## 核心技术

1. **创建跟踪**

https://msdn.microsoft.com/zh-cn/library/ms190362(v=sql.110).aspx

sp\_trace\_create：创建跟踪定义。  新的跟踪将处于停止状态。

语法：sp\_trace\_create [ @traceid = ] trace\_id OUTPUT

          , [ @options = ] option\_value

          , [ @tracefile = ] 'trace\_file'

     [ , [ @maxfilesize = ] max\_file\_size ]

     [ , [ @stoptime = ] 'stop\_time' ]

     [ , [ @filecount = ] 'max\_rollover\_files' ]

[ @traceid= ] trace\_id ：

Microsoft SQL Server 为新跟踪分配的编号。  用户提供的任何输入都被忽略。 trace\_id 的数据类型为 int，默认值为 NULL。 用户使用 trace\_id 值标识、修改和控制此存储过程定义的跟踪

[ @options= ] option\_value

指定为跟踪设置的选项。  option\_value 的数据类型为 int，无默认值。 用户可以通过指定所选出选项之和来选择这些选项的组合。 例如，若要启用 TRACE\_FILE\_ROLLOVER 和 SHUTDOWN\_ON\_ERROR，请为 option\_value 指定 6。

TRACE\_FILE\_ROLLOVER : 2

指定当达到 max\_file\_size 时，将关闭当前跟踪文件并创建新文件。  所有新记录都将写入新文件。 新文件将与前一个文件同名，但是在文件名后将附加一个整数以指示其序列。 例如，如果命名原始跟踪文件为 filename.trc，则命名下个跟踪文件为 filename\_1.trc，命名再下一个跟踪文件为 filename\_2.trc，等等。

随着更多滚动更新跟踪文件的创建，附加到文件名的整数值继续增加。

如果指定该选项时未指定 max\_file\_size 的值，则 SQL Server 使用 max\_file\_size 的默认值 (5 MB)。

SHUTDOWN\_ON\_ERROR  :4

指定无论任何原因，如果不能将跟踪写入文件，则 SQL Server 将关闭。  执行安全审核跟踪时，该选项很有用。

TRACE\_PRODUCE\_BLACKBOX :8

指定服务器产生的最后 5 MB 跟踪信息记录将由服务器保存。  TRACE\_PRODUCE\_BLACKBOX 与所有其他选项不兼容。

[ @tracefile= ] 'trace\_file'

指定跟踪将写入的位置和文件名。  trace\_file 的数据类型为 nvarchar(245)，无默认值。 trace\_file 可以是本地目录(例如 N 'C:\MSSQL\Trace\trace.trc')，或共享或路径的 UNC (N'\\Servername\Sharename\Directory\trace.trc')。

SQL Server 会将 .trc 扩展名附加到所有的跟踪文件名。  如果已指定 TRACE\_FILE\_ROLLOVER 选项和 max\_file\_size，则当原始跟踪文件增长到其最大值时，SQL Server 将创建新的跟踪文件。 新文件与原始文件同名，但是在其后附加 \_n 以指示它的序列，序列从 1 开始。 例如，如果第一个跟踪文件名为 filename.trc，则将第二个跟踪文件命名为 filename\_1.trc。

如果您使用 TRACE\_FILE\_ROLLOVER 选项，我们建议您在原始跟踪文件名中不要使用下划线字符。  如果您使用了下划线，则会发生以下行为：

• SQL Server Profiler 不自动加载或者提示您加载滚动更新文件（如果配置了上述文件滚动更新选项之一）。

• fn\_trace\_gettable 函数将不会加载原始文件名以下划线和数值结尾的滚动更新文件（在通过使用 number\_files 参数指定时）。  （这不适用于在文件滚动更新时自动追加的下划线和数字。）

[ @maxfilesize= ] max\_file\_size

指定跟踪文件可以增长到的最大文件大小 (MB)。  max\_file\_size 的数据类型为 bigint，默认值为 5。

如果指定该参数时未带有 TRACE\_FILE\_ROLLOVER 选项，则当使用的磁盘空间超过 max\_file\_size 指定的空间量时跟踪将停止对文件的记录。

[ @stoptime= ] 'stop\_time'

指定停止跟踪的日期和时间。  stop\_time 的数据类型为 datetime，默认值为 NULL。 如果为 NULL，该跟踪将一直运行，直到它被手动停止或服务器关闭。

