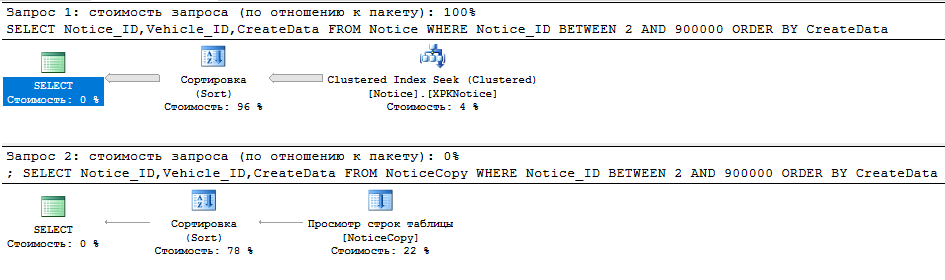
# Просмотр и интерпретация плана выполнения запросов, использование статистики

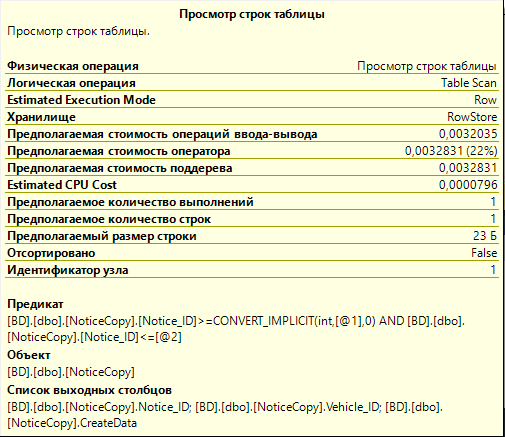
Просмотр плана запросов будем проводить на примере исследования влияния наличия кластерного индекса в таблице на обработку запроса.

В исследовании используются две таблицы с одинаковыми столбцами и строками.

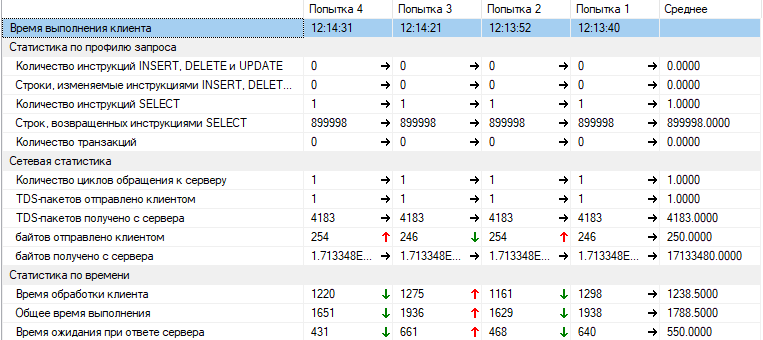
SELECT Notice\_ID,Vehicle\_ID,CreateData FROM Notice WHERE Notice\_ID BETWEEN 2 AND 900000 ORDER BY CreateData;

SELECT Notice\_ID,Vehicle\_ID,CreateData FROM NoticeCopy WHERE Notice\_ID BETWEEN 2 AND 900000 ORDER BY CreateData;





Ниже приведена статистика где 1 и 3 запросы выполнялись с кластеризованым индексом, остальные без.

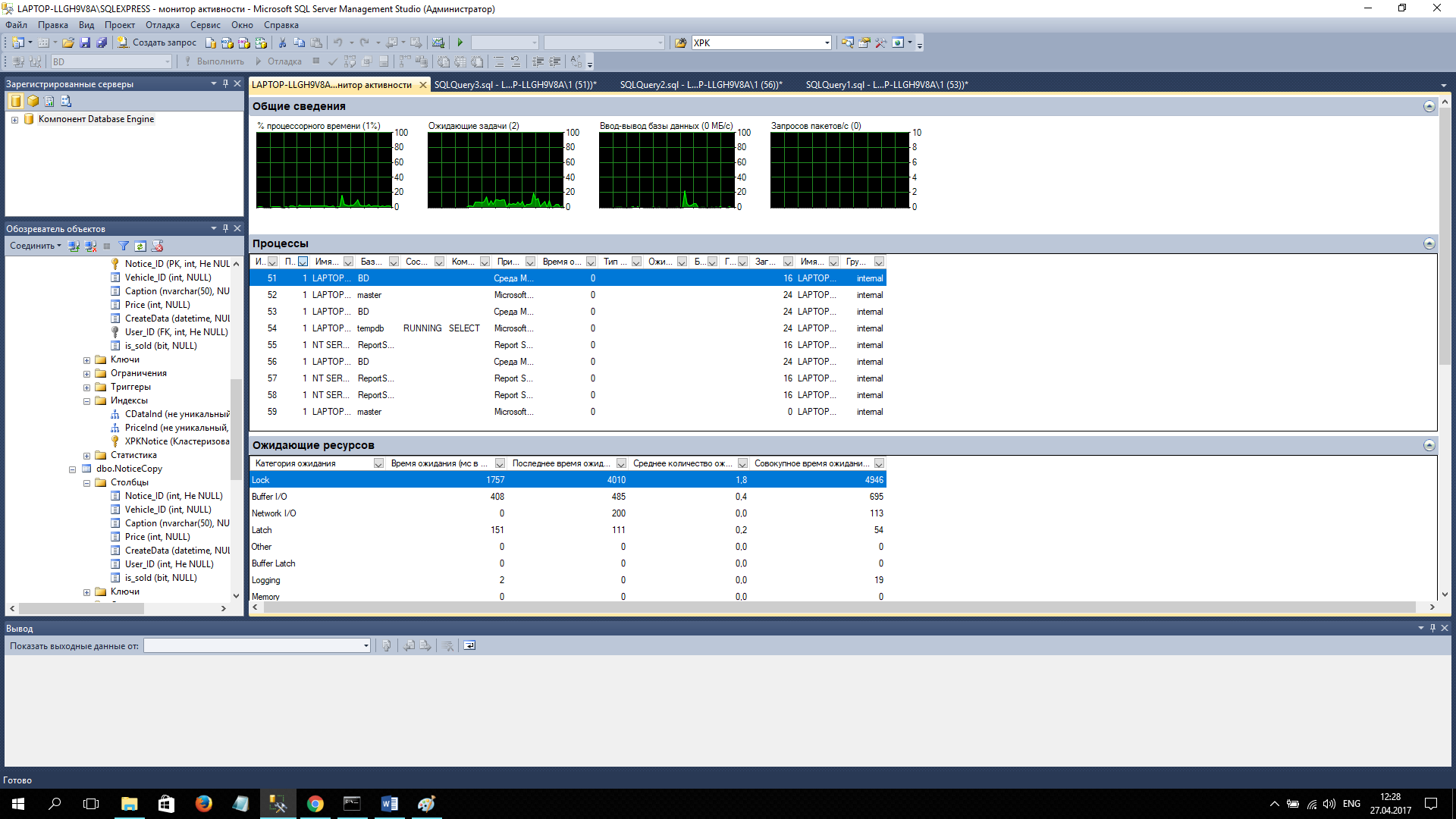


# Монитор производительности

Монитор производительности или как он называется в русской версии « Монитор активности» можно вызвать с помощью комбинации клавиш Ctrl+Alt+A.

