بسمه تعالی

# رویدادنگاری، ممیزی و جرم‌شناسی در پایگاه داده‌ی MSSQL

# فهرست مطالب

# ۱ مقدمه

امروزه پایگاه‌های داده در برنامه‌های کاربردی مختلف کاربرد دارند. با توجه به ذخیره‌ی حجم زیادی از اطلاعات حساس در پایگاه‌های داده، لازم است که عملیات و تراکنش‌های رخ‌داده روی پایگاه‌های داده، موردبررسی قرار گیرد تا مشخص شود هرکسی، چه عملیاتی را در چه زمانی انجام داده است. همچنین در بحث جرم‌شناسی پایگاه‌های داده، جمع‌آوری شواهد و اطلاعات برای تجزیه‌وتحلیل از اهمیت زیادی برخوردار است. شواهد در پایگاه داده شامل رخدادهای ثبت‌شده و ممیزی‌ها است. ازاین‌رو، ثبت رویدادها و تهیه‌ی ممیزی در تمامی پایگاه‌های داده موردتوجه قرار گرفته است.

با توجه به اهمیت ثبت وقایع و ممیزی برای نظارت بر اتفاقات درون پایگاه‌های داده، در این گزارش پایگاه داده MSSQL و روش‌های متفاوت ثبت وقایع و ممیزی در آن و ابزارهای جرم‌شناسی موجود برای این پایگاه داده موردبررسی قرار گرفته است.

بر روی کارگزار عملیاتی SQL، فایل‌های رویدادنگاری[1] مختلفی ایجاد می‌شوند که در بررسی مشکلات پیش‌آمده و مسائل مربوط به کارایی و خطاها کمک خواهند کرد. در بخش ۳، انواع فایل‌های رویدادنگاری در پایگاه داده MSSQL موردبررسی قرار گرفته‌اند. ویژگی ممیزی در SQL Server امکان تهیه‌ی ممیزی از تمام اتفاقات رخ‌داده در کارگزار را فراهم می‌کند. در بخش ۴، نحوه‌ی انجام ممیزی در پایگاه داده MSSQL توضیح داده شده است. در بخش ۵، دو ابزار جرم‌شناسی معروف برای پایگاه داده‌ی MSSQL معرفی‌شده‌اند. بیشتر ابزارهای جرم‌شناسی SQL Server در حالت برون‌خط[2] اجرا می‌شوند؛ زیرا تحلیل در حالت زنده و آنلاین ممکن است باعث توقف فعالیت کارگزار و از دست رفتن داده‌ها شود.

# ۲ نحوه‌ی ثبت وقایع در پایگاه داده MSSQL

در این بخش انواع فایل‌های رویدادنگاری در پایگاه داده MSSQL موردبررسی قرار گرفته‌اند.

[1] Log files

[2] Offline

لازم به ذکر است که بررسی‌های انجام‌شده در این گزارش بر روی سیستم‌عامل ویندوز ۷ و سیستم مدیریت پایگاه داده‌ی MSSQL Server 2012 صورت گرفته است.

## ۲-۱ انواع فایل‌های رویدادنگاری

بر روی کارگزار SQL عملیاتی، فایل‌های رویدادنگاری مختلفی ایجاد می‌شوند که در بررسی مشکلات پیش‌آمده و مسائل مربوط به کارایی و خطاها کمک خواهند کرد. انواع فایل‌های رویدادنگاری موجود در این پایگاه داده، در جدول ۱ معرفی شده‌اند.

**جدول ۱ انواع فایل‌های رویدادنگاری در MSSQL**

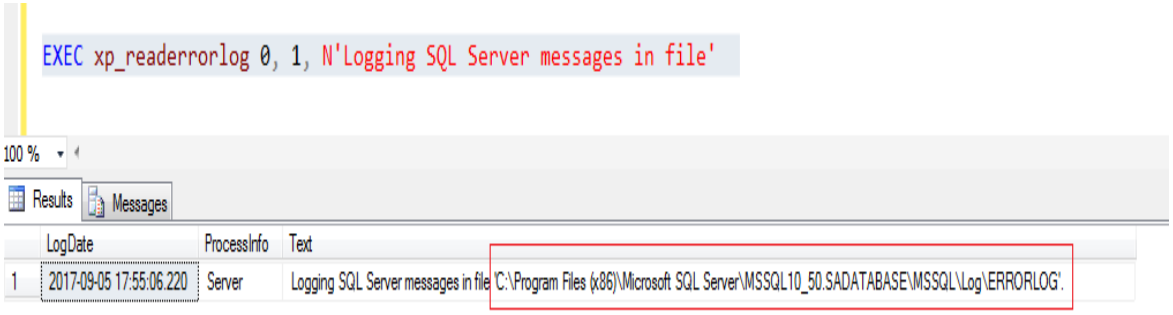
| نوع فایل رویدادنگاری | اطلاعات ثبت‌شده در فایل |
|---|---|
| فایل رویدادنگار خطا | رویدادهای خطا و پیغام‌های اطلاعاتی |
| ثبت رویدادها در ویندوز | اطلاعات بسیار دقیق درباره‌ی سلامت کارگزار SQL |
| ثبت رویدادهای مربوط به SQL Server Agent | پیام‌های هشدار و خطای مرتبط با کارهای اجراشده توسط SQL Server Agent |
| ثبت رویدادهای مربوط به SQL Server Profiler | ثبت فعالیت‌های جاری پایگاه داده‌ی کارگزار برای رفع خطاهای مربوط به برنامه‌های کاربردی پایگاه داده |
| فایل رویدادنگار تراکنش | تمامی تغییرات پایگاه داده |

در ادامه هر یک از فایل‌های فوق شرح داده شده‌اند.

### ۲-۱-۱ فایل رویدادنگار خطا

مهم‌ترین فایل رویدادنگاری استفاده‌شده توسط کارگزار SQL، فایل رویدادنگار خطا است. این فایل برای رفع مشکلات کلی سیستم به کار می‌رود. فایل رویدادنگار خطا، شامل رویدادهای خطا و پیغام‌های اطلاعاتی است. برای یافتن محل ذخیره‌سازی فایل‌های رویدادنگار خطا از دستور زیر استفاده می‌شود (شکل ۱ محل فایل‌های رویدادنگار خطا در SQL Server را نشان می‌دهد):

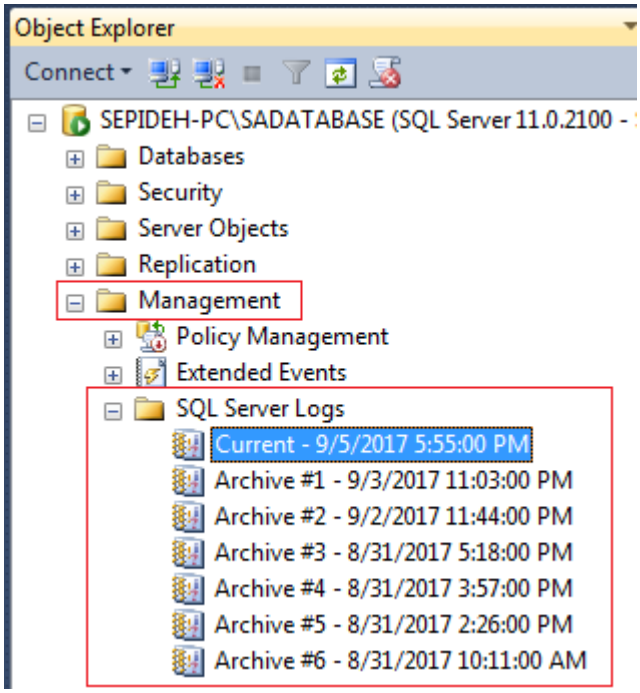
```
EXEC xp_readerrorlog 0, 1, N'Logging SQL Server messages in file'
```




**شکل ۱ محل فایل‌های رویدادنگار خطا در SQL Server**

نام فایل رویدادنگار خطای فعلی، ERRORLOG است. کارگزار SQL، شش فایل رویدادنگار اخیر را نگه می‌دارد و هر یک از فایل‌ها را به‌صورت ERRORLOG.n با پسوند عددی یک تا شش نام‌گذاری می‌کند [۱]. فایل رویدادنگار خطای جدید با هر بار راه‌اندازی SQL Server instance، ایجاد می‌شود. همچنین می‌توان از رویه‌ی سیستمی sp_cycle_errorlog به‌منظور ایجاد فایل رویدادنگار خطای جدید بدون نیاز به راه‌اندازی مجدد SQL Server instance استفاده کرد [۲].

به‌منظور مشاهده‌ی محتوای فایل رویدادنگار خطا، می‌توان آن را با برنامه‌ی ویرایشگر متن، باز کرد. همچنین می‌توان محتوای این فایل‌ها را از Microsoft SQL Server Management studio مشاهده کرد (شکل ۲).




**شکل ۲ مشاهده محتوای فایل‌های رویدادنگار خطا در SSMS**

راه‌حل دیگر، استفاده از رویه‌ی ذخیره‌شده‌ی سیستمی sp_readerrorlog یا رویه‌ی ذخیره‌شده‌ی توسعه‌یافته‌ی xp_readerrorlog است. رویه‌ی sp_readerrorlog، چهار پارامتر زیر را دریافت می‌کند [۳]:

۱. مقداری برای مشخص شدن فایل رویدادنگار خطا به‌منظور خواندن: ۰ = فایل جاری، ۱ = فایل آرشیوشده‌ی اول و ...

۲. نوع فایل رویدادنگاری: ۱ یا Null = فایل رویدادنگار خطا، ۲ = SQL Agent log

۳. رشته‌ی شماره‌ی یک برای جستجو

۴. رشته‌ی شماره‌ی دو برای جستجو

درصورتی‌که به رویه هیچ پارامتری داده نشود، محتوای فایل رویدادنگار خطای جاری را نشان می‌دهد (شکل ۳).

```
EXEC sp_readerrorlog 0, 1;
EXEC xp_readerrorlog 0, 1;
```

100 %

Results | Messages

| | LogDate | ProcessInfo | Text |
|---|---|---|---|
| 1 | 2017-09-05 17:55:06.030 | Server | Microsoft SQL Server 2012 - 11.0.2100.60 (Intel X86) F... |
| 2 | 2017-09-05 17:55:06.170 | Server | (c) Microsoft Corporation. |
| 3 | 2017-09-05 17:55:06.170 | Server | All rights reserved. |
| 4 | 2017-09-05 17:55:06.170 | Server | Server process ID is 1168. |
| 5 | 2017-09-05 17:55:06.170 | Server | System Manufacturer: 'Apple Inc.', System Model: 'MacBo... |
| 6 | 2017-09-05 17:55:06.190 | Server | Authentication mode is MIXED. |
| 7 | 2017-09-05 17:55:06.220 | Server | Logging SQL Server messages in file 'C:\Program Files (x8... |
| 8 | 2017-09-05 17:55:06.220 | Server | The service account is 'WORKGROUP\SEPIDEH-PC$'. ... |
| 9 | 2017-09-05 17:55:06.260 | Server | Registry startup parameters: -d C:\Program Files (x86)\... |
| 10 | 2017-09-05 17:55:06.260 | Server | Command Line Startup Parameters: -s "SADATABASE" |
| 11 | 2017-09-05 17:56:19.520 | Server | SQL Server detected 1 sockets with 4 cores per socket a... |
| 12 | 2017-09-05 17:56:19.520 | Server | SQL Server is starting at normal priority base (=7). This is a... |
| 13 | 2017-09-05 17:56:19.520 | Server | Detected 8102 MB of RAM. This is an informational mess... |
| 14 | 2017-09-05 17:56:19.600 | Server | Using conventional memory in the memory manager. |
| 15 | 2017-09-05 17:56:29.960 | Server | This instance of SQL Server last reported using a process ... |

