Особенности миграции с Microsoft SQL Server на СУБД Firebird

Алексей Бехтин
Ведущий программист Финам

FIREBIRD CONF 2024 | РЕДСОФТ

О чём доклад?

Будни прикладного разработчика в Firebird после отказа от MS SQL Server

Основные моменты

База бэк-офиса:

- Терабайты данных
- Миллионы поручений в день
- Таблицы миллиардники
- Два десятка разработчиков на одну БД
- 2300 таблиц
- 2M строк T-SQL кода

Хранимки в 2024?

Хранимки

+

- Расчёты рядом с данными это удобно
- Скорость обработки
- Объёмы данных превышают оперативную память
- Транзакционность
- Применяются и правятся на лету
- Альтернативы сложнее/дороже в поддержке
- Не отменяют сервисного подхода

-

- Много антипаттернов
  - Меняем код для отладки/тестирования
  - Плохая декомпозиция -> Copy/Paste
  - Процедуры с большим количеством параметров
- Требуется опыт
- Требуют определённого процесса
- Могут не всё
- Есть проблемы масштабирования

Как с этим жить?

- Организовать процесс: Спринты, аналитики, ТЗ, код-ревью, тестирование, приёмка
- По возможности использовать лучшие инструменты, но есть и бесплатные
- Jira, Trello, Gitlab, gogs, gitea
- Jenkins, Gitlab CI
- Flyway, Liquibase, написать своё
- Стиль кодирования и устоявшиеся практики
- Документирование (Markdown, HTML, AsciiDoc, Obsidian, Draw.io, Excalidraw, tldraw, VS Code + плагины)
- Отдельно – понимание от бизнеса
- Контуры разработки, тестирования, препрода

Работа над задачей

- Сделали ветку в Git
- Изменения в имеющемся коде, создание нового, скрипты миграции
- Выкладываем на тестирование
- Скрипты должны корректно обрабатывать повторное выполнение

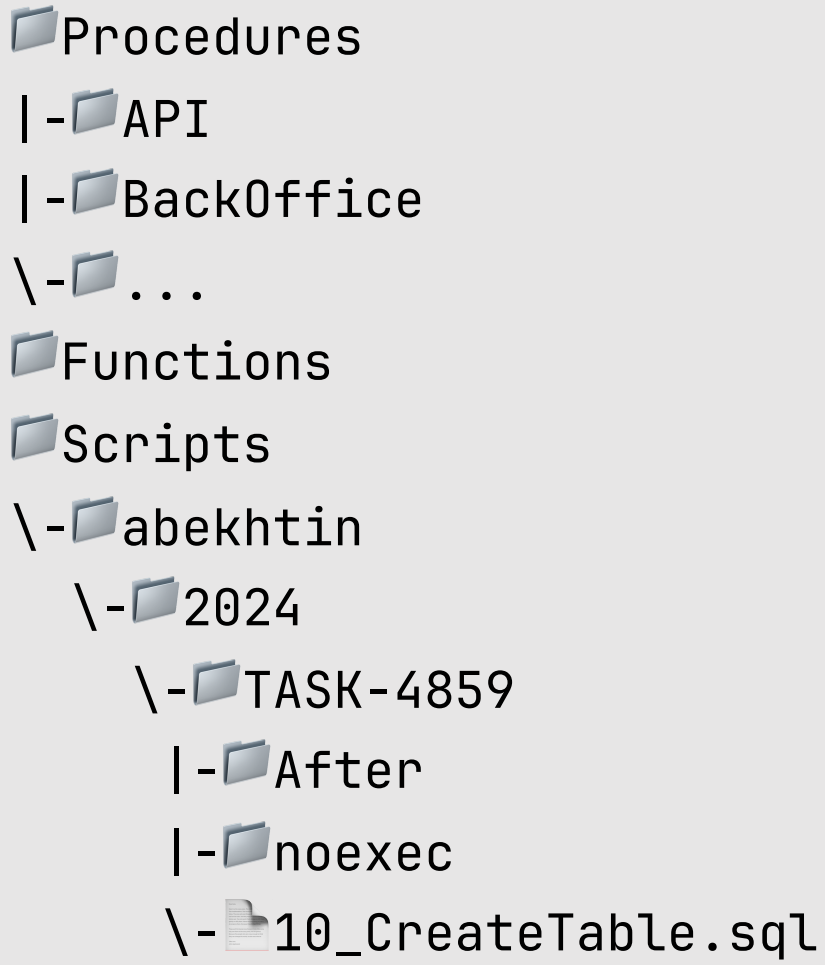
Создание поля таблицы MSSQL

```
USE [BackOffice]
GO

IF NOT EXISTS(SELECT * FROM sys.columns WHERE name = 'nnUserEdit'
  AND [object_id] = OBJECT_ID('[dbo].[Operations]')) BEGIN

  ALTER TABLE [dbo].[Operations]
    ADD [nnUserEdit] INT DEFAULT NULL

  EXEC [SQL].[Comment]
    @ObjectName   = '[dbo].[Operations]',
    @ParamName    = 'nnUserEdit',
    @Description  = 'ID персоны, кто последним редактировал операцию'
END
```

