



PG BootCamp Russia 2025 Ekaterinburg

PGBootCamp.ru



СТАТИЧЕСКИЙ АНАЛИЗ структуры баз данных PostgreSQL против скрытых ошибок

Никонов Василий

support@dbint.ru

Вас привлекали для аудита, связанного с СУБД PostgreSQL?

Вас привлекали для аудита, связанного с СУБД PostgreSQL?

Производительность:

Вас привлекали для аудита, связанного с СУБД PostgreSQL?

Производительность:

- уже на этапе эксплуатации
- решение уже возникшей проблемы
- возможен широкий спектр проблем и решений для них (анализ запросов, тюнинг структуры и настройки СУБД, реплики и др.)

Вас привлекали для аудита, связанного с СУБД PostgreSQL?

Производительность

Настройка СУБД:

Вас привлекали для аудита, связанного с СУБД PostgreSQL?

Производительность

Настройка СУБД:

- перед началом эксплуатации
- широкий спектр возможностей (предоценка нагрузки, «калькуляторы» настроек, выбор типов реплик, пулы соединений)

Вас привлекали для аудита, связанного с СУБД PostgreSQL?

Производительность

Настройка СУБД

Настройка резервного копирования:

Вас привлекали для аудита, связанного с СУБД PostgreSQL?

Производительность

Настройка СУБД

Настройка резервного копирования:

- *желательно* перед началом эксплуатации
- не объединяем с предыдущими пунктами, т.к. админы делятся на два типа ...
- и здесь также широкий спектр возможностей (pg_dump, архивирование WAL, реплики, тестирование восстановлением)

Вас привлекали для аудита, связанного с СУБД PostgreSQL?

Производительность

Настройка СУБД

Настройка резервного копирования

Безопасность:

Вас привлекали для аудита, связанного с СУБД PostgreSQL?

Производительность

Настройка СУБД

Настройка резервного копирования

Безопасность:

- *желательно*, чтобы это был непрерывный процесс
- и здесь также широкий спектр возможностей (актуальность версий, учетные записи, ограничение доступа к серверам и к сущностям БД, сканеры уязвимостей)

Вас привлекали для аудита, связанного с СУБД PostgreSQL?

Производительность

Настройка СУБД

Настройка резервного копирования

Безопасность

Этап разработки:

Вас привлекали для аудита, связанного с СУБД PostgreSQL?

Производительность

Настройка СУБД

Настройка резервного копирования

Безопасность

Этап разработки:

- системы управления миграциями
- pgTar (для unit-тестов)
- нагрузочное тестирование

Аудит на этапах разработки или приёмки на сопровождение

Аудит на этапах разработки или приёмки на сопровождение

Системы управления миграциями:

Аудит на этапах разработки или приёмки на сопровождение

Системы управления миграциями:

- хорошо, когда они есть
 - да, встречаются проекты, где структура БД не воспроизводится миграциями
 - часто в проектах миграции «только вверх»

Аудит на этапах разработки или приёмки на сопровождение

Системы управления миграциями:

- хорошо, когда они есть
 - да, встречаются проекты, где структура БД не воспроизводится миграциями
 - часто в проектах миграции «только вверх»
- внешняя система миграций vs ORM

Аудит на этапах разработки или приёмки на сопровождение

Системы управления миграциями:

- хорошо, когда они есть
 - да, встречаются проекты, где структура БД не воспроизводится миграциями
 - часто в проектах миграции «только вверх»
- внешняя система миграций vs ORM
 - ORM, как правило, быстрый старт
 - ORM, как правило, не поддерживают всех возможностей СУБД
 - при расширении или смене стека миграции на базе ORM могут стать «гирей»

Аудит на этапах разработки или приёмки на сопровождение

Системы управления миграциями:

- хорошо, когда они есть
 - да, встречаются проекты, где структура БД не воспроизводится миграциями
 - часто в проектах миграции «только вверх»
- внешняя система миграций vs ORM
 - ORM, как правило, быстрый старт
 - ORM, как правило, не поддерживают всех возможностей СУБД
 - при расширении или смене стека миграции на базе ORM могут стать «гирей»
- разбор подходов к организации скриптов миграций
потянет на отдельный доклад

Аудит на этапах разработки или приёмки на сопровождение

Системы управления миграциями:

- хорошо, когда они есть
 - да, встречаются проекты, где структура БД не воспроизводится миграциями
 - часто в проектах миграции «только вверх»
- внешняя система миграций vs ORM
 - ORM, как правило, быстрый старт
 - ORM, как правило, не поддерживают всех возможностей СУБД
 - при расширении или смене стека миграции на базе ORM могут стать «гирей»
- разбор подходов к организации скриптов миграций потянет на отдельный доклад

Статический анализ структуры базы данных как инструмент на этапах разработки и приёмки проекта на сопровождение

Статический анализ структуры базы данных как инструмент на этапах разработки и приёмки проекта на сопровождение

Что такое статический анализ?