**شکل ۳ مشاهده محتوای فایل رویدادنگار خطای جاری**

### ۲-۱-۲ ثبت رویدادها در ویندوز

ثبت رویدادها در ویندوز[3] به‌طور انحصاری توسط SQL server استفاده نمی‌شود؛ ولی یک منبع مهم اطلاعاتی برای رفع خطاهای کارگزار SQL به‌حساب می‌آید [۴]. ثبت رویدادها در ویندوز حاوی اطلاعات بسیار دقیقی درباره‌ی سلامت کارگزار SQL است. سه مورد اصلی برای ثبت رویداد در ویندوز شامل موارد زیر بوده که هر یک حاوی اطلاعاتی درباره‌ی زیرسیستم‌های مختلف کارگزار SQL هستند [۱]:

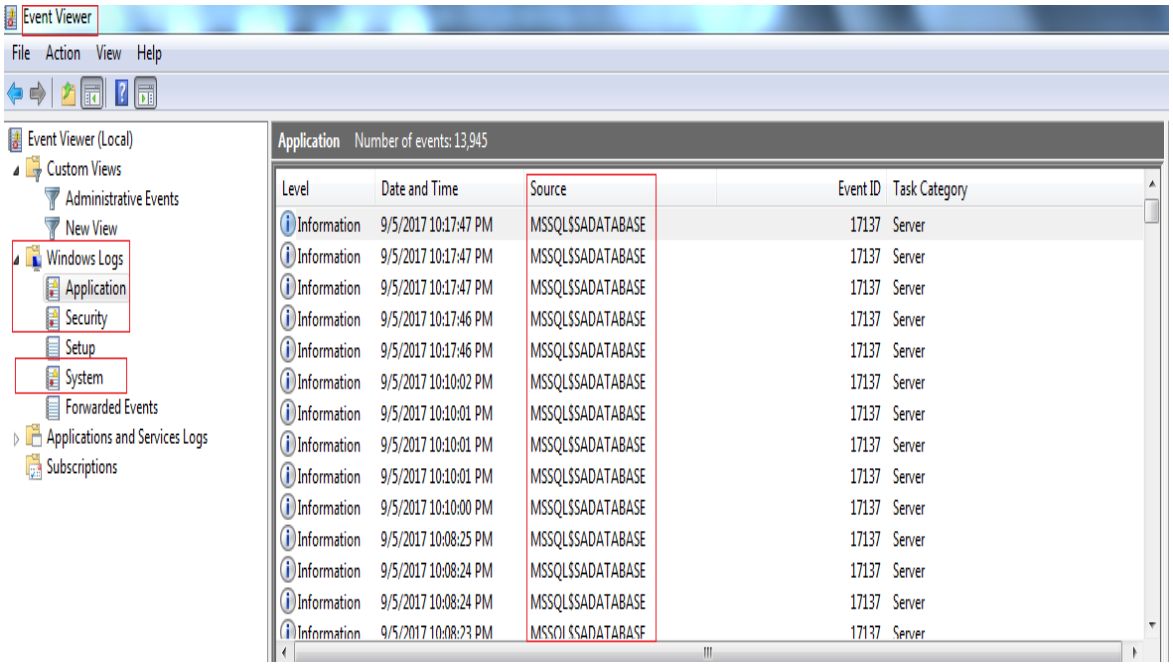
۱. برنامه‌ی کاربردی: رویدادهای درون کارگزار SQL و SQL Server agent را ثبت می‌کند.

۲. امنیت: اطلاعاتی در مورد تصدیق اصالت ثبت می‌شود.

۳. سیستم: رویدادهای راه‌اندازی و توقف سرویس را ثبت می‌کند.

[3] Windows event log

به‌منظور مشاهده‌ی این دسته از رویدادهای ثبت‌شده، از event viewer استفاده می‌شود (شکل ۴).




**شکل ۴ ثبت رویدادها در ویندوز**

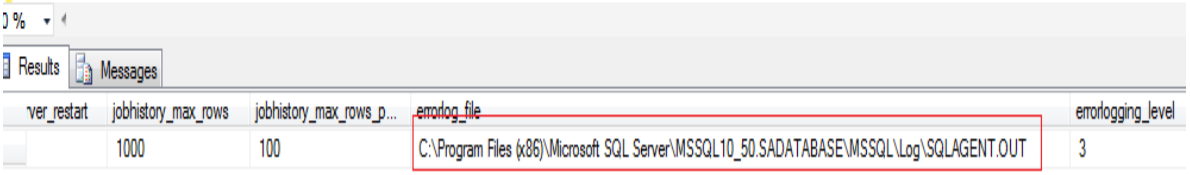
### ۳-۱-۲ ثبت رویدادهای مربوط به SQL Server Agent

SQL Server agent زیرسیستم برنامه‌ریز کار[4] است. این زیرسیستم، پیام‌های هشدار و خطای مرتبط با کارهایی که اجرا می‌کند را در فایل‌های رویدادنگاری خود می‌نویسد [۱]. برای یافتن محل فایل‌های رویدادنگاری مرتبط با SQL Server agent دستور زیر اجرا می‌شود (شکل ۵).

```
USE MASTER
GO
EXEC msdb..sp_get_sqlagent_properties
GO
```

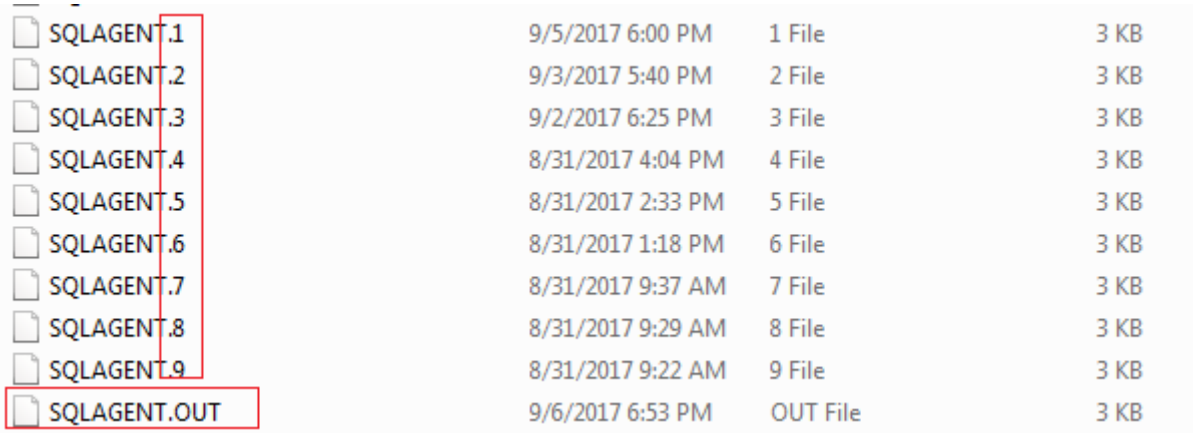
[4] Job scheduling subsystem

```
USE MASTER
GO
EXEC msdb..sp_get_sqlagent_properties
GO
```




**شکل ۵ محل فایل‌های رویدادنگاری مرتبط با SQL Server agent**

کارگزار SQL، ۹ فایل رویدادنگاری خطای مربوط به SQL Server agent را نگه می‌دارد. فایل رویدادنگاری جاری SQLAGENT.OUT نام دارد و فایل‌های آرشیوشده به ترتیب از یک تا نه (مطابق با شکل ۶) شماره‌گذاری شده‌اند [۱].




**شکل ۶ فایل‌های رویدادنگاری مرتبط با SQL Server agent**

همان‌طور که بیان شد، به‌منظور خواندن فایل رویدادنگاری جاری مربوط به SQL Server agent از دستور زیر (مطابق با شکل ۷) استفاده می‌شود [۳].

```
EXEC xp_readerrorlog 0, 2;
```

```
EXEC xp_readerrorlog 0, 2;
```

100 %

Results | Messages

| | LogDate | ErrorLevel | Text |
|---|---|---|---|
| 1 | 2017-09-06 18:49:22.000 | 3 | [100] Microsoft SQLServerAgent version 11.0.2100.60 (x86 unicode retail build) : Process ID 2348 |
| 2 | 2017-09-06 18:49:22.000 | 3 | [495] The SQL Server Agent startup service account is WORKGROUP\SEPIDEH-PC$. |
| 3 | 2017-09-06 18:49:36.000 | 1 | [298] SQLServer Error: 229, The EXECUTE permission was denied on the object 'sp_sqlagent_update_agent_xps', database 'msdb', schema 'dbo'. [SQLSTATE 42000]... |
| 4 | 2017-09-06 18:49:36.000 | 1 | [000] The EXECUTE permission was denied on the object 'sp_sqlagent_update_agent_xps', database 'msdb', schema 'dbo'. [SQLSTATE 42000] (Error 229) |
| 5 | 2017-09-06 18:49:36.000 | 1 | [298] SQLServer Error: 229, The EXECUTE permission was denied on the object 'sp_sqlagent_update_agent_xps', database 'msdb', schema 'dbo'. [SQLSTATE 42000]... |
| 6 | 2017-09-06 18:49:36.000 | 1 | [000] The EXECUTE permission was denied on the object 'sp_sqlagent_update_agent_xps', database 'msdb', schema 'dbo'. [SQLSTATE 42000] (Error 229) |
| 7 | 2017-09-06 18:53:33.000 | 3 | [098] SQLServerAgent terminated (normally) |

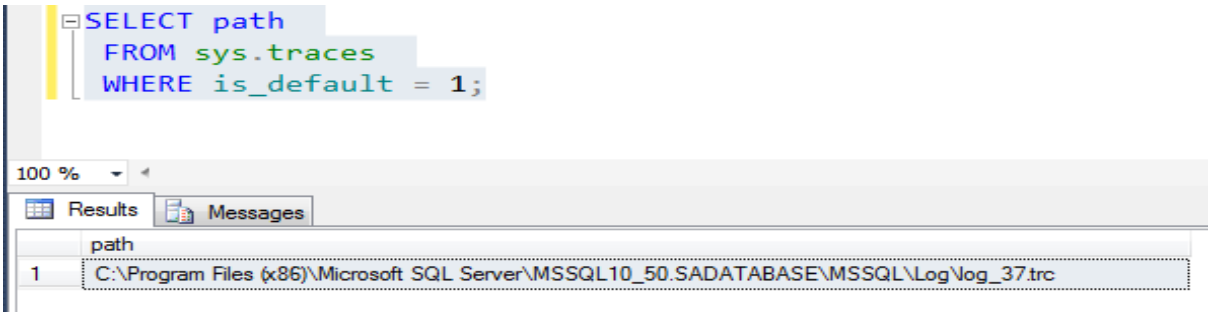
**شکل ۷ مشاهده‌ی محتوای فایل رویدادنگاری جاری مرتبط با SQL Server agent**

### ٤-١-٢ ثبت رویدادهای مربوط به SQL Server Profiler

SQL Server Profiler ابزار اصلی ردیابی برنامه‌ی کاربردی[5] است و برای رفع خطاهای مربوط به برنامه‌های کاربردی پایگاه داده مورداستفاده قرار می‌گیرد. SQL Server Profiler فعالیت‌های جاری پایگاه داده‌ی کارگزار را برای تحلیل‌های آتی در فایلی می‌نویسد. فایل‌های رویدادنگاری مربوط به SQL Server Profiler دارای ساختار log_<log_number>.trc هستند [۴]. برای یافتن مسیر پیش‌فرض ذخیره‌سازی این دسته از فایل‌های رویدادنگاری از دستور زیر استفاده می‌شود (شکل ۸):

```
SELECT path
 FROM sys.traces
 WHERE is_default = 1;
```