Создание поля таблицы Firebird

```
EXECUTE BLOCK AS
BEGIN
  IF (NOT EXISTS(SELECT * FROM RDB$RELATION_FIELDS
  WHERE RDB$RELATION_NAME = 'OPERATIONS' AND RDB$FIELD_NAME = 'nnUserEdit')) THEN BEGIN
    EXECUTE STATEMENT Q'{
       ALTER TABLE Operations
         ADD "nnUserEdit" INT
       }'
    WITH AUTONOMOUS TRANSACTION;
    EXECUTE STATEMENT Q'{
       COMMENT ON COLUMN OPERATIONS."nnUserEdit" IS 'ID персоны, кто...'
       }'
    WITH AUTONOMOUS TRANSACTION;
  END
END
```

Всё изменения хранятся в Git



Выкладываем изменения

- А завтра оно накатится?
- А что если «пересеклись»?
- То, что прошёл merge, не значит, что код рабочий
- Примечания о редактировании в самом коде PSQL

ALTER зависимых объектов

- Если в Firebird скомпилировалось, то код с большой вероятностью работает
- Если в MSSQL скомпилировалось, то это может не значить ничего
- Переименование поля с большим количеством зависимостей
  - Уточнение имени
  - Изменение типа
- Переименование таблицы с большим количеством зависимостей
  - Уточнение имени
  - Построение индекса на большой таблице

Отладка через принты в MSSQL

```
DECLARE
  @Msg       varchar(100),
  @StepID    int           = 3,
  @StepCount int           = 5
PRINT 'Выполнение шага 1/5'
SET @Msg = 'Выполнение шага 2/5'
PRINT @Msg
RAISERROR('Выполнение шага %d/%d', 0, 0, @StepID, @StepCount) WITH NOWAIT
```

Отладка через принты в Firebird

```
CREATE TABLE DEBUG_MESSAGES (
  ID  INT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
  MSG VARCHAR(1024))

CREATE FUNCTION PRINT_DEBUG(MSG VARCHAR(1024)) RETURNS INT AS BEGIN
  IN AUTONOMOUS TRANSACTION DO
     INSERT INTO DEBUG_MESSAGES(MSG) VALUES(:MSG);
END

EXECUTE BLOCK AS BEGIN
  PRINT_DEBUG('Выполнение шага 1/3');
END
```

Отладка через принты в Firebird

- IN AUTONOMOUS TRANSACTION DO
- Внешние таблицы
- UDR IBSurgeon/http_client_udr + web-сервер на Go, Node.js
- Ред База RDB$TRACE_MSG
- Самописная UDR

Отладочные датасеты в MSSQL

```
CREATE PROCEDURE ...
  @Debug bit = 0
...
IF @Debug = 1
  SELECT [#Tariffs_Step_1]='', * FROM #Tariffs
...
DROP TABLE IF EXISTS [Temp].[Tariffs_Step_1]
SELECT *
INTO [Temp].[Tariffs_Step_1]
FROM #Tariffs
```

