**Параметры аудита доступа к данным в SQL Server**

SQL Server предоставляет множество инструментов, которые можно использовать для аудита доступа к данным. Как правило, ни один инструмент не отвечаетвсем возможным требованиям аудита, и часто требуется сочетание функций.

Цели:

* Описание аудита доступа к данным.
* Описание функции соответствия общим критериям.
* Использование триггеров для аудита.
* Использование временных таблицы для аудита.

Включение соответствия общим критериям в SQL Server

SQL Server предоставляет параметр "Включено соответствие общим критериям", который можно задать с помощью

системной хранимой процедуры sp\_configure. Эта опция доступна в выпуске Enterprise для использования в производственных системах (она также доступна в выпусках для Developer и Evaluation выпусках для непроизводственного использования).

Когда этот параметр включен, в способ работы SQL Server вносятся три изменения:

* Остаточная защита информации (RIP). Память всегда перезаписывается известным битовым шаблоном перед повторным использованием. Внедрение RIP повышает безопасность, но может негативно повлиять на производительность системы.
* Возможность просмотра статистики входа в систему. Аудит учетных записей включается автоматически.
* Grant столбца не переопределяет Deny таблицы. Это изменяет поведение системы разрешений по умолчанию.

**Аудит с помощью триггеров**

Триггеры - это форма хранимой процедуры, которая запускается автоматически в ответ на событие. SQL Server поддерживает различные типы триггеров, включая:

* Триггеры языка обработки данных (DML). Эти триггеры связаны с таблицей и запускаются при изменении данных в таблице.
* Триггеры входа в систему. Триггер входа в систему запускается в ответ на событие входа в систему Триггеры языка определения данных (DDL). Эти триггеры связаны с инструкциями DDL, которые создают, изменяют или удаляют объекты базы данных.

Все эти типы триггеров могут играть определенную роль в аудите. Все типы триггеров создаются с помощью инструкции CREATE TRIGGER. В контексте аудита чаще всего используются триггеры AFTER.

**Триггеры DML**

Триггеры DML настроены на срабатывание при выполнении инструкций INSERT, UPDATE и/или DELETE для таблицы. Вы можете получить доступ к исходной и новой информации с помощью внутренних вставленных и удаленных таблиц в коде триггера. При использовании для аудита триггеры обычно используются для записи сведений об изменении в другую таблицу — таблица, содержащая проверенную копию данных, может находиться в другой базе данных на том же экземпляре SQL Server.

В следующем примере показан триггер DML, который запускается при обновлении строки в таблице dbo.Employee. Исходные данные регистрируются в таблице аудита заработной платы dbo.Employee.

CREATE TRIGGER TR\_Employee\_Salary\_UPDATE

ON dbo.Employee

FOR UPDATE

AS BEGIN SET NOCOUNT ON;

IF UPDATE(Salary) BEGIN

INSERT dbo.EmployeeSalaryAudit (EmployeeID, OldSalary, NewSalary, UpdatedBy, UpdatedAt)

SELECT i.EmployeeID, d.Salary, i.Salary, SUSER\_NAME(), GETDATE()

FROM inserted AS i

JOIN deleted AS d

ON i.EmployeeID = d.EmployeeID;

END;

END;

GO

**Триггеры входа** **в систему (Logon Triggers)**

Триггер входа срабатывает в ответ на вход, прошедший проверку подлинности, но до установления сеанса. Триггеры входа в систему можно использовать для записи входа в систему либо в таблицу, либо в журнал ошибок SQL Server.

**Триггеры языка определения данных DDL**

Триггер DDL срабатывает в ответ на событие DDL, которое создает, удаляет и/или изменяет либо определенный класс объекта базы данных, либо все объекты базы данных. В контексте аудита триггеры DDL обычно используются для отслеживания изменений в схеме базы данных путем записи инструкций DDL в таблицу.

