

Тема 7.3: PL/pgSQL: динамические команды.



- 1) Причины использования.**
- 2) Выполнение динамического запроса.**
- 3) Способы формирования динамического запроса.**

Текст SQL-команды формируется в момент выполнения.

Причины использования:

- дополнительная гибкость в приложении
- формирование нескольких конкретных запросов
- вместо одного универсального для оптимизации

Цена:

- операторы не подготавливаются
- возрастает риск внедрения SQL-кода
- возрастает сложность сопровождения

Оператор EXECUTE

выполняет строковое представление SQL-запроса

позволяет использовать параметры

переменные PL/pgSQL не становятся неявными параметрами

Может использоваться вместо SQL-запроса

сам по себе

при открытии курсора

в цикле по запросу

в предложении RETURN QUERY

Подстановка значений параметров

предложение USING

гарантируется невозможность внедрения SQL-кода

Экранирование значений

идентификаторы: `format('%I')`, `quote_ident`

литералы: `format('%L')`, `quote_literal`, `quote_nullable`

внедрение SQL-кода невозможно при правильном использовании

Обычные строковые функции

конкатенация и др.

возможно внедрение SQL-кода!

Динамические команды дают дополнительную гибкость.

Формирование отдельных запросов для разных значений параметров с целью оптимизации.

Не подходят для коротких, частых запросов.

Увеличивается сложность поддержки.

1. Измените функцию `get_catalog` так, чтобы запрос к представлению `catalog_v` формировался динамически и содержал условия только на те поля, которые заполнены на форме поиска в «Магазине». Проверьте работу функции в приложении.