



70-761^{Q&As}

Querying Data with Transact-SQL

Pass Microsoft 70-761 Exam with 100% Guarantee

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

<https://www.lead4pass.com/70-761.html>

100% Passing Guarantee
100% Money Back Assurance

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

-  **Instant Download** After Purchase
-  **100% Money Back** Guarantee
-  **365 Days** Free Update
-  **800,000+** Satisfied Customers





QUESTION 1

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

You create a table by running the following Transact-SQL statement:

```
CREATE TABLE Customers (  
    CustomerID int NOT NULL PRIMARY KEY CLUSTERED,  
    FirstName nvarchar(100) NOT NULL,  
    LastName nvarchar(100) NOT NULL,  
    TaxIdNumber varchar(20) NOT NULL,  
    Address nvarchar(1024) NOT NULL,  
    AnnualRevenue decimal(19,2) NOT NULL,  
    DateCreated datetime2(2) NOT NULL,  
    ValidFrom datetime2(2) GENERATED ALWAYS AS ROW START NOT NULL,  
    ValidTo datetime2(2) GENERATED ALWAYS AS ROW END NOT NULL,  
    PERIOD FOR SYSTEM_TIME(ValidFrom, ValidTo)  
)  
WITH (SYSTEM_VERSIONING = ON (HISTORY_TABLE = CustomersHistory))
```

You are developing a report that displays customer information. The report must contain a grand total column.

You need to write a query that returns the data for the report.

Which Transact-SQL statement should you run?



- A. `SELECT CustomerID, FirstName, LastName, TaxIdNumber, Address, AnnualRevenue, DateCreated
FROM Customers
GROUP BY GROUPING SETS((FirstName, LastName), (Address), (CustomerID, AnnualRevenue), (CustomerID), ())
ORDER BY CustomerID, FirstName, LastName, Address, AnnualRevenue`
- B. `SELECT FirstName, LastName, Address
FROM Customers
FOR SYSTEM_TIME ALL ORDER BY ValidFrom`
- C. `SELECT c.CustomerID, c.FirstName, c.LastName, c.Address, c.ValidFrom, c.ValidTo
FROM Customers AS c
ORDER BY c.CustomerID
FOR JSON AUTO, ROOT('Customers')`
- D. `SELECT * FROM (SELECT CustomerID, FirstName, LastName, Address, AnnualRevenue, DateCreated
FROM Customers) AS Customers PIVOT(AVG(AnnualRevenue)
FOR DateCreated IN([2014])) AS PivotCustomers
ORDER BY LastName, FirstName`
- E. `SELECT CustomerID, AVG(AnnualRevenue)
AS AverageAnnualRevenue, FirstName, LastName, Address, DateCreated
FROM Customers WHERE YEAR(DateCreated) >= 2014
GROUP BY CustomerID, FirstName, LastName, Address, DateCreated`
- F. `SELECT c.CustomerID, c.FirstName, c.LastName, c.Address, c.ValidFrom, c.ValidTo
FROM Customers AS c ORDER BY c.CustomerID
FOR XML PATH ('CustomerData'), root ('Customers')`
- G. `SELECT CustomerID, FirstName, LastName, TaxIdNumber, Address, ValidFrom, ValidTo
FROM Customers FOR SYSTEM_TIME
BETWEEN '2014-01-01 00:00:00.000000' AND '2015-01-01 00:00:00.000000'`
- H. `SELECT CustomerID, FirstName, LastName, TaxIdNumber, Address, ValidFrom, ValidTo
FROM Customers
WHERE DateCreated
BETWEEN '20140101' AND '20141231'`

A. B. C. D. E. F. G. H.

Correct Answer: E

Calculate aggregate column through AVG function and GROUP BY clause.

QUESTION 2

DRAG DROP You have a database that stored information about servers and application errors. The database contains the following tables. Servers

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

Errors



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

You are building a webpage that shows the three most common errors for each server.

You need to return the data for the webpage.

How should you complete the Transact-SQL statement? To answer, drag the appropriate 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:

Transact-SQL segments

- svr.ServerID
- errs.ServerID
- INNER JOIN
- CROSS APPLY
- WITHIN GROUP
- WHERE ServerID = svr.ServerID
- WHERE ServerID = errs.ErrorID

Answer Area

```

SELECT      Transact-SQL segment , errs.LogMessage
FROM Servers AS svr
Transact-SQL segment
(
    SELECT TOP 3 LogMessage
    FROM Errors
    Transact-SQL segment
    ORDER BY Occurrences
) AS errs
  
```

Correct Answer:



Transact-SQL segments

Answer Area

-
- errs.ServerID
- INNER JOIN
-
- WITHIN GROUP
-
- WHERE ServerID = errs.ErrorID

```
SELECT svr.ServerID , errs.LogMessage
FROM Servers AS svr
CROSS APPLY
(
    SELECT TOP 3 LogMessage
    FROM Errors
    WHERE ServerID = svr.ServerID
    ORDER BY Occurrences
) AS errs
```

QUESTION 3

HOTSPOT

You need to develop a function that returns a list of courses grouped by the total number of students in a course.