Отладочные датасеты в Firebird

```
INSERT INTO DEBUG_MESSAGES (MSG) VALUES(
  (SELECT LIST(
      'RDB$RELATION_NAME' || COALESCE(TRIM(R.RDB$RELATION_NAME), 'NULL')||
     ',RDB$RELATION_TYPE' || COALESCE(TRIM(R.RDB$RELATION_TYPE), 'NULL')||
     ',RDB$SYSTEM_FLAG'   || COALESCE(TRIM(R.RDB$SYSTEM_FLAG)  , 'NULL'))
   FROM RDB$RELATIONS R)
)
```

Отладочные датасеты в Ред Базе

```
INSERT INTO DEBUG_MESSAGES (MSG) VALUES(
            (SELECT JSON_ARRAYAGG(
               JSON_OBJECT(
                  'RDB$RELATION_NAME' : TRIM(R.RDB$RELATION_NAME),
                  'RDB$RELATION_TYPE' : TRIM(R.RDB$RELATION_TYPE),
                  'RDB$SYSTEM_FLAG'   : TRIM(R.RDB$SYSTEM_FLAG)
                  RETURNING BLOB SUB_TYPE TEXT)
              RETURNING BLOB SUB_TYPE TEXT)
            FROM RDB$RELATIONS R)
)
```

Привыкли использовать в MSSQL

- Табличные переменные
- Стартовать транзакции внутри процедуры
- SELECT из другой БД и другой СУБД
- JOIN к другой таблице в другой БД
- Табличная инлайн-функция (view с параметрами)
- Триггеры уровня выражения
- UPDATE FROM, DELETE FROM
- Отладка на больших объёмах/нагрузках это логи

Табличные переменные в MSSQL

```
CREATE TYPE [dbo].[TActivesStates] AS TABLE(
       [StateId] TINYINT      NOT NULL,
       [Code]    VARCHAR(200) NOT NULL,
       [Sign]    SMALLINT     NOT NULL
)

DECLARE @States TActivesStates

INSERT @States ([StateId], [Code], [Sign])
        VALUES (1508,      'SUM',  -1)

SELECT * FROM [dbo].[GetStatesInfo](@States)
```

Табличные переменные в Firebird

- Иногда это список значений с разделителем
  - Хранимая процедура SPLIT() на PSQL
  - split_udr
  - UNLIST (Firebird 6)
- GTT
- JSON (Ред База 5)

Транзакции в MSSQL

```
BEGIN TRY
  BEGIN TRAN
  -- Код в транзакции
  COMMIT TRAN
END TRY
BEGIN CATCH
  IF @@TRANCOUNT > 0
    ROLLBACK TRAN
  EXEC [System].[ReRaiseError] ...
END CATCH
```

Транзакции в Firebird

```
EXECUTE BLOCK AS BEGIN
  IN AUTONOMOUS TRANSACTION DO BEGIN
    -- Автономная транзакция в текущей БД
  END
  EXECUTE STATEMENT 'SELECT ...' ON EXTERNAL DATA SOURCE '...'
  WITH AUTONOMOUS TRANSACTION DO BEGIN
    -- Автономная транзакция в другой БД
  END
  BEGIN -- Неявный savepoint
    WHEN ANY DO
    -- Обработка ошибки
  END
END
```

Табличная инлайн-функция (view с параметрами)

```
CREATE FUNCTION ... (@CodeId INT)
RETURS TABLE
AS
RETURN
  SELECT
    ...
  FROM
  WHERE [CodeId] = @CodeId
  GROUP BY ...
```

UPDATE FROM, DELETE FROM

```
UPDATE FROM T
  [Sum] = ...
FROM Transactions T
INNER JOIN ... ON ... = T.[Id]
WHERE ...

DELETE FROM T
FROM Transactions T
INNER JOIN ... ON ... = T.[Id]
WHERE ...
```

Псевдонимы полей в MSSQL

```
INSERT ... ([AccountName], [IsValid], ...)
SELECT
  [AccountName] = A.AccountName,
  [IsValid]     = CASE
                    WHEN ... THEN ...
                    WHEN ... THEN ...
                    ELSE ...
                  END,
   ...
FROM
```

Временные таблицы vs GTT

MSSQL

- Могут порождать большие запросы
- Порождают ошибки при совпадении имён
- Часть неявного API (!)

