70-762 Q&As

Developing SQL Databases

Pass Microsoft 70-762 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.passapply.com/70-762.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Microsoft Official Exam Center

QUESTION 1

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution. Determine whether the solution meets the stated goals.

You need to create a stored procedure that updates the Customer, CustomerInfo, OrderHeader, and OrderDetails tables in order.

You need to ensure that the stored procedure:

Runs within a single transaction.

Commits updates to the Customer and CustomerInfo tables regardless of the status of updates to the OrderHeader and OrderDetail tables.

Commits changes to all four tables when updates to all four tables are successful.

Solution: You create a stored procedure that includes the following Transact-SQL segment:

```
DECLARE @CustomerComplete bit = 0
BEGIN TRY
    BEGIN TRAN
        UPDATE Customer ...
        UPDATE CustomerInfo ...

        SET @CustomerComplete = 1

        SAVE TRAN TR1

        UPDATE OrderHeader ...
        UPDATE OrderDetail ...
    COMMIT TRAN
END TRAN
BEGIN CATCH
    IF (@CustomerComplete = 1) AND (XACT_STATE() = 1)
    BEGIN
        COMMIT TRAN
    END
    ELSE IF XACT_STATE() = 1
        ROLLBACK TRAN
END CATCH
```


Does the solution meet the goal?

A. Yes

B. No

Correct Answer: A

Need to handle the case where the first two updates (OrderHeader, OrderDetail) are successful, but either the 3rd or the 4th (OrderHeader, OrderDetail) fail. We add the @CustomerComplete variable in the BEGIN TRY block, and test it in

the BEGIN CATCH block.

Note: XACT_STATE indicates whether the request has an active user transaction, and whether the transaction is capable of being committed.

XACT_STATE =1: the current request has an active user transaction. The request can perform any actions, including writing data and committing the transaction.

References:

https://docs.microsoft.com/en-us/sql/t-sql/functions/xact-state-transact-sql

QUESTION 2

Your company runs end-of-the-month accounting reports. While the reports run, other financial records are updated in the database.

Users report that the reports take longer than expected to run.

You need to reduce the amount of time it takes for the reports to run. The reports must show committed data only.

What should you do?

A. Use the NOLOCK option.

B. Execute the DBCC UPDATEUSAGE statement.

C. Use the max worker threads option.

D. Use a table-valued parameter.

E. Set SET ALLOW_SNAPSHOT_ISOLATION to ON.

F. Set SET XACT_ABORT to ON.

G. Execute the ALTER TABLE T1 SET (LOCK_ESCALATION = AUTO); statement.

H. Use the OUTPUT parameters.

Correct Answer: E

Snapshot isolation enhances concurrency for OLTP applications.

Once snapshot isolation is enabled, updated row versions for each transaction are maintained in tempdb. A unique transaction sequence number identifies each transaction, and these unique numbers are recorded for each row version. The transaction works with the most recent row versions having a sequence number before the sequence number of the transaction. Newer row versions created after the transaction has begun are ignored by the transaction. References: https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/sql/snapshot-isolation-in-sql-server

QUESTION 3

Note: This question is part of a series of questions that use the same scenario. For your convenience, the scenario is repeated in each question. Each question presents a different goal and answer choices, but the text of the scenario is exactly the same in each question in this series. You have a database named DB1 that contains the following tables: Customer, CustomerToAccountBridge, and CustomerDetails. The three tables are part of the Sales schema. The database also contains a schema named Website. You

```
CREATE TABLE Customer
(
    CustomerNumber int NOT NULL,
    CustomerName varchar(50) NOT NULL,
    CreateDate date NOT NULL,
    Gender bit,
    Address varchar(50)
    City varchar(50)
    State char(2),
    CustomerStatus bit NOT NULL,
    MaritalStatus bit,
    Segment varchar(5),
    CountryCode char(2),
    Birthday date,
    PostalCode char(5),
    PhoneNumber varchar(20),
    Account1 char(7),
    Account1Status bit,
    Account2 char(7),
    Account2Status bit,
    CONSTRAINT PK_Customer PRIMARY KEY CLUSTERED (CustomerNumber)
);
```


create the Customer table by running the following Transact-SQL statement:

The value of the CustomerStatus column is equal to one for active customers. The value of the Account1Status and Account2Status columns are equal to one for active accounts. The following table displays selected columns and rows from the Customer table.