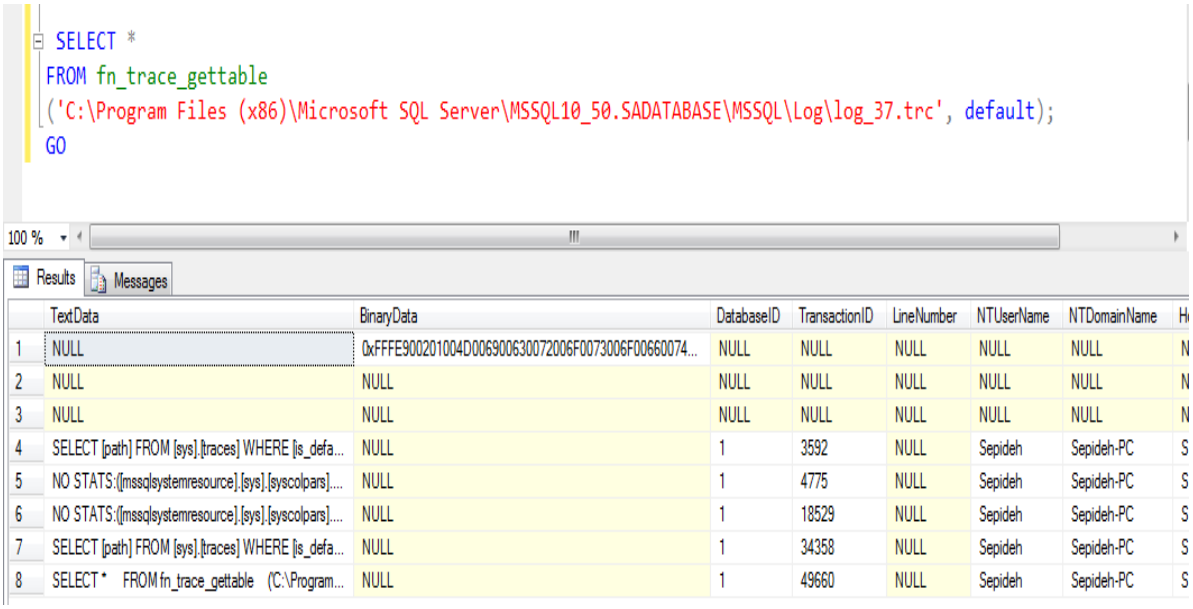
[5] Primary application tracing tool




**شکل ۸ محل فایل‌های رویدادنگاری مربوط به SQL Server Profiler**

برای مشاهده‌ی محتوای فایل رویدادنگاری جاری از دستور زیر استفاده می‌شود (شکل ۹):

```
SELECT *
FROM fn_trace_gettable
('C:\Program Files (x86)\Microsoft SQL
Server\MSSQL10_50.SADATABASE\MSSQL\Log\log_37.trc', default);
GO
```




**شکل ۹ مشاهده‌ی محتوای فایل رویدادنگاری جاری مربوط به SQL Server Profiler**

### ۵-۱-۲ فایل رویدادنگاری تراکنش

هر پایگاه داده در SQL Server instance، دارای یک فایل رویدادنگاری است که تمامی تغییرات پایگاه داده در آن ثبت می‌شوند. به دلیل آنکه این نوع از فایل‌های رویدادنگاری به‌صورت مستقل پیش از اعمال تغییرات

نوشته می‌شود، در مواقع شکست[۶] در سخت‌افزار یا خطای برنامه‌ی کاربردی، امکان بازگرداندن[۷] یا برگشت به عقب[۸] تراکنش‌ها را فراهم می‌کند. به دلیل اهمیت نقش فایل‌های رویدادنگاری تراکنش، رخدادها در یک یا چندین فایل رویدادنگاری مجزا از فایل‌های داده ذخیره می‌شوند. رکوردهای رویدادنگاری، پیش از آنکه محتوای تغییر داده شده در حافظه‌ی بافر[۹] بر روی فایل‌های داده نوشته شوند، در فایل‌های رویدادنگاری تراکنش نوشته می‌شوند [۵].

ثبت رویداد تراکنش برای هر پایگاه داده موارد زیر را پشتیبانی می‌کند [۵]:

۱. درصورتی‌که عبارت ROLLBACK اجرا شود یا موتور پایگاه داده خطایی را شناسایی کند، امکان بازگشت به عقب تراکنش وجود دارد.

۲. درصورتی‌که کارگزار دچار مشکل شود و تراکنشی ناتمام بماند، امکان بازگشت به عقب تراکنش وجود دارد. زمانی که کارگزار SQL مجدداً راه‌اندازی شود، تراکنش به عقب بازگردانده می‌شود.

۳. تراکنش‌های ناتمامی که به دلیل شکست[۱۰] در کارگزار در فایل‌های رویدادنگاری نوشته‌شده ولی در فایل‌های داده نوشته‌نشده‌اند، قابل بازیابی هستند. زمانی که کارگزار SQL مجدداً راه‌اندازی شود، تراکنش‌ها در فایل‌های داده نوشته می‌شوند.

۴. یک پایگاه داده، گروه فایل[۱۱]، فایل یا صفحه‌ی بازسازی‌شده را می‌توان به نقطه‌ای که شکست در سخت‌افزار رخ داده است، روبه‌جلو راند[۱۲]. پس‌ازآنکه آخرین فایل‌های پشتیبان کامل یا پشتیبان اختلافی[۱۳] اعمال شدند، می‌توان تراکنش‌ها را روبه‌جلو راند.

---

[6] Failure
[7] Restore
[8] Roll back
[9] Buffer cache
[10] Failure
[11] Filegroup
[12] Roll forward
[13] Differential backup

فایل یا فایل‌هایی که فایل‌های رویدادنگاری تراکنش را تشکیل می‌دهند به فایل‌های رویدادنگاری مجازی[14] تقسیم می‌شوند. سایز فایل‌های مجازی و تعداد آن‌ها توسط موتور پایگاه داده تعیین می‌شوند. تعیین حداقل و حداکثر سایز فایل رویدادنگاری فیزیکی بر عهده‌ی مدیر سیستم است. همچنین مدیر می‌تواند فایل‌های فیزیکی جدیدی به فایل‌های رویدادنگاری اضافه کند، از آن فایل‌هایی را حذف کند و سایز فایل‌ها را تغییر دهد [۵].

به‌منظور به دست آوردن اطلاعاتی در مورد فایل رویدادنگاری تراکنش مربوط به یک پایگاه داده از دستور زیر استفاده می‌شود (شکل ۱۰).

```
USE AdventureWorks2012;

SELECT name,
 size, -- in 8-KB pages
 max_size, -- in 8-KB pages
 growth,
 is_percent_growth
FROM sys.database_files
WHERE type_desc = 'LOG'
```

[14] Virtual log files

```
USE AdventureWorks2012;

SELECT name,
  size, -- in 8-KB pages
  max_size, -- in 8-KB pages
  growth,
  is_percent_growth
FROM sys.database_files
WHERE type_desc = 'LOG'
```

100 %

Results | Messages

| | name | size | max_size | growth | is_percent_growth |
|---|---|---|---|---|---|
| 1 | AdventureWorks2012_log | 112 | 268435456 | 10 | 1 |

**شکل ۱۰ اطلاعاتی در مورد فایل رویدادنگاری تراکنش مربوط به یک پایگاه داده**

در شکل ۱۰ به دلیل آنکه پایگاه داده تنها دارای یک فایل رویدادنگاری تراکنش است، خروجی یک سطر دارد.

برای به دست آوردن مسیر فایل رویدادنگاری تراکنش مربوط به یک پایگاه داده از دستور زیر استفاده می‌شود (شکل ):

```
SELECT * FROM sys.master_files WHERE name =
'AdventureWorks2012_log';
```

```
select * from sys.master_files where name = 'AdventureWorks2012_log';
```

.00 %

Results | Messages

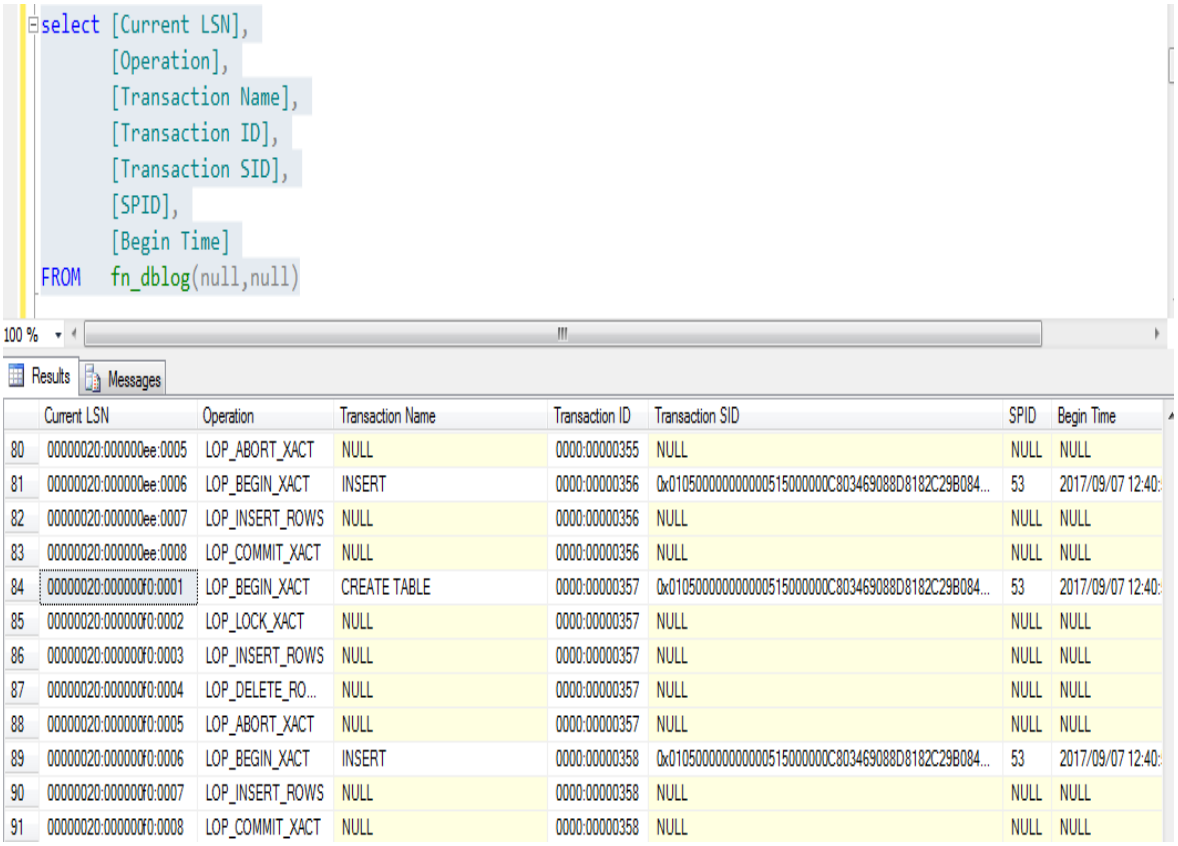
| | _desc | data_space_id | name | physical_name | state | state_desc |
|---|---|---|---|---|---|---|
| 1 | G | 0 | AdventureWorks2012_log | C:\Program Files (x86)\Microsoft SQL Server\MSSQL10_50.SADATABASE\MSSQL\DATA\AdventureWorks2012_log.ldf | 0 | ONLINE |