+ Передача датасетов между процедурами
+ Элемент декомпозиции запросов

GTT

- Нельзя создать из селективного запроса по месту
- Меньше методов доступа
- Нельзя наполнить данными и потом построить индекс

+ Передача датасетов между процедурами
+ Присутствуют в схеме БД постоянно (!)

Курсоры в MSSQL

```
DECLARE CurAccounts CURSOR LOCAL FAST_FORWARD FOR
      SELECT [AccountId] ...
OPEN CurAccounts
WHILE 1 = 1 BEGIN
  FETCH NEXT FROM CurAccounts INTO @AccountId
  IF @@FETCH_STATUS <> 0 BREAK
  -- Работа с данными курсора
END
CLOSE CurAccounts
DEALLOCATE CurAccounts
```

Курсоры в Firebird

```
FOR SELECT
  AccountId
FROM ...
INTO :AccountID DO
BEGIN
  -- Работа с данными курсора
END
```

Эквиваленты

MSSQL	Firebird
CROSS/OUTER APPLY	LATERAL
UPDATE FROM, DELETE FROM	MERGE
Активные запросы, блокировки	Таблицы мониторинга, консольные утилиты
Хинты оптимизатора	+0, PLAN, переписывание через FOR SELECT
Методы доступа MERGE/HASH LEFT JOIN	Firebird 6
Метод доступа INDEX SCAN	Денормализация 1:1
Асинхронные операции и сервис брокер	Ред База планировщик, POST_EVENT, Сторонние внешние решения
Именованные параметры	Firebird 6

Статистика выполнения

Процедуры

Статистика выполнения процедуры

MSSQL

- Include Actual Execution Plan
- `SET STATISTICS XML`

Firebird

- `SET PER_TABLE_STATS`

Статистика выполнения

Процедуры

Запроса внутри процедуры

Статистика выполнения запроса MSSQL

```
CREATE ROCEDURE …
…
SET STATISTICS XML ON
SELECT …
SET STATISTICS XML OFF
…
```

Статистика выполнения запроса Firebird

```
CREATE PROCEDURE …
…
SAVE_STAT          -- сохраняем $MON
SELECT …
SAVE_STAT
…

SELECT * FROM STAT_DIFF
```

Статистика выполнения запроса Firebird

Редактор запросов - скрипт2.sql | Профайлер [2]

База Данных: TEST_50_... Соединение: 127.0.0.1/... Старт | Пауза | Возобновить | Стоп | Отменить | Очистить

ИМЯ ПРОЦЕССА	ОБЩЕЕ ВРЕМЯ	СРЕДНЕЕ ВРЕМЯ	КОЛ-ВО ВЫЗОВОВ
Profiler Session [ID: 35]	4 ms [100.00%]	4 ms	1
execute procedure TEST_INS	4 ms [100.00%]	4 ms	1
TEST_INS	4 ms [99.83%]	4 ms	1
2: execute procedure ins;	2 ms [63.01%]	2 ms	1
INS	2 ms [98.53%]	2 ms	1
7: insert into tab values (:n, mult(:n, 2));	2 ms [83.93%]	4558 ns	500
8: n = n + 1;	217600 ns [8.01%]	217 ns	1000
6: if (mod(n, 2) = 1) then	133800 ns [4.93%]	133 ns	1000
3: while (n <= 100000)	83000 ns [3.06%]	82 ns	1001
1: DECLARE n integer = 1;	1700 ns [0.06%]	1700 ns	1
3: delete from tab;	1 ms [36.97%]	1 ms	1

