DUMPSBOSS.

Developing SQL Databases

Microsoft 70-762

Version Demo

Total Demo Questions: 10

Total Premium Questions: 181

Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

QUESTION NO: 1

You have the following stored procedure that is called by other stored procedures and applications:

```
CREATE PROCEDURE UpdateCustomer @CustomerId INT
AS
BEGIN
    EXEC ProcessCustomer_Internal @CustomerId
    DECLARE @Status INT
    SELECT @Status = Status FROM Customer WHERE CustomerId =
@CustomerId
END
```

You need to modify the stored procedure to meet the following requirements:

- Always return a value to the caller.
- Return 0 if @Status is NULL.
- Callers must be able to use @Status as a variable.

Which two actions should you perform? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

A. Replace NULL values with 0. Add a PRINT statement to return @Status.

B. Add a RETURN statement.

C. Replace NULL values with 0. Add an output parameter to return @Status.

D. Replace NULL values with 0. Add a SELECT statement to return @Status.

E. Add a PRINT statement.

F. Add a SELECT statement to return @Status.

G. Add an output parameter to return @Status.

ANSWER: B C

Explanation:

There are three ways of returning data from a procedure to a calling program: result sets, output parameters, and return codes.

References: https://docs.microsoft.com/en-us/sql/relational-databases/stored-procedures/return-data-from-a-stored-procedure?view=sql-server-2017

QUESTION NO: 2 - (HOTSPOT)

HOTSPOT

You have a Microsoft Azure SQL database.

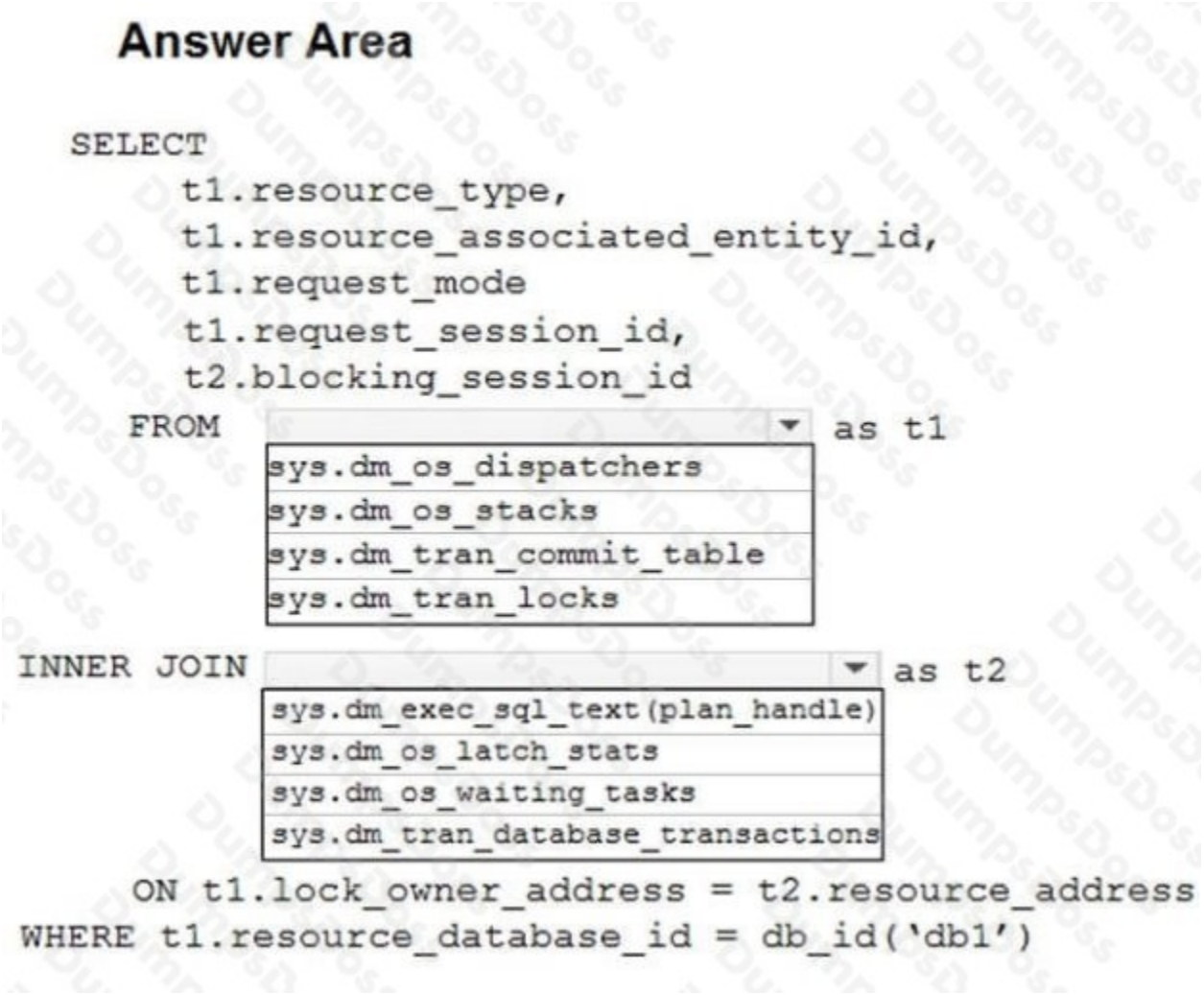
Some queries take a long time to execute.

You need to identify whether any of the long-running queries prevent users from updating data in the database.

