

**Microsoft.Premium.70-762.by.VCEplus.85q**

Number: 70-762 VCEplus  
Passing Score: 800  
Time Limit: 120 min  
File Version: 2.0



**Exam Code:** 70-762

**Exam Name:** Developing SQL Databases

**Certification Provider:** Microsoft

**Corresponding Certifications:** MCP, MCSA, MCSA: SQL 2016

**Website:** [www.vceplus.com](http://www.vceplus.com)

**Free Exam:** <https://vceplus.com/exam-70-762/>

Questions & Answers Exam Engine is rigorously checked before being put up for sale. We make sure there is nothing irrelevant in 762 exam products and you get latest questions. We strive to deliver the best 70-762 exam product for top grades in your first attempt.

**VCE to PDF Converter :** <https://vceplus.com/vce-to-pdf/>

**Facebook:** <https://www.facebook.com/VCE.For.All.VN/>

**Twitter :** [https://twitter.com/VCE\\_Plus](https://twitter.com/VCE_Plus)

**Google+ :** <https://plus.google.com/+Vcepluscom>

**LinkedIn :** <https://www.linkedin.com/company/vceplus>

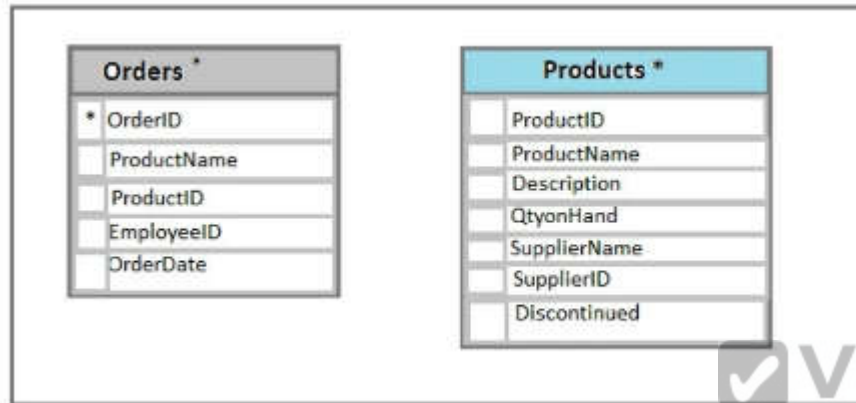
## Exam A

### QUESTION 1

#### 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 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.

Changes to the price of any product must be less a 25 percent increase from the current price. The shipping department must be notified about order and shipping details when an order is entered into the database.

You need to implement the appropriate table objects.

Which object should you use for each table? To answer, drag the appropriate objects to the correct tables. 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**

Foreign key constraint	Instead of trigger
Check constraint	Primary key constraint
Unique constraint	After insert trigger

**Answer Area**

Table	Objects
Orders	
Products	

**Correct Answer:**

**Objects**

	Instead of trigger
Check constraint	
Unique constraint	After insert trigger

**Answer Area**

Table	Objects
Orders	Foreign key constraint
Products	Primary key constraint

**Section: (none)**

**Explanation**

**Explanation/Reference:**

The Products table needs a primary key constraint on the ProductID field.

The Orders table needs a foreign key constraint on the productID field, with a reference to the ProductID field in the Products table

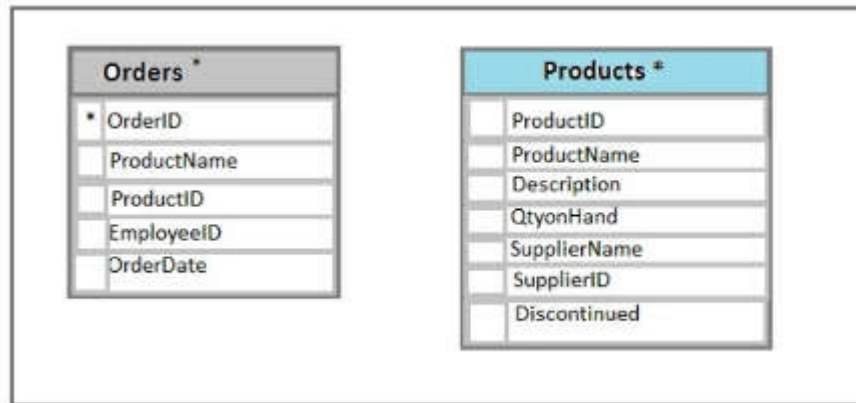
## QUESTION 2

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:**

Requirement	Transact-SQL segment
Handle errors	Try/Parse
	Select @@error
	Begin Tran/Rollback Tran
	Try/Catch*
Display error message	ERROR MESSAGE()
	PRINT
	RAISERROR
	RETURN