Монитор имеет несколько разделов информации.

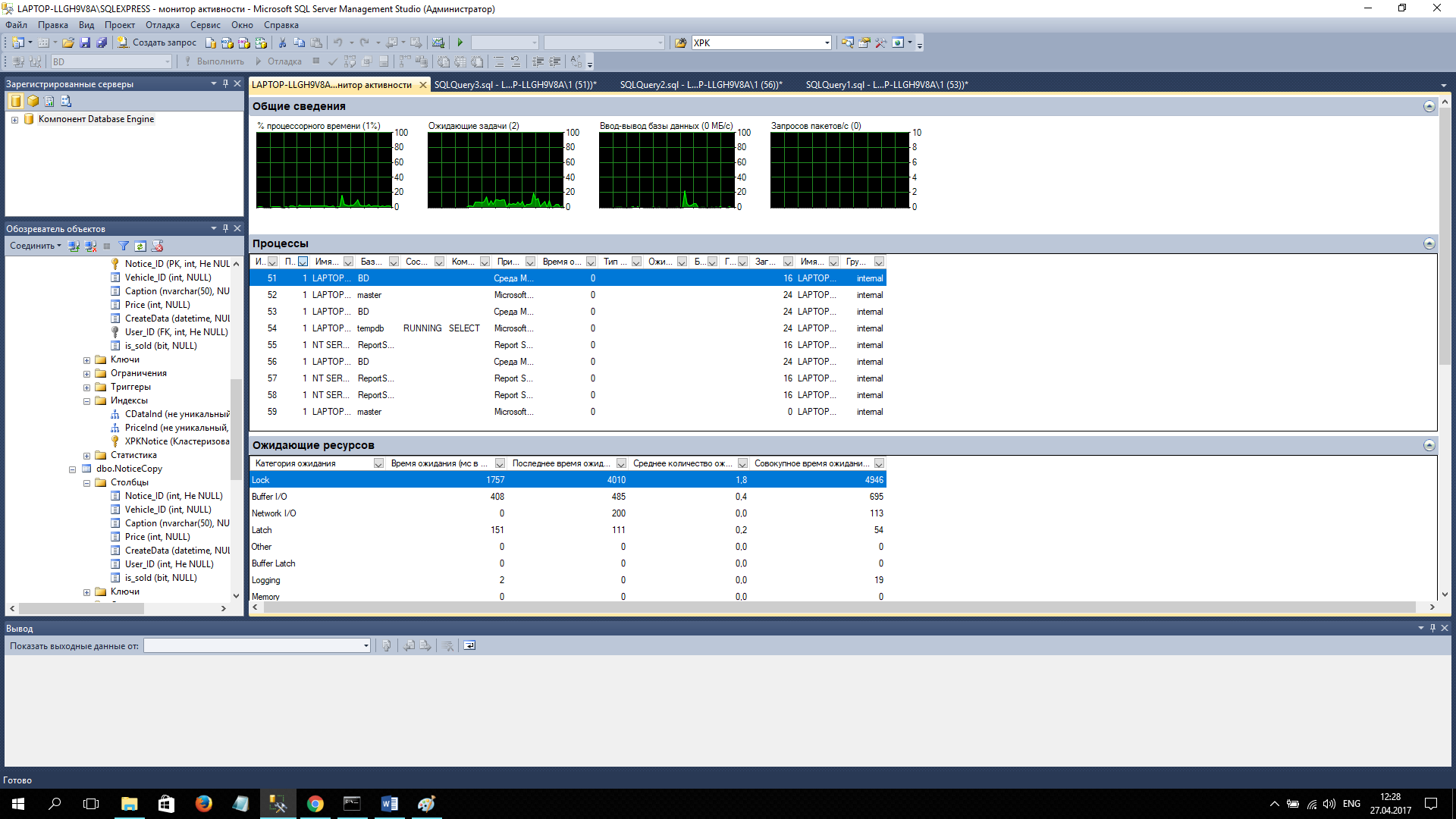
## Общие сведения.