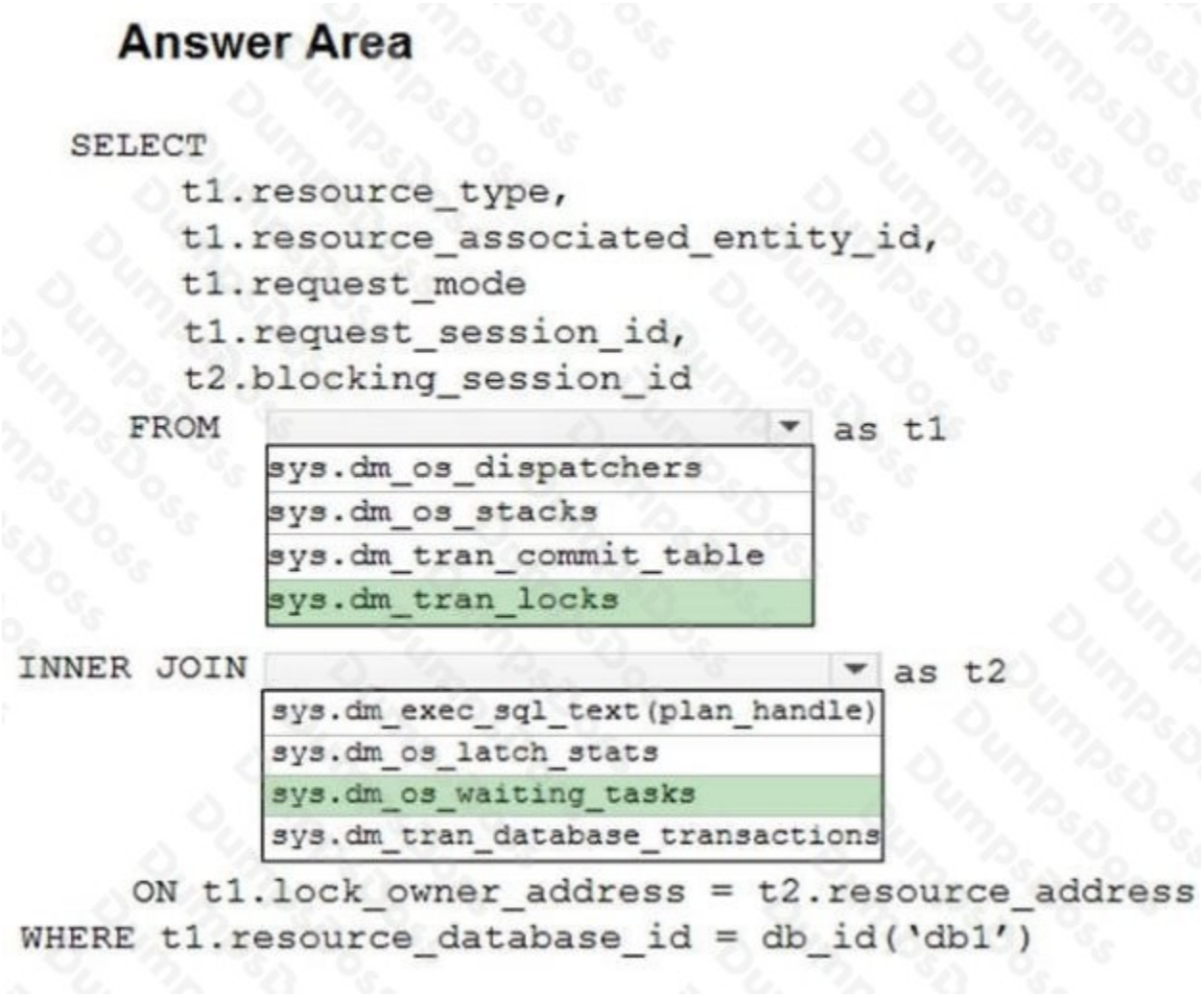
Which query should you execute? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Hot Area:



ANSWER:



Explanation:

Box 1: sys.dm_tran_locks

sys.dm_tran_locks returns information about currently active lock manager resources in SQL Server 2017. Each row represents a currently active request to the lock manager for a lock that has been granted or is waiting to be granted.

Box 2: sys.dm_os_waiting_tasks

sys.dm_tran_locks.lock_owner_address is the memory address of the internal data structure that is used to track this request. This column can be joined the with resource_address column in sys.dm_os_waiting_tasks.

References: https://docs.microsoft.com/en-us/sql/relational-databases/system-dynamic-management-views/sys-dm-tran-locks-transact-sql?view=sqlserver-2017

QUESTION NO: 3

You have a database with multiple tables. You must insert data into the tables by using views.

Which of the following two statements best describes an updatable view?

NOTE: Each correct selection is worth one point.

A. The view must not include the primary key of the table.

B. The view may include GROUP BY or HAVING statements.

C. The view may include a composite field.

D. Each view must reference columns from one table.

E. The view must not include subqueries.

ANSWER: B D

Explanation:

B: The columns being modified must not be affected by GROUP BY, HAVING, or DISTINCT clauses.

D: Any modifications, including UPDATE, INSERT, and DELETE statements, must reference columns from only one base table.

Incorrect Answers:

A: Primary keys are allowed.

C: The columns cannot be derived in any other way, such as through the following:

A computation. The column cannot be computed from an expression that uses other columns.

E: The restrictions apply to any subqueries in the FROM clause of the view, just as they apply to the view itself.

References:

https://docs.microsoft.com/en-us/sql/t-sql/statements/create-view-transact-sql?view=sql-server-2017

QUESTION NO: 4

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 named dbReporting. Users run a large number of read-only ad hoc queries against the database. The application and all ad hoc queries use default database transaction isolation levels. You set the value of the READ_COMMITTED_SNAPSHOT database option to ON.

