Vendor: Microsoft

Exam Code: 70-433

Exam Name: TS: Microsoft SQL Server 2008, Database Development

Version: Demo

QUESTION 1
You have a user named John. He has SELECT access to the Sales schema. You need to eliminate John's SELECT access rights from the Sales.SalesOrder table without affecting his other permissions. Which Transact-SQL statement should you use?

A. DROP USER John;
B. DENY SELECT ON Sales.SalesOrder TO John;
C. GRANT DELETE ON Sales.SalesOrder TO John;
D. REVOKE SELECT ON Sales.SalesOrder FROM John;

Correct Answer: B
Explanation:
REVOKE - permanently removes both granted an denied permissions on an object, resulting in no permissions.Main thing you have to remember is, this does not restrict user accessing the object completely. If user is in a role that has permission on the object for the operation, user will be able to perform the operation.

DENY - Denies permission to the object for an operation. Once it set, it takes precedence over all other GRANT permissions, user will not be able to perform the operation against the object.

To sum up, because John does not have GRANT permissions on the Sales.SalesOrder table (instead it has GRANT permission on Sales schema), then REVOKE SELECT ON Sales.SalesOrder from John will not remove any permissions.

Here is a code that shows it clearly. - create a login and user

```
CREATE LOGIN [John] WITH PASSWORD = '1', CHECK_POLICY = OFF GO
USE [AdventureWorks2008]
GO
CREATE USER [John] FOR LOGIN [John]
WITH DEFAULT_SCHEMA = [dbo]
GO - grant permission on Sales schema
GRANT SELECT ON SCHEMA :: Sales TO [John] - Run SELECT with John's credentials and see
 - He sees records
EXECUTE AS USER = 'John'
SELECT * FROM Sales.SalesOrderHeader
REVERT - Revoke permisson for the table from him
REVOKE SELECT ON Sales.SalesOrderHeader FROM [John] -- He still sees data
EXECUTE AS USER = 'John'
SELECT * FROM Sales.SalesOrderHeader
REVERT - This explicitly denies permission on SalesOrderHeader to John -- Once this is executed, he will not be able to see data -- even we grant him again.
DENY SELECT ON Sales.SalesOrderHeader TO [John] - He sees error message: The SELECT permission was denied on the object 'SalesOrderHeader', database 'AdventureWorks2008', schema 'Sales'.
EXECUTE AS USER = 'John'
SELECT * FROM Sales.SalesOrderHeader
REVERT
```

QUESTION 2
You need to create a column that allows you to create a unique constraint. Which two column definitions should you choose? (Each correct answer presents a complete solution. Choose two.)

A. nvarchar(100) NULL
B. nvarchar(max) NOT NULL
C. nvarchar(100) NOT NULL
D. nvarchar(100) SPARSE NULL

Correct Answer: AC

QUESTION 3
You manage a SQL Server 2008 database that is located at your company's corporate headquarters. The database contains a table named dbo.Sales. You need to create different views of the dbo.Sales table that will be used by each region to insert, update, and delete rows. Each regional office must only be able to insert, update, and delete rows for their respective region. Which view should you create for Region1?

A. CREATE VIEW dbo.Region1Sales
AS
SELECT SalesID,OrderQty,SalespersonID,RegionID FROM dbo.Sales WHERE RegionID = 1;
B. CREATE VIEW dbo.Region1Sales
AS
SELECT SalesID,OrderQty,SalespersonID,RegionID FROM dbo.Sales WHERE RegionID = 1
WITH CHECK OPTION;
C. CREATE VIEW dbo.Region1Sales
WITH SCHEMABINDING
AS SELECT SalesID,OrderQty,SalespersonID,RegionID FROM dbo.Sales WHERE RegionID = 1;
D. CREATE VIEW dbo.Region1Sales
WITH VIEW_METADATA
AS
SELECT SalesID,OrderQty,SalespersonID,RegionID FROM dbo.Sales WHERE RegionID = 1;

Correct Answer: B
Explanation:
CHECK OPTION
Forces all data modification statements executed against the view to follow the criteria set within select_statement. When a row is modified through a view, the WITH CHECK OPTION makes sure the data remains visible through the view after the modification is committed.

QUESTION 4
You administer a SQL Server 2008 database that contains a table name dbo.Sales, which contains the following table definition:

CREATE TABLE [dbo].[Sales](

[SalesID] [int] IDENTITY(1,1) NOT NULL PRIMARY KEY CLUSTERED,

[OrderDate] [datetime] NOT NULL,

[CustomerID] [int] NOT NULL,

[SalesPersonID] [int] NULL,

[CommentDate] [date] NULL);

This table contains millions of orders. You run the following query to determine when sales persons comment in the dbo.Sales table:

SELECT SalesID,CustomerID,SalesPersonID,CommentDate

FROM dbo.Sales

WHERE CommentDate IS NOT NULL

AND SalesPersonID IS NOT NULL;

You discover that this query runs slow. After examining the data, you find only 1% of rows have comment dates and the SalesPersonID is null on 10% of the rows. You need to create an index to optimize the query. The index must conserve disk space while optimizing your query.

Which index should you create?