Статистика выполнения

Процедуры

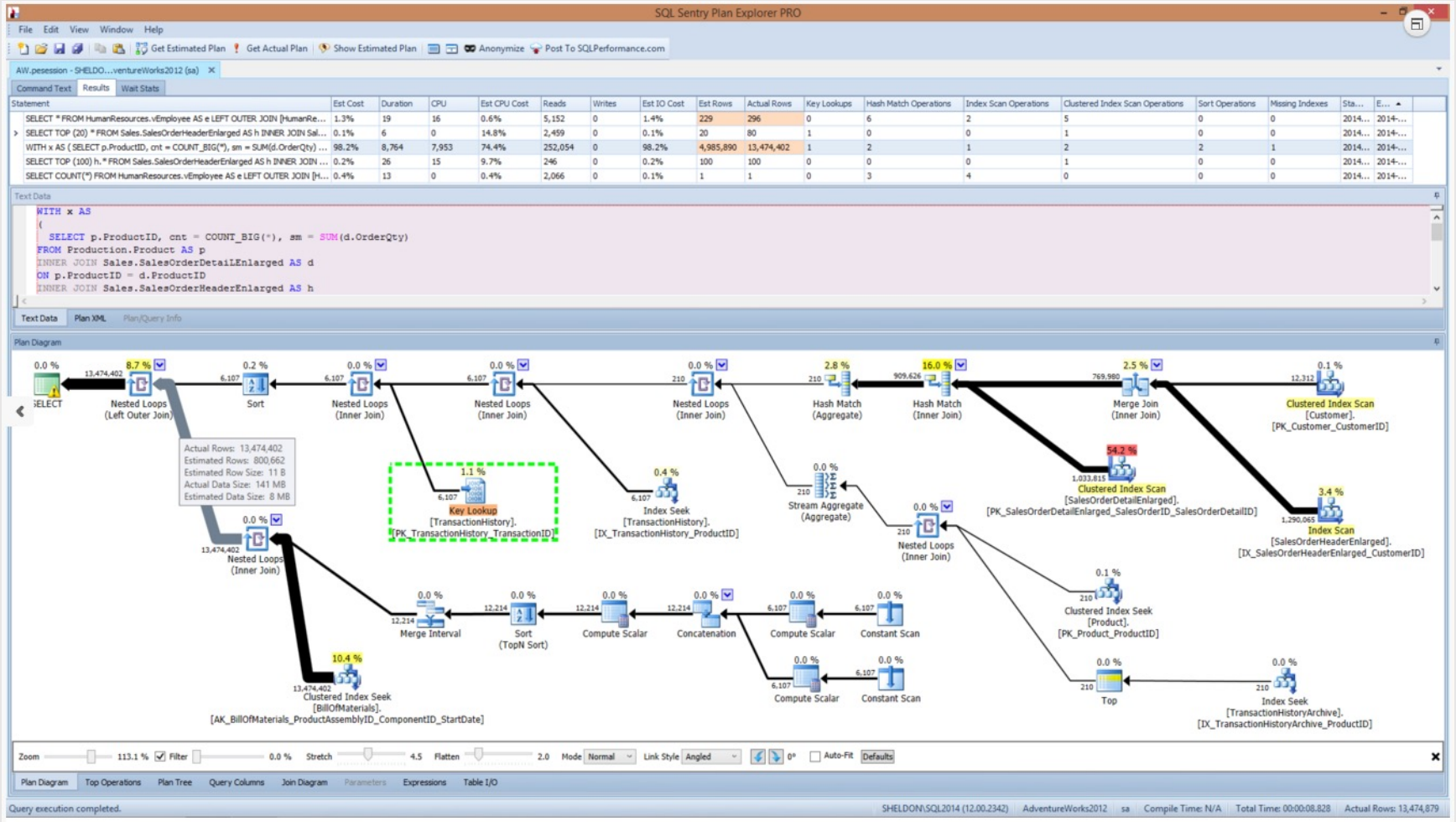
Запроса внутри процедуры

Узлов плана

Статистика по узлам плана MSSQL

```
CREATE PROCEDURE …
…
SET STATISTICS XML ON
SELECT …
SET STATISTICS XML OFF
…
```

Статистика по узлам плана MSSQL



Статистика по узлам плана Firebird

Как?

