SQL Anywhere 10 でデータベースへの接続試行をモニタする方法

このマニュアルでは、接続試行に関する追加情報をトランザクション・ログに記録するために、監査を有効にする手順について説明します。

概要

監査を有効にすると、データベースについて実行されたアクティビティを追跡できます。権限のあるユーザのみがデータベースにアクセスできます。しかし、権限のないユーザもデータベースに接続しようとするかもしれません。監査を有効にすれば、データベースへの接続の成功および不成功に関するトランザクション・ログ内の情報を取得できます。

このマニュアルは、Microsoft Windows XP 上で稼働している SQL Anywhere 10.0.0.2788 を使用して記述されています。ただし、記載されている手順は、Linux や Solaris など、それ以外のシステムで稼働している SQL Anywhere 10 にも使用できます。このマニュアルに記載されている手順は、バージョン 9.0.0 以降の SQL Anywhere に適用できます。

監査の有効化

監査を有効にするには、Interactive SQL から監査 (auditing) オプションを設定します。

監査の有効化

1. Interactive SQL を起動し、データベースに接続します。
2. 以下のコマンドを実行します。

   ```
   SET option public.auditing = 'On';
   ```

3. 監査タイプを指定します。

監査を有効にしたら、取得したい監査情報のタイプを指定します。ストアド・プロシージャの sa_enable_auditing_type を使用します。このプロシージャは、取得する情報のタイプを示す文字列パラメータを 1 つ使用します。文字列は、以下の表に示されているパラメータをカンマで区切って構成します。sa_enable_auditing_type のデフォルト設定は all です。all は、成功および不成功の接続試行、DDL 文、パブリック・オプション、パーミッション・チェック、およびトリガの監査情報を取得します。

引数	説明
all	すべてのタイプの監査
connect	成功および不成功のすべての接続試行の監査
connectFailed	失敗したすべての接続試行の監査

DDL	DDL 文の監査
options	パブリック・オプションの設定の監査
permission	パーミッション／権限／ユーザ・チェックの監査
permissionDenied	失敗したパーミッション／権限／ユーザ・チェックの監査
triggers	トリガの監査

データベースへの接続試行のみを監査するよう指定するには、connect パラメータを指定してこのプロシージャを呼び出します。たとえば、Interactive SQL で以下のコマンドを実行します。

CALL sa_enable_auditing_type('connect');

監査情報の取り出し

dbtran ユーティリティを使用して、取得した監査情報が保存されているファイルを生成できます。データベースが稼働中であるかどうかにかかわらず、データベースに対して dbtran を使用できます。

注: 以下のプロシージャで dbtran コマンドを実行すると、"WARNING: Chronological ordered output must not be applied to a database (警告:日付順の出力をデータベースに適用しないでください)" という警告が表示されます。この警告は、データをリストアする場合、この変換されたログ・ファイルをデータベースに適用しないよう警告しています。

稼働中でないデータベースのトランザクション・ログの変換

1. dbtran ユーティリティに -g オプションを指定して使用します。
   -g オプションは、監査情報をトランザクション・ログに追加します。-g オプションを使用する場合、変換したいログ・ファイルを指定する必要があります。たとえば、コマンド・プロンプトで以下のコマンドを実行します。

   dbtran -g demo.log

   -g オプションは、-a、-d、および -t が使用されることを意味します。詳細については、製品マニュアルで dbtran ユーティリティの構文を参照してください。
   dbtran が完了すると、データベースの監査情報が保存されている *demo.sql* ファイルが作成されます。

稼働中のデータベースのトランザクション・ログの変換

1. dbtran コマンドとともに、監査情報をトランザクション・ログに追加する -g オプションを使用します。さらに、-c オプションで接続情報、-n オプションで出力 .sql ファイルを指定します。たとえば、コマンド・プロンプトで以下のコマンドを実行します。

```
dbtran -g -c "UID=DBA;PWD=sql;DBN=demo" -n auditing_output.sql
```

このコマンドにより、データベースの監査情報が保存されている *auditing_output.sql* というファイルが作成されます。

監査情報について

.sql ファイルを開き、データベースへの失敗した接続試行と成功した接続試行を探します。両タイプの接続試行について、どこから接続されているかがわかります。以下の例では、ローカル・マシンから接続されています。ただし、TCP/IP を介する接続試行があった場合、その接続試行と接続元のアドレスが表示されます。これは、組織外の人物からの接続試行をモニタする場合に非常に便利です。

失敗した接続試行

失敗した接続試行がある場合、変換された .sql ファイル内に以下のような情報が表示されます。

```
--CONNECT-1007-0000402747-failure-2007-02-01 14:28
----AUDIT-1007-0000402762 -- 2007/02/01 14:28:35.188 Connection attempt (machine (local)) Port SharedMemory - Failed
----ROLLBACK-1007-0000402795
```

トランザクション・ログのオフセット 402747 に、2007 年 2 月 1 日午後 2:28 に発生した接続試行が記録されています。オフセット 402762 で、この監査情報が再び 2007 年 2 月 1 日午後 2:28 に記録されます。失敗した接続試行は、ローカル・マシンの共有メモリからのものでした。トランザクション・ログのオフセット 402795 で、ロールバックが行われています。この失敗した接続試行については、ユーザ名が表示されません。1007 はデータベースへの特定の接続試行です。

