[Sep-2017 Dumps PassLeader 98q 70-761 PDF Study Guide with Free VCE Dumps Collection

New Updated 70-761 Exam Questions from PassLeader 70-761 PDF dumps! Welcome to download the newest PassLeader 70-761 VCE dumps: https://www.passleader.com/70-761.html (98 Q&As) Keywords: 70-761 exam dumps, 70-761 exam questions, 70-761 VCE dumps, 70-761 PDF dumps, 70-761 practice tests, 70-761 study guide, 70-761 braindumps, Querying Data with Transact-SQL Exam P.S. Free 70-761 dumps download from Google Drive:
https://drive.google.com/open?id=0B-ob6L_QjGLpaEZzRVFnOE9OenM Free 70-762 dumps download from Google Drive:
https://drive.google.com/open?id=0B-ob6L_QjGLpN3RVQ25sVUM5dkU Free 70-764 dumps download from Google Drive:
https://drive.google.com/open?id=0B-ob6L_QjGLpN3N6eHJ6Z2EzZWc Free 70-765 dumps download from Google Drive:
https://drive.google.com/open?id=0B-ob6L_QjGLpZHlHSG5KM09xUms Free 70-767 dumps download from Google Drive:
https://drive.google.com/open?id=0B-ob6L_QjGLpcXZXWUl4dHhIUVk Free 70-768 dumps download from Google Drive:
https://drive.google.com/open?id=0B-ob6L_QjGLpeXAxaUJkWEZnVlU NEW QUESTION 75Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution. After you answer a question in this section. You will NOT be able to return to it. As a result, these questions will not appear in the review screen.A database has two tables as shown in the following database diagram:
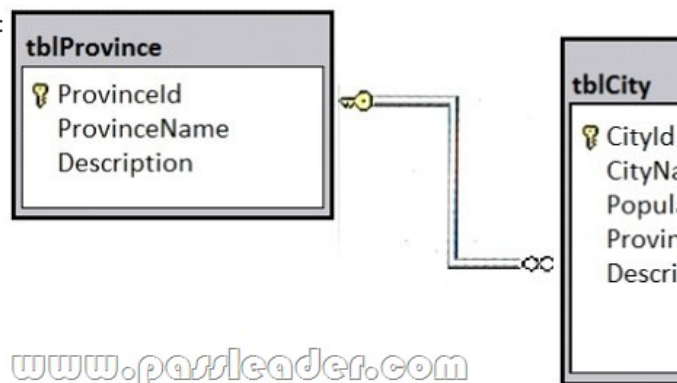
You need to list all provinces that have at least two large cities. A large city is defined as having a population of at least one million residents. The query must return the following columns:

- tblProvince.ProvinceId
- tblProvince.ProvinceName
- a derived column named LargeCityCount that presents the total count of large cities for t

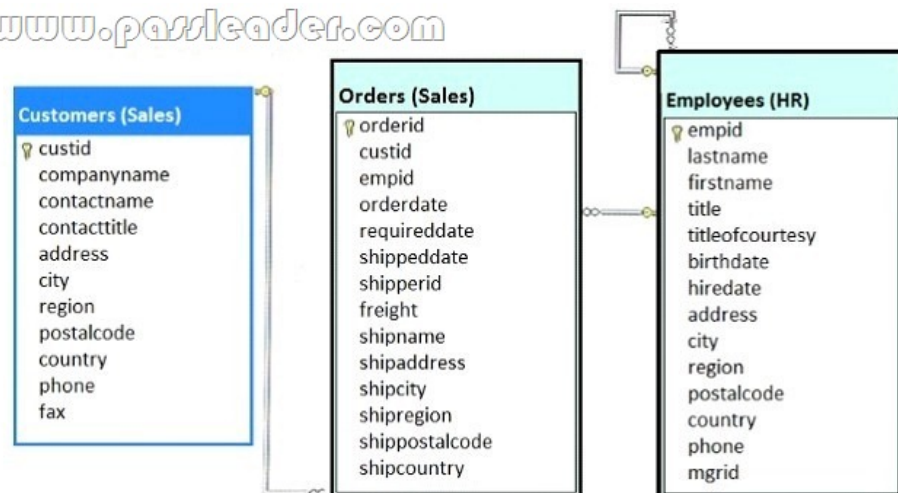Solution: You run the following Transact-SQL statement:

```
SELECT P.ProvinceId, P.ProvinceName, CitySummary.LargeCityCount
FROM tblProvince P
CROSS APPLY (
    SELECT COUNT(*) AS LargeCityCount FROM tblCity C
    WHERE C.Population>=1000000 AND C.ProvinceId = P.ProvinceId
) CitySummary
WHERE CitySummary.LargeCityCount >=2
```

