



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.passapply.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 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 tracks orders and deliveries for customers in North America. The database contains the following tables:

Sales.Customers

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 new values
StandardDiscountPercentage	int	does not allow new values
CreditLimit	decimal(18,2)	null values are permitted
IsOnCreditHold	bit	does not allow new values
DeliveryLocation	geography	does not allow new values
PhoneNumber	nvarchar(20)	does not allow new values

Application.Cities

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

Sales.CustomerCategories

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

Your company is developing a new social application that connects customers to each other based on the distance



between their delivery locations.

You need to write a query that returns the nearest customer.

Solution: You run the following Transact-SQL statement:

```
SELECT TOP 1 B.CustomerID, A.DeliveryLocation.STDistance(B.DeliveryLocation) AS Dist
FROM Sales.Customers AS A
CROSS JOIN Sales.Customers AS B
WHERE A.CustomerID = @custID AND A.CustomerID <> B.CustomerID
ORDER BY Dist
```

The variable @custID is set to a valid customer. Does the solution meet the goal?

A. Yes

B. No

Correct Answer: B

QUESTION 2

DRAG DROP

Note: 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 in this series.

Start of repeated scenario

You have a database that contains the tables shown in the exhibit. (Click the Exhibit button.)

SalesSummary			
Column Name	Data Type	Allow Nulls	
SalesSummaryKey	int	<input type="checkbox"/>	
SalesYear	smallint	<input type="checkbox"/>	
SalesQuarter	smallint	<input type="checkbox"/>	
SalesMonth	smallint	<input type="checkbox"/>	
SalesDate	date	<input type="checkbox"/>	
ProductCode	char(12)	<input type="checkbox"/>	
CustomerCode	char(6)	<input type="checkbox"/>	
EmployeeCode	char(6)	<input type="checkbox"/>	
RegionCode	char(2)	<input checked="" type="checkbox"/>	
SalesAmount	money	<input type="checkbox"/>	

Employee			
Column Name	Data Type	Allow Nulls	
EmployeeID	smallint	<input type="checkbox"/>	
EmployeeCode	char(6)	<input type="checkbox"/>	
FirstName	varchar(30)	<input checked="" type="checkbox"/>	
MiddleName	varchar(30)	<input checked="" type="checkbox"/>	
LastName	varchar(40)	<input type="checkbox"/>	
Title	varchar(50)	<input type="checkbox"/>	
ManagerID	smallint	<input checked="" type="checkbox"/>	
		<input type="checkbox"/>	

You review the Employee table and make the following observations:



Every record has a value in the ManagerID except for the Chief Executive Officer (CEO).

The FirstName and MiddleName columns contain null values for some records.

The valid values for the Title column are Sales Representative manager, and CEO.

You review the SalesSummary table and make the following observations:

The ProductCode column contains two parts: The first five digits represent a product code, and the last seven digits represent the unit price. The unit price uses the following pattern: #####.##.

You observe that for many records, the unit price portion of the ProductCode column contains values.

The RegionCode column contains NULL for some records.

Sales data is only recorded for sales representatives.

You are developing a series of reports and procedures to support the business. Details for each report or procedure follow. Sales Summary report: This report aggregates data by year and quarter. The report must resemble the following table.

SalesYear	SalesQuarter	YearSalesAmount	QuarterSalesAmount
2015	1	2000.00	1000.00
2015	2	2000.00	500.00
2015	3	2000.00	250.00
2015	4	2000.00	250.00
2016	1	3500.00	500.00
2016	2	3500.00	1000.00

Sales Manager report: This report lists each sales manager and the total sales amount for all employees that report to the sales manager.

Sales by Region report: This report lists the total sales amount by employee and by region. The report must include the following columns: EmployeeCode, MiddleName, LastName, RegionCode, and SalesAmount. If MiddleName is NULL, FirstName must be displayed. If both FirstName and MiddleName have null values, the word Unknown must be displayed/ If RegionCode is NULL, the word Unknown must be displayed.