The function must list only courses that have more than a specific number of students. The specific number of students is defined as an input variable for the function.

How should you complete the function? To answer, select the appropriate Transact-SQL segments in the answer area.

NOTE: Each correct selection is worth one point.

Hot Area:



Answer Area

```

CREATE FUNCTION CoursesWithMoreThan (@totalStudents INT)
RETURNS 

|                                    |   |
|------------------------------------|---|
|                                    | ▼ |
| TABLE                              |   |
| HAVING                             |   |
| WHERE                              |   |
| INT                                |   |
| SUM(cp.NumStudents) AS NumStudents |   |
| SUM(cp.NumStudents)                |   |


AS
RETRUN 

|                                    |   |
|------------------------------------|---|
|                                    | ▼ |
| TABLE                              |   |
| HAVING                             |   |
| WHERE                              |   |
| INT                                |   |
| SUM(cp.NumStudents) AS NumStudents |   |
| SUM(cp.NumStudents)                |   |


SELECT c.Course, 

|                                    |   |
|------------------------------------|---|
|                                    | ▼ |
| TABLE                              |   |
| HAVING                             |   |
| WHERE                              |   |
| INT                                |   |
| SUM(cp.NumStudents) AS NumStudents |   |
| SUM(cp.NumStudents)                |   |


FROM dbo.Courses
INNER JOIN dbo.CourseStudents cp ON c.CourseID = cp.CourseID


|                                    |   |
|------------------------------------|---|
|                                    | ▼ |
| TABLE                              |   |
| HAVING                             |   |
| WHERE                              |   |
| INT                                |   |
| SUM(cp.NumStudents) AS NumStudents |   |
| SUM(cp.NumStudents)                |   |

 SUM(cp.NumStudents) > @totalStudents

```

Correct Answer:



Answer Area

CREATE FUNCTION CoursesWithMoreThan (@totalStudents INT)
 RETURNS

	▼
TABLE	
HAVING	
WHERE	
INT	
SUM(cp.NumStudents) AS NumStudents	
SUM(cp.NumStudents)	

 AS
 RETURN

	▼
TABLE	
HAVING	
WHERE	
INT	
SUM(cp.NumStudents) AS NumStudents	
SUM(cp.NumStudents)	

 SELECT c.Course,

	▼
TABLE	
HAVING	
WHERE	
INT	
SUM(cp.NumStudents) AS NumStudents	
SUM(cp.NumStudents)	

 FROM dbo.Courses
 INNER JOIN dbo.CourseStudents cp ON c.CourseID = cp.CourseID

	▼
TABLE	
HAVING	
WHERE	
INT	
SUM(cp.NumStudents) AS NumStudents	
SUM(cp.NumStudents)	

 SUM(cp.NumStudents) > @totalStudents

QUESTION 4

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 this series.

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

You create a table named Customers. Data stored in the table must be exchanged between web pages and web servers by using AJAX calls that use REST endpoint.

You need to return all customer information by using a data exchange format that is text-based and lightweight.

Which Transact-SQL statement should you run?



- A. `SELECT CustomerID, FirstName, LastName, TaxIdNumber, Address, AnnualRevenue, DateCreated
FROM Customers
GROUP BY GROUPING SETS(FirstName, LastName), (Address), (CustomerID, AnnualRevenue), (CustomerID), ()
ORDER BY CustomerID, FirstName, LastName, Address, AnnualRevenue`
- B. `SELECT FirstName, LastName, Address
FROM Customers
FOR SYSTEM_TIME ALL ORDER BY ValidFrom`
- C. `SELECT c.CustomerID, c.FirstName, c.LastName, c.Address, c.ValidFrom, c.ValidTo
FROM Customers AS c
ORDER BY c.CustomerID
FOR JSON AUTO, ROOT('Customers')`
- D. `SELECT * FROM (SELECT CustomerID, FirstName, LastName, Address, AnnualRevenue, DateCreated
FROM Customers) AS Customers PIVOT(AVG(AnnualRevenue)
FOR DateCreated IN([2014])) AS PivotCustomers
ORDER BY LastName, FirstName`
- E. `SELECT CustomerID, AVG(AnnualRevenue)
AS AverageAnnualRevenue, FirstName, LastName, Address, DateCreated
FROM Customers WHERE YEAR(DateCreated) >= 2014
GROUP BY CustomerID, FirstName, LastName, Address, DateCreated`
- F. `SELECT c.CustomerID, c.FirstName, c.LastName, c.Address, c.ValidFrom, c.ValidTo
FROM Customers AS c ORDER BY c.CustomerID
FOR XML PATH ('CustomerData'), root ('Customers')`
- G. `SELECT CustomerID, FirstName, LastName, TaxIdNumber, Address, ValidFrom, ValidTo
FROM Customers FOR SYSTEM_TIME
BETWEEN '2014-01-01 00:00:00.000000' AND '2015-01-01 00:00:00.000000'`
- H. `SELECT CustomerID, FirstName, LastName, TaxIdNumber, Address, ValidFrom, ValidTo
FROM Customers
WHERE DateCreated
BETWEEN '20140101' AND '20141231'`

A. B. C. D. E. F. G. H.

Correct Answer: C

JSON can be used to pass AJAX updates between the client and the server.