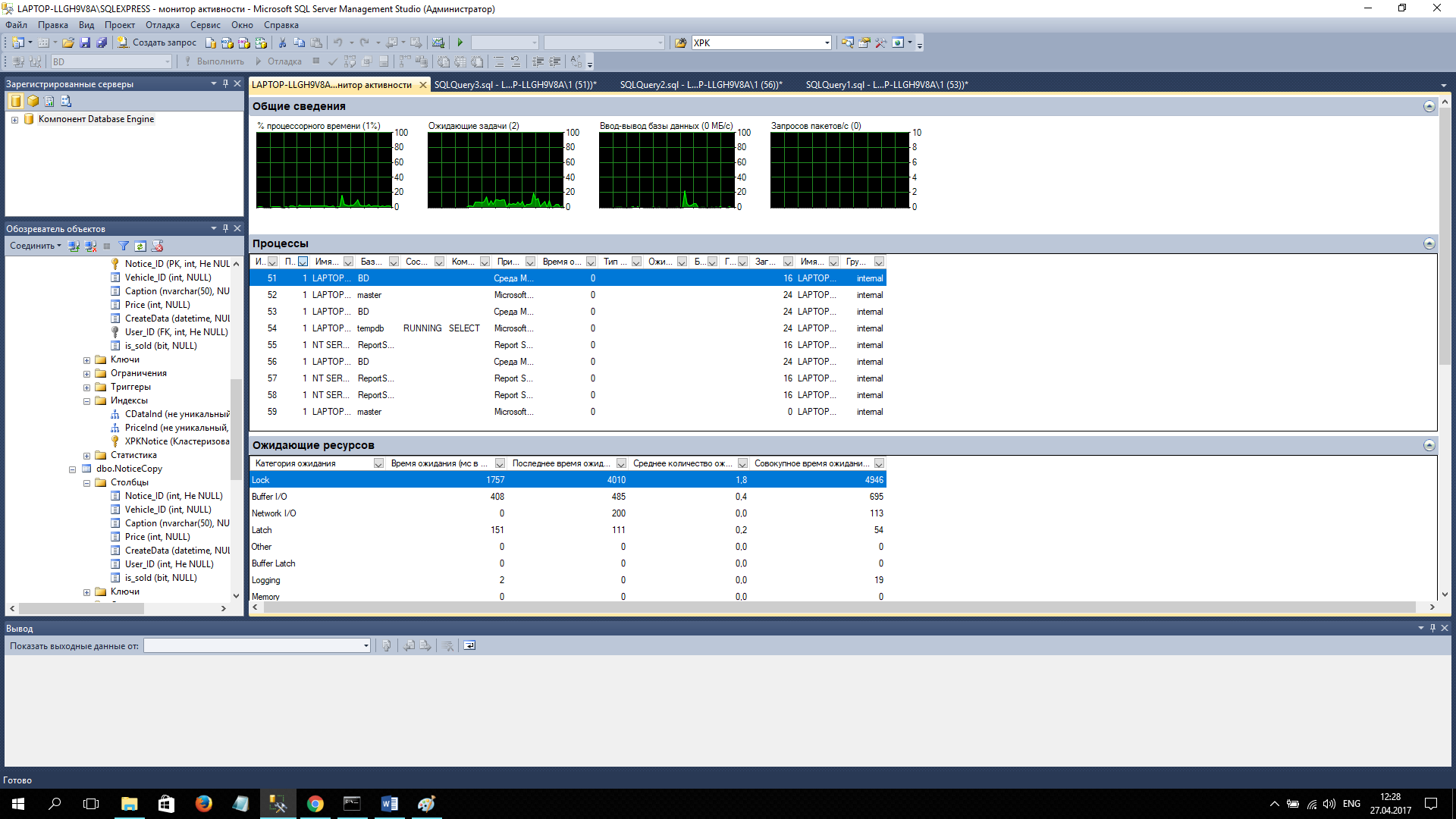
## Процессы

Таблица содержит следующую информацию:

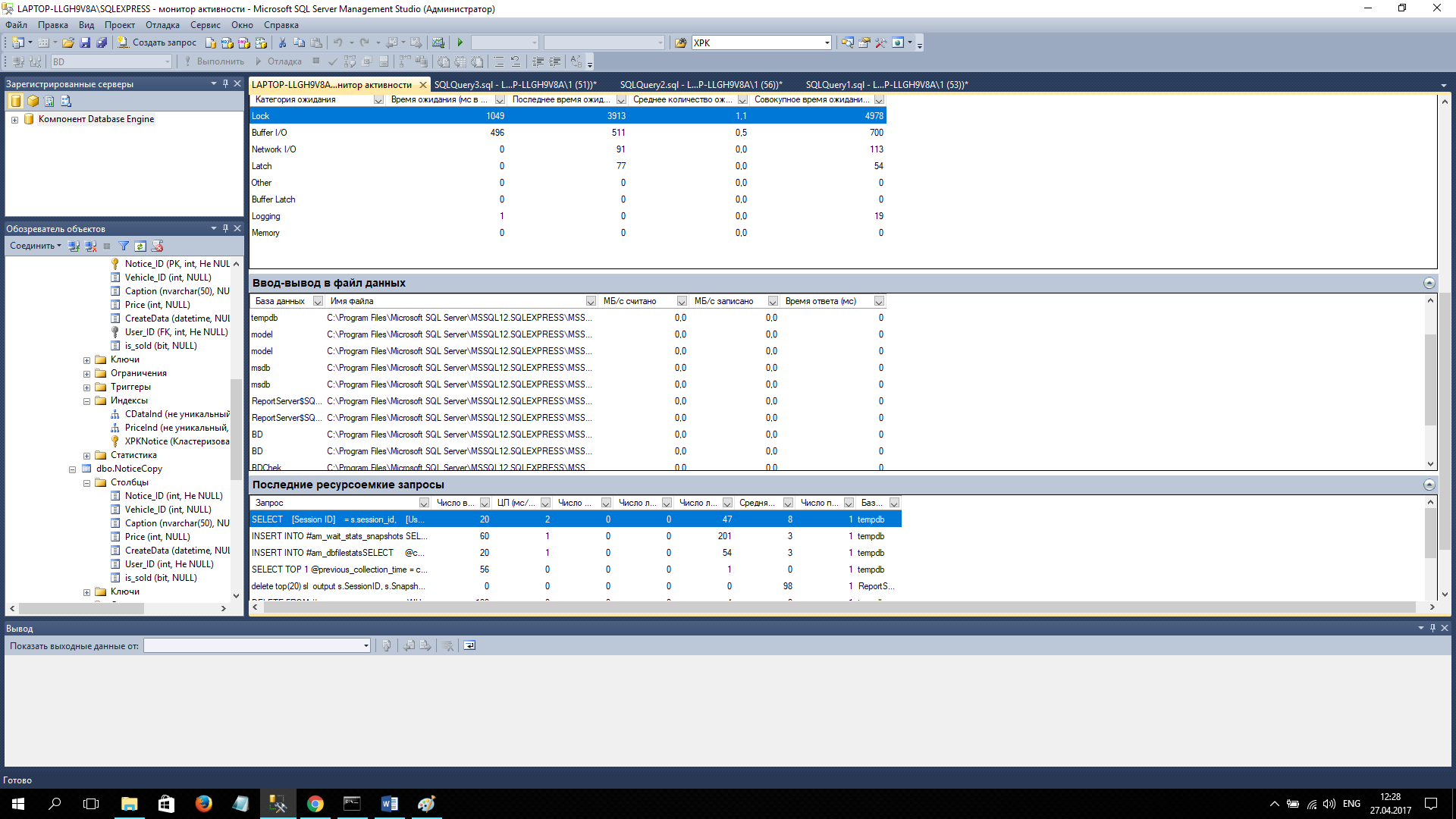
Идентификатор сеанса, Пользовательский процесс, Имя входа, База данных, Состояние задачи,Команда,Приложение,Время ожидание(мс),Тип ожидания,Ожидание ресурса,Блокирующий сеанс,Главная причина блокировки,Загрузка памяти(КБ),Имя узла,Группа рабочей нагрузки.