Report1: This report joins data from SalesSummary with the Employee table and other tables. You plan to create an object to support Report1. The object has the following requirements:

be joinable with the SELECT statement that supplies data for the report

can be used multiple times with the SELECT statement for the report

be usable only with the SELECT statement for the report

not be saved as a permanent object

Report2: This report joins data from SalesSummary with the Employee table and other tables.

You plan to create an object to support Report1. The object has the following requirements:

be joinable with the SELECT statement that supplies data for the report



can be used multiple times for this report and other reports

accept parameters

be saved as a permanent object

Sales Hierarchy report: This report aggregates rows, creates subtotal rows, and super-aggregates rows over the SalesAmount column in a single result-set. The report uses SaleYear, SaleQuarter, and SaleMonth as a hierarchy. The result

set must not contain a grand total or cross-tabulation aggregate rows.

Current Price Stored Procedure: This stored procedure must return the unit price for a product when a product code is supplied. The unit price must include a dollar sign at the beginning. In addition, the unit price must contain a comma every

three digits to the left of the decimal point, and must display two digits to the left of the decimal point. The stored procedure must not throw errors, even if the product code contains invalid data.

End of Repeated Scenario

You are creating the queries for Report1 and Report2.

You need to create the objects necessary to support the queries.

Which object should you use to join the SalesSummary table with the other tables that each report uses? To answer, drag the appropriate objects to the correct reports. each object 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.

Select and Place:

Objects

-
-
-
-
-
-
-
-

Answer area

Report	Object
Report1	Object
Report2	Object

Correct Answer:



Objects	Answer area
<input type="text"/>	Report
<input type="text"/>	Object
<input type="text" value="indexed view"/>	Report1 <input type="text" value="common table expression (CTE)"/>
<input type="text" value="subquery"/>	Report2 <input type="text" value="view"/>
<input type="text" value="scalar function"/>	
<input type="text" value="table-valued function"/>	
<input type="text" value="stored procedure"/>	
<input type="text" value="derived table"/>	
<input type="text"/>	

Box 1: common table expression (CTE)

A common table expression (CTE) can be thought of as a temporary result set that is defined within the execution scope of a single SELECT, INSERT, UPDATE, DELETE, or CREATE VIEW statement. A CTE is similar to a derived table in that it is not stored as an object and lasts only for the duration of the query. Unlike a derived table, a CTE can be self-referencing and can be referenced multiple times in the same query.

A CTE can be used to:

Create a recursive query. For more information, see Recursive Queries Using Common Table Expressions.

Substitute for a view when the general use of a view is not required; that is, you do not have to store the definition in metadata.

Enable grouping by a column that is derived from a scalar subselect, or a function that is either not deterministic or has external access.

Reference the resulting table multiple times in the same statement.

From Scenario: Report1: This report joins data from SalesSummary with the Employee table and other tables. You plan to create an object to support Report1. The object has the following requirements:

be joinable with the SELECT statement that supplies data for the report

can be used multiple times with the SELECT statement for the report

be usable only with the SELECT statement for the report

not be saved as a permanent object

Box 2: view



From scenario: Report2: This report joins data from SalesSummary with the Employee table and other tables.

You plan to create an object to support Report1. The object has the following requirements:

be joinable with the SELECT statement that supplies data for the report can be used multiple times for this report and other reports accept parameters be saved as a permanent object

References: [https://technet.microsoft.com/en-us/library/ms190766\(v=sql.105\).aspx](https://technet.microsoft.com/en-us/library/ms190766(v=sql.105).aspx)

QUESTION 3

DRAG DROP

Note: 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 in this series.

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

Task level is defined using the following rules:

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.

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.