Does the solution meet the goal? A.   YesB.   No Answer: AExplanation:The requirement to list all provinces that have at least two large cities is meet by the WHERE CitySummary.LargeCityCount >=2 clause. CROSS APPLY will work fine here.
https://technet.microsoft.com/en-us/library/ms175156(v=sql.105).aspx NEW QUESTION 76You create a table named Products.Sales by running the following Transact-SQL statement:CREATE TABLE Products.Sales (SalesId int IDENTIFY(1,1) PRIMARY KEY,SalesDate DateTime NOT NULL,SalesAmount decimal(18,2) NULL)You add the following data to the table:

SalesID	SalesData	SalesAmount
1	2015-03-05 16:37:23.630	65.00
2	2014-08-25 16:37:23.633	98.00
3	2014-10-15 16:37:23.633	39.00
4	2016-04-06 16:37:23.633	118.00
5	2014-08-29 16:37:23.633	79.00
6	2015-07-17 16:37:23.633	68.00
7	2016-01-03 16:37:23.637	115.00
8	2015-10-23 16:37:23.637	52.00
9	2014-12-07 16:37:23.637	109.00
10	2016-06-15 16:37:23.637	83.00

You are developing a report to display monthly sales data. You need to create a Transact-SQL query to meet the following requirements:- Retrieve a column for the year followed by a column for each month from January through December.- Include the total sales amount for each month.- Aggregate columns by year, month, and then amount.Construct the query using the following guidelines:- Use the MONTH keyword as the interval when using the DATANAME function.- Do not modify the provided IN clause.- Do not surround object names with square brackets.- Do not use implicit joins.- Do not use the DATEPART function.Part of the correct Transact-SQL has been provided in the answer area below. Enter the code in the answer area that resolves the problem and meets the stated goals or requirements. You can add code within the code that has been provided as well as below it.1. SELECT * FROM2. (SELECT YEAR(SalesData)) AS Year, DATENAME (MONTH, SalesDate) AS Month, SalesAmount AS Amount3.4.) AS MonthlySalesData5.6. FOR Month IN (January, February, March, April, May, June, July, August, September, October, November, December))AS MonthNamePivot?? NEW QUESTION 77Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution. After you answer a question in this section. You will NOT be able to return to it. As a result, these questions will not appear in the review screen.You have a database that includes the tables shown in the exhibit:

Customers (Sales)
- custid
- companyname
- contactname
- contacttitle
- address
- city
- region
- postalcode
- country
- phone
- fax

Orders (Sales)
- orderid
- custid
- empid
- orderdate
- requireddate
- shippeddate
- shipperid
- freight
- shipname
- shipaddress
- shipcity
- shipregion
- shippostalcode
- shipcountry

Employees (HR)
- empid
- lastname
- firstname
- title
- titleofcourtesy
- birthdate
- hiredate
- address
- city
- region
- postalcode
- country
- phone
- mgrid

You need to create a Transact-SQL query that returns the following information:- the customer number- the customer contact name-

the date the order was placed, with a name of DateofOrder- a column named Salesperson, formatted with the employee first name, a space, and the employee last name- orders for customers where the employee identifier equals 4The output must be sorted by order date, with the newest orders first. The solution must return only the most recent order for each customer. Solution: You run the following Transact-SQL statement:

```
SELECT c.custid, contactname, MAX(orderdate) AS DateofOrder,
       e.firstname + ' ' + e.lastname AS Salesperson
FROM Sales.Customers AS c
INNER JOIN Sales.Orders AS o ON c.custid = o.custid
INNER JOIN HR.Employees AS e ON o.empid = e.empid
GROUP BY c.custid, contactname, firstname, lastname, o.empid
HAVING o.empid = 4
ORDER BY DateofOrder DESC
```

Does the solution meet the goal? A.   YesB.   No Answer: BExplanation:We should use a WHERE clause, not a HAVING clause. The HAVING clause would refer to aggregate data. NEW QUESTION 78Note: 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 this series. Information and details provided in a question apply only to that question.Multiple processes use the data from a table named Sales and place it in other databases across the organization. Some of the processes are not completely aware of the data types in the Sales table. This leads to data type conversion errors. You need to implement a method that returns a NULL value id data conversion fails instead of throwing an error. What should you implement? A.   the COALESCE functionB.   a viewC.   a table-valued functionD.   the TRY_PARSE functionE.   a stored procedureF.   the ISNULL functionG.   a scalar functionH.   the TRY_CONVERT function Answer: HExplanation:TRY_CONVERT returns a value cast to the specified data type if the cast succeeds; otherwise, returns null.https://docs.microsoft.com/en-us/sql/t-sql/functions/try-convert-transact-sql NEW QUESTION 79You need to create a table named MiscellaneousPayment that meets the following requirements:

