|  |
| --- |
| Photo displaying partial image of two pie charts on a canvas-textured page |
| XE: eXtended Events data management system. Общее описание  Версия 5.0 |
| |  |  |  | | --- | --- | --- | | sotivoli | 10/14/24 | Санкт-Петербург | |

Contents

[Введение 3](#_Toc179749450)

[Описание возможностей 3](#_Toc179749451)

[Преимущества eXtended Events перед Profiler 3](#_Toc179749452)

[Структура данных 6](#_Toc179749453)

[Модель данных 6](#_Toc179749454)

[Перечень таблиц: 7](#_Toc179749455)

[Описание обработки 8](#_Toc179749456)

[Инсталляция системы 10](#_Toc179749457)

[Инсталляция. Шаг 1: Сохранение хранимых процедур 10](#_Toc179749458)

[Инсталляция. Шаги 2 и 3: Создание таблиц и сессии XE 11](#_Toc179749459)

[Параметры процедуры XE\_Install для инсталляции 11](#_Toc179749460)

[Инсталляция. Шаг 4: корректировка настроек 13](#_Toc179749461)

[Инсталляция. Шаг 5: импорт данных 13](#_Toc179749462)

[Инсталляция. Шаг 6: назначение заданий в планировщик 13](#_Toc179749463)

[Инсталляция. Шаг 7: настройка отчетов 13](#_Toc179749464)

[Параметры и опции, общие для всех процедур 14](#_Toc179749465)

[Основные хранимые процедуры 16](#_Toc179749466)

[Хранимая процедура XE\_Xel 16](#_Toc179749467)

[Считывание и фильтрация данных 18](#_Toc179749468)

[Вывод диагностической информации: 18](#_Toc179749469)

[Обновление справочников полей 19](#_Toc179749470)

[Обновление Dic\_Source 20](#_Toc179749471)

[Хранимая процедура XE\_TopSum 20](#_Toc179749472)

[Хранимые процедуры XE\_Count\_JDE и XE\_Count\_Dummy 22](#_Toc179749473)

[Хранимая процедура XE\_Daily 23](#_Toc179749474)

[Вспомогательные процедуры 23](#_Toc179749475)

[Хранимая процедура XE\_Day 23](#_Toc179749476)

[Хранимая процедура XE\_TextData 24](#_Toc179749477)

[Хранимая процедура XE\_Convert 25](#_Toc179749478)

[Хранимая процедура BaReTa (Backup Restore Tables) 28](#_Toc179749479)

[Служебные процедуры 29](#_Toc179749480)

[Хранимые процедуры XE\_Import\_XE44, XE\_Import\_Profiler 29](#_Toc179749481)

[Хранимая процедура XE\_ExecLog 29](#_Toc179749482)

[Хранимая процедура XE\_CheckClear 30](#_Toc179749483)

[Хранимая процедура XE\_CheckParams 30](#_Toc179749484)

[Справочник XEd\_Set 31](#_Toc179749485)

[Анализ данных XE 33](#_Toc179749486)

[Представления для таблиц XE 33](#_Toc179749487)

[Доступ к данным из Excel и из Power BI 34](#_Toc179749488)

[Приложение 1. Сравнение данных конвертации 37](#_Toc179749489)

# Введение

# Описание возможностей

Система ***XE (eXtended Events)*** представляет собой набор хранимых процедур, таблиц, вспомогательных программ, и файлов, предназначенных для сбора, хранения и анализа информации, собираемой службой MS SQL Server extended events.

Возможности системы позволяют отбирать и сохранять в нормализованном виде следующую информацию:

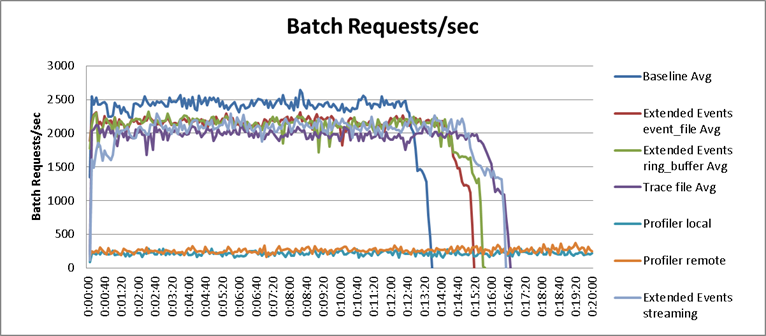
1. Полные данные, собранные сессией Extended events (таблицы вида *XE\_ИМЯ*)
2. Суммарную информацию по потреблению ресурсов в течении суток (таблица *XE\_Sum*)
3. Наиболее агрессивные запросы, отобранные по трем критериям, примененным к ресурсам CPU, Duration, Physical Reads, Logical Reads, Writes, Spills (таблица *XE\_Top*)
4. Справочники полей данных (таблицs вида *XEd\_справочник*)

Система имеет возможности гибкой настройки посредством параметров вызова и / или параметров, определенных в таблице настройки (

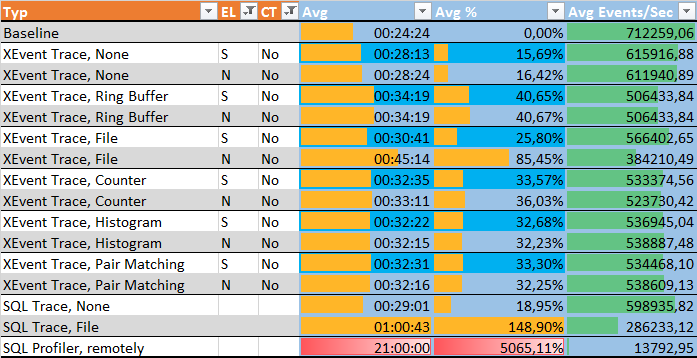
Система позволяет обрабатывать данные от различных сессий MS SQL Server extended events в единой базе данных, при этом полные данных могут храниться как в одной, так и в различных таблицах, суммарные данные и топ запросы хранятся в общих таблицах, но каждая запись имеет идентификатор, позволяющий идентифицировать, к какой сессии она относится.

## Преимущества eXtended Events перед Profiler

* Использование Extended Events вместо Profiler имеет ряд преимуществ:
  + Значительно меньшее влияние на производительность при включенном сборе данных, причем имеются расширенные настройки, с помощью которых на это можно влиять.  
    Сравнительные тесты показывают значительно большую производительность XEvents относительно profiler (подробнее см. [SQL Server Extended Events vs Profiler vs Trace](https://www.mssqltips.com/sqlservertutorial/9210/sql-server-extended-events-vs-profiler-vs-trace/))



Схожие результаты показывает и исследование [Performance overhead of tracing with Extended Event targets vs SQL Trace under CPU Load](https://andreas-wolter.com/en/performance-overhead-of-tracing-with-extended-event-targets-vs-sql-trace-under-cpu-load/)



* + Настройки сбора данных (события, фильтры и др.) можно менять на активных сессиях, прямо во время сбора данных.
  + Можно настроить различные способы хранения логов, причем одновременно в нескольких вариантах.
  + Встроенные инструменты в SQL Server Management Studio и инструкции TSQL для работы с ними.
  + Поддержка PowerShell
  + Может делать сбор данных в течении длительного периода
  + Profiler не развивается с версии 2008, считается устаревшим инструментом и может быть удален в будущем.
  + В XEvents включены встроенные шаблоны, аналогичные profiler

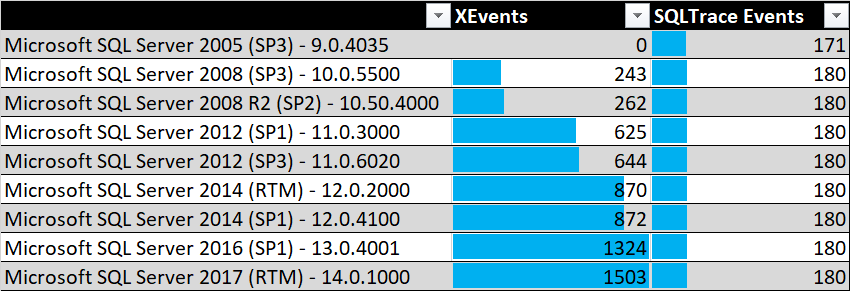


Рисунок 1 Сравнение число регистрируемых событий Profiler и XEvents

## Возможности системы XE для сбора данных

* Система XE предоставляет дополнительные параметры выполнения запросов по сравнению с данными от profiler:

|  |  |
| --- | --- |
| Page\_server\_reads | Число страниц, перемещенных с диска в буфер данных в памяти |
| Reads\_physical | Число страниц данных, считанных с диска, т.к. они не были доступны в памяти. |
| Reads\_logical | Считано страниц данных из памяти (соответствует Reads в profiler) |
| Spills | показывает, что буфер связи в параллельном плане запроса временно записан в базу данных tempdb . Это происходит редко и только если план запроса содержит несколько просмотров диапазона.[[1]](#footnote-1) |
| Row\_count | Возвращает число строк, затронутых при выполнении последней инструкции |
| Is\_System |  |
| Database\_name | Имя базы данных – в дополнение к id базы. Данное значение важно, когда данные анализируются на сервере SQL, отличном от того, на котором выполнялся запрос |
| Result\_Code, Result\_Text | Код выполнения (Result\_Code)и соответствует ему текст (Result\_Text). |
| Output\_parameters |  |

* Система позволяет вести сбор данных с различных сессий сбора данных, в т.ч. и на различных серверах с общими справочниками, при этом параметры сбора и хранения информации можно гибко настраивать для отдельных сессий и / или таблиц с помощью таблицы настройки XEd\_Set
* Параллельно с хранением полной информации идет отбор наиболее агрессивных запросов с целью долговременного хранения информации о них в таблице XE\_Top. При этом отбор идет по показателям CPU, Duration, Reads\_physical, Reads\_logical, Writes, Spills, Row\_count. При этом каждый показатель отбирается по трем критериям (с условием ИЛИ для этих критериев):
  + Отбирается N наиболее агрессивных запросов. Например, 200 топ запросов.
  + Отбираются все запросы, каждый из которых потребляет не менее p % от общего потребления ресурса за сутки. Например, 0,05%
  + Отбираются наиболее агрессивные запросы, которые в сумме потребили более q % от общего потребления ресурса за сутки. Например, 80%
* Ведется таблица XE\_Sum с суммарными показателями потребления ресурсов за день, что облегчает анализ общего потребления ресурсов по каждой сессии в отдельности или по всем сессиям вместе. В число показателей вносится также количественная информация о работе соответствующей прикладной системы, при этом для каждой сессии это может быть отельный показатель. Например, число заказов в JDE за сутки.
* Ранее обработанные данные не считываются и не записываются, что исключает дублирование информации и ускоряет обработку.
* Данные хранятся в нормализованной форме, что снижает потребление объема данных за счет дедупликации текстовых данных
* Из текста данных вычленяется (когда возможно) имя первой вызываемой процедуры и имя первой таблицы из параметра FROM, при этом из текста запроса предварительно вырезаются все комментарии (как типа – comment, так и типа /\* comment \*/). Это позволяет иметь более актуальную информацию. При этом имеется возможность сохранять обработанный текст запроса или не сохранять для экономии места.
* Система хранит текст запроса в формате nvarchar(max), что гарантирует доступность как минимум 4000 символов
* По тексту поля [TextData] и полю с удаленными комментариями вычисляется хеш текста запроса (соответственно [Hash] и HashText), что позволяет идентифицировать одинаковые тексты запросов в таблице полных данных и таблице *XE\_Top*
* Имеется средство для импорта данных из profiler в XE, что позволяет не потерять исторические данные
* Имеется средство для простой установки XE
* Настройка осуществляется посредством записей параметров в таблицу *XE\_Set*, либо указанием параметров при вызове хранимых процедур.
* Имеется возможность фиксации обращений к хранимым процедурам в журнале *XEd\_Log*
* Для таблиц с нормализированными данными автоматически формируются представления, в которых добавлены join на все справочники, что позволяет легко строить по ним отчеты. Например, для таблицы *XE\_SQL251* будет автоматически создано представление *XEv\_SQL251*, для таблицы *XE\_Top* – представление *XEv\_Top* и т.д.