A. CREATE NONCLUSTERED INDEX idx1
ON dbo.Sales (CustomerID)
INCLUDE (CommentDate,SalesPersonID);

B. CREATE NONCLUSTERED INDEX idx1
ON dbo.Sales (SalesPersonID)
INCLUDE (CommentDate,CustomerID);

C. CREATE NONCLUSTERED INDEX idx1
ON dbo.Sales (CustomerID)
INCLUDE(CommentDate)
WHERE SalesPersonID IS NOT NULL;

D. CREATE NONCLUSTERED INDEX idx1
ON dbo.Sales (CommentDate, SalesPersonID)
INCLUDE(CustomerID)
WHERE CommentDate IS NOT NULL;

Correct Answer: D

QUESTION 5

Your database is 5GB and contains a table named SalesHistory. Sales information is frequently inserted and updated. You discover that excessive page splitting is occurring. You need to reduce the occurrence of page splitting in the SalesHistory table. Which code segment should you use?.

A. ALTER DATABASE Sales
MODIFY FILE
(NAME = Salesdat3,
SIZE = 10GB);

B. ALTER INDEX ALL ON Sales.SalesHistory
REBUILD WITH (FILLFACTOR = 60);

C. EXEC sys.sp_configure 'fill factor (%)', '60';

D. UPDATE STATISTICS Sales.SalesHistory(Products) WITH FULLSCAN, NORECOMPUTE;

Correct Answer: B
Explanation:

Fill Factor
The fill-factor option is provided for fine-tuning index data storage and performance. When an index is created or rebuilt, the fill-factor value determines the percentage of space on each leaf-level page to be filled with data, reserving the remainder on each page as free space for future growth. For example, specifying a fill-factor value of 60 means that 40 percent of each leaf-level page will be left empty, providing space for index expansion as data is added to the underlying table.

Page Splits
A correctly chosen fill-factor value can reduce potential page splits by providing enough space for index expansion as data is added to the underlying table.When a new row is added to a full index page, the Database Engine moves approximately half the rows to a new page to make room for the new row. This reorganization is known as a page split. A page split makes room for new records, but can take time to perform and is a resource intensive operation. Also, it can cause fragmentation that causes increased I/O operations.

Fragmentation
Fragmentation breaks down to physical and logical fragmentation (or, internal and external fragmentation).
Physical/Internal fragmentation is caused when there is wasted space in the index caused by page splits, deletes, FILLFACTOR and PAD_INDEX. Logical/External fragmentation is when the pages that make up the leaf levels of the index are not in good order. This is usually caused by page splits. Both cause performance problems. Internal fragmentation causes problems because the indexes get bigger and bigger requiring more and more pages to store the data, which means that reads from the index slow things down. External fragmentation causes problems because when the data needs to be read from the index it has to skip all over the place following the links between pages, again, slowing things down.

The SQL Server Database Engine automatically maintains indexes whenever insert, update, or delete operations are made to the underlying data. Over time these modifications can cause the information in the index to become scattered in the database (fragmented). Fragmentation exists when indexes have pages in which the logical ordering, based on the key value, does not match the physical ordering inside the data file. You can remedy index fragmentation by either reorganizing an index or by rebuilding an index.
To reorganize one or more indexes, use the ALTER INDEX statement with the REORGANIZE clause.

Reorganizing an index defragments the leaf level of clustered and nonclustered indexes on tables and views by physically reordering the leaf-level pages to match the logical order (left to right) of the leaf nodes. Having the pages in order improves index-scanning performance. The index is reorganized within the existing pages allocated to it; no new pages are allocated.

Rebuilding an index drops the index and creates a new one. In doing this, fragmentation is removed, disk space is reclaimed by compacting the pages using the specified or existing fill factor setting, and the index rows are reordered in contiguous pages (allocating new pages as needed). This can improve disk performance by reducing the number of page reads required to obtain the requested data.
You can use the CREATE INDEX with the DROP_EXISTING clause or ALTER INDEX with the REBUILD clause statements to set the fill-factor value for individual indexes. EXEC sys.sp_configure 'fill factor (%)', '60'; will modify the server default fill factor value.

QUESTION 6

You have a table named dbo.Customers. The table was created by using the following Transact-SQL statement:

```
CREATE TABLE dbo.Customers

(

CustomerID int IDENTITY(1,1) PRIMARY KEY CLUSTERED,

AccountNumber nvarchar(25) NOT NULL,

FirstName nvarchar(50) NOT NULL,

LastName nvarchar(50) NOT NULL,

AddressLine1 nvarchar(255) NOT NULL,

AddressLine2 nvarchar(255) NOT NULL,

City nvarchar(50) NOT NULL,

StateProvince nvarchar(50) NOT NULL,

Country nvarchar(50) NOT NULL,

PostalCode nvarchar(50) NOT NULL,

CreateDate datetime NOT NULL DEFAULT(GETDATE()),

ModifiedDate datetime NOT NULL DEFAULT(GETDATE())

)
```

You create a stored procedure that includes the AccountNumber, Country, and StateProvince columns from the dbo.Customers table. The stored procedure accepts a parameter to filter the output on the AccountNumber column.

You need to optimize the performance of the stored procedure. You must not change the existing structure of the table.

Which Transact-SQL statement should you use?

