



PostgreSQL

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



# PL/pgSQL Отладка



Проверки корректности

Отладчик PL/pgSQL

Служебные сообщения и различные способы их реализации

Трассировка сеансов

## Проверки времени компиляции и времени выполнения

*plpgsql.extra\_warnings*

*plpgsql.extra\_errors*

дополнительные проверки в расширении plpgsql\_check

## Проверки в коде

команда ASSERT

## Тестирование

## Состав

- расширение pldbgapi

- встроенная поддержка в некоторых графических средах

## Возможности

- установка точек прерывания

- пошаговое выполнение

- проверка и установка значений переменных

- не требуется изменение кода

- отладка работающих приложений

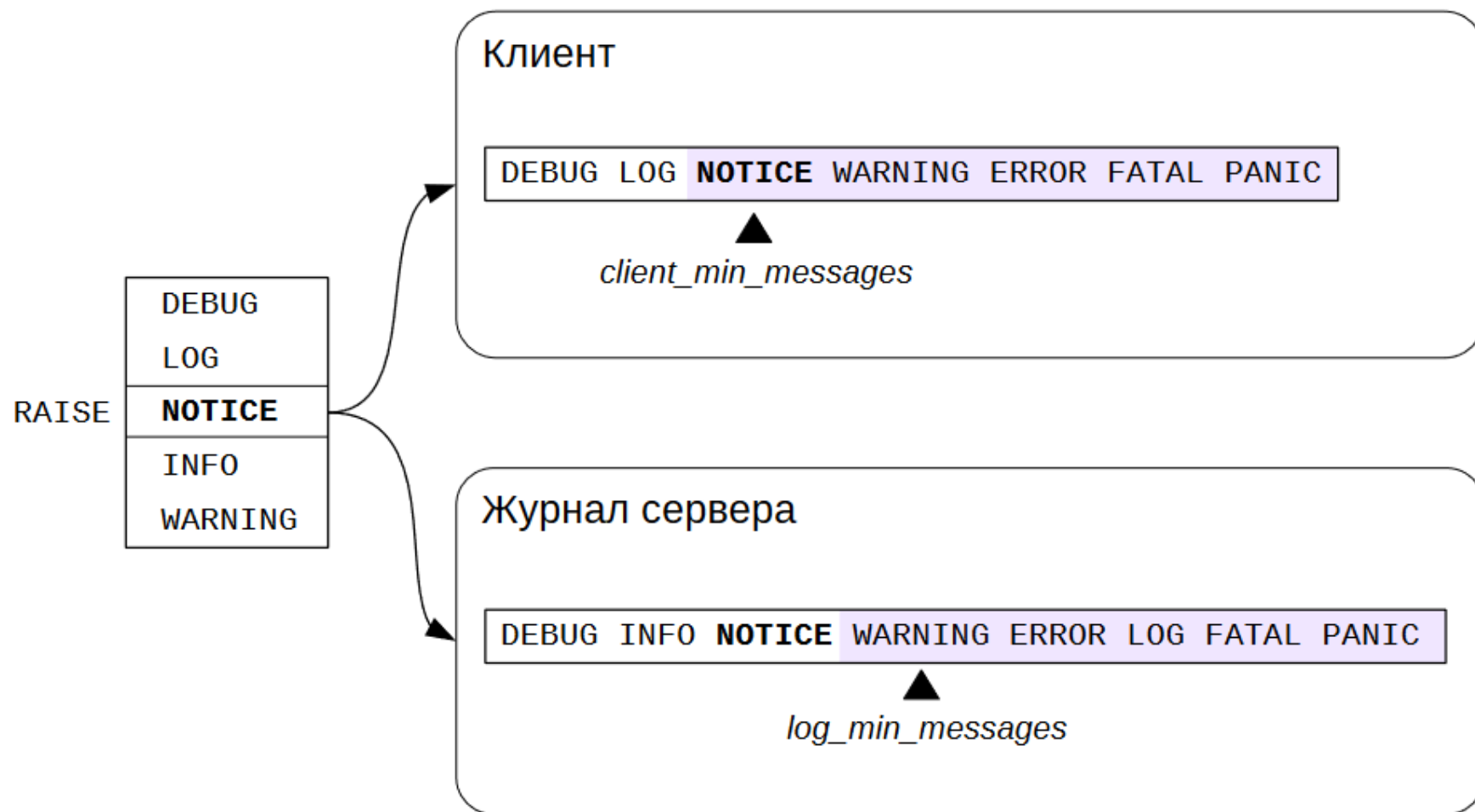
## Не только отладка кода

- мониторинг долго выполняющихся процессов
- ведение журнала приложения

## Подходы к реализации

- вывод на консоль или в журнал сервера
- запись в таблицу или в файл
- передача информации другим процессам

# Команда RAISE





# Процесс → процесс (IPC)

## NOTIFY → LISTEN

команды SQL

транзакционное выполнение, неудобно для отладки

## Статус сеанса

параметр *application\_name*

виден в представлении `pg_stat_activity` и выводе команды `ps`

можно использовать в журнальных сообщениях



## Расширение dblink

входит в состав сервера

накладные расходы на создание соединения

## Автономные транзакции

Postgres Pro Enterprise

## Расширение adminpack

входит в состав сервера

в том числе позволяет записывать текстовые файлы

## Недоверенные языки

например, PL/Perl

## Стандартная трассировка в журнал сообщений

- накладные расходы на запись в журнал
- большой размер файла журнала
- требуется инструменты для анализа
- нужен доступ к журналу (безопасность)

## Настройки

долго выполняющиеся команды	<i>log_min_duration_statement</i>
какие команды записывать	<i>log_statement</i>
контекст сообщения	<i>log_line_prefix</i>
...	

## Расширение `auto_explain`

запись в журнал планов выполнения запросов  
трассировка вложенных запросов

## Настройки

планы долгих команд  
вложенные запросы  
...

`auto_explain.log_min_duration`  
`auto_explain.log_nested_statements`

## Расширение plpgsql\_check

накладные расходы на запись сообщений  
большой объем выдачи

## Основные настройки

включение трассировки

*plpgsql\_check.enable\_tracer*

*plpgsql\_check.tracer*

уровень сообщений

*plpgsql\_check.tracer\_errlevel*

PL/pgSQL Debugger — API отладчика, используется в графических средах разработки

Служебные сообщения — вывод на консоль, запись в журнал сообщений сервера, в таблицу или в файл, передача другим процессам

Возможность трассировки сеансов



1. Измените функцию `get_catalog` так, чтобы динамически формируемый текст запроса записывался в журнал сообщений сервера.

В приложении выполните несколько раз поиск, заполняя разные поля, и убедитесь, что команды SQL формируются правильно.

2. Включите трассировку команд SQL на уровне сервера.

Поработайте в приложении и проверьте, какие команды попадают в журнал сообщений.

Выключите трассировку.



1. Включите трассировку PL/pgSQL-кода средствами расширения `plpgsql_check` и проверьте ее работу на примере нескольких подпрограмм, вызывающих одна другую.
2. При выводе отладочных сообщений из PL/pgSQL-кода важно понимать, к какой подпрограмме они относятся. В демонстрации имя функции выводилось вручную. Реализуйте функционал, автоматически добавляющий к тексту сообщений имя текущей функции или процедуры.