成功した接続試行

成功した接続試行がある場合、変換された .sql ファイル内に以下のような情報が表示されます。

```
--CONNECT-1009-0000402857-DBA-2007-02-01 14:28
----AUDIT-1009-0000402868 -- 2007/02/01 14:28:44.016 Checking DBA authority - OK
----AUDIT-1009-0000402892 -- 2007/02/01 14:28:44.016 Connection attempt (DBA - machine (local)) Port SharedMemory - OK
```

トランザクション・ログのオフセット 402857 に、2007 年 2 月 1 日午後 2:28 に発生した DBA による接続試行があります。オフセット 402868 で記録されている監査情報は、DBA がデータベースへの接続権限を持っていることを示しています。402892 で、ローカル・マシンの共有メモリからの接続が成功していることを示す監査情報がさらに記録されます。ここでも、1009 はデータベースへの特定の接続試行です。

auditing_options データベース・オプション

auditing_options オプションを設定すると、監査に影響が及ぶ可能性があります。このオプションの値は変更しないでください。これは、内部使用専用のオプションです。

監査コメントの追加

DBA 権限を持つユーザは、sa_audit_string ストアド・プロシージャに最大 200 バイトまでの文字列を指定して、トランザクション・ログに監査コメントを追加できます。sa_audit_string ストアド・プロシージャは、監査が有効な場合にのみ使用できます。たとえば、無効なログイン情報をテストしようとしていることを示すコメントをトランザクション・ログに追加するには、以下のコマンドを実行します。

```
CALL sa_audit_string( 'Testing invalid username and password.' );
```

無効な接続情報のテストを行う前に、このコメントがトランザクション・ログに記録されます。

監査の代替手段

監査の代わりに要求ログを使用することもできます。要求ログは、アプリケーションとやり取りするメッセージを記録してパフォーマンスのモニタや問題解決に使用しますが、監査に役立つ情報も記録します。ただし、要求ログは、監査の代わりに実行されるわけではありません。要求ログを有効にするには、RequestLogging パラメータを使用して sa_server_option ストアド・プロシージャを呼び出してから、RequestLogFile パラメータを使用して再びこのプロシージャを呼び出します。要求ログを使用する場合、すべての情報を取得するように指定します。タイムスタンプ、接続 ID、要求タイプなどの情報が記録されます。RequestLogFile パラメータを使用するには、情報を記録する出力ファイルを指定する必要があります。

要求ログの有効化

1. Interactive SQL を開いてデータベースに接続し、以下の文を実行します。

   ```
   CALL sa_server_option( 'RequestLogging','all' );
   CALL sa_server_option( 'RequestLogFile','c:¥request_output.sql' );
   ```

   上記の例は、すべての要求ログ情報を取得し、*auditing_output.sql* というファイルに保存します。

要求ログの有効化とログ・ファイルの指定は、サーバの -zr オプション (RequestLogging オプションに相当) と -zo オプション (RequestLogFile オプションに相当) でも行えます。-zr オプションの後に all 値を指定し、-zo オプションの後に出力ファイル名を指定します。たとえば、以下のように指定します。

```
dbeng10 -zr all -zo c:¥request_output.sql
```

要求ログ出力ファイルのサイズ指定

サーバ・オプションの -zs は、要求ログ出力ファイルの最大サイズをバイト単位で指定します。この値は、sa_server_option ストアド・プロシージャに RequestLogMaxSize を指定する方法でも設定できます。サイズの後に k、m、g を付けることにより、キロバイト、メガバイト、ギガバイトを指定します。たとえば、以下のコマンドは、要求ログを有効にしてデータベース・サーバを起動し、最大サイズが 10 キロバイトの *request_output.sql* ファイルにログ情報を記録します。

```
dbeng10 -zr all -zo c:¥request_output.sql -zs 10k
```

ログ出力ファイルが最大サイズに達すると、ファイル名に *extension.old* が付いたファイルが既存のファイルと置き換えられて、新しい出力ファイルが作成されます。

要求ログ出力ファイル数の指定

-zn オプションを使用して、保持する古い監査出力ファイルの数を指定できます。ただし、このオプションは -zs オプションが指定されている場合にのみ有効になります。sa_server_option ストアド・プロシージャに RequestLogNumFiles オプションと数値を渡す方法でもこのファイル数を指定できます。以下のコマンドは、要求ログを有効にしてサーバを起動し、最大サイズが 100 キロバイトで、古い監査出力ファイルが 3 つ保持される request_output.sql ファイルに情報を記録します。

```
dbeng10 -zr all -zo c:¥request_output.sql -zs 100k -zn 3
```

パフォーマンスの考慮

監査または要求ログを有効にすると、通常よりも大きい情報量が取得されてトランザクション・ログに記録されます。そのため、通常よりもはるかに大きいディスク・スペースが必要になり、パフォーマンスに影響が及びます。これらのオプションを有効化する期間を決める際には、このことを考慮する必要があります。監査および要求ログは、長期間有効にしたままにしないことをおすすめします。