Correct Answer:

Requirement	Transact-SQL segment
Handle errors	Try/Parse
	Select @@error
	Begin Tran/Rollback Tran
	Try/Catch*
Display error message	ERROR MESSAGE()
	PRINT
	RAISERROR
	RETURN

**Section: (none)**

**Explanation**

**Explanation/Reference:**

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.

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

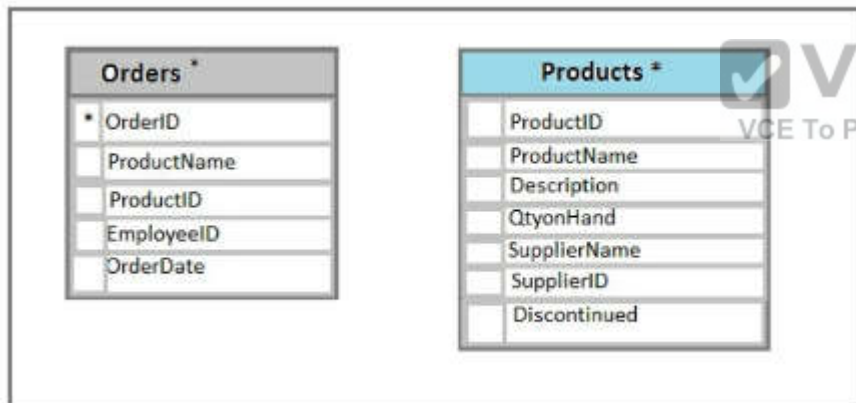
**QUESTION 3**

**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 create triggers that meet the following requirements:

!Optimize the performance and data integrity of the tables.

!Provide a custom error if a user attempts to create an order for a customer that does not exist.

!In the Customers table, update the value for the last order placed.

!Complete all actions as part of the original transaction.

In the table below, identify the trigger types that meet the requirements.

NOTE: Make only selection in each column. Each correct selection is worth one point

Hot Area:

**Answer Area**

Trigger type	Provide custom	Update Customer table
AFTER INSERT trigger	<input type="checkbox"/>	<input type="checkbox"/>
INSTEAD OF INSERT trigger	<input type="checkbox"/>	<input type="checkbox"/>
AFTER UPDATE trigger	<input type="checkbox"/>	<input type="checkbox"/>
INSTEAD OF UPDATE trigger	<input type="checkbox"/>	<input type="checkbox"/>

Correct Answer:

**Answer Area**

Trigger type	Provide custom	Update Customer table
AFTER INSERT trigger	<input checked="" type="checkbox"/>	<input type="checkbox"/>
INSTEAD OF INSERT trigger	<input type="checkbox"/>	<input type="checkbox"/>
AFTER UPDATE trigger	<input type="checkbox"/>	<input checked="" type="checkbox"/>
INSTEAD OF UPDATE trigger	<input type="checkbox"/>	<input type="checkbox"/>

Section: (none)

