70-461 Q&As

Querying Microsoft SQL Server 2012/2014

Pass Microsoft 70-461 Exam with 100% Guarantee

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

https://www.passapply.com/70-461.html

100% Passing Guarantee
100% Money Back Assurance

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

QUESTION 1

You use Microsoft SQL Server to develop a database application.

Your application sends data to a VARCHAR(50) variable named @var.

You need to write a Transact-SQL statement that will return information on a successful or unsuccessful cast to an integer in a table.

Which Transact-SQL statement should you run?

A.
```
SELECT
    CASE
    WHEN CONVERT(int, @var) IS NULL
    THEN 'True'
    ELSE 'False'
    END
 As BadCast
```

B.
```
SELECT
    IIF(TRY_PARSE(@var AS int) IS NULL,
   'True',
   'False'
)
As BadCast
```

C.
```
TRY(
  SELECT CONVERT(int, @var)
  SELECT 'True' As BadCast
)
CATCH
  SELECT 'False' As BadCast
)
```

D.
```
BEGIN TRY
SELECT
  CONVERT(int, @var) as Value,
  'True' As BadCast
END TRY
BEGIN CATCH
SELECT
  CONVERT(int, @var) as Value,
'False' As BadCast
END CATCH
```

A. Option A

B. Option B

C. Option C

D. Option D

Correct Answer: B

TRY_PARSE returns the result of an expression, translated to the requested data type, or null if the cast fails in SQL Server. Use TRY_PARSE only for converting from string to date/time and number types. References: https://docs.microsoft.com/en-us/sql/t-sql/functions/try-parse-transact-sql?view=sql-server-2017

QUESTION 2

You are developing an SQL Server database. The database contains two tables and a function that are defined by the following Transact-SQL statements.

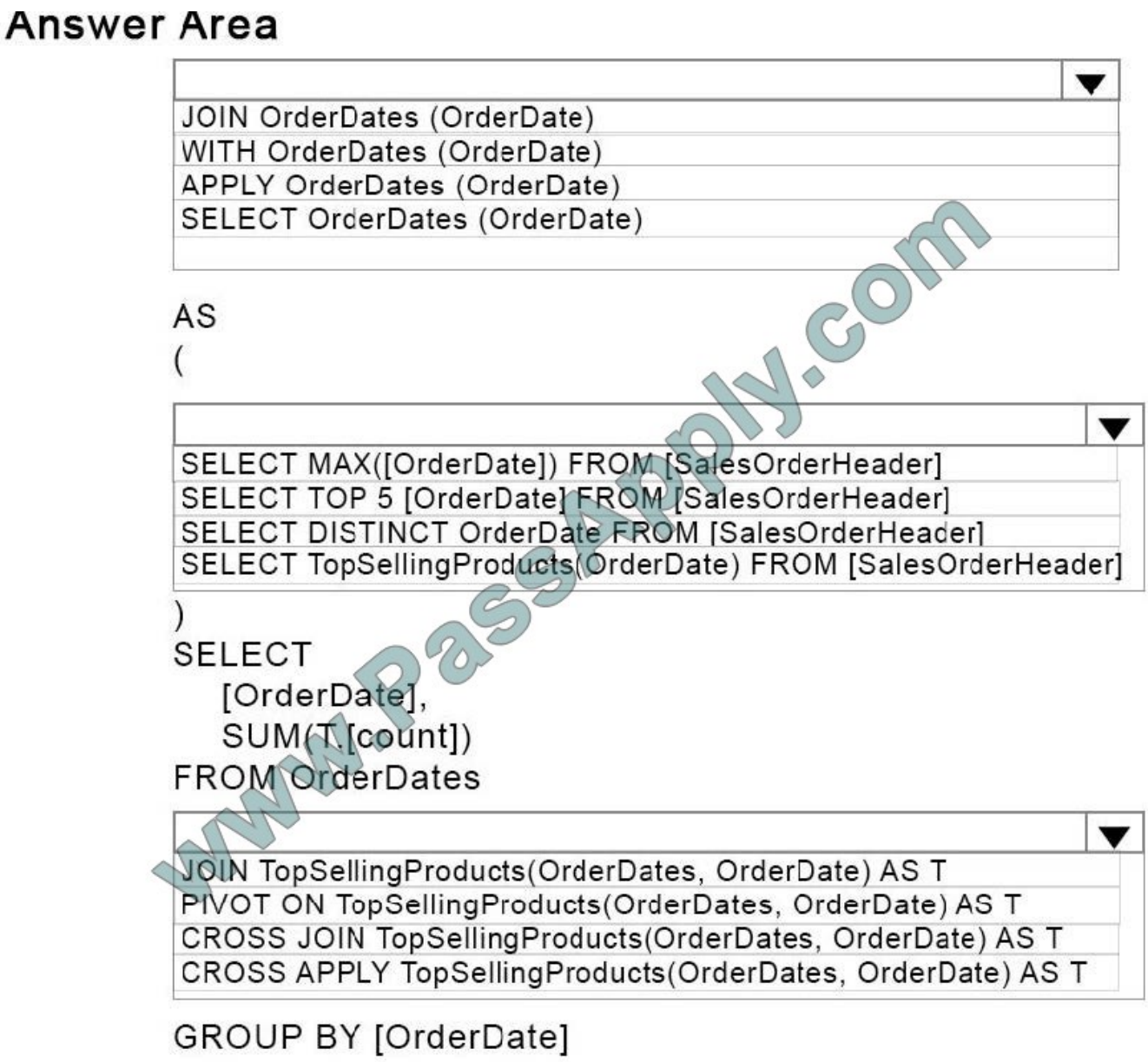
```
CREATE TABLE [dbo].[SalesOrderDetail](
    [SalesOrderID] [int] NOT NULL,
    [SalesOrderDetailID] [int] IDENTITY(1,1) NOT NULL,
    [OrderQty] [smallint] NOT NULL,
    [ProductID] [int] NOT NULL,
    [UnitPrice] [money] NOT NULL,
    [LineTotal] [numeric](38, 6) NOT NULL,
CONSTRAINT [PK_SalesOrderDetail] PRIMARY KEY CLUSTERED
(
[SalesOrderDetailID] ASC
))

CREATE TABLE [dbo].[SalesOrderHeader](
    [SalesOrderID] [int] IDENTITY(1,1) NOT NULL,
    [OrderDate] [datetime] NOT NULL,
    [Status] [tinyint] NOT NULL,
    [PurchaseOrderNumber] [nvarchar](25) NULL,
    [AccountNumber] [nvarchar](15) NULL,
    [CustomerID] [int] NOT NULL,
    [TotalDue] [money] NOT NULL,
CONSTRAINT [PK_SalesOrderHeader] PRIMARY KEY CLUSTERED
(
[SalesOrderID] ASC
))

CREATE FUNCTION TopSellingProducts
(
   @date datetime
)
RETURNS TABLE
AS
RETURN
(
   SELECT TOP 5
      COUNT([SalesOrderDetail].ProductID) [count],
      [SalesOrderDetail].ProductID
   FROM [SalesOrderHeader]
   INNER JOIN [SalesOrderDetail] ON [SalesOrderHeader].[SalesOrderID] = [SalesOrderDetail].[SalesOrderID]
   WHERE [OrderDate] >= dateadd(day,datediff(day,1,@date),0)
     AND [OrderDate] < dateadd(day,datediff(day,0,@date),0)
   GROUP BY [SalesOrderDetail].ProductID
   ORDER BY COUNT ([SalesOrderDetail].ProductID) DESC
)
```


