

70-762^{Q&As}

Developing SQL Databases

Pass Microsoft 70-762 Exam with 100% Guarantee

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

<https://www.leads4pass.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

You have a view that includes an aggregate.

You must be able to change the values of columns in the view. The changes must be reflected in the tables that the view uses.

You need to ensure that you can update the view.

What should you create?

- A. table-valued function
- B. a schema-bound view
- C. a partitioned view
- D. a DML trigger

Correct Answer: D

QUESTION 2

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 stored information about the employees of your company. You need to implement the following auditing rules for the Employees table:

- Record any changes that are made to the data in the Employees table.

-

Customize the data recorded by the audit operations.

Solution: You implement a user-defined function on the Employees table.

Does the solution meet the goal?

- A.
Yes
- B.
No

Correct Answer: A

SQL Server 2016 provides two features that track changes to data in a database: change data capture and change tracking. These features enable applications to determine the DML changes (insert, update, and delete operations) that were

made to user tables in a database.

Change data is made available to change data capture consumers through table-valued functions (TVFs).

References: <https://msdn.microsoft.com/en-us/library/cc645858.aspx>

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 a check constraint on the table.

Does the solution meet the goal?

A. Yes

B. No

Correct Answer: A

References: https://en.wikipedia.org/wiki/Check_constraint

QUESTION 4

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 have a database named DB1 that contains the following tables: Customer, CustomerToAccountBridge, and CustomerDetails. The three tables are part of the Sales schema. The database also contains a schema named Website. You create the Customer table by running the following Transact-SQL statement: The value of the CustomerStatus column is equal to one for active customers. The value of the Account1Status and Account2Status columns are equal to one for active accounts. The following table displays selected columns and rows from the Customer table.

```
CREATE TABLE Customer
(
    CustomerNumber int NOT NULL,
    CustomerName varchar(50) NOT NULL,
    CreateDate date NOT NULL,
    Gender bit,
    Address varchar(50)
    City varchar(50)
    State char(2),
    CustomerStatus bit NOT NULL,
    MaritalStatus bit,
    Segment varchar(5),
    CountryCode char(2),
    Birthday date,
    PostalCode char(5),
    PhoneNumber varchar(20),
    Account1 char(7),
    Account1Status bit,
    Account2 char(7),
    Account2Status bit,
    CONSTRAINT PK_Customer PRIMARY KEY CLUSTERED (CustomerNumber)
);
```

| Customer ID | CustomerName | Gender | Account1 | Account1Status | Account2 | Account2Status |
|-------------|--------------|--------|----------|----------------|----------|----------------|
| 101 | Name A | 0 | 0001001 | 0 | 0001002 | 1 |
| 102 | Name B | 1 | 0002001 | 1 | 0002002 | 0 |
| 103 | Name C | 0 | 0003001 | 1 | 0003002 | 1 |

You plan to create a view named Website.Customer and a view named Sales.FemaleCustomers.

Website.Customer must meet the following requirements:

Allow users access to the CustomerName and CustomerNumber columns for active customers.

Allow changes to the columns that the view references. Modified data must be visible through the view.

Prevent the view from being published as part of Microsoft SQL Server replication.

Sales.Female.Customers must meet the following requirements:

Allow users access to the CustomerName, Address, City, State and PostalCode columns.

Prevent changes to the columns that the view references.

Only allow updates through the views that adhere to the view filter.

You have the following stored procedures: spDeleteCustAcctRelationship and spUpdateCustomerSummary. The spUpdateCustomerSummary stored procedure was created by running the following Transacr-SQL statement:

```
CREATE PROCEDURE uspUpdateCustomerSummary
@CustomerId INT
AS
BEGIN
    SET NOCOUNT on;
    UPDATE CustomerDetails SET TotalDepositAccountCount = TotalDepositAccountCount + 1 WHERE CustomerID = @CustomerId;
    BEGIN TRAN;
        BEGIN TRY
            UPDATE CustomerDetails SET TotalAccountCount = TotalAccountCount + 1 WHERE CustomerID = @CustomerId;
        END TRY
        BEGIN CATCH
            IF @@TRANCOUNT > 0
                ROLLBACK TRAN;
        END CATCH
        IF @@TRANCOUNT > 0
            COMMIT TRAN;
```

You run the `uspUpdateCustomerSummary` stored procedure to make changes to customer account summaries. Other stored procedures call the `spDeleteCustAcctRelationship` to delete records from the `CustomerToAccountBridge` table.

