

## 70-762<sup>Q&As</sup>

Developing SQL Databases

### Pass Microsoft 70-762 Exam with 100% Guarantee

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

<https://www.lead4pass.com/70-762.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**

Database users report that SELECT statements take a long time to return results. You run the following Transact-SQL statement:

```
SELECT OBJECT_NAME([object_id]) AS [object_name],
d.equality_columns, d.inequality_columns, d.included_columns
FROM sys.dm_db_missing_index_details;
```

Object_name	Equality_columns	Inequality_columns	Included_columns
[Users]	[CountryCode]	[UserStatus]	[UserName]

You need to create one nonclustered covering index that contains all of the columns in the above table. You must minimize index key size. Which Transact-SQL statement should you run?

- A. CREATE NONCLUSTERED INDEX IX\_User ON Users (CountryCode, UserName);
- B. CREATE NONCLUSTERED INDEX IX\_User ON Users (CountryCode, UserStatus) INCLUDE (UserName);
- C. CREATE NONCLUSTERED INDEX IX\_User ON Users (CountryCode, UserStatus, UserName);
- D. CREATE NONCLUSTERED INDEX IX\_User ON Users (UserStatus, CountryCode) INCLUDE (UserName);

Correct Answer: D

Use the UserStatus as the first column in the index, as it is an in\_equality column. Incorrect Answers:

A: UserStatus is not included.

References: <https://docs.microsoft.com/en-us/sql/relational-databases/indexes/create-indexes-with-included-columns>

---

**QUESTION 2**

You have the following stored procedure:

```
CREATE PROCEDURE AddNextNumber @Number INT
AS
BEGIN
    SET ANSI_DEFAULTS ON
    INSERT INTO Numbers (Number) VALUES (@Number)
END
```

The Numbers table becomes unavailable when you run the stored procedure. The stored procedure obtains an exclusive lock on the table and does not release the lock.

What are two possible ways to resolve the issue? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

- A. Remove the implicit transaction and the SET ANSI\_DEFAULTS ON statement.
- B. Set the ANSI\_DEFAULT statement to OFF and add a COMMIT TRANSACTION statement after the INSERT statement.
- C. Add a COMMIT TRANSACTION statement after the INSERT statement.
- D. Remove the SET ANSI\_DEFAULTS ON statement.

Correct Answer: CD

SET ANSI\_DEFAULTS is a server-side setting that the client does not modify. When enabled (ON), this option enables SET IMPLICIT\_TRANSACTIONS (and some other options).

The SET IMPLICIT\_TRANSACTIONS, when ON, the system is in implicit transaction mode.

This means that if @@TRANCOUNT = 0, any of the following Transact-SQL statements begins a new transaction. It is equivalent to an unseen BEGIN TRANSACTION being executed first: ALTER TABLE, FETCH, REVOKE, BEGIN

TRANSACTION, GRANT, SELECT, CREATE, INSERT, TRUNCATE TABLE, DELETE, OPEN, UPDATE, DROP.

References: <https://docs.microsoft.com/en-us/sql/t-sql/statements/set-implicit-transactions-transact-sql?view=sql-server-2017>

---

### QUESTION 3

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution. Determine whether the solution meets the stated goals.

You have a database that contains a table named Employees. The table stores information about the employees of your company.

You need to implement and enforce the following business rules:

Limit the values that are accepted by the Salary column.

Prevent salaries less than \$15,000 and greater than \$300,000 from being entered.

Determine valid values by using logical expressions.

Do not validate data integrity when running DELETE statements.

Solution: You implement cascading referential integrity constraints on the table.

Does the solution meet the goal?

- A. Yes
- B. No

Correct Answer: A

References: [https://technet.microsoft.com/en-us/library/ms186973\(v=sql.105\).aspx](https://technet.microsoft.com/en-us/library/ms186973(v=sql.105).aspx)

---

**QUESTION 4**

You run the following Transact-SQL statements:

```
CREATE TABLE Customers (  
    CustomerID INT NOT NULL IDENTITY PRIMARY KEY CLUSTERED,  
    CustomerName NVARCHAR (100) UNIQUE NOT NULL  
)  
  
CREATE TABLE Orders (  
    OrderID INT NOT NULL IDENTITY PRIMARY KEY CLUSTERED,  
    CustomerID INT NOT NULL REFERENCES Customers (CustomerID),  
    OrderDate DATE NOT NULL  
)  
  
CREATE VIEW v_CustomerOrder  
AS SELECT  
    b.CustomerName, a.OrderID, a.OrderDate,  
    (SELECT COUNT(*) FROM Orders c WHERE c.CustomerID = a.CustomerID) AS CustomerOrderCount  
FROM Orders a  
INNER JOIN Customers b ON a.CustomerID = b.CustomerID
```

Records must only be added to the Orders table by using the view. If a customer name does not exist, then a new customer name must be created. You need to ensure that you can insert rows into the Orders table by using the view.

- A. Add the CustomerID column from the Orders table and the WITH CHECK OPTION statement to the view.
- B. Create an INSTEAD OF trigger on the view.
- C. Add the WITH SCHEMABINDING statement to the view statement and create a clustered index on the view.
- D. Remove the subquery from the view, add the WITH SCHEMABINDING statement, and add a trigger to the Orders table to perform the required logic.

Correct Answer: A

The WITH CHECK OPTION clause forces all data-modification statements executed against the view to adhere to the criteria set within the WHERE clause of the SELECT statement defining the view. Rows cannot be modified in a way that causes them to vanish from the view.

References: <http://www.informit.com/articles/article.aspx?p=130855andseqNum=4>

---

**QUESTION 5**

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 questions 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 is 130 GB and contains 500 million rows of data.

Granular transactions and mass batch data imports change the database frequently throughout the day. Microsoft SQL Server Reporting Services (SSRS) uses the database to generate various reports by using several filters.

You discover that some reports time out before they complete.

You need to reduce the likelihood that the reports will time out.

Solution: You create a file group for the indexes and a file group for the data files. You store the files for each file group on separate disks.

Does this meet the goal?

A. Yes

B. No

Correct Answer: A

Consider creating two additional File Groups: Tables and Indexes. It is best not to put your stuff in PRIMARY as that is where SQL SERVER stores all of its data and meta-data about your objects. You create your Table and Clustered Index (as that is the data for the table) on [Tables] and all Non-Clustered indexes on [Indexes].

[70-762 PDF Dumps](#)

[70-762 VCE Dumps](#)

[70-762 Practice Test](#)