Статический анализ структуры базы данных как инструмент на этапах разработки и приёмки проекта на сопровождение

Что такое статический анализ?

Какие возможности предоставляет PostgreSQL?

Статический анализ структуры базы данных как инструмент на этапах разработки и приёмки проекта на сопровождение

Что такое статический анализ?

Какие возможности предоставляет PostgreSQL?

- системный каталог (pg_catalog)
- информационная схема (information_schema)
- для анализа достаточно SQL запросов

Статический анализ структуры базы данных как инструмент на этапах разработки и приёмки проекта на сопровождение

Что такое статический анализ?

Какие возможности предоставляет PostgreSQL?

- системный каталог (pg_catalog)
- информационная схема (information_schema)
- для анализа достаточно SQL запросов

Необходимы:

Статический анализ структуры базы данных как инструмент на этапах разработки и приёмки проекта на сопровождение

Что такое статический анализ?

Какие возможности предоставляет PostgreSQL?

- системный каталог (pg_catalog)
- информационная схема (information_schema)
- для анализа достаточно SQL запросов

Необходимы:

- систематизация проверок и соответствующих SQL-запросов
- упрощение интеграции в CI

Какие проверки можно выполнять?

Какие проверки можно выполнять

Структура (схема) данных:

- наличие РК или уникальных ограничений
- соответствие типов полей для FK
- совпадение (высокая схожесть) индексов

Какие проверки можно выполнять?

Структура (схема) данных:

- наличие РК или уникальных ограничений
- соответствие типов полей для FK
- совпадение (высокая схожесть) индексов

Состояние служебных объектов:

- признак, что ограничения не были применены ко всем данным
- близкое исчерпание последовательностей

Какие проверки можно выполнять?

Структура (схема) данных:

- наличие РК или уникальных ограничений
- соответствие типов полей для FK
- совпадение (высокая схожесть) индексов

Состояние служебных объектов:

- признак, что ограничения не были применены ко всем данным
- близкое исчерпание последовательностей

Эвристические проверки:

- b-tree индекс для массивов
- текстовые поля для хранения uuid данных

Пример

https://github.com/sdblist/db_verifier/blob/main/shards/c1001.sql

```
-- c1001 - constraint not validated  
SELECT * FROM pg_catalog.pg_constraint AS c  
INNER JOIN pg_catalog.pg_class AS t ON t.oid = c.conrelid  
INNER JOIN pg_catalog.pg_namespace AS n ON t.relnamespace = n.oid  
WHERE  
    c.contype IN ('c', 'f')  
    AND (NOT c.convalidated)
```

Пример

https://github.com/sdblist/db_verifier/blob/main/shards/fk1007.sql

```
-- fk1007 - not involved in foreign keys
SELECT * FROM pg_catalog.pg_class AS t
INNER JOIN pg_catalog.pg_namespace AS n ON t.relnamespace = n.oid
WHERE
    t.relkind IN ('r', 'p')
    AND n.nspname NOT IN ('information_schema', 'pg_catalog')
    AND t.oid NOT IN (SELECT conrelid FROM pg_catalog.pg_constraint WHERE contype IN ('f'))
    AND t.oid NOT IN (SELECT confrelid FROM pg_catalog.pg_constraint WHERE contype IN ('f'))
```

Пример

https://github.com/sdblist/db_verifier/blob/main/shards/fk1001.sql

```
-- fk1001 - fk uses mismatched types
SELECT * FROM pg_catalog.pg_constraint AS cfk
CROSS JOIN LATERAL UNNEST(cfk.conkey) WITH ORDINALITY AS cfk_conkey(conkey_number, conkey_order)
LEFT JOIN LATERAL UNNEST(cfk.confkey) WITH ORDINALITY AS cfk_confkey(confkey_number, confkey_order)
    ON cfk_conkey.conkey_order = cfk_confkey.confkey_order
LEFT JOIN pg_catalog.pg_attribute AS rel_att
    ON rel_att.attrelid = cfk.conrelid AND rel_att.attnum = cfk_conkey.conkey_number
LEFT JOIN pg_catalog.pg_attribute AS frel_att
    ON frel_att.attrelid = cfk.confrelid AND frel_att.attnum = cfk_confkey.confkey_number
WHERE cfk.contype IN ('f')

((rel_att.atttypid <> frel_att.atttypid) OR (rel_att.atttypmod <> frel_att.atttypmod))
```