# Структура данных

## Модель данных

Нормализованная структура данных представлена на рисунке на примере таблицы полных данных XE\_SOTIVOLI

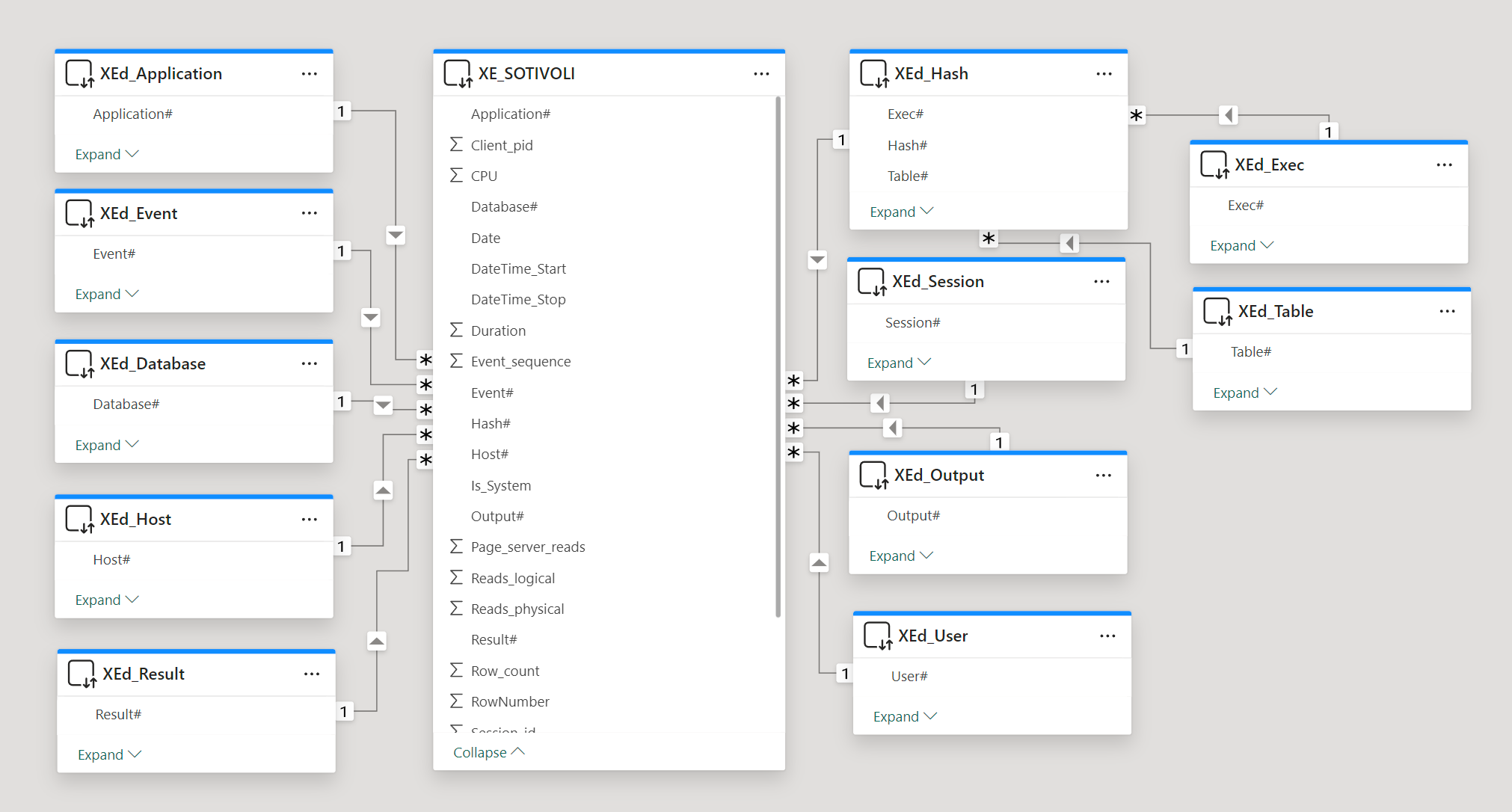


Рисунок 2 Схема модели данных XE

## Перечень таблиц:

### Таблицы с данными

| ***Таблица*** | ***Назначение*** | ***Кто изменяет таблицу*** |
| --- | --- | --- |
| ***XE\_<ИМЯ>*** | Хранение полных данных XE. Может быть определено несколько таблиц. ИМЯ всегда состоит из заглавных символов во избежание путаницы.  Например, *XE­\_SQL251, XE\_SOTIVOLI* | Создается процедурой XE\_Installво время инсталляции, либо при попытке первой записи в нее  Данные добавляют процедура считывания данных из .xel - файлов XE\_Xel, процедуры импорта данныхXE\_Import\_Profiler, XE\_Import\_XE44 |
| ***XE\_Sum*** | Таблица ежедневных суммарных значений потребления ресурсов в разрезе сессий | Процедуры обновления XE\_TopSumи XE\_Day |
| ***XE\_Top*** | Топ наиболее агрессивных запросов за сутки | Процедуры обновления XE\_TopSumи XE\_Day |

### Таблицы со справочниками

**Замечание**: процедуры импорта данных XE\_Import\_Profile, XE\_Import\_XE44 (вызываются из процедуры XE\_Convert) изменяют те же таблицы, что и процедура XE\_Xel

| ***Таблица*** | ***Назначение*** | ***Кто изменяет таблицу*** |
| --- | --- | --- |
| ***XEd\_Application*** | Справочник приложений | XE\_Xel |
| ***XEd\_Database*** | Справочник баз данных | XE\_Xel |
| ***XEd\_Event*** | Справочник событий XE (Profiler) | XE\_Xel |
| ***XEd\_Exec*** | Справочник вызываемых запросами процедур | XE\_TextData  (может вызываться из XE\_Xel ) |
| ***XEd\_Hash*** | Справочник хешей и текстов запросов | XE\_Xel  , а также XE\_TextData  (может вызываться из XE\_Xel ) |
| ***XEd\_Host*** | Справочник хостов | XE\_Xel |
| ***XEd\_Output*** | Справочник полей Output | XE\_Xel |
| ***XEd\_Result*** | Справочник полей результата выполнения запроса | XE\_Xel |
| ***XEd\_Session*** | Справочник сессий XE | XE\_Xel |
| ***XEd\_Table*** | Справочник таблиц, читаемых запросами | XE\_TextData  (может вызываться из XE\_Xel ) |
| ***XEd\_User*** | Справочник пользователей | XE\_Xel |

### Служебные таблицы

| ***Таблица*** | ***Назначение*** | ***Кто изменяет таблицу*** |
| --- | --- | --- |
| ***XEd\_Log*** | Регистрация выполнения процедур XE | *XE\_ExecLog* |
| ***XEd\_Set*** | Параметры работы процедур XE | Создается процедурой XE\_Installво время инсталляции  Данные в таблице могут добавляться, удаляться и изменяться пользователем |
| ***XEd\_Source*** | Считанные файлы .xel и их параметры на момент последнего считывания в разрезе сессий. | ТолькоXE\_Xel  (но не процедуры импорта) |

# Описание обработки

Схема обработки данных приведена на рисунке ниже

1. Первичные данные о выполняемых запросах сессия SQL Server eXtended Events собирает в файлы типа .xel в папке, определенной при запуске сессии.  
   Данные записываются в последовательно организуемые файлы, число и размер файлов определяется при запуске сессии и может быть изменен. Эти параметры при работе содержатся в параметре Source, который может задаваться либо при вызове процедур, либо быть определен в справочнике *XEd\_Set*.
2. Регулярно запускаемая (например, ежечасно или ежедневно) процедура XE\_Xel считывает данные из тех файлов .xel, которые были изменены с момента предыдущего своего запуска о тех запросах, которые были выполнены позже самого последнего, считанного ранее (т.е. только более свежие запросы).  
   Данные записываются в таблицу полных данных: таблицу с именем вида *XE\_ИМЯ*, например, в таблицу *XE\_SQL251*.

Возможно определение различных серверов XE, тогда для каждого из них будет создана отдельная таблица.

1. Данные нормализуются для полей, имеющих текстовое значение, эти значения заносятся в соответствующие справочники, а таблица полных данных содержит ссылку на индекс в соответствующем справочнике, см. описание модели данных ниже.
2. На основании текста запроса (поле [TextData]) вычисляется хеш запроса. Это позволяет, с одной стороны, уменьшать объем хранимых данных, т.к. хранятся только уникальные тексты запросов, а с другой стороны – отслеживать статистику запросов, которые имеют одинаковый хеш.  
   Так как полные данные имеют большой размер, таблицы ограничены по времени хранения данных, которое задается в справочнике *XEd\_Set.* Для различных серверов XE и справочников возможно задать различные сроки хранения. Но в любом случае, срок хранения данных в справочниках будет не менее, чем максимальное время хранения для любой из таблиц в этом сервере – таким образом обеспечивается целостность данных.
3. На основании данных в поле [TextData] обновляются справочники таблиц и процедур, которые использовались запросом. Данная процедура может вызываться явно, или неявно (по умолчанию) процедурой XE\_Xel*.* Параметр *Skip* этой процедуры отменяет неявный вызов.
4. Процедура XE\_Dayформирует таблицы суммарного потребления ресурсов по дням *XE\_Sum* и таблицу наиболее интенсивных запросов *XE\_Top*. Последняя таблица позволяет более эффективно выявлять основные источники потребления тех или иных ресурсов.



Рисунок 3 Схема работы и потоков данных XE

Почти все процедуры имеют параметры (ключевые слова, задаваемые через переменную @Option, которые позволяют модифицировать процесс выполнения.

Для выполнения ежедневной обработки данных может использоваться процедура XE\_Daily. Данная процедура может выполнять, например, следующую последовательность:

* 1. Последовательно выполнять процедуру XE\_Xel для каждой из обрабатываемых сессий.
  2. Если при вызове процедур XE\_Xel использовалась опция Skip, то в этом случае необходимо явно вызвать процедуру XE\_TextData для каждой из таблиц полных данных для обработки новых строк в справочнике *XEd\_Hash*.  
     **Замечание**: Использование параметра *Skip* и явный вызов процедуры *XE\_TextData* целесообразен только если в таблицу полных данных одновременно записываются данные от различных сессий. В этом случае это позволит несколько снизить общее время выполнения.
  3. Необходимо выполнить процедуру *XE\_TopSum*, которая обработает данные за предыдущий период и заполнит таблицы *XE\_Top* и XE\_Sum.

Процедура XE\_Daily вызывает процедуры через процедуру XE\_ExecLog, которая регистрирует действия в журнале XEd\_Log

Данные о считанных файлах типа .xel и их параметрах на момент последнего чтения сохраняются в таблице XEd\_Source

Кроме того, имеются вспомогательные процедуры, которые облегчают ряд действий.

# Инсталляция системы

Для инсталляции системы, либо ее обновления с версии 4.4 на версию 5.0 необходимо выполнить следующие действия:

1. Сохранить хранимые процедуры.
2. Создать необходимые таблицы
3. Создать сессию extended events
4. При необходимости скорректировать таблицу настроек XEd\_Set
5. При необходимости выполнить импорт существующих данных
6. Скорректировать процедуру XE\_Daily и назначить ее выполнение (либо аналогичное выполнение необходимых процедур) в планировщике заданий
7. При необходимости скорректировать отчеты Excel и PowerBI

## Инсталляция. Шаг 1: Сохранение хранимых процедур

Перечень хранимых процедур определен в переменной @Procedures процедуры XE\_Install и включает в себя минимальный набор процедур:

XE\_Install  
XE\_Xel  
XE\_TextData  
XE\_ExecLog  
XE\_TopSum  
XE\_Day  
XE\_Count\_JDE  
XE\_Clear  
XE\_CheckParams

Остальные процедуры могут быть сохранены либо на этом этапе, либо по мере необходимости в них.

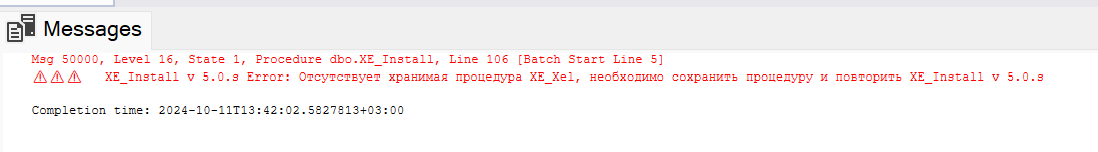
## Инсталляция. Шаги 2 и 3: Создание таблиц и сессии XE

Для создания таблиц и сессии необходимо вызвать процедуру XE\_Install.