## Ожидающие ресурсов



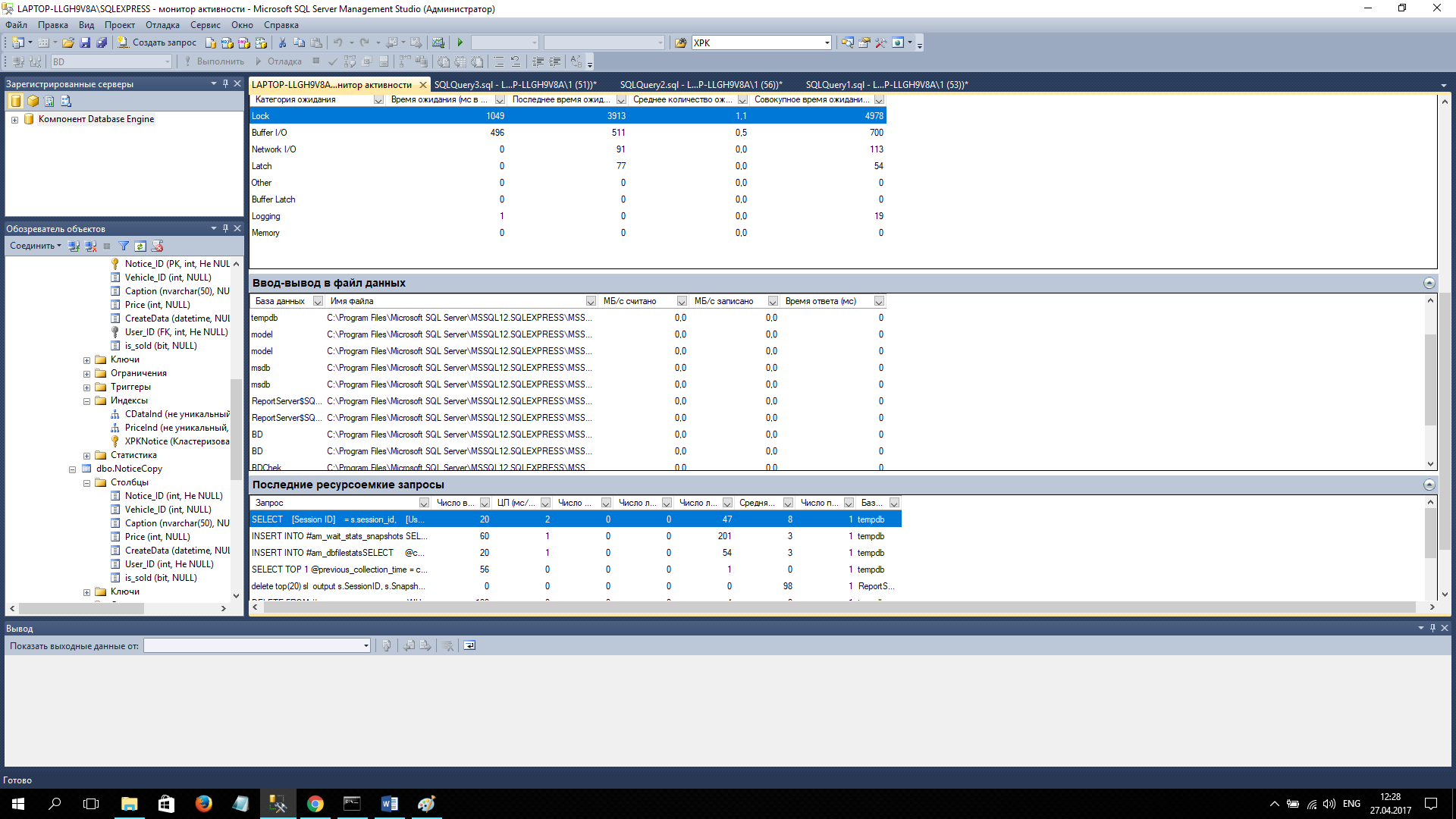
## Ввод-вывод в файл данных

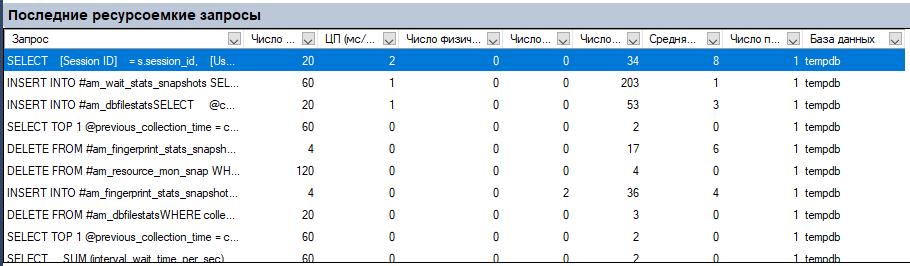


## Последние ресурсоемкие запросы

Данная таблица содержит следующую информацию:

Запрос, Число выполнений в минут, ЦП(м/с),Число физических операций чтения в секунду, Число логических операций записи в секунду, Число логических операций чтения в секунду,Средняя продолжительность(мсек),Число планов,База данных.



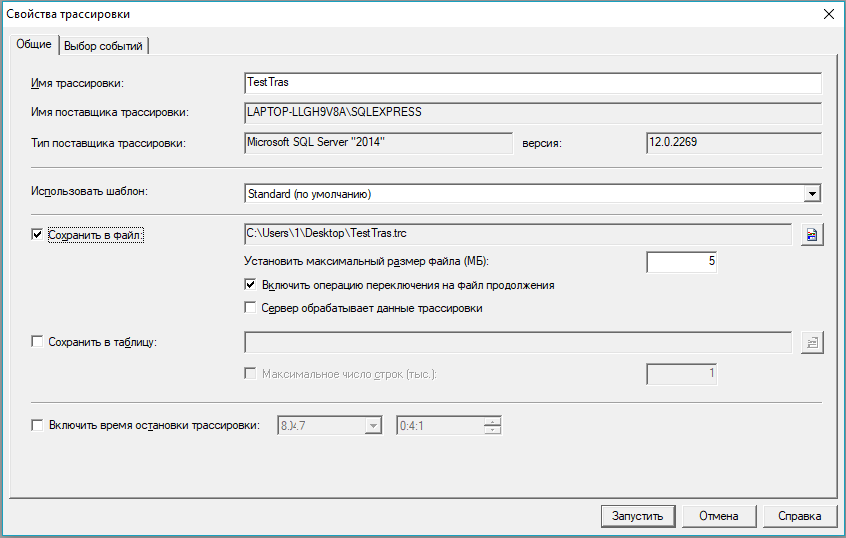


# SQL Server Profiler

Создание трассировки

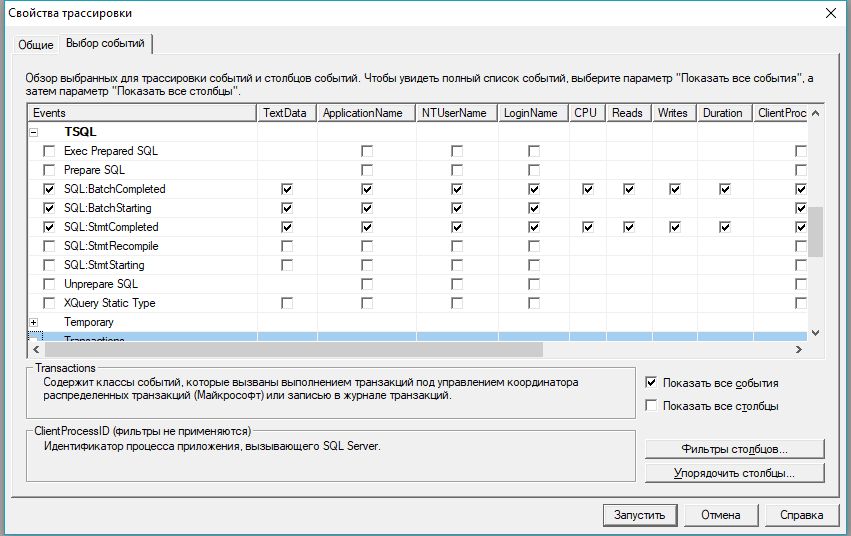
Для самой простой настройки достаточно просто ввести имя трассировки.