When you start `uspUpdateCustomerSummary`, there are no active transactions. The procedure fails at the second update statement due to a CHECK constraint violation on the `TotalDepositAccountCount` column.

What is the impact of the stored procedure on the `CustomerDetails` table?

- A. A. The value of the `TotalAccountCount` column decreased.
- B. The value of the `TotalDepositAccountCount` column is not changed.
- C. The statement that modifies `TotalDepositAccountCount` is excluded from the transaction.
- D. The value of the `TotalAccountCount` column is not changed.
- E. The value of the `TotalDepositAccountCount` column is increased.
- F. The statement that modifies `TotalAccountCount` column is excluded from the transaction.
- G. The value of the `TotalDepositAccountCount` column is decreased.

Correct Answer: D

QUESTION 5

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

Your company has employees in different regions around the world. You need to create a database table that stores the following employee attendance information: Employee ID date and time employee checked in to work date and time employee checked out of work

Date and time information must be time zone aware and must not store fractional seconds.

Solution: You run the following Transact-SQL statement:

```
CREATE TABLE [dbo].[EmployeeAttendance] (  
    EmployeeID int NOT NULL,  
    DateCheckedIn datetimeoffset(0) NOT NULL,  
    DateCheckedOut datetimeoffset(0) NOT NULL)
```

Does the solution meet the goal?

- A. Yes
- B. No

Correct Answer: A

Datetimeoffset defines a date that is combined with a time of a day that has time zone awareness and is based on a 24-hour clock.

Syntax: datetimeoffset [(fractional seconds precision)]

For the use "datetimeoffset(0)", the Fractional seconds precision is 0, which is required here.

References: <https://msdn.microsoft.com/en-us/library/bb630289.aspx>

QUESTION 6

You are experiencing performance issues with the database server.

You need to evaluate schema locking issues, plan cache memory pressure points, and backup I/O problems.

What should you create?

- A. a System Monitor report
- B. a sys.dm_exec_query_stats dynamic management view query
- C. a sys.dm_exec_session_wait_stats dynamic management view query
- D. an Activity Monitor session in Microsoft SQL Management Studio.

Correct Answer: C

sys.dm_exec_session_wait_stats returns information about all the waits encountered by threads that executed for each session. You can use this view to diagnose performance issues with the SQL Server session and also with specific queries and batches.

Note: SQL Server wait stats are, at their highest conceptual level, grouped into two broad categories: signal waits and resource waits. A signal wait is accumulated by processes running on SQL Server which are waiting for a CPU to become available (so called because the process has "signaled" that it is ready for processing). A resource wait is accumulated by processes running on SQL Server which are waiting for a specific resource to become available, such as waiting for the release of a lock on a specific record.

QUESTION 7

You are designing a stored procedure for a database named DB1.

The following requirements must be met during the entire execution of the stored procedure:

The stored procedure must only read changes that are persisted to the database.

SELECT statements within the stored procedure should only show changes to the data that are made by the stored procedure.

You need to configure the transaction isolation level for the stored procedure.

Which Transact-SQL statement or statements should you run?

- A. SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED ALTER DATABASE DB1 SET READ_COMMITTED_SNAPSHOT ON
- B. SET TRANSACTION ISOLATION LEVEL READ COMMITTED ALTER DATABASE DB1 SET READ_COMMITTED_SNAPSHOT OFF
- C. SET TRANSACTION ISOLATION LEVEL SERIALIZABLE
- D. SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED ALTER DATABASE SET READ_COMMITTED_SNAPSHOT OFF

Correct Answer: B

READ COMMITTED specifies that statements cannot read data that has been modified but not committed by other transactions. This prevents dirty reads. Data can be changed by other transactions between individual statements within the

current transaction, resulting in nonrepeatable reads or phantom data. This option is the SQL Server default.