You have an application that updates 10 tables sequentially and modifies a large volume of records in a single transaction. The updates are isolated from each other.

Users report an error which indicates that the version store us full.

You need to reduce the number of occurrences of the error.

Solution: You increase the maximum database size for the tempdb database.

Does the solution meet the goal?

A. Yes

B. No

ANSWER: A

Explanation:

References: https://docs.microsoft.com/en-us/sql/relational-databases/databases/tempdb-database?view=sql-server-2017

QUESTION NO: 5

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.

ANSWER: C D

Explanation:

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 NO: 6

You have multiple queries that take a long time to complete.

You need to identify the cause by using detailed information about the Transact-SQL statements in the queries. The Transact-SQL statements must not run as part of the analysis.

Which Transact-SQL statement should you run?

A. SET STATISTICS IO ON

B. SET SHOWPLAN_TEXT ON

C. SET STATISTICS XML ON

D. SET STATISTICS PROFILE ON

ANSWER: B

Explanation:

SET SHOWPLAN_TEXT ON causes Microsoft SQL Server not to execute Transact-SQL statements. Instead, SQL Server returns detailed information about how the statements are executed.

Incorrect Answers:

C: SET STATISTICS XML ON causes Microsoft SQL Server to execute Transact-SQL statements and generate detailed information about how the statements were executed in the form of a well-defined XML document.

D: When STATISTICS PROFILE is ON, each executed query returns its regular result set, followed by an additional result set that shows a profile of the query execution.

The additional result set contains the SHOWPLAN_ALL columns for the query and these additional columns.

Column name, Description

Rows: Actual number of rows produced by each operator Executes: Number of times the operator has been executed

References:

https://docs.microsoft.com/en-us/sql/t-sql/statements/set-showplan-text-transact-sql

QUESTION NO: 7

You are developing a database reporting solution for a table that contains 900 million rows and is 103 GB.

The table is updated thousands of times a day, but data is not deleted.

The SELECT statements vary in the number of columns used and the amount of rows retrieved.

You need to reduce the amount of time it takes to retrieve data from the table. The must prevent data duplication.

Which indexing strategy should you use?

A. a nonclustered index for each column in the table

B. a clustered columnstore index for the table

C. a hash index for the table

D. a clustered index for the table and nonclustered indexes for nonkey columns

ANSWER: B

Explanation:

Columnstore indexes are the standard for storing and querying large data warehousing fact tables. It uses column-based data storage and query processing to achieve up to 10x query performance gains in your data warehouse over traditional row-oriented storage.

A clustered columnstore index is the physical storage for the entire table.

Generally, you should define the clustered index key with as few columns as possible.

A nonclustered index contains the index key values and row locators that point to the storage location of the table data. You can create multiple nonclustered indexes on a table or indexed view. Generally, nonclustered indexes should be designed to improve the performance of frequently used queries that are not covered by the clustered index.

References: https://docs.microsoft.com/en-us/sql/relational-databases/indexes/columnstore-indexes-overview?view=sql-server-2017

QUESTION NO: 8 - (DRAG DROP)

DRAG DROP

You have a database that contains three encrypted store procedures named dbo.Proc1, dbo.Proc2 and dbo.Proc3. The stored procedures include INSERT, UPDATE, DELETE and BACKUP DATABASE statements.

You have the following requirements:

- You must run all the stored procedures within the same transaction.

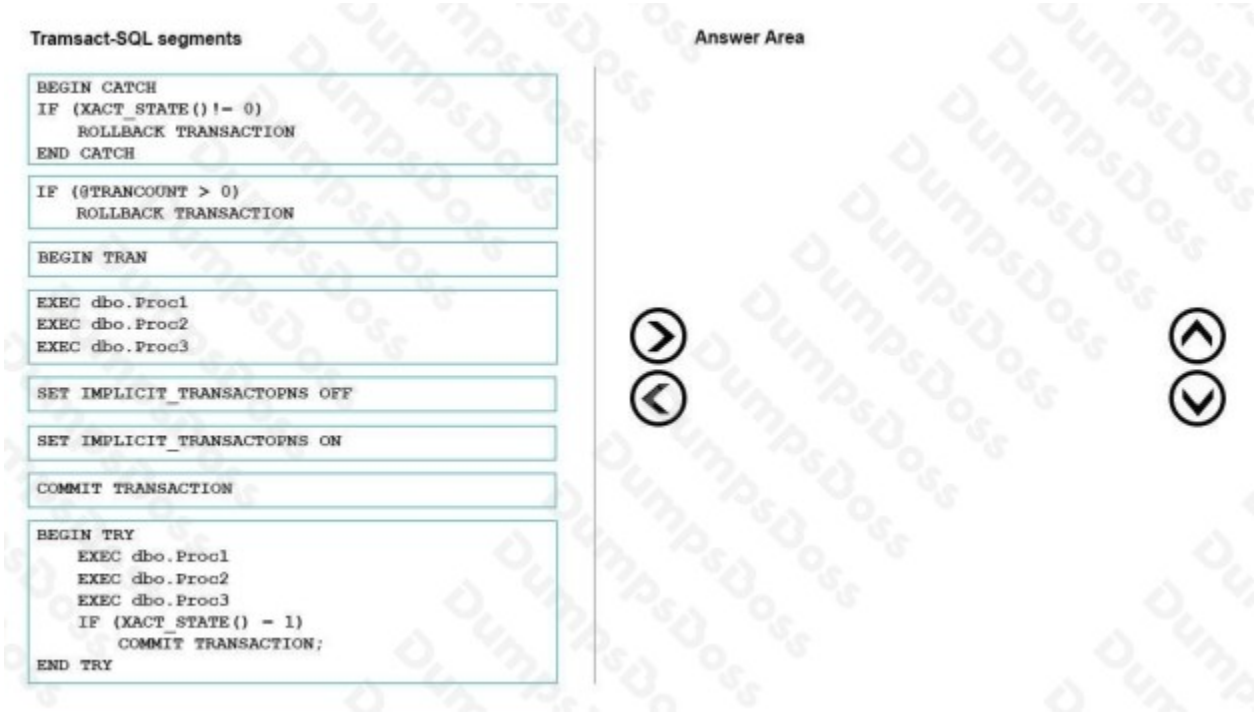
- You must automatically start a transaction when stored procedures include DML statements.