Проект db_verifier для статического для анализа структуры БД

Проект db_verifier для статического для анализа структуры БД

Цели:

- систематизировать и упростить рутинные проверки
- автоматизировать проверки, упростить их интеграцию в CI
- популяризировать использование статического анализа структур БД, чтобы он стал обыденным инструментом

Проект db_verifier для статического для анализа структуры БД

Цели:

- систематизировать и упростить рутинные проверки
- автоматизировать проверки, упростить их интеграцию в CI
- популяризировать использование статического анализа структур БД, чтобы он стал обыденным инструментом

Приоритеты:

- только SQL (не нужны расширения или PLSql)
- гибкость и настраиваемость
- безопасность для пользователя (весь код можно прочитать, не нужны пользовательские данные, применимо для БД без данных)

Проект `db_verifier` для статического для анализа структуры БД

Коллекция проверок в формате SQL:

- `db_verifier.sql` – скрипт «всё в одном», все проверки в одном файле для снижения порога для начала использования и упрощения встраивания проверок в CI
- `shards` – каталог, где те же проверки в отдельных файлах

Проект `db_verifier` для статического для анализа структуры БД

Коллекция проверок в формате SQL:

- `db_verifier.sql` – скрипт «всё в одном», все проверки в одном файле для снижения порога для начала использования и упрощения встраивания проверок в CI
- `shards` – каталог, где те же проверки в отдельных файлах

Все проверки:

- «покрашены» по уровню
- снабжены описанием (RU / EN)

Проект `db_verifier` для статического для анализа структуры БД

Коллекция проверок в формате SQL:

- `db_verifier.sql` – скрипт «всё в одном», все проверки в одном файле для снижения порога для начала использования и упрощения встраивания проверок в CI
- `shards` – каталог, где те же проверки в отдельных файлах

Все проверки:

- «покрашены» по уровню
- снабжены описанием (RU / EN)

Набор тестов для скриптов, их проверка на PG с 12 по 17

Проект `db_verifier` для статического для анализа структуры БД

«Грубая» интеграция в CI:

- `db_verifier.sql` оборачивается в `examples/cumulative_score.sql`
- при первом запуске необходимо получить пороговое значение
- скрипт вызывается (при помощи `plsql` или системы миграции) после применения изменений или по расписанию
- результат каждой проверки на основе уровня критичности, получает числовую оценку, оценки суммируются в одно числовое значение
- полученное числовое значение сравнивается с ранее заданной константой (получают при первом запуске)
- `alarm`, если превышено пороговое значение

https://github.com/sdblist/db_verifier/blob/main/examples/cumulative_score.sql

```
SELECT
    COALESCE(SUM(cumulative_score_value), 0) AS cumulative_score
FROM (
    VALUES
        ('critical', 55),
        ('error', 25),
        ('warning', 12),
        ('notice', 3)
    ) AS t(check_level, cumulative_score_value)
    INNER JOIN (
-- >>> db_verifier
) AS r ON t.check_level = r.check_level
;
```


Проект `db_verifier` для статического для анализа структуры БД

Интеграция в CI с полноценным baseline:

- `db_verifier.sql` объединяется с `examples/where.sql`
- при первом запуске фиксируем срабатывания, добавляем их в блок `WHERE`, чтобы заглушить как ранее известные
- скрипт вызывается (при помощи `plsql` или системы миграции) после применения изменений или по расписанию
- `alarm`, если проверка вывела хотя бы одну строку, т.к. все известные ошибки заглушены и появилось что-то новое

Формат результата

Колонка	Описание
object_id	id (oid) объекта в соответствующей системной таблице
object_name	наименование объекта, в некоторых случаях со схемой
object_type	тип проверяемого объекта (relation, constraint, index, sequence, attribute)
check_code	код проверки (сейчас более 30 проверок)
check_level	уровень важности/критичности результата
check_name	наименование проверки
check_result_json	подробные результаты проверки в формате json содержит все перечисленные выше данные, дополнительное описание в зависимости от выбранного языка, дополнительные данные, зависящие от конкретной проверки



PG BootCamp Russia 2025 Ekaterinburg

PGBootCamp.ru



Ваши вопросы?

Никонов Василий

support@dbint.ru

https://github.com/sdblist/db_verifier

