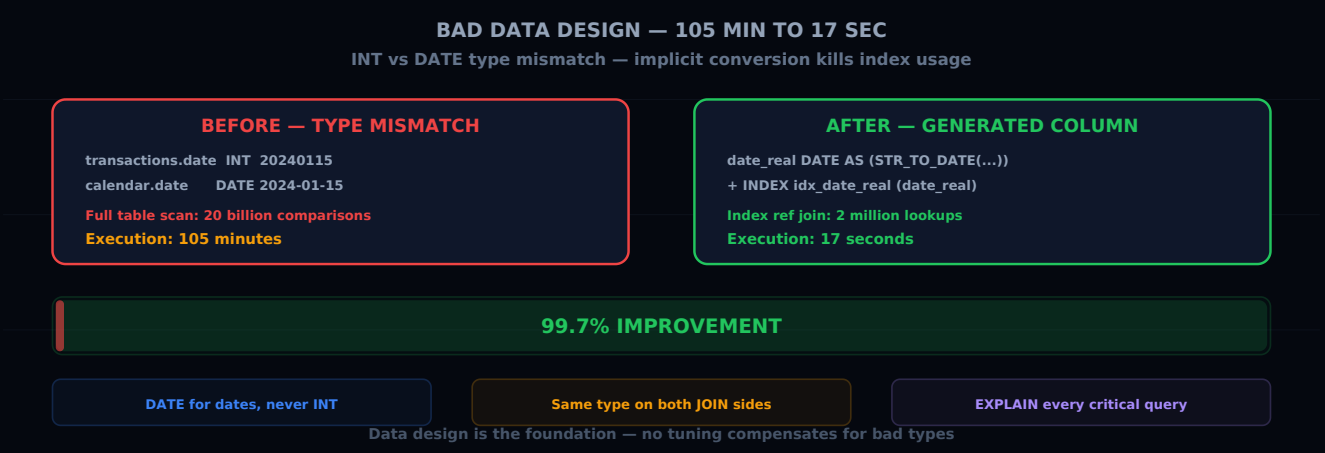
Плохой дизайн данных ведёт к плохой производительности: от 105 минут до 17 секунд

Sylvain ARBAUDIE · July 23, 2025

MARIADB PERFORMANCE OPTIMIZATION DATA-DESIGN



Симптом: 105 минут на один запрос

Клиент звонит в панике. Их ночной batch-процесс, формирующий ежедневные отчёты, занимает всё больше времени. То, что год назад выполнялось за 10 минут, теперь длится **105 минут**. Объём данных, конечно, вырос, но не настолько, чтобы объяснить десятикратное увеличение времени выполнения.

Проблемный запрос — классический `JOIN` между таблицей транзакций и таблицей календаря:

```
SELECT
    t.transaction_id,
    t.amount,
    t.transaction_date,
    c.fiscal_year,
    c.fiscal_quarter
FROM transactions t
JOIN calendar c ON t.transaction_date = c.calendar_date
WHERE t.created_at >= '2024-01-01';
```

Ничего примечательного на первый взгляд. Две таблицы, соединение по дате, временной фильтр. И тем не менее, 105 минут.

Диагноз: несоответствие типов

Анализ плана выполнения (`EXPLAIN`) выявляет full table scan по таблице `calendar` . Странно для соединения по тому, что должно быть первичным ключом.

При рассмотрении структур таблиц проблема бросается в глаза:

```
-- Таблица transactions
CREATE TABLE transactions (
    transaction_id BIGINT AUTO_INCREMENT PRIMARY KEY,
    amount DECIMAL(10,2),
    transaction_date INT NOT NULL,  -- ← хранится как YYYYMMDD
    created_at DATETIME
);

-- Таблица calendar
CREATE TABLE calendar (
    calendar_date DATE NOT NULL PRIMARY KEY,
    fiscal_year SMALLINT,
    fiscal_quarter TINYINT
);
```

Столбец `transaction_date` в таблице `transactions` — это `INT` , хранящий дату в формате `YYYYMMDD` (например, `20240115` для 15 января 2024). Столбец `calendar_date` в таблице `calendar` — настоящий `DATE` .

Когда MariaDB / MySQL выполняет `JOIN` , движку приходится сравнивать `INT` с `DATE` . Для каждой строки из `transactions` он неявно преобразует `DATE` в `INT` (или наоборот) для каждой строки `calendar` . Это неявное преобразование делает индекс по `calendar_date` неиспользуемым. Результат: full table scan по `calendar` для каждой строки из `transactions` .