- You must not automatically start a transaction when stored procedures include DDL statements.

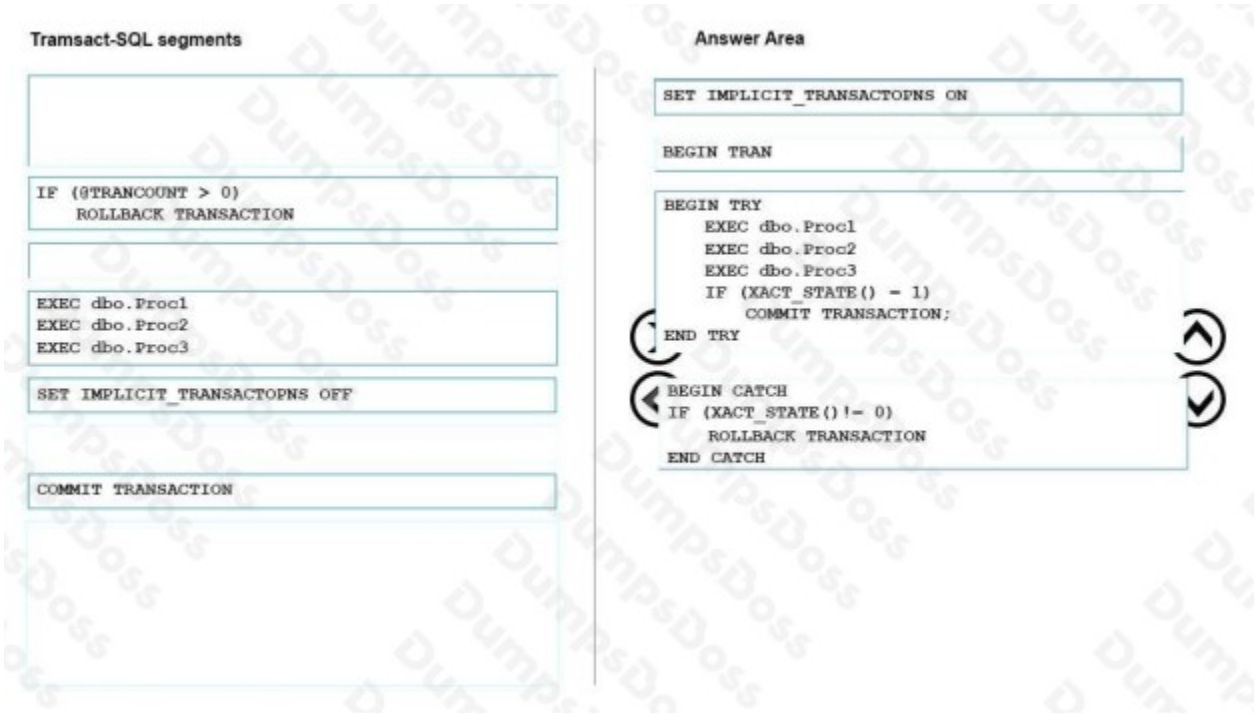
You need to run all three stored procedures.

Which four Transact-SQL segments should you use to develop the solution? To answer, move the appropriate Transact-SQL segments to the answer area and arrange then in the correct order.

Select and Place:



ANSWER:



Explanation:

Note:

Implicit transaction mode remains in effect until the connection executes a SET IMPLICIT_TRANSACTIONS OFF statement, which returns the connection to autocommit mode. In autocommit mode, allindividual statements are committed if they complete successfully.

When a connection is in implicit transaction mode and the connection is not currently in a transaction, executing any of the following statements starts a transaction:

ALTER TABLE (DDL)

FETCH

REVOKE

BEGIN TRANSACTION

GRANT

SELECT

CREATE (DDL)

INSERT

TRUNCATE TABLE

DELETE (DML)

OPEN

UPDATE (DML) DROP (DDL)

Note 2: XACT_STATE returns the following values.

1 The current request has an active user transaction. The request can perform any actions, including writing data and committing the transaction. The transaction is committable.

-1 The current request has an active user transaction, but an error has occurred that has caused the transaction to be classified as an uncommittable transaction.the transaction is uncommittable and should be rolled back.

0 There is no active user transaction for the current request. Acommit or rollback operation would generate an error.

References: https://technet.microsoft.com/en-us/library/ms187807(v=sql.105).aspx https://technet.microsoft.com/en-us/library/ms189797(v=sql.110).aspx

QUESTION NO: 9 - (DRAG DROP)

Case Study

Background

You have a database named HR1 that includes a table named Employee.