Explanation

**Explanation/Reference:**

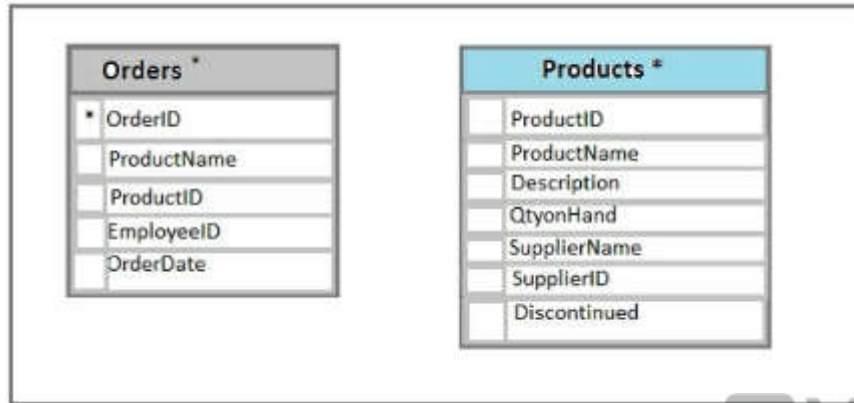
INSTEAD OF INSERT triggers can be defined on a view or table to replace the standard action of the INSERT statement. AFTER specifies that the DML trigger is fired only when all operations specified in the triggering SQL statement have executed successfully.  
[https://technet.microsoft.com/en-us/library/ms175089\(v=sql.105\).aspx](https://technet.microsoft.com/en-us/library/ms175089(v=sql.105).aspx)

#### QUESTION 4

##### 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.

The Leads table must include the columns described in the following table.

Column name	Description
LeadID	This column stores a unique value that is automatically assigned for each lead.
IsCustomer	This column indicates whether the lead is for a current customer.

The data types chosen must consume the least amount of storage possible,

You need to select the appropriate data types for the Leads table.

In the table below, identify the data type that must be used for each table column NOTE: Make only one selection in each column.

**Hot Area:**



**Answer Area**

Data type	LeadID	IsCustomer
smallint	<input type="radio"/>	<input type="radio"/>
int	<input type="radio"/>	<input type="radio"/>
binary	<input type="radio"/>	<input type="radio"/>
numeric	<input type="radio"/>	<input type="radio"/>
bit	<input type="radio"/>	<input type="radio"/>

**Correct Answer:**

**Answer Area**

Data type	LeadID	IsCustomer
smallint	<input type="radio"/>	<input type="radio"/>
int	<input type="radio"/>	<input type="radio"/>
binary	<input type="radio"/>	<input type="radio"/>
numeric	<input type="radio"/>	<input type="radio"/>
bit	<input type="radio"/>	<input type="radio"/>

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Bit is a Transact-SQL integer data type that can take a value of 1,0, or NULL.

Smallint is a Transact-SQL integer data type that can take a value in the range from -32,768 to 32,767. int, bigint, smallint, and tinyint (Transact-SQL) Exact-number data types that use integer data.

Data type	Range	Storage
bigint	-2 <sup>63</sup> (-9,223,372,036,854,775,808) to 2 <sup>63</sup> -1 (9,223,372,036,854,775,807)	8 Bytes
int	-2 <sup>31</sup> (-2,147,483,648) to 2 <sup>31</sup> -1 (2,147,483,647)	4 Bytes
smallint	-2 <sup>15</sup> (-32,768) to 2 <sup>15</sup> -1 (32,767)	2 Bytes
tinyint	0 to 255	1 Byte

References: <https://msdn.microsoft.com/en-us/library/ms187745.aspx>  
<https://msdn.microsoft.com/en-us/library/ms177603.aspx>

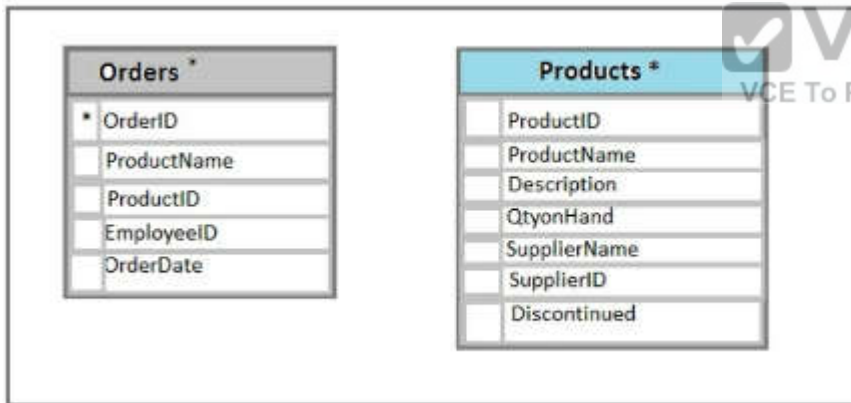
### QUESTION 5

#### 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 modify the database design to meet the following requirements:

- Rows in the Orders table must always have a valid value for the ProductID column.
- Rows in the Products table must not be deleted if they are part of any rows in the Orders table.
- All rows in both tables must be unique.