Customer ID	CustomerName	Gender	Account1	Account1Status	Account2	Account2Status
101	Name A	0	0001001	0	0001002	1
102	Name B	1	0002001	1	0002002	0
103	Name C	0	0003001	1	0003002	1

You plan to create a view named Website.Customer and a view named Sales.FemaleCustomers. Website.Customer must meet the following requirements:

1. Allow users access to the CustomerName and CustomerNumber columns for active customers.

2. Allow changes to the columns that the view references. Modified data must be visible through the view.

3. Prevent the view from being published as part of Microsoft SQL Server replication.

Sales.Female.Customers must meet the following requirements:

1. Allow users access to the CustomerName, Address, City, State and PostalCode columns.

2. Prevent changes to the columns that the view references.

3. Only allow updates through the views that adhere to the view filter.

You have the following stored procedures: spDeleteCustAcctRelationship and spUpdateCustomerSummary. The spUpdateCustomerSummary stored procedure was created by running the following Transacr-SQL statement:

```
CREATE PROCEDURE uspUpdateCustomerSummary
@CustomerId INT
AS
BEGIN
    SET NOCOUNT on;
    UPDATE CustomerDetails SET TotalDepositAccountCOunt - TotalDepositAccountCount + 1 WHERE CustomerID = @CustomerID;
    BEGIN TRAN;
        BEGIN TRY
            UPDATE CustomerDetails SET TotalAccountCount = TotalAccountCount + 1 WHERE CustomerID = @CustomerID;
        END TRY
        BEGIN CATCH
            IF @@TRANCOUNT > 0
                ROLLBACK TRAN;
        END CATCH
        IF @@TRANCOUNT > 0
            COMMIT TRAN;
```


You run the spUpdateCustomerSummary stored procedure to make changes to customer account summaries. Other stored procedures call the spDeleteCustAcctRelationship to delete records from the CustomerToAccountBridge table.

When a procedure calls spDeleteCustAcctRelationship, if the calling stored procedures has already started an active transaction, all the detections made by the spDeleteCustAccRelationship stored procedure must be committed by the caller;

otherwise changes must be committed within the spDeleteCustAcctRelationship stored procedure.

If any error occurs during the delete operation, only the deletes made by the soDeleteCustACCTRelationships stored procedure must be rolled back and the status must be updated.

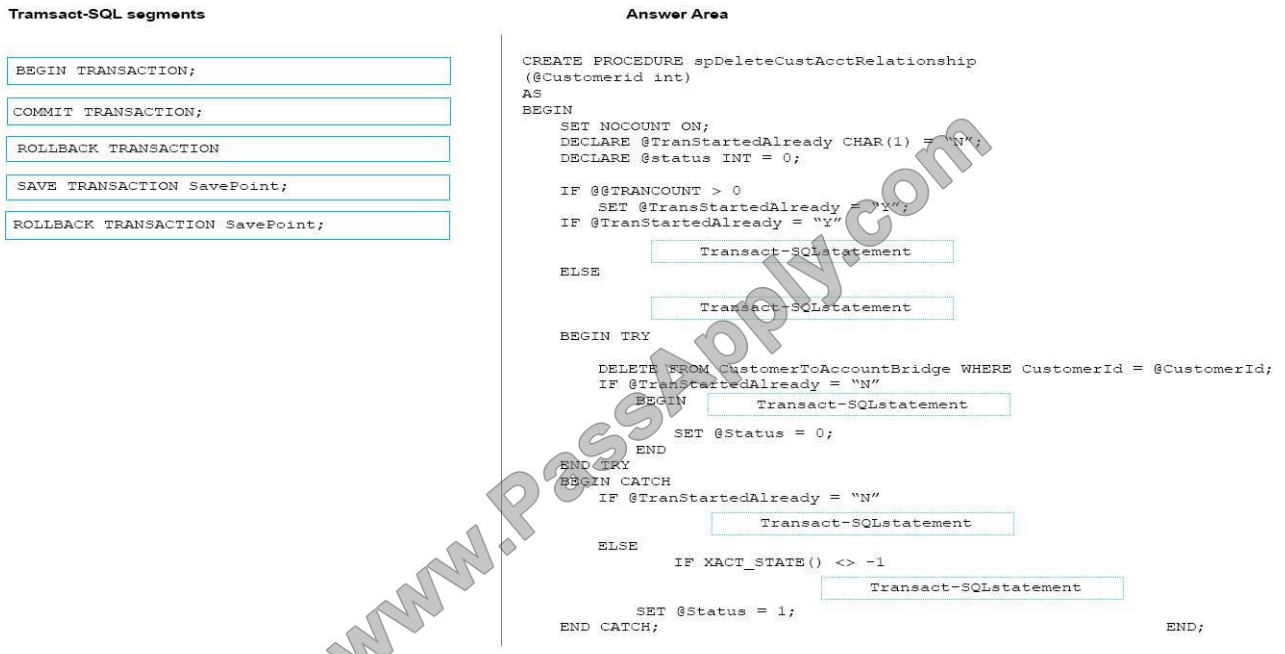
You need to complete the stored procedure to ensure all the requirements are met.

