

Тема 7.7: PL/pgSQL: триггеры.



- 1) Триггеры и триггерные функции.
- 2) В какой момент срабатывают триггеры.
- 3) Контекст выполнения триггерной функции.
- 4) Возвращаемое значение.
- 5) Для чего можно применять триггеры и для чего не нужно.
- 6) Событийные триггеры.

Триггер

объект базы данных — список обрабатываемых событий
при возникновении события вызывается триггерная функция и ей
передается контекст

Триггерная функция

объект базы данных — код обработки события
выполняется в той же транзакции, что и основная операция
соглашение: функция не принимает параметры, возвращает
значение псевдотипа `trigger` (фактически `record`)
может использоваться в нескольких триггерах

INSERT, UPDATE, DELETE

таблицы before/after statement

before/after row

представления before/after statement

instead of row

TRUNCATE

таблицы before/after statement

Условие WHEN

устанавливает дополнительный фильтр

Срабатывает

до операции

Возвращаемое значение

игнорируется

Контекст

TG-переменные



Срабатывает

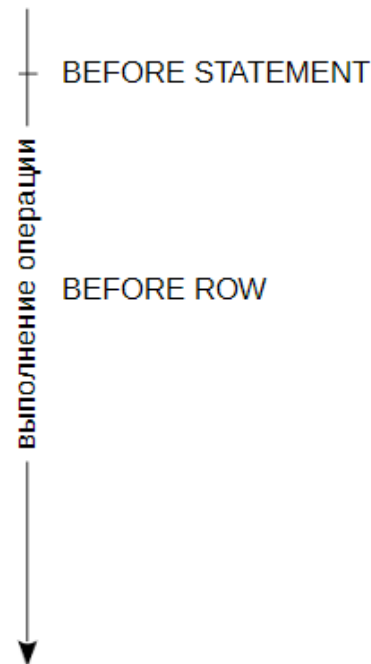
перед действием со строкой
в процессе выполнения операции

Возвращаемое значение

строка (возможно, измененная)
null отменяет действие

Контекст

OLD update, delete
NEW insert, update
TG-переменные



Срабатывает

вместо действия со строкой
для представлений

Возвращаемое значение

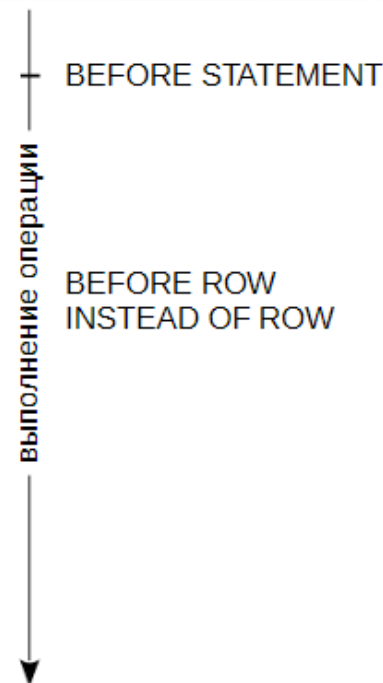
строка (возможно, измененная) —
будет видна в RETURNING
null отменяет действие

Контекст

OLD update, delete

NEW insert, update

TG-переменные



Срабатывает

после выполнения операции
очередь из прошедших условие WHEN

Возвращаемое значение

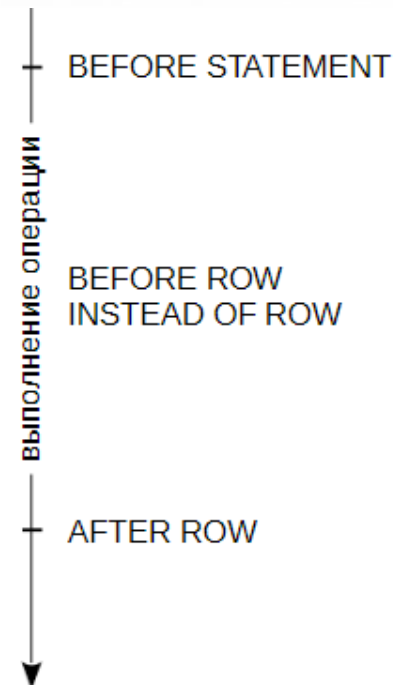
Игнорируется

Контекст

OLD, OLD TABLE update, delete

NEW, NEW TABLE insert, update

TG-переменные



Срабатывает

после операции

(даже если не затронута ни одна строка)