如果同时指定 stop\_time 和 max\_file\_size，但不指定 TRACE\_FILE\_ROLLOVER，则当达到指定停止时间或最大文件大小时跟踪将停止。  如果同时指定了 stop\_time、max\_file\_size 和 TRACE\_FILE\_ROLLOVER，并假设跟踪未填满驱动器，则跟踪将在指定停止时间停止。

[ @filecount= ] 'max\_rollover\_files'

指定使用同一基准文件名维护的最大跟踪文件数。  max\_rollover\_files 的数据类型为 int，其值大于 1。 此参数仅在指定了 TRACE\_FILE\_ROLLOVER 选项时有效。 指定 max\_rollover\_files 后，SQL Server 会在打开新的跟踪文件前删除最旧的跟踪文件，以便维护不超过 max\_rollover\_files 个跟踪文件。 SQL Server 通过向基准文件名追加数字来跟踪跟踪文件的新旧程度。

例如，将 trace\_file 参数指定为“c:\mytrace”时，名为“c:\mytrace\_123.trc”的文件比名为“c:\mytrace\_124.trc”的文件旧。  如果 max\_rollover\_files 设置为 2，则 SQL Server 在创建跟踪文件“c:\mytrace\_125.trc”前删除文件“c:\mytrace\_123.trc”。

请注意，SQL Server 一次仅尝试删除一个文件，且不能删除另一进程正在使用的文件。  因此，如果在跟踪运行时另一应用程序正在使用跟踪文件，则 SQL Server 会将这些跟踪文件保留在文件系统中。

返回值：

0 ：没有错误。

1 ：未知错误。

10 ：无效选项。  指定的选项不兼容时返回此代码。

12 ：文件未创建。

13 ：内存不足。  在没有足够内存执行指定的操作时返回此代码。

14 ：无效停止时间。  在指定的停止时间已发生时返回此代码。

15 ：参数无效。  在用户已提供不兼容的参数时返回此代码。

1. **修改跟踪**

https://msdn.microsoft.com/library/aa260311.aspx

sp\_trace\_setstatus：修改指定跟踪的当前状态。

语法：sp\_trace\_setstatus [ @traceid = ] trace\_id , [ @status = ] status

status可取值：

0 停止指定的跟踪。

1 启动指定的跟踪。

2 关闭指定的跟踪并从服务器中删除其定义 。

返回值：

0 没有错误。

1 未知错误。

8 指定的状态无效。

9 指定的跟踪句柄无效。

13 内存不足。  在没有足够内存执行指定的操作时返回此代码。

1. **查询跟踪**

首先得开始跟踪，才能进行查询，查询前首先创建跟踪文件c:\master.mdf..2016.06.21.10.00.23.trc

select \* from (select ROW\_NUMBER() OVER (order by StartTime) as RowNum, \* from fn\_trace\_gettable('c:\master.mdf..2016.06.21.10.00.23.trc', default)) as dt where RowNum > 0 ORDER BY RowNum

