END;
```

```
/
```

Проверка:

```
BEGIN
```

```
restructure_company;
```

```
END;
```

```
/
```

Затем пишем SELECT * FROM DEPT; для проверки изменений.

Для MS SQL:

1. Написать функцию для определения количества сотрудников в указанном регионе.

```
CREATE FUNCTION get_emp_count_by_region (  
    @region NVARCHAR(50)  
)  
RETURNS INT  
AS  
BEGIN  
    DECLARE @count INT;  
    SELECT @count = COUNT(*)  
    FROM EMP E  
    JOIN DEPT D ON E.d_no = D.d_no  
    WHERE D.location = @region;  
  
    RETURN @count;  
END;
```

Проверка:

```
SELECT dbo.get_emp_count_by_region('New-York');
```

Создать подпрограммы, которые будут включать в себя следующие операции:

а) изменение зарплаты указанному сотруднику на заданную величину (2 параметра: фамилия сотрудника и величина изменения з/пл (если число положительное, то зарплата увеличивается, если отрицательное, то зарплата уменьшается)

исключение – нет сотрудника;

```
CREATE PROCEDURE change_salary
```

```
    @name NVARCHAR(50),
```

```
    @delta MONEY
```

```
AS
```

```
BEGIN
```

```
-- Проверка, есть ли сотрудник с такой фамилией
```

```
IF NOT EXISTS (SELECT 1 FROM EMP WHERE e_name = @name)
```

```
BEGIN
```

```
    RAISERROR('Сотрудник не найден', 16, 1);
```

```
    RETURN;
```

```
END
```

```
-- Обновление зарплаты
```

```
UPDATE EMP
```

```
SET salary = salary + @delta
```

```
WHERE e_name = @name;
```

```
END;
```

Проверка:

```
EXEC change_salary @name = 'Petrov', @delta = 300;  
EXEC change_salary @name = 'Smith', @delta = -200;
```

Затем также прописываем `SELECT * FROM EMP`; для проверки изменений.

б) Реструктуризация фирмы – отделы с названием Sales перевести в Dallas, остальные в New York (процедура без параметров) исключений нет.

```
CREATE PROCEDURE restructure_company
```

```
AS
```

```
BEGIN
```

```
    UPDATE DEPT
```

```
    SET location = CASE
```

```
        WHEN d_name = 'Sales' THEN 'Dallas'
```

```
        ELSE 'New-York'
```

```
    END;
```

```
END;
```

Проверка:

```
EXEC restructure_company;
```

Для проверки изменений пишем команду `SELECT * FROM DEPT`;

Лабораторная работа №11а

1. Сотрудникам, которые устроились на фирму после 01.01.2022, увеличить зарплату на 5%, тем, которые устроились на фирму с 02.01.2021 по 30.12.2021 — увеличить зарплату на 3%, остальным сотрудникам увеличить зарплату на 1%. (Учебная БД №2)

Вывести на экран следующие данные: фамилия, имя сотрудника, дата устройства на работу, старая зарплата, новая зарплата, процент увеличения зарплаты.

Для MS SQL:

```
CREATE PROCEDURE update_salary_by_hire_date
```

```
AS
```

```
BEGIN
```

```
-- Временная таблица для вывода результата
```

```
CREATE TABLE #Result (
```

```
    f_name VARCHAR(50),
```

```
    l_name VARCHAR(50),
```

```
    date_hire DATE,
```

```
    old_salary INT,
```

```
    new_salary INT,
```

```
    increase_percent INT
```

```
);
```

```
INSERT INTO #Result (f_name, l_name, date_hire, old_salary, new_salary,  
increase_percent)
```

```

SELECT
    f_name,
    l_name,
    date_hire,
    p_rate,
    CASE
        WHEN date_hire > '2022-01-01' THEN CAST(p_rate * 1.05 AS INT)
        WHEN date_hire BETWEEN '2021-01-02' AND '2021-12-30' THEN
CAST(p_rate * 1.03 AS INT)
        ELSE CAST(p_rate * 1.01 AS INT)
    END AS new_salary,
    CASE
        WHEN date_hire > '2022-01-01' THEN 5
        WHEN date_hire BETWEEN '2021-01-02' AND '2021-12-30' THEN 3
        ELSE 1
    END AS increase_percent
FROM EMPLOYEE;

-- Обновление зарплат
UPDATE EMPLOYEE
SET p_rate = r.new_salary
FROM EMPLOYEE e
JOIN #Result r ON e.f_name = r.f_name AND e.l_name = r.l_name;

```



```
-- Вывод результата

SELECT * FROM #Result;

DROP TABLE #Result;

END;
```

Проверка:

```
EXEC update_salary_by_hire_date;
```

2. Определить зарплату сотрудников за период: 01.02.2011 – 15.02.2011 (Учебная БД №1)

Вывести на экран следующие данные: имя и фамилию сотрудника, его комиссионные, зарплату за указанный период

```
CREATE PROCEDURE get_salary_by_period
AS
BEGIN
    SELECT
        sp.sp_name,
        sp.comm,
        SUM(p.price * s.qty * sp.comm / 100.0) AS total_salary
    FROM SALE s
    JOIN SPERSON sp ON s.sp_id = sp.sp_id
    JOIN PRODUCT p ON s.p_id = p.p_id
    WHERE s.data BETWEEN '2011-02-01' AND '2011-02-15'
    GROUP BY sp.sp_name, sp.comm;
END;
```

Проверка:

