➢ Vendor: Microsoft

➢ Exam Code: 70-761

➢ Exam Name: Querying Data with Transact-SQL

➢ New Questions (May/2018)

NEW QUESTION 125

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

- accepts a product identifies as input
- calculates the total quantity of a specific product, including quantity on hand and quantity on - order
- caches and reuses execution plan
- returns a value
- can be called from within a SELECT statement
- can be used in a JOIN clause

What should you create?

A. a temporary table that has a columnstore index
B. a user-defined table-valued function
C. a memory-optimized table that has updated statistics
D. a natively-complied stored procedure that has an OUTPUT parameter

Answer: B
Explanation:

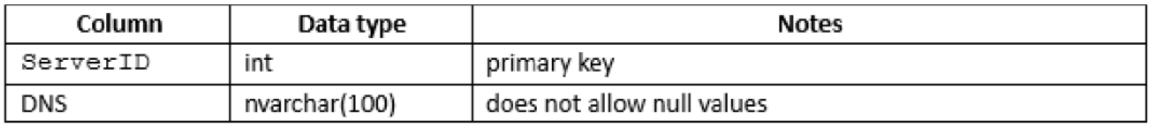
A table-valued user-defined function can also replace stored procedures that return a single result set. The table returned by a user-defined function can be referenced in the FROM clause of a Transact-SQL statement, but stored procedures that return result sets cannot.
https://technet.microsoft.com/en-us/library/ms191165(v=sql.105).aspx

NEW QUESTION 126

Drag and Drop

You have a database containing the following tables:

Servers



Column	Data type	Notes
ServerID	int	primary key
DNS	nvarchar(100)	does not allow null values

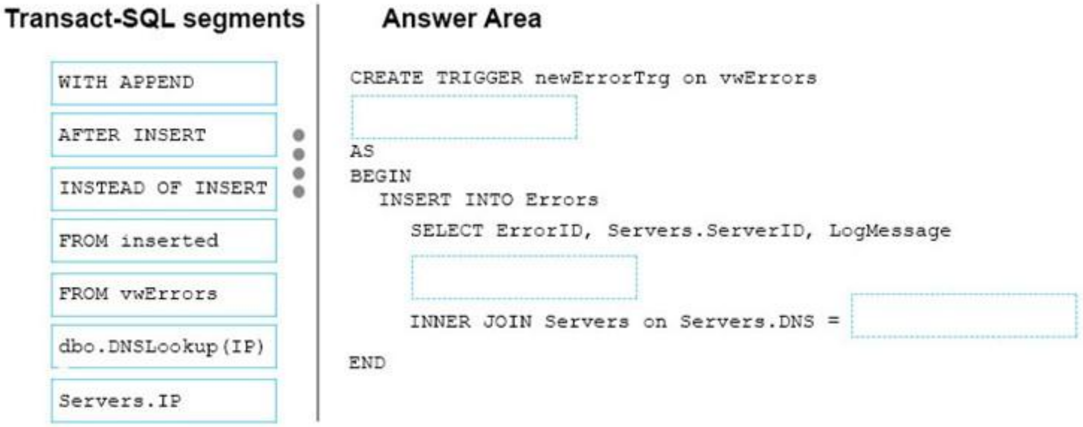
Errors

Column	Data type	Notes
ErrorID	int	primary key
ServerID	int	does not allow null values, foreign key to Servers table
LogMessage	nvarchar(max)	does not allow null values

You have a user-defined, scalar function named IPLookup that takes a DNS name as a parameter and returns the IP address of the server. You have an additional user-defined, scalar function named DNSLookup, that takes an IP address as a parameter and returns a DNS name. You create a view named vwErrors by running the following Transact-SQL statement:

```
CREATE VIEW vwErrors
AS
        SELECT ErrorID,IPLookup(DNS) as IP,LogMessage
        FROM Errors
        INNER JOIN Servers ON Errors.ServerID = Servers.ServerID
```

You need to insert data by using the view. How should you complete the Transact-SQL statement? (To answer, drag the appropriate Transact-SQL segments to the correct location. Each Transact-SQL segments 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.)