How should you complete the procedure? To answer, drag the Transact-SQL segments to the correct location. Each transact-SQL segment may be used once, more than once or not at all. You may need to drag the split bar between panes

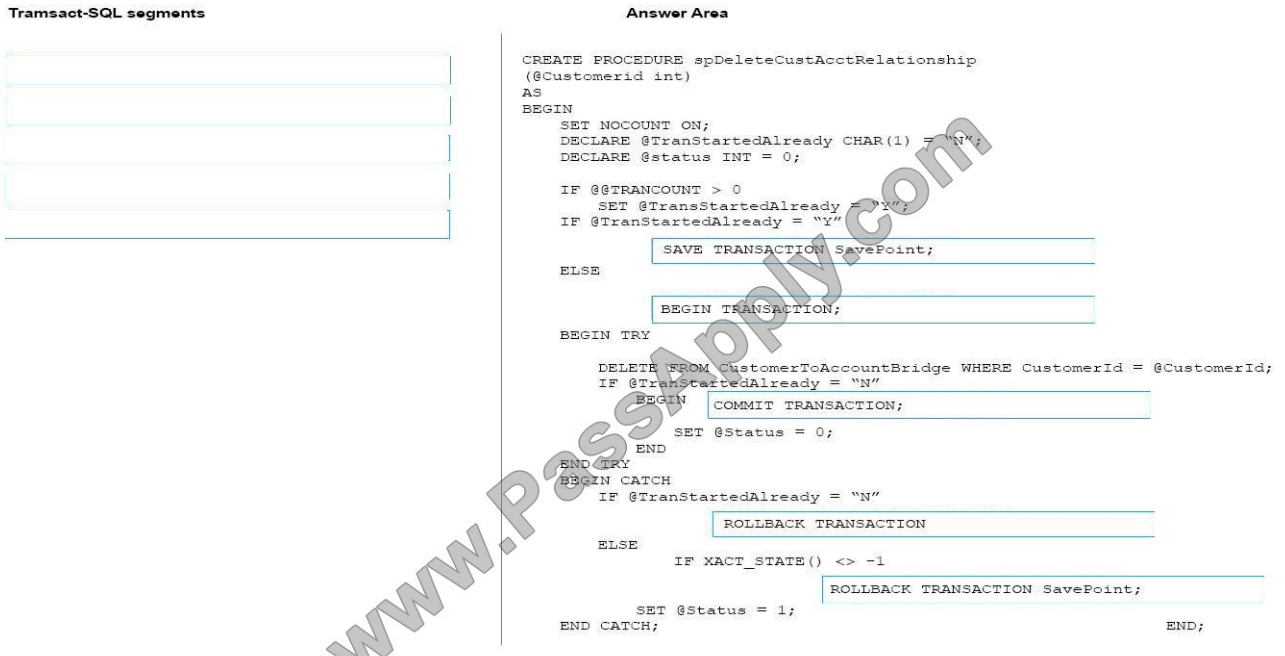
or scroll to view content.

NOTE: Each correct selection is worth one point.

Select and Place:



Correct Answer:



Savepoints offer a mechanism to roll back portions of transactions. You create a savepoint using the SAVE TRANSACTION savepoint_name statement. Later, you execute a ROLLBACK TRANSACTION savepoint_name

statement to roll back to the savepoint instead of rolling back to the start of the transaction.