Трассировку можно сохранять в файл, что позволит открыть и просмотреть результаты трассировки после закрытия Profiler.

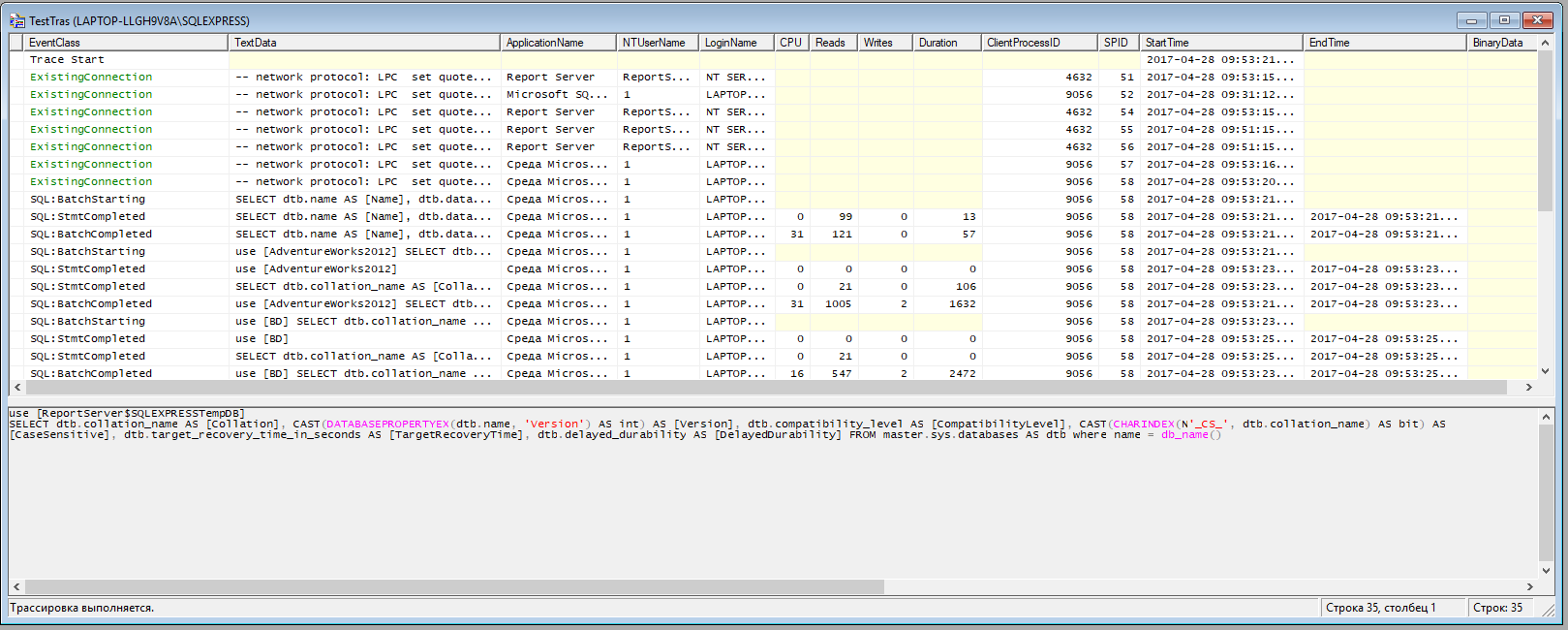


При настройке трассировки требуется выбрать те события, которые мы хотим отследить.

Данная настройка очень обширна как по количеству различных событий, так и по сведениям о каждом событии.



После запуска трассировки



Для отслеживания мы выполнили два запроса из первого пункта.

Строки отслеживания запроса с кластерным индексом.



Строки отслеживания запроса без индекса.



Так же отслеживать события можно с помощью расширенных событий – инструмента, который полностью заменит Profiler в 2016 версии в будущем.

# Настройка индексов

Для того чтобы просмотреть имеющиеся индексы в базе данных создан скрипт.

Индексы отсортированы по уровню фрагментации в порядке убывания.

SELECT tabl.name AS 'TableName' ,

indexs.name AS 'IndexName',

avg\_fragmentation\_in\_percent AS 'PercentOfFragmentaion'

FROM sys.dm\_db\_index\_physical\_stats(DB\_ID(N'BD'),

OBJECT\_ID(N'all'),NULL,NULL,NULL) AS a

JOIN sys.indexes AS indexs ON a.object\_id = indexs.object\_id AND a.index\_id = indexs.index\_id

JOIN sys.tables AS tabl ON tabl.object\_id = a.object\_id

ORDER BY avg\_fragmentation\_in\_percent DESC;

|  |  |  |
| --- | --- | --- |
| **TableName** | **IndexName** | **PercentOfFragmentaion** |
| Notice | PriceInd | 99,42503 |
| Comments | XPKComments | 50 |
| Notice | CDataInd | 1,344086 |
| Notice | XPKNotice | 0,499762 |
| NoticeCopy | NULL | 0,189753 |
| NoticeCopy1 | NULL | 0 |
| NNoticeCopy1 | NULL | 0 |
| NoticeCopy2 | NULL | 0 |
| Region | XPKRegion | 0 |
| Seller | XPKSeller | 0 |
| Statys | XPKStatys | 0 |
| User\_log | XPKUser\_log | 0 |
| Users\_profiles | XPKUsers\_profiles | 0 |
| sysdiagrams | PK\_\_sysdiagr\_\_C2B05B61CD9E3B08 | 0 |
| sysdiagrams | PK\_\_sysdiagr\_\_C2B05B61CD9E3B08 | 0 |
| sysdiagrams | UK\_principal\_name | 0 |
| Comments | CreateDataInd | 0 |
| Gearing | XPKGearing | 0 |
| Lable | XPKLable | 0 |
| Model | XPKModel | 0 |
| Motor | XPKMotor | 0 |
| History | XPKHistory | 0 |
| History | CreateDataInd | 0 |
| History | SoldDataInd | 0 |
| History | SellingPriceInd | 0 |
| kek | NULL | 0 |
| Vehicle | XPKVehicle | 0 |
| Vehicle | MileageInd | 0 |
| Vehicle | PowerInd | 0 |
| Vehicle | VolumeInd | 0 |
| Vehicle | CreateDataInd | 0 |
| Case | XPKCase | 0 |
| City | XPKCity | 0 |

Далее для индексов, степень фрагментации которых выше 30 % требуется выполнить перестроение. Для индексов со степенью фрагментации от 5% до 30% рекомендуется реорганизация.