**شکل ۱۱ محل ذخیره‌سازی فایل رویدادنگاری تراکنش**

به‌منظور مشاهده محتوای فایل رویدادنگاری تراکنش از تابع fn_dblog() استفاده می‌شود (شکل ۱۱):

```
select [Current LSN],
```

```
 [Operation],
 [Transaction Name],
 [Transaction ID],
 [Transaction SID],
 [SPID],
 [Begin Time]
FROM fn_dblog(null,null)
```



```
select [Current LSN],
       [Operation],
       [Transaction Name],
       [Transaction ID],
       [Transaction SID],
       [SPID],
       [Begin Time]
FROM   fn_dblog(null,null)
```

100 %

Results | Messages

| | Current LSN | Operation | Transaction Name | Transaction ID | Transaction SID | SPID | Begin Time |
|---|---|---|---|---|---|---|---|
| 80 | 00000020:000000ee:0005 | LOP_ABORT_XACT | NULL | 0000:00000355 | NULL | NULL | NULL |
| 81 | 00000020:000000ee:0006 | LOP_BEGIN_XACT | INSERT | 0000:00000356 | 0x0105000000000005150000000C803469088D8182C29B084... | 53 | 2017/09/07 12:40: |
| 82 | 00000020:000000ee:0007 | LOP_INSERT_ROWS | NULL | 0000:00000356 | NULL | NULL | NULL |
| 83 | 00000020:000000ee:0008 | LOP_COMMIT_XACT | NULL | 0000:00000356 | NULL | NULL | NULL |
| 84 | 00000020:000000f0:0001 | LOP_BEGIN_XACT | CREATE TABLE | 0000:00000357 | 0x0105000000000005150000000C803469088D8182C29B084... | 53 | 2017/09/07 12:40: |
| 85 | 00000020:000000f0:0002 | LOP_LOCK_XACT | NULL | 0000:00000357 | NULL | NULL | NULL |
| 86 | 00000020:000000f0:0003 | LOP_INSERT_ROWS | NULL | 0000:00000357 | NULL | NULL | NULL |
| 87 | 00000020:000000f0:0004 | LOP_DELETE_RO... | NULL | 0000:00000357 | NULL | NULL | NULL |
| 88 | 00000020:000000f0:0005 | LOP_ABORT_XACT | NULL | 0000:00000357 | NULL | NULL | NULL |
| 89 | 00000020:000000f0:0006 | LOP_BEGIN_XACT | INSERT | 0000:00000358 | 0x0105000000000005150000000C803469088D8182C29B084... | 53 | 2017/09/07 12:40: |
| 90 | 00000020:000000f0:0007 | LOP_INSERT_ROWS | NULL | 0000:00000358 | NULL | NULL | NULL |
| 91 | 00000020:000000f0:0008 | LOP_COMMIT_XACT | NULL | 0000:00000358 | NULL | NULL | NULL |

**شکل ۱۱ مشاهده محتوای فایل رویدادنگاری تراکنش**

# ۳ نحوه‌ی انجام ممیزی در پایگاه داده MSSQL

ویژگی ممیزی[۱۵] در SQL Server امکان تهیه‌ی ممیزی از تمام اتفاقات رخ داده در کارگزار را فراهم می‌کند. رویدادها از تغییرات در تنظیمات کارگزار تا تغییر در مقداری در جدول خاصی از پایگاه داده را شامل

[15] Auditing

می‌شوند. رکوردهای ممیزی در محل ثبت رویدادهای امنیتی ویندوز[16]، محل ثبت رویدادهای کاربردی ویندوز[17] یا در فایل نوشته می‌شوند. در SQL Server 2012 امکان ممیزی در سطح کارگزار در تمامی نسخه‌ها وجود دارد ولی ممیزی پایگاه داده تنها در نسخه‌های Developer، Evaluation و Enterprise قابل‌اعمال است[۶،۷].

## ۳-۱ مؤلفه‌های ممیزی

قابلیت ممیزی در SQL Server شامل سه جزء اصلی است [۶]:

۱. ممیزی کارگزار[18]
۲. مشخصات ممیزی کارگزار[19]
۳. مشخصات ممیزی پایگاه داده[20]

### ۳-۱-۱ ممیزی کارگزار

ممیزی کارگزار، سرآغازی برای تعریف مشخصات ممیزی در SQL Server است. ممیزی کارگزار در پایگاه داده‌ی master قرار دارد و برای تعریف موارد زیر قابل‌استفاده است [۶]:

۱. نام ممیزی کارگزار
۲. محل ذخیره‌ی اطلاعات ممیزی

محل ذخیره‌سازی، شامل موارد زیر است:

- فایل
- ثبت رویدادهای امنیتی در ویندوز[21]

---

[16] Windows security log
[17] Windows application log
[18] Server Audit
[19] Server Audit specification
[20] Database audit specification

- ثبت رویدادهای مرتبط با برنامه‌های کاربردی در ویندوز[۲۲]

۳. خط‌مشی بازنویسی بر روی فایل[۲۳]

۴. تأخیر صف: حداکثر مقدار برحسب میلی‌ثانیه که سیستم ممکن است پیش از پردازش ممیزی صبر کند.

۵. عکس‌العمل SQL Server در مواردی که تهیه‌ی ممیزی امکان‌پذیر نیست.

گزینه‌های ممکن شامل موارد زیر هستند:

- ادامه و نادیده گرفتن ثبت رویداد
- خاموش کردن کارگزار
- شکست عملیات[۲۴]

۶. محدودیت سایز فایل و تعداد فایل‌های ممیزی

۷. حداکثر تعداد فایل‌های ممیزی بدون استفاده از خط‌مشی بازنویسی[۲۵]

۸. رزرو یا عدم رزرو فضای دیسک برای فایل‌های رویدادنگاری ممیزی

در ادامه، پرسمانی به عنوان مثال به‌منظور تعریف ممیزی کارگزار آورده شده است [۸]:

```
CREATE SERVER AUDIT [AdventureWorksAudit_DDL_Access] TO FILE
(FILEPATH = N'C:\Users\Sepideh\Desktop\mssqlAudit'
,MAXSIZE = 10 MB
)
WITH
(QUEUE_DELAY = 1000
,ON_FAILURE = CONTINUE
)
```

[21] Windows security log

[22] Windows application log

[23] File roll over policy

[24] Fail the operation

[25] Rollover

ممیزی در وضعیت غیرفعال ایجاد می‌شود و به‌صورت خودکار رویدادها را ثبت نمی‌کند؛ بنابراین پس از تعریف ممیزی باید آن را فعال کرد تا مقصد ممیزی، داده‌ها را دریافت کند.

```
ALTER SERVER AUDIT [AdventureWorksAudit_DDL_Access]WITH (STATE =
ON)
```

### ۲-۱-۳ مشخصات ممیزی کارگزار

از برخی از رویدادها همچون تهیه‌ی ممیزی از رویداد بازیابی (SELECT از یک جدول)، به‌صورت انفرادی ممیزی تهیه می‌شود. به این رویدادهای انفرادی عمل ممیزی[۲۶] می‌گویند. در بیش‌تر موارد عمل‌های ممیزی با هم گروه‌بندی می‌شوند و گروه اعمال ممیزی[۲۷] را شکل می‌دهند. بدین ترتیب با گروه‌بندی و ایجاد واحدهای منطقی، تنظیم مشخصات ممیزی آسان‌تر می‌شود [۶].

مشخصات ممیزی کارگزار که در تمامی نسخه‌های SQL Server پشتیبانی می‌شود، برای تعریف ممیزی در سطح کارگزار مورداستفاده قرار می‌گیرد.

سه مورد زیر در تعریف مشخصه ممیزی کارگزار مشخص می‌شوند [ ۶] .

۱. نام مشخصه‌ی ممیزی: این مورد اختیاری است و در صورت عدم تعیین، نام پیش‌فرضی اختصاص داده می‌شود.

۲. ممیزی کارگزار: مقصدی که باید رویدادهای انتخاب‌شده ثبت شوند را تعیین می‌کند.

۳. نوع عمل ممیزی: رویدادهایی هستند که باید از آن‌ها ممیزی تهیه شود.

در مشخصه‌ی کارگزار تمامی رویدادها در قالب گروه اعمال ممیزی گروه‌بندی شده‌اند. ازجمله این گروه‌ها می‌توان به موارد زیر اشاره کرد:

- SUCCESSFUL_LOGIN_GROUP
- FAILED_LOGIN_GROUP
- DBCC_GROUP

[26] Audit action

[27] Audit action group

با پرسمان زیر می‌توان مشخصه کارگزار را ایجاد کرد و آن را که به‌صورت پیش‌فرض غیرفعال است را در وضعیت فعال قرار داد [۸]:

```
CREATE SERVER AUDIT SPECIFICATION [SQL2012_FailedLogins_DbChanges]
FOR SERVER AUDIT [AdventureWorksAudit_DDL_Access]
ADD (FAILED_LOGIN_GROUP),
ADD (DATABASE_CHANGE_GROUP)
WITH (STATE = ON)
GO
```

به‌منظور ایجاد مشخصه‌ی ممیزی کارگزار، یک کاربر نیاز به مجوز ALTER ANY SERVER AUDIT دارد و مجوز CONTROL SERVER اجازه‌ی مشاهده‌ی ممیزی را به کاربر می‌دهد.

```
GRANT ALTER ANY SERVER AUDIT TO test11

GRANT CONTROL SERVER TO test11
```

### ۳-۱-۳ مشخصات ممیزی پایگاه داده

مشخصات ممیزی پایگاه داده، از رویدادها در سطح پایگاه داده ممیزی می‌گیرد. با استفاده از مشخصه‌ی ممیزی پایگاه داده، ممیزی می‌تواند در سطح شیئ یا کاربر انجام شود ولی در سطح ستون قابل انجام نیست.

به‌منظور ایجاد مشخصه‌ی ممیزی پایگاه داده، موارد زیر تنظیم می‌شوند [۸]:

۱. نام مشخصه‌ی ممیزی پایگاه داده. درصورتی‌که نامی مشخص نشود، مقدار پیش‌فرضی اختصاص داده می‌شود.

۲. ممیزی کارگزار که مشخصه‌ی ممیزی پایگاه داده به آن مرتبط می‌شود.

۳. نوع عمل ممیزی. در این فیلد می‌توان عمل ممیزی و گروه اعمال ممیزی را مشخص کرد.

۴. در صورت تعیین عمل ممیزی، نام شیئ برای تهیه‌ی ممیزی باید تعیین شود.

۵. شمای شیئ انتخاب‌شده

۶. نام principal یا استفاده از کلیدواژه public برای تهیه ممیزی از تمامی کاربران.

مثالی از ایجاد مشخصه‌ی ممیزی پایگاه داده در ادامه آورده شده است [۸]:

```
CREATE DATABASE AUDIT SPECIFICATION [AW_DDL_Access_dbSpec]
FOR SERVER AUDIT [AdventureWorksAudit_DDL_Access]
ADD (SCHEMA_OBJECT_ACCESS_GROUP),
ADD (SCHEMA_OBJECT_CHANGE_GROUP)
WITH (STATE = ON)
GO
```

به‌منظور تنظیم مشخصات ممیزی پایگاه داده، به کاربر باید مجوزهای ALTER ANY DATABASE AUDIT یا ALTER یا CONTROL روی پایگاه داده اعطا شوند.

```
GRANT ALTER ANY DATABASE AUDIT TO test11;

GRANT CONTROL TO test11;

GRANT ALTER TO test11;
```

## ۲-۳ خواندن داده‌های فایل ممیزی

اطلاعات ممیزی در فایل مقصد به‌صورت دودویی[۲۸] نوشته می‌شوند. به‌منظور خواندن فایل‌های دودویی ممیزی از تابع fn_get_audit_file() استفاده می‌شود. این تابع سه پارامتر نیاز دارد [۶]:

۱. الگوی فایل: مسیر یا مسیر فایل به همراه نام آن در این پارامتر مشخص می‌شوند. به‌منظور خواندن تمامی فایل‌های یک فولدر، پس از مسیر فولدر از * استفاده می‌شود.