При этом можно выполнить эти шаги за один вызов, либо отдельно таблицы и сессию.

Процедура выполняет следующие действия:

1. Выполняет проверку того, что минимальный набор процедур был сохранен. Если будет обнаружено, что какая-либо из требуемых процедур не сохранена, то будет выдано сообщение об ошибке.



В этом случае необходимо будет сохранить указанную процедуру и повторить выполнение инсталляции.

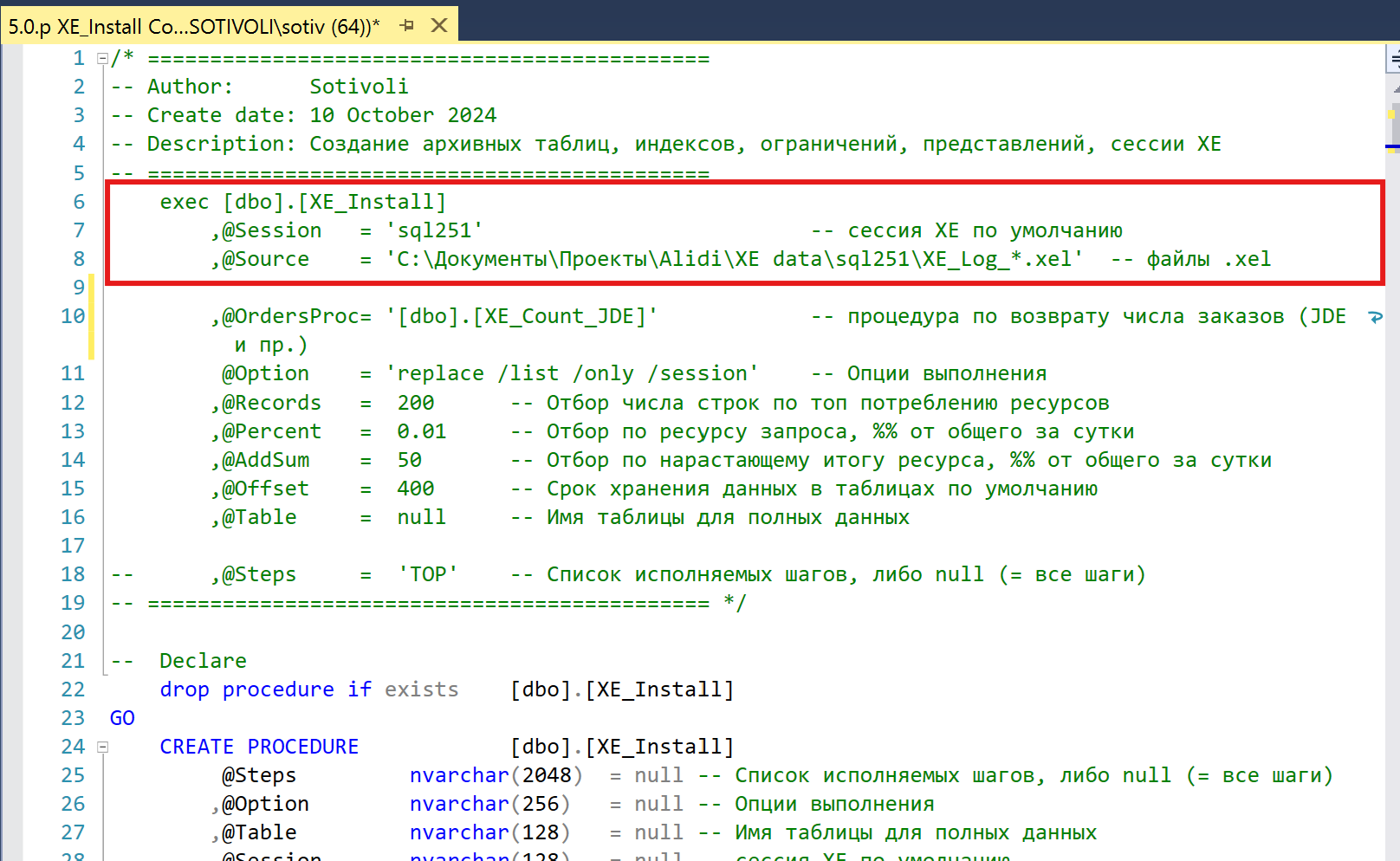
1. Создаются все необходимые справочники (таблицы с именем вида XEd\_справочник**)**
2. На основании введенных параметров формируются данные в таблице настроек XEd\_Set
3. На основании введенных параметров формируется одна таблица для полных данных (с названием вида XE\_ИМЯ) и представление для нее (с названием вида XEv\_ИМЯ)
4. Создаются таблицы XE\_Topи XE\_Sumи представления для них XEv\_Topи XEv\_Sum
5. Только в случае, если явно указана опция Create, создается сессия extended\_events с именем ***XEs\_имя***. При необходимости сессию можно сделать вручную, при этом необходимо согласовать ее параметры с параметрами в таблице XEd\_Set

При вызове процедуры целесообразно задать ей необходимые параметры. Полный перечень параметров приведен ниже.

## Параметры процедуры XE\_Install для инсталляции

Процедура имеет ряд параметров, что позволяет модифицировать выполнение инсталляции.

В комментариях процедуры приводится пример параметров для стандартной инсталляции, этот вызов можно скорректировать и использовать для вызова самой процедуры



Несмотря на то, что процедура имеет значения по умолчанию для всех параметров, а также любые из параметров можно ввести или изменить позже, тем не менее целесообразно передавать, как минимум, следующие параметры процедуры (выделены красной рамкой):

| ***Параметр*** | ***Назначение*** |
| --- | --- |
| @Session = имя | Имя, которое входит в название таблицы полных данных, сессии extended events и пр. места. Определяет логический набор данных XE. Целесообразно делать отдельные имена для каждого из сервера баз данных |
| @Source = путь | Путь к файлам и шаблон имен файлов. Используется как для создания сессии extended events, так и для чтения файлов с данными. Значение будет занесено в справочник XEd\_Set как значение по умолчанию |
| @Option = ‘параметры’ | В строке указываются через пробел параметры. Для создания сессии extended events необходимо указать параметр **Create**  Если данный параметр не указан, то сессия extended events не будет создана. |

**Замечание:** Повторный вызов процедуры XE\_Install выполняет только те действия, которые не были выполнены на текущий момент. Если необходимо повторно выполнить все действия, то необходимо указать в @Option параметр Replace, для выполнения отдельных шагов – указать их перечень в переменной @Steps

Дополнительные параметры можно посмотреть в описании процедуры. Например, можно выполнить только один шаг процедуры, указать название таблицы для полных данных и т.п.

Процедура XE\_Install используется в ряде случаев другими хранимыми процедурами, поэтому ее нельзя удалять после завершения инсталляции.

Для хранения полных данных будет автоматически создана таблица XE\_СЕССИЯ и представление XEv\_СЕССИЯ, где СЕССИЯ – имя заданной при инсталляции сессии.

## Инсталляция. Шаг 4: корректировка настроек

При необходимости можно провести корректировку настроек. Все настройки см. в описании таблицы настройки XEd\_Set

Замечание: любые настройки можно изменить в любой момент времени. Часть настроек, например, параметры отбора топ запросов или время хранения данных целесообразно проводить уже после того, как будет собран значимый объем данных на основании их статистики.

## Инсталляция. Шаг 5: импорт данных

Если есть необходимость, можно выполнить импорт имеющихся данных из числа собранных профайлером или предыдущей версией XE (4.4).

Эту операцию лучше выполнить до начала сбора данных, чтобы первичный ключ [RowNumber] возрастал по времени выполнения запросов.

**Замечание**: Если импорт сделать позже, когда таблицы уже наполнены данными, последовательность [RowNumber] может уже не соответствовать порядку выполнения запросов. Тогда можно сделать рефакторинг по полю [RowNumber] путем копирования данных во временную таблицу, очистки исходной таблицы и копирования данных обратно (без поля [RowNumber] с сортировкой по полю [Date] desc – в этом случае поле [RowNumber] будет автозаполнено корректно.  
См. описание процедуры XE\_Convert.

**Замечание**: Для выполнения XE\_Convert необходимо таже предварительно сохранить процедуры XE\_Import\_Profile и XE\_Import\_XE44

## Инсталляция. Шаг 6: назначение заданий в планировщик

Рекомендуется использовать ежедневный запуск процедуры XE\_Daily, ее следует скорректировать, в частности, скорректировать поля @Source, @Session – см. описание процедуры.

## Инсталляция. Шаг 7: настройка отчетов

В файлах отчета следует изменить на актуальные названия таблиц, представления и сервера базы данных. См. соответствующие описание.

# Параметры и опции, общие для всех процедур

Все процедуры имеют параметры, которые определяют источники данных, таблицы для записи, даты, параметры фильтрации данных и т.д.

Там, где это возможно, параметры определяются по умолчанию, если они не указаны (значение NULL).

При этом используется, где это возможно, следующий алгоритм назначения параметров:

1. Если параметр указан, и он не противоречит остальным параметрам, то он принимается за рабочий.   
   Если он противоречит другим параметрам, имеющимся данным или логике, то он может быть изменен. Например, если при вызове было указано значение даты @Start больше, чем значение даты @Stop, то они будут взаимно заменены.
2. Если параметр не определен, то ищется его значение в таблице XEd\_Set в следующем порядке:
   1. С максимальной детализацией и указанной сессией XE.   
      Например, ищется значение отбора **Records** в XEd\_Set для конкретной таблицы @Table и сессии XE @Session.
   2. Только с указанной сессией XE.
   3. С максимальной детализацией, но без указанной сессии XE.   
      Например, ищется значение отбора Records в XEd\_Set для конкретной таблицы @Table и сессии XE @Session = NULL.
   4. Без указания на детализацию и сессию.
3. Если параметр по-прежнему не определен, то, если это допустимо, он может быть оставлен в значении NULL, или ему может быть присвоено значение по умолчанию, определенное в конкретной процедуре.

Кроме того, при вызове процедур используется параметр @Option, который позволяет изменять стандартный ход обработки. Параметр @Option (nvarchar(256)) может содержать список опций, разделенных одним или несколькими пробелами, при этом каждая опция определяет ту или иную модификацию работы процедуры.

Опции, которые не используются в процедуре игнорируются. Т.о., чтобы сделать опцию недействительной, достаточно, например, изменить опцию **List** на значение **XList**. Или **\List**

Процедура может передавать опции в другую процедуру при ее вызове, при этом опции могут в некоторых случаях модифицироваться.

Опции не чувствительны к регистру.