A. CREATE STATISTICS ST_Customer_AccountNumber ON dbo.Customer (AccountNumber) WITH FULLSCAN;
B. CREATE CLUSTERED INDEX IX_Customer_AccountNumber ON dbo.Customer (AccountNumber);
C. CREATE NONCLUSTERED INDEX IX_Customer_AccountNumber ON dbo.Customer (AccountNumber) WHERE AccountNumber = '';
D. CREATE NONCLUSTERED INDEX IX_Customer_AccountNumber ON dbo.Customer (AccountNumber) INCLUDE (Country, StateProvince);

Correct Answer: D

QUESTION 7

You have a table named Customer.

You need to ensure that customer data in the table meets the following requirements:

credit limit must be zero unless customer identification has been verified.

credit limit must be less than 10,000.

Which constraint should you use?

A. CHECK (CreditLimt BETWEEN 1 AND 10000)
B. CHECK (Verified = 1 AND CreditLimt BETWEEN 1 AND 10000)
C. CHECK ((CreditLimt = 0 AND Verified = 0) OR (CreditLimt BETWEEN 1 AND 10000 AND Verified = 1))
D. CHECK ((CreditLimt = 0 AND Verified = 0) AND (CreditLimt BETWEEN 1 AND 10000 AND Verified = 1))

Correct Answer: C

QUESTION 8

You have a table named AccountsReceivable. The table has no indexes. There are 75,000 rows in the table. You have a partition function named FG_AccountData. The AccountsReceivable table is defined in the following Transact-SQL statement:

CREATE TABLE AccountsReceivable (

column_a INT NOT NULL,

column_b VARCHAR(20) NULL)

ON [PRIMARY];

You need to move the AccountsReceivable table from the PRIMARY file group to FG_AccountData.

Which Transact-SQL statement should you use?

A. CREATE CLUSTERED INDEX idx_AccountsReceivable ON AccountsReceivable(column_a) ON [FG_AccountData];
B. CREATE NONCLUSTERED INDEX idx_AccountsReceivable ON AccountsReceivable(column_a) ON [FG_AccountData];
C. CREATE CLUSTERED INDEX idx_AccountsReceivable ON AccountsReceivable(column_a) ON FG_AccountData(column_a);
D. CREATE NONCLUSTERED INDEX idx_AccountsReceivable ON AccountsReceivable(column_a) ON FG_AccountData(column_a);

Correct Answer: C

QUESTION 9
You have a SQL Server 2008 database named Contoso with a table named Invoice. The primary key of the table is InvoiceId, and it is populated by using the identity property. The Invoice table is related to the InvoiceLineItem table. You remove all constraints from the Invoice table during a data load to increase load speed. You notice that while the constraints were removed, a row with InvoiceId = 10 was removed from the database. You need to re-insert the row into the Invoice table with the same InvoiceId value. Which Transact-SQL statement should you use?

A. INSERT INTO Invoice (InvoiceId, ...VALUES (10, ...
B. SET IDENTITY_INSERT Invoice ON;
INSERT INTO Invoice (InvoiceId, ...
VALUES (10, ...
SET IDENTITY_INSERT Invoice OFF;
C. ALTER TABLE Invoice;
ALTER COLUMN InvoiceId int;
INSERT INTO Invoice (InvoiceId, ...
VALUES (10, ...
D. ALTER DATABASE Contoso SET SINGLE_USER;
INSERT INTO Invoice (InvoiceId, ...
VALUES (10, ...
ALTER DATABASE Contoso SET MULTI_USER;

Correct Answer: B

QUESTION 10
You are developing a new database. The database contains two tables named SalesOrderDetail and Product. You need to ensure that all products referenced in the SalesOrderDetail table have a corresponding record in the Product table. Which method should you use?

A. JOIN
B. DDL trigger
C. Foreign key constraint
D. Primary key constraint

Correct Answer: C

QUESTION 11
You are creating a table that stores the GPS location of customers. You need to ensure that the table allows you to identify customers within a specified sales boundary and to calculate the distance between a customer and the nearest store. Which data type should you use?

A. geometry
B. geography
C. nvarchar(max)
D. varbinary(max) FILESTREAM

Correct Answer: B
Explanation:
The GEOGRAPHY data type is used to store ellipsoidal (round-earth) data. It is used to store latitude and longitude coordinates that represent points, lines, and polygons on the earth's surface. For example, GPS data that represents the lay of the land is one example of data that can be stored in the GEOGRAPHY data type.

QUESTION 12
You plan to add a new column named SmallKey to the Sales.Product table that will be used in a unique constraint. You are required to ensure that the following information is applied when adding the new column:

'a1' and 'A1' are treated as different values

'a' and 'A' sort before 'b' and 'B' in an ORDER BY clause

You need to select the collation that meets the requirements for the new column. Which collation should you select?

A. Latin1_General_BIN
B. SQL_Latin1_General_CP1_CI_AI
C. SQL_Latin1_General_CP1_CI_AS
D. SQL_Latin1_General_CP1_CS_AS

Correct Answer: D
Explanation:
SQL Server Collation Name consists of several parts, one of them is responsible for CaseSensitivity - CI specifies case-insensitive, CS specifies case-sensitive.
BIN specifies the binary sort order to be used.
So, because we want case-sensitive location, B and C are not suitable. Latin1_General_BIN use binary sort order, but we want linguistical sort order (according to rules of language), not based on the code point values of characters.

QUESTION 13
You have multiple tables that represent properties of the same kind of entities. The property values are comprised of text, geometry, varchar(max), and user-defined types specified as 'bit NOT NULL' data types.

You plan to consolidate the data from multiple tables into a single table. The table will use semi-structured storage by taking advantage of the SPARSE option.

You are tasked to identify the data types that are compatible with the SPARSE option.

Which data type is compatible with the SPARSE option?

A. text
B. geometry
C. varchar(max)
D. A user-defined type defined as 'bit NOT NULL'

Correct Answer: C
Explanation:
Sparse columns are ordinary columns that have an optimized storage for null values. Sparse columns reduce the space requirements for null values at the cost of more overhead to retrieve nonnull values.
The following data types cannot be specified as SPARSE:

QUESTION 14
You currently store date information in two columns. One column contains the date in local time and one column contains the difference between local time and UTC time. You need to store this data in a single column. Which data type should you use?

A. time
B. datetime2
C. datetime2(5)
D. datetimeoffset

Correct Answer: D
Explanation:
datetimeoffset defines a date that is combined with a time of a day that has time zone awareness.

QUESTION 15
You have two partitioned tables named Transaction and TransactionHistory. You need to archive one of the partitions of the Transaction table to the TransactionHistory table. Which method should you use?

A. ALTER TABLE ...
SWITCH ...
B. INSERT ... SELECT ...; TRUNCATE TABLE
C. ALTER PARTITION FUNCTION ... MERGE ...
D. ALTER PARTITION FUNCTION ...
SPLIT ...

Correct Answer: B

QUESTION 16
You are creating a new table in a database. Your business requires you to store data in the table for only seven days. You need to implement a partitioned table to meet this business requirement. Which tasks should you complete?

A. Create the partition function
Create the partition scheme
Create the table
B. Create the partition function
Create the table
Create a filtered index
C. Add a secondary file to the primary filegroups Create the table
Create the distributed partitioned view
D. Create the partition function
Create the partition scheme
Create the distributed partitioned view

Correct Answer: A

QUESTION 17
You need to alter stored procedures to use the WITH RECOMPILE option. Which types of stored procedures should you alter? (Each correct answer represents a complete solution. Choose two.)

A. Stored procedures implemented from CLR assemblies.
B. Stored procedures that require the FOR REPLICATION option.
C. Stored procedures that require the WITH ENCRYPTION option.
D. Stored procedures that contain queries that use the OPTION (RECOMPILE) hint.

Correct Answer: CD
Explanation:
As a database is changed by such actions as adding indexes or changing data in indexed columns, the original query plans used to access its tables should be optimized again by recompiling them. This optimization happens automatically the first time a stored procedure is run after Microsoft SQL Server is restarted. It also occurs if an underlying table used by the stored procedure changes. But if a new index is added from which the stored procedure might benefit, optimization does not happen until the next time the stored procedure is run after SQL Server is restarted. In this situation, it can be useful to force the stored procedure to recompile the next time it executes.

SQL Server provides three ways to force a stored procedure to recompile:
The sp_recompile system stored procedure forces a recompile of a stored procedure the next time that it is run. It does this by deleting the existing plan from the procedure cache forcing a new plan to be created the next time that the procedure is run.

Creating a stored procedure that specifies the WITH RECOMPILE option in its definition indicates that SQL Server does not cache a plan for this stored procedure; the stored procedure is recompiled every time that it is executed. Use of this option is uncommon and causes the stored procedure to execute more slowly, because the stored procedure must be recompiled every time that it is executed.

You can force the stored procedure to be recompiled by specifying the WITH RECOMPILE option when you execute the stored procedure.

A.CREATE and ALTER PROCEDURE syntax for CLR Stored Procedure does not have RECOMPILE option.
B.The RECOMPILE option is ignored for procedures created with FOR REPLICATION.
C.ENCRYPTION option and RECOMPILE option can go together.

QUESTION 18
You have a SQL Server database. The database contains two schemas named Marketing and Sales. The Marketing schema is owned by a user named MarketingManager. The Sales schema is owned by a user named SalesManager.

A user named John must be able to access the Sales.Orders table by using a stored procedure named Marketing.GetSalesSummary. John is not granted a SELECT permission on the Sales.Orders table. A user named SalesUser does have SELECT permission on the Sales.Orders table. You need to implement appropriate permissions for John and the stored procedure Marketing.GetSalesSummary.

What should you do?

A. Marketing.GetSalesSummary should be created by using the EXECUTE AS 'SalesUser' clause. John should be granted EXECUTE permission on Marketing.GetSalesSummary.

B. Marketing.GetSalesSummary should be created by using the EXECUTE AS OWNER clause. John should be granted EXECUTE WITH GRANT OPTION on Marketing.GetSalesSummary.
C. Marketing.GetSalesSummary should be created by using the EXECUTE AS CALLER clause. John should be granted IMPERSONATE permission for the user named SalesUser.
D. Marketing.GetSalesSummary should be created without an EXECUTE AS clause. John should be granted SELECT permission on the Sales.Orders table.