ALTER INDEX PriceInd ON Notice REBUILD;

ALTER INDEX XPKComments ON Comments REBUILD;

Результаты

|  |  |  |
| --- | --- | --- |
| Notice | PriceInd | 0 |
| Comments | XPKComments | 40 |

Команда для реорганизации :

ALTER INDEX XPKComments ON Comments REORGANIZE;

# Системные динамические представления

Динамические административные представления и функции организованы в следующие категории:

|  |  |
| --- | --- |
| [Динамические административные представления, относящиеся к системе отслеживания измененных данных](https://technet.microsoft.com/ru-ru/library/bb522478(v=sql.105).aspx) | [Динамические административные представления и функции, связанные с вводом-выводом](https://technet.microsoft.com/ru-ru/library/ms190314(v=sql.105).aspx) |
| [Динамические административные представления, относящиеся к отслеживанию изменений](https://technet.microsoft.com/ru-ru/library/cc627419(v=sql.105).aspx) | [Динамические административные представления и соответствующие функции, связанные с объектами](https://technet.microsoft.com/ru-ru/library/bb630390(v=sql.105).aspx) |
| [Динамические административные представления, связанные со средой CLR](https://technet.microsoft.com/ru-ru/library/ms179982(v=sql.105).aspx) | [Динамические административные представления, связанные с уведомлениями запроса](https://technet.microsoft.com/ru-ru/library/ms187407(v=sql.105).aspx) |
| [Динамические административные представления, связанные с зеркальным отображением базы данных](https://technet.microsoft.com/ru-ru/library/ms173571(v=sql.105).aspx) | [Динамические административные представления, относящиеся к репликации](https://technet.microsoft.com/ru-ru/library/ms176053(v=sql.105).aspx) |
| [Динамические административные представления базы данных](https://technet.microsoft.com/ru-ru/library/ms181626(v=sql.105).aspx) | [Динамические административные представления для регулятора ресурсов](https://technet.microsoft.com/ru-ru/library/bb934218(v=sql.105).aspx) |
| [Динамические административные представления и соответствующие функции, связанные с выполнением](https://technet.microsoft.com/ru-ru/library/ms188068(v=sql.105).aspx) | [Динамические административные представления компонента Service Broker](https://technet.microsoft.com/ru-ru/library/ms176110(v=sql.105).aspx) |
| [Динамические административные представления расширенных событий](https://technet.microsoft.com/ru-ru/library/bb677293(v=sql.105).aspx) | [Динамические административные представления, относящиеся к операционной системе SQL Server](https://technet.microsoft.com/ru-ru/library/ms176083(v=sql.105).aspx) |
| [Динамические административные представления, связанные с полнотекстовым поиском](https://technet.microsoft.com/ru-ru/library/ms174971(v=sql.105).aspx) | [Динамические административные представления и функции, связанные с транзакциями](https://technet.microsoft.com/ru-ru/library/ms178621(v=sql.105).aspx) |
| [Динамические административные представления и функции, связанные с индексами](https://technet.microsoft.com/ru-ru/library/ms187974(v=sql.105).aspx) | [Динамические административные представления, связанные с безопасностью](https://technet.microsoft.com/ru-ru/library/bb677257(v=sql.105).aspx) |
| [Динамические административные представления, связанные с файловыми потоками (Transact-SQL)](https://technet.microsoft.com/ru-ru/library/ff773106(v=sql.105).aspx) |  |

Вот примеры результатов выборки по некоторым из них:

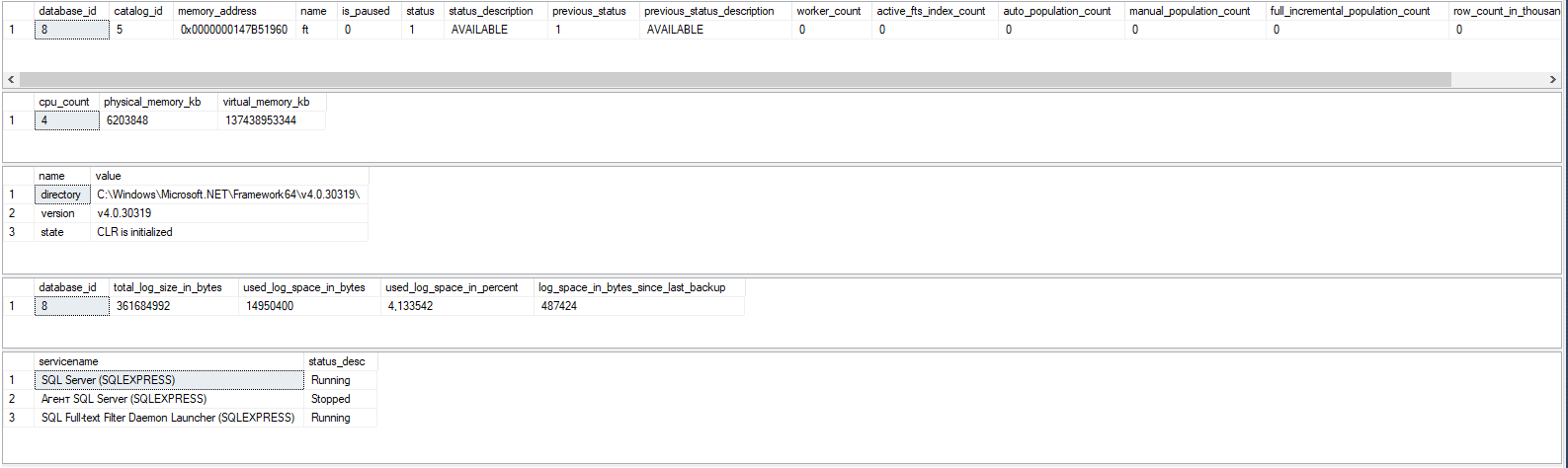
SELECT \* FROM sys.dm\_fts\_active\_catalogs

SELECT cpu\_count,physical\_memory\_kb,virtual\_memory\_kb FROM sys.dm\_os\_sys\_info