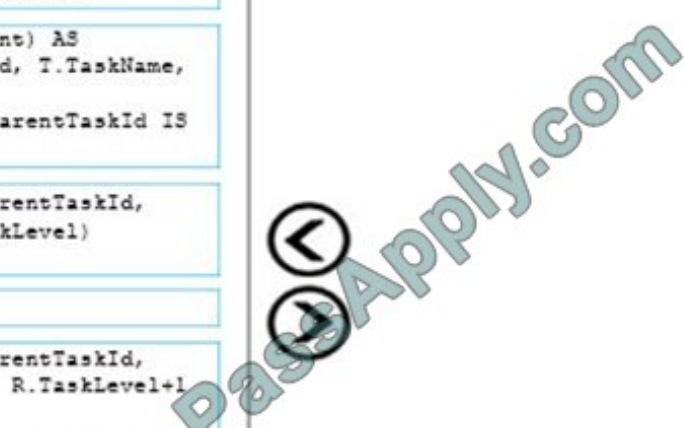



Task category	Task level definition
Tasks that have no parent task	[Task Level] = 0
Tasks that have a parent task	[Task Level] = [Parent Task's Level] + 1

You need to determine the task level for each task in the hierarchy.



Which five Transact-SQL segments should you use to develop the solution? To answer, move the appropriate Transact-SQL segments from the list of Transact-SQL segments to the answer area and arrange them in the correct order.

Select and Place:

Transact-SQL segments	Answer Area
<pre>) SELECT * FROM TaskWithLevel</pre>	 <div style="display: flex; justify-content: space-around; align-items: center; margin-top: 20px;"> <div style="text-align: center;">   </div> <div style="text-align: center;">   </div> </div>
<pre>SELECT CAST(NULL AS int) AS ParentTaskId, T.TaskId, T.TaskName, 0 AS TaskLevel FROM Task T WHERE T.ParentTaskId IS NULL</pre>	
<pre>With TaskWithLevel (ParentTaskId, TaskId, TaskName, TaskLevel) As {</pre>	
<pre>UNION</pre>	
<pre>SELECT R.TaskId AS ParentTaskId, T.TaskId, T.TaskName, R.TaskLevel+1 AS TaskLevel FROM Task T INNER JOIN TaskWithLevel R ON T.ParentTaskId = R.TaskId</pre>	
<pre>SELECT T.TaskId AS ParentTaskId, CAST(null AS int) AS TaskId, T.TaskName, 0 AS TaskLevel FROM Task T WHERE T.ParentTaskId IS NULL</pre>	
<pre>UNION ALL</pre>	

Correct Answer:



Transact-SQL segments

```

)
SELECT * FROM TaskWithLevel

SELECT CAST(NULL AS int) AS
ParentTaskId, T.TaskId, T.TaskName,
0 AS TaskLevel
FROM Task T WHERE T.ParentTaskId IS
NULL

With TaskWithLevel (ParentTaskId,
TaskId, TaskName, TaskLevel)
As (

UNION

SELECT R.TaskId AS ParentTaskId,
T.TaskId, T.TaskName, R.TaskLevel+1
AS TaskLevel
FROM Task T INNER JOIN TaskWithLevel
R ON T.ParentTaskId = R.TaskId

SELECT T.TaskId AS ParentTaskId,
CAST(null AS int) AS TaskId,
T.TaskName, 0 AS TaskLevel
FROM Task T WHERE T.ParentTaskId IS
NULL

UNION ALL

```

Answer Area

```

SELECT CAST(NULL AS int) AS
ParentTaskId, T.TaskId, T.TaskName,
0 AS TaskLevel
FROM Task T WHERE T.ParentTaskId IS
NULL

UNION

SELECT R.TaskId AS ParentTaskId,
T.TaskId, T.TaskName, R.TaskLevel+1
AS TaskLevel
FROM Task T INNER JOIN TaskWithLevel
R ON T.ParentTaskId = R.TaskId

With TaskWithLevel (ParentTaskId,
TaskId, TaskName, TaskLevel)
As (

)

SELECT * FROM TaskWithLevel

```

Box 1: SELECT CAST (NULL AS INT) AS ParentTaskID, etc.

This statement selects all tasks with task level 0.

The ParentTaskID could be null so we should use CAST (NULL AS INT) AS ParentTaskID.

Box 2: UNION

We should use UNION and not UNION ALL as we do not want duplicate rows.

UNION specifies that multiple result sets are to be combined and returned as a single result set.

Incorrect Answers:

Not UNION ALL: ALL incorporates all rows into the results. This includes duplicates. If not specified, duplicate rows are removed.