In the table below, identify the constraint that must be configured for each table.

NOTE: Make only one selection in each column.

Hot Area:

Constraint	Orders table	Products table
Check constraint on OrderID	<input type="radio"/>	<input type="radio"/>
Foreign key constraint on ProductID	<input type="radio"/>	<input type="radio"/>
Check constraint on ProductID	<input type="radio"/>	<input type="radio"/>
Foreign key constraint on OrderID	<input type="radio"/>	<input type="radio"/>

Correct Answer:

Constraint	Orders table	Products table
Check constraint on OrderID	<input type="radio"/>	<input type="radio"/>
Foreign key constraint on ProductID	<input checked="" type="radio"/>	<input type="radio"/>
Check constraint on ProductID	<input type="radio"/>	<input checked="" type="radio"/>
Foreign key constraint on OrderID	<input type="radio"/>	<input type="radio"/>

Section: (none)

Explanation

**Explanation/Reference:**

A FOREIGN KEY in one table points to a PRIMARY KEY in another table. Here the foreign key constraint is put on the ProductID in the Orders, and points to the ProductID of the Products table.

With a check constraint on the ProductID we can ensure that the Products table contains only unique rows.

References: [http://www.w3schools.com/sql/sql\\_foreignkey.asp](http://www.w3schools.com/sql/sql_foreignkey.asp)

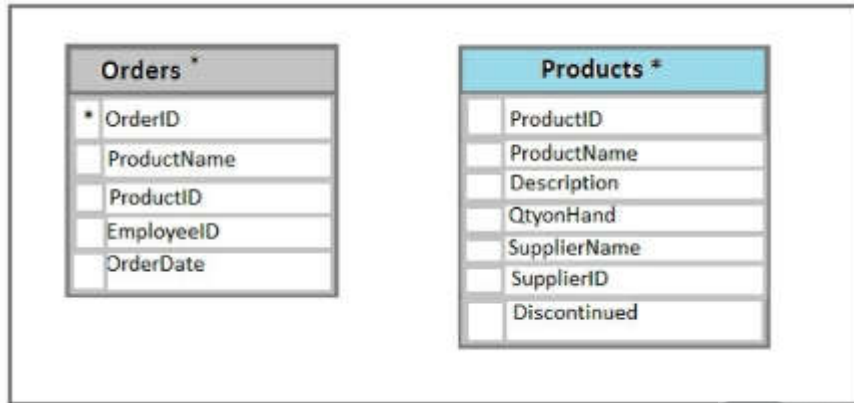
## QUESTION 6

### 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 the series.

Each question presents a different goal and answer choices, but the text of the scenario is exactly the same in each question in the series.

You have a database named Sales that contains the following database tables. Customer, Order, and Products The Products table and the order table shown in the following diagram.



The Customer table includes a column that stores the date 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 begin to modify the table design to adhere to third normal form.

Which column should you remove for each table? To answer? drag the appropriate column names to the correct locations. Each column name 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:**

**Columns**

ProductID
ProductName
Description
EmployeeID
OrderDate
SupplierName
SupplierID
Discontinued

**Answer Area**

Table	Column to remove
Products	Column
Orders	Column

**Correct Answer:**

The screenshot shows a database normalization tool interface. On the left, under the heading "Columns", there is a list of columns for a table: ProductID, Description, EmployeeID, OrderDate, SupplierID, and Discontinued. On the right, under the heading "Answer Area", there is a section titled "Table" with two tables listed: "Products" and "Orders". To the right of the "Table" section, under the heading "Column to remove", there is a list of columns: SupplierName and ProductName. The interface also features several "VCEplus" watermarks and logos.

**Section: (none)**

**Explanation**

**Explanation/Reference:**

In the Products table the SupplierName is dependant on the SupplierID, not on the ProductID.

