



PostgreSQL

Разработка серверной части приложений PostgreSQL 16 **(dev-1)**



PL/pgSQL

Выполнение запросов



Использование команд SQL в коде PL/pgSQL

Устранение неоднозначностей именования

Проверка статуса команды

Табличные функции

Команды SQL встраиваются в код PL/pgSQL

как и в выражениях:

запрос подготавливается,

переменные PL/pgSQL подставляются как параметры

SELECT → PERFORM

удобно для вызова функций с побочными эффектами

запросы, начинающиеся на WITH, надо «оборачивать» в SELECT

INSERT, UPDATE, DELETE и другие команды SQL

кроме служебных команд

управление транзакциями — только в процедурах и анонимных блоках

Одна строка результата

SELECT ... INTO

получение первой по порядку строки результата

одна переменная составного типа

или подходящее количество скалярных переменных

INSERT, UPDATE, DELETE RETURNING ... INTO

получение вставленной (измененной, удаленной) строки

одна переменная составного типа

или подходящее количество скалярных переменных

INTO STRICT

гарантия получения ровно одной строки — ни больше, ни меньше

Диагностика ROW_COUNT

число строк, возвращенных (обработанных) последней командой SQL

Переменная FOUND

после команды SQL: истина, если команда вернула (обработала) строку

после цикла: признак того, что выполнялась хотя бы одна итерация

Строки запроса

`RETURN QUERY запрос;`

Одна строка

`RETURN NEXT выражение;`

если нет выходных параметров

`RETURN NEXT;`

если есть выходные параметры

Особенности

строки добавляются к результату,
но выполнение функции не прекращается

команды можно выполнять несколько раз

результат не возвращается, пока функция не завершится

PL/pgSQL тесно интегрирован с SQL

- в процедурном коде можно выполнять запросы (оформленные как выражения или отдельные команды)
- в запросах можно использовать переменные
- можно получать результаты запросов и их статус

Нужно следить за неоднозначностями разрешения имен



1. Напишите функцию `add_author` для добавления новых авторов. Функция должна принимать три параметра (фамилия, имя, отчество) и возвращать идентификатор нового автора.

Проверьте, что приложение позволяет добавлять авторов.

2. Напишите функцию `buy_book` для покупки книги. Функция принимает идентификатор книги и уменьшает количество таких книг на складе на единицу. Возвращаемое значение отсутствует.

Проверьте, что в «Магазине» появилась возможность покупки книг.

Напишите игру, в которой сервер пытается угадать загаданное пользователем животное, задавая последовательные уточняющие вопросы, на которые можно отвечать «да» или «нет».

Если сервер предложил неправильный вариант, он запрашивает у пользователя имя животного и отличающий вопрос. Эта новая информация запоминается и используется в следующих играх.

1. Создайте таблицу для представления информации.
2. Придумайте интерфейс и реализуйте необходимые функции.
3. Проверьте реализацию.