**Ограничения**

Триггеры имеют некоторые ограничения в качестве инструмента аудита:

* На производительность системы могут существенно повлиять триггеры DML, работающие параллельно с обычной нагрузкой на сервер.
* Пользователи с соответствующими разрешениями могут отключать триггеры. Это может быть серьезной проблемой для требований аудита.
* Триггеры DML нельзя использовать для аудита доступа к данным с помощью инструкций SELECT.
* Предусмотрена только ограниченная возможность управления порядком срабатывания спускового крючка. Чтобы убедиться, что он фиксирует все изменения, внесенные другими триггерами, аудит обычно должен быть последним срабатывающим триггером DML, который можно указать только с помощью системной процедуры **sp\_settriggerorder**.

**Аудит с помощью временных таблиц**

Временные таблицы с системной версией предоставляют механизм для записи изменений данных в таблицу в таблицу истории; во многих отношениях это похоже на метод аудита изменений данных с помощью триггеров DML, за исключением того, что компонент database engine автоматически управляет таблицей истории.

Можно использовать функцию временной таблицы для записи всех изменений в ваших данных. Чтобы использовать эту функцию, начините с создания таблицы с системной версией, либо изменяя существующую.

При создании новой таблицы SQL Server фактически создает две таблицы — одну для хранения текущих данных и одну для хранения исторических данных. Два datetime2 столбца добавляются как в текущую, так и в историческую таблицы, в которых SQL Server хранит допустимый диапазон дат данных: SysStartTime и SysEndTime. Текущая строка будет иметь значение SYSENDTIME 9999-1231, и все записи, вставленные в одну транзакцию, будут иметь одинаковое время UTC. При обновление строки, SQL Server копирует данные в таблицу истории перед обновлением; он также устанавливает в столбце SysEndTime дату изменения данных. Затем он обновляет строку в текущей таблице, чтобы отразить это изменение.

Создание Временной Таблицы

CREATE TABLE dbo.Employee (

EmployeeID int NOT NULL PRIMARY KEY CLUSTERED,

ManagerID int NULL,

FirstName varchar(50) NOT NULL,

LastName varchar(50) NOT NULL, SysStartTime datetime2 GENERATED ALWAYS AS ROW START NOT NULL,

SysEndTime datetime2 GENERATED ALWAYS AS ROW END NOT NULL,

PERIOD FOR SYSTEM\_TIME (SysStartTime,SysEndTime)

) WITH (SYSTEM\_VERSIONING = ON (HISTORY\_TABLE = dbo.EmployeeHistory));

GO

При использовании в качестве механизма аудита временные таблицы имеют некоторые ограничения:

* Временные таблицы нельзя использовать для аудита доступа к данным с помощью инструкций SELECT.
* При аудите с временными таблицами вы не можете определять различные действия для инструкций INSERT, UPDATE и DELETE — в отличие от триггеров DML.
* Таблица истории, используемая для создания временной таблицы, должна быть создана в той же базе данных, что и текущая таблица.
* Отслеживание личности пользователя, внесшего изменение, требует, чтобы вы изменили определение таблицы, включив в нее столбец, по умолчанию имеющий значение SUSER\_SNAME.

Реализация аудита SQL Server

Цели:

* Описать расширенные события.
* Описать аудит SQL Server.
* Создать аудиты.
* Объясните действия по аудиту и группы действий.
* Создайте спецификации аудита сервера.
* Создайте спецификации аудита базы данных.
* Мониторинг аудитов с использованием связанных с аудитом динамических объектов управления и системных представлений.
* Используйте пользовательские события аудита.

Введение в Расширенные события

Аудит SQL Server основан на механизме мониторинга, управляемом событиями, называемом Расширенными событиями.

Расширенные события - это инструмент мониторинга активности, управляемый событиями, который следует шаблону проектирования со свободной связью. События и их цели не тесно связаны; любое событие может быть привязано к любой цели. Это означает, что обработка и фильтрация данных могут выполняться независимо от сбора данных, что снижает производительность других решений для аудита.

Расширенные события позволяют определять сложные фильтры для захваченных данных. В дополнение к фильтрам значений события могут быть отфильтрованы путем выборки. Данные могут быть объединены в момент их сбора. Расширенными событиями можно управлять либо с помощью графического интерфейса в среде SQL Server Management Studio, либо с помощью инструкций Transact-SQL.