In the Orders table the ProductName is dependant on the ProductID, not on the OrderID,

Note:

A table is in third normal form when the following conditions are met:

\*It is in second normal form.

\*All nonprimary fields are dependent on the primary key.

Second normal form states that it should meet all the rules for First 1 Nominal Form and there must be no partial dependences of any of the columns on the primary key.

First normal form (1NF) sets the very basic rules for an organized database:

\*Define the data items required, because they become the columns in a table Place related data items in a table.

\*Ensure that there are no repeating groups of data.

\*Ensure that there is a primary key.

<https://www.tutorialspoint.com/sql/third-normal-form.htm>

**QUESTION 7**  
**HOTSPOT**

```
CREATE TABLE BlogCategory
(
    CategoryID int NOT NULL PRIMARY KEY,
    CategoryName nvarchar (20)
);

CREATE TABLE BlogEntry
(
    Entry int NOT PRIMARY KEY,
    Entrytitle nvarchar (50),
    Category int NOT NULL FOREIGN KEY REFERENCES BlogCategory
(CategoryID)
);

CREATE TABLE dbo.ProductReview
(
    ProductReviewID IDENTITY(1,1) PRIMARY KEY,
    Product int NOT NULL,
    Review varchar (1000) NOT NULL
);

CREATE TABLE dbo.Product
(
    ProductID int Identity(1,1) PRIMARY KEY,
    Name varchar(1000) NOT NULL
);

CREATE TABLE dbo.SalesPerson
(
    SalesPersonID int IDENTITY(1,1) PRIMARY KEY,
    Name varchar (1000) NOT NULL,
    SalesID Money
)
```



You must modify the ProductReview Table to meet the following requirements:

- 1 The table must reference the ProductID column in the Product table
- 2.Existing records in the ProductReview table must not be validated with the Product table.
- 3.Deleting records in the Product table must not be allowed if records are referenced by the ProductReview table.
- 4.Changes to records in the Product table must propagate to the ProductReview table.

You also have the following database tables: Order, ProductTypes, and SalesHistory, The transact-SQL statements for these tables are not available.

You must modify the Orders table to meet the following requirements:

- 1,Create new rows in the table without granting INSERT permissions to the table.
- 2.Notify the sales person who places an order whether or not the order was completed.

You must add the following constraints to the SalesHistory table:

- a constraint on the SaleID column that allows the field to be used as a record identifier
- a constant that uses the ProductID column to reference the Product column of the ProductTypes table
- a constraint on the CategoryID column that allows one row with a null value in the column

-a constraint that limits the SalePrice column to values greater than four

Finance department users must be able to retrieve data from the SalesHistory table for sales persons where the value of the SalesYTD column is above a certain threshold.

You plan to create a memory-optimized table named SalesOrder. The table must meet the following requirements:

-The table must hold 10 million unique sales orders.

-The table must use checkpoints to minimize I/O operations and must not use transaction logging.

-Data loss is acceptable.

Performance for queries against the SalesOrder table that use Where clauses with exact equality operations must be optimized.

You need to create an object that allows finance users to be able to retrieve the required data. The object must not have a negative performance impact.

How should you complete the Transact-SQL statements? To answer, select the appropriate Transact-SQL segments in the answer area.

**Hot Area:**

**Answer Area**

CREATE **FUNCTION** Sales.YIDSalesByPerson

PROCEDURE  
TRIGGER  
VIEW

(@SalesPersonID int, @minYIDSales money)

**RETURNS TABLE**  
WITH SCHEMABINDING  
WITH ENCRYPTION  
RETURNS INT

AS  
RETURN (SELECT TOP(@SalesPersonID) BusinessEntityID, SalesYID  
FROM Sales.SalesPerson  
WHERE SalesYID > @minYIDSales  
ORDER BY SalesYID desc);

**Correct Answer:**



**Answer Area**

CREATE **FUNCTION** Sales.YIDSalesByPerson

PROCEDURE  
TRIGGER  
**VIEW**

(@SalesPersonID int, @minYIDSales money)

**RETURNS TABLE**  
**WITH SCHEMABINDING**  
WITH ENCRYPTION  
RETURNS INT

AS  
RETURN (SELECT TOP(@SalesPersonID) BusinessEntityID, SalesYID  
FROM Sales.SalesPerson  
WHERE SalesYID > @minYIDSales  
ORDER BY SalesYID desc);

**Section: (none)**

**Explanation**

**Explanation/Reference:**

From question: Finance department users must be able to retrieve data from the SalesHistory table for sales persons where the value of the SalesYTD column is above a certain threshold.