You have several read-only, historical reports that contain regularly changing totals. The reports use multiple queries to estimate payroll expenses. The queries run concurrently. Users report that the payroll estimate reports do not always run. You must monitor the database to identify issues that prevent the reports from running.

You plan to deploy the application to a database server that supports other applications. You must minimize the amount of storage that the database requires.

Employee Table

You use the following Transact-SQL statements to create, configure, and populate the Employee table:

```
CREATE TABLE dbo.Employee
    (EmployeeId INT PRIMARY KEY,
    LastName varchar(50),
    FirstName varchar(50),
    DepartmentId int,
    HireDate datetime,
    TerminationDate datetime,
    SupervisorId int,
    CostCenterNumber int,
    EmployeeStatus int,
    EmployeePayRate int)
) GO
CREATE INDEX IX_1 on dbo.Employee (LastName, FirstName, DepartmentId) INCLUDE (HireDate)
CREATE INDEX IX_2 on dbo.Employee (LastName) INCLUDE (EmployeeId, FirstName, DepartmentId)
CREATE INDEX IX_3 on dbo.Employee (LastName, FirstName) INCLUDE (DepartmentId)
CREATE INDEX IX_4 on dbo.Employee (LastName, FirstName) INCLUDE (HireDate, DepartmentId)
GO

INSERT INTO Employee (EmployeeID, LastName, CostCenterNumber) VALUES(1001, 'Employee A', 3001001)
INSERT INTO Employee (EmployeeID, LastName, CostCenterNumber) VALUES(1002, 'Employee B', 3001001)
GO
```

Application

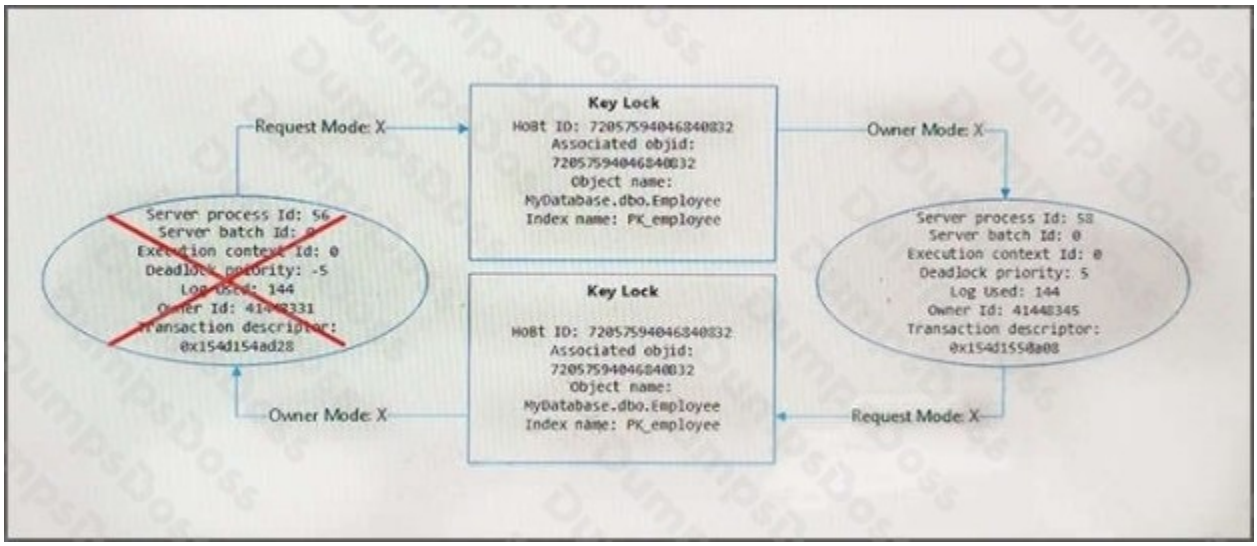
You have an application that updates the Employees table. The application calls the following stored procedures simultaneously and asynchronously: ▪ UspA: This stored procedure updates only the EmployeeStatus column.

▪ UspB: This stored procedure updates only the EmployeePayRate column.

The application uses views to control access to data. Views must meet the following requirements:

▪ Allow users access to all columns in the tables that the view accesses. ▪ Restrict updates to only the rows that the view returns.