Incorrect Answers:

A, D: READ UNCOMMITTED specifies that statements can read rows that have been modified by other transactions but not yet committed.

References: <https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/sql/snapshot-isolation-in-sql-server>

QUESTION 8

Note: This question is part of a series of questions that present the same scenario. Each question in this series contains a unique solution. Determine whether the solution meets the stated goals. The Account table was created using the following Transact-SQL statement:


```
CREATE TABLE Account
(
    AccountNumber int NOT NULL,
    ProductCode char(2) NOT NULL,
    Status tinyint NOT NULL,
    OpenDate date NOT NULL,
    CloseDate date,
    Balance decimal(15,2),
    AvailableBalance decimal(15,2)
);
```

There are more than 1 billion records in the Account table. The Account Number column uniquely identifies each account. The ProductCode column has 100 different values. The values are evenly distributed in the table. Table statistics are refreshed and up to date.

You frequently run the following Transact-SQL SELECT statements:

```
SELECT ProductCode, SUM(Balance) AS TotalSUM FROM Account WHERE ProductCode
<> 'CD' GROUP BY ProductCode;
SELECT AccountNumber, Balance FROM Account WHERE ProductCode = 'CD'
```

You must avoid table scans when you run the queries. You need to create one or more indexes for the table. Solution: You run the following Transact-SQL statement:

```
CREATE NONCLUSTERED INDEX PK_Account (AccountNumber) ;
CREATE NONCLUSTERED INDEX IX_Account_productCode On Account (ProductCode)
INCLUDE (Balance);
```

Does the solution meet the goal?

A. Yes

B. No

Correct Answer: B

Create a clustered index on the AccountNumber column as it is unique, not a non nonclustered one.

References:<https://msdn.microsoft.com/en-us/library/ms190457.aspx>

QUESTION 9

You are designing a solution for a company that operates retail stores. Each store has a database that tracks sales transactions. You create a summary table in the database at the corporate office. You plan to use the table to record the

quantity of each product sold at each store on each day. Managers will use this data to identify reorder levels for

products.

Every evening stores, must transmit sales data to the corporate office. The data must be inserted into the summary table that includes the StoreID, ProductID, Qtysold, Totprodsales, and Datesold columns.

You need to prevent duplicate rows in the summary table. Each row must uniquely identify the store that sold the product and the total amount sold for that store on a specific date.

What should you include in your solution?

- A. Create a unique constraint.
- B. Create a foreign key constraint to the StoreID column in each of the store tables.
- C. Create a rule and bind it to the StoreID column.
- D. Create a check constraint.
- E. Create a table-valued user-defined function.

Correct Answer: A

You can use UNIQUE constraints to make sure that no duplicate values are entered in specific columns that do not participate in a primary key. Although both a UNIQUE constraint and a PRIMARY KEY constraint enforce uniqueness, use a UNIQUE constraint instead of a PRIMARY KEY constraint when you want to enforce the uniqueness of a column, or combination of columns, that is not the primary key.

Incorrect Answers:

D: CHECK constraints enforce domain integrity by limiting the values that are accepted by one or more columns.

References: <https://docs.microsoft.com/en-us/sql/relational-databases/tables/unique-constraints-and-check-constraints?view=sql-server-2017>

QUESTION 10

Note: This question is part of a series of questions that use the same or similar answer choices. As 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 have a Microsoft SQL Server database named DB1 that contains the following tables:

| Table name | Description |
|------------|---|
| TBL1 | <ul style="list-style-type: none">• Column1 is configured as the primary key.• Column2 will store the year.• TBL1 only stores data for the year 2016.• The table will contain 1 million records. |
| TBL2 | <ul style="list-style-type: none">• Column1 is configured as the primary key.• Column2 will store the year.• TBL2 only stores data for the year 2015.• The table will contain 1 million records. |

Users frequently run the following query. The users report that the query takes a long time to return results.

```
SELECT Column1, Column2, Column3
FROM (
    SELECT Column1, Column2, Column3
    FROM TBL1
    UNION ALL
    SELECT Column1, Column2, Column3
    FROM TBL2)
WHERE Column2 = <year> AND Column3 = 1
```

You need to minimize the amount of time required for the query to return data.