CREATE VIEW (Transact-SQL) creates a virtual table whose contents (columns and rows) are defined by a query. Use this statement to create a view of the data in one or more tables in the database.

SCHEMABINDING binds the view to the schema of the underlying table or tables. When SCHEMABINDING is specified, the base table or tables cannot be modified in a way that would affect the view definition.

<https://msdn.microsoft.com/en-us/library/ms187956.aspx>

## QUESTION 8

### 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 that contains the following tables: BlogCategory, BlogEntry, ProductReview, Product, and Salesperson. The tables were created using the following Transact SQL statements:

```

CREATE TABLE BlogCategory
(
    CategoryID int NOT NULL PRIMARY KEY,
    CategoryName nvarchar (20)
);

CREATE TABLE BlogEntry
(
    Entry int NOT PRIMARY KEY,
    Entrytitle nvarchar (50),
    Category int NOT NULL FOREIGN KEY REFERENCES BlogCategory
(CategoryID)
);

CREATE TABLE dbo.ProductReview
(
    ProductReviewID IDENTITY(1,1) PRIMARY KEY,
    Product int NOT NULL,
    Review varchar (1000) NOT NULL
);

CREATE TABLE dbo.Product
(
    ProductID int Identity(1,1) PRIMARY KEY,
    Name varchar(1000) NOT NULL
);

CREATE TABLE dbo.SalesPerson
(
    SalesPersonID int IDENTITY(1,1) PRIMARY KEY,
    Name varchar (1000) NOT NULL,
    SalesID Money
)

```

You must modify the ProductReview Table to meet the following requirements:

- 1.The table must reference the ProductID column in the Product table
- 2.Existing records in the ProductReview table must not be validated with the Product table.
- 3.Deleting records in the Product table must not be allowed if records are referenced by the ProductReview table.
- 4.Changes to records in the Product table must propagate to the ProductReview table.

You also have the following database tables: Order, ProductTypes, and SalesHistory, The transact-SQL statements for these tables are not available.

You must modify the Orders table to meet the following requirements:

- 1.Create new rows in the table without granting INSERT permissions to the table.
- 2.Notify the sales person who places an order whether or not the order was completed.

You must add the following constraints to the SalesHistory table:

- a constraint on the SaleID column that allows the field to be used as a record identifier
- a constant that uses the ProductID column to reference the Product column of the ProductTypes table
- a constraint on the CategoryID column that allows one row with a null value in the column
- a constraint that limits the SalePrice column to values greater than four

Finance department users must be able to retrieve data from the SalesHistory table for sales persons where the value of the SalesYTD column is above a certain threshold.

You plan to create a memory-optimized table named SalesOrder. The table must meet the following requirements:

- The table must hold 10 million unique sales orders.
- The table must use checkpoints to minimize I/O operations and must not use transaction logging.
- Data loss is acceptable.

Performance for queries against the SalesOrder table that use Where clauses with exact equality operations must be optimized.  
You need to update the SalesHistory table  
How should you complete the Transact\_SQL statement? To answer? select the appropriate Transact-SQL, segments in the answer area.