Correct Answer: A
Explanation:
1. When the module is executed, the Database Engine first verifies that the user executing the module has EXECUTE permission on the module. So John should be granted EXECUTE permission on Marketing.
GetSalesSummary stored procedure.
2. Additional permissions checks on objects that are accessed by the module are performed against the user account specified in the EXECUTE AS clause. The user executing the module is, in effect, impersonating the specified user. Because John is not granted a SELECT permission on the Sales.Orders table which is referenced by the stored procedure, EXECUTE AS CALLER is not suitable. (CALLER specifies the statements inside the module are executed in the context of the caller of the module. The user executing the module must have appropriate permissions not only on the module itself, but also on any database objects that are referenced by the module.) Because the user named SalesUser DOES have SELECT permission on the Sales.Orders table, he can be specified in EXECUTE AS clause. It means that Marketing.
GetSalesSummary stored procedure should be created by using the EXECUTE AS 'SalesUser' clause.

QUESTION 19
You need to create a stored procedure that accepts a table-valued parameter named @Customers. Which code segment should you use?

A. CREATE PROCEDURE AddCustomers (@Customers varchar(max))
B. CREATE PROCEDURE AddCustomers (@Customers Customer READONLY)
C. CREATE PROCEDURE AddCustomers (@Customers CustomerType OUTPUT)
D. CREATE PROCEDURE ADDCUSTOMERS
(@Customers varchar (max)) AS EXTERNAL NAME Customer.Add.NewCustomer

Correct Answer: B
Explanation:
To create and use table-valued parameters, follow these steps:

1. Create a table type and define the table structure.

```
/* Create a table type. */
CREATE TYPE LocationTableType AS TABLE
( LocationName VARCHAR(50)
, CostRate INT );
GO
```

2. Declare a routine that has a parameter of the table type.

```
/* Create a procedure to receive data for the table-valued parameter. */ CREATE PROCEDURE
usp_InsertProductionLocation
@TVP LocationTableType READONLY
AS
SET NOCOUNT ON
INSERT INTO [AdventureWorks2008R2].[Production].[Location]
([Name]
,[CostRate]
,[Availability]
,[ModifiedDate])
SELECT *, 0, GETDATE()
FROM @TVP;
GO
```

3. Declare a variable of the table type, and reference the table type.

```
/* Declare a variable that references the type. */ DECLARE @LocationTVP
AS LocationTableType;
```

4. Fill the table variable by using an INSERT statement.

```
/* Add data to the table variable. */
INSERT INTO @LocationTVP (LocationName, CostRate)
SELECT [Name], 0.00
FROM
[AdventureWorks2008R2].[Person].[StateProvince];
```

5. After the table variable is created and filled, you can pass the variable to a routine.

```
/* Pass the table variable data to a stored procedure. */ EXEC usp_InsertProductionLocation
@LocationTVP;
```

Restrictions

Table-valued parameters must be passed as input READONLY parameters to Transact- SQL routines. You cannot perform DML operations such as UPDATE, DELETE, or INSERT on a table-valued parameter in the body of a routine.

QUESTION 20

You have a computed column that is implemented with a user-defined function. The user- defined function returns a formatted account number. The column must be indexed to provide adequate search performance. You plan to create an index on the computed column. You need to identify the valid combination of ObjectPropertyEX values for the user-defined function. Which combination should you use?

A. IsDeterministic = True
IsSystemVerified = True
UserDataAccess = False
SystemDataAccess = False

B. IsDeterministic = True
IsSystemVerified = True
IsPrecise = True
IsTableFunction = True

C. IsDeterministic = False
IsSystemVerified = True
UserDataAccess = False
SystemDataAccess = False

D. IsDeterministic = False
IsSystemVerified = True
IsPrecise = True
SystemDataAccess = False

Correct Answer: A

QUESTION 21
You need to identify, within a given clause, if the month of February will contain 29 days for a specified year. Which object should you use?

A. DML trigger
B. Stored procedure
C. Table-valued function
D. Scalar-valued function

Correct Answer: D

QUESTION 22
You are creating a function that references a table.

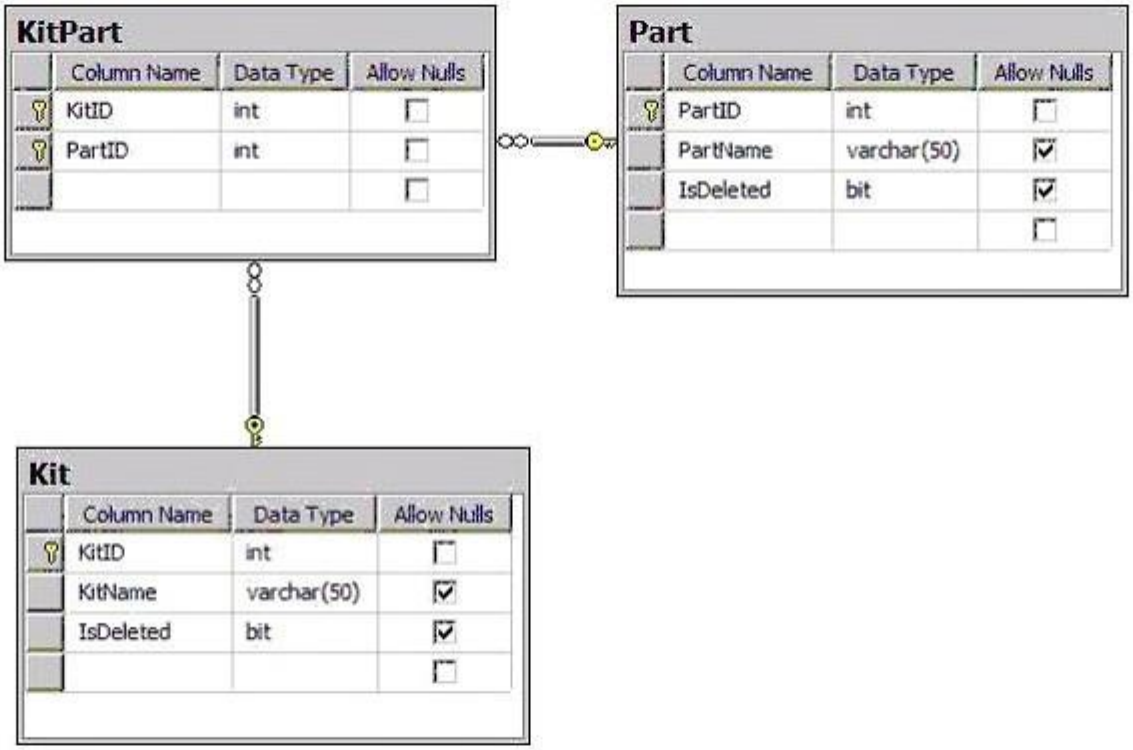
You need to prevent the table from being dropped.

Which option should you use when you create the function?

A. WITH ENCRYPTION
B. WITH EXECUTE AS
C. WITH SCHEMABINDING
D. WITH RETURNS NULL ON NULL INPUT

Correct Answer: C

QUESTION 23
Click the Exhibit button.



KitPart

	Column Name	Data Type	Allow Nulls
🔑	KitID	int	☐
🔑	PartID	int	☐
			☐

Part

	Column Name	Data Type	Allow Nulls
🔑	PartID	int	☐
	PartName	varchar(50)	☑
	IsDeleted	bit	☑
			☐

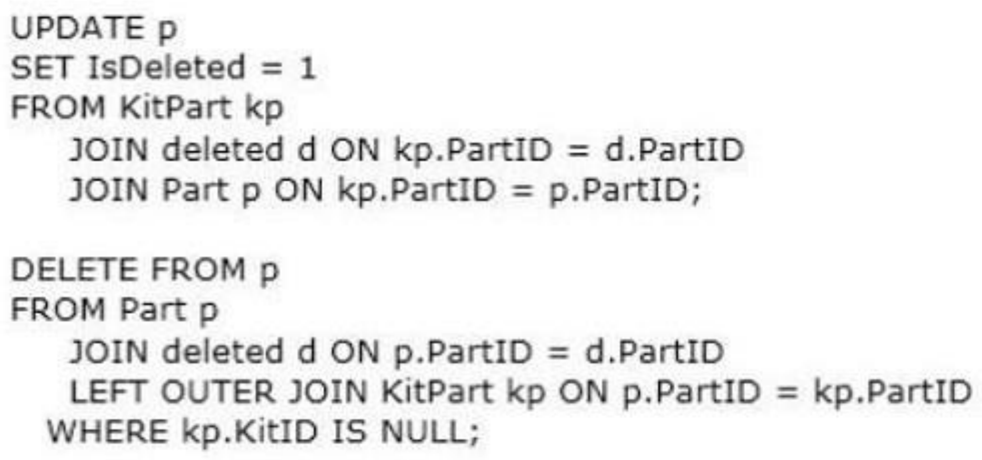
Kit

	Column Name	Data Type	Allow Nulls
🔑	KitID	int	☐
	KitName	varchar(50)	☑
	IsDeleted	bit	☑
			☐

You are developing a database using Microsoft SQL Server 2008. The database contains the tables shown in the exhibit.

You are required to prevent parts from being deleted if they belong to a kit. If a part belongs to a kit, the delete should not occur and the IsDeleted column for the row should be changed to 'True'. Parts can be deleted if they do not belong to a kit.

You have the following Transact-SQL statement to be used in a trigger:



```
UPDATE p
SET IsDeleted = 1
FROM KitPart kp
    JOIN deleted d ON kp.PartID = d.PartID
    JOIN Part p ON kp.PartID = p.PartID;

DELETE FROM p
FROM Part p
    JOIN deleted d ON p.PartID = d.PartID
    LEFT OUTER JOIN KitPart kp ON p.PartID = kp.PartID
  WHERE kp.KitID IS NULL;
```

You need to implement the Transact-SQL statement in a trigger.

Which trigger syntax should you use?

A. CREATE TRIGGER tr_Part_d ON Part
AFTER DELETE AS
BEGIN
...
END

B. CREATE TRIGGER tr_Part_d ON Part
INSTEAD OF DELETE AS
BEGIN
...
END

C. CREATE TRIGGER tr_KitPart_d ON KitPart
AFTER DELETE AS
BEGIN
...
END

D. CREATE TRIGGER tr_KitPart_d ON KitPart
INSTEAD OF DELETE AS
BEGIN
...
END

Correct Answer: B

QUESTION 24
You have a third-party application that inserts data directly into a table. You add two new columns to the table. These columns cannot accept NULL values and cannot use default constraints. You need to ensure that the new columns do not break the third-party application. What should you do?

A. Create a DDL trigger.
B. Create a stored procedure.
C. Create an AFTER INSERT trigger.
D. Create an INSTEAD OF INSERT trigger.

Correct Answer: D

QUESTION 25
Your database contains two tables named Order and OrderDetails that store order information. They relate to each other using the OrderID column in each table. Your business requires that the LastModifiedDate column in the Order table must reflect the date and time when a change is made in the OrderDetails table for the related order. You need to create a trigger to implement this business requirement. Which Transact-SQL statement should you use?

A.
```
CREATE TRIGGER [uModDate] ON [OrderDetails]
INSTEAD OF UPDATE FOR REPLICATION
AS
UPDATE [Order]
SET [LastModifiedDate] = GETDATE()
FROM inserted
WHERE inserted.[OrderID] = [Order].[OrderID];
```
B.
```
CREATE TRIGGER [uModDate] ON [Order]
INSTEAD OF UPDATE NOT FOR REPLICATION
AS
UPDATE [Order]
SET [LastModifiedDate] = GETDATE()
FROM inserted
WHERE inserted.[OrderID] = [Order].[OrderID];
```
C.
```
CREATE TRIGGER [uModDate] ON [Order]
AFTER UPDATE FOR REPLICATION
AS
UPDATE [Order]
SET [LastModifiedDate] = GETDATE()
FROM inserted
WHERE inserted.[OrderID] = [Order].[OrderID];
```
D.
```
CREATE TRIGGER [uModDate] ON [OrderDetails]
AFTER UPDATE NOT FOR REPLICATION
AS
UPDATE [Order]
SET [LastModifiedDate] = GETDATE()
FROM inserted
WHERE inserted.[OrderID] = [Order].[OrderID];
```

Correct Answer: D

QUESTION 26
You need to ensure that tables are not dropped from your database. What should you do?

A. Create a DDL trigger that contains COMMIT.
B. Create a DML trigger that contains COMMIT.
C. Create a DDL trigger that contains ROLLBACK.
D. Create a DML trigger that contains ROLLBACK.

Correct Answer: C
Explanation:
DDL triggers can execute either when a DDL statement is executed or when the user logs on to the SQL Server instance.
DDL triggers can be scoped at either the database or instance level. To scope a DDL trigger at the instance level, you use the ON ALL SERVER option. To scope a DDL trigger at the database level, you use the ON DATABASE option.

The following is an example of a DDL trigger:

```
CREATE TRIGGER tddl_tabledropalterprevent
ON DATABASE
FOR DROP_TABLE, ALTER_TABLE
AS
PRINT 'You are attempting to drop or alter tables in production!'
ROLLBACK;
```

Almost all DDL commands run within the context of a transaction. Because a DDL trigger also runs within the same transaction context, any DDL statement running in the context of a transaction can be rolled back.

QUESTION 27
You are responsible for a SQL Server database. You require the tables to be added or altered only on the first day of the month. You need to ensure that if the tables are attempted to be modified or created on any other day, an error is received and the attempt is not successful. Which Transact-SQL statement should you use?

A.
```
CREATE TRIGGER TRG_TABLES_ON_FIRST
ON DATABASE FOR CREATE_TABLE
AS
IF DATEPART(day,getdate())>1
BEGIN
RAISERROR ('Must wait til next month.', 16, 1)
END
```
B.
```
CREATE TRIGGER TRG_TABLES_ON_FIRST
ON DATABASE FOR CREATE_TABLE,ALTER_TABLE
AS
IF DATEPART(day,getdate())>1
BEGIN
RAISERROR ('Must wait til next month.', 16, 1)
END
```
C.
```
CREATE TRIGGER TRG_TABLES_ON_FIRST
ON DATABASE FOR CREATE_TABLE,ALTER_TABLE
AS
IF DATEPART(day,getdate())>1
BEGIN
```

ROLLBACK
RAISERROR ('Must wait til next month.', 16, 1)
END

D. CREATE TRIGGER TRG_TABLES_ON_FIRST
ON ALL SERVER FOR ALTER_DATABASE
AS
IF DATEPART(day,getdate())>1
BEGIN
ROLLBACK
RAISERROR ('Must wait til next month.', 16, 1)
END

Correct Answer: C

QUESTION 28

You have a single CLR assembly in your database. The assembly only references blessed assemblies from the Microsoft .NET Framework and does not access external resources. You need to deploy this assembly by using the minimum required permissions. You must ensure that your database remains as secure as possible. Which options should you set?

A. PERMISSION_SET = SAFE
TRUSTWORTHY ON

B. PERMISSION_SET = SAFE
TRUSTWORTHY OFF

C. PERMISSION_SET = UNSAFE
TRUSTWORTHY ON

D. PERMISSION_SET = EXTERNAL_ACCESS
TRUSTWORTHY OFF

Correct Answer: B

QUESTION 29

You have created an assembly that utilizes unmanaged code to access external resources. You need to deploy the assembly with the appropriate permissions. Which permission set should you use?

A. SAFE
B. UNSAFE
C. EXTERNAL_ACCESS
D. Default permission set

Correct Answer: B

QUESTION 30

You have tables named Products and OrderDetails. The Products table has a foreign key relationship with the OrderDetails table on the ProductID column. You have the following Transact-SQL batch:

BEGIN TRY

BEGIN TRANSACTION

```
DELETE FROM Products WHERE ProductID = 5;

BEGIN TRANSACTION

INSERT INTO OrderDetails

( OrderID, ProductID, Quantity )

VALUES

( 1234, 5, 12 );

COMMIT TRANSACTION

COMMIT TRANSACTION

END TRY

BEGIN CATCH

ROLLBACK TRANSACTION

PRINT ERROR_MESSAGE();

END CATCH
```

You need to analyze the result of executing this batch. What should be the expected outcome?