Статистика по узлам плана Firebird

Профайлер!

Инструменты

MS SQL

- SQL Server Management Studio
  + Red Gate SQL Prompt
  + SSMS Boost
  + SQL Sentry Plan Explorer

Firebird

- IBExpert
- RedExpert
- EMS SQL Manager

Подсказки для вычисляемых полей

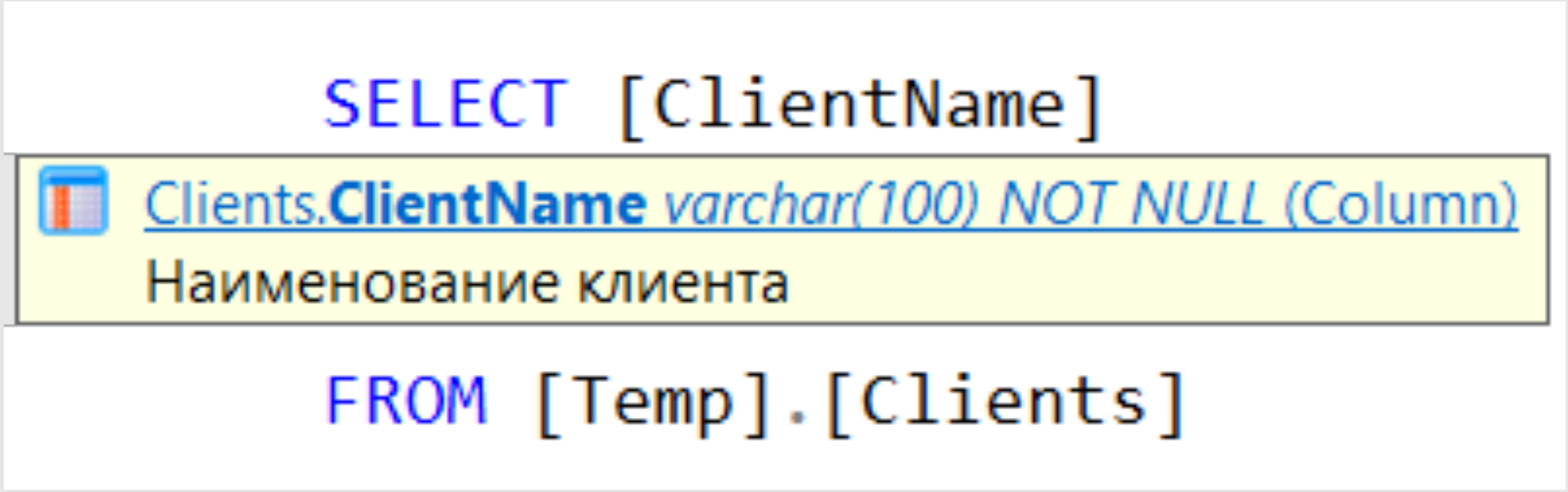
```
WITH A AS
(
   SELECT [Id] = 123
)
SELECT A.| FROM A
```