Возвращаемое значение

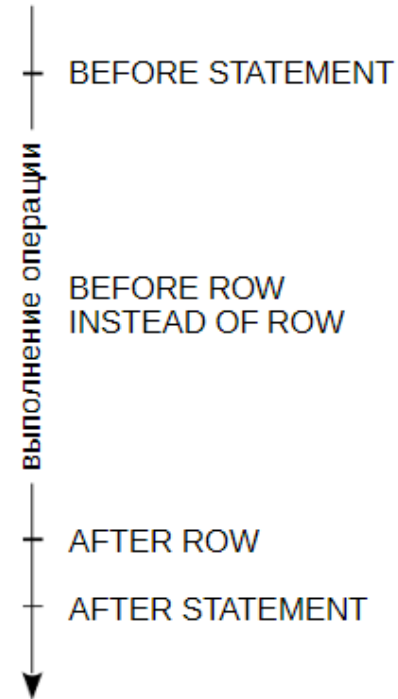
игнорируется

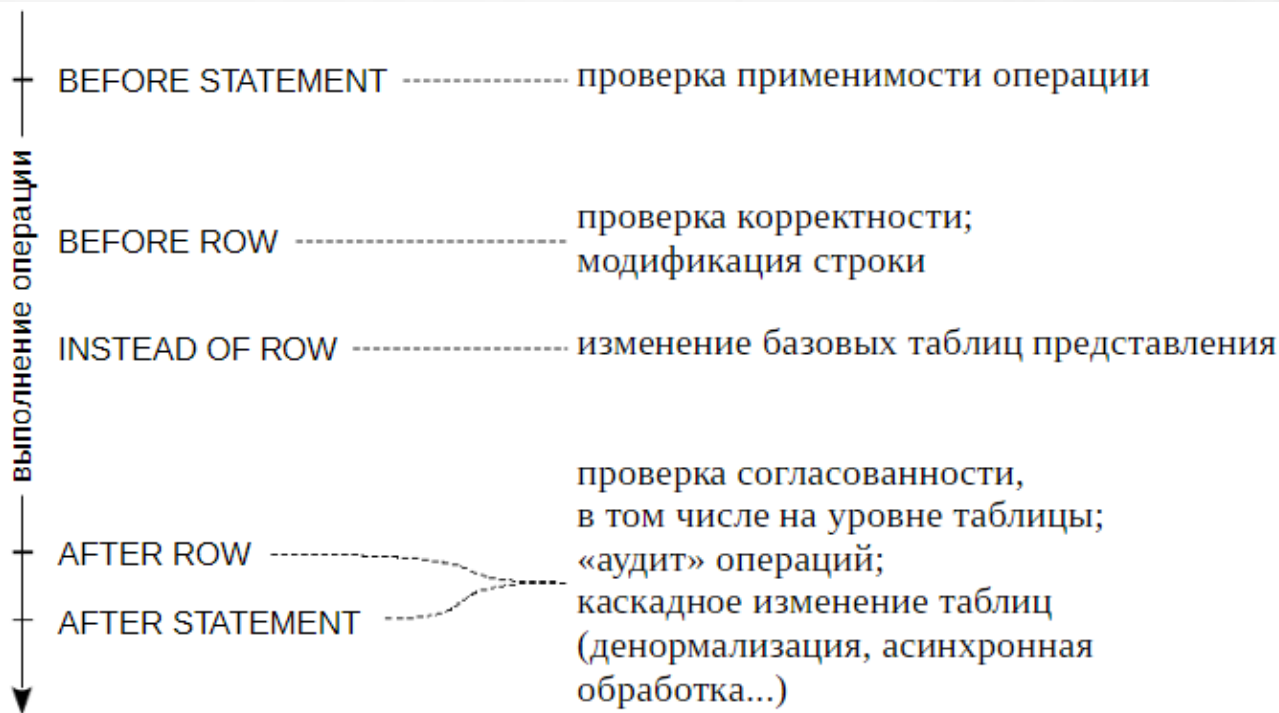
Контекст

OLD TABLE update, delete

NEW TABLE insert, update

TG-переменные





Код вызывается неявно

сложно отследить логику выполнения

Правила видимости изменчивой триггерной функции

виден результат триггеров BEFORE ROW или INSTEAD OF ROW

Порядок вызова триггеров для одного события

триггеры отрабатывают в алфавитном порядке

Не предотвращается зацикливание

триггер может вызвать срабатывание других триггеров

Можно нарушить ограничения целостности

например, исключив из обработки строки, которые должны удалиться

Событийный триггер

похож на обычный «табличный» триггер, но другой объект

Триггерная функция

соглашение: функция не принимает параметры,

возвращает значение псевдотипа `event_trigger`

для получения контекста служат специальные функции

События

`DDL_COMMAND_START` перед выполнением команды

`DDL_COMMAND_END` после выполнения команды

`TABLE_REWRITE` перед перезаписью таблицы

`SQL_DROP` после удаления объектов

Триггер — способ отреагировать на возникновение события.

С помощью триггера можно отменить операцию, изменить ее результат или выполнить дополнительные действия.

Триггер выполняется как часть транзакции;

ошибка в триггере приводит к откату транзакции.

Использование триггеров AFTER ROW и переходных таблиц удорожает обработку.

Все хорошо в меру: сложную логику трудно отлаживать из-за неявного выполнения триггеров.

1. Создайте триггер, обрабатывающий обновление поля `onhand_qty` представления `catalog_v`. Проверьте, что в «Каталоге» появилась возможность заказывать книги.
2. Обеспечьте выполнение требования согласованности: количество книг на складе не может быть отрицательным (нельзя купить книгу, которой нет в наличии). Внимательно проверьте правильность реализации, учитывая, что с приложением могут одновременно работать несколько пользователей.