

Храним сложные объекты просто

Банальности, проверенные опытом

О себе



Когда-то MS SQL Dev

Сейчас Java архитектор
платежные системы, букмекерские
системы, b2b, *вот это вот все*

Oracle, DB2, PostgreSQL

О проекте

Платежная система (аналог Яндекс.Денег)

Год в бою

Не очень большая нагрузка

Не очень большие объемы

О нас

Небольшая команда

Java-разработчики
не очень умеющие СУБД

Быстро меняющаяся постановка

Проблема

Много сущностей

Пользователи

Платежи

Контексты платежа

около десятка разных

Контексты регистрации

около десятка

Персональные данные

очень много сущностей

Провайдеры

Контрагенты

Проводки

Интеграции

Нотификации

И довольно сложных сущностей

[illegible]

контекст платежа

И довольно сложных сущностей

Много таблиц и связей

[illegible]

контекст платежа

И довольно сложных сущностей

Много таблиц и связей

Сложно менять структуру

[illegible]

контекст платежа

Безопасно?

```
alter table TEST  
  add column NULLABLECOLUMN varchar;
```

Безопасно?

```
alter table TEST  
  add column NULLABLECOLUMN varchar;
```

В некоторых сценариях может полностью блокировать работу с таблицей TEST на долгое время.

www.facebook.com/groups/postgresql/permalink/506304452899640/

И довольно сложных сущностей

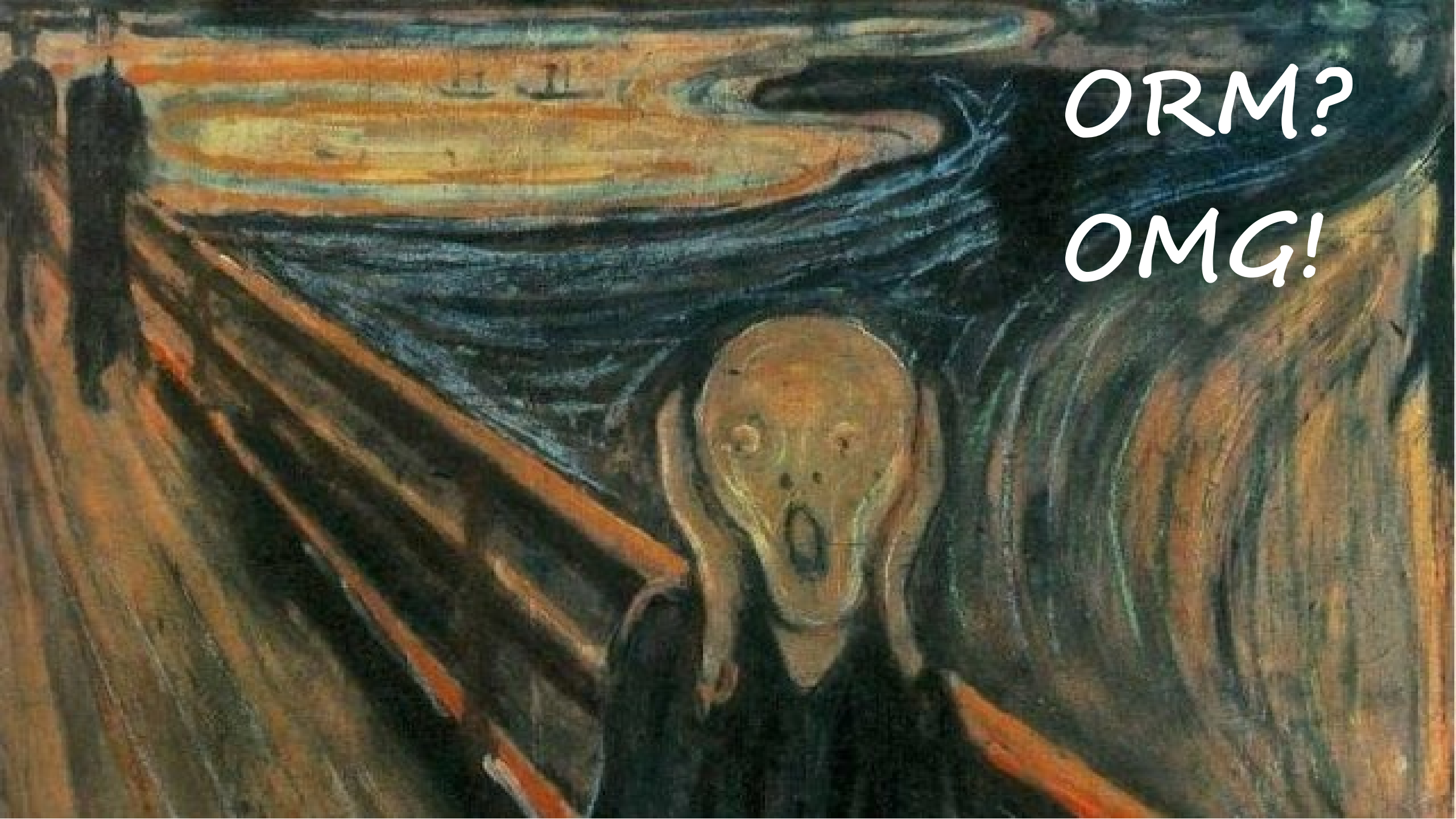
Много таблиц и связей

Сложно менять структуру

Без ORM не выжить

[illegible]

контекст платежа

A painting featuring a central figure with a pale, skull-like face and wide, staring eyes, appearing to be in a state of shock or distress. The figure is positioned in the lower center, possibly within a boat or a small enclosure, with dark, swirling lines around them. The background is composed of horizontal bands of color, including shades of blue, green, and brown, suggesting a landscape or a sky. The overall style is expressive and somewhat abstract, with visible brushstrokes and a somber, dramatic atmosphere.

ORM?
OMG!

Хочется как-нибудь иначе!

А что нам нужно?

Прочитать сложную сущность из БД

Сохранить ее изменения

Найти сущность по каким-то полям
обычно РК и что-то еще

Например, так

```
create table operations (  
  id bigint not null primary key,  
  userId bigint,  
  state varchar,  
  data text );
```


Что сделали?

Храним сложные сущности внутри записей
как-то их при этом сериализуем
например в json

Придумываем, как быстро искать сущности
в отдельные поля выделяем то, где ищем

Что сделали?

Храним сложные сущности внутри записей
как-то их при этом сериализуем
например в json

Придумываем, как быстро искать сущности
в отдельные поля выделяем то, где ищем

И даже не нарушили 1НФ ;)

Что это дает

Простота

Что это дает

Простота

Удобство изменения

Что это дает

Простота

Удобство изменения

Производительность

Что это дает

Простота

Удобство изменения

Производительность

Атомарность

Как хранить

::json валидация, расширение SQL для доступа
когда нужно хранить json, но не надо с ним ничего делать

Как хранить

- ::json валидация, расширение SQL для доступа
когда нужно хранить json, но не надо с ним ничего делать
- ::jsonb эффективное бинарное хранение и доступ
когда мы планируем много работать с внутренностями

Как хранить

`::json` валидация, расширение SQL для доступа
когда нужно хранить json, но не надо с ним ничего делать

`::jsonb` эффективное бинарное хранение и доступ
когда мы планируем много работать с внутренностями

`::text` синоним для `varchar`
когда мы не уверены, что хранить будем json

Как хранить

`::json` валидация, расширение SQL для доступа
когда нужно хранить json, но не надо с ним ничего делать

`::jsonb` эффективное бинарное хранение и доступ
когда мы планируем много работать с внутренностями

`::text` синоним для `varchar`
когда мы не уверены, что хранить будем json

`::bytea` просто храним байтики
шифруемся или используем свой формат сериализации

Чем плохо

Чем плохо

Не поддерживается ОРМами

Чем плохо

Не поддерживается ОРМами
хотя может быть это как раз и хорошо

Чем плохо

Не поддерживается ОРМами

Сложность для построения отчетов

Отчеты: быстро

::json / ::jsonb

```
select id,  
       data->'amount'->>'amount',  
       data->'amount'->>'currency'  
from operations;
```

Отчеты: быстро

::json / ::jsonb

```
select id,  
       data->'amount'->>'amount',  
       data->'amount'->>'currency'  
from operations;
```

Лучше так не делать

Отчеты: а как еще?

Не надо смешивать OLTP и OLAP/Reporting

Выгружайте данные в BI-системы в удобном виде и
делайте с ними что хотите

Чем плохо

Не поддерживается ОРМами

Сложность для построения отчетов

Поддержка целостности

Все это просто, но ...

Подстилаем соломку...

Храним номер версии сериализации
он точно будет быстро расти

Как это выглядит в БД

```
create table operations (  
  id bigint not null primary key,  
  userId bigint,  
  state varchar,  
  schema bigint not null,  
  data json );
```

Подстилаем соломку...

Храним номер версии сериализации
он точно будет быстро расти

Прописываем приоритет индексных полей над содержимым
или не сериализуем лишнее

Как это выглядит в БД

```
create table operations (  
  id bigint not null primary key,  
  userId bigint,  
  state varchar,  
  schema bigint not null,  
  data json );
```

Подстилаем соломку...

Храним номер версии сериализации
он точно будет быстро расти

Прописываем приоритет индексных полей над содержимым
или не сериализуем лишнее

Заранее готовим код для обновления версий
он точно пригодится

Подстилаем соломку...

Храним номер версии сериализации
он точно будет быстро расти

Прописываем приоритет индексных полей над содержимым
или не сериализуем лишнее

Заранее готовим код для обновления версий
он точно пригодится

Учим десериализатор поддерживать конфликты
а лучше бы и полиморфизм

Jackson (Java world)