Расширенные события могут быть интегрированы с платформой отслеживания событий для Windows (ETW - это служба, которая позволяет получать события от одного или нескольких поставщиков событий в режиме реального времени или из файла \*.etl за некоторый временной период), что означает, что активность SQL Server можно отслеживать вместе с другими компонентами Windows ® .

Расширенные события важны, поскольку аудит SQL Server основан на инфраструктуре расширенных событий. Механизм расширенных событий не привязан к определенным типам событий — механизм написан таким образом, что он может обрабатывать события любого типа.

Исполняемые файлы и исполняемые модули могут предоставлять один или несколько пакетов расширенных событий во время выполнения. Пакеты действуют как контейнеры для объектов расширенных событий и их определений; пакет может предоставлять доступ к любому из следующих типов объектов:

* Events (События). Точки интереса, достигнутые во время выполнения кода.
* Targets (Цели). Цели ведения журнала, такие как файлы в файловой системе.
* Actions (Действия). Дополнительные данные, которые могут быть собраны при срабатывании события.
* Predicates (Предикаты). Фильтры, используемые для ограничения того, какие события регистрируются.
* Types and Maps (Типы и карты). Справочные данные; определения типов данных и значений подстановки.
* Sessions (Сеансы). Связывает события и действия, отфильтрованные по предикатам, с одной или несколькими целями. Как правило, сеансы определяются пользователем.

Аудит SQL Server - это специальный пакет в рамках расширенных событий; вы не можете изменить его внутреннюю конфигурацию.

Введение в аудит SQL Server

Аудит SQL Server является основным инструментом аудита в SQL Server. Вы можете использовать его для отслеживания событий на уровне сервера и базы данных на экземпляре SQL Server, а затем регистрировать эти события в файлах аудита или журналах событий.

|  |  |
| --- | --- |
| Объект | Описание |
| Аудит сервера | Определяет, где будут храниться данные аудита. |
| Спецификация аудита сервера | Собирает множество групп действий на уровне сервера, созданных расширенными событиями. По одному на аудит. |
| Спецификация аудита базы данных | Собирает действия аудита на уровне базы данных, вызванные расширенными событиями. По одному на базу данных за аудит. |
| Действия | Конкретные действия, которые могут вызывать события и добавляться в аудит. Например, Select операции над таблицей. |
| Группы действий | Логические группы действий для упрощения создания спецификаций аудита. Например, BACKUP\_RESTORE\_GROUP, которая включает в себя любые команды резервного копирования или восстановления. |
| Цель | Получает и сохраняет результаты аудита. Это может быть файл, журнал событий безопасности Windows или журнал событий приложений Windows. |

Определение аудита сервера

Аудит сервера определяет, где и как регистрируются события аудита. Каждый аудит сервера определяет цель (например, файл или журнал событий Windows), интервал времени до записи событий и действие, которое должен предпринять SQL Server, если у цели заканчивается место на диске.

Вы можете создавать аудиты с помощью инструкции CREATE SERVER AUDIT или с помощью графического интерфейса SQL Server Management Studio (SSMS). Есть несколько параметров, которые вы можете настроить, в том числе:

* Имя. Удобное имя для ссылки на аудит.
* Задержка в очереди. Время в миллисекундах, в течение которого SQL Server может буферизировать результаты аудита, прежде чем отправлять их целевому объекту.
* В случае неудачи. Действия, которые необходимо предпринять, если журнал аудита недоступен — продолжить, завершить работу сервера или завершить операции, подлежащие аудиту.
* Место назначения аудита. Местоположение цели.
* Максимальный размер файла. Максимальный размер каждого файла аудита (в МБ).
* Зарезервируйте место на диске. Указывает, следует ли заранее зарезервировать место на диске для файлов аудита.