Column name
Id
Reason
Amount

Which Transact-SQL statement should you run? A.   CREATE TABLE MiscellaneousPayment (Id uniqueidentifier DEFAULT NEWSEQUENTIALID() PRIMARY KEY,Reason varchar(500),Amount money)B.   CREATE TABLE MiscellaneousPayment (Id int identify(1,1)PRIMARY KEY,Reason nvarchar(500),Amount numeric(19,4))C.   CREATE TABLE MiscellaneousPayment (Id uniqueidentifier DEFAULT NEWSEQUENTIALID() PRIMARY KEY,Reason varchar(500),Amount decimal(19,4))D.   CREATE TABLE MiscellaneousPayment (Id uniqueidentifier DEFAULT NEWID() PRIMARY KEY,Reason nvarchar(500),Amount decimal(19,4)) Answer: D NEW QUESTION 80You have a database that includes the tables shown in the exhibit:
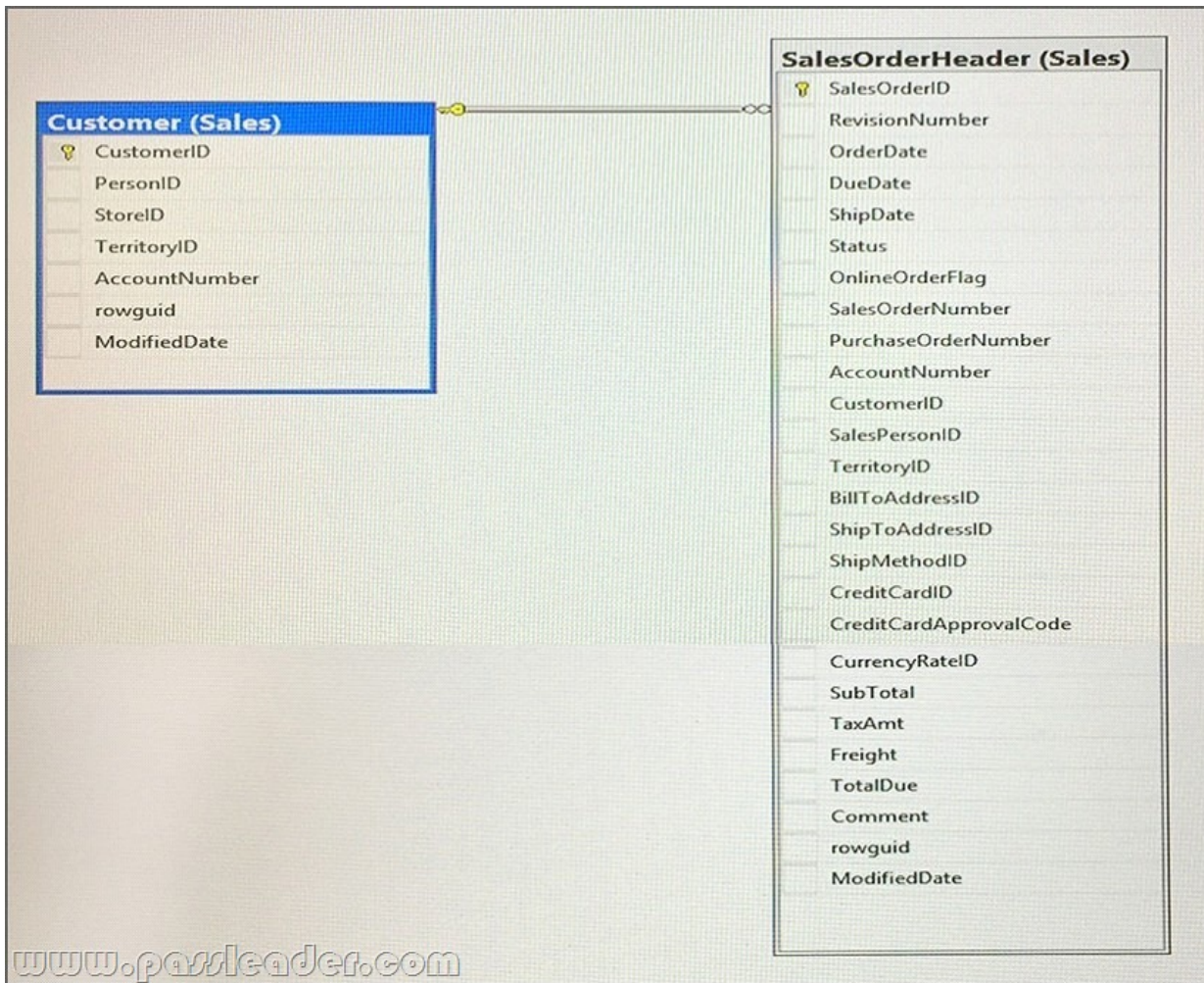
You need to create a list of all customers, the order ID for the last order that the customer placed, and the date that the order was placed. For customers who have not placed orders, you must substitute a zero for the order ID and 01/01/1990 for the date. Which Transact-SQL statement should you run?