Если иное не оговорен отдельно, то следующие параметры используются во всех процедурах следующим образом:

| ***Параметр*** | ***Описание*** | ***Замечание*** |
| --- | --- | --- |
| @Session | Значение «Сeссия XE» - имя для набора XE, обрабатываемых и хранимых в одной и той же таблице  ***Пример****:*  *@Session = ’sql251’* | Если параметр не указан, то берется значение по умолчанию из таблицы XEd\_Set  Значение параметра @Sessoin может влиять на остальные параметры выполнения, поэтому он проверяется и устанавливается в процедурах первым. |
| @Source | Иcходные данные для работы процедуры. Это:   * либо путь к файлам .xel (для процедуры XE\_Xel), * либо таблицы, откуда импортируются полные данные (для процедур конвертации)   ***Пример****:*  @Source = N'C:\Документы\Проекты\XE data\sql1250\XE\_Log\_\*.xel'  @Source = '[alidi backup].[dbo].[Log\_XE\_SQL251]' | Если параметр не указан, то последовательно выбираются, пока не будет найдено не нулевое значение:   1. Значение в XEd\_Set для сессии (см. параметр @Session) 2. Значение в XEd\_Set по умолчанию (без указания сессии)   Если параметр @Source не удалось определить, будет выдано сообщение об ошибке и обработка не будет осуществляться. |
| @Table | Определяет имя таблицы для хранения полных данных extended events, т.к. имя таблицы может различным для различных сессий XE  ***Пример****:*  *@Table = 'XE\_SQL251’* | Если параметр не указан, то он выбирается на основании данных в справочнике Xed\_Set |
| @Start,  @Stop | Определяет диапазон дат (начало и конец), данные за который необходимо обработать.  ***Пример****:*  *@Start = '2024-08-06'*  ,@Stop = null | При указании значения null ему будет присвоено, соответственно, самое меньше и самое больше из возможных значений на основании имеющихся данных в файлах .xel и справочника XEd\_Source |
| @Option | Список опций, модифицирующих ход выполнения процедуры, разделенных одним или несколькими пробелами  ***Пример****:*  *@Option = 'notext nodetail \only'* | См. список допустимых опций.  Недействительные опции игнорируются. |

Перечень доступных опций приведен ниже

| **Опция** | **Описание** |
| --- | --- |
| List | В процессе выполнения выводятся дополнительные диагностические данные (такие, например, как актуальные параметры выполнения, информация об отобранных записях и т.п.)  Данный параметр можно использовать для предварительного анализа исходных данных или отладки. |
| Only | Осуществляется отбор данных, выдача диагностических сообщений, однако запись в таблицы и обновление справочников не производится.  Используется совместно с опцией List в диагностических целях. |
| NoClear | При выполнении обновления таблиц и обновлении справочников не выполняется очистка целевых таблиц от архивных данных.  По умолчанию из таблиц и справочников перед обновлением удаляются данные с датой, отстающей от текущей на число дней, определяемых параметрами в таблице XEd\_Set |
| Compact | может применяться для уменьшения объема хранимых данных. В этом случае в таблице XEd\_Hash поле [Text] будет содержать NULL вместо модифицированного текста запроса.  Данный параметр непосредственно в процедуре не обрабатывается, но транслируется при вызове в процедуру XE\_TextData |
| Skip | По умолчанию после записи данных в таблицу полных данных и обновления справочников вызывается процедура XE\_TextDataдля обработки справочников XEd\_Hash, XEd\_Exec, XEd\_Table  Параметр Skip отменяет ее вызов. |

# Основные хранимые процедуры

## Хранимая процедура XE\_Xel

**Назначение**: считывание данных из файлов, записанных extended events и запись в таблицы типа XE

**Допустимые параметры**

| **Опция** | **Описание** | **Замечание** |
| --- | --- | --- |
| @Session | Сессия |  |
| @Source | Путь и шаблон файлов с данными, записанными сессией extended events. |  |
| @Table | Имя таблицы для полных данных |  |
| @Start,  @Stop | Границы интервала отбираемых из @Source данных |  |
| @Option | В качестве значений опций можно указать следующие ключевые значения:   * *List* * *Only* * *NoClear* * *Compact* * *Skip* |  |

**Алгоритм работы**:

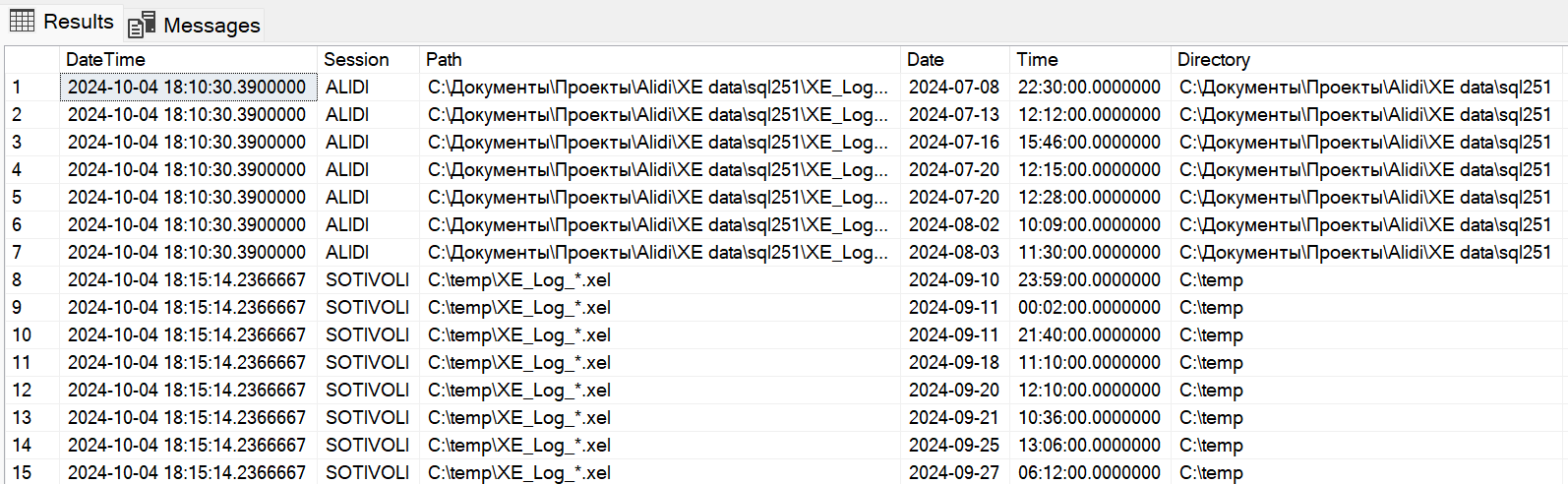
Общая последовательность работы:

1. Из файлов с типом .xel, соответствующих шаблону @Source выбираются только те, которые обновлялись с момента последнего вызова процедуры с тем же параметром @Session.
2. Из отобранных файлов отбираются данные о событиях, которые закончились позднее, чем самое позднее событие из таблицы @Table.   
   При этом записи файлов парсятся из xml в поля временной таблицы, при необходимости преобразуются.
3. На основании отобранных данных обновляются справочники значений полей, а сами поля заменяются на индекс. Индекс имеет имя поля с добавленным символом #.
4. Данные помещаются в таблицу @Table. Если таблица отсутствует, то она создается путем вызова процедуры XE\_Install
5. Если не указана опция *Skip,* то вызывается хранимая процедура XE\_TextData*, которой* передаются параметры @Session и @Option
6. Обновляется справочник считанных файлов XEd\_Source

Если при вызове процедуры была указана опция **List**, то будет выводиться дополнительная информация о работе процедуры.

Процедура обрабатывает исходные данные, созданные сессией extended events.

Путь и шаблон названия файлов определяется параметром @Source, или, если параметр отсутствует – соответствующими значениями в таблице XEd\_Set *(см. параметры* SOURCEсобытия extended events и их параметрах на момент последнего считывания (дата модификации, число строк).



Считываются только те файлы, которые:

* либо не указаны в XEd\_Source
* либо дата обновления или число строк в файле отлично от тех значений, которые зафиксированы на момент последнего чтения в XEd\_Source

Остальные файлы игнорируются, о чем выводится соответствующее сообщение:

Information: Обработка файла XE\_Log\_-\_0\_133649406260170000.xel пропускается

**Рекомендация**: предпочтительно определять путь и шаблон имен файлов через значения, в XEd\_Set, а параметр *@Source* использовать в тех случаях, когда надо добавить данные из других источников, т.к. это упрощает вызов процедуры и уменьшает вероятность ошибок при указании параметра.

### Считывание и фильтрация данных

Фильтрация данных производится в случае, если желаемая дата данных была указана в параметрах @Start *и* @Stop.

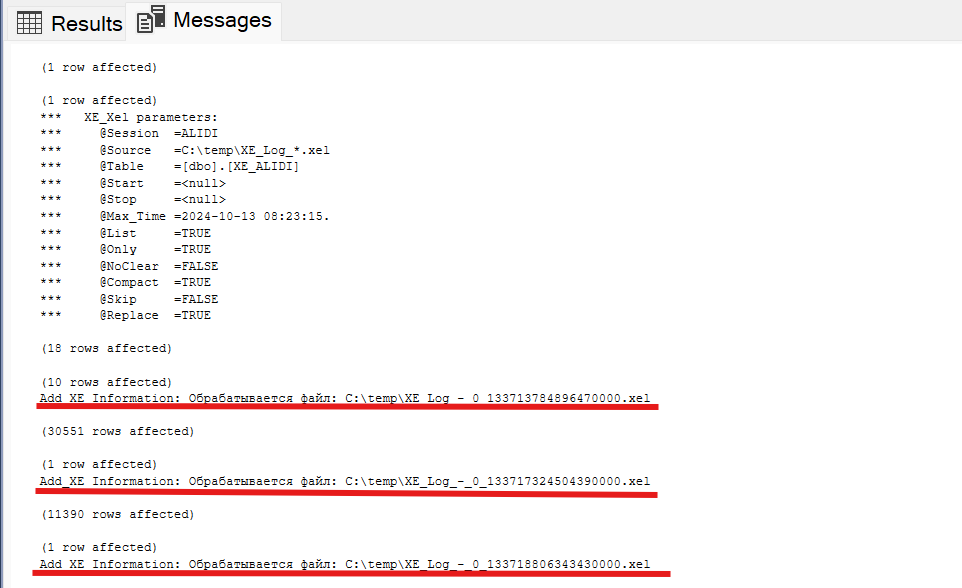
В этом случае будут отобраны только записи о тех событиях, которые **закончились** в указанный диапазон.

При этом если не указана опция **Replace**, то диапазон дополнительно ограничивается снизу наибольшей датой события в @Table*.*

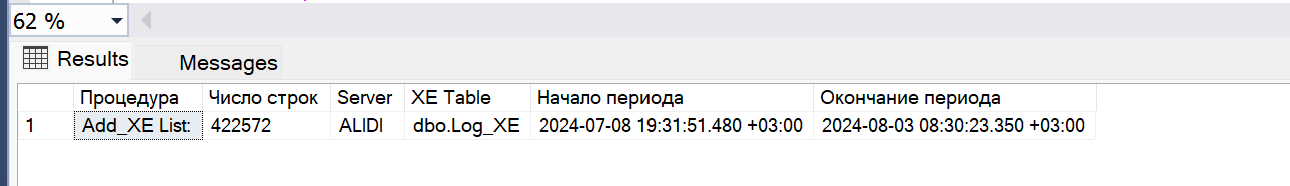
Если указана опция Replace, то выбранные данные замещаются в @Table т.е. предварительно в таблицу @Table удаляются все данные в диапазоне отобранных дат из файлов @Source.

### Вывод диагностической информации:

При указании опции **List** выводится следующая диагностическая информация:

1. Параметры вызова процедуры
2. Информация об обрабатываемых и пропускаемых файлах  
   
3. Через select выводятся параметры считанных файлов

При указании опции *Only* запись в таблицу *Log\_XE* и в справочник не производится.



### Обновление справочников полей

На основании считанных данных производится обновление полей в следующих справочниках

|  |  |  |  |
| --- | --- | --- | --- |
| **Таблица** | **Поля** | **Замечание** | |
| XEd\_Application | [Application] | | Поле Client\_app\_name события extended events |
| XEd\_Database | [Database\_name]  [DatabaseID] | | Пара этих значений образуют уникальный ключ |
| XEd\_Event | [Session] – сервер XE  [Package] – пакет событий  [Event] – тип события | | Три этих значения образуют уникальный ключ |
| XEd\_Hash | [Hash] - хеш, расчитанный по полю TextData  [TextData] - текст запроса | | 1. Хеш расчитывается как HASHBYTES('SHA2\_256',[TextData]) 2. Остальные поля справочника заполняет процедура Add\_Detail |
| XEd\_Host | [HostName] | |  |
| XEd\_Output | [Output\_parameters] | |  |
| XEd\_Result | [Result\_Code]  [Result\_Text] | | Пара этих значений образуют уникальный ключ |
| XEd\_Session | [Session] | | сессия XE |
| XEd\_Source | [DateTime]  [Session]  [Source]  [Date]  [Time]  [Directory]  [File\_Name]  [Records]  [Events] | | Данные о считанных файлах .xel |
| XEd\_User | Разделенное на части поле *Nt\_username* события:  [Nt\_AD]  [Nt\_username]  Разделенное на части поле *LoginName* события:  [Login\_AD]  [LoginName] | | Четыре этих значения образуют уникальный ключ |

Кроме того, во всех справочных таблицах обновляются следующие поля:

* [Date] – самая свежая дата события, где присутствовало данное значение
* [First\_Date] – самое первая дата события
* [Cnt] - число событий, нарастающим итогом, где присутствует данное значение

Данные поля позволяют оценить интегральные значения по значениям параметров

### Обновление Dic\_Source

В справочнике XEd\_Source производится обновление и / или добавление информации о файлах, которые считывались. Это предотвращает повторное считывание и дублирование информации.