Цели аудита

Аудиты могут быть направлены на одну из следующих трех целей:

* Двоичный файл. Вывод файлов обеспечивает высочайшую производительность и является самым простым вариантом настройки.
* Журнал событий приложений Windows. Избегайте отправки слишком большого количества сведений в этот журнал, так как сетевым администраторам, как правило, не нравятся приложения, которые записывают слишком много содержимого в любой из журналов событий. Не используйте этот целевой объект для конфиденциальных данных, поскольку любой пользователь, прошедший проверку подлинности, может просматривать журнал.
* Журнал событий безопасности Windows. Это наиболее безопасный вариант аудита данных, но перед его использованием необходимо добавить учетную запись службы SQL Server в политику Создания проверок безопасности.

Следует периодически просматривать содержимое целевого объекта, который вы используете, и архивировать его содержимое.

В следующем примере кода создается и включается аудит сервера, который использует двоичный файл в качестве целевого объекта:

CREATE SERVER AUDIT [Audit]

TO SECURITY\_LOG WITH (QUEUE\_DELAY = 1000, ON\_FAILURE = CONTINUE)

WITH (STATE = ON);

GO

Действия по аудиту и Группы действий

После определения аудита можно указать события, которые будут отслеживаться в ходе аудита; эти события могут быть на уровне сервера, базы данных или аудита.

Можно добавлять события, которые будут отслеживаться аудитом на уровне отдельных событий, называемых действиями, такими как инструкция SELECT в определенной таблице, или в предопределенных группах действий, которые собирают связанные действия вместе, например SUCCESSFUL\_LOGIN\_GROUP, которая включает все действия, связанные с успешным входом в систему. Действия и группы действий связаны с аудитом посредством спецификации аудита.

Предопределенные группы действий доступны на уровне сервера, уровне базы данных и уровне аудита. Действия доступны только на уровне базы данных.

Создание спецификаций аудита сервера

После создания аудита можно добавить спецификации аудита сервера, чтобы включить в аудит одну или несколько групп действий. Когда вы создаете спецификацию аудита сервера, она автоматически отключается, поэтому вы должны не забыть включить ее, когда будете готовы к ее запуску.

Спецификация аудита сервера детализирует действия для аудита. Действия на уровне сервера сгруппированы в предопределенные группы действий, включая:

* BACKUP\_RESTORE\_GROUP. Включает все действия по резервному копированию и восстановлению.
* DATABASE\_CHANGE\_GROUP. Включает действия, которые создают, изменяют или удаляют базу данных.
* FAILED\_LOGIN\_GROUP. Включает сведения о неудачных попытках входа в систему. SUCCESSFUL\_LOGIN\_GROUP. Включает сведения об успешных входах в систему. SERVER\_OPERATION\_GROUP. Включает действия, которые изменяют настройки сервера.

В следующем примере создается и включается спецификация аудита сервера для отслеживания неудачных и успешных попыток входа в систему. Этот код предполагает, что Аудит безопасности аудита сервера уже создан:

CREATE SERVER AUDIT SPECIFICATION [AuditLogins]

FOR SERVER AUDIT SecurityAudit

ADD (FAILED\_LOGIN\_GROUP),

ADD (SUCCESSFUL\_LOGIN\_GROUP)

ALTER SERVER AUDIT SPECIFICATION AuditLogins

WITH (STATE = ON);

GO

Создание спецификации аудита базы данных аналогично созданию спецификации аудита сервера. Спецификация аудита базы данных детализирует действия или группы действий для аудита на уровне базы данных.

Вы можете выполнять аудит отдельных действий на уровне базы данных, включая: SELECT, INSERT, DELETE, UPDATE, EXECUTE

Кроме того, SQL Server определяет группы действий на уровне базы данных, включая:

* APPLICATION\_ROLE\_CHANGE\_PASSWORD\_GROUP. Возникает при изменении пароля роли приложения.
* DATABASE\_OBJECT\_ACCESS\_GROUP. Включает в себя весь доступ к объектам базы данных.
* DATABASE\_PRINCIPAL\_CHANGE\_GROUP. Включает все события, когда участник уровня базы данных создается, изменяется или удаляется.
* SCHEMA\_OBJECT\_ACCESS\_GROUP. Включает весь доступ к объектам в указанной схеме.
* SCHEMA\_OBJECT\_PERMISSION\_CHANGE\_GROUP. Включает все изменения разрешений на объекты.