You need to create a query to determine the total number of products that are sold each day for the live top-selling

products on that particular day.

How should you complete the relevant Transact-SQL script? To answer, select the appropriate Transact-SQL statements from each list in the answer area.

Hot Area:



Correct Answer:

Answer Area

The APPLY operator allows you to invoke a table-valued function for each row returned by an outer table expression of a query.

There are two forms of APPLY: CROSS APPLY and OUTER APPLY. CROSS APPLY returns only rows from the outer table that produce a result set from the table-valued function. OUTER APPLY returns both rows that produce a result set,

and rows that do not, with NULL values in the columns produced by the table-valued function.

QUESTION 3

When referring in the OUTPUT clause to columns from the inserted rows, when should you prefix the columns with the keyword inserted?

A. Always

B. Never

C. Only when the statement is UPDATE

D. Only when the statement is MERGE

Correct Answer: A

QUESTION 4

Why are window functions allowed only in the SELECT and ORDER BY clauses of a query?

A. Because they are supposed to operate on the underlying query\\'s result, which is achieved when logical query processing gets to the SELECT phase.

B. Because Microsoft didn\\'t have time to implement them in other clauses.

C. Because you never need to filter or group data based on the result of window functions.

D. Because in the other clauses, the functions are considered door functions (also known as backdoor functions).

Correct Answer: A

QUESTION 5

What is the difference between the simple CASE expression and the searched CASE expression?

A. The simple CASE expression is used when the database recovery model is simple, and the searched CASE expression is used when it\\'s full or bulk logged.

B. The simple CASE expression compares an input expression to multiple possible expressions in the WHEN clauses, and the searched CASE expression uses independent predicates in the WHEN clauses.

C. The simple CASE expression can be used anywhere in a query, and the searched CASE expression can be used only in the WHERE clause.

D. The simple CASE expression can be used anywhere in a query, and the searched CASE expression can be used only in query filters (ON, WHERE, HAVING).

Correct Answer: B