**Hot Area:**



**Answer Area**

VCE To PDF - Free Practice Exam

```
IF OBJECT_ID(*SalesHistory*)>0 DROP TABLE SalesHistory
GO
IF OBJECT_ID(*ProductTypes*)>0 DROP TABLE ProductTypes
GO
CREATE TABLE ProductTypes
(
    ProductID SMALLINT,
    ProductDescription VARCHAR(255),
    CONSTRAINT pk_ProductID PRIMARY KEY (ProductID)
)
GO
CREATE TABLE [dbp].[SalesHistoryK]
[SalesID] [int]
```

▼

IDENTITY(1,4)  
IDENTITY(1,4) NOT NULL PRIMARY KEY  
UNIQUE

```
[ProductID] SMALLINT NULL
[SalesDate] [datetime] NULL
[SalesPrice] [money]
```

▼

NOT NULL  
NULL CHECK (SalesPrice > 4)  
UNIQUE

```
[CategoryID] [smallint]
```

▼

NOT NULL  
NULL CHECK (SalesPrice > 4)  
UNIQUE

▼

CONSTRAINT fk\_SalesHistoryProductID FOREIGN KEY (ProductID) REFERENCES SalesHistory(CategoryID)  
CONSTRAINT fk\_SalesHistoryProductID FOREIGN KEY (ProductID) REFERENCES ProductTypes(ProductID)

```
)
GO
```

**Correct Answer:**



**Answer Area**

VCE To PDF - Free Practice Exam

```
IF OBJECT_ID(*SalesHistory*)>0 DROP TABLE SalesHistory
GO
IF OBJECT_ID(*ProductTypes*)>0 DROP TABLE ProductTypes
GO
CREATE TABLE ProductTypes
(
    ProductID SMALLINT,
    ProductDescription VARCHAR(255),
    CONSTRAINT pk_ProductID PRIMARY KEY (ProductID)
)
GO
CREATE TABLE [dbp].[SalesHistoryK]
[SalesID] [int]
```

IDENTITY(1,4)  
IDENTITY(1,4) NOT NULL PRIMARY KEY  
UNIQUE

```
[ProductID] SMALLINT NULL
[SalesDate] [datetime] NULL
[SalesPrice] [money]
```

NOT NULL  
NULL CHECK (SalesPrice > 4)  
UNIQUE

```
[CategoryID] [smallint]
```

NOT NULL  
NULL CHECK (SalesPrice > 4)  
UNIQUE

```
CONSTRAINT fk_SalesHistoryProductID FOREIGN KEY (ProductID) REFERENCES SalesHistory(CategoryID)
CONSTRAINT fk_SalesHistoryProductID FOREIGN KEY (ProductID) REFERENCES ProductTypes(ProductID)
```

```
)
GO
```

**Section: (none)**

**Explanation**

**Explanation/Reference:**

### QUESTION 9

#### 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 that contains the following tables: BlogCategory, BlogEntry, ProductReview, Product, and Salesperson. The tables were created using the following Transact SQL statements:

```
CREATE TABLE BlogCategory
(
    CategoryID int NOT NULL PRIMARY KEY,
    CategoryName nvarchar (20)
);

CREATE TABLE BlogEntry
(
    Entry int NOT PRIMARY KEY,
    Entrytitle nvarchar (50),
    Category int NOT NULL FOREIGN KEY REFERENCES BlogCategory
(CategoryID)
);

CREATE TABLE dbo.ProductReview
(
    ProductReviewID IDENTITY(1,1) PRIMARY KEY,
    Product int NOT NULL,
    Review varchar (1000) NOT NULL
);

CREATE TABLE dbo.Product
(
    ProductID int Identity(1,1) PRIMARY KEY,
    Name varchar(1000) NOT NULL
);

CREATE TABLE dbo.SalesPerson
(
    SalesPersonID int IDENTITY(1,1) PRIMARY KEY,
    Name varchar (1000) NOT NULL,
    SalesID Money
);
```

You must modify the ProductReview Table to meet the following requirements:

1.The table must reference the ProductID column in the Product table

2.Existing records in the ProductReview table must not be validated with the Product table.  
3.Deleting records in the Product table must not be allowed if records are referenced by the ProductReview table.  
4.Changes to records in the Product table must propagate to the ProductReview table.  
You also have the following database tables: Order, ProductTypes, and SalesHistory, The transact-SQL statements for these tables are not available.

You must modify the Orders table to meet the following requirements:

- 1.Create new rows in the table without granting INSERT permissions to the table.
- 2.Notify the sales person v/ho places an order whether or not :he order was completed.