A
```
SELECT C.CustomerID, ISNULL(SOH.SalesOrderID, 0) AS OrderID, ISNULL(MAX(OrderDate), '')
FROM Sales.Customer C LEFT OUTER JOIN Sales.SalesOrderHeader SOH
ON C.CustomerID = SOH.CustomerID
GROUP BY C.CustomerID, SOH.SalesOrderID
ORDER BY C.CustomerID
```

B
```
SELECT C.CustomerID, SOH.SalesOrderID, MAX(OrderDate)
FROM Sales.Customer C INNER JOIN Sales.SalesOrderHeader SOH
ON C.CustomerID = SOH.CustomerID
GROUP BY C.CustomerID, SOH.SalesOrderID
ORDER BY C.CustomerID
```

C
```
SELECT C.CustomerID, SOH.SalesOrderID, MAX(OrderDate)
FROM Sales.Customer C CROSS JOIN Sales.SalesOrderHeader SOH
ON C.CustomerID = SOH.CustomerID
GROUP BY C.CustomerID, SOH.SalesOrderID
ORDER BY C.CustomerID
```

D
```
SELECT C.CustomerID, SOH.SalesOrderID, MAX(OrderDate)
FROM Sales.Customer C RIGHT OUTER JOIN Sales.SalesOrderHeader SOH
ON C.CustomerID = SOH.CustomerID
GROUP BY C.CustomerID, SOH.SalesOrderID
ORDER BY C.CustomerID
```

A.   Option AB.   Option BC.   Option CD.   Option D Answer: AExplanation:- ISNULL Syntax: ISNULL (check_expression , replacement_value) author:"Luxemburg, Rosa"- The ISNULL function replaces NULL with the specified replacement value. The

value of check_expression is returned if it is not NULL; otherwise, replacement_value is returned after it is implicitly converted to the type of check_expression.https://msdn.microsoft.com/en-us/library/ms184325.aspx NEW QUESTION 81Note: 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 this series. Information and details provided in a question apply to that question.You have a database for a banking system. The database has two tables named tblDepositAcct and tblLoanAcct that store deposit and loan accounts, respectively. Both tables contain the following columns:

Column name	Data type
CustNo	int
AcctNo	int
ProdCode	varchar(3)

You need to determine the total number of customers who have only loan accounts. Which Transact-SQL statement should you run? A.   SELECT COUNT(*)FROM (SELECT AcctNoFROM tblDepositAcctINTERSECTSELECT AcctNoFROM tblLoanAcct) RB.   SELECT COUNT(*)FROM (SELECT CustNoFROM tblDepositAcctUNIONSELECT CustNoFROM tblLoanAcct) RC.   SELECT COUNT(*)FROM (SELECT CustNoFROM tblDepositAcctUNION ALLSELECT CustNoFROM tblLoanAcct) RD.   SELECT COUNT (DISTINCTD.CustNo)FROM tblDepositAcct D, tblLoanAcct LWHERED.CustNo = L.CustNoE.   SELECT COUNT(DISTINCT L.CustNo)FROM tblDepositAcct DRIGHT JOIN tblLoanAcct L OND.CustNo = L.CustNoWHERED.CustNo IS NULLF.   SELECT COUNT(*)FROM (SELECT CustNoFROM tblDepositAcctEXCEPTSELECT CustNoFROM tblLoanAcct) RG.   SELECT COUNT (DISTINCT COALESCE(D.CustNo, L.CustNo))FROM tblDepositAcct DFULL JOIN tblLoanAcct L ON D.CustNo = L.CustNoWHERED.CustNo IS NULL OR L.CustNo IS NULLH.   SELECT COUNT(*)FROM tblDepositAcct DFULL JOIN tblLoanAcct L OND.CustNo = L.CustNo Answer: EExplanation:The RIGHT JOIN keyword returns all records from the right table (table2), and the matched records from the left table (table1). The result is NULL from the left side, when there is no match.https://www.w3schools.com/sql/sql_join_right.asp NEW QUESTION 82Note: 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 on this series.You have a database that tracks orders and deliveries for customers in North America. System versioning is enabled for all tables. The database contains the Sales.Customers, Application.Cities, and Sales.CustomerCategories tables. Details for the Sales.Customers table are shown in the following table:

