

Тема 7: PL/pgSQL: обзор и конструкции языка.



- 1) История PL/pgSQL.
- 2) Структура блока, объявление переменных.
- 3) Анонимные блоки.
- 4) Функции на языке PL/pgSQL.
- 5) Условные операторы и циклы.
- 6) Вычисление выражений.

PL/pgSQL – язык программирования, который используется в СУБД PostgreSQL для написания функций, триггеров и других управляющих конструкций.

Появился в версии 6.4 в 1998 году

устанавливается по умолчанию с версии 9.0

Цели создания

простой язык для написания пользовательских функций и триггеров

добавить управляющие структуры к языку SQL

сохранить возможность использования любых пользовательских типов, функций и операторов

Родословная: Oracle PL/SQL, Ада

Метка блока

Объявления переменных

область действия — блок

область видимости может перекрываться вложенными блоками,

но можно сослаться на переменную, используя метку блока

любые типы SQL, ссылки на типы объектов (%TYPE)

Операторы

управляющие конструкции

операторы SQL, кроме служебных

Обработка исключительных ситуаций

Разовое выполнение процедурного кода:

- без создания хранимой подпрограммы;
- без возможности передать параметры;
- без возможности вернуть значение.

Оператор DO языка SQL.

Заголовок подпрограммы не зависит от языка:

- имя, входные и выходные параметры;
- для функций: возвращаемое значение, категория

изменчивости.

Указание `LANGUAGE plpgsql`.

Возврат значений:

- оператор `RETURN`;
- присвоение значений выходным (`INOUT`, `OUT`) параметрам.

IF

стандартный условный оператор

CASE

похож на CASE языка SQL, но не возвращает значение

Внимание: трехзначная логика

условие должно быть истинно; false и NULL не подходят

Цикл FOR по диапазону чисел.

Цикл WHILE с предусловием.

Бесконечный цикл.

Цикл может иметь метку, как блок.

Управление:

- выход из цикла (EXIT);**
- переход на новую итерацию (CONTINUE).**

Любое выражение вычисляется в контексте SQL

- выражение автоматически преобразуется в запрос;
- запрос подготавливается;
- переменные PL/pgSQL подставляются как параметры.

Особенности

- можно использовать все возможности SQL, включая подзапросы
- невысокая скорость выполнения, хотя разобранный запрос (и, возможно, план запроса) кешируются неоднозначности при разрешении имен требуют внимания

PL/pgSQL — доступный по умолчанию, интегрированный с SQL, удобный и простой в использовании язык.

Управление подпрограммами на PL/pgSQL не отличается от работы с подпрограммами на других языках.

DO — команда SQL для выполнения анонимного блока.

Переменные PL/pgSQL могут использовать любые типы SQL.

Язык поддерживает обычные управляющие конструкции, такие как условные операторы и циклы.

1. Измените функцию `book_name` так, чтобы длина возвращаемого значения не превышала 45 символов. Если название книги при этом обрезается, оно должно завершаться на троеточие. Проверьте реализацию в SQL и в приложении; при необходимости добавьте книг с длинными названиями.
2. Снова измените функцию `book_name` так, чтобы избыточно длинное название уменьшалось на целое слово. Проверьте реализацию.

1. Например:

Путешествия в некоторые удалённые страны мира в четырёх частях: сочинение Лемюэля Гулливера, сначала хирурга, а затем капитана нескольких кораблей →
→ Путешествия в некоторые удалённые страны м...

Вот некоторые случаи, которые имеет смысл проверить:

- длина названия меньше 45 символов (не должно измениться);
- - длина названия ровно 45 символов (не должно измениться);
- - длина названия 46 символов (от названия должны быть отрезаны 4 символа, т. к. добавятся еще три точки).

Лучше всего написать и отладить отдельную функцию укорачивания, которую затем использовать в `book_name`. Это полезно и по другим соображениям:

- такая функция может пригодиться где-то еще;
- каждая функция будет выполнять ровно одну задачу.

2. Например:

Путешествия в некоторые удалённые страны мира в четырёх частях:
сочинение Лемюэля Гулливера, сначала хирурга, а затем капитана нескольких
кораблей →

→ Путешествия в некоторые удалённые страны...

Как поведет себя ваша реализация, если название состоит из одного длинного
слова без пробелов?