При 2 миллионах транзакций и 10 000 строках в `calendar` это порождает **20 миллиардов сравнений** с преобразованием типа.

Почему нельзя просто изменить тип?

Очевидный ответ — преобразовать столбец `transaction_date` из `INT` в `DATE`. Но в реальности production-систем:

- Таблица занимает 15 ГБ. `ALTER TABLE` займёт часы и заблокирует таблицу.
- 47 хранимых процедур и 12 представлений ссылаются на `transaction_date` как на `INT`.
- PHP-приложение использует арифметические сравнения по этому столбцу (`WHERE transaction_date > 20240101`).
- ETL-процесс загрузки передаёт даты в формате `INT` из legacy-системы.

Изменение типа — правильное решение в долгосрочной перспективе, но не то немедленное решение, которое нужно клиенту сегодня вечером.

Решение: виртуальный генерируемый столбец

MariaDB / MySQL поддерживает виртуальные столбцы (generated columns). Это столбцы, вычисляемые динамически из других столбцов, без физического хранения (VIRTUAL) или с хранением (STORED).

```
ALTER TABLE transactions
ADD COLUMN transaction_date_real DATE AS (
    STR_TO_DATE(CAST(transaction_date AS CHAR(8)), '%Y%m%d')
) VIRTUAL;
```

Этот столбец преобразует `INT` в `DATE` на лету. Но один лишь виртуальный столбец не решает проблему производительности. Нужен индекс:

```
ALTER TABLE transactions
ADD COLUMN transaction_date_real DATE AS (
    STR_TO_DATE(CAST(transaction_date AS CHAR(8)), '%Y%m%d')
) STORED,
ADD INDEX idx_transaction_date_real (transaction_date_real);
```

Мы используем `STORED` вместо `VIRTUAL`, чтобы иметь возможность создать индекс. Столбец физически хранится, и индекс поддерживается автоматически при вставках и обновлениях.

Исправленный запрос

```
SELECT
    t.transaction_id,
    t.amount,
    t.transaction_date,
    c.fiscal_year,
    c.fiscal_quarter
FROM transactions t
JOIN calendar c ON t.transaction_date_real = c.calendar_date
WHERE t.created_at >= '2024-01-01';
```

`JOIN` теперь сравнивает `DATE` с `DATE`. Индекс используется. План выполнения показывает `ref` вместо full scan.

Результат: 17 секунд

Метрика	До	После	Улучшение
Время выполнения	105 мин	17 сек	99,7%
Просмотренных строк	~20 млрд	~2 млн	99,99%
Тип сканирования	Full scan	Index ref	—

Со 105 минут до 17 секунд. Улучшение на **99,7%** без изменения существующей схемы, без модификации приложения, без правок хранимых процедур.

Почему неявные преобразования — это ловушка

Этот случай иллюстрирует фундаментальную проблему: неявные преобразования типов в JOIN-ах и условиях WHERE — это тихие убийцы производительности.

MariaDB / MySQL выполняет неявные преобразования во множестве случаев:

- `INT` сравнивается с `VARCHAR`: INT преобразуется в VARCHAR
- `INT` сравнивается с `DATE`: DATE преобразуется в число
- `VARCHAR(utf8)` сравнивается с `VARCHAR(latin1)`: преобразование charset
- `DECIMAL` сравнивается с `FLOAT`: преобразование в число с плавающей запятой

В каждом случае преобразование делает индекс неиспользуемым, поскольку движок не может выполнить прямой поиск в B-tree индексе, если значение сначала нужно преобразовать.

Урок: дизайн данных — это фундамент

Производительность базы данных определяется на этапе проектирования, а не на этапе тюнинга. Никакой индекс, никакая конфигурация buffer pool, никакое оборудование не компенсируют плохой выбор типа данных.

Фундаментальные правила:

1. **Дата должна храниться как `DATE` или `DATETIME`**, никогда как `INT` или `VARCHAR`.
2. **Столбцы соединения должны иметь одинаковый тип и одинаковый charset/collation.**
3. **Используйте `EXPLAIN` систематически**, чтобы проверять, что соединения используют индексы.
4. **Отслеживайте неявные преобразования** с помощью `EXPLAIN ANALYZE` (MariaDB 10.1+).

Дизайн данных — это не гламурно. Это не так захватывающе, как тюнинг системных переменных или развёртывание кластера Galera. Но это фундамент. И когда фундамент плохой, всё остальное рушится — по 105 минут за раз.

Эта статья была первоначально опубликована на Medium.