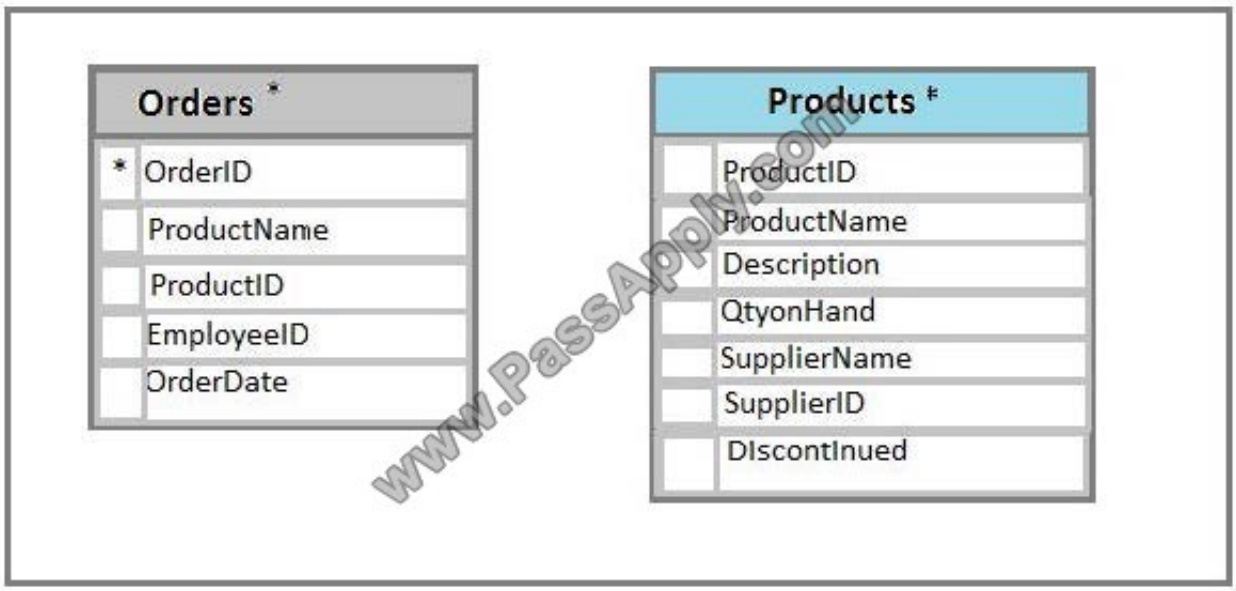
References: https://technet.microsoft.com/en-us/library/ms178157(v=sql.105).aspx

QUESTION 4

Note: This question is part of a series of questions that use the same scenario. For your convenience, the scenario is repeated in each question. Each question presents a different goal and answer choices, but the text of the scenario is

exactly the same in each question in this series.

You have a database named Sales that contains the following database tables: Customer, Order, and Products. The Products table and the Order table are shown in the following diagram.