Связанные с аудитом Динамические представления управления и Системные представления

SQL Server предоставляет ряд динамических представлений управления (DMVs) и системных представлений, которые могут помочь получить сведения о конфигурации аудита SQL Server.

Аудит DMVs

Следующие DMVs возвращают метаданные об аудитах:

* sys.dm\_audit\_акции. Возвращает полный список доступных действий аудита и групп действий аудита.
* sys.dm\_audit\_класс\_тип\_карты. Возвращает список справочных данных, сопоставляющий коды классов аудита описаниям.
* sys.dm\_server\_аудит\_статус. Возвращает список всех аудитов, определенных на экземпляре SQL Server, и их текущее состояние.

Представления системы Аудита

Следующие системные представления возвращают метаданные об аудитах:

* sys. server\_audit. Возвращает список всех аудитов, определенных на экземпляре SQL Server.
* sys.server\_file\_audits. Возвращает список всех аудитов, которые записывают данные в целевой файл.
* sys.server\_audit\_specifications. Возвращает список высокоуровневой информации о спецификациях аудита сервера на экземпляре.
* sys.server\_audit\_specifications\_details. Возвращает список групп действий, связанных с каждой спецификацией аудита сервера. Это представление можно объединить с sys.dm\_audit\_actions для разрешения имен групп действий.
* sys.database\_audit\_specifications. Возвращает список высокоуровневой информации о спецификациях аудита базы данных на экземпляре.
* sys.audit\_database\_specifications\_details. Возвращает список действий и групп действий, связанных с каждой спецификацией аудита базы данных. Это представление можно объединить с sys.dm\_audit\_actions для разрешения имен действий и групп действий.

**Управление аудитом SQL Server**

После настройки аудита SQL Server вам необходимо знать, как работать с данными аудита. В этом уроке рассматриваются различные способы доступа к данным аудита и способы работы с форматом записи аудита. В нем также рассматриваются некоторые потенциальные проблемы, с которыми вы можете столкнуться при работе с серверами и базами данных, на которых включен аудит SQL Server.

Цели:

* Получить данные аудита.
* Работа со структурой записей аудита.
* Объяснит потенциальные проблемы, с которыми вы можете столкнуться при использовании аудита SQL Server.

Получение Данных Аудита

Метод, который вы используете для получения данных аудита, будет зависеть от цели, указанной в определении аудита.

Цели журнала событий Windows

Если целью аудита является журнал приложений Windows или Журнал безопасности Windows, вы можете использовать средство, которое может считывать журналы событий Windows, например средство просмотра событий, для доступа к данным аудита.

Цели Двоичных Файлов

Файлы аудита, созданные с помощью аудита SQL Server, можно открыть с помощью системной табличной функции sys.fn\_get\_audit\_file, которая может считывать содержимое одного или нескольких файлов журнала аудита и возвращать содержимое в виде таблицы, которую можно обрабатывать и фильтровать с помощью стандартных инструкций Transact-SQL.

Функция sys.fn\_get\_audit\_file принимает три параметра — шаблон файла, начальное имя файла и смещение записи аудита. Шаблон файла может быть в одном из трех форматов:

<path>\\*, который собирает файлы аудита в указанном расположении. Символ звездочки является подстановочным знаком.

<path>\<audit name>\_ {GUID}, который собирает все файлы аудита, имеющие указанное имя и пару идентификаторов GUID.

<path>\<file name>, который собирает определенный файл аудита.

В этом примере показано, как использовать файл sys.fn\_get\_audit\_file для чтения всех файлов аудита в определенном каталоге: файл

sys.fn\_get\_audit\_file — основное использование

SELECT \* FROM sys.fn\_get\_audit\_file ('X:\AuditFiles\\*',default,default);