analyticsexam.com
#1 Big Data & Analytics Certification Guide

70-461 – QUERYING MICROSOFT SQL SERVER CERTIFICATION QUESTIONS AND STUDY GUIDE

Querying Microsoft SQL Server 2012/2014 (70-461)

Contents

Querying Microsoft SQL Server Certification Details

Exam Name	Querying Microsoft SQL Server 2012/2014
Exam Code	70-461
Duration	120 Minutes
Passing Percentage	700 out of 1000
Negative Marking	No Negative Marking
Partial Credit	No Partial credit
Reference Book	Training Kit (Exam 70-461): Querying Microsoft SQL Server 2012
Schedule Your exam	Querying Microsoft SQL Server 2012/2014
Sample Questions	Querying Microsoft SQL Server Certification Sample Question
Recommended Practice tool	**Querying Microsoft SQL Server Certification Practice Exam**

Querying Microsoft SQL Server Certification Syllabus for 70-461 (Study Aid)

Create database objects (20–25%)

1. Create and alter tables using T-SQL syntax (simple statements)
    - Create tables without using the built in tools; ALTER; DROP; ALTER COLUMN; CREATE
2. Create and alter views (simple statements)
    - Create indexed views; create views without using the built in tools; CREATE, ALTER, DROP
3. Design views
    - Ensure code non regression by keeping consistent signature for procedure, views and function (interfaces); security implications
4. Create and modify constraints (simple statements)
    - Create constraints on tables; define constraints; unique constraints; default constraints; primary and foreign key constraints
5. Create and alter DML triggers
    - Inserted and deleted tables; nested triggers; types of triggers; update functions; handle multiple rows in a session; performance implications of triggers

Work with data (25–30%)

1. Query data by using SELECT statements
    - Use the ranking function to select top(X) rows for multiple categories in a single query; write and perform queries efficiently using the new (SQL 2005/8->) code items such as synonyms, and joins (except, intersect); implement logic which uses dynamic SQL and system metadata; write efficient, technically complex SQL queries, including all types of joins versus the use of derived tables; determine what code may or may not execute based on the tables provided; given a table with constraints, determine which statement set would load a table; use and understand different data access technologies; case versus isnull versus coalesce

2. Implement sub-queries

- Identify problematic elements in query plans; pivot and unpivot; apply operator; cte statement; with statement

3. Implement data types

- Use appropriate data; understand the uses and limitations of each data type; impact of GUID (newid, newsequentialid) on database performance, when to use what data type for columns

4. Implement aggregate queries

- New analytic functions; grouping sets; spatial aggregates; apply ranking functions

5. Query and manage XML data

- Understand XML datatypes and their schemas and interop w/, limitations and restrictions; implement XML schemas and handling of XML data; XML data: how to handle it in SQL Server and when and when not to use it, including XML namespaces; import and export XML; XML indexing

Modify data (20–25%)

1. Create and alter stored procedures (simple statements)

- Write a stored procedure to meet a given set of requirements; branching logic; create stored procedures and other programmatic objects; techniques for developing stored procedures; different types of storeproc result; create stored procedure for data access layer; program stored procedures, triggers, functions with T-SQL

2. Modify data by using INSERT, UPDATE, and DELETE statements

- Given a set of code with defaults, constraints, and triggers, determine the output of a set of DDL; know which SQL statements are best to solve common requirements; use output statement

3. Combine datasets

- Difference between UNION and UNION all; case versus isnull versus coalesce; modify data by using MERGE statements

4. Work with functions

- Understand deterministic, non-deterministic functions; scalar and table values; apply built-in scalar functions; create and alter user-defined functions (UDFs)

Troubleshoot and optimize (25–30%)

1. Optimize queries

- Understand statistics; read query plans; plan guides; DMVs; hints; statistics IO; dynamic vs. parameterized queries; describe the different join types (HASH, MERGE, LOOP) and describe the scenarios they would be used in

2. Manage transactions

- Mark a transaction; understand begin tran, commit, and rollback; implicit vs explicit transactions; isolation levels; scope and type of locks; trancount

3. Evaluate the use of row-based operations vs. set-based operations

- When to use cursors; impact of scalar UDFs; combine multiple DML operations

4. Implement error handling

- Implement try/catch/throw; use set based rather than row based logic; transaction management

Querying Microsoft SQL Server Exam (70-461) Sample Questions

- Below are the 10 sample questions which will help you be familiar with Querying Microsoft SQL Server 2012/2014 (70-461) exam style and Structure.