Box 3, Box 4, Box 5:

These statements select all tasks with task level >0.

References:

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

QUESTION 4

DRAG DROP



Note: 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	Notes
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	Notes
CustomerCategoryID	int	primary key
CustomerCategoryName	nvarchar(50)	does not allow null values

You are creating a report to show when the first customer account was opened in each city. The report contains a line chart with the following characteristics:

The chart contains a data point for each city, with lines connecting the points.

The X axis contains the position that the city occupies relative to other cities.



The Y axis contains the date that the first account in any city was opened. An example chart is shown below for five cities:



During a sales promotion, customers from various cities open new accounts on the same date. You need to write a query that returns the data for the chart.

How should you complete the Transact-SQL statement? To answer, drag the appropriate Transact-SQL segments to the correct locations. 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

-
-
-
-
-
-
-
-

Answer Area

```
SELECT
    CityID,
    MIN(AccountOpenedDate),
    
    
FROM Application.Cities
INNER JOIN Sales.Customers ON CityID = PostalCityID

ORDER BY MIN(AccountOpenedDate) DESC
```

Correct Answer:

Transact-SQL segments

-
-
-
-
-
-
-
-

Answer Area

```
SELECT
    CityID,
    MIN(AccountOpenedDate),
    
    
FROM Application.Cities
INNER JOIN Sales.Customers ON CityID = PostalCityID

ORDER BY MIN(AccountOpenedDate) DESC
```

Box 1: RANK() OVER

RANK returns the rank of each row within the partition of a result set. The rank of a row is one plus the number of ranks that come before the row in question.

ROW_NUMBER and RANK are similar. ROW_NUMBER numbers all rows sequentially (for example 1, 2, 3, 4, 5).



Incorrect Answers:

DENSE_RANK returns the rank of rows within the partition of a result set, without any gaps in the ranking. The rank of a row is one plus the number of distinct ranks that come before the row in question.

Box 2: (PARTITION BY CityID ORDER BY MIN(AccountOpenedDate) DESC)

Syntax for RANK: RANK () OVER ([partition_by_clause] order_by_clause)

Box 3: GROUP BY CityID

References: <https://msdn.microsoft.com/en-us/library/ms176102.aspx>

QUESTION 5

DRAG DROP

Note: 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 in this series.

You are developing a database to track customer orders. The database contains the following tables: Sales.Customers, Sales.Orders, and Sales.OrderLines. The following table describes the columns in Sales.Customers.

Column name	Data type	Constraints
CustomerID	int	primary key
CustomerName	nvarchar(100)	does not allow null values
PhoneNumber	nvarchar(20)	does not allow null values
AccountOpenedDate	date	does not allow null values
StandardDiscountPercentage	decimal(18,3)	does not allow null values
CreditLimit	decimal(18,2)	null values are permitted
IsOnCreditHold	bit	does not allow null values
DeliveryLocation	geography	does not allow null values
PhoneNumber	nvarchar(20)	does not allow null values

The following table describes the columns in Sales.Orders.

Column name	Data type	Constraints
OrderID	int	primary key
CustomerID	int	foreign key to the Sales.Customers table
OrderDate	date	does not allow null values

The following table describes the columns in Sales.OrderLines.



Column name	Data type	Constraints
OrderLineID	int	primary key
OrderID	int	foreign key to the Sales.Orders table
Quantity	int	does not allow null values
UnitPrice	decimal(18,2)	null values are permitted
TaxRate	decimal(18,3)	does not allow null values

You need to create a stored procedure that inserts data into the Customers table. The stored procedure must meet the following requirements: Data changes occur as a single unit of work. Data modifications that are successful are committed and a value of 0 is returned. Data modifications that are unsuccessful are rolled back. The exception severity level is set to 16 and a value of -1 is returned. The stored procedure uses a built-in scalar function to evaluate the current condition of data modifications. The entire unit of work is terminated and rolled back if a run-time error occurs during execution of the stored procedure.