- A. Create clustered indexes on TBL1 and TBL2.
- B. Create a clustered index on TBL1. Create a nonclustered index on TBL2 and add the most frequently queried column as included columns.
- C. Create a nonclustered index on TBL2 only.
- D. Create UNIQUE constraints on both TBL1 and TBL2. Create a partitioned view that combines columns from TBL1 and TBL2.
- E. Drop existing indexes on TBL1 and then create a clustered columnstore index. Create a nonclustered columnstore index on TBL1. Create a nonclustered index on TBL2.
- F. Drop existing indexes on TBL1 and then create a clustered columnstore index. Create a nonclustered columnstore index on TBL1. Make no changes to TBL2.
- G. Create CHECK constraints on both TBL1 and TBL2. Create a partitioned view that combines columns from TBL1 and TBL2.
- H. Create an indexed view that combines columns from TBL1 and TBL2.

Correct Answer: D

A partitioned view is a view defined by a UNION ALL of member tables structured in the same way, but stored separately as multiple tables in either the same instance of SQL Server or in a group of autonomous instances of SQL Server servers, called federated database servers.

References: <https://docs.microsoft.com/en-us/sql/t-sql/statements/create-view-transact-sql?view=sql-server-2017#partitioned-views>

QUESTION 11

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 independent of the other questions in this series.

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

You are a database developer for a company. The company has a server that has multiple physical disks. The disks are not part of a RAID array. The server hosts three Microsoft SQL Server instances. There are many SQL jobs that run during off-peak hours.

You must monitor and optimize the SQL Server to maximize throughput, response time, and overall SQL performance.

You need to identify previous situations where a modification has prevented queries from selecting data in tables.

What should you do?

- A. Create a `sys.dm_os_waiting_tasks` query.
- B. Create a `sys.dm_exec_sessions` query.
- C. Create a Performance Monitor Data Collector Set.
- D. Create a `sys.dm_os_memory_objects` query.
- E. Create a `sp_configure 'max server memory'` query.
- F. Create a SQL Profiler trace.
- G. Create a `sys.dm_os_wait_stats` query.
- H. Create an Extended Event.

Correct Answer: G

`sys.dm_os_wait_stats` returns information about all the waits encountered by threads that executed. You can use this aggregated view to diagnose performance issues with SQL Server and also with specific queries and batches.

QUESTION 12

Note: This question is part of a series of questions that use the same 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 are developing an application to track customer sales. You create tables to support the application. You need to create a database object that meets the following data entry requirements: What should you create?

| Table name | Data entry requirements |
|----------------------|---|
| Customers | Inserts must be grouped in a batch of Transact-SQL statements. |
| CustomerTransactions | Inserts must be grouped in a batch of Transact-SQL statements. Inserts must be grouped in a batch of Transact-SQL statements. All of the data modifications made must become a permanent part of the database if data entry is successful. |
| Invoices | All data entry modifications must be erased if an error occurs. Inserts must be grouped in a batch of Transact-SQL statements. All of the data modifications made must become a permanent part of the database if data entry is successful. |
| Orders | All data entry modifications must be erased if an error occurs. |

- A. extended procedure
- B. CLR procedure
- C. user-defined procedure
- D. DML trigger
- E. DDL trigger
- F. scalar-valued function
- G. table-valued function

Correct Answer: C

References: <https://msdn.microsoft.com/en-us/library/ms345075.aspx>

QUESTION 13

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 need to identify the indexes that are not being used so that you can remove them from the database.

Solution: You run the sys.dm_db_index_usage_stats dynamic management view.

Does the solution meet the goal?