```
EXEC get_salary_by_period;
```

3. Вывести на экран следующие данные: наименование товара, id товара, его закупочную стоимость для указанной в качестве параметра фирмы. (функция)

```
CREATE FUNCTION get_products_by_manufacturer (  
    @manufacturer_name NVARCHAR(100)  
)  
  
RETURNS TABLE  
  
AS  
  
RETURN (  
  
    SELECT  
  
        p.p_desc,  
  
        p.p_id,  
  
        p.cost  
  
    FROM PRODUCT p  
  
    JOIN MANUFACT m ON p.m_id = m.m_id  
  
    WHERE m.m_name = @manufacturer_name  
  
);
```

Проверка:

```
SELECT * FROM get_products_by_manufacturer('Lampy Lama');
```

Для Oracle Apex:

1. Сотрудникам, которые устроились на фирму после 01.01.2022, увеличить зарплату на 5%, тем, которые устроились на фирму с 02.01.2021 по 30.12.2021 — увеличить зарплату на 3%, остальным сотрудникам увеличить зарплату на 1%. (Учебная БД №2)

Вывести на экран следующие данные: фамилия, имя сотрудника, дата устройства на работу, старая зарплата, новая зарплата, процент увеличения зарплаты.

```
CREATE OR REPLACE PROCEDURE update_salary_by_hire_date IS
BEGIN
    FOR rec IN (
        SELECT e_no, f_name, l_name, date_hire, p_rate
        FROM EMPLOYEE
    ) LOOP
        DECLARE
            new_salary NUMBER;
            percent NUMBER;
        BEGIN
            IF rec.date_hire > TO_DATE('01.01.2022', 'DD.MM.YYYY') THEN
                percent := 5;
                new_salary := rec.p_rate * 1.05;
            ELSIF rec.date_hire BETWEEN TO_DATE('02.01.2021',
            'DD.MM.YYYY') AND TO_DATE('30.12.2021', 'DD.MM.YYYY') THEN
                percent := 3;
                new_salary := rec.p_rate * 1.03;
            ELSE
                percent := 1;
                new_salary := rec.p_rate * 1.01;
            END IF;

            -- Обновляем зарплату
            UPDATE EMPLOYEE
            SET p_rate = ROUND(new_salary)
            WHERE e_no = rec.e_no;
```

```

-- Вывод результата
DBMS_OUTPUT.PUT_LINE(
    'Сотрудник: ' || rec.l_name || ' ' || rec.f_name ||
    ', Дата найма: ' || TO_CHAR(rec.date_hire, 'DD.MM.YYYY') ||
    ', Старая з/п: ' || rec.p_rate ||
    ', Новая з/п: ' || ROUND(new_salary) ||
    ', Увеличение: ' || percent || '%'
);
END;
END LOOP;
END;
/

```

Проверка:

```

BEGIN
    update_salary_by_hire_date;
END;

```

2. Определить зарплату сотрудников за период: **01.02.2011 – 15.02.2011** (Учебная БД №1)

Вывести на экран следующие данные: **имя и фамилию сотрудника, его комиссионные, зарплату за указанный период**

```

CREATE OR REPLACE PROCEDURE get_salary_by_period IS

```

```

BEGIN

```

```

    FOR rec IN (

```

```

        SELECT

```

```

            sp.sp_name,

```

```

            sp.comm,

```

```

            SUM(p.price * s.qty * sp.comm / 100) AS total_salary

```

```

        FROM SALE s

```

```

        JOIN SPERSON sp ON s.sp_id = sp.sp_id

        JOIN PRODUCT p ON s.p_id = p.p_id

        WHERE s.data BETWEEN TO_DATE('01.02.2011', 'DD.MM.YYYY')
        AND TO_DATE('15.02.2011', 'DD.MM.YYYY')

        GROUP BY sp.sp_name, sp.comm
    )

    LOOP

        DBMS_OUTPUT.PUT_LINE(

            'Имя: ' || rec.sp_name ||

            ', Комиссия: ' || rec.comm || '%' ||

            ', Зарплата: ' || ROUND(rec.total_salary, 2)

        );

    END LOOP;

END;

/

```

Проверка:

```

BEGIN

    get_salary_by_period;

END;

```

3. Вывести на экран следующие данные: наименование товара, id товара, его закупочную стоимость для указанной в качестве параметра фирмы. (функция)

Тип строки, которая содержит нужные поля: описание товара, ID и закупочную цену:

```
CREATE OR REPLACE TYPE product_info_row AS OBJECT (  
    p_desc VARCHAR2(100),  
    p_id  NUMBER,  
    cost  NUMBER  
);  
/
```

Тип "таблица строк", чтобы функция могла возвращать несколько записей:

```
CREATE OR REPLACE TYPE product_info_table AS TABLE OF  
product_info_row;  
/
```

Функция возвращает все товары указанной фирмы:

```
CREATE OR REPLACE FUNCTION get_products_by_manufacturer (  
    p_manufacturer_name IN VARCHAR2  
) RETURN product_info_table  
PIPELINED  
IS  
BEGIN
```

```
FOR rec IN (  
    SELECT p.p_desc, p.p_id, p.cost  
    FROM PRODUCT p  
    JOIN MANUFACT m ON p.m_id = m.m_id  
    WHERE m.m_name = p_manufacturer_name  
)  
LOOP  
    PIPE ROW(product_info_row(rec.p_desc, rec.p_id, rec.cost));  
END LOOP;  
  
RETURN;  
  
END;  
  
/
```

Проверка:

```
SELECT * FROM TABLE(get_products_by_manufacturer('Lampy Lama'));
```