## Хранимая процедура XE\_TopSum

**Назначение**: формирование за один день данных суммарного потребления ресурсов (в таблицу XE\_Sum) и топ запросов (в таблицу XE\_Top) из таблицы полных данных @Table, сформированных процедурой XE\_Xel

Типичное использование процедуры XE\_Day – ежедневная обработка данных, накопленных в таблицах с полными данными.

Поскольку во всех процедурах XE анализ идет по дате и времени завершения запросов, то данную программу можно выполнять сразу непосредственно по окончании суток, предварительно выполнив вызов XE\_Xel.

Так как таблицы XE\_Top и XE\_Sum являются общими для всех сессий XE (в отличии от таблиц с полными данными, которые индивидуальны для каждой из сессий), то записи имеют индекс сессии [Session#] в каждой строке.

Параметры:

| **Опция** | **Описание** | **Замечание** |
| --- | --- | --- |
| @Session |  |  |
| @Start,  @Stop | Начало и конец периода данных |  |
| @Table | Имя таблицы с полными данными |  |
|  | Число отбираемых топ запросов | По умолчанию берется значение из XEd\_Set |
|  | %% от общего потребления ресурса за день для запроса | По умолчанию берется значение из XEd\_Set |
|  | %% от общего потребления ресурса за день – нарастающим итогом наиболее агрессивных запросов | По умолчанию берется значение из XEd\_Set |
|  | Доступны опции:   * List * Only |  |

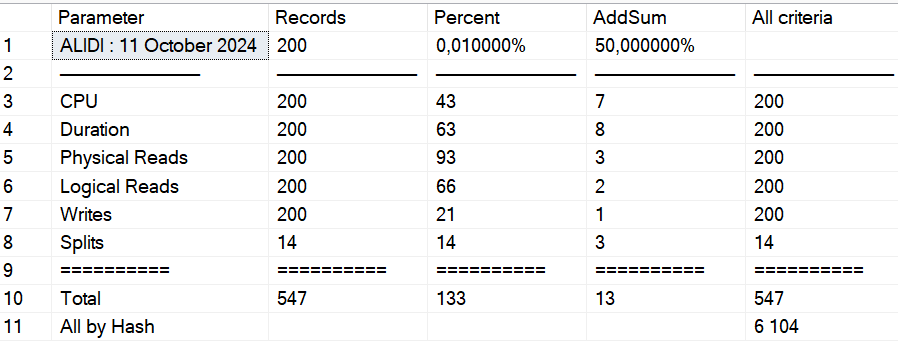
Алгоритм работы:

1. Для указанного диапазона дат считываются данные за день из файла полных данных, определяемых параметрами @Session и @Table
2. Вызывается процедура, определенная в XEd\_Set (параметр OrdersProc), которая обращается к бизнес-системе и возвращает целое число (например, число запросов, обработанных JDE за день). Это значение вместе с суммарным потреблением ресурсов заносится в строку таблицы Log\_Sum.
3. Для каждой отобранной из таблицы с полными данными строки определяются параметры (ранг, процент потребления, процент потребления нарастающим итогом по рангу) для следующих ресурсов:
   1. [CPU]
   2. [Duration]
   3. [Reads\_physical]
   4. [Reads\_logical]
   5. [Writes]
   6. [Spills]
4. Из справочника XEd\_Set считываются и применяются параметры отбора:
   1. Records - число топ строк
   2. Percent - % использования ресурсов от общедневного
   3. AddSum - % использования ресурсов от общедневного, нарастающим итогом для наиболее агрессивных запросов
5. В таблицу XE\_Top помещаются записи о запросах, которые:
   1. Удовлетворяют одному из критериев отбора
   2. Имеют хеш, для которого есть хотя бы один запрос, удовлетворяющий критериям отбора

Очевидно, что критерии отбора образуют декартово произведение 3 x 6 = 18 критериев. При этом отдельные запросы могут удовлетворять одновременно различенным критериям.

Для того, чтобы определить, как критерии определяют число отобранных строк в таблицу XE\_Top, нужно запустить процедуру с параметром List и, если не надо обновлять данные – также и с параметром Only.

Например, вызов процедуры XE\_TopSum @Option = ‘List only’ выведет следующую информацию:

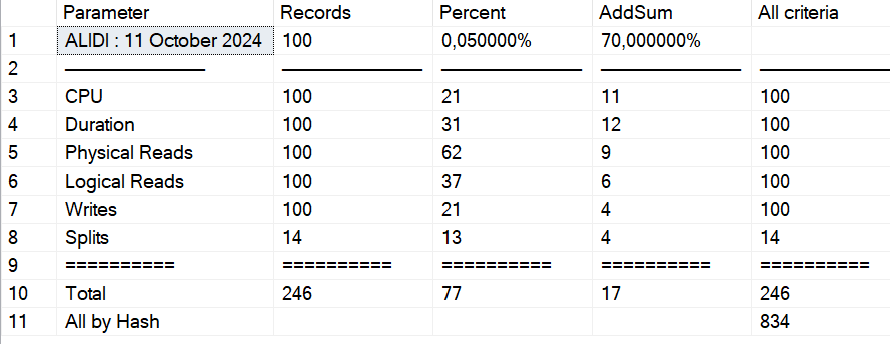


Видно, что в этом случае 547 строк было отобрано по критериям, и еще 6 104 – 547 записей – по хэшу, совпадающему с отобранными по критериям запросами.

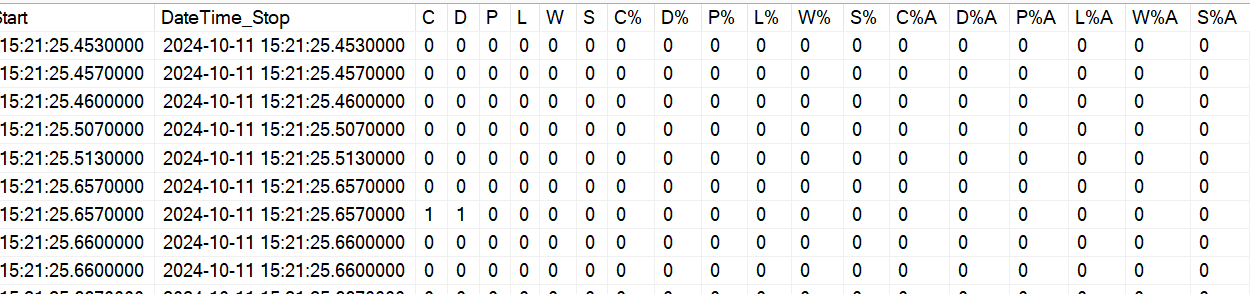
Чтобы подобрать параметры, можно вызывать процедуру, указывая параметры явно, например:

*exec [dbo].[XE\_Day] @Option='List only'  
 ,@Records = 100  
 ,@Percent = 0.05  
 ,@Addsum = 70*

Сравним результаты отбора:



Строки в XE\_Top дополнены флагами (битовые величины), определяющими критерий их отбора.



Если все флаги равны 0, то непосредственно эта строка не соответствует критериям отбора, однако есть запрос с тем же хэшем, который соответствует хотя бы одному из критериев отбора.

Наименование флагов критериев отбора следующее:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Ресурс*** | ***Отбор по Records*** | ***Отбор по Percent*** | ***Отбор по AddSum*** |
| [CPU] | C | C% | C%A |
| [Duration] | D | D% | D%A |
| [Physical Reads] | P | P% | P%A |
| [Logical Reads] | L | L% | L%A |
| [Writes] | W | W% | W%A |
| [Spills] | S | S% | S%A |

### Хранимые процедуры XE\_Count\_JDE и XE\_Count\_Dummy

Для каждой сессии можно в строку файла Log\_Sum записывать информацию от бизнес-системы в поле [Orders] (bigint). Файл XE\_Count\_JDE содержит код для JDE (после инсталляции необходимо удалить комментарий и проверить текст запроса.

XE\_Count\_Dummy – шаблон для создания таких процедур для других бизнес-систем.

То, какая процедура будет вызываться (и будет ли вызываться) для каждой из сессий определяется в справочнике XEd\_Set

Параметры:

| **Опция** | **Описание** | **Замечание** |
| --- | --- | --- |
| *@Date* | Дата, за которую запрашивается получение информации |  |
| *@Orders* | Число (bigint) – возвращаемое число |  |

## Хранимая процедура XE\_Daily

**Назначение**: Ежедневная пакетная обработка данных XE. Включает в себя вызов всех необходимых регулярных процедур

| **Опция** | **Описание** | **Замечание** |
| --- | --- | --- |
| @Session |  |  |
| @Source | Для вызова процедуры XE\_Xel | *См. описание* XE\_Xel |
| @Table | Для вызова процедуры XE\_Xel | *См. описание XE\_Xel* |
| @Steps | Перечень выполняемых шагов, разделенные пробелом:   * Xel * TextData * TopSum | *NULLl соответствует выполнению всех шагов* |
| @Option | Любые параметры для процедур XE\_Xel, XE\_TextData, XE\_TopSum |  |

Вызов процедур осуществляется через служебную процедуру XE\_ExecLog ,которая делает соответствующую запись в журнале выполнения XEd\_Log.

# Вспомогательные процедуры

## Хранимая процедура XE\_Day

Используется процедурой XE\_TopSum для обработки данных за один день.

Т.к. в процедуре отсутствует проверка корректности параметров, она может быть использована для обработки данных в случаях, когда это не позволяет осуществить XE\_TopSum

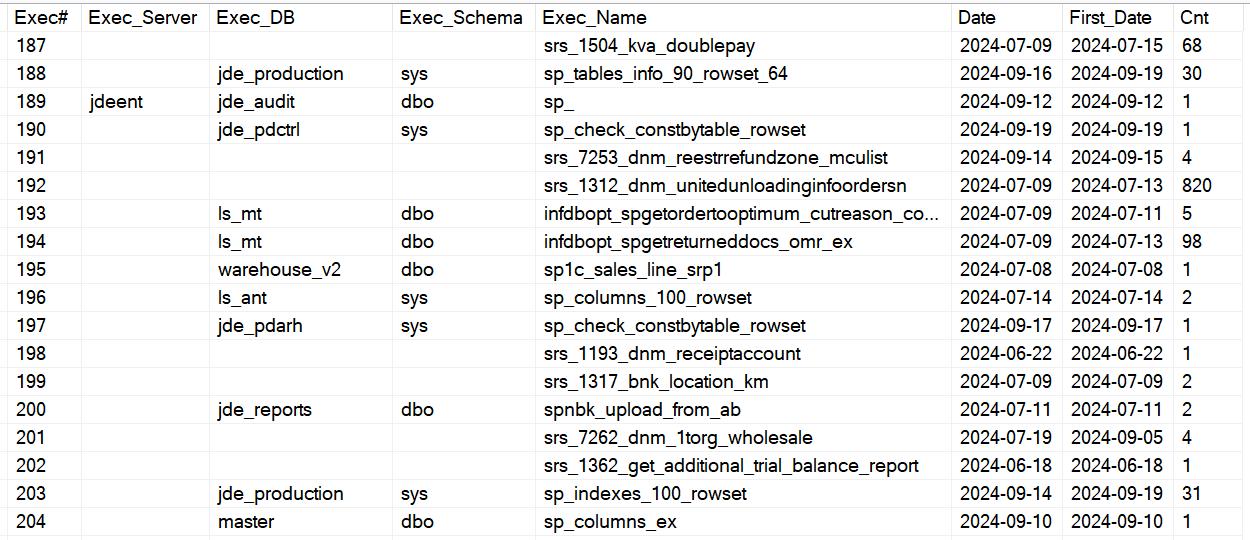
Параметры: такие же, как и у XE\_TopSum, за исключением того, что вместо параметров начала и конца периода времени (@Start и @Stop) используется параметр даты обработки @Date