- A. Yes
- B. No

Correct Answer: A

References: <https://www.mssqltips.com/sqlservertutorial/256/discovering-unused-indexes/>

<https://www.sqlshack.com/how-to-identify-and-monitor-unused-indexes-in-sql-server/>

QUESTION 14

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 need to create a stored procedure that updates the Customer, CustomerInfo, OrderHeader, and OrderDetails tables in order.

You need to ensure that the stored procedure:

Runs within a single transaction.

Commits updates to the Customer and CustomerInfo tables regardless of the status of updates to the OrderHeader and OrderDetail tables.

Commits changes to all four tables when updates to all four tables are successful.

Solution: You create a stored procedure that includes the following Transact-SQL segment:

```
DECLARE @CustomerComplete bit = 0
BEGIN TRY
    BEGIN TRAN
        UPDATE Customer ...
        UPDATE CustomerInfo ...

        SET @CustomerComplete = 1

        SAVE TRAN TR1

        UPDATE OrderHeader ...
        UPDATE OrderDetail ...
    COMMIT TRAN
END TRAN
BEGIN CATCH
    IF (@CustomerComplete = 1) AND (XACT_STATE() = 1)
    BEGIN
        COMMIT TRAN
    END
    ELSE IF XACT_STATE() = 1
        ROLLBACK TRAN
END CATCH
```

Does the solution meet the goal?

- A. Yes
- B. No

Correct Answer: A

Need to handle the case where the first two updates (OrderHeader, OrderDetail) are successful, but either the 3rd or the 4th (OrderHeader, OrderDetail) fail. We add the @CustomerComplete variable in the BEGIN TRY block, and test it in the BEGIN CATCH block.

Note: XACT_STATE indicates whether the request has an active user transaction, and whether the transaction is capable of being committed. XACT_STATE =1: the current request has an active user transaction. The request can perform any actions, including writing data and committing the transaction.

References: <https://docs.microsoft.com/en-us/sql/t-sql/functions/xact-state-transact-sql>

QUESTION 15

You have several real-time applications that constantly update data in a database. The applications run more than 400 transactions per second that insert and update new metrics from sensors.

A new web dashboard is released to present the data from the sensors. Engineers report that the applications take longer than expected to commit updates.

You need to change the dashboard queries to improve concurrency and to support reading uncommitted data.

What should you do?

- A. Use the NOLOCK option.
- B. Execute the DBCC UPDATEUSAGE statement.
- C. Use the max worker threads option.
- D. Use a table-valued parameter.
- E. Set SET ALLOW_SNAPSHOT_ISOLATION to ON.
- F. Set SET XACT_ABORT to ON.
- G. Execute the ALTER TABLE T1 SET (LOCK_ESCALATION = AUTO); statement.
- H. Use the OUTPUT parameters.

Correct Answer: A

The NOLOCK hint allows SQL to read data from tables by ignoring any locks and therefore not being blocked by other processes. This can improve query performance, but also introduces the possibility of dirty reads.

Incorrect Answers:

F: When SET XACT_ABORT is ON, if a Transact-SQL statement raises a run-time error, the entire transaction is terminated and rolled back.

G: DISABLE, not AUTO, would be better.

There are two more lock escalation modes: AUTO and DISABLE.

The AUTO mode enables lock escalation for partitioned tables only for the locked partition. For non-partitioned tables it

works like TABLE. The DISABLE mode removes the lock escalation capability for the table and that is important when concurrency issues are more important than memory needs for specific tables.

Note: SQL Server's locking mechanism uses memory resources to maintain locks. In situations where the number of row or page locks increases to a level that decreases the server's memory resources to a minimal level, SQL Server's locking strategy converts these locks to entire table locks, thus freeing memory held by the many single row or page locks to one table lock. This process is called lock escalation, which frees memory, but reduces table concurrency.

References: <https://www.mssqltips.com/sqlservertip/2470/understanding-the-sql-server-nolock-hint/>

[Latest 70-762 Dumps](#)

[70-762 VCE Dumps](#)

[70-762 Study Guide](#)