Transact-SQL segments

- WITH APPEND
- AFTER INSERT
- INSTEAD OF INSERT
- FROM inserted
- FROM vwErrors
- dbo.DNSLookup(IP)
- Servers.IP

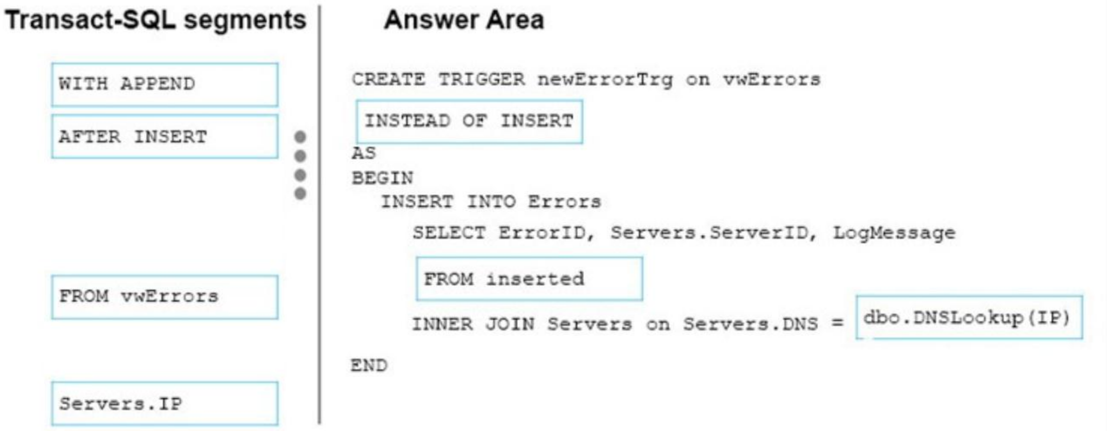
Answer Area

```
CREATE TRIGGER newErrorTrg on vwErrors

AS
BEGIN
  INSERT INTO Errors
     SELECT ErrorID, Servers.ServerID, LogMessage

     INNER JOIN Servers on Servers.DNS =
END
```

Answer:



Transact-SQL segments

- WITH APPEND
- AFTER INSERT
- FROM vwErrors
- Servers.IP

Answer Area

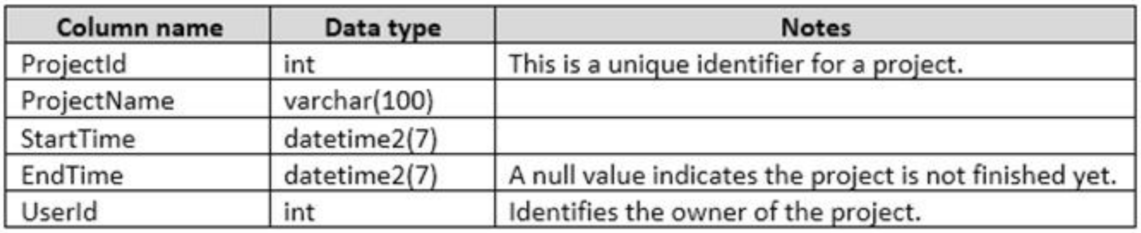
```
CREATE TRIGGER newErrorTrg on vwErrors
 INSTEAD OF INSERT
AS
BEGIN
  INSERT INTO Errors
     SELECT ErrorID, Servers.ServerID, LogMessage
      FROM inserted
     INNER JOIN Servers on Servers.DNS = dbo.DNSLookup(IP)
END
```

Explanation:
https://docs.microsoft.com/en-us/sql/t-sql/queries/output-clause-transact-sql

NEW QUESTION 127

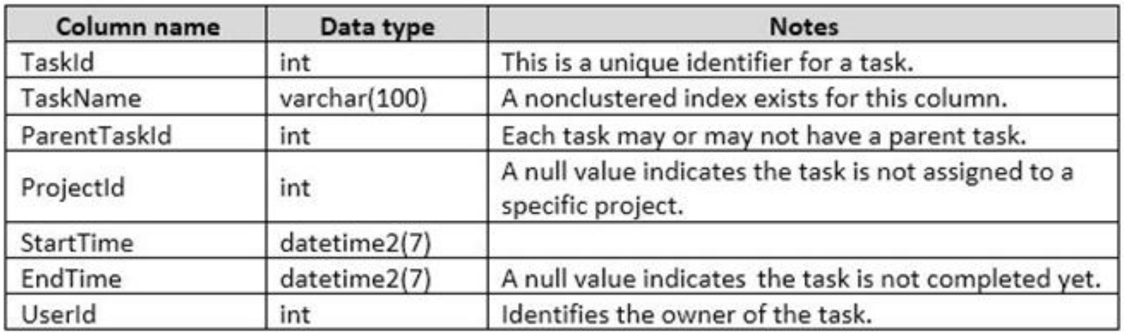
Hotspot

You query a database that includes two tables: Project and Task. The Project table includes the following columns:



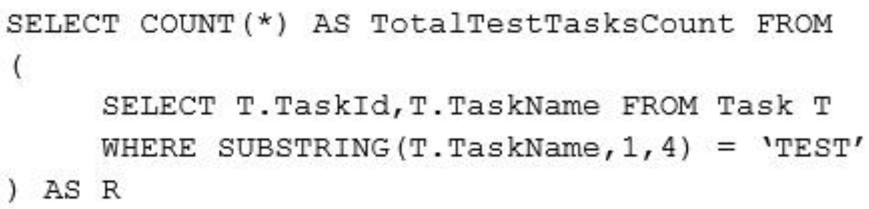
Column name	Data type	Notes
ProjectId	int	This is a unique identifier for a project.
ProjectName	varchar(100)	
StartTime	datetime2(7)	
EndTime	datetime2(7)	A null value indicates the project is not finished yet.
UserId	int	Identifies the owner of the project.