SELECT \* FROM sys.dm\_clr\_properties

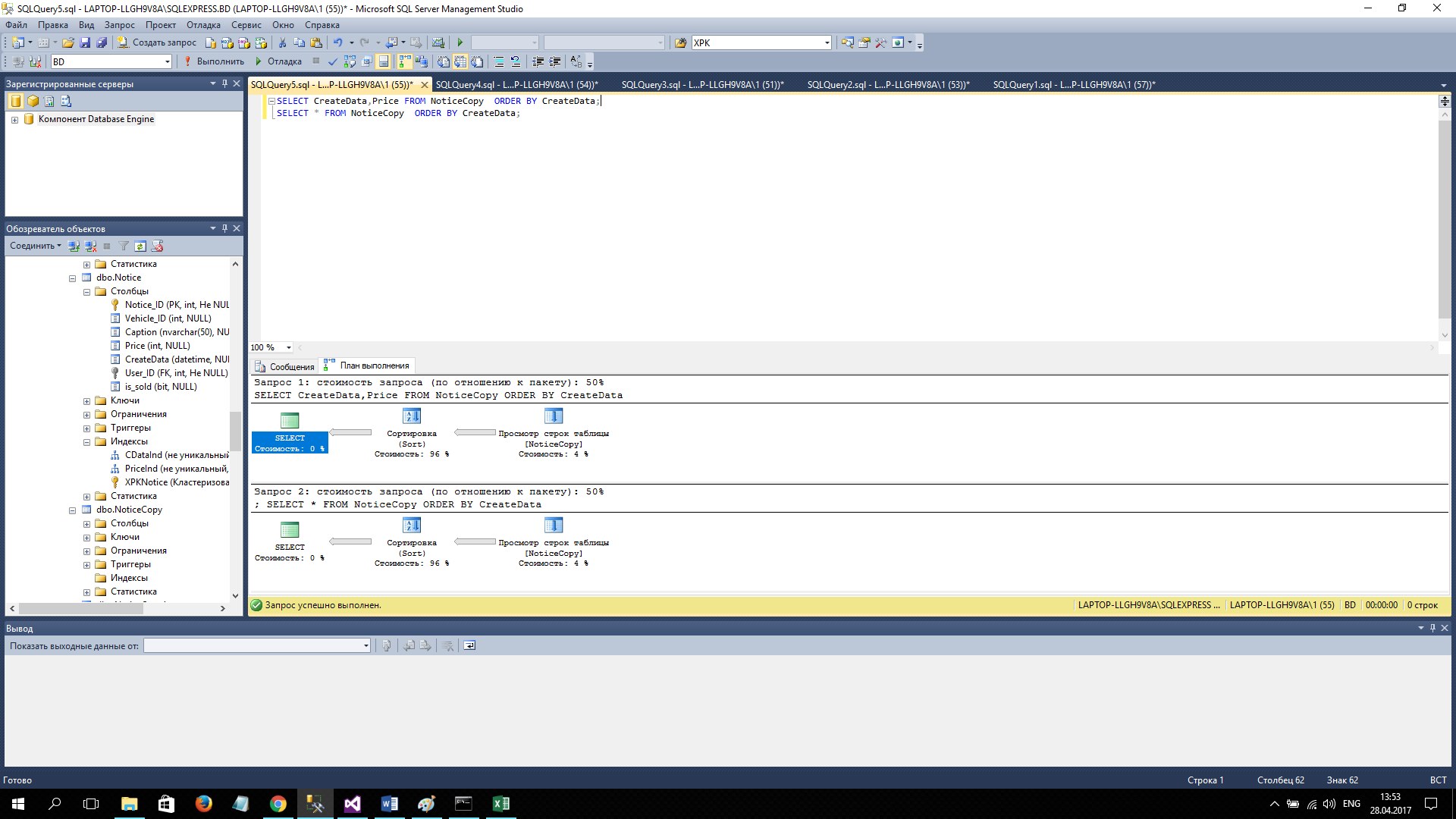
SELECT \* FROM sys.dm\_db\_log\_space\_usage