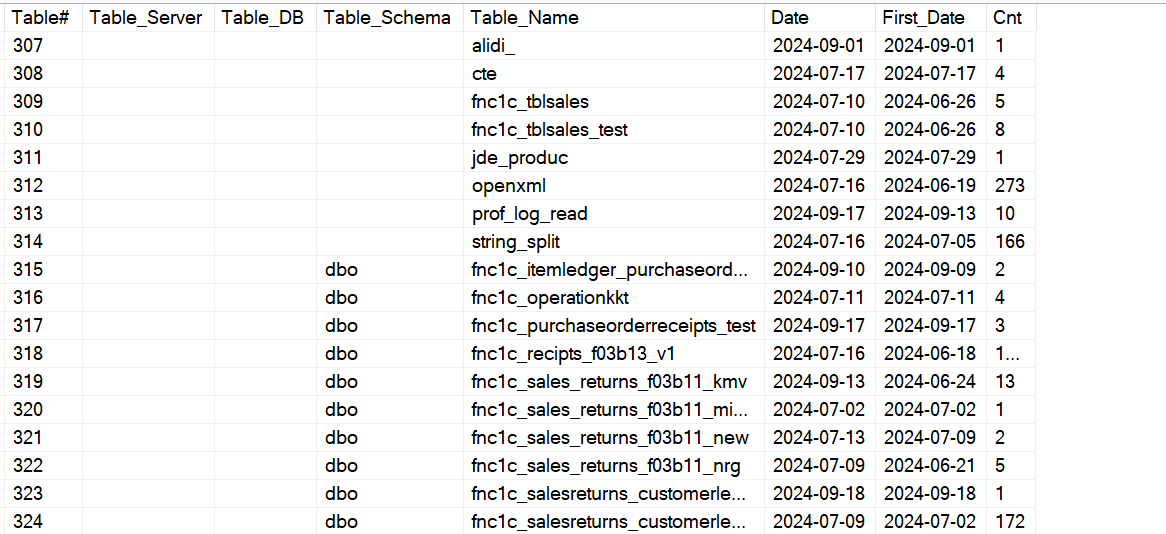
## Хранимая процедура XE\_TextData

**Назначение**: выполняет анализ полей TextData, вычленяя из него данные о первой вызванной процедуре и первой таблице, указанной после ключевого слова FROM.

| **Опция** | **Описание** | **Замечание** |
| --- | --- | --- |
| @Session |  |  |
| @Option | Допустимые опции:   * List * Only * NoClear * Compact |  |

На основании анализа текста вычленяется первая вызванная процедура и первая считанная таблица. Данные помещаются в соответствующие поля и справочники XEd\_Exec и XEd\_Table





Процедура неявно вызывается из процедуры XE\_Xel (если не указана опция Skip)

Алгоритм работы:

1. Из справочника XEd\_Hash считываются строки, у которых не заполнено поле [HashText] (имеет значение null)
2. Поле [TextData] корректируется следующим образом:
   1. Удаляются все однострочные комментарии типа   
      --текст комментария
   2. Удаляются все комментарии вида

/\* комментарий \*/

* 1. Все служебные символы (табуляция, возврат каретки, перевод строки)
  2. Все группы пробелов заменяются на единичный пробел
  3. Проводится ряд модификаций, не меняющих смысл запроса, но делающий однозначным его трактование
  4. Выделяется имя вызываемой процедуры (если таковая есть)
  5. Выделяется имя читаемой таблицы (если таковая есть)
  6. Обновляется справочник процедур, в строку вставляется индекс на него
  7. Обновляется справочник таблиц, в строку вставляется индекс на него
  8. Если не был задан параметр Compact, то в строку XEd\_Hash записывается скорректированный текст запроса [Text]
  9. записывается хеш поля [Text]

## Хранимая процедура XE\_Convert

**Назначение**: считывание данных из импортируемых таблиц, организация цикла по интервалам и вызов соответствующих процедур, осуществляющих импорт одного интервала: XE\_Import\_XE44 и XE\_Import\_Profiler

Тип конвертации (из таблиц XE версии 4.4 или из таблиц данных profiler) определяется соответствующей параметром @Action.

**Параметры:**

| **Опция** | **Описание** | **Замечание** |
| --- | --- | --- |
| @Action | Определяет тип конвертации | **Profiler** – данные берутся из таблиц с данными из профайлера  **XE44** – данные берутся из таблиц с данными XE версии 4.4 |
| @Session |  |  |
| @Source | Исходные данные  **Пример:**  @Source = '[alidi backup].[dbo].[Prof\_log\_251\_char\_View]' | *Наименование таблицы, в которой находятся данные для конвертации.* |
| @Table | Место хранения данных | Как правило, но не обязательно – таблица для хранения полных данных (значение @Table *для процедуры* XE\_Xel |
| @Uncertainty | Неопределённость начала и конца интервала времени (см. описание ниже). | По умолчанию – 0 секунд |
| @Option | Допустимые опции:   * @List * @Only * @Profiler * @XE * @Replace * @Skip | ***XE*** *–* указывает, что конвертируются данные XE версии 4.4  ***Profiler***– указывает, что конвертируются данные от *Profiler*  ***@Replace*** *–* данные в *@Table* замещаются данными конвертации |

Алгоритм работы:

Данные считываются и анализируются подневно на их пересечение с данными, которые уже есть в таблице @Table.

В течении каждого из дней возможны разные варианты пересечения данных в конвертируемо таблице @Source и в таблице назначения @Table – см. схемы ниже.

Для каждого случая выбирается от 0 до 2 интервалов, которые импортируются в @Table за этот день.

Т.к. в конвертируемой системе и в системе, куда импортируются данные могут иметь место несинхронности во времени, то можно задать расширение интервалов данных в @Table на n секунд – это значение задается параметром @Uncertainty*. По умолчанию* @Uncertainty *= 0 секунд*

| ***Варианты пересечения интервалов*** | ***Импортируемые временные промежутки*** |
| --- | --- |
|  | В @Table уже есть данные за весь период в текущий день.  Нет интервалов для импорта, импорт данного дня пропускается |
|  | Производится импорт с начала суток до времени, начиная с которого данные присутствуют в @Table |
|  | Производится импорт данных со времени, начиная с которого в @Table гарантированно отсутствуют данные и до конца текущих суток |
|  | Импортируется два интервала – комбинированный случай предыдущих двух |
|  | Данные в @Table за данные сутки отсутствуют.  Импортируются данные за все сутки |

Процедура делает цикл по всем датам, начиная с наименьшей даты с данными из @Source и кончая наибольшей датой данных в @Source.

Определив интервалы импорта, процедура вызывает для, собственно, обработки данных одну из двух процедур импорта: XE\_Import\_XE44 для данных XE версии 4.4 или XE\_Import\_Profiler для импорта данных из таблицы профайлера

**Замечание**: для импорта данных из таблицы XE версии 4.4 необходимо, чтобы вместе с таблицей, указанной в @Source существовали таже и все справочники версии 4.4 (кроме Dic\_Exec и Dic\_Table), т.к. процедура восстанавливает значения всех полей и по ним заново обновляет поля в справочниках XE версии 5.0  
Если не соблюсти это требование, то часть или все ссылки на поля будут отсутствовать (значение 0), а при выполнении будут выдаваться ошибки о невозможности записи значений NULL в справочники на основании существующих constraints.

## Хранимая процедура BaReTa (Backup Restore Tables)

**Назначение**: Копирование набора таблиц (временное сохранение)

Позволяет делать резервные копии таблиц, их восстанавливать, либо очищать резервные копии. Действие определяется значением параметра @Action

Перечень исходных таблиц указывается через пробел или группу пробелов в параметре @Table

Таблицы копии создаются в соответствии с параметром @Prefix.

В простейшем случае @Prefix задает префикс, который добавляется к имени таблицы для копий.

Если префикс указан в виде квалифицированного имени, то копии создаются в базе и схеме, указанные в @Prefix, имя по-прежнему содержит префикс.

Например, команда

Prefix @Table=’a b c’, @Action=’Backup’, @Prefix=’Copy\_’

выполнит следующее копирование:

A команда

Prefix @Table=’abc’, @Action=’Backup’, @Prefix=’[INT\_DEM].[dbo].[Copy\_]’

Выполнит копирование

| **Опция** | **Описание** | **Замечание** |
| --- | --- | --- |
| *@Action* | Одно из действий:   * Backup * Restore * Clear | **Backup** – копирование таблиц из списка @Table в место и имя, определенное @Prefix  **Restore** – восстановление в таблицы списка @Table из копии, определенной @Prefix  **Clear** – удаление таблиц копии, заданной @Prefix |
| *@Prefix* | Задает префикс для таблиц резервного копирования – см. выше |  |
| *@Table* | Список исходных таблиц |  |
| *@Option* | Доступные опции:   * List * Only * Replace | Для замещения имеющихся таблиц необходимо использовать опцию Replace |

# Служебные процедуры

## Хранимые процедуры XE\_Import\_XE44, XE\_Import\_Profiler

Технические процедуры импорта данных за указанный период из таблиц данных XE версии 4.4 (XE\_Import\_XE44) или профайлера (XE\_Imort\_Profiler)

Параметры вызова: такие же, как у процедуры XE\_Convert

## Хранимая процедура XE\_ExecLog

**Назначение**: выполнение процедур XE с регистрацией их параметров и хода выполнения в журнале XEd\_Log

**Параметры:**

| **Опция** | **Описание** | **Замечание** |
| --- | --- | --- |
| @Proc | Имя вызываемой процедуры |  |
| @Caller | Имя того модуля, откуда происходит вызов | Как правило, используется выражение @Caller = @Version |
| @Error | Если передано значение @Error, не равное NULL, то вызов процедуры @Proc не осуществляется, но делается запись о ее выполнении с ошибкой @@Error = @Error |  |
| Параметры, передаваемые процедуре @Proc | Допустимые параметры, передаваемые процедуре @Proc:   * @Option * @Comm * @Table * @Session * @Date * @Records * @Percent * @AddSum * @Start * @Stop * @Source * @Offset * @Orders * @OrdersProc * @Steps |  |

## Хранимая процедура XE\_CheckClear

Очистка таблицы @Table от архивных данных (на основании параметров из справочника XEd\_Set)

| **Опция** | **Описание** | **Замечание** |
| --- | --- | --- |
| @Session |  |  |
| @Table | Имя таблицы, архивные данные которой необходимо удалить |  |
| @Option | Допустимые опции:   * List * Only |  |

## Хранимая процедура XE\_CheckParams

**Назначение**: Вспомогательна процедура для модификации входных параметров:

* Установка значения по умолчанию, если параметра null
* Приведение параметра к стандартной для него форме

Может использоваться всеми процедурами XE.

**Параметры** определяют входные параметры, которые надо проверить и выполнить операции по подстановке значения по умолчанию или приведению параметров к стандартному виду.

Модификатор output при вызове говорит о том, что надо вернуть скорректированное значение и является обязательным для каждого параметра.

Пример вызова:

exec [dbo].[XE\_CheckParams]

@Session = @Session output

,@Table = @Table output

,@Source = @Sourc output

При определении значений по умолчанию берутся значения из таблицы XEd\_Set. При этом значения полей в этой таблице null и ‘’ (пусто) трактуются как «неуказанные»

| **Опция** | **Назначение** | **Значение по умолчанию, приведение к стандартному виду** |
| --- | --- | --- |
| ***@Session*** |  |  |
| ***@Session#*** | Id сессии @Session (после коррекции) | Значение поля [Session#] в справочнике XEd\_Session для сервера XE @Session  Для поиска используется значение @Session после корректировки |
| @Table | Имя таблицы для записи полных данных из файлов .xel для сессии XE @Session | Берется последовательно ненулевое значение [Object] из таблицы XEd\_Set:   1. Значение для сессии [Session]=@Session 2. Значение для неуказанного [Session]   ‘XE\_Xel’  Значение приводится к квалифицированному имени таблицы с указанием БД (если известно), схемы (по умолчанию [dbo]) и имени таблицы   1. Если имя таблицы не начинается на XE\_, то к нему добавляется этот префикс 2. Имя таблицы переводится в верхний регистр |
| @Source | Путь и шаблон файлов типа .xel | Берется последовательно ненулевое значение [Object] из таблицы XEd\_Set:   1. Значение для сервера [Session]=@Session 2. Значение для неуказанного [Session] |
| @OrdersProc | Имя процедуры, определяющей число заказов бизнес-приложения за указанный день | Берется последовательно ненулевое значение [Object] из таблицы XEd\_Set:   1. Значение для сервера [Session]=@Session 2. Значение для неуказанного [Session] |
| @Records  @Percent  @AddSum | Параметры отбора top запросов за день. | Берется последовательно ненулевое значение [Object] из таблицы XEd\_Set для строки с оответствующим значением в столбце [Parameter]:   1. Значение для сервера [Session]=@Session и таблицы [Object] = @Table 2. Значение для для сервера [Session]=@Session 3. Значение для таблицы [Object] = @Table 4. Значение с неуказанных @Session и @Table |
| @Object | Имя объекта, для которого ведется поиск значений @Records, @Percent, @AddSum | Значение приводится к квалифицированному виду, при этом имя таблицы для записи полных данных формируются в заглавных буквах.  Например |

# Справочник XEd\_Set

Структура справочника приводится ниже. Строки могут следовать в произвольном порядке.