Exhibit

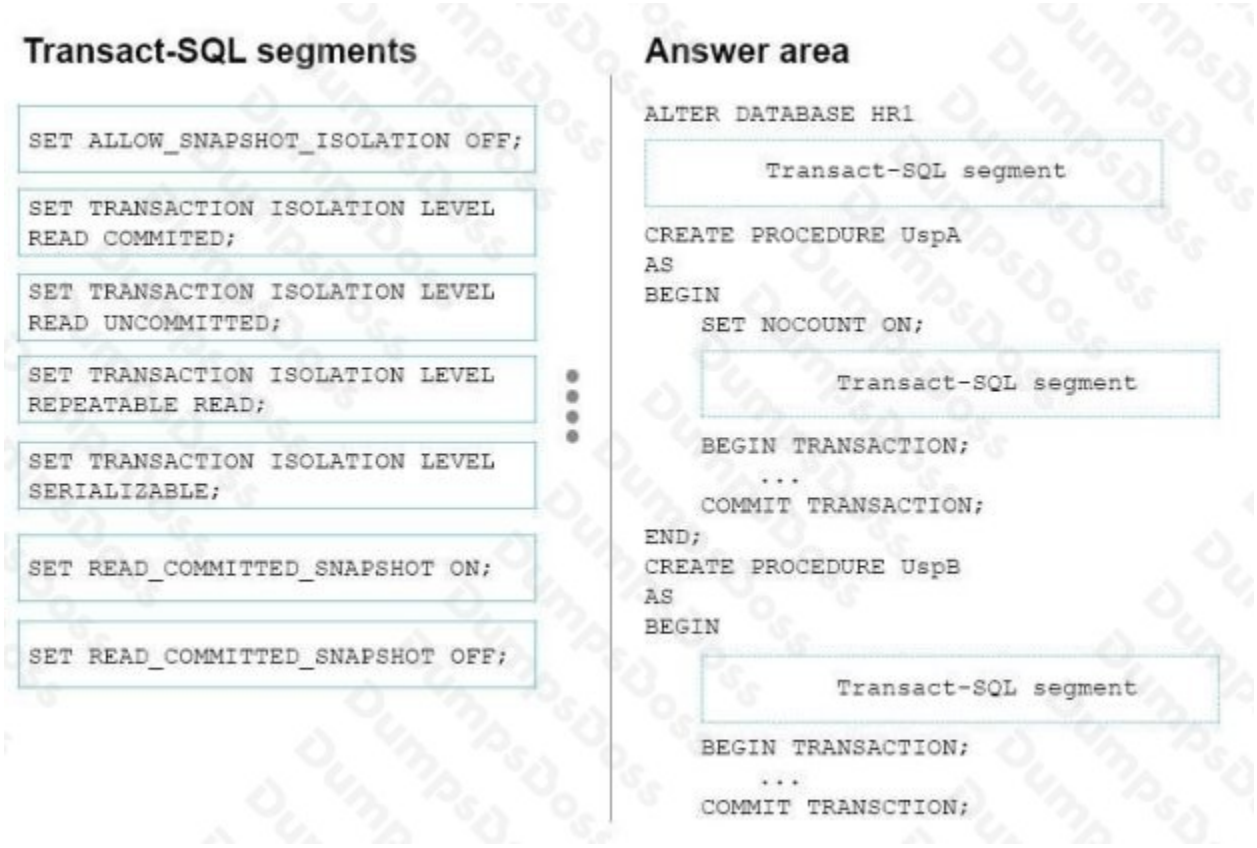


Both of the stored procedures experience blocking issues. UspB must not abort if UspA commits changes to a row before UspB commits changes to the same row. UspA must not abort if UspB commits changes to a row before UspA commits changes to the same row.

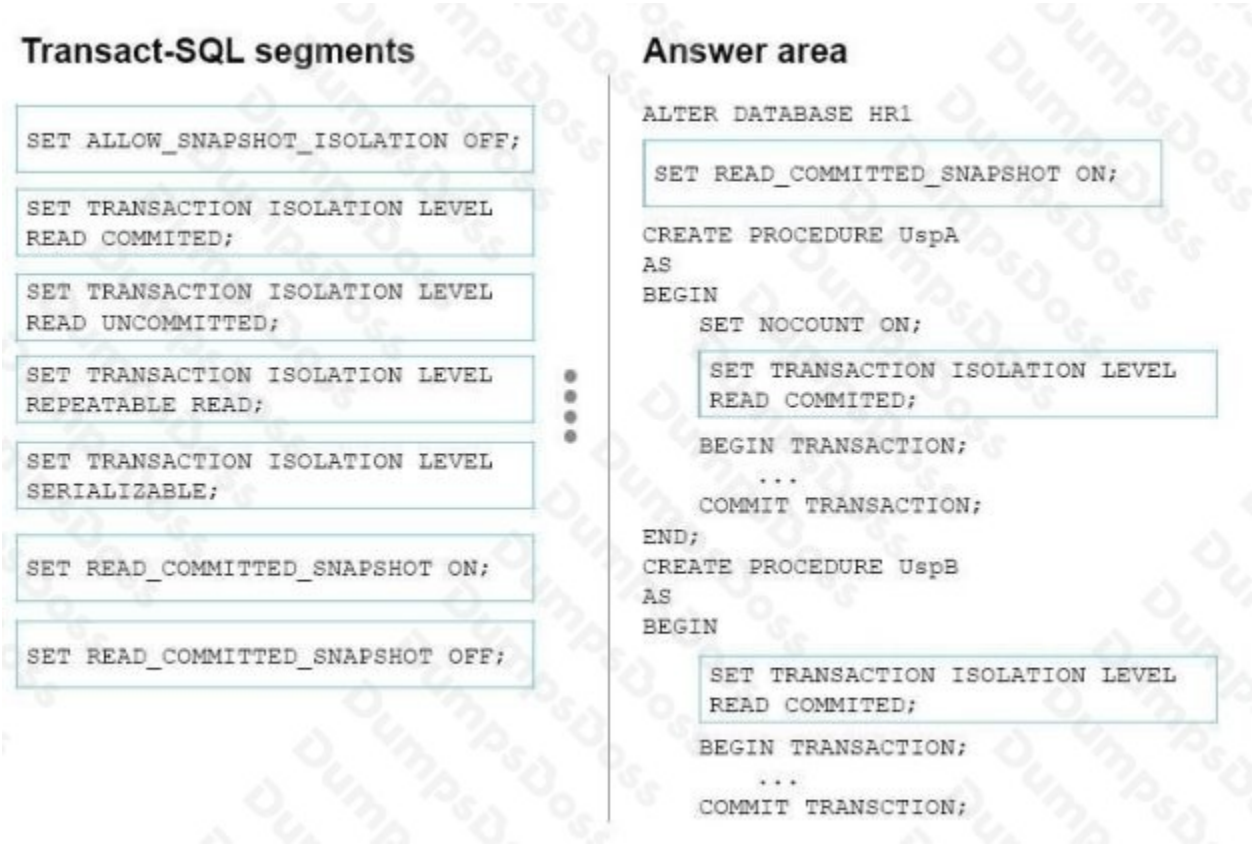
You need to specify the transaction isolation levels to enable row versioning.