How should complete the stored procedure definition? To answer, drag the appropriate Transact-SQL segments to the correct targets. 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

- RAISERROR
- THROW
- XACT_ABORT
- XACT_STATE
- @@TRANCOUNT
- ROLLBACK
- COMMIT
- END

Answer Area

```

CREATE PROCEDURE Sales.InsertCustomer
    @CustomerName nvarchar(100),
    @PhoneNumber nvarchar(20),
    @AccountOpenedDate date,
    @StandardDiscountPercentage decimal(18,3),
    @CreditLimit decimal(18,2),
    @IsCreditOnHold bit,
    @DeliveryLongitude nvarchar(50),
    @DeliveryLatitude nvarchar(50)
AS
BEGIN
    SET NOCOUNT ON
    SET  ON

    BEGIN TRY
        BEGIN TRANSACTION
        INSERT INTO Sales.Customers (CustomerName, PhoneNumber, AccountOpenedDate,
            StandardDiscountPercentage, CreditLimit, IsOnCreditHold, DeliveryLocation)
        VALUES
            (@CustomerName, @PhoneNumber, @AccountOpenedDate, @StandardDiscountPercentage,
            @CreditLimit, @IsCreditOnHold, geography::Point(ISNULL(@DeliveryLongitude, ''),
            ISNULL(@DeliveryLatitude, ''), 4326))
         TRANSACTION
    END TRY
    BEGIN CATCH
        IF  (<> 0)  TRANSACTION
        PRINT 'Unable to create the customer record.'
        
        RETURN -1
    END CATCH
    RETURN 0
END
        
```



Correct Answer:

Transact-SQL segments	Answer Area
<input type="text" value="RAISERROR"/>	<pre>CREATE PROCEDURE Sales.InsertCustomer @CustomerName nvarchar(100), @PhoneNumber nvarchar(20), @AccountOpenedDate date, @StandardDiscountPercentage decimal(18,3), @CreditLimit decimal(18,2), @IsCreditOnHold bit, @DeliveryLongitude nvarchar(50), @DeliveryLatitude nvarchar(50) AS BEGIN SET NOCOUNT ON SET <input type="text" value="XACT_ABORT"/> ON BEGIN TRY BEGIN TRANSACTION INSERT INTO Sales.Customers (CustomerName, PhoneNumber, AccountOpenedDate, StandardDiscountPercentage, CreditLimit, IsOnCreditHold, DeliveryLocation) VALUES (@CustomerName, @PhoneNumber, @AccountOpenedDate, @StandardDiscountPercentage, @CreditLimit, @IsCreditOnHold, geography::Point(ISNULL(@DeliveryLongitude, ''), ISNULL(@DeliveryLatitude, ''), 4326)) <input type="text" value="COMMIT"/> TRANSACTION END TRY BEGIN CATCH IF <input type="text" value="XACT_STATE"/> () <> 0 <input type="text" value="ROLLBACK"/> TRANSACTION PRINT 'Unable to create the customer record.' <input type="text" value="THROW"/> RETURN -1 END CATCH RETURN 0 END</pre>
<input type="text" value="@@TRANCOUNT"/>	
<input type="text" value="END"/>	

Box 1: XACT_ABORT

XACT_ABORT specifies whether SQL Server automatically rolls back the current transaction when a Transact-SQL statement raises a run-time error. When SET XACT_ABORT is ON, if a Transact-SQL statement raises a run-time error, the

entire transaction is terminated and rolled back.

Box 2: COMMIT

Commit the transaction.

Box 3: XACT_STATE

Box 4: ROLLBACK

Rollback the transaction

Box 5: THROW

THROW raises an exception and the severity is set to 16.

Requirement: Data modifications that are unsuccessful are rolled back. The exception severity level is set to 16 and a value of -1 is returned.



References:

<https://msdn.microsoft.com/en-us/library/ms188792.aspx>

<https://msdn.microsoft.com/en-us/library/ee677615.aspx>

[Latest 70-761 Dumps](#)

[70-761 Practice Test](#)

[70-761 Study Guide](#)



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.passapply.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 © passapply, All Rights Reserved.