Export data from SQL Server as JSON, or format query results as JSON, by adding the FOR JSON clause to a SELECT statement.

When you use the FOR JSON clause, you can specify the structure of the output explicitly, or let the structure of the SELECT statement determine the output.

References: <https://msdn.microsoft.com/en-us/library/dn921882.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 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	Primary key column	Description
CustNo	int	No	This column uniquely identifies a customer in the bank. A customer may have both deposit and loan accounts.
AcctNo	int	Yes	This column uniquely identifies a customer in the bank.
ProdCode	varchar(3)	No	This column identifies the product type of an account. A customer may have multiple accounts for the same product type.

You need to determine the total number of deposit and loan accounts. Which Transact-SQL statement should you run?

```
SELECT COUNT(*)  
FROM (SELECT AcctNo  
FROM tblDepositAcct  
INTERSECT  
SELECT AcctNo  
FROM tblLoanAcct) R
```



- A. `SELECT COUNT(*)
FROM (SELECT AcctNo
FROM tblDepositAcct
INTERSECT
SELECT AcctNo
FROM tblLoanAcct) R`
- B. `SELECT COUNT(*)
FROM (SELECT CustNo
FROM tblDepositAcct
UNION
SELECT CustNo
FROM tblLoanAcct) R`
- C. `SELECT COUNT(*)
FROM (SELECT CustNo
FROMtblDepositAcct
UNION ALL
SELECT CustNo
FROM tblLoanAcct) R`
- D. `SELECT COUNT (DISTINCT D.CustNo)
FROM tblDepositAcct D, tblLoanAcct L
WHERE D.CustNo = L.CustNo`
- E. `SELECT COUNT(DISTINCT L.CustNo)
FROM tblDepositAcct D
RIGHT JOIN tblLoanAcct L ON D.CustNo =L.CustNo
WHERE D.CustNo IS NULL`
- F. `SELECT COUNT(*)
FROM (SELECT CustNo
FROM tblDepositAcct
EXCEPT
SELECT CustNo
FROM tblLoanAcct) R`
- G. `SELECT COUNT (DISTINCT COALESCE(D.CustNo, L.CustNo))
FROM tblDepositAcct D
FULL JOIN tblLoanAcct L ON D.CustNo =L.CustNo
WHERE D.CustNo IS NULL OR L.CustNo IS NULL`
- H. `SELECT COUNT(*)
FROM tblDepositAcct D
FULL JOIN tblLoanAcct L ON D.CustNo = L.CustNo`



A. B. C. D. E. F. G. H.

Correct Answer: C

Would list the customers with duplicates, which would equal the number of accounts. Incorrect Answers:

A: INTERSECT returns distinct rows that are output by both the left and right input queries operator.

B: Would list the customers without duplicates.

D: Number of customers.

F: EXCEPT returns distinct rows from the left input query that aren't output by the right input query.

References: <https://msdn.microsoft.com/en-us/library/ms180026.aspx>

[Latest 70-761 Dumps](#)

[70-761 PDF Dumps](#)

[70-761 Exam Questions](#)



To Read the [Whole Q&As](#), please purchase the [Complete Version](#) from [Our website](#).

Try our product !

100% Guaranteed Success

100% Money Back Guarantee

365 Days Free Update

Instant Download After Purchase

24x7 Customer Support

Average 99.9% Success Rate

More than 800,000 Satisfied Customers Worldwide

Multi-Platform capabilities - [Windows](#), [Mac](#), [Android](#), [iPhone](#), [iPod](#), [iPad](#), [Kindle](#)

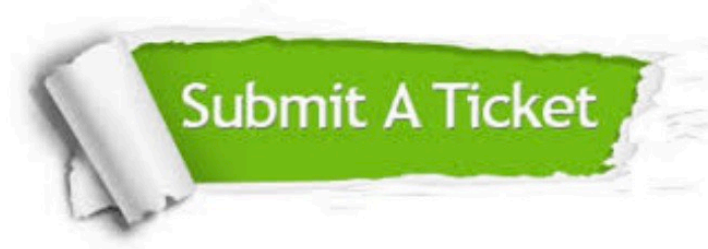
We provide exam PDF and VCE of Cisco, Microsoft, IBM, CompTIA, Oracle and other IT Certifications. You can view Vendor list of All Certification Exams offered:

<https://www.lead4pass.com/allproducts>

Need Help

Please provide as much detail as possible so we can best assist you.

To update a previously submitted ticket:



 <p>One Year Free Update Free update is available within One Year after your purchase. After One Year, you will get 50% discounts for updating. And we are proud to boast a 24/7 efficient Customer Support system via Email.</p>	 <p>Money Back Guarantee To ensure that you are spending on quality products, we provide 100% money back guarantee for 30 days from the date of purchase.</p>	 <p>Security & Privacy We respect customer privacy. We use McAfee's security service to provide you with utmost security for your personal information & peace of mind.</p>
---	---	--

Any charges made through this site will appear as Global Simulators Limited.

All trademarks are the property of their respective owners.

Copyright © lead4pass, All Rights Reserved.