Id

Метаданные

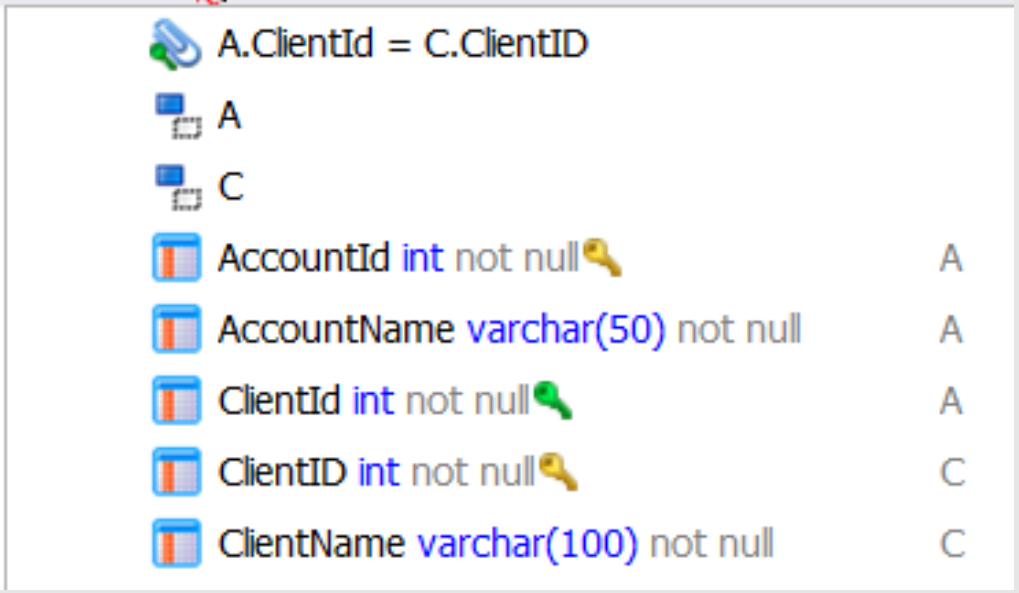
```
CREATE TABLE [Temp].[Clients]
(
  [ClientID]   int PRIMARY KEY,
  [ClientName] varchar(100) NOT NULL
)

CREATE TABLE [Temp].[Acounts]
(
  [AccountId]   int PRIMARY KEY,
  [AccountName] varchar(50) NOT NULL,
  [ClientId]    int         NOT NULL
)

ALTER TABLE [Temp].[Acounts]
ADD CONSTRAINT [FK_Temp.Acounts#ClientId]
FOREIGN KEY ([ClientId]) REFERENCES [Temp].[Clients] ([ClientID])
```

Тип поля и описание



Подсказка для JOIN

```
SELECT
  C.[ClientName],
  COUNT(*)
FROM [Temp].[Clients] C
INNER JOIN [Temp].[Acounts] A ON
```



Подсказка для GROUP BY

```
SELECT
   C.[ClientName],
   COUNT(*)
FROM [Temp].[Clients] C
INNER JOIN [Temp].[Acounts] A ON A.[ClientId] = C.[ClientID]
GROUP BY
```

All non-aggregated columns

Есть в Firebird и нет в MSSQL

- Больше возможности декомпозировать и переиспользовать код, не теряя производительности и читаемости
- Локальные процедуры и функции
- В PSQL-функциях нет ограничений на использование SQL
- Автономные транзакции
- Стек вызовов при ошибке
- Пакеты
- Детерминированные функции, как константы
- IS [NOT] DISTINCT FROM (EXCEPT/ INTERSECT, MSSQL 2022)
- И ещё целый ряд полезностей для разработчика

Что не рассмотрено?

- Администрирование
- Партиционирование/Кластеризация
- Репликация
- Always On
- Компоненты доступа
- Типы данных
- SQL Server Integration Services (SSIS)
- SQL Server Analysis Services (SSAS)
- Колоночные таблицы
- Таблицы, оптимизированные для памяти

Вместо выводов

- Выстроить процесс
- Не рефакторить при переносе, или только то, что иначе никак
- Заручиться поддержкой бизнеса
- Выделить ресурс
- Неудачная попытка перехода - урок
- Чеклисты

Полезные ссылки

- Migration from MS-SQL to Firebird (2011)
  https://www.firebirdsql.org/manual/migration-mssql.html
- IBSurgeon HTTP Client UDR
  https://github.com/IBSurgeon/http_client_udr
- IBSurgeon Full Text Search UDR
  https://github.com/IBSurgeon/lucene_udr
- Библиотека SplitUdr
  https://github.com/sim1984/split_udr
- Спортмастер Разработка БАЗ ДАННЫХ! / Как готовят Oracle в Спортмастере / Максим Пермяков
  https://youtu.be/_liOxMtBNR8

Спасибо за внимание!