The Task table includes the following columns:



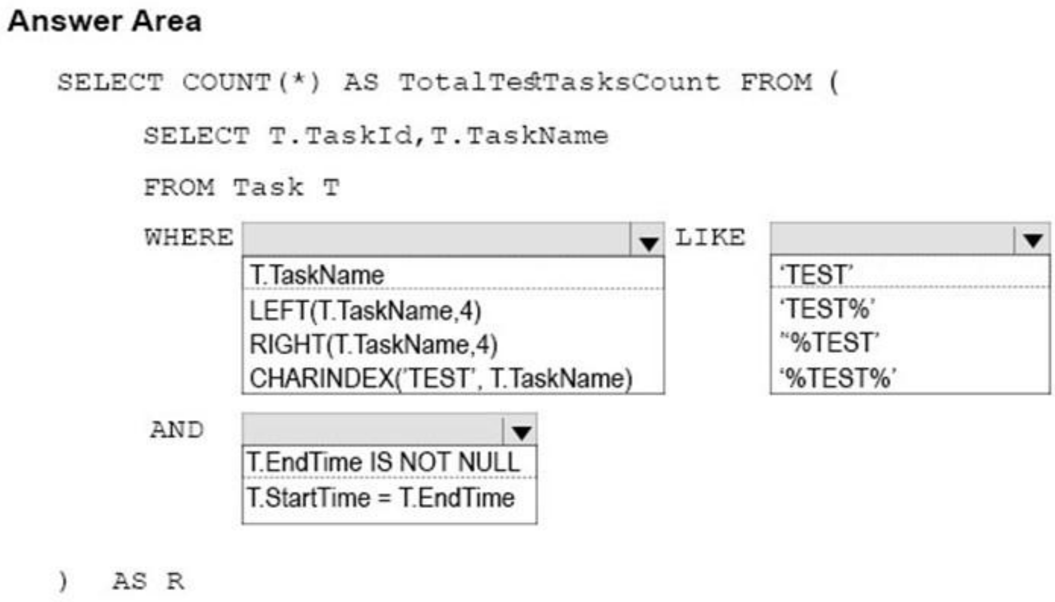
Column name	Data type	Notes
TaskId	int	This is a unique identifier for a task.
TaskName	varchar(100)	A nonclustered index exists for this column.
ParentTaskId	int	Each task may or may not have a parent task.
ProjectId	int	A null value indicates the task is not assigned to a specific project.
StartTime	datetime2(7)	
EndTime	datetime2(7)	A null value indicates the task is not completed yet.
UserId	int	Identifies the owner of the task.

Users report performance issues when they run the following query:



```
SELECT COUNT(*) AS TotalTestTasksCount FROM
(
     SELECT T.TaskId,T.TaskName FROM Task T
     WHERE SUBSTRING(T.TaskName,1,4) = ‘TEST’
) AS R
```

You need to improve query performance and limit results to projects that specify an end date. How should you complete the Transact-SQL statement? (To answer, select the appropriate Transact-SQL segments in the answer area.)



