

FIREBIRD  CONF 2025

Дмитрий Еманов

Firebird и Ред База Данных:

приоритеты развития и
резервы производительности

Сопровождение

Старые версии

- Релизы 3.0.12 и 4.0.5 выпущены в августе 2024
- В 3.0 — только багфиксы
- В 4.0 — десяток мелких улучшений плюс багфиксы
- Свежие поинт-релизы ожидаются летом
- Возможно для Firebird 3.0 он будет последним
- Ред База Данных 3.0 — поддерживается до конца 2025

Firebird 5.0

- Релиз 5.0.2 выпущен в феврале 2025
- Улучшен сетевой протокол в плане выборки блобов
- Оптимизирован алгоритм выборки ORDER для NULL-ключей
- Более точная оценка кардинальности таблиц
- Другие мелкие улучшения плюс около полусотни багфиксов
- Релиз 5.0.3 планируется летом

Ред База Данных 5.0

- Релиз 5.0.5 выпущен в апреле 2025
- Исправления JSON и оптимизатора, а также репликации
- Фиксы улучшения надежности

+ изменения из Firebird 5.0.x

Статус разработки

Firebird 6.0 : прогресс

- Общий кеш метаданных — на финишной прямой
- Поддержка схем по SQL-стандарту — PR на ревью
- Поддержка табличных пространств — PR на ревью
- JSON-функции по SQL-стандарту — на доработке
- Тип данных ROW — подготовка PR

Firebird 6.0 : прогресс

- Встроенная функция UNLIST
- Поддержка блобов более 4ГБ
- Стоимостная оценка SORT vs ORDER планов
- Недесятичные литералы — на ревью
- Стандартная функция LISTAGG — подготовка PR
- Гетерогенные запросы — подготовка PR
- Локальные временные таблицы — дискуссия

Firebird 6.0 : nporpccc

- Roadmap:

<https://www.firebirdsql.org/en/roadmap-v6/>

Ред База Данных 6.0 : прогресс

- *Тип данных ROW*
- *Встроенная функция UNLIST*
- *Стоимостная оценка SORT vs ORDER планов*
- *Недесятичные литералы*
- *Стандартная функция LISTAGG*
- *Гетерогенные запросы*
- *Пакетные константы*

Ред База Данных 6.0 : прогресс

- Геометрические типы данных
- OpenGIS и SQL/MM (Part 3: Spatial)
- GEOMETRY / GEOGRAPHY
- POINT, LINESTRING, POLYGON, MULTIPOINT, ...
- Реализовано поверх блобов
- Пока без индексов, варианты —
R-TREE либо B-TREE с Hilbert curve / Z-order curve

Ред База Данных 6.0 : прогресс

- **Партиционирование**
- RANGE — синтаксис и функция разбиения
- Поддержка глобальных индексов (включая РК, УК, FK)
- Поддержка локальных индексов, РК (если создан по полю разбиения), УК и FK (в пределах партии)
- Обновление значения поля разбиения записи с перемещением в другую партицию (с обновлением индексов)
- Поддержка партиций в DBKEY / MAKE_DBKEY

Ред База Данных 6.0 : прогресс

- Поддержка табличных пространств в pbackup
- Расширение JSON-функционала
- Статистика для временных файлов
- Улучшения производительности

Производительность

Оптимизатор запросов

- Максимально влияет на время выполнения селектов
- Всегда являлся миксом стоимостного (*inner join*) и эвристического (все остальное)
- Недавно были добавлены оценки стоимости: *hash vs nested loop, hash vs merge, order vs sort*
- Также добавлена сквозная оценка кардинальности (местами наивная)
- Реализованы первые методы трансформации запросов

Оптимизатор запросов : недостатки

- Упрощенный учет селективности сложных предикатов
- Весьма грубая оценка кардинальности таблиц
- Нет возможности считать стоимость подзапросов до их оптимизации
- Нет гистограмм распределения значений
- Не хватает хранимой статистики и ее автоматического обновления
- Смешанные джойны не переупорядочиваются

Методы доступа к данным

- Читать свежую редакцию одноименной статьи :)
- HASH/MERGE OUTER JOIN — реализованы алгоритмически, но пока недоступны со стороны оптимизатора
- Реализовали TOP-сортировку (*РБД 6.0*)
- HASH-группировка — следующая на очереди (*РБД 6.0*)
- Профилирование и оптимизация существующих алгоритмов — индексный скан, сортировка, хеширование
- Index-only scan?

Параллельность

- Сделали бекап/рестор, CREATE INDEX, sweep
- Также валидация и gstat (РБД)
- Не все сделано идеально, есть пути к улучшению
- На очереди методы доступа
- В первую очередь фуллскан, но и для других алгоритмов есть варианты

Работа с диском

- Нет префетча страниц — неэффективность работы с O_DIRECT
- Хотелось бы еще многоблочное чтение (экстентами)
- Есть патч от Влада Хорсуна, но не все пока замечательно
- Самое главное — I/O не в топе профайлера
(TPC-C: 15/30%, фуллскан: 17%)
- Есть смысл работать, но сначала надо разобраться с другими проблемами

Асинхронный I/O

- Оптимизация для *careful writes*
- Позволяет отвязать *bulk load* (рестор в том числе) от диска
- Можно использовать также и для префетча
- Реализации: *aio / io_uring* на Linux, *IOCP* на Windows
- В рамках РБД 6.0 реализован асинхронный диспетчер на основе *io_uring*, тестируем (время сброса кеша < в ~2 раза)

Борьба с legacy

- Неэффективный API для больших выборок
- Высокая стоимость входа в движок
- Излишние копирования и преобразования входных и выходных параметров
- На скорую руку: скорость фетча $389s \rightarrow 301s = 27\%$
(и еще есть резервы)

Вопросы



29 МАЯ | МОСКВА

FIREBIRD  CONF

2025