SELECT servicename, status\_desc FROM sys.dm\_server\_services

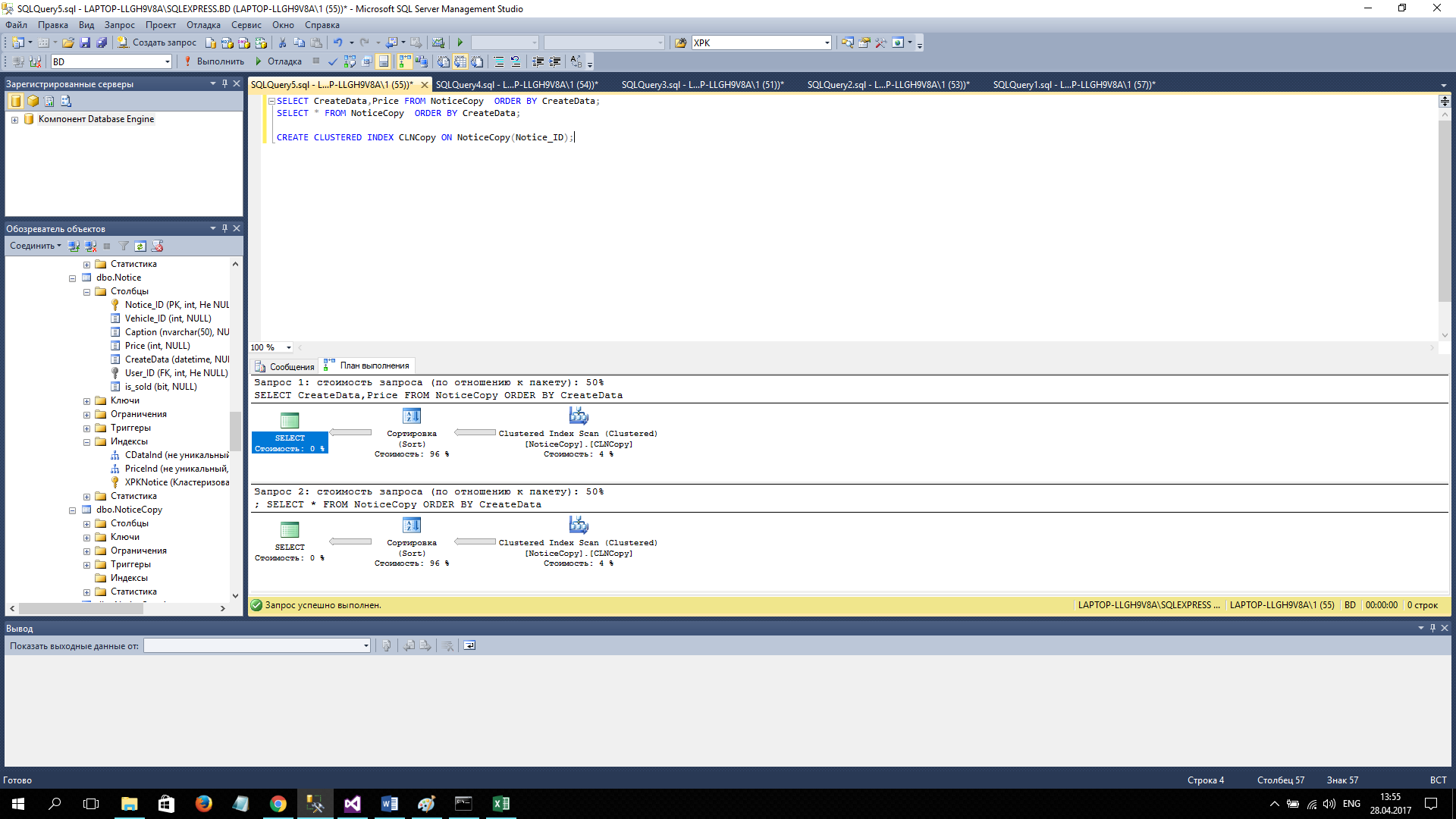


# Покрывающие индексы

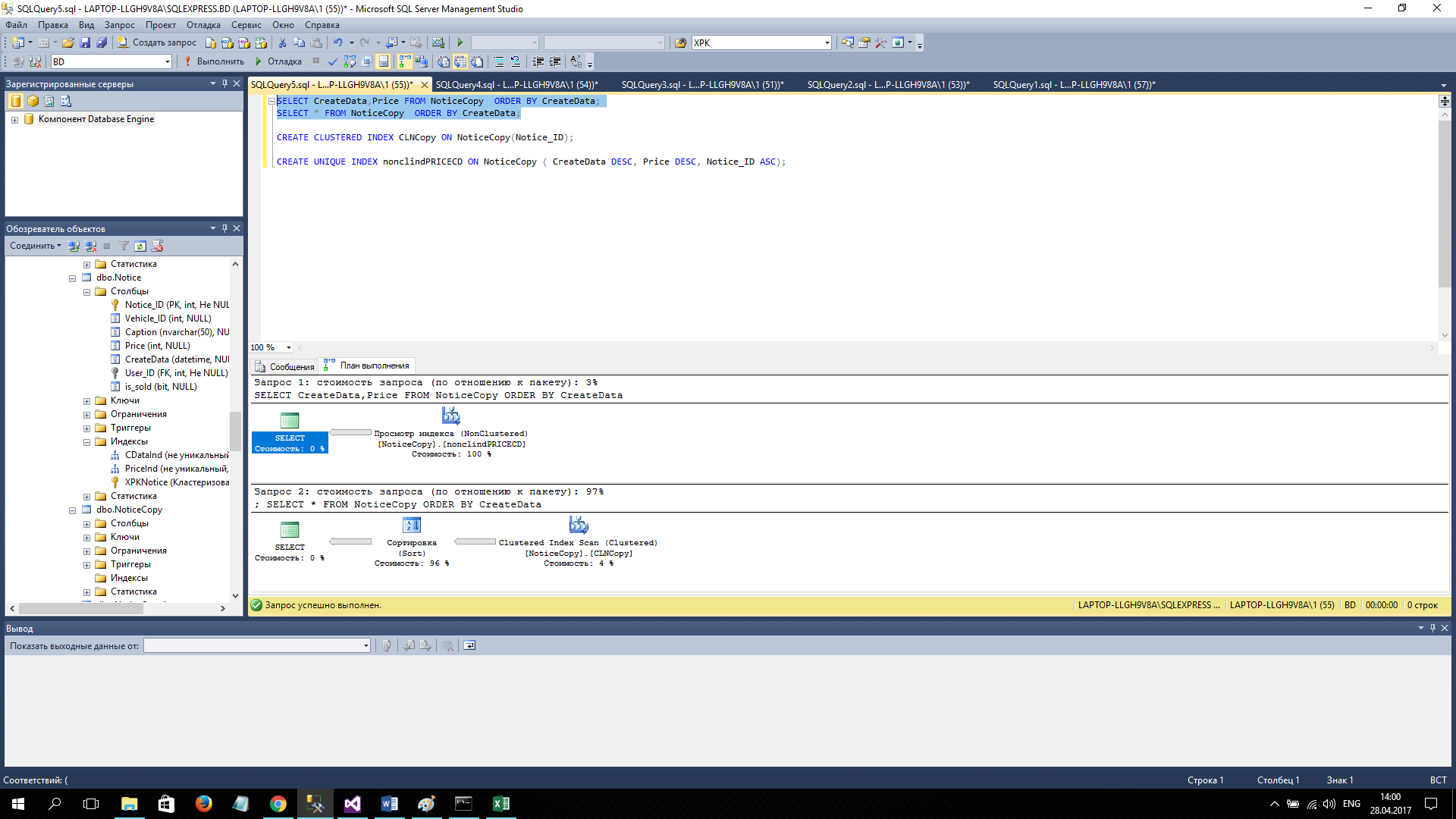
Проанализируем две выборки из таблицы без индексов.



Теперь создаем кластерный индекс и анализируем изменение плана выполнения запроса.



А теперь добавляем покрывающий индекс для верхнего запроса по двум полям.



Как видно из плана, стоимость запроса при обращении к кластерному индексу существенно больше, чем если задействуется только покрывающий индекс.

# XML индексирование

Подготавливаем данные

SELECT \* INTO NewCommentsCopy FROM Comments;

ALTER TABLE NewCommentsCopy

ALTER COLuMN Text XML;

Создаем первичный индекс

Так как таблица скопирована с помощью SELECT INTO в ней нет первичного ключа и предварительно перед созданием первичного индекса нужно создать первичный кластерный ключ.

ALTER TABLE NewCommentsCopy

ADD CONSTRAINT PK PRIMARY KEY CLUSTERED (Comments\_ID);

CREATE PRIMARY XML INDEX PrimarXMLInd ON NewCommentsCopy (Text);

Вторичный индекс – значение

CREATE XML INDEX valueIndex ON NewCommentsCopy (Text)

USING XML INDEX PrimarXMLInd FOR VALUE;

Индекс – путь

CREATE XML INDEX pathIndex ON NewCommentsCopy (Text)

USING XML INDEX PrimarXMLInd FOR PATH;

Индекс – свойство

CREATE XML INDEX PropertiesIndex ON NewCommentsCopy (Text)

USING XML INDEX PrimarXMLInd FOR PROPERTY;

Пример выполнения запроса

SELECT Comments\_ID AS 'ID', Text.value ('(/)[1]', 'varchar(30)') FROM NewCommentsCopy;