۲. نام فایل اولیه: مسیر و نام فایلی که خواندن اطلاعات از آن شروع می‌شود.

۳. محل شروع در فایل اولیه در این قسمت مشخص می‌شود.

به‌عنوان مثال می‌توان از تابع fn_get_audit_file() به صورت زیر استفاده کرد:

```
SELECT * FROM
fn_get_audit_file('C:\Users\Sepideh\Desktop\mssqlAudit\*',
default, default);
```

در شکل با توجه به تنظیمات انجام‌شده در بخش‌های پیشین، ممیزی مربوط به عدم موفقیت در ورود به کارگزار نشان داده شده است.

[28] Binary

| | istance_name | database_name | schema_name | statement | additional_information |
|---|---|---|---|---|---|
| 1 | H-PC\SADATABASE | | | | <action_info xmlns="http://sch |
| 2 | H-PC\SADATABASE | | | Login failed for user 'dr_B'. Reason: Password did not match that for the login provided. [CLIENT: <local machine>] | <action_info xmlns="http://sch |
| 3 | H-PC\SADATABASE | | | Login failed for user 'dr_B'. Reason: Password did not match that for the login provided. [CLIENT: <local machine>] | <action_info xmlns="http://sch |
| 4 | H-PC\SADATABASE | | | | <action_info xmlns="http://sch |
| 5 | H-PC\SADATABASE | | | Network error code 0x6d occurred while establishing a connection; the connection has been closed. This may have ... | <action_info xmlns="http://sch |

**شکل۱۳ اطلاعات ممیزی**

اطلاعات ممیزی نوشته‌شده در windows security log یا application log با استفاده از **event viewer**[۲۹] قابل خواندن هستند.

## ۳-۳ بهترین روش‌ها

در ادامه برخی روش‌های پیشنهادی برای ممیزی بیان شده‌اند [۶]:

۱. نوشتن ممیزی در محل متمرکز

۲. تسهیل پردازش داده‌های ممیزی شده، با بارگذاری رویدادهای ثبت‌شده در پایگاه داده

۳. استفاده از یک فایل مقصد برای کارایی بهینه

٤. استفاده از ممیزی هدفمند به‌منظور به حداقل رساندن داده‌های جمع‌آوری‌شده و کارایی بهتر

۵. اطمینان از عدم مغایرت خط‌مشی بازنویسی بر روی فایل‌های رویدادنگاری در ویندوز با استراتژی ممیزی، در صورت نوشتن اطلاعات در Windows logs

# ٤ ابزارهای جرم‌شناسی

روش‌های جرم‌شناسی موجود در MSSQL Server، می‌توانند برای بررسی دسترسی‌های غیرمجاز به داده‌ها، کلاه‌برداری و جمع‌آوری اطلاعات برای تشخیص نفوذ استفاده شوند. در حقیقت جرم‌شناسی در SQL Server در موارد زیر کاربرد دارد [۹]:

[۲۹] ابزاری از ابزارهای مدیریتی ویندوز که در مسیر control panel -> administrative tools قرار دارد.

۱. تزریق SQL روشی است برای دسترسی به داده‌های حساس به‌صورت غیرمجاز که توسط مهاجمین استفاده می‌شود. در این مواقع جرم‌شناسی SQL Server، روشی است برای تشخیص تغییرات ایجادشده در پایگاه داده.

۲. تشخیص داده‌های تغییر داده شده

۳. تشخیص منشأ اصلی تغییرات و حملات

بیشتر ابزارهای جرم‌شناسی SQL Server در حالت برون‌خط[۳۰] اجرا می‌شوند؛ زیرا تحلیل در حالت زنده و برخط ممکن است باعث توقف فعالیت کارگزار و از دست رفتن داده‌ها شود [۱۰].

## ۴-۱ SysTools SQL Recovery

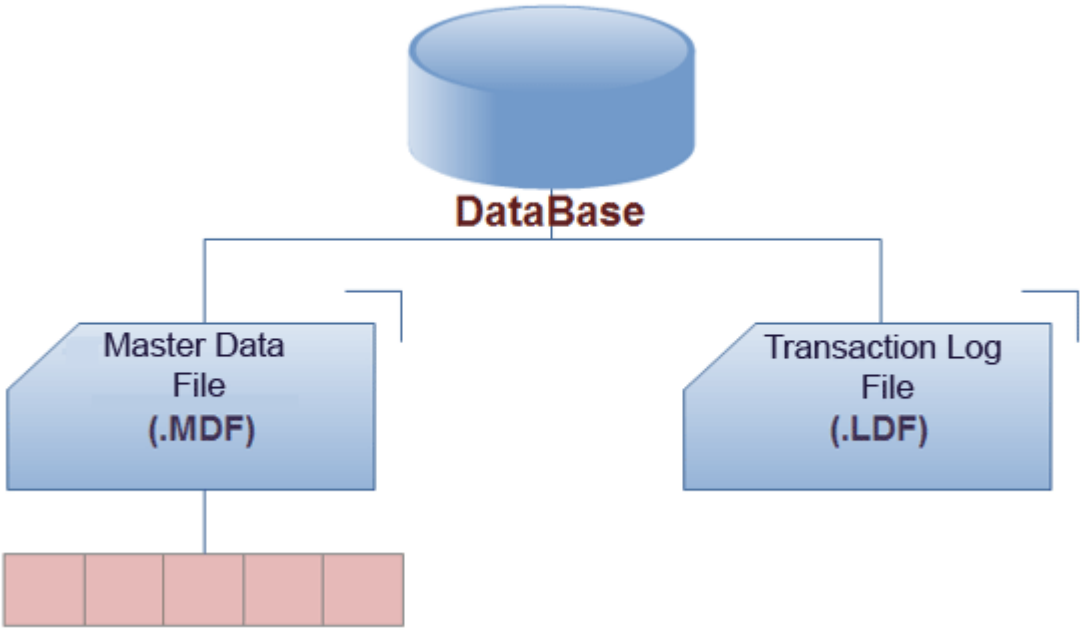
SQL Server پایگاه‌های داده خود را در سه دسته فایل ذخیره می‌کند. این فایل‌ها برای تحلیل و بررسی جرم‌شناسی بسیار مفید و سودمند هستند [۹]. سه دسته فایل شامل موارد زیر هستند (شکل ۱۲):

۱. **فایل mdf.**: این دسته از فایل‌ها، داده‌های اصلی شامل جداول، توابع، قوانین و رویه‌های ذخیره‌شده را در خود نگه می‌دارند.

۲. **فایل ndf.**: دسته‌ی دیگری از فایل‌های پایگاه داده هستند که توسط کاربر تعریف شده‌اند.

۳. **فایل ldf.**: SQL Server از این دسته از فایل‌ها برای ذخیره‌سازی اطلاعات رخدادها استفاده می‌کند. تغییرات در پایگاه داده شامل درج، حذف و به‌روزرسانی در فایل‌های رویدادنگاری تراکنش ذخیره می‌شوند.

[30] Offline

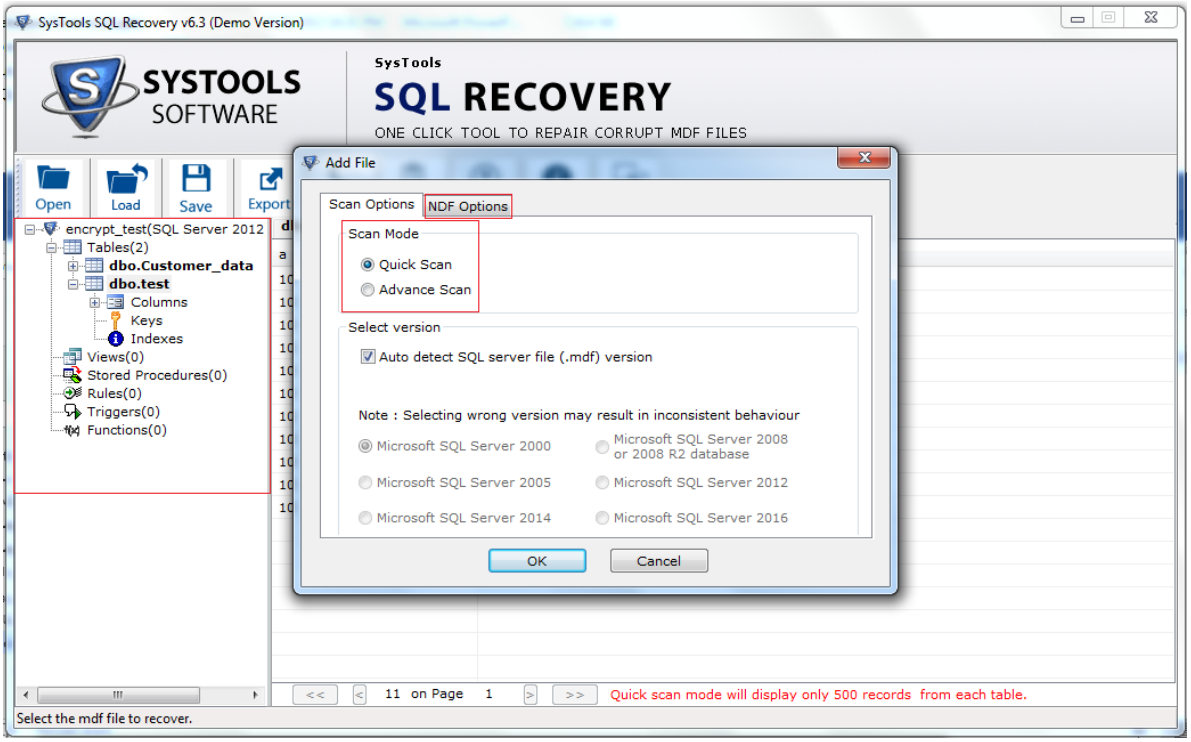



**شکل ۱۲ فایل‌های ذخیره‌سازی پایگاه داده**

تحلیل فایل‌های مذکور در موارد مختلفی ازجمله موارد زیر کاربرد دارد [۹]:

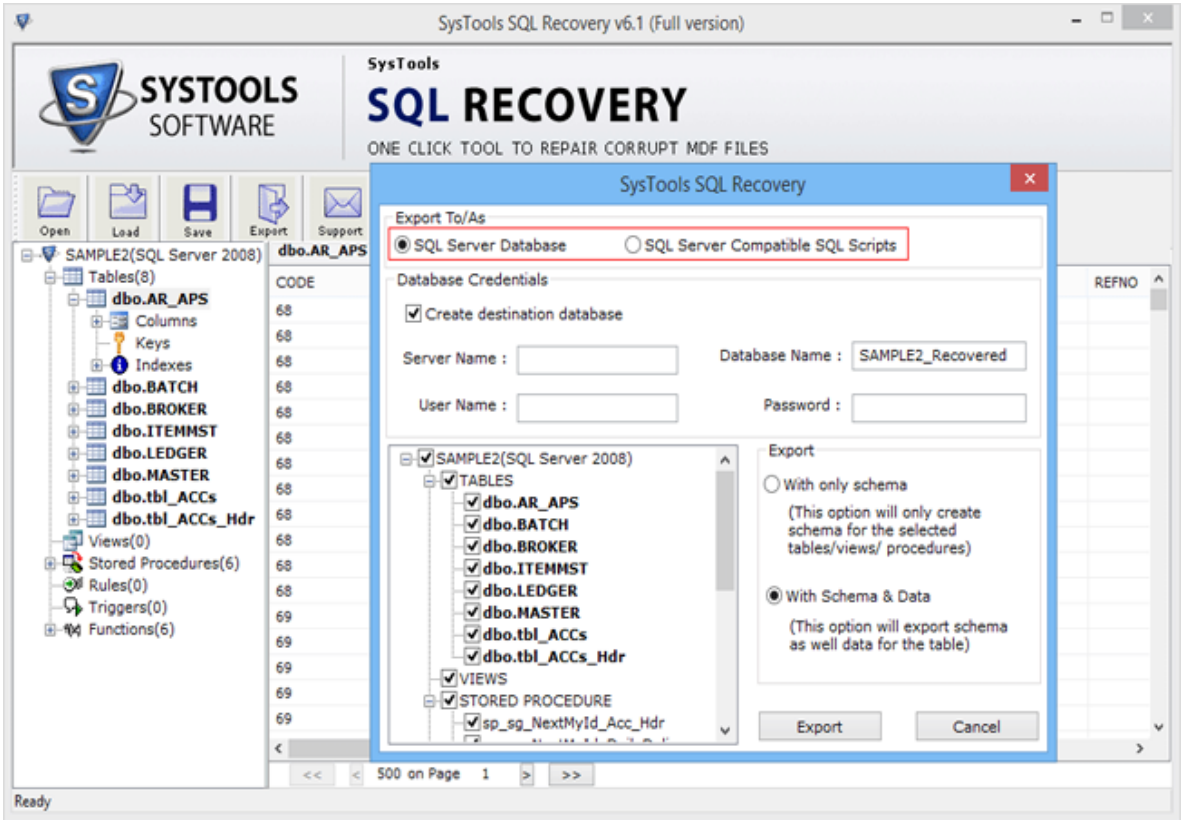
۱. شناسایی نقض امنیت داده
۲. شناسایی سطوح نفوذ به پایگاه داده
۳. بررسی اطلاعات ذخیره‌شده

ابزار SysTools SQL Recovery برای تحلیل فایل‌های پایگاه داده MDF مورد استفاده قرار می‌گیرد. این ابزار می‌تواند تمامی تریگرها، قوانین، توابع و جداول ذخیره‌شده را بازیابی کند. این ابزار قابلیت تحلیل فایل‌های NDF را نیز دارد. با استفاده از الگوریتم‌های Quick scanning و advance scanning در این ابزار، می‌توان فایل‌های MDF و LDF را ترمیم و بازیابی کرد (شکل ۱۳). همچنین قابلیت بازیابی داده‌های حذف‌شده‌ی جداول که نتیجه‌ی فعالیت خرابکارانه‌ی مهاجمین است را دارد [۱۰]. لازم به ذکر است که این ابزار، ابزار متن‌بازی نیست و دارای نسخه‌ی آزمایشی با ویژگی‌های محدودی است.




**شکل ۱۳ ابزار SysTools SQL Recovery**

در هنگام تجزیه‌وتحلیل قانونی SQL Server، چالش بزرگی که محققان با آن روبه‌رو هستند، استخراج شواهد است. در این ابزار، محققان می‌توانند فایل SQL را در پایگاه داده SQL Server یا در اسکریپتی سازگار با SQL Server، استخراج کنند(شکل ۱۴) [۱۱].




**شکل ۱۴ استخراج شواهد در SysTools SQL Recovery**

## ۴-۲ ابزار ApexSQL

ApexSQL شامل ابزارهای مختلفی است که برخی از آن‌ها همچون موارد زیر در جرم‌شناسی کاربرد دارند:

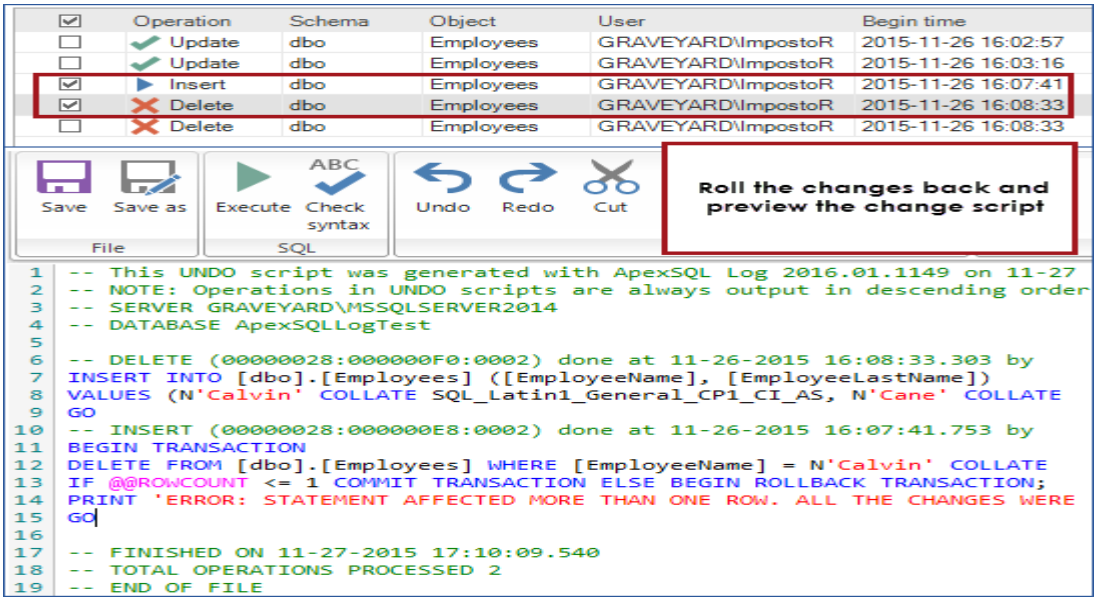
۱. ApexSQL Log
۲. ApexSQL Audit
۳. ApexSQL Recover

هیچ‌یک از این ابزارها متن‌باز نیستند.

### ۴-۲-۱ ابزار ApexSQL Log

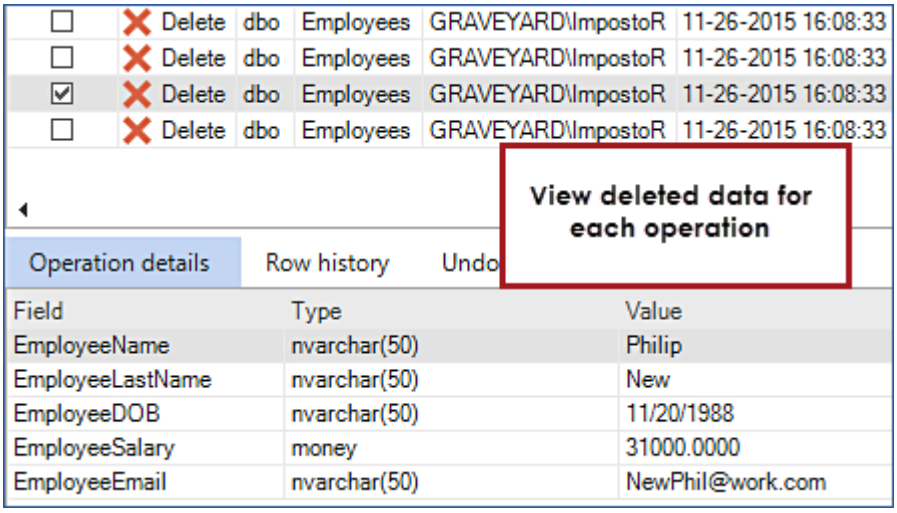
این ابزار در موارد زیر کاربرد دارد [۱۲]:

۱. تهیه ممیزی از تغییرات در داده‌ها، شِما و مجوزها
۲. عقب‌گرد یا اجرای مجدد تراکنش پایگاه داده (شکل ۱۵)




**شکل ۱۵ عقب‌گرد یا اجرای مجدد تراکنش پایگاه داده**

۳. اجرای معکوس تراکنش‌های غیرعمدی یا مخرب پایگاه داده به‌منظور ترمیم داده: بازیابی داده‌های خراب‌شده یا ازدست‌رفته، بدون تکیه‌بر بازگرداندن کل پایگاه داده (شکل ۱۶).

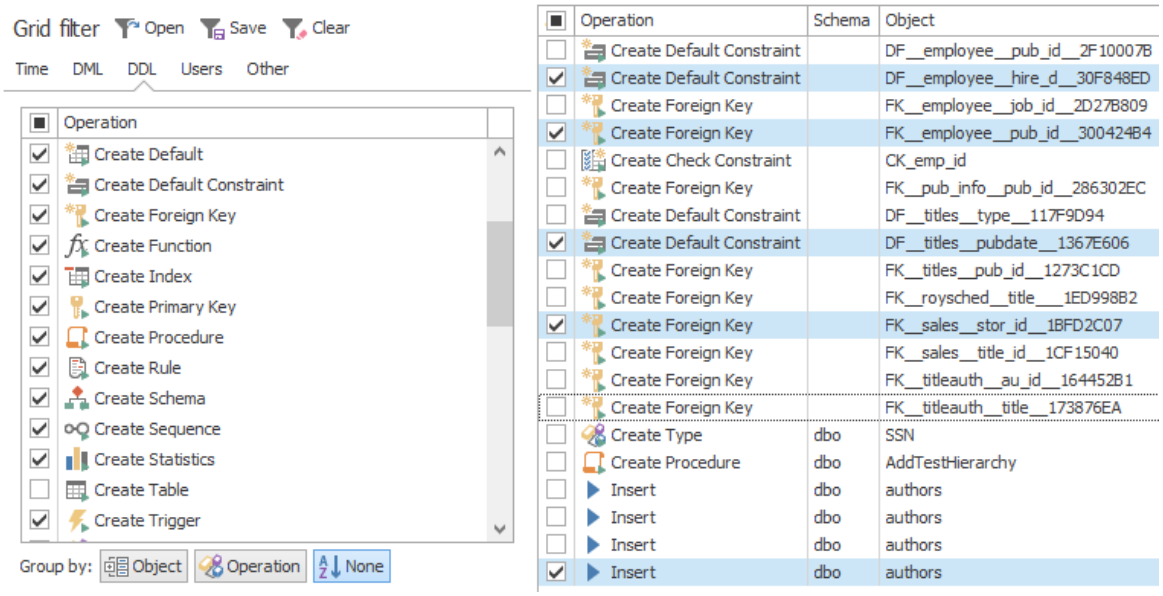



**شکل ۱۶ اجرای معکوس تراکنش‌های غیرعمدی یا مخرب پایگاه داده**

۴. بررسی قانونی اینکه چه کسی، چه چیزی را در چه زمانی تغییر داده است.