Answer Area

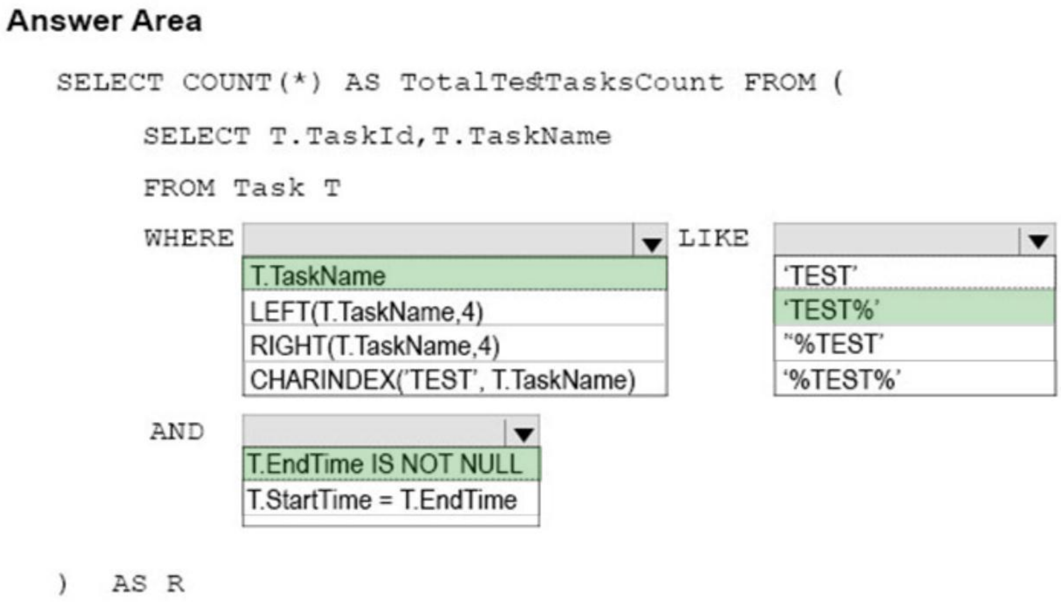
```
SELECT COUNT(*) AS TotalTestTasksCount FROM (
        SELECT T.TaskId,T.TaskName
        FROM Task T
        WHERE [▼] LIKE [▼]
        AND [▼]
) AS R
```

WHERE options:
- T.TaskName
- LEFT(T.TaskName,4)
- RIGHT(T.TaskName,4)
- CHARINDEX('TEST', T.TaskName)

LIKE options:
- 'TEST'
- 'TEST%'
- "%TEST'
- '%TEST%'

AND options:
- T.EndTime IS NOT NULL
- T.StartTime = T.EndTime

Answer:



Answer Area

```
SELECT COUNT(*) AS TotalTestTasksCount FROM (
        SELECT T.TaskId,T.TaskName
        FROM Task T
        WHERE [▼] LIKE [▼]
        AND [▼]
) AS R
```

WHERE options:
- [✓ T.TaskName]
- LEFT(T.TaskName,4)
- RIGHT(T.TaskName,4)
- CHARINDEX('TEST', T.TaskName)

LIKE options:
- 'TEST'
- [✓ 'TEST%']
- "%TEST'
- '%TEST%'

AND options:
- [✓ T.EndTime IS NOT NULL]
- T.StartTime = T.EndTime

Explanation:
Wildcard character %: Any string of zero or more characters. For example: If the LIKE '5%' symbol is specified, the Database Engine searches for the number 5 followed by any string of zero or more characters.
https://docs.microsoft.com/en-us/sql/t-sql/language-elements/like-transact-sql

NEW QUESTION 128
You are building a stored procedure that will update data in a table named Table1 by using a complex query as the data source. You need to ensure that the SELECT statement in the stored procedure meets the following requirements:

```
- Data being processed must be usable in several statements in the stored
```

```
procedure.
- Data being processed must contain statistics.
```

What should you do?

A. Update Table1 by using a common table expression (CTE).
B. Insert the data into a temporary table, and then update Table1 from the temporary table.
C. Place the SELECT statement in a derived table, and then update Table1 by using a JOIN to the derived table.
D. Insert the data into a table variable, and then update Table1 from the table variable.

Answer: B
Explanation:
Incorrect:
Not A: CTEs do not have dedicated stats. They rely on stats on the underlying objects.
Not C: Unlike a derived table, a CTE can be self-referencing and can be referenced multiple times in the same query.
https://technet.microsoft.com/en-us/library/ms190766(v=sql.105).aspx
https://dba.stackexchange.com/questions/13112/whats-the-difference-between-a-cte-and-a-temp-table

NEW QUESTION 129
You have a disk-based table that contains 15 columns. You query the table for the number of new rows created during the current day. You need to create an index for the query. The solution must generate the smallest possible index. Which type of index should you create?