A. 1. The product will be deleted from the Products table.
   2. The order details will be inserted into the OrderDetails table.
B. 1. The product will be deleted from the Products table.
   2. The order details will not be inserted into the OrderDetails table.
C. 1. The product will not be deleted from the Products table.
   2. The order details will be inserted into the OrderDetails table.
D. 1. The product will not be deleted from the Products table.
   2. The order details will not be inserted into the OrderDetails table.

Correct Answer: D
Explanation:

```
ROLLBACK { TRAN | TRANSACTION }
[ transaction_name | @tran_name_variable
| savepoint_name | @savepoint_variable ]
[ ; ]
```

transaction_name
Is the name assigned to the transaction on BEGIN TRANSACTION. When nesting transactions, transaction_name must be the name from the outermost BEGIN TRANSACTION statement.
savepoint_name
Is savepoint_name from a SAVE TRANSACTION statement. Use savepoint_name when a conditional rollback should affect only part of the transaction.

ROLLBACK TRANSACTION without a savepoint_name or transaction_name rolls back to the beginning of the transaction. When nesting transactions, this same statement rolls back all inner transactions to the outermost BEGIN TRANSACTION statement. In both cases, ROLLBACK TRANSACTION decrements the @@TRANCOUNT system function to 0. ROLLBACK

TRANSACTION savepoint_name does not decrement @@TRANCOUNT.

A transaction cannot be rolled back after a COMMIT TRANSACTION statement is executed, except when the COMMIT TRANSACTION is associated with a nested transaction that is contained within the transaction being rolled back. In this instance, the nested transaction will also be rolled back, even if you have issued a COMMIT TRANSACTION for it.
SQL Server 2008 error handling best practice
CREATE PROCEDURE SaveTranExample
@InputCandidateID INT
AS - Detect whether the procedure was called from an active transaction and save that for later use. - In the procedure, @hasOuterTransaction = 0 means there was no active transaction -- and the procedure started one. - @hasOuterTransaction > 0 means an active transaction was started before the -- procedure was called.
DECLARE @hasOuterTransaction BIT = CASE WHEN @@TRANCOUNT > 0 THEN 1 ELSE 0 END;
 - Save points need unique names if modules can nest otherwise you can rollback -- to the wrong save point. The solution is to use a GUID to name the save points. DECLARE @rollbackPoint nchar(32) = REPLACE(CONVERT(NCHAR(36), NEWID()), N'-', N'');
IF @hasOuterTransaction > 0
BEGIN - Procedure called when there is an active transaction. -- Create a savepoint to be able to roll back only the work done in the procedure if there is an error.
SAVE TRANSACTION @rollbackPoint;
END
ELSE - Procedure must start its own transaction.
BEGIN TRANSACTION @rollbackPoint; - Modify database.
BEGIN TRY - Do work;
DELETE HumanResources.JobCandidate
WHERE JobCandidateID = @InputCandidateID; - Get here if no errors; must commit - any transaction started in the - procedure, but not commit a transaction - started before the transaction was called.
IF @hasOuterTransaction = 0
BEGIN - @hasOuterTransaction = 0 means no transaction was started before the procedure was called. - The procedure must commit the transaction it started.
COMMIT TRANSACTION;
END
END TRY
BEGIN CATCH - An error occurred; - If the transaction is still valid
IF XACT_STATE() = 1 - The XACT_STATE function can return the following values:
- 1 An open transaction exists that can be either committed or rolled back.
- 0 There is no open transaction.
- -1 An open transaction exists, but it is in a doomed state. Due to the type of error that was raised, the transaction can only be rolled back.

BEGIN
- Because the syntax for ROLLBACK TRANSACTION is the same for the transaction and for a savepoint
- (ROLLBACK TRANSACTION [transaction_name | @tran_name_variable | savepoint_name | @savepoint_variable])

- we can write the following:
ROLLBACK TRANSACTION @rollbackPoint;
- In case @rollbackPoint has the name of a transaction, roll back to the beginning of the transaction.
- In case @rollbackPoint has the name of a savepoint, roll back to the savepoint.
END;
ELSE IF XACT_STATE() = -1

```
BEGIN
IF @hasOuterTransaction = 0
BEGIN - Transaction started in procedure. - Roll back complete transaction.
ROLLBACK TRANSACTION;
END
- If the transaction is uncommitable, a rollback to the savepoint is not allowed
- because the savepoint rollback writes to the log. Just return to the caller, which
- should roll back the outer transaction.
END
- Execute Standard module error handler;
- After the appropriate rollback, echo error information to the caller.
DECLARE @ErrorMessage NVARCHAR(4000);
DECLARE @ErrorSeverity INT;
DECLARE @ErrorState INT;
SELECT @ErrorMessage = ERROR_MESSAGE();
SELECT @ErrorSeverity = ERROR_SEVERITY();
SELECT @ErrorState = ERROR_STATE();
RAISERROR (@ErrorMessage, -- Message text.
@ErrorSeverity, -- Severity.
@ErrorState -- State.
);
END CATCH
GO
```