## SQL跟踪事件

（参考：https://msdn.microsoft.com/zh-cn/library/ms175481(v=sql.110).aspx）

|  |  |
| --- | --- |
| RPCCompleted | 某个远程过程调用已经完成。 |
| RPCStarting | 远程过程调用已启动。 |
| SQLBatchCompleted | Transact-SQL 批处理已完成。 |
| SQLBatchStarting | 正在启动 Transact-SQL 批处理。 |
| AuditLogin | 用户已成功登录到 MicrosoftSQL Server。 此类中的事件由新连接或从连接池中重用的连接触发。 |
| AuditLogout | 用户已注销 MicrosoftSQL Server。 此类中的事件由新连接或从连接池中重用的连接触发。 |
| LockReleased | 已释放某个资源（例如页）的锁。  Lock:Acquired 和 Lock:Released 事件类可以用于监视锁定对象的时间、使用的锁类型以及持有锁的时间。 保留较长时间的锁可能导致争用问题，应进行调查。 例如，应用程序可以为表中的行获取锁，然后等待用户输入。 由于用户输入可能需要较长时间，所以锁可能会阻塞其他用户。 在这种情况下，应重新设计应用程序，以便只在需要时才请求锁，并且在获取锁后不需要用户输入。 |
| LockAcquired | 已获取某个资源（如数据页）的锁。  Lock:Acquired 和 Lock:Released 事件类可以用于监视锁定对象的时间、使用的锁类型以及持有锁的时间。 保留较长时间的锁可能导致争用问题，应进行调查。 例如，应用程序可以为表中的行获取锁，然后等待用户输入。 由于用户输入可能需要较长时间，所以锁可能会阻塞其他用户。 在这种情况下，应重新设计应用程序，以便只在需要时才请求锁，并且在获取锁后不需要用户输入。 |
| LockDeadlock | 对于死锁中的每个参与者都会产生 Lock:Deadlock Chain 事件类。  使用 Lock:Deadlock Chain 事件类可以监视何时出现死锁情况。 此信息有助于确定死锁是否会对应用程序的性能造成重大影响，以及会涉及哪些对象。 可以检查用于修改这些对象的应用程序代码，以便确定是否可以做出更改以便将死锁的情况减到最少。 |
| LockCancel | 已取消获取某个资源的锁；例如，由于查询被取消。 |
| LockTimeout | Lock:Timeout 事件类指示由于其他事务持有所需资源的阻塞锁而使对资源（例如页）锁的请求超时。  超时由 @@LOCK\_TIMEOUT 系统函数决定，可用 SET LOCK\_TIMEOUT 语句设置。 超时情况出现时，使用 Lock:Timeout 事件类进行监视。 此信息有助于确定超时是否对应用程序的性能造成重大影响，以及涉及哪些对象。 您可以检查修改这些对象的应用程序代码，以确定是否可以进行更改以将超时减到最小。持续时间为 0 的 Lock:Timeout 事件通常是内部锁探测的结果，并不表示存在问题。 可以使用 Lock:Timeout (timeout > 0) 事件以忽略持续时间为 0 的超时。 |
| SQLStmtStarting | Transact-SQL 语句已开始执行。 |
| SQLStmtCompleted | Transact-SQL 语句已完成。 |
| SPStarting | 存储过程将要开始执行。 |
| SPCompleted | 存储过程已执行完毕。 |
| SPStmtStarting | 已开始执行存储过程中的 Transact-SQL 语句。 |
| SPStmtCompleted | 存储过程中的 Transact-SQL 语句已执行完毕。 |
| SQLTransaction | 可以监视事务开始和完成的时间，尤其是当您测试应用程序、触发器或存储过程时。 |
| ScanStarted | 开始扫描表或索引时，会发生 Scan:Started 事件类。 |
| ScanStopped | 停止扫描表或索引时，会发生 Scan:Stopped 事件类。 |
| CursorOpen | 说明在应用程序编程接口 (API) 游标中发生的游标打开事件。  SQL Server 数据库引擎定义与游标和游标选项相关联的 SQL 语句时发生游标打开事件，然后填充游标。   在记录游标性能的跟踪中包括 CursorOpen 事件类。 在跟踪中包括 CursorOpen 事件类时，产生的开销取决于跟踪期间数据库中使用游标的频率。 如果广泛使用游标，则跟踪可能会显著地降低性能。 |
| TransactionLog | 可以监视 SQL Server 数据库引擎实例的事务日志中的活动。 |
| LockDeadlockChain | 对于死锁中的每个参与者都会产生 Lock:Deadlock Chain 事件类。   使用 Lock:Deadlock Chain 事件类可以监视何时出现死锁情况。 此信息有助于确定死锁是否会对应用程序的性能造成重大影响，以及会涉及哪些对象。 可以检查用于修改这些对象的应用程序代码，以便确定是否可以做出更改以便将死锁的情况减到最少。 |
| LockEscalation | 较细粒度的锁已转换为较粗粒度的锁；例如，行锁已转换为对象锁。 升级事件类是事件 ID 为 60 的事件类。 |
| ExecutionWarnings | 在执行 SQL Server 语句或存储过程期间出现的内存授予警告。   监视此事件类可确定查询在继续进行之前是否必须等待一秒或几秒再获取内存，或确定获取内存的初始尝试是否失败。 查询等待时间的信息有助于揭示影响性能的系统争用问题。 |
| SQLFullTextQuery | 当 SQL Server 执行全文查询时，会发生 SQL:FullTextQuery 事件类。  请将此事件类包括在监视与全文目录相关的问题的跟踪中。  当包含 SQL:FullTextQuery 事件类时，开销量将很高。 如果此类事件频繁发生，则跟踪可能会显著地降低性能。 为了尽量减小这种影响，此事件类的使用应限于短期监视特定问题的跟踪。 |
| DeadlockGraph | 有关死锁的 XML 描述。 该类与 Lock:Deadlock 事件类同时发生。 |
| SQLStmtRecompile | 由下列所有类型的批处理引起的语句级重新编译：存储过程、触发器、即席批查询和查询。  可以通过使用 sp\_executesql、动态 SQL、“准备”方法、“执行”方法或类似接口来提交查询。 应该用 SP:Recompile 事件类来代替 SQL:StmtRecompile 事件类。 |
| TMBeginTranStarting | 正在启动 BEGIN TRANSACTION 请求。 将通过事务管理接口从客户端发送请求。 |
| TMBeginTranCompleted | 已完成 BEGIN TRANSACTION 请求。 该请求是通过事务管理界面从客户端发送的。 |
| TMPromoteTranStarting | 正在启动 PROMOTE TRANSACTION 请求。 将通过事务管理接口从客户端发送请求。 |
| TMPromoteTranCompleted | PROMOTE TRANSACTION 请求已完成。 将通过事务管理接口从客户端发送请求。 |
| TMCommitTranStarting | 正在启动 COMMIT TRANSACTION 请求。 将通过事务管理接口从客户端发送请求。EventSubClass 列指示在提交当前事务之后是否启动新事务。 |
| TMCommitTranCompleted | COMMIT TRANSACTION 请求已完成。 该请求是通过事务管理界面从客户端发送的。EventSubClass 列指示在提交当前事务之后是否启动新事务。 |
| TMRollbackTranStarting | 正在启动 ROLLBACK TRANSACTION 请求。 客户端通过事务管理界面发送请求。EventSubClass 列指示在当前事务回滚后是否启动新事务。 |
| TMRollbackTranCompleted | ROLLBACK TRANSACTION 请求已完成。 该请求是通过事务管理界面从客户端发送的。EventSubClass 列指示在当前事务回滚后是否启动新事务。 |
| TMSaveTranstarting | 正在启动 SAVE TRANSACTION 请求。 该请求是从客户端通过事务管理界面发送的。 |
| TMSaveTrancompleted | SAVE TRANSACTION 请求已完成。 该请求是通过事务管理界面从客户端发送的。 |