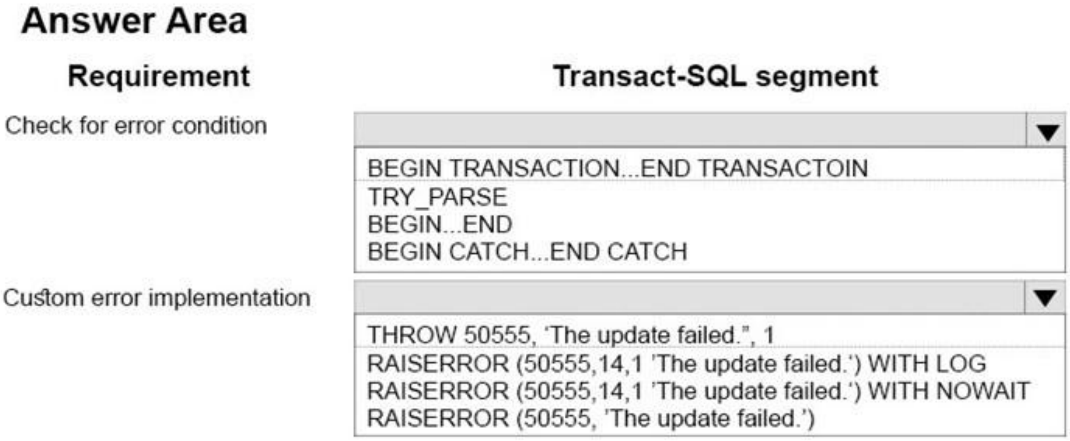
A. clustered
B. filtered nonclustered with a getdate() predicate in the WHERE statement clause
C. hash
D. nonclustered with compression enabled

Answer: B
Explanation:
A filtered index is an optimized nonclustered index especially suited to cover queries that select from a well-defined subset of data. It uses a filter predicate to index a portion of rows in the table. A well-designed filtered index can improve query performance as well as reduce index maintenance and storage costs compared with full-table indexes. Creating a filtered index can reduce disk storage for nonclustered indexes when a full-table index is not necessary.
https://docs.microsoft.com/en-us/sql/relational-databases/indexes/create-filtered-indexes

NEW QUESTION 130
Hotspot
You need to develop a Transact-SQL statement that meets the following requirements:

```
- The statement must return a custom error when there are problems
updating a table.
- The error number must be the value 50555.
- The error severity level must be 14.
- A Microsoft SQL Server alert must be triggered when the error condition
occurs.
```

Which Transact-SQL segment should you use for each requirement? (To answer, select the appropriate Transact-SQL segments in the answer area.)



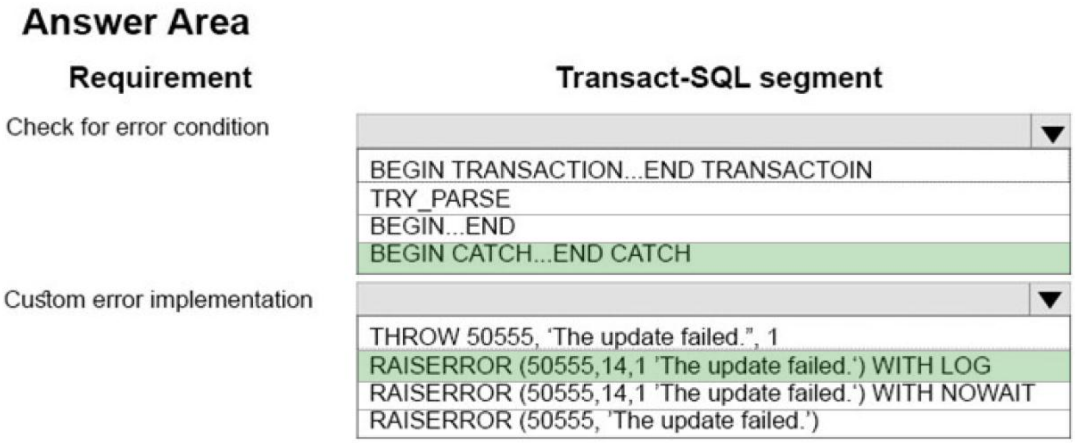
Answer Area

Requirement	Transact-SQL segment

Check for error condition	BEGIN TRANSACTION...END TRANSACTOIN
	TRY_PARSE
	BEGIN...END
	BEGIN CATCH...END CATCH

Custom error implementation	THROW 50555, 'The update failed.", 1 RAISERROR (50555,14,1 'The update failed.') WITH LOG RAISERROR (50555,14,1 'The update failed.') WITH NOWAIT RAISERROR (50555, 'The update failed.')

Answer:



Answer Area

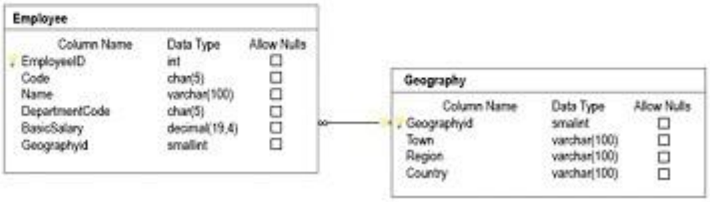
Requirement	Transact-SQL segment

Check for error condition	BEGIN TRANSACTION...END TRANSACTOIN
	TRY_PARSE
	BEGIN...END
	[✓ BEGIN CATCH...END CATCH]

Custom error implementation	THROW 50555, 'The update failed.", 1 [✓ RAISERROR (50555,14,1 'The update failed.') WITH LOG] RAISERROR (50555,14,1 'The update failed.') WITH NOWAIT RAISERROR (50555, 'The update failed.')