How should you complete the Transact-SQL statements? To answer, drag the 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.

Select and Place:


Transact-SQL segments
SET ALLOW_SNAPSHOT_ISOLATION OFF;
SET TRANSACTION ISOLATION LEVEL READ COMMITED;
SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
SET READ_COMMITTED_SNAPSHOT ON;
SET READ_COMMITTED_SNAPSHOT OFF;
Answer area
ALTER DATABASE HR1
Transact-SQL segment
CREATE PROCEDURE UspA
AS
BEGIN
SET NOCOUNT ON;
Transact-SQL segment
BEGIN TRANSACTION;
...
COMMIT TRANSACTION;
END;
CREATE PROCEDURE UspB
AS
BEGIN
Transact-SQL segment
BEGIN TRANSACTION;
...
COMMIT TRANSCTION;

ANSWER:



Explanation:

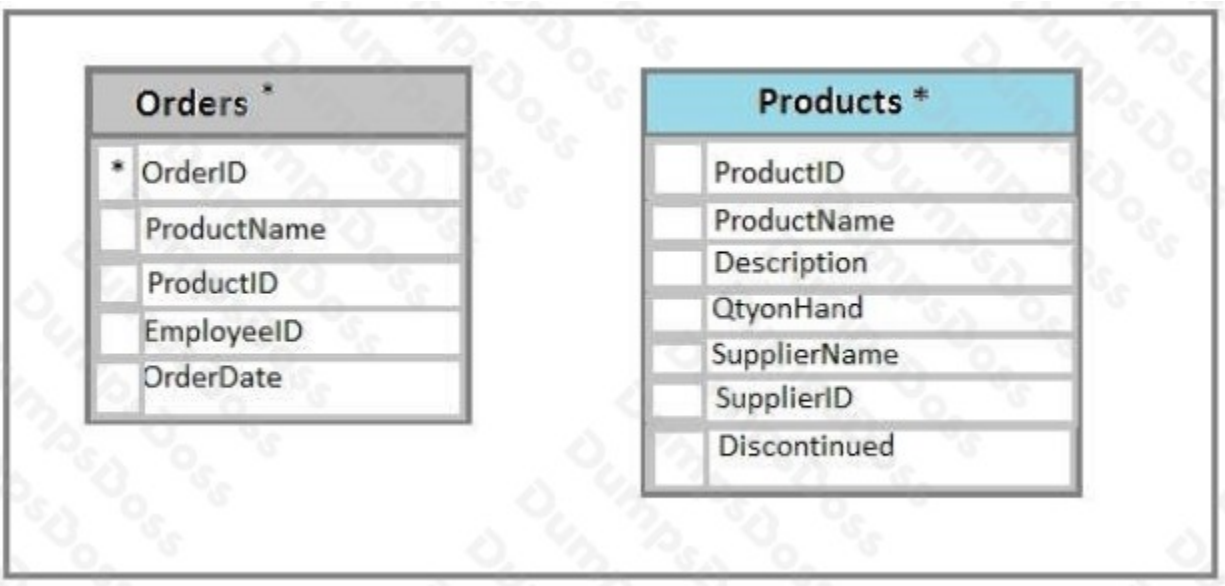
References: https://technet.microsoft.com/en-us/library/ms175095(v=sql.105).aspx https://technet.microsoft.com/en-us/library/ms173763(v=sql.105).aspx

QUESTION NO: 10 - (HOTSPOT)

HOTSPOT

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 Sales that contains the following database tables: Customer, Order, and Products. The Products table and the Order table are shown in the following diagram.



The customer table includes a column that stores the data for the last order that the customer placed.

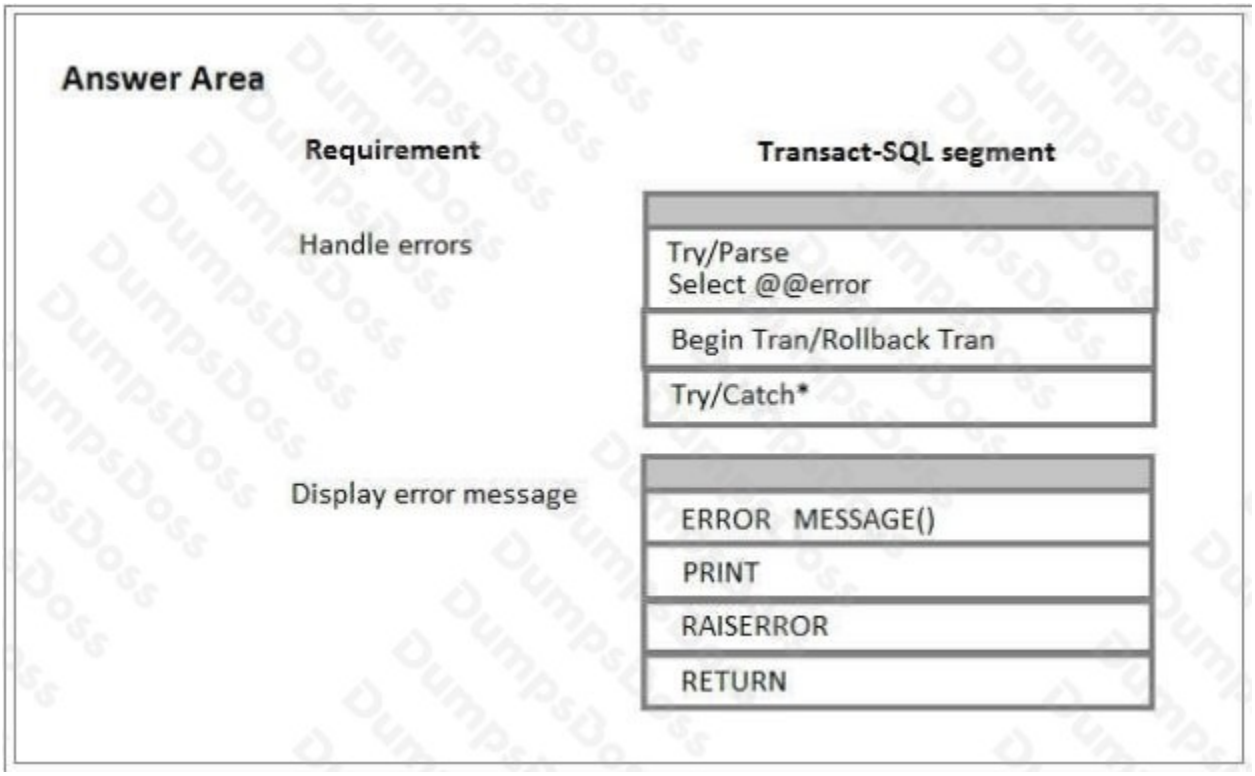
You plan to create a table named Leads. The Leads table is expected to contain approximately 20,000 records. Storage requirements for the Leads table must be minimized.