При наличии дублирующих значений числовое значение будет выбираться максимальным.

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Session** | **Object** | **Value** |
| *NULL* | ***=-- Сессия по умолчанию*** | *NULL* | *NULL* |
| Session | Alidi | NULL | NULL |

Если при вызове процедур параметр @Session не будет указан, то его значением по умолчанию будет @Session = ‘ALIDI’ (имя сессии преобразуется в заглавные символы)

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Session** | **Object** | **Value** |
| *NULL* | ***=-- Параметры отбора в XE\_Top*** | *NULL* | *NULL* |
| Records | NULL | NULL | 200.000000 |
| Percent | NULL | NULL | 0.010000 |
| AddSum | NULL | NULL | 50.000000 |
| Records | sotivoli | XE\_sotivoli | 300.000000 |

Для сессии SOTIVOLI значением @Records для отбора по умолчанию будет значение 300, а для всех остальных сессий – 200

Значение по умолчанию для всех сессий для @Percent будет 0,01 %, а для @AddSum – 50%

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Session** | **Object** | **Value** |
| *NULL* | ***=-- Путь к файлам данных XE*** | *NULL* | *NULL* |
| Source | NULL | C:\temp\XE\_Log\_\*.xel | NULL |
| Source | Alidi | C:\Data\XE\_Log\_\*.xel | NULL |

Для сессии ALIDI путь к файлам .xel будет равен ‘C:\Data\XE\_Log\_\*.xel’, для всех остальных сессий – ‘C:\temp\XE\_Log\_\*.xel’

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Session** | **Object** | **Value** |
| *NULL* | ***=-- Таблица полных данных*** | *NULL* | *NULL* |
| xel | NULL | XE\_Xel | 400.000000 |
| xel | Alidi | [dbo].[XE\_Alidi] | 400.000000 |
| xel | sotivoli | XE\_sotivoli | 1000.000000 |

Таблица полных данных для сессии SOTIVOLI по умолчанию будет XE\_SOTIVOLI,

Для сессии ALIDI – [dbo].[XE\_ALIDI], для остальных сессий = XE\_XEL

Замечание: во избежание путаницы имя полных таблиц всегда переводится в заглавные буквы. Однако имя схемы или базы данных остается чувствительным к регистру.

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Session** | **Object** | **Value** |
| *NULL* | ***=-- Минимальный срок хранения данных в справочниках*** | *NULL* | *NULL* |
| Offset | NULL | NULL | 400.000000 |
| Offset | Alidi | NULL | 200.000000 |
| Offset | sotivoli | XEd\_Hash | 3650.000000 |

Максимальный срок хранения данных в таблице XEd\_Hash для сессии SOTIVOLI составит 3 650 дней,

Для всех таблиц сессии Alidi - не менее 200 дней, для всех остальных сессии – 400 дней.

Однако, для сохранения целостности, данные в справочниках хранятся не менее максимального значения срока хранения, поэтому все справочники будут хранить информацию не 200 или 400 дней, а 3 650 дней.

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Session** | **Object** | **Value** |
| *NULL* | ***-- Имя процедуры, возращающей @Orders*** | *NULL* | *NULL* |
| OrdersProc | NULL | NULL | NULL |
| OrdersProc | Alidi | XE\_Count\_JDE | NULL |

Для сессии ALIDI в качестве процедуры, определяющей поле [Orders], будет использоваться процедура XE\_Count\_JDE, для всех остальных сессий процедура не будет вызываться.

# Анализ данных XE

## Представления для таблиц XE

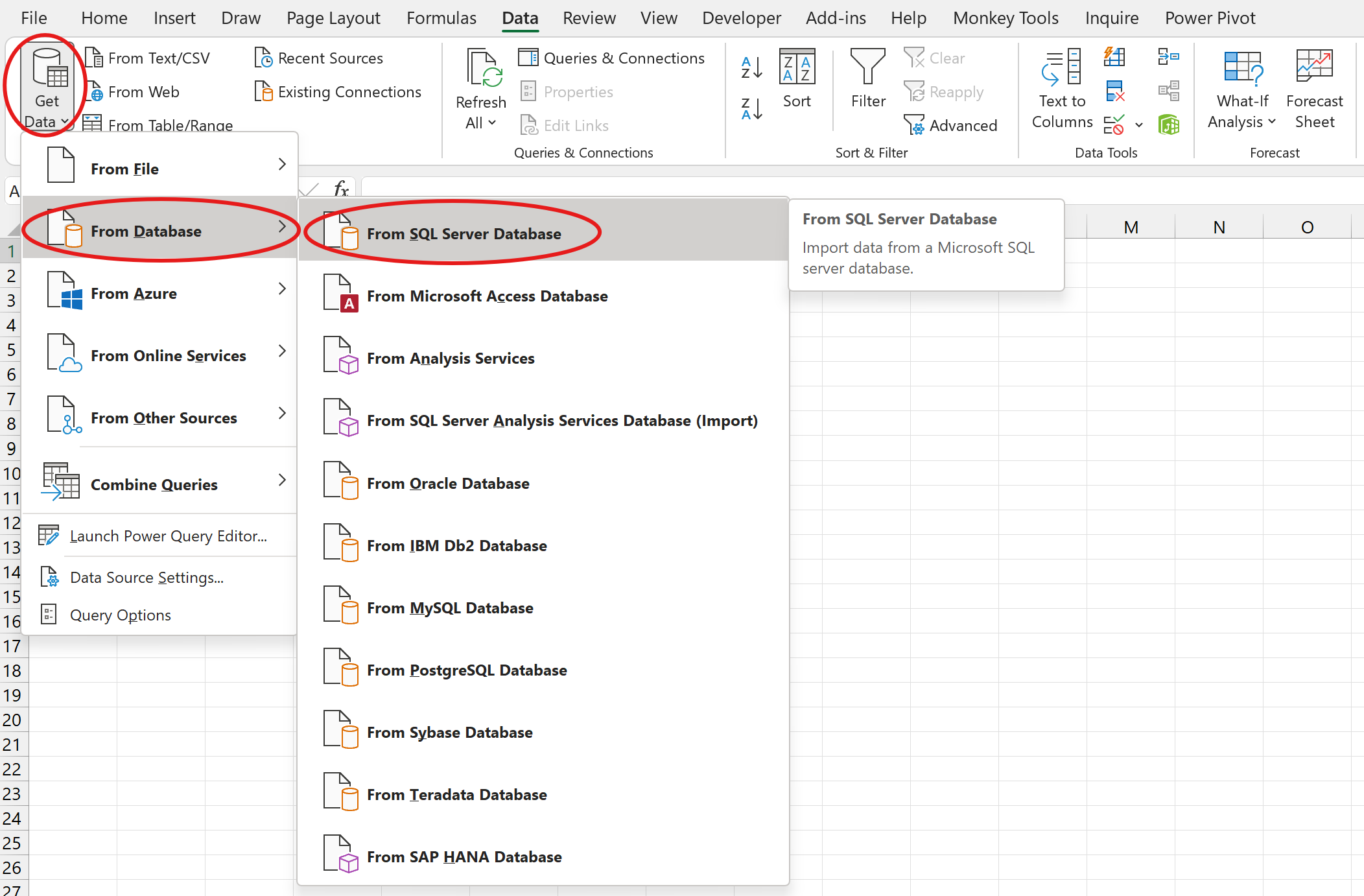
Для доступа к данным можно использовать представления (View), которые создаются процедурой XE\_Install для следующих объектов:

|  |  |
| --- | --- |
| Таблица | Представление |
| XE\_Top | XEv\_Top |
| Таблицы полных данных типа XE\_ИМЯ  Пример:  *XE\_SQL251*  *XE\_ALIDI* | Xev\_ИМЯ  Создаются автоматически при создании таблицы для полных данных  Пример:  XEv\_SQL251  XEv\_ALIDI |

Преимуществом представлений является то, что в них таблицы данных объединены со справочниками

## Доступ к данным из Excel и из Power BI

Для доступа к данным из Excel необходимо в меню Data выбрать From Database > From SQL Server Database



Далее возможно два варианта действия:

**Вариант 1: доступ к таблицам**

В окне выбора необходимо ввести Имя базы данных.



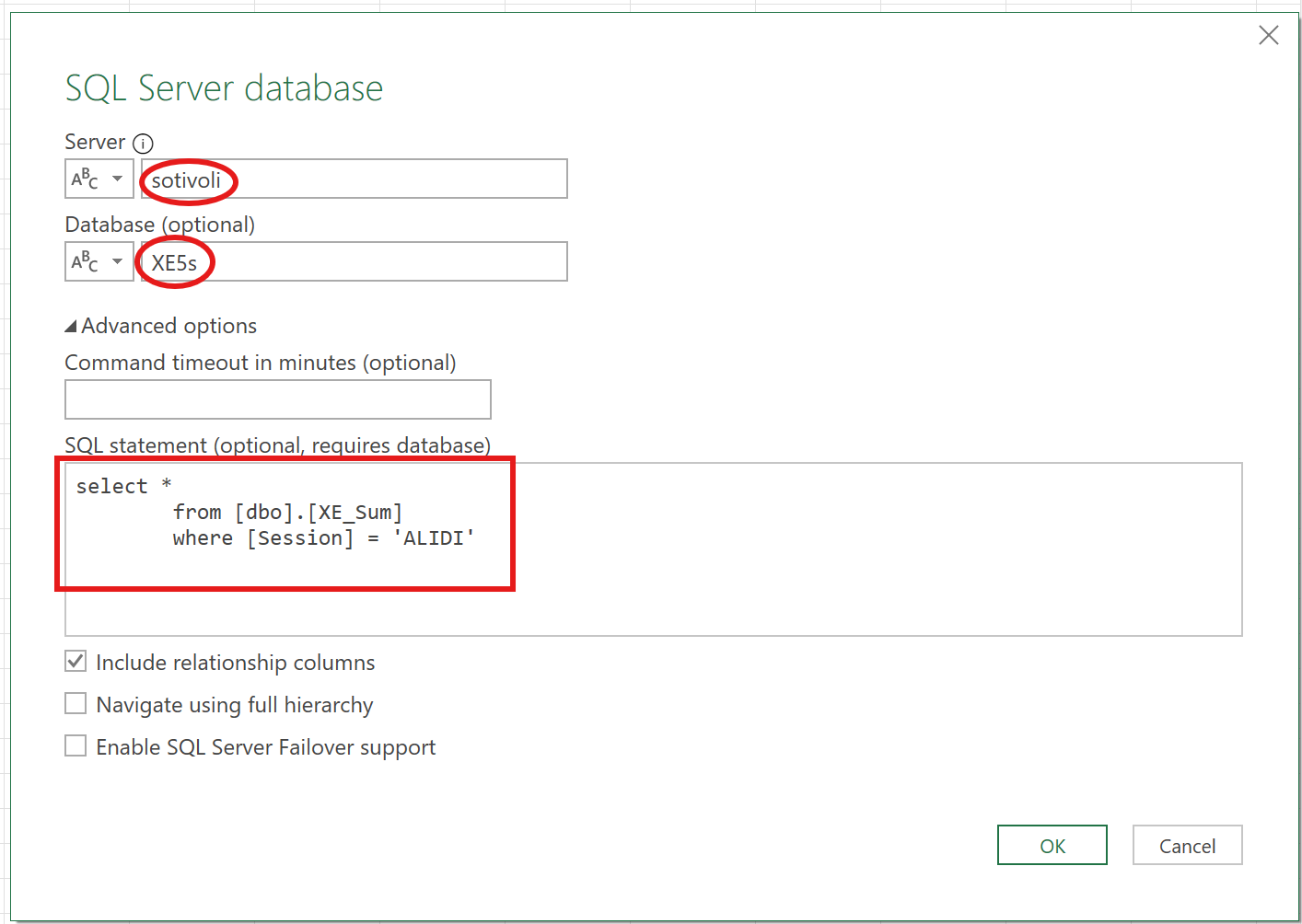
Далее в окне навигатора выбрать базу и таблицу, после чего нажать на клавишу Загрузить