۵. نمایش تاریخچه کامل تغییرات سطرها

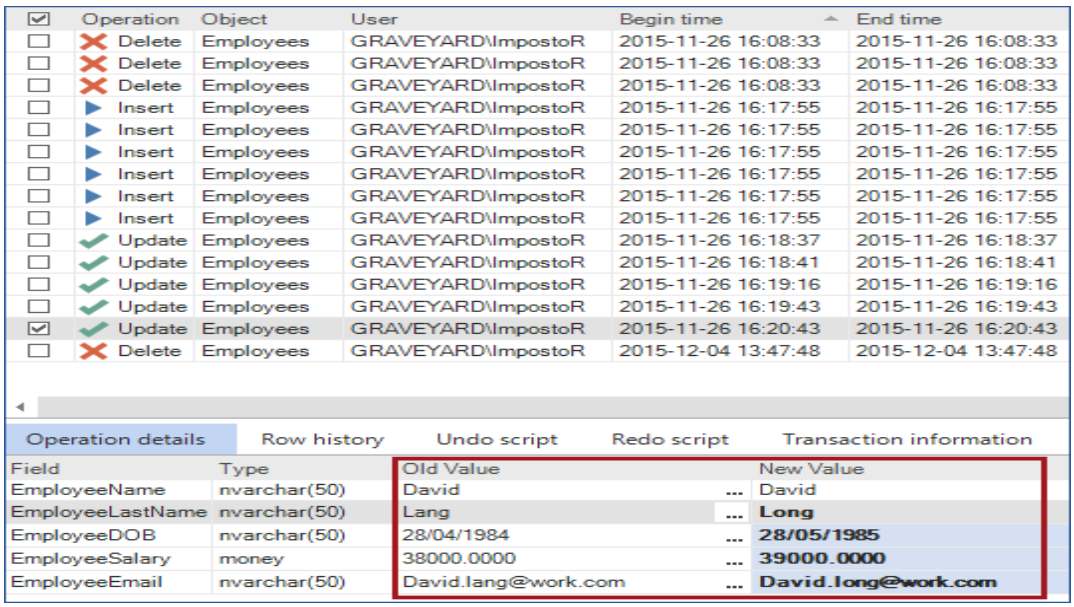
۶. تجسم تراکنش[۳۱]: نمایش، دسته‌بندی و مرتب‌سازی تراکنش‌ها همراه با گزینه‌های پیشرفته برای فیلترینگ (شکل ۱۷)




**شکل ۱۷ تجسم تراکنش**

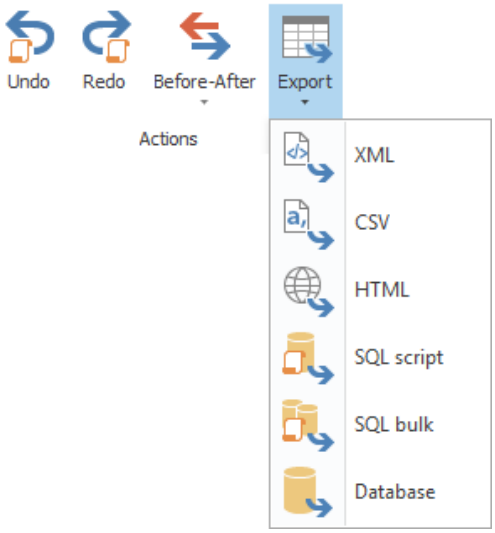
۷. نمایش مقادیر قبلی و بعدی در عملیات به‌روزرسانی (شکل ۱۸)

[31] Transaction visualization




**شکل ۱۸ نمایش مقادیر قبلی و بعدی در عملیات به‌روزرسانی**

۸. نمایش اطلاعات رویدادهای تراکنش در قالب جدول یا استخراج آن در قالب‌های مختلف همچون HTML، CSV، XML، SQL script (شکل ۱۹)

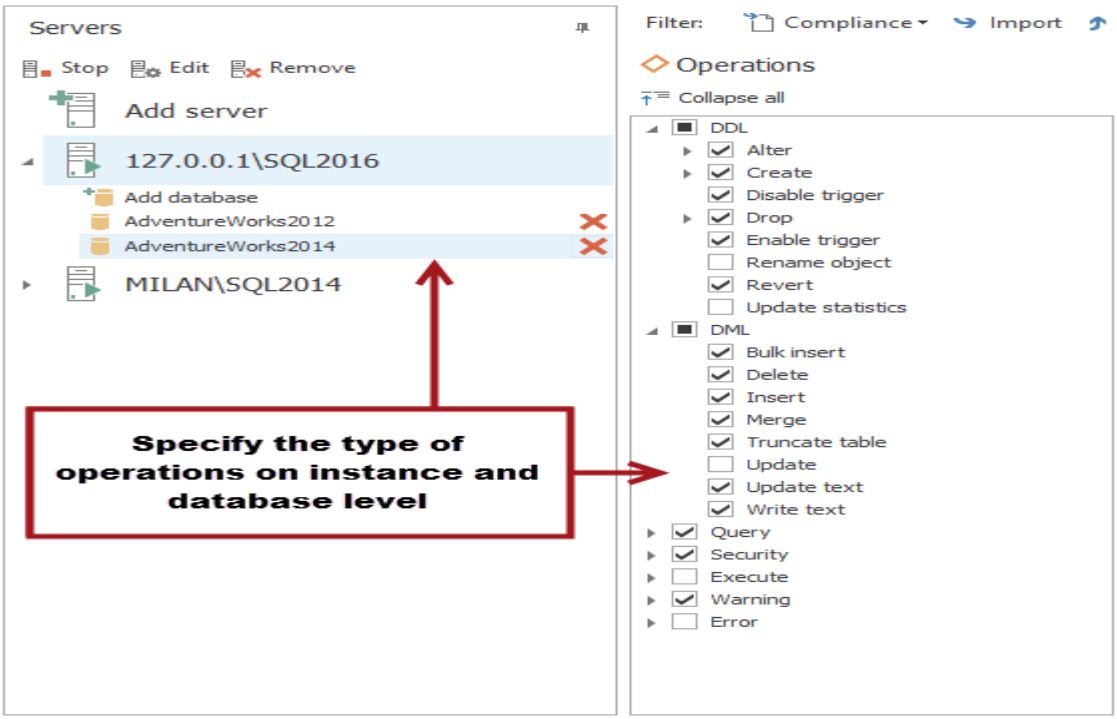



**شکل ۱۹ استخراج اطلاعات رویدادهای تراکنش**

### ۴-۲-۲ ابزار ApexSQL Audit

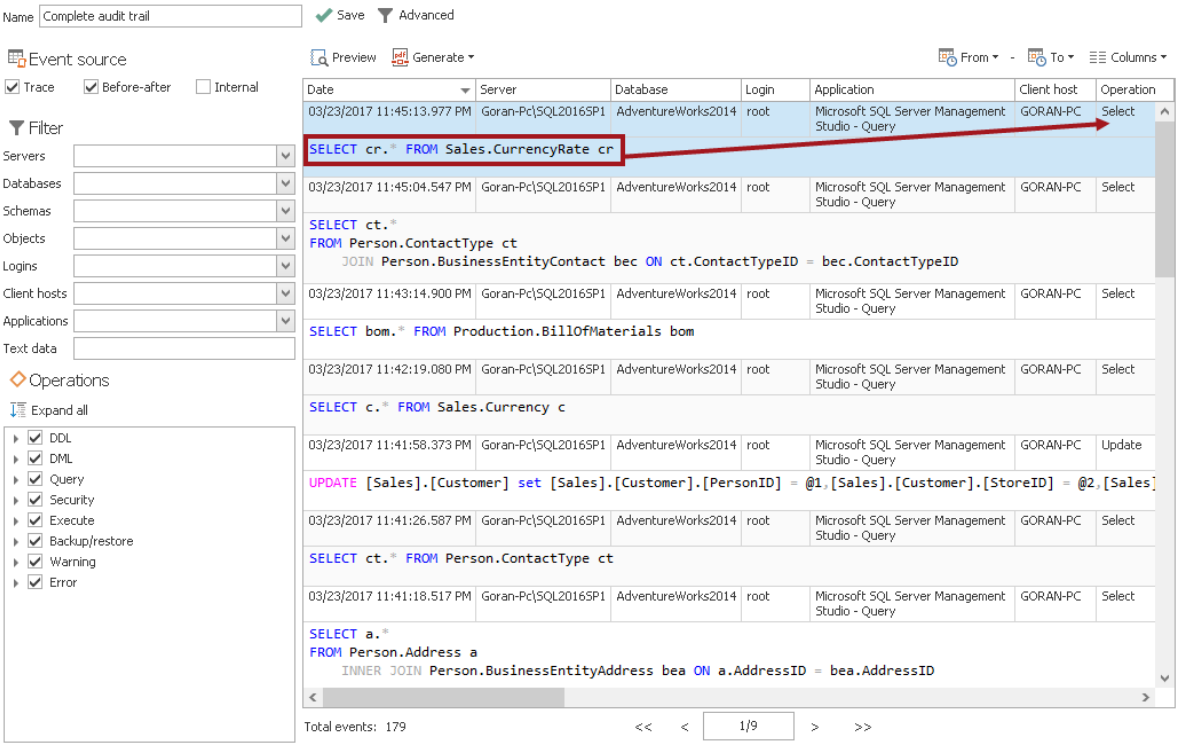
ابزار ApexSQL Audit برای تهیه ممیزی کاربرد داشته که ازجمله ویژگی‌های آن می‌توان به موارد زیر اشاره کرد [۱۲]:

۱. ممیزی از فعالیت‌های SQL Server: تهیه‌ی ممیزی از تمامی عملیات اجراشده بر روی SQL instance شامل تغییرات در DDL و DML، عبارات SELECT و فعالیت‌های مرتبط با ورود به پایگاه داده، کاربران و مجوزها (شکل ۲۰).



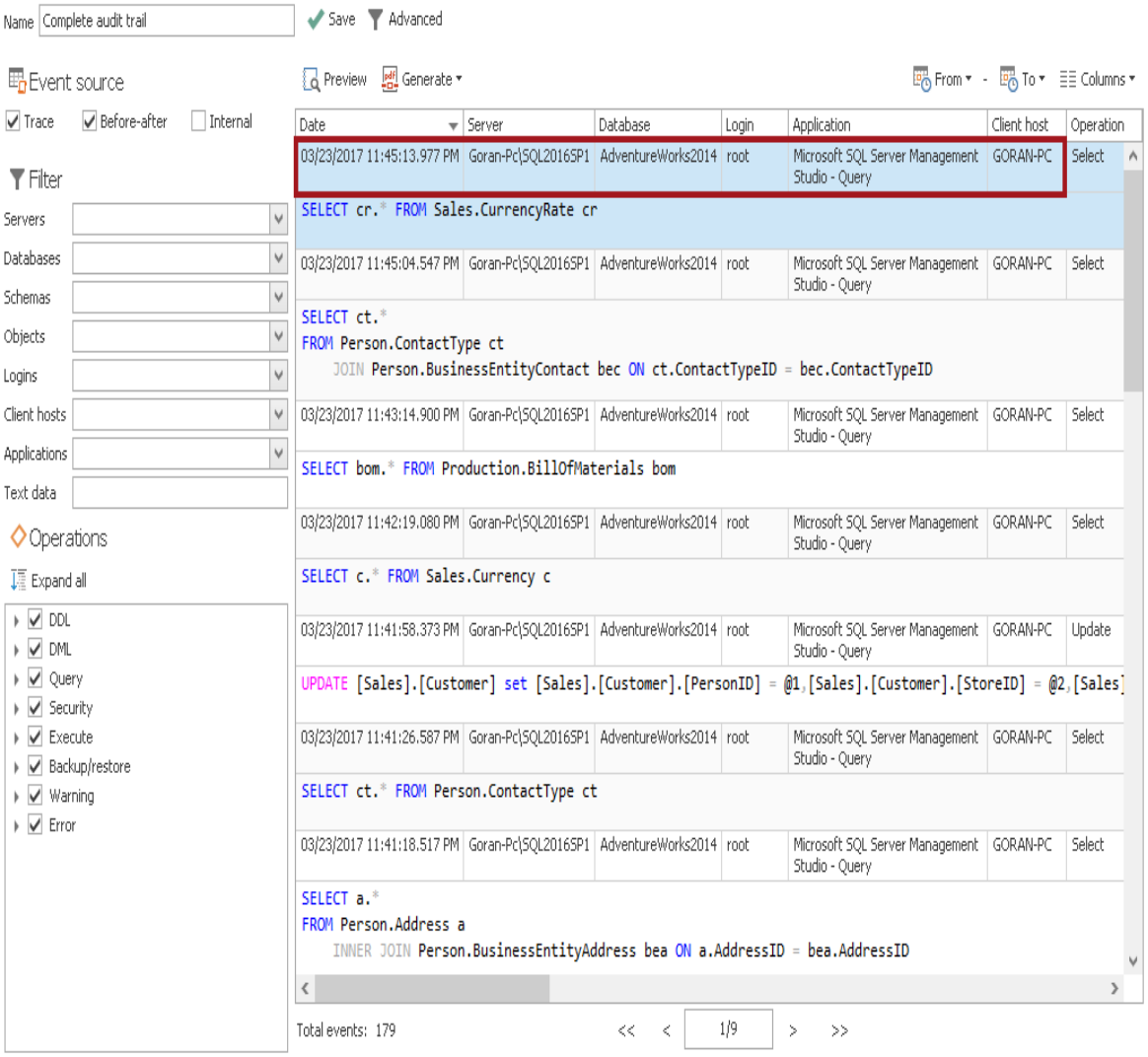

**شکل ۲۰ ممیزی از فعالیت‌های SQL Server**

۲. نمایش عبارات SQL اجراشده (شکل ۲۱)