## 过滤字段说明

(参考：https://msdn.microsoft.com/zh-cn/library/hh245121(v=sql.110).aspx)

|  |  |
| --- | --- |
| TextData | 依赖于跟踪中捕获的事件类的文本值。 但是，如果跟踪参数化查询，则不以 **TextData** 列中的数据值显示变量。 |
| BinaryData | 依赖于跟踪中捕获的事件类的二进制值 |
| DatabaseID | 由 USE database\_name 语句指定的数据库的 ID；如果未对给定实例发出 USE database\_name语句，则为默认数据库的 ID。 如果在跟踪中捕获 **ServerName** 数据列而且服务器可用，则 SQL Server Profiler 将显示数据库名。 若要确定数据库的值，请使用 DB\_ID 函数。 |
| TransactionID | 系统为事务分配的 ID。 |
| NTUserName | Windows 用户名。 |
| NTDomainName | 用户所属的 Microsoft Windows 域。 |
| HostName | 正在运行客户端程序的计算机的名称。 如果客户端提供了主机名，则填充此数据列。 若要确定主机名，请使用 HOST\_NAME 函数。 |
| ClientProcessID | 由主机分配给正在运行客户端应用程序的进程的 ID。 如果客户端提供了客户端进程 ID，则填充此数据列。 |
| ApplicationName | 客户端应用程序的名称，该客户端应用程序创建了指向 SQL Server 实例的连接。 此列由该应用程序传递的值填充，而不是由程序名填充的。 |
| LoginName | 用户的登录名（SQL Server 安全登录名或 Windows 登录凭据，格式为“域/用户名”）。 |
| SPID | SQL Server 为客户端的相关进程分配的服务器进程 ID (SPID)。 |
| Duration | 事件的持续时间（微秒）。 |
| StartTime | 事件（如果有）的开始时间。 |
| EndTime | 事件的结束时间。 对指示事件开始的事件类（例如 **SQL:BatchStarting** 或**SP:Starting**）将不填充此列。 |
| Reads | 由服务器代表事件读取逻辑磁盘的次数。 这些读取操作数包含在语句执行期间读取表和缓冲区的次数。 |
| Writes | 由服务器代表事件写入物理磁盘的次数。 |
| CPU | 事件使用的 CPU 时间（毫秒）。 |
| ObjectID | 系统分配的对象 ID。 |
| ServerName | 正在跟踪的 SQL Server 实例的名称。 |
| EventClass | 捕获的事件类的类型。 |
| ProviderName | OLEDB 访问接口的名称。 |
| LoginSID | 已登录的用户的安全标识符 (SID)。 您可以在 **master** 数据库的 **sys.server\_principals** 视图中找到此信息。 服务器中的每个登录名都具有唯一的 ID。 |
| DataBaseName | 正在运行用户语句的数据库的名称。 |