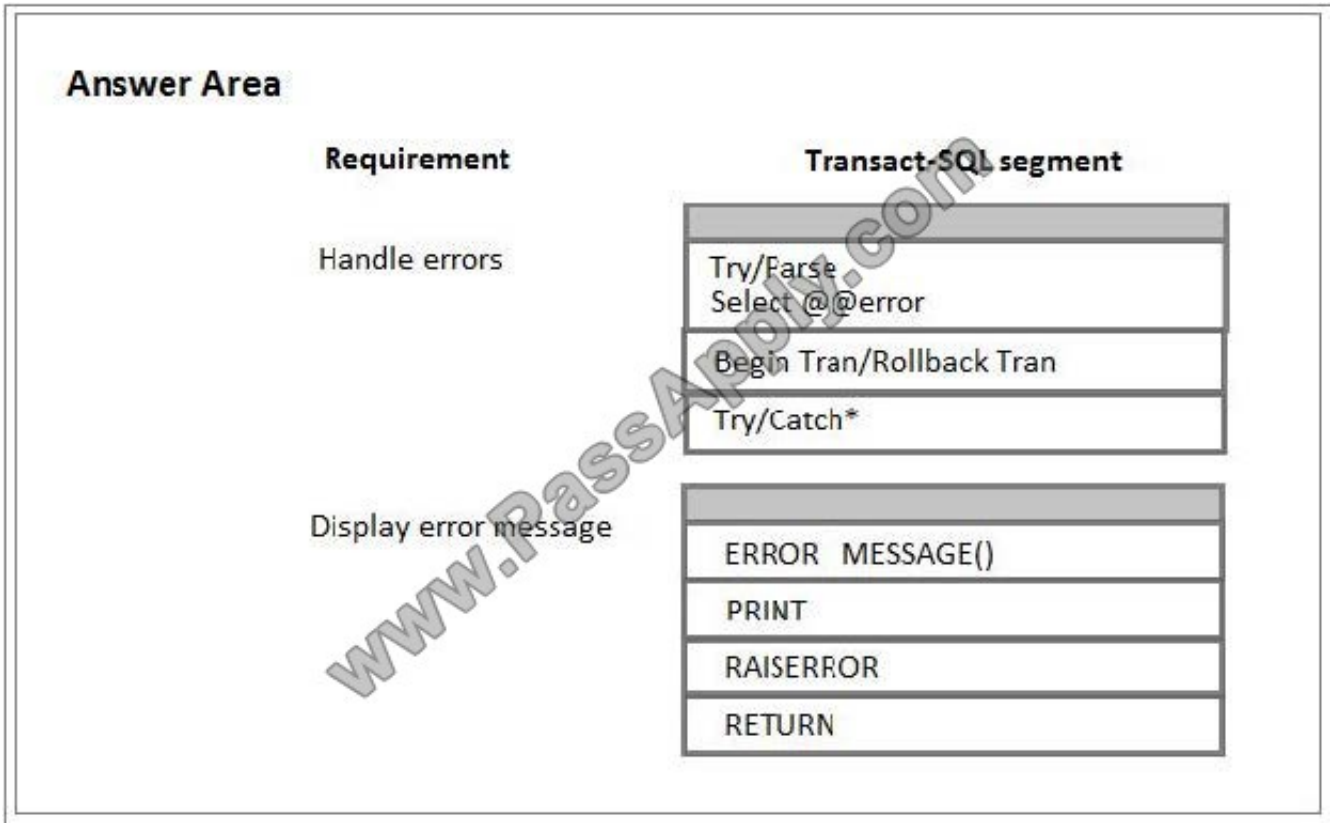
The customer table includes a column that stores the data for the last order that the customer placed.

You plan to create a table named Leads. The Leads table is expected to contain approximately 20,000 records. Storage requirements for the Leads table must be minimized.

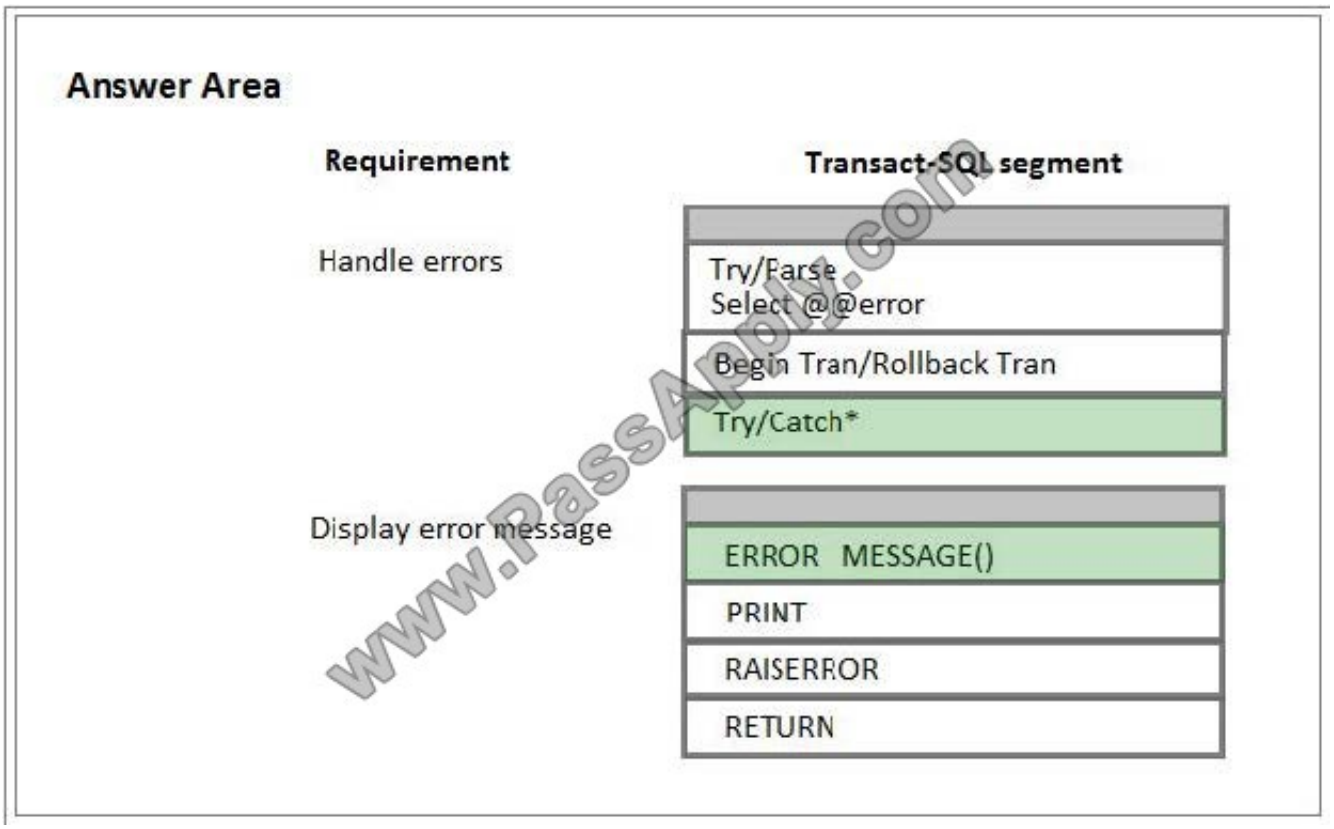
You need to implement a stored procedure that deletes a discontinued product from the Products table. You identify the following requirements:

What should you do? To answer, select the appropriate Transact-SQL segments in the answer area.

Hot Area:



Correct Answer:



Using TRY...CATCH inTransact-SQL

Errors in Transact-SQL code can be processed by using a TRY…CATCH construct.

TRY…CATCH can use the following error function to capture error information:

ERROR_MESSAGE() returns the complete text of the error message. The text includes the values supplied for any substitutable parameters such as lengths, object names, or times.

References:

https://technet.microsoft.com/en-us/library/ms179296(v=sql.105).aspx

QUESTION 5

Note: this question is part of a series of questions that use the same or similar answer choices. An answer choice may be correct for more than one question in the series. Each question is independent of the other questions in the series.

Information and details provided in a question apply only to that question.

You are developing an application to track customer sales.

You need to create a database object that meets the following requirements:

Launch when table data is modified.

Evaluate the state a table before and after a data modification and take action based on the difference.

Prevent malicious or incorrect table data operations.

Prevent changes that violate referential integrity by cancelling the attempted data modification.

Run managed code packaged in an assembly that is created in the Microsoft.NET Framework and located into Microsoft SQL Server.

What should you create?

A. extended procedure

B. CLR procedure

C. user-defined procedure

D. DML trigger

E. scalar-valued function

F. table-valued function

Correct Answer: B

You can create a database object inside SQL Server that is programmed in an assembly created in the Microsoft .NET Framework common language runtime (CLR). Database objects that can leverage the rich programmingmodel provided

by the CLR include DML triggers, DDL triggers, stored procedures, functions, aggregate functions, and types.

Creating a CLR trigger (DML or DDL) in SQL Server involves the following steps:

Define the trigger as a class in a .NETFramework-supported language. For more information about how to program triggers in the CLR, see CLR Triggers. Then, compile the class to build an assembly in the .NET Framework using the

appropriate language compiler.

Register the assembly in SQL Server using the CREATE ASSEMBLY statement. For more information about assemblies in SQL Server, see Assemblies (Database Engine).

Create the trigger that references the registered assembly.

References:https://msdn.microsoft.com/en-us/library/ms179562.aspx