You must add the following constraints to the SalesHistory table:

- a constraint on the SaleID column that allows the field to be used as a record identifier
- a constant that uses the ProductID column to reference the Product column of the ProductTypes table
- a constraint on the CategoryID column that allows one row with a null value in the column
- a constraint that limits the SalePrice column to values greater than four

Finance department users must be able to retrieve data from the SalesHistory table for sales persons where the value of the SalesYTD column is above a certain threshold.

You plan to create a memory-optimized table named SalesOrder. The table must meet the following requirments:

- The table must hold 10 million unique sales orders.
- The table must use checkpoints to minimize I/O operations and must not use transaction logging.
- Data loss is acceptable.

Performance for queries against the SalesOrder table that use Where clauses with exact equality operations must be optimized. You need to create the Sales Order table

How should you complete the table definition? To answer? select the appropriate Transact-SQL segments in the answer area.

**Hot Area:**

**Answer Area**

```
CREATE TABLE dbo.SalesOrder
(
    SalesOrderID int Primary KEY NONSELECTED HASH WITH (BUCKET_COUNT = 10000000),
    SalesOrderCode char(5) NOT NULL INDEX IX_SalesOrder,
    EmployeeID int not null,
    CustomerID int not null,
    SalesAmount money
)
WITH
    (Memory_OPTIMIZED = ON),
    DURABILITY =
```

SCHEMA\_ONLY  
SCHEMA\_AND\_DATA

CLUSTERED  
NONCLUSTERED  
NONCLUSTERED HASH WITH (BUCKET\_COUNT = 10000)  
NONCLUSTERED HASH WITH (BUCKET\_COUNT = 10000000)

**Correct Answer:**



**Answer Area**

```
CREATE TABLE dbo.SalesOrder
(
    SalesOrderID int Primary KEY NONSELECTED HASH WITH (BUCKET_COUNT = 10000000),
    SalesOrderCode char(5) NOT NULL INDEX IX_SalesOrder
    EmployeeID int not null,
    CustomerID int not null,
    SalesAmount money
)
WITH
    (Memory_OPTIMIZED = ON),
    DURABILITY =
```

SCHEMA\_ONLY

SCHEMA\_AND\_DATA

CLUSTERED

NONCLUSTERED

NONCLUSTERED HASH WITH (BUCKET\_COUNT = 10000)

NONCLUSTERED HASH WITH (BUCKET\_COUNT = 10000000)

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Box 1: NONCLUSTERED HASHWITH (BUCKET.COUNT = 10000000)

Hash index is preferable over a nonclustered index when queries test the indexed columns by use of a WHERE clause with an exact equality on all index key columns. We should use a bucket count of 10 million.

Box 2: SCHEMA\_ONLY

Durability: The value of SCHEMA\_AND\_DATA indicates that the table is durable, meaning that changes are persisted on disk and survive restart or failover. SCHEMA\_AND\_DATA is the default value

The value of SCHEMA\_ONLY indicates that the table is non-durable. The table schema is persisted but any data updates are not persisted upon a restart or failover of the database. DURABILITY=SCHEMA\_ONLY is only allowed with MEMORY\_OPTIMIZED=ON

<https://msdn.microsoft.com/en-us/library/mt670614.aspx>

## QUESTION 10

### 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 that contains the following tables: BlogCategory, BlogEntry, ProductReview, Product, and Salesperson. The tables were created using the following Transact SQL statements:

```
CREATE TABLE BlogCategory
(
    CategoryID int NOT NULL PRIMARY KEY,
    CategoryName nvarchar (20)
);

CREATE TABLE BlogEntry
(
    Entry int NOT PRIMARY KEY,
    Entrytitle nvarchar (50),
    Category int NOT NULL FOREIGN KEY REFERENCES BlogCategory
(CategoryID)
);

CREATE TABLE dbo.ProductReview
(
    ProductReviewID IDENTITY(1,1) PRIMARY KEY,
    Product int NOT NULL,
    Review varchar (1000) NOT NULL
);

CREATE TABLE dbo.Product
(
    ProductID int Identity(1,1) PRIMARY KEY,
    Name varchar(1000) NOT NULL
);

CREATE TABLE dbo.SalesPerson
(
    SalesPersonID int IDENTITY(1,1) