



Опыт успешной миграции НСИ Строительного Комплекса Московской Государственной Экспертизы с Oracle на Postgres Pro

Сергей Прохоров, ООО Айбим, Технический директор



План презентации

- Немного о себе
- Коротко о заказчике
- О проекте миграции
- С чем столкнулись в процессе миграции
- О организационных и технических трудностях
- Результаты проекта
- Вопросы и ответы

О СЕБЕ

Сергей Прохоров



**Технический директор
«Айбим»**



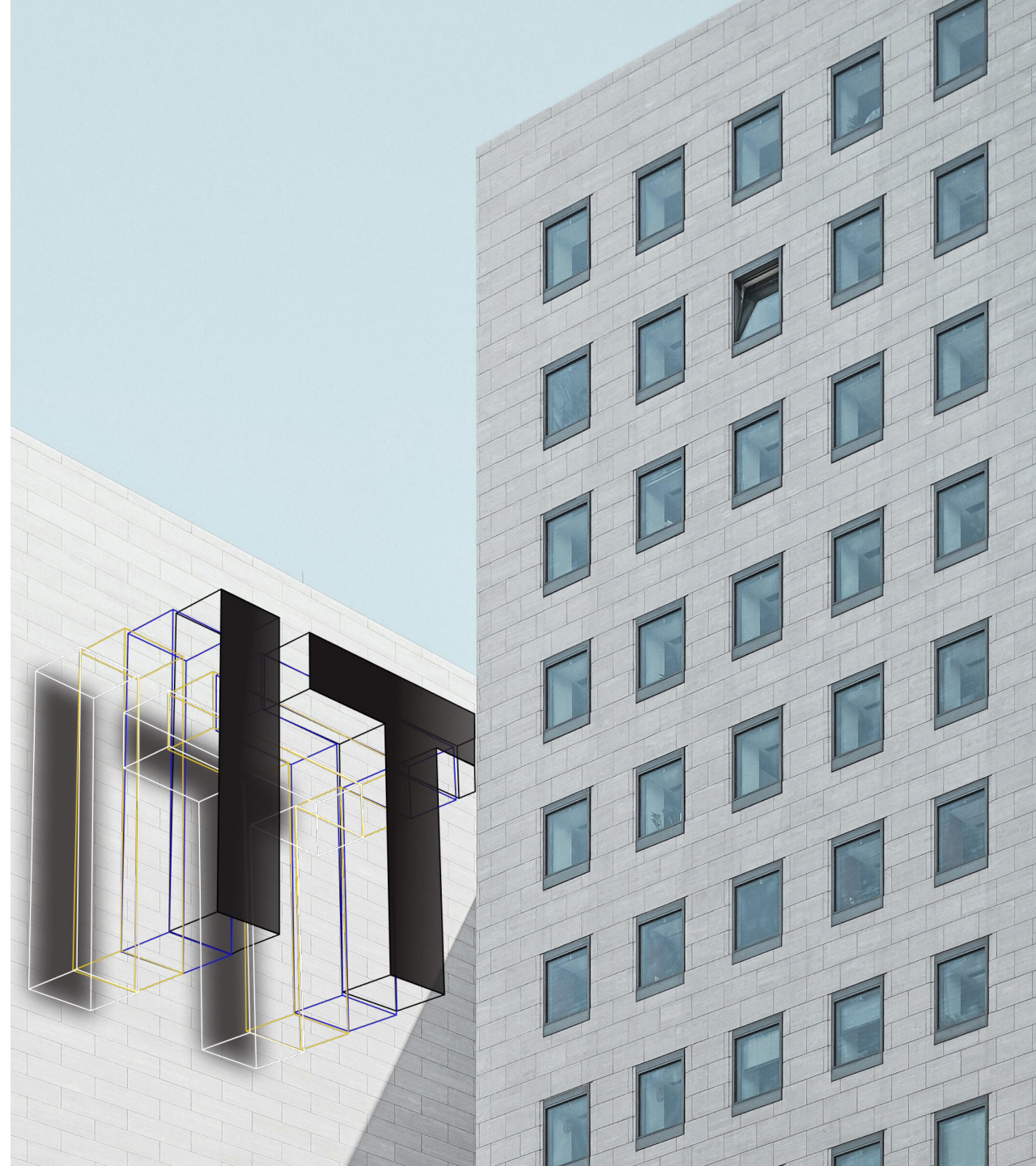
Предыдущий опыт



Архитектор хранилищ данных
и BI систем



Presale Senior Consultant
Consulting Senior Consultant





RBI
THE ART
OF REAL ESTATE



АТОМЭНЕРГОМАШ
РОСАТОМ



О КОМПАНИИ

Команда

90 профессионалов

Эксперты в своей области

Портфолио

200+ проектов

Клиенты

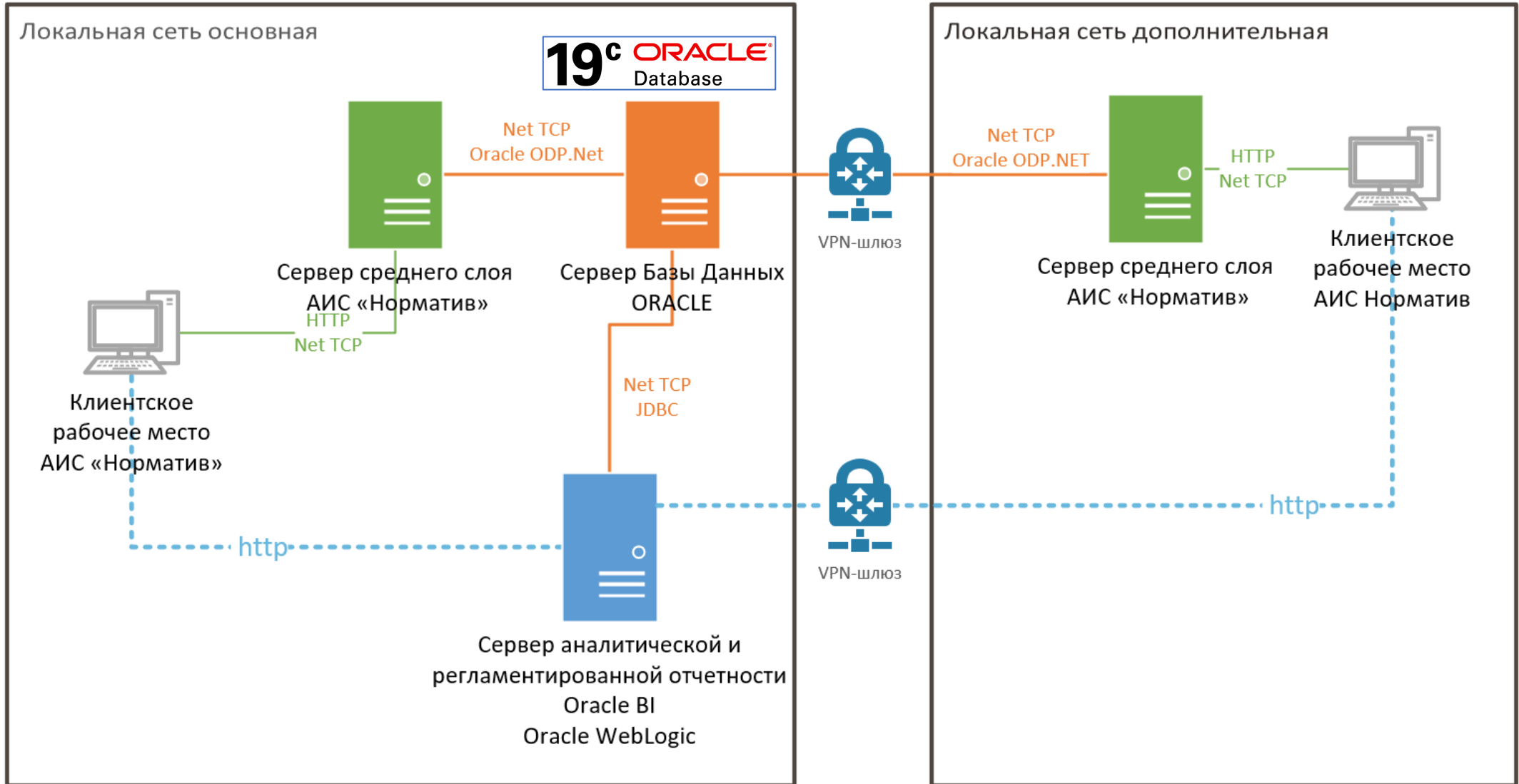
100 компаний

Из различных областей: от девелопмента до медицины

АИС “Норматив” – автоматизация расчетов стоимости строительства и публикации нормативных баз данных

- СУБД: Oracle Database 19c Enterprise Edition
- Объем БД: ~ 100GB
- Бизнес-логика реализована на PL/SQL в виде пакетов БД
- Объем кода: 11 пакетов, 152 функций и процедур, общий объем кода (включая процедуры триггеры и view) ~ 20 тыс. строк
- Количество таблиц: ~100
- Количество View: 20
- Интенсивно используются вычисляемые поля в таблицах (Virtual Columns)
- На основе Oracle BI реализовано и использует до 40 отчетов, из них 10% высокой сложности, 20% средней сложности и 70% стандартные

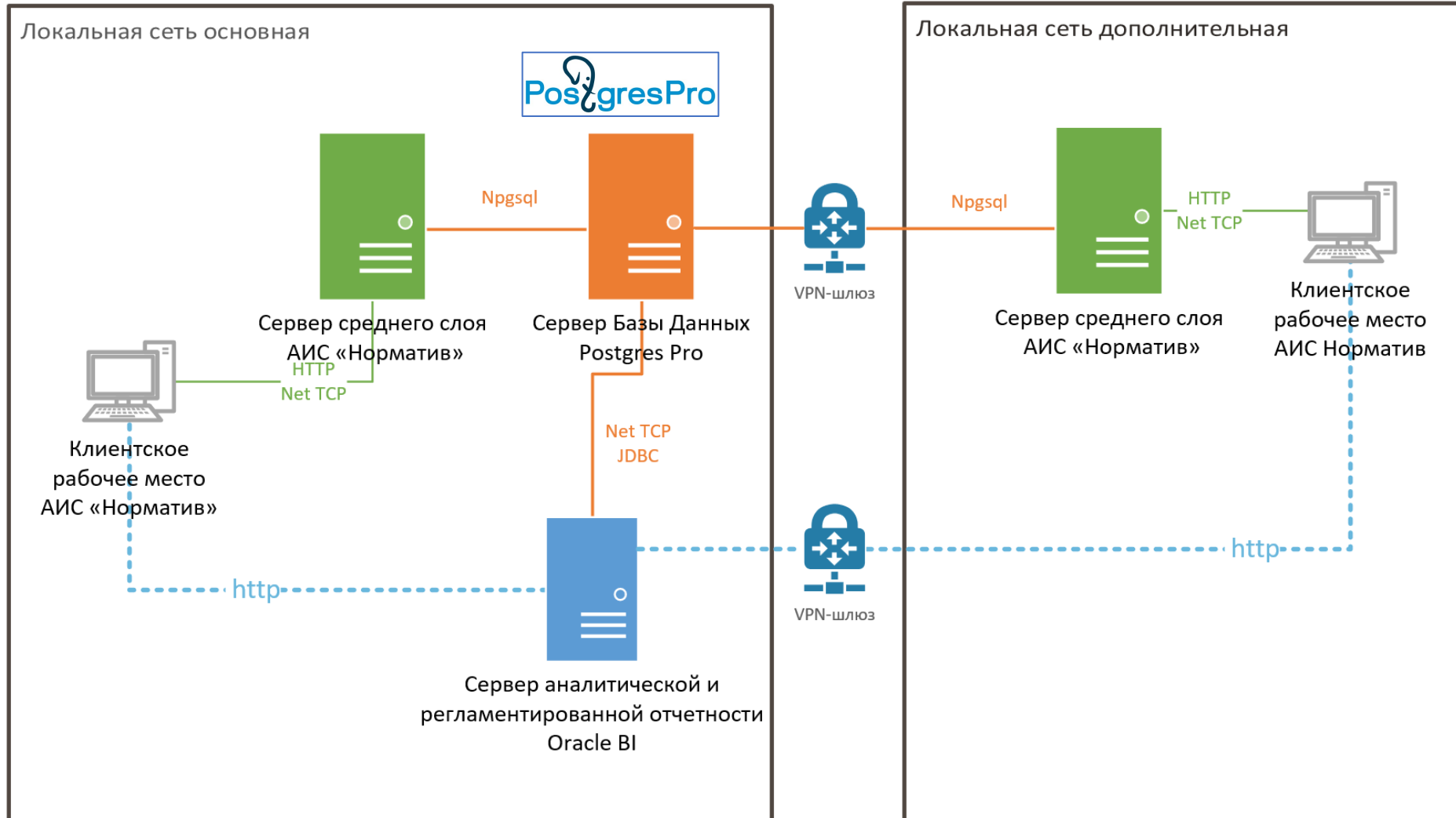
Приложение для миграции АИС Норматив – архитектура на Oracle



АИС “Норматив” – целевая архитектура (меняется только СУБД!)

- СУБД: Postgres Pro Standard 14 (в ходе проекта сделали апгрейд на PG Pro Std 15)
- Бизнес-логика остается в БД на хранимых процедурах PL/pgSQL
- App Tier модифицируется для поддержки Postgres Pro
- Отчеты на Oracle BI модифицируются для работы с Postgres Pro

Целевая архитектура с СУБД Postgres Pro



Этапы проекта по миграции с СУБД Oracle на Postgres Pro

- Миграция метаданных (определений таблиц, индексов, constraints и т.д.) с помощью утилиты ORA2PG, корректировка DDL-скриптов
- Миграция бизнес-логики с Oracle PL/SQL на PL/pgSQL с помощью ORA2PG и дальнейшая ручная корректировка исходного кода
- Миграция данных с помощью ORA2PG (downtime в 2 суток позволял это сделать)
- Переписывание запросов и адаптация бизнес-логики
- Тестирование
- Переход в опытную эксплуатацию
- Переход в промышленную эксплуатацию

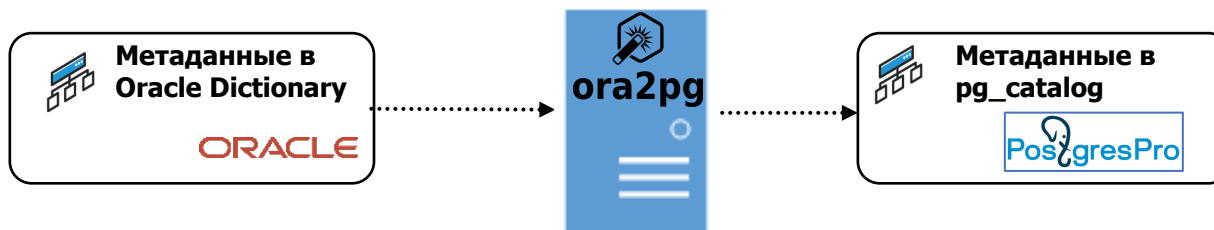
Миграция метаданных с ORA2PG

Подключение к СУБД Oracle и чтение метаданных из Oracle Dictionary

Формирование DDL-скриптов для Postgres Pro

Проблема: неверные типы в Id-столбцах, в Oracle в столбце был указан тип NUMBER, а не NUMBER(9,0).

Неверные типы данных в .NET Entity Framework - потребовалась ручная корректировка скриптов!



```
CREATE TABLE app_version (  
  id real NOT NULL, BIGINT NOT NULL,  
  major bigint NOT NULL,  
  minor bigint NOT NULL,  
  cmt varchar(200),  
  patch_number varchar(50),  
  created_on timestamp NOT NULL DEFAULT  
  LOCALTIMESTAMP,  
  created_by varchar(50) NOT NULL,  
  version_number bigint NOT NULL DEFAULT 1,  
  app_version varchar(200));
```

Миграция PL/SQL-кода в PL/pgSQL

- Получение первоначального варианта кода с помощью ORA2PG – полностью нерабочий код ☹
 - Некорректная конвертация запросов Connect By Prior в рекурсивый WITH
 - Некорректная конвертация операторов UPDATE – синтаксис Oracle не работает
 - Нет локальных контекстов сессии (SYS_CONTEXT), эмулируются через параметры (set_config)
 - Отсутствие поддержки коллекций и методов-итераторов коллекций
 - Другой синтаксис поддержки XML в запросах
- Ручная корректировка исходного кода в соответствии с синтаксисом и правилами PL/pgSQL
- Отсутствие поддержки оператора MERGE в Postgres Pro Standard 14
 - Апгрейд до Postgres Pro Standard 15 (как только вышел – сразу перешли!)
- Активное использование расширения ORAFCE для эмуляции функций REGEXP_* и системных представлений Oracle

Пример PL/pgSQL-кода после ORA2PG

```
UPDATE sql_report t
SET
t.title          = p_title,
t.content        = p_content,
t.cmt           = CASE
                        WHEN p_cmt=commonv.vp_default_text_value THEN
                            t.cmt
                        ELSE p_cmt
                    END,
t.modified_on = clock_timestamp(),
t.modified_by = coalesce(p_user, get_user) get_user()
WHERE
    id = p_id;
```

Не работает partition pruning, если ключ секции в подзапросе

```
SELECT *
FROM
  resources
WHERE
  period_id = (SELECT period_id FROM periods WHERE period_name=:a)
```

```
-> Hash Join (cost=8.30..1142934.76 rows=12526 width=1928) (actual time=3424.783..3443.614
rows=13720 loops=3)
```

```
    Hash Cond: (resources.period = period.id)
```

```
      -> Parallel Append (cost=0.00..1128516.97 rows=5436240 width=1929) (actual
time=0.041..3031.622 rows=4349004 loops=3)
```

```
        -> Seq Scan on resources_p_150826697 resources_373 (cost=0.00..0.00 rows=1
width=1518) (actual time=0.019..0.020 rows=0 loops=1)
```

```
        -> Seq Scan on resources_p_150855043 resources_377 (cost=0.00..0.00 rows=1
width=1545) (actual time=0.005..0.005 rows=0 loops=1)
```

```
    . . . . .
```

Не работает partition pruning – workaround (переписать запрос)

```
SELECT  
    period_id  
INTO  
    xPeriod_Id  
FROM  
    periods  
WHERE period_name = a;
```

```
Seq Scan on resources_p_149000015 resources (cost=0.00..4099.51 rows=41161 width=1539)  
(actual time=0.016..29.817 rows=41161 loops=1)
```

```
    Filter: (period = 149000015)
```

```
Planning Time: 0.249 ms
```

```
Execution Time: 31.282 ms.
```

С чем еще столкнулись

- Доработка приложения под Postgres Pro
 - Неверное преобразование типов данных между Oracle и Postgres библиотеки для Entity Framework
 - Нет Autolist partitioning – пришлось делать автоматическое создание секций через триггеры
 - Нет поддержки Package – заменили на схемы и функции/процедуры
 - Переделка синтаксиса вызова функции на стороне сервера приложений (SELECT ...FROM dual работает некорректно)
- Установка минорного обновления Postgres Pro Standard 15.2 устранило ошибку
 - Проблема неработающего расчета виртуальных колонок при изменении данных командой MERGE
- Нет возможности корректно логировать ошибки снаружи блока exception
- Изолированные транзакции

Как вводили в эксплуатацию

- Нет возможности остановить развитие продукта на время проекта миграции
- Заказчик очень переживал за достоверность информации
 - Очень высокая ответственность за цифры
 - Эффект «черного ящика» для системы – которая давно введена в эксплуатацию
 - Диссонанс необходимости и целесообразности
 - Из за этого - постоянная отсрочка полного перехода на Postgres
- Тестирование и проверка
 - Классическое тестирование функциональности через клиентское приложение
 - Сверка данных путем параллельного экспорта ключевых агрегирующих таблиц и построчной выверки данных с помощью автоматизации excel
 - Сверка отчетов – печатных форм
 - Сверка XML отчетов – путем загрузки данных в БД
- Цель – прийти к абсолютно **идентичным данным** либо **иметь объяснения их расхождения**

Результаты: “АИС Норматив” на Postgres Pro Standard 15

- Система находится в промышленной эксплуатации!
- Что понравилось в Postgres Pro Standard
 - Легкий и быстрый “движок” СУБД – ничего лишнего
 - Идеально подходит для небольших и средних баз данных
 - Отчеты работают быстрее, чем Oracle!
 - Гибкая расширяемость за счет наличия большого числа расширений, библиотек и инструментов разработки



Трудозатраты: миграция “АИС Норматив” с Oracle на Postgres Pro

	Этап работ	Трудозатраты чел/мес
1	Установка настройка СУБД, ora2pg, корректировка DDL-скриптов, отладка миграции данных, тестирование	2
2	Переписывание хранимых процедур и тестирование	2
3	Переделка отчетов и тестирование	1
4	Доработка сервера приложений и тестирование	1
	Итого	6





КОНТАКТЫ

Всего в шаге от изменений

Если у вас остались вопросы,
свяжитесь с нами любым удобным
способом

+7 (495) 796-69-09

sales@bim-info.ru