**ЛАБОРАТОРНАЯ РАБОТА № 4. ХРАНИМЫЕ ПРОЦЕДУРЫ**

В соответствии с описанием раздела 2.4 создайте хранимые процедуры:

* содержащие входные параметры;
* содержащие выходные параметры;
* содержащие входные и выходные параметры;
* возвращающие набор строк;
* использующие переменные;
* использующие конструкции IF ... THEN ... ELSE.

Продемонстрируйте изменение и удаление созданной ранее хранимой процедуры.

Хранимая процедура − это отдельная программа, написанная на процедурном языке ис-пользуемого сервера БД. Существует две разновидности хранимых процедур:

* процедуры выбора (аналог SELECT-запросов), используются в операторе SELECT вме-сто таблицы и, следовательно, возвращающие наборы данных (в заголовке процедуры обяза-тельно должен быть определен набор выходных параметров);
* исполняемые процедуры, предназначенные для исполнения команд, например, UPDATE.

Выходные параметры определяют набор данных, возвращаемых хранимой процедурой. Тип величин, возвращаемой функцией Pl/pgSQL, должен соответствовать типу возвращаемого значения, указанному при создании функции командой create function. Значение возвращается командой return. Команда return находится в конце функции, но она также часто встречается в командах if или других командах, осуществляющих передачу управления программе. Даже если команда return вызывается в одной из этих команд, функция все равно должна заканчиваться командой return (даже если управлене никогда не будет передано этой завершающей команде). Синтаксис команды return приведен ниже:

CREATE FUNCTION ФУНКЦИЯ (АРГУМЕНТЫ) RETURN ТИП AS

$BODY$

DECLARE

ОБЪЯВЛЕНИЕ:

[…]

BEGIN

КОМАНДА:

[…]

RETURN {ПЕРЕМЕННАЯ | ЗНАЧЕНИЕ}

END

$BODY$

LANGUAGE 'PLPGSQL';

Конструкция IF ... THEN ... ELSE в языке SQL применяется точно так же, как и в других языках программирования. Если условие, заданное после ключевого слова IF, истинно, то выполняется блок операторов после ключевого слова THEN. В противном случае выполняется блок операторов после ключевого слова ELSE.

Циклы, как и условная команда, аналогичны циклам из обычных языков программирования.

Цикл FOR выполняет одну итерацию для каждого значения переменной переменнаяв интервале, границы которого определяются выражениями выражение1 и выражение2(включительно).

При вызове функции PL/pgSQL из кода PL/pgSQL имя функции обычно включа­ется в команду SQL SELECT или в команду присваивания.

Команда drop procedure – удалят БД. Ее синтаксис:

DROP procedure имя\_процедуры

**Примеры процедур и их вызов:**

CREATE OR REPLACE FUNCTION zakazy.count\_fio()

RETURNS integer AS

$BODY$

declare

num INTEGER;

begin

select into num count(fio\_klient) from zakazy.klient;

RETURN num;

end;

$BODY$

LANGUAGE 'plpgsql';

SELECT zakazy.count\_fio()

CREATE OR REPLACE FUNCTION zakazy.count\_product()

RETURNS integer AS

$BODY$

declare

num INTEGER;

begin

select into num count(name\_product) from zakazy.tovar;

RETURN num;

end;

$BODY$

LANGUAGE 'plpgsql';

SELECT zakazy.count\_product()

CREATE OR REPLACE FUNCTION zakazy.get\_slzh(id\_klient1 integer)

RETURNS character varying AS

$BODY$

DECLARE

id\_klient1 ALIAS FOR $1;

fio\_klient1 text;

name\_klient1 text;

BEGIN

SELECT INTO fio\_klient1, name\_klient1

fio\_klient , name\_klient

FROM zakazy.klient

WHERE klient.id\_klient = id\_klient1;

RETURN fio\_klient1 || ' ' || name\_klient1;

END;

$BODY$

LANGUAGE 'plpgsql'

SELECT zakazy.get\_slzh(11)

CREATE OR REPLACE FUNCTION zakazy.delete\_product(id\_product1 integer)

RETURNS integer AS

$BODY$

DECLARE

num\_product integer := 0;

BEGIN

SELECT INTO num\_product COUNT(id\_product)

FROM zakazy.tovar

WHERE id\_product1 = $1;

IF num\_product > 0 THEN RETURN -1;

ELSE

DELETE FROM zakazy.tovar

WHERE id\_product1 = $1;

RETURN 1;

END IF;

END;

$BODY$

LANGUAGE 'plpgsql'

SELECT zakazy.delete\_product (4)

CREATE OR REPLACE FUNCTION zakazy.get\_fio()

RETURNS character varying AS

$BODY$

DECLARE

row zakazy.sotrydnik%rowtype;

str text := '';

BEGIN

FOR row IN SELECT \*

FROM zakazy.sotrydnik LOOP

str := str || row.fio\_sotrydnik || ' ' || row.name\_sotrydnik || ' ' || row.nomer || E'\n';

END loop;

RETURN str;

END;

$BODY$

LANGUAGE 'plpgsql'

SELECT zakazy.get\_fio()