Explanation:
https://msdn.microsoft.com/en-us/library/ms178592.aspx

NEW QUESTION 131
Hotspot
You have two tables as shown in the following image:



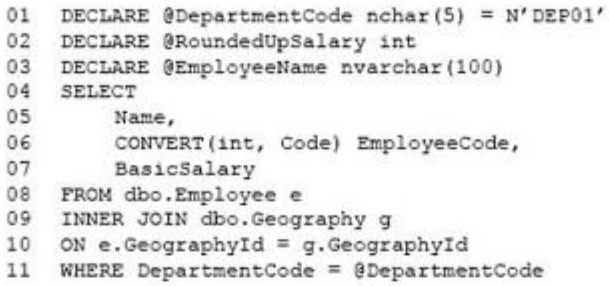
Employee

Column Name	Data Type	Allow Nulls
EmployeeID	int	□
Code	char(5)	□
Name	varchar(100)	□
DepartmentCode	char(5)	□
BasicSalary	decimal(19,4)	□
Geographyid	smallint	□

Geography

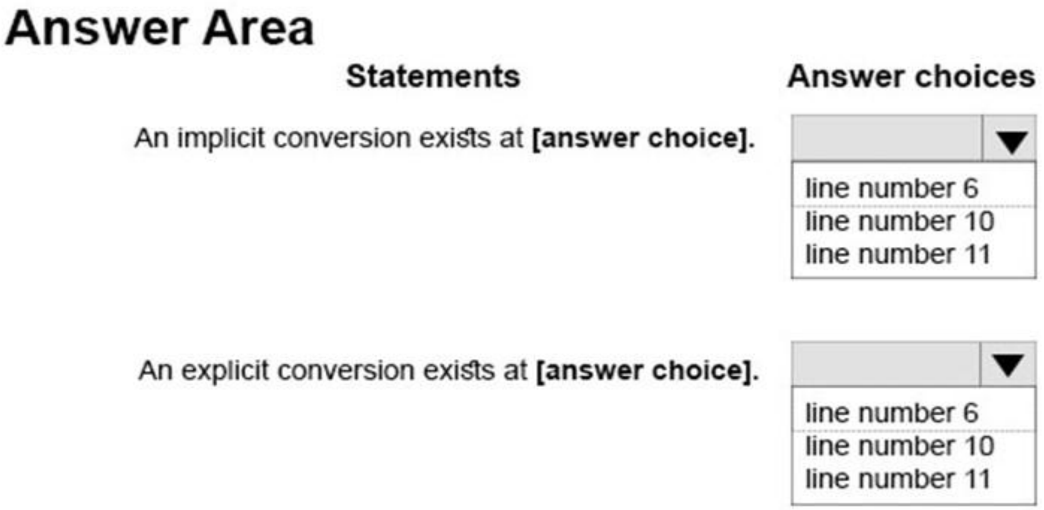
Column Name	Data Type	Allow Nulls
Geographyid	smallint	□
Town	varchar(100)	□
Region	varchar(100)	□
Country	varchar(100)	□

You need to analyze the following query. (Line numbers are included for reference only.)



```
01  DECLARE @DepartmentCode nchar(5) = N'DEP01'
02  DECLARE @RoundedUpSalary int
03  DECLARE @EmployeeName nvarchar(100)
04  SELECT
05      Name,
06      CONVERT(int, Code) EmployeeCode,
07      BasicSalary
08  FROM dbo.Employee e
09  INNER JOIN dbo.Geography g
10  ON e.GeographyId = g.GeographyId
11  WHERE DepartmentCode = @DepartmentCode
```