Column	Data type	Notes
CustomerId	int	primary key
CustomerCategoryId	int	foreign key to the Sales.CustomerCategories table
PostalCityID	int	foreign key to the Application.Cities table
DeliveryCityID	int	foreign key to the Application.Cities table
AccountOpenedDate	datetime	does not allow values
StandardDiscountPercentage	int	does not allow values
CreditLimit	decimal(18,2)	null values are permitted
IsOnCreditHold	bit	does not allow values
DeliveryLocation	geography	does not allow values
PhoneNumber	nvarchar(20)	does not allow values
ValidFrom	datetime2(7)	does not allow values, GENERATED ALWAYS AS ROW START
ValidTo	datetime2(7)	does not allow values, GENERATED ALWAYS AS ROW END

Details for the Application.Cities table are shown in the following table:

Column	Data type	Note
CityID	int	primary key
LatestRecordedPopulation	bigint	null values are permitted

Details for the Sales.CustomerCategories table are shown in the following table:

Column	Data type	
CustomerCategoryID	int	primary key
CustomerCategoryName	nvarchar(50)	does not allow

You discover an application bug that impacts customer data for records created on or after January 1, 2014. In order to fix the data impacted by the bug, application programmers require a report that contains customer data as it existed on December 31, 2013. You need to provide the query for the report. Which Transact-SQL statement should you use?

A
```
DECLARE @sdate DATETIME, @edate DATETIME
SET @sdate = DATEFROMPARTS (2013, 12, 31)
set @edate = DATEADD(d, 1, @sdate)
SELECT * FROM Sales.Customers FOR SYSTEM_TIME A
WHERE ValidFrom > @sdate AND ValidTo < @edate
```

B
```
DECLARE @sdate DATETIME, @edate DATETIME
SET @sdate = DATEFROMPARTS (2013, 12, 31)
set @edate = DATEADD(d, -1, @sdate)
SELECT * FROM Sales.Customers FOR SYSTEM_TIME B
```

C
```
DECLARE @date DATE
SET @date = DATEFROMPARTS (2013, 12, 31)
SELECT * FROM Sales.Customers FOR SYSTEM_TIME A
```

D
```
DECLARE @date DATE
SET @date = DATEFROMPARTS (2013, 12, 31)
SELECT * FROM Sales.Customers WHERE @date BETWEE
```

A.   Option AB.   Option BC.   Option CD.   Option D Answer: DExplanation:The datetime datetype defines a date that is combined with a time of day with fractional seconds that is based on a 24-hour clock. The DATEFROMPARTS function returns a date value for the specified year, month, and day. NEW QUESTION 83??   Download the newest PassLeader 70-761 dumps from passleader.com now! 100% Pass Guarantee! 70-761 PDF dumps & 70-761 VCE dumps: https://www.passleader.com/70-761.html (98 Q&As) (New Questions Are 100% Available and Wrong Answers Have Been Corrected! Free VCE simulator!) P.S. Free 70-761 dumps download from Google Drive: https://drive.google.com/open?id=0B-ob6L_QjGLpaEZzRVFnOE9OenM Free 70-762 dumps download from Google Drive: https://drive.google.com/open?id=0B-ob6L_QjGLpN3RVQ25sVUM5dkU Free 70-764 dumps download from Google Drive: https://drive.google.com/open?id=0B-ob6L_QjGLpN3N6eHJ6Z2EzZWc Free 70-765 dumps download from Google Drive: https://drive.google.com/open?id=0B-ob6L_QjGLpZHlHSG5KM09xUms Free 70-767 dumps download from Google Drive: https://drive.google.com/open?id=0B-ob6L_QjGLpcXZXWUl4dHhIUVk Free 70-768 dumps download from Google Drive: https://drive.google.com/open?id=0B-ob6L_QjGLpeXAxaUJkWEZnVlU