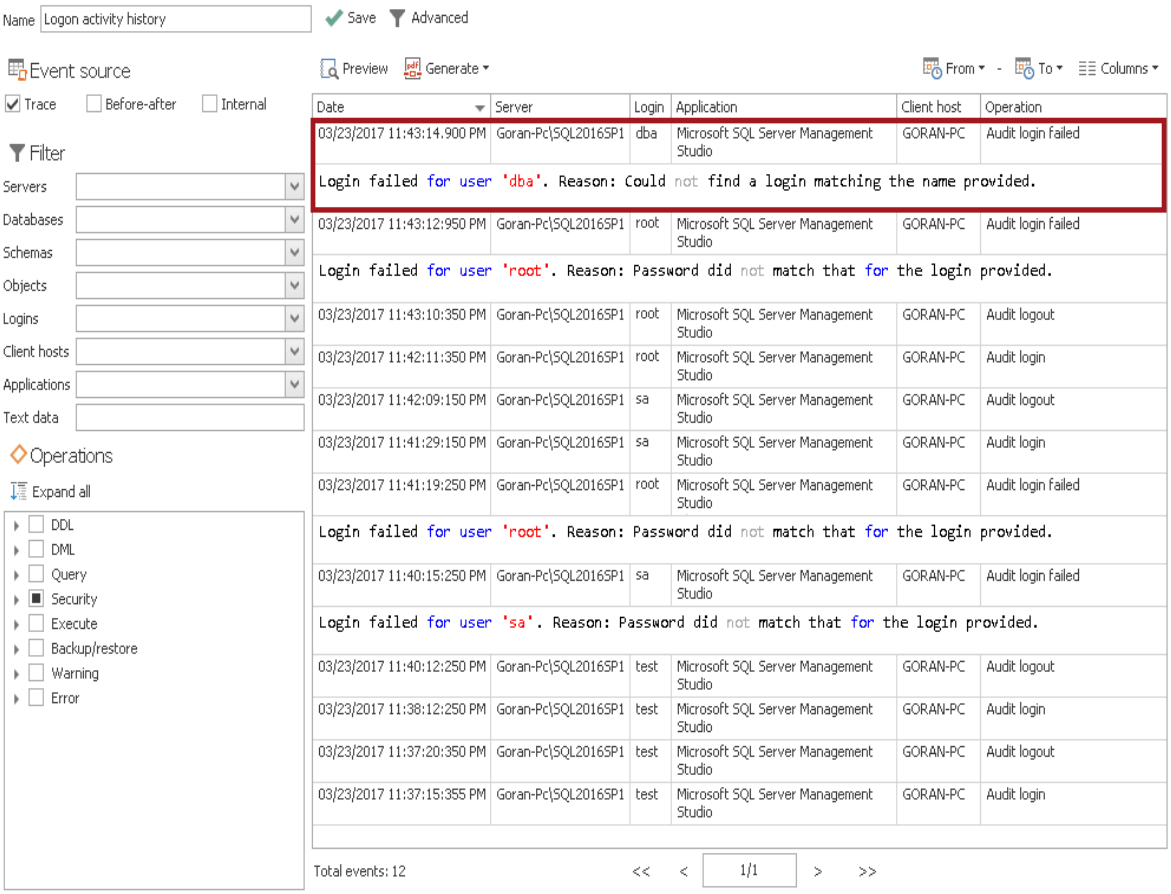

**شکل ۲۱ نمایش عبارات SQL اجراشده**

۳. نمایش اینکه چه کسی، چه چیزی را در چه زمانی انجام داده است (شکل ۲۲)




**شکل ۲۲ نمایش جزئیات مربوط به تراکنش‌ها**

۴. نمایش تلاش‌های ناموفق برای ورود (شکل ۲۳)




**شکل ۲۳ نمایش تلاش‌های ناموفق برای ورود**

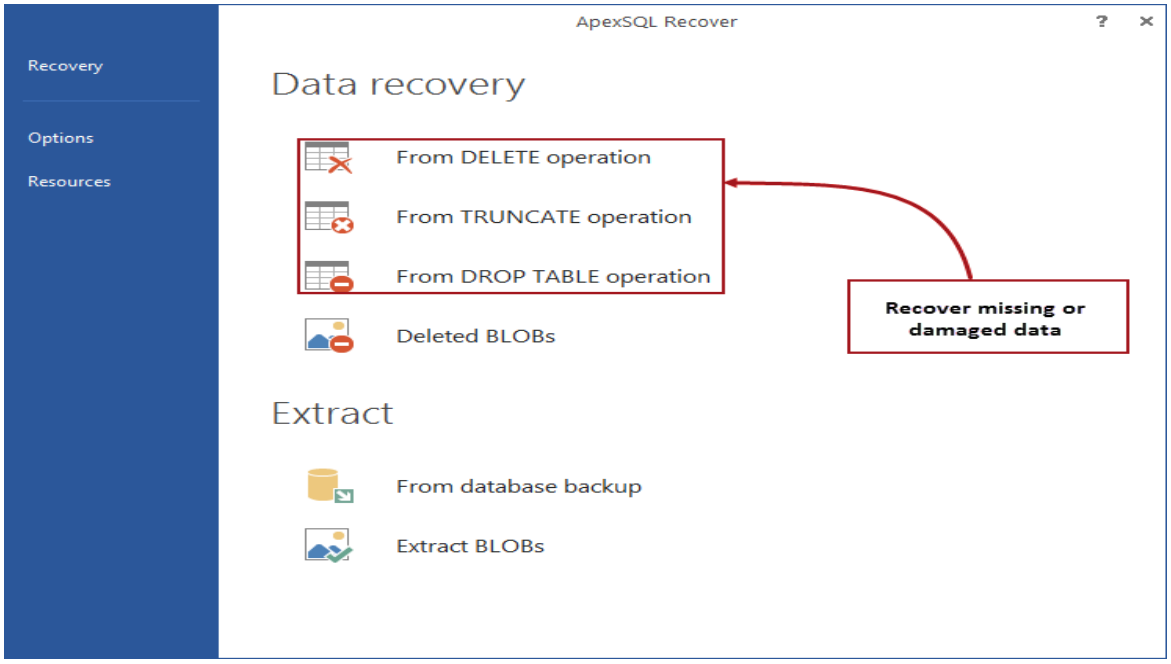
۵. تهیه ممیزی از عملیات درج، به‌روزرسانی و حذف، قبل و بعد از انجام عملیات

۶. مشخص کردن تغییر در مجموعه‌ای از داده‌ها

### ۳-۲-٤ ابزار ApexSQL Recover

ابزار ApexSQL Recover برای ترمیم و بازیابی داده‌های حذف‌شده کاربرد دارد. ازجمله ویژگی‌های این ابزار می‌توان به موارد زیر اشاره کرد [۱۲]:

۱. بازیابی داده‌های ازدست‌رفته با عملیات DELETE، TRUNCATE یا DROP TABLE (شکل ۲۴)




**شکل ۲۴ بازیابی داده‌های ازدست‌رفته**

۲. بازیابی داده‌ها مستقیماً در پایگاه داده: بازیابی جداول ازدست‌رفته مستقیماً در پایگاه داده

۳. بازیابی BLOB-های حذف‌شده و استخراج آن‌ها از فایل‌های داده

۴. فیلترینگ با جزئیات و تعیین محدوده‌ی زمانی که داده یا اشیا ازدست‌رفته‌اند

۵. بازیابی ساختارهای جداول ازدست‌رفته

# ۵ جمع‌بندی

بر روی کارگزار SQL، فایل‌های رویدادنگاری مختلفی ایجاد می‌شوند که در بررسی مشکلات پیش‌آمده و مسائل مربوط به کارایی و خطاها کمک خواهند کرد. ویژگی ممیزی در SQL Server امکان تهیه‌ی ممیزی از تمام اتفاقات رخ داده در کارگزار را فراهم می‌کند. رویدادها از تغییرات در تنظیمات کارگزار تا تغییر در مقداری در جدول خاصی از پایگاه داده را شامل می‌شوند. در SQL Server 2012 امکان ممیزی در سطح کارگزار در تمامی نسخه‌ها وجود دارد؛ ولی ممیزی پایگاه داده تنها در نسخه‌های Developer، Evaluation و Enterprise قابل‌اعمال است.

روش‌های جرم‌شناسی موجود در MSSQL Server، می‌توانند برای بررسی دسترسی‌های غیرمجاز به داده‌ها، کلاه‌برداری و جمع‌آوری اطلاعات برای تشخیص نفوذ استفاده شوند. در حقیقت جرم‌شناسی در SQL Server برای تشخیص تغییرات ایجادشده در پایگاه داده و منشأ اصلی تغییرات و حملات کاربرد دارد.

بیش‌تر ابزارهای جرم‌شناسی SQL Server در حالت برون‌خط اجرا می‌شوند؛ زیرا تحلیل در حالت زنده و آنلاین ممکن است باعث توقف فعالیت کارگزار و از دست رفتن داده‌ها شود.

# ٦ منابع


[1]. https://docs.mendix.com/refguide5/review-log-files-ms-sql-server

[2]. https://docs.microsoft.com/en-us/sql/tools/configuration-manager/viewing-the-sql-server-error-log

[3]. https://www.mssqltips.com/sqlservertip/1476/reading-the-sql-server-log-files-using-tsql/

[4]. http://sqlmag.com/database-administration/sql-server-log-files-update

[5]. https://www.red-gate.com/simple-talk/sql/learn-sql-server/managing-transaction-logs-in-sql-server/

[6]. https://www.sqlshack.com/understanding-sql-server-audit/

[7]. https://docs.microsoft.com/en-us/sql/relational-databases/security/auditing/sql-server-audit-database-engine

[8]. https://solutioncenter.apexsql.com/how-to-setup-and-use-sql-server-audit-feature/

[9]. https://www.systoolsgroup.com/forensics/sql-server/

[10]. http://www.dataforensics.org/sql-mdf-forensics/

[11]. http://www.xploreforensics.com/blog/sql-server-mdf.html

[12]. https://www.apexsql.com