**Вариант 2: выборка данных посредством Select:**

В окне выбора базы данных выбрать сервер, базу данных

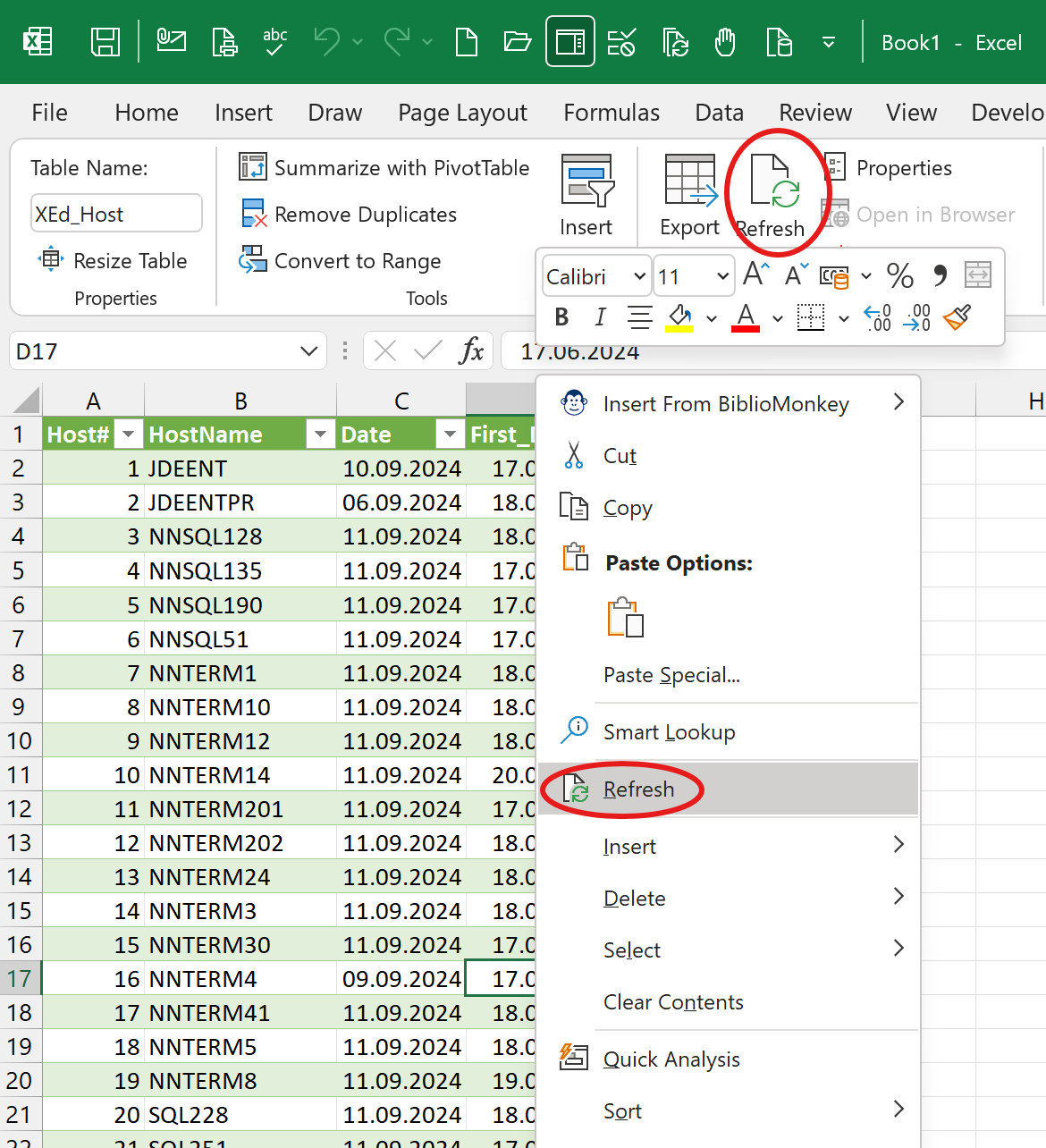
Раскрыть меню Advanced options и в поле SQL Statement ввести требуемую команду Select

Нажать OK

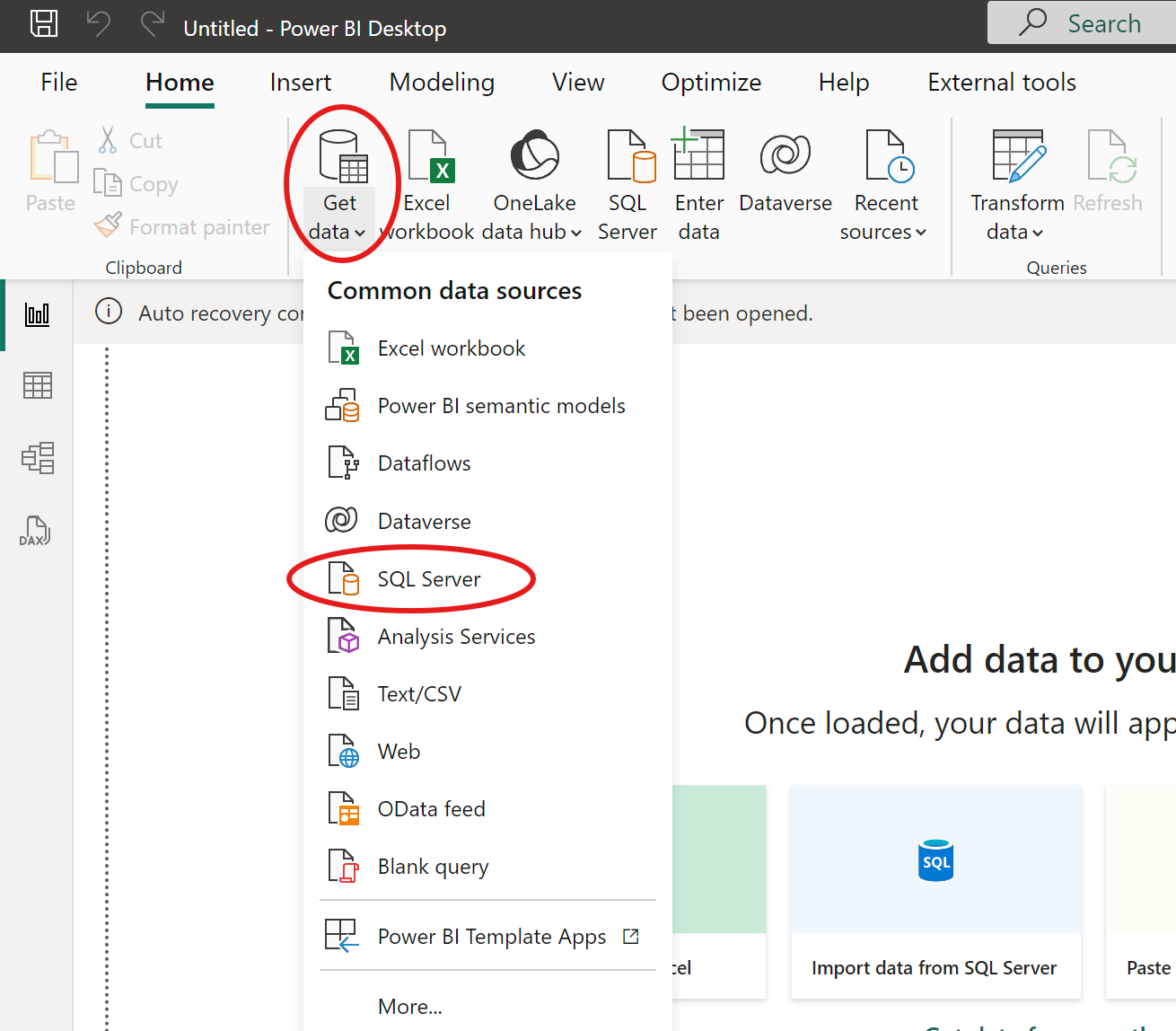
****

После выполнения запроса появится окно с данными. Если все устраивает, то нажать OK

Полученные данные можно обновить в любой момент, нажав либо клавишу в меню Data, либо такую же клавишу в контекстном меню – будет повторно выполнен оператор Select

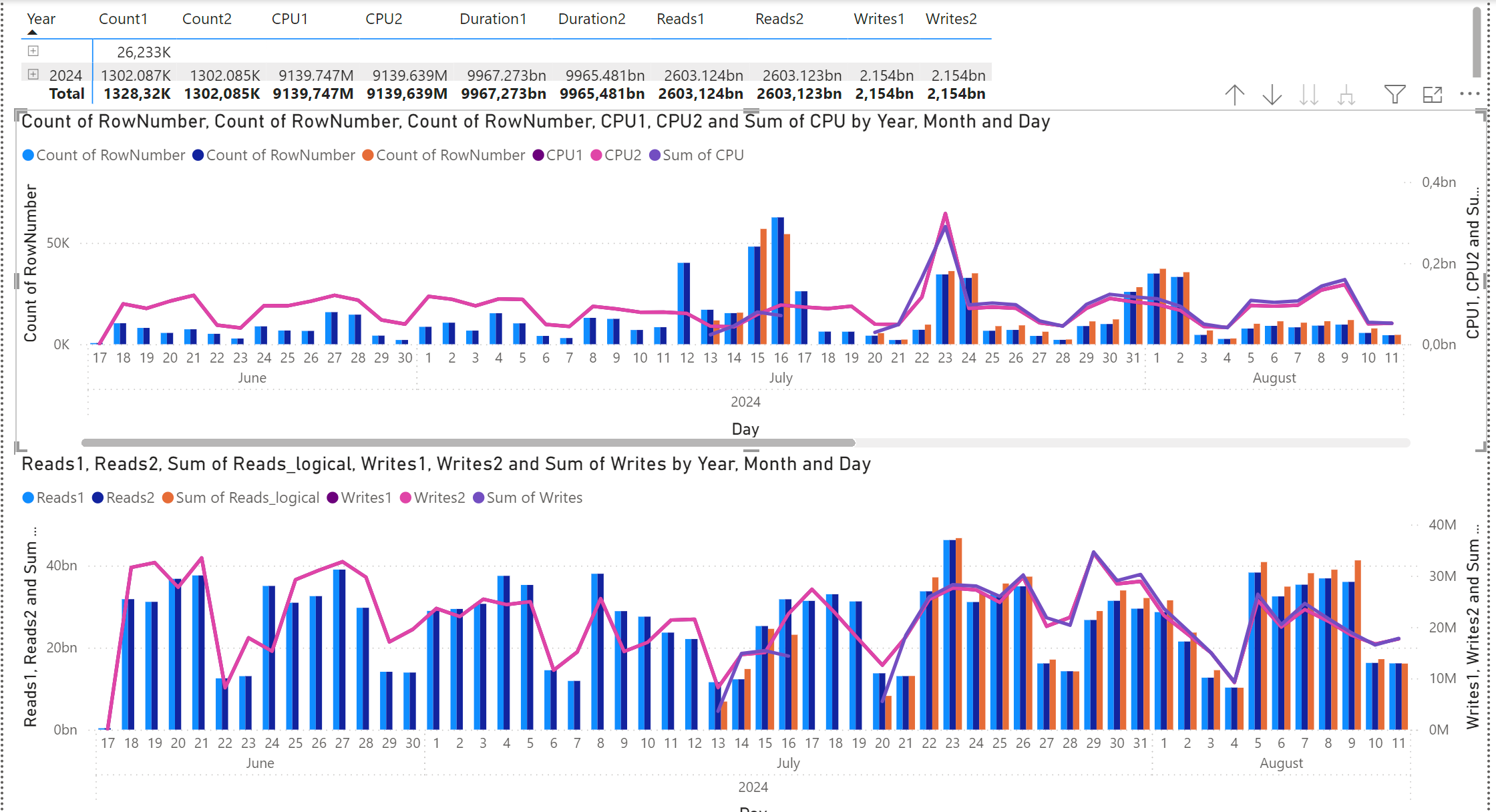


Аналогично вводятся данные и в Power BI:



# Приложение 1. Сравнение данных конвертации

Данные сравнения конвертации данных из profiler и XE версии 4.4 приведены ниже.



1. См. <https://learn.microsoft.com/ru-ru/sql/relational-databases/event-classes/exchange-spill-event-class?view=sql-server-ver16> [↑](#footnote-ref-1)