Управляется аннотациями

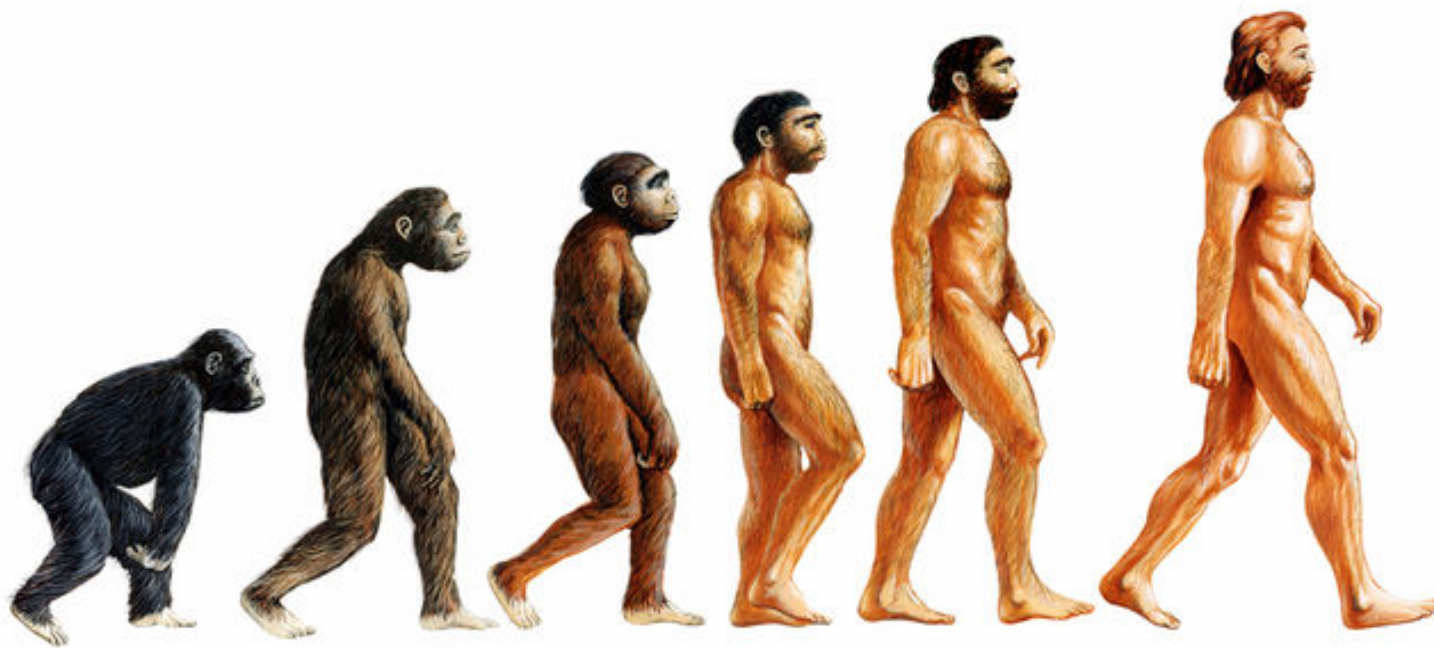
Работает с приватными полями

Можно фильтровать, какие атрибуты сериализовать

Умеет «досериализовывать» уже созданные объекты

Умеет полиморфизм

Изменения в структуре



Изменения в структуре

Обычно ничего делать не надо

Изменения в структуре

Обычно ничего делать не надо
но лучше не хранить в БД совсем старые структуры
для обновления нужно поле `schema` и логика миграции

Изменения в структуре

Обычно ничего делать не надо

Иногда нужно писать специальный код десериализации

Изменения в структуре

Обычно ничего делать не надо

Иногда нужно писать специальный код десериализации

`@JsonAnySetter`

`@JsonSerialize/@JsonDeserialize`

Изменения в структуре

Обычно ничего делать не надо

Иногда нужно писать специальный код

Иногда надо хранить старые «версии» классов

Изменения в структуре

Обычно ничего делать не надо

Иногда нужно писать специальный код

Иногда надо хранить старые «версии» классов

```
if (schemaVersion==1) {  
    OldOperations oldo = mapper.readValue(data,OldOperations.class);  
    Operations op = Operations.renew(oldo);  
    return op;  
}
```

Бонусы

Оптимистические блокировки

Бонусы

Оптимистические блокировки
добавим поле с версией сущности

Бонусы

Оптимистические блокировки

История изменений

Бонусы

Оптимистические блокировки

История изменений

добавим поле с версией сущности

Бонусы

Оптимистические блокировки

История изменений

Шифрование

Бонусы

Оптимистические блокировки

История изменений

Шифрование

вот тут и пригодится `bytea`

Почему PostgreSQL?

Почему PostgreSQL?

TOAST

The Oversized-Attribute Storage Technique

Почему PostgreSQL?

TOAST

::json

и все богатство операций с JSON

Почему PostgreSQL?

TOAST

::json

Бесплатный и с поддержкой

Итого

Из PostgreSQL получается прекрасная document-oriented СУБД

Итого

Из PostgreSQL получается прекрасная document-oriented СУБД
но только надежная, транзакционная и популярная

Итого

Из PostgreSQL получается прекрасная document-oriented СУБД

Разумный баланс облегчает жизнь

Итого

Из PostgreSQL получается прекрасная document-oriented СУБД

Разумный баланс облегчает жизнь

между СУБД и уровнем приложения

между нормализацией и денормализацией

между OLTP и OLAP

Итого

Из PostgreSQL получается прекрасная document-oriented СУБД

Разумный баланс облегчает жизнь

Запросы на N страниц все равно будут

Итого

Из PostgreSQL получается прекрасная document-oriented СУБД

Разумный баланс облегчает жизнь

Запросы на N страниц все равно будут
если не в коде, то при поддержке

Вопросы и контакты

Дельгадо Филипп,

dph.main@gmail.com

phd@itasystems.ru

vk.com/dphil

ИТИС Интерактивные
Технологии
и Системы

<http://itasystems.ru/>