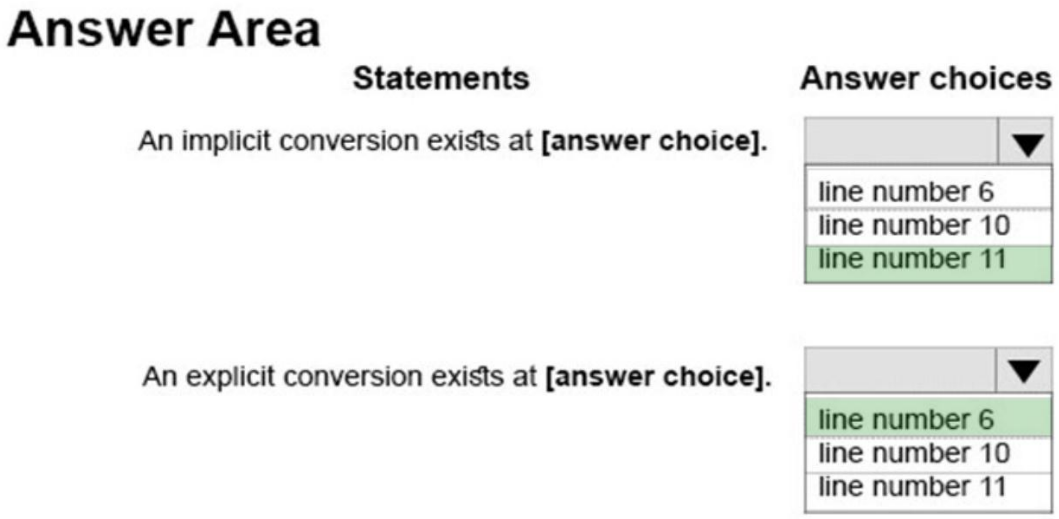
Use the drop-down menus to select the answer choice that completes each statement based on the information presented in the graphic.



Answer Area

Statements	Answer choices
An implicit conversion exists at **[answer choice]**.	line number 6 line number 10 line number 11
An explicit conversion exists at **[answer choice]**.	line number 6 line number 10 line number 11

Answer:



Answer Area

Statements	Answer choices
An implicit conversion exists at **[answer choice]**.	line number 6 line number 10 [✓ line number 11]
An explicit conversion exists at **[answer choice]**.	[✓ line number 6] line number 10 line number 11

Explanation:

To compare char(5) and nchar(5) an implicit conversion has to take place. Explicit conversions use the CAST or CONVERT functions, as in line number 6.

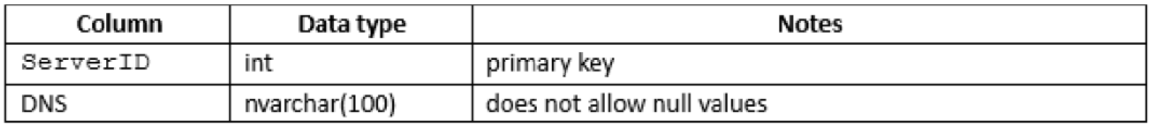
https://docs.microsoft.com/en-us/sql/t-sql/data-types/data-type-conversion-database-engine#implicit-and-explicit-conversion

NEW QUESTION 132

Drag and Drop

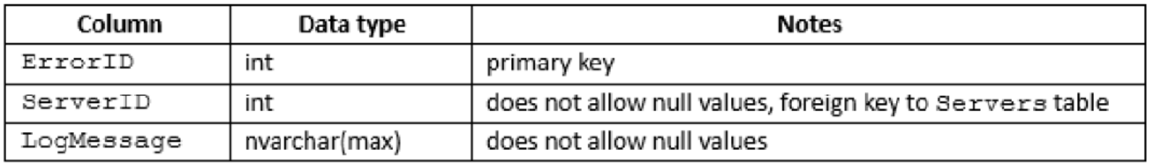
You have a database containing the following tables:

Servers



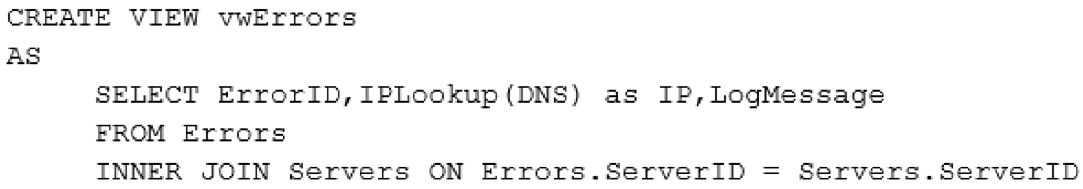
Column	Data type	Notes
ServerID	int	primary key
DNS	nvarchar(100)	does not allow null values