|  |  |
| --- | --- |
| RPCCompleted | 某个远程过程调用已经完成。 |
| RPCStarting | 远程过程调用已启动。 |
| SQLBatchCompleted | Transact-SQL 批处理已完成。 |
| SQLBatchStarting | 正在启动 Transact-SQL 批处理。 |
| AuditLogin | 用户已成功登录到 MicrosoftSQL Server。 此类中的事件由新连接或从连接池中重用的连接触发。 |
| AuditLogout | 用户已注销 MicrosoftSQL Server。 此类中的事件由新连接或从连接池中重用的连接触发。 |
| LockReleased | 已释放某个资源（例如页）的锁。 |
| LockAcquired | 已获取某个资源（如数据页）的锁。 |
| LockDeadlock | 对于死锁中的每个参与者都会产生 Lock:Deadlock Chain 事件类。 |
| LockCancel | 已取消获取某个资源的锁；例如，由于查询被取消。 |
| LockTimeout | Lock:Timeout 事件类指示由于其他事务持有所需资源的阻塞锁而使对资源（例如页）锁的请求超时。 |
| SQLStmtStarting | Transact-SQL 语句已开始执行。 |
| SQLStmtCompleted | Transact-SQL 语句已完成。 |
| SPStarting | 存储过程将要开始执行。 |
| SPCompleted | 存储过程已执行完毕。 |
| SPStmtStarting | 已开始执行存储过程中的 Transact-SQL 语句。 |
| SPStmtCompleted | 存储过程中的 Transact-SQL 语句已执行完毕。 |
| SQLTransaction | 可以监视事务开始和完成的时间，尤其是当您测试应用程序、触发器或存储过程时。 |
| ScanStarted | 开始扫描表或索引时，会发生 Scan:Started 事件类。 |
| ScanStopped | 停止扫描表或索引时，会发生 Scan:Stopped 事件类。 |
| CursorOpen | 说明在应用程序编程接口 (API) 游标中发生的游标打开事件。 |
| TransactionLog | 可以监视 SQL Server 数据库引擎实例的事务日志中的活动。 |
| LockDeadlockChain | 对于死锁中的每个参与者都会产生 Lock:Deadlock Chain 事件类。 |
| LockEscalation | 较细粒度的锁已转换为较粗粒度的锁；例如，行锁已转换为对象锁。 |
| ExecutionWarnings | 在执行 SQL Server 语句或存储过程期间出现的内存授予警告。 |
| SQLFullTextQuery | 当 SQL Server 执行全文查询时，会发生 SQL:FullTextQuery 事件类。 |
| DeadlockGraph | 有关死锁的 XML 描述。 该类与 Lock:Deadlock 事件类同时发生。 |
| SQLStmtRecompile | 由下列所有类型的批处理引起的语句级重新编译：存储过程、触发器、即席批查询和查询。 |
| TMBeginTranStarting | 正在启动 BEGIN TRANSACTION 请求。 将通过事务管理接口从客户端发送请求。 |
| TMBeginTranCompleted | 已完成 BEGIN TRANSACTION 请求。 该请求是通过事务管理界面从客户端发送的。 |
| TMPromoteTranStarting | 正在启动 PROMOTE TRANSACTION 请求。 将通过事务管理接口从客户端发送请求。 |
| TMPromoteTranCompleted | PROMOTE TRANSACTION 请求已完成。 将通过事务管理接口从客户端发送请求。 |
| TMCommitTranStarting | 正在启动 COMMIT TRANSACTION 请求。 将通过事务管理接口从客户端发送请求。EventSubClass 列指示在提交当前事务之后是否启动新事务。 |
| TMCommitTranCompleted | COMMIT TRANSACTION 请求已完成。 该请求是通过事务管理界面从客户端发送的。EventSubClass 列指示在提交当前事务之后是否启动新事务。 |
| TMRollbackTranStarting | 正在启动 ROLLBACK TRANSACTION 请求。 客户端通过事务管理界面发送请求。EventSubClass 列指示在当前事务回滚后是否启动新事务。 |
| TMRollbackTranCompleted | ROLLBACK TRANSACTION 请求已完成。 该请求是通过事务管理界面从客户端发送的。EventSubClass 列指示在当前事务回滚后是否启动新事务。 |
| TMSaveTranstarting | 正在启动 SAVE TRANSACTION 请求。 该请求是从客户端通过事务管理界面发送的。 |
| TMSaveTrancompleted | SAVE TRANSACTION 请求已完成。 该请求是通过事务管理界面从客户端发送的。 |