- These questions are just for demonstration purpose, there are many scenario based question are included in **Premium Querying Microsoft SQL Server Practice Exam**

- Access to all 230+ questions is available only through premium practice exam available to members at www.analyticsexam .com

Q 1: Your database contains two tables named DomesticSalesOrders and InternationalSalesOrders. Both tables contain more than 100 million rows. Each table has a Primary Key column named SalesOrderId. The data in the two tables is distinct from one another.

Business users want a report that includes aggregate information about the total number of global sales and total sales amounts. You need to ensure that your query executes in the minimum possible time. Which query should you use?
Options:

A. SELECT COUNT(*) AS NumberOfSales, SUM(SalesAmount) AS TotalSalesAmount FROM DomesticSalesOrders UNION SELECT COUNT(*) AS NumberOfSales, SUM(SalesAmount) AS TotalSalesAmount FROM InternationalSalesOrders

B. SELECT COUNT(*) AS NumberOfSales, SUM(SalesAmount) AS TotalSalesAmount FROM (SELECT SalesOrderId, SalesAmount FROM DomesticSalesOrders UNION ALL SELECT SalesOrderId, SalesAmount FROM InternationalSalesOrders) AS p

C. SELECT COUNT(*) AS NumberOfSales, SUM(SalesAmount) AS TotalSalesAmount FROM (SELECT SalesOrderId, SalesAmount FROM DomesticSalesOrders UNION SELECT SalesOrderId, SalesAmount FROM InternationalSalesOrders) AS p

D. SELECT COUNT(*) AS NumberOfSales, SUM(SalesAmount) AS TotalSalesAmount FROM DomesticSalesOrders UNION ALL SELECT COUNT(*) AS NumberOfSales, SUM(SalesAmount) AS TotalSalesAmount FROM InternationalSalesOrders

Q 2: Your database contains tables named Products and ProductsPriceLog. The Products table contains columns named ProductCode and Price. The ProductsPriceLog table contains columns named ProductCode, OldPrice, and NewPrice.

The ProductsPriceLog table stores the previous price in the OldPrice column and the new price in the NewPrice column.

You need to increase the values in the Price column of all products in the Products table by 5 percent. You also need to log the changes to the ProductsPriceLog table.

Which Transact-SQL query should you use?

Options:

A. DATE Products SET Price = Price * 1.05
OUTPUT inserted.ProductCode, deleted.Price, inserted.Price INTO ProductsPriceLog(ProductCode, OldPrice, NewPrice)

B. UPDATE Products SET Price = Price * 1.05
OUTPUT inserted.ProductCode, inserted.Price, deleted.Price INTO ProductsPriceLog(ProductCode, OldPrice, NewPrice)

C. UPDATE Products SET Price = Price * 1.05
OUTPUT inserted.ProductCode, deleted.Price, inserted.Price * 1.05 INTO ProductsPriceLog(ProductCode, OldPrice, NewPrice)

D. UPDATE Products SET Price = Price * 1.05
INSERT INTO ProductsPriceLog(ProductCode, OldPrice, NewPrice) SELECT ProductCode, Price, Price * 1.05 FROM Products

Q 3: Your database contains a table named SalesOrders. The table includes a DATETIME column named OrderTime that stores the date and time each order is placed. There is a nonclustered index on the OrderTime column.

The business team wants a report that displays the total number of orders placed on the current day. You need to write a query that will return the correct results in the most efficient manner. Which Transact-SQL query should you use?
Options:

A. SELECT COUNT(*) FROM SalesOrders WHERE OrderTime = GETDATE()

B. SELECT COUNT(*) FROM SalesOrders WHERE CONVERT(VARCHAR, OrderTime, 112) = CONVERT(VARCHAR, GETDATE(), 112))

C. SELECT COUNT(*) FROM SalesOrders WHERE OrderTime >= CONVERT(DATE, GETDATE()) AND OrderTime < DATEADD(DAY, 1, CONVERT(DATE, GETDATE()))

D. SELECT COUNT(*) FROM SalesOrders WHERE OrderTime = CONVERT(DATE, GETDATE())

Q 4: Your database contains a table named Purchases. The table includes a DATETIME column named PurchaseTime that stores the date and time each purchase is made.
There is a nonclustered index on the PurchaseTime column. The business team wants a report that displays the total number of purchases made on the current day.
You need to write a query that will return the correct results in the most efficient manner. Which Transact-SQL query should you use?
Options:

A. SELECT COUNT(*) FROM Purchases WHERE PurchaseTime = CONVERT(DATE, GETDATE())

B. SELECT COUNT(*) FROM Purchases WHERE PurchaseTime = GETDATE()

C. SELECT COUNT(*) FROM Purchases WHERE CONVERT(VARCHAR, PurchaseTime, 112) = CONVERT(VARCHAR, GETDATE(), 112)

D. SELECT COUNT(*) FROM Purchases WHERE PurchaseTime >= CONVERT(DATE, GETDATE()) AND PurchaseTime < DATEADD(DAY, 1, CONVERT(DATE, GETDATE()))

Q 5: Your database contains a table named Customer that has columns named CustomerID and Name. You want to write a query that retrieves data from the Customer table sorted by Name listing 20 rows at a time. You need to view rows 41 through 60. Which Transact-SQL query should you create?
/files/sas-exam.com/files/user22601/70-461_131.png
Options:

A. Option D

B. Option B

C. Option A

D. Option C

Q 6: Your application contains a stored procedure for each country. Each stored procedure accepts an employee identification number through the @EmpID parameter.

You need to build a single process for each employee that will execute the appropriate stored procedure based on the country of residence.

Which approach should you use?
Options:

A. A SELECT statement that includes CASE

B. Cursor

C. BULK INSERT

D. View

E. A user-defined function

Q 7: You use Microsoft SQL Server 2012 to write code for a transaction that contains several statements. There is high contention between readers and writers on several tables used by your transaction. You need to minimize the use of the tempdb space.

You also need to prevent reading queries from blocking writing queries. Which isolation level should you use?
Options:

A. SERIALIZABLE

B. SNAPSHOT

C. READ COMMITTED SNAPSHOT

D. REPEATABLE READ

Q 8: You use Microsoft SQL Server 2012 to develop a database application. Your application sends data to an NVARCHAR(MAX) variable named @var. You need to write a Transact-SQL statement that will find out the success of a cast to a decimal (36,9). Which code segment should you use?

Options:

A. SELECT IF(TRY_PARSE(@var AS decimal(36,9)) IS NULL, 'True', 'False') AS BadCast

B. SELECT CASE WHEN convert (decimal(36,9), @var) IS NULL THEN 'True' ELSE 'False' END AS BadCast

C. TRY(SELECT convert (decimal(36,9), @var) SELECT 'True' As BadCast) CATCH(SELECT 'False' As BadCast)

D. BEGIN TRY SELECT convert (decimal(36,9), @var) as Value, 'True' As BadCast END TRY BEGIN CATCH SELECT convert (decimal(36,9), @var) as Value, 'False' As BadCast END CATCH

Q 9: You use Microsoft SQL Server 2012 to develop a database application. You need to implement a computed column that references a lookup table by using an INNER JOIN against another table. What should you do?
Options:

A. Reference a user-defined function within the computed column.

B. Create a BEFORE trigger that maintains the state of the computed column.

C. Add a default constraint to the computed column that implements hard-coded values.

D. Add a default constraint to the computed column that implements hard-coded CASE statements.

Q 10: You use Microsoft SQL Server 2012 to develop a database application. You need to create an object that meets the following requirements:

- Takes an input parameter
- Returns a table of values
- Can be referenced within a view

Which object should you use?
Options:

A. inline table-valued function

B. user-defined data type

C. stored procedure

D. scalar-valued function

Answers:

Question: 1	Answer:B	Question: 2	Answer:A
Question: 3	Answer:C	Question: 4	Answer:D
Question: 5	Answer:B	Question: 6	Answer:E
Question: 7	Answer:C	Question: 8	Answer:A
Question: 9	Answer:A	Question: 10	Answer:A