Errors



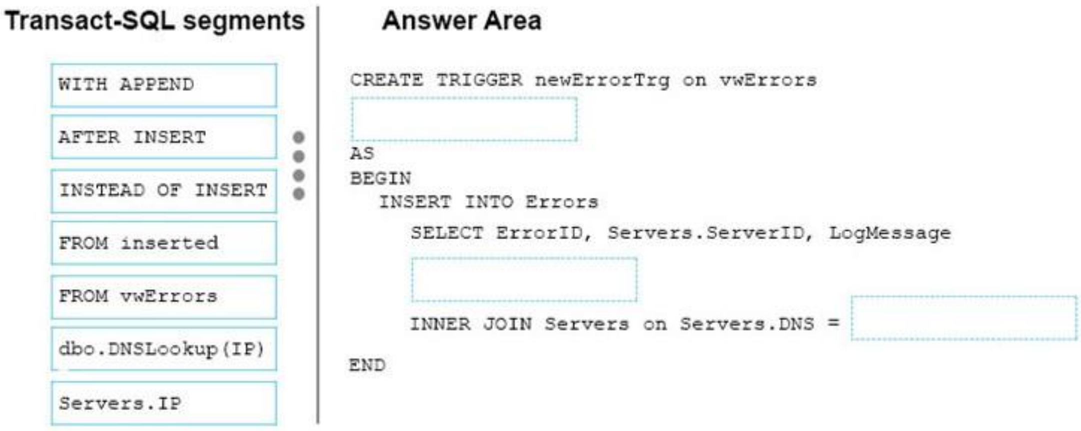
Column	Data type	Notes
ErrorID	int	primary key
ServerID	int	does not allow null values, foreign key to Servers table
LogMessage	nvarchar(max)	does not allow null values

You have a user-defined, scalar function named IPLookup that takes a DNS name as a parameter and returns the IP address of the server. You have an additional user-defined, scalar function named DNSLookup, that takes an IP address as a parameter and returns a DNS name. You create a view named vwErrors by running the following Transact-SQL statement:



```
CREATE VIEW vwErrors
AS
      SELECT ErrorID,IPLookup(DNS) as IP,LogMessage
      FROM Errors
      INNER JOIN Servers ON Errors.ServerID = Servers.ServerID
```

You need to insert data by using the view. How should you complete the Transact-SQL statement? (To answer, drag the appropriate Transact-SQL segments to the correct location. Each Transact-SQL segments 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.)



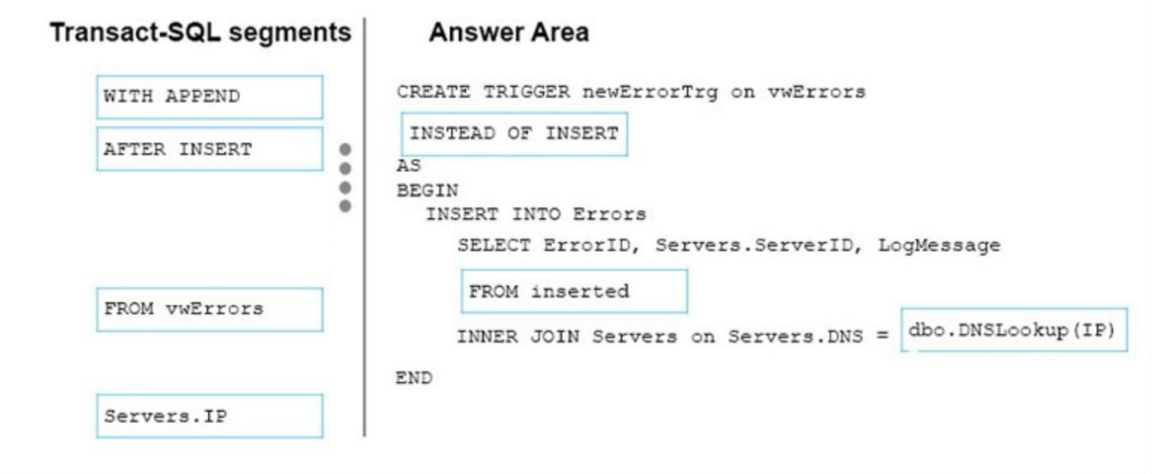
Transact-SQL segments

- WITH APPEND
- AFTER INSERT
- INSTEAD OF INSERT
- FROM inserted
- FROM vwErrors
- dbo.DNSLookup(IP)
- Servers.IP

Answer Area

```
CREATE TRIGGER newErrorTrg on vwErrors
□
AS
BEGIN
  INSERT INTO Errors
     SELECT ErrorID, Servers.ServerID, LogMessage
     □
     INNER JOIN Servers on Servers.DNS = □
END
```

Answer:



Transact-SQL segments

WITH APPEND

AFTER INSERT

FROM vwErrors

Servers.IP

Answer Area

```
CREATE TRIGGER newErrorTrg on vwErrors
 INSTEAD OF INSERT
AS
BEGIN
  INSERT INTO Errors
     SELECT ErrorID, Servers.ServerID, LogMessage
      FROM inserted
     INNER JOIN Servers on Servers.DNS = dbo.DNSLookup(IP)
END
```

Explanation:

https://docs.microsoft.com/en-us/sql/t-sql/queries/output-clause-transact-sql

NEW QUESTION 133

......