**ТЕМА 10. ФУНКЦИИ PLPGSQL**

1. **Введение в PL/pgSQL**

PL/pgSQL - это язык программирования, который используется в PostgreSQL для создания хранимых процедур, триггеров и функций. Вот простой пример PL/pgSQL-функции:

-- Создание функции

CREATE OR REPLACE FUNCTION greet(name text) RETURNS text AS $$

BEGIN

RETURN 'Привет, ' || name || '!';

END;

$$ LANGUAGE plpgsql;

-- Вызов функции

SELECT greet('Петя');

1. **Возврат и присвоение**

Примеры операций возврата и присвоения значений переменным:

-- Присвоение переменной значения

DECLARE

my\_variable text;

BEGIN

my\_variable := 'Значение';

-- Возврат значения

RETURN my\_variable;

END;

-- Вызов функции и получение результата

SELECT my\_function();

1. **Декларация переменных в SQL**

Пример декларации переменных в SQL:

-- Декларация переменных

DECLARE

var1 INT;

var2 TEXT;

BEGIN

-- Присвоение значений переменным

var1 := 10;

var2 := 'Пример';

-- Использование переменных

-- ...

END;

1. **Логика с if else**

Использование условных операторов IF и ELSE:

-- Пример условного оператора

DECLARE

x INT := 5;

BEGIN

IF x > 0 THEN

-- Действие при выполнении условия

RAISE NOTICE 'x положительное число';

ELSE

-- Действие при не выполнении условия

RAISE NOTICE 'x не положительное число';

END IF;

END;

1. **Циклы в PL/pgSQL**

Использование циклов FOR в PL/pgSQL:

-- Пример цикла FOR

DECLARE

i INT;

BEGIN

FOR i IN 1..5 LOOP

-- Действие в цикле

RAISE NOTICE 'Итерация %', i;

END LOOP;

END;

1. **RETURN NEXT**

Использование RETURN NEXT для возврата набора записей:

-- Пример использования RETURN NEXT

CREATE OR REPLACE FUNCTION get\_employees() RETURNS SETOF employees AS $$ DECLARE

emp\_record employees%ROWTYPE;

BEGIN

FOR emp\_record IN (SELECT \* FROM employees) LOOP

-- Возврат записи

RETURN NEXT emp\_record;

END LOOP;

RETURN;

END;

$$ LANGUAGE plpgsql;

-- Вызов функции для получения набора записей

SELECT \* FROM get\_employees();

**Код для добавления столбца и обновления данных:**

-- Добавление столбца "salary" в таблицу "employees"

ALTER TABLE employees

ADD COLUMN salary numeric(12,2);

-- Обновление данных

UPDATE employees

SET salary=64.47 WHERE employee\_id=1;

UPDATE employees

SET salary=52.42 WHERE employee\_id=2;

UPDATE employees

SET salary=78.47 WHERE employee\_id=3;

UPDATE employees

SET salary=62.95 WHERE employee\_id=4;

UPDATE employees

SET salary=55.56 WHERE employee\_id=5;

UPDATE employees

SET salary=54.92 WHERE employee\_id=6;

UPDATE employees

SET salary=64.35 WHERE employee\_id=7;

UPDATE employees

SET salary=75.60 WHERE employee\_id=8;

UPDATE employees

SET salary=0.00 WHERE employee\_id=9;

-- Выборка данных

SELECT employee\_id, salary FROM employees

ORDER BY employee\_id;