You need to implement a stored procedure that deletes a discontinued product from the Products table. You identify the following requirements:

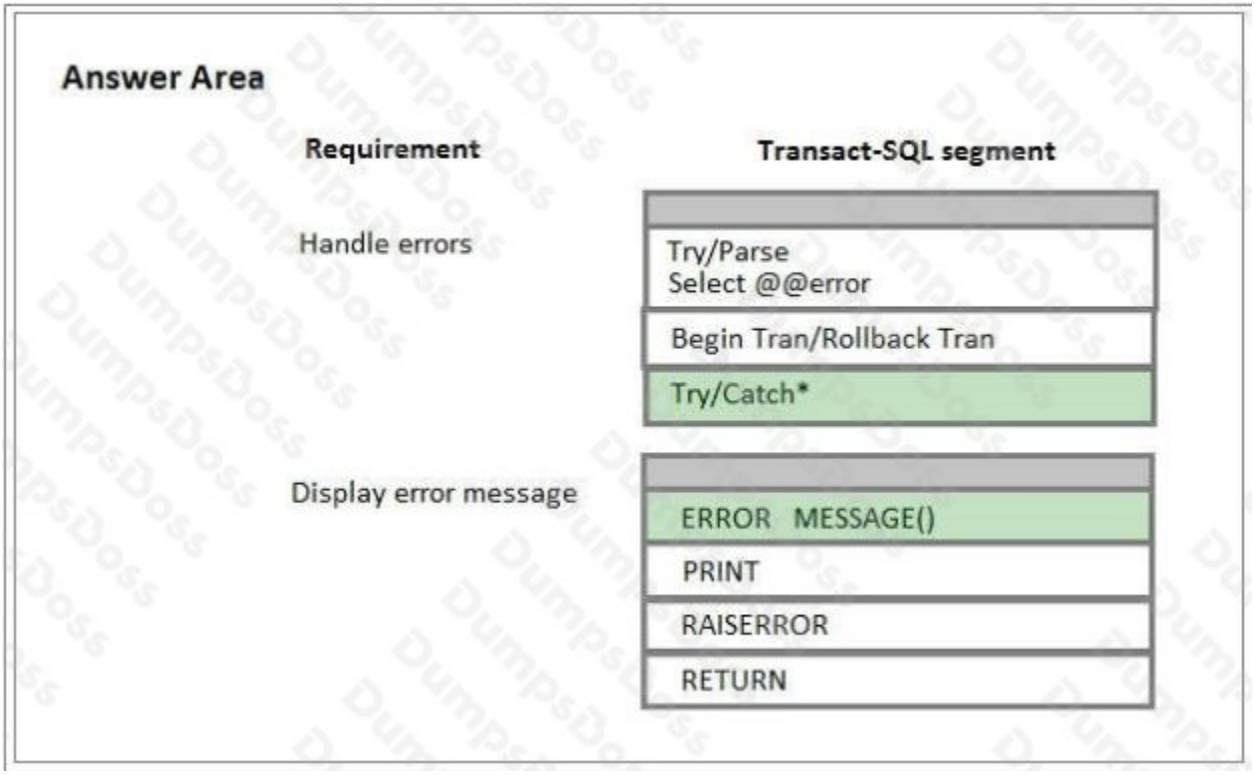
- If an open order includes a discontinued product, the records for the product must not be deleted.
- The stored procedure must return a custom error message if a product record cannot be deleted. The message must identify the OrderID for the open order.

What should you do? To answer, select the appropriate Transact-SQL segments in the answer area.

Hot Area:



ANSWER:



Explanation:

Using TRY...CATCH in Transact-SQL

Errors in Transact-SQL code can be processed by using a TRY…CATCH construct.

TRY…CATCH can use the following error function to capture error information:

ERROR_MESSAGE() returns the complete text of the error message. The text includes the values supplied for any substitutable parameters such as lengths, object names, or times.

References: https://technet.microsoft.com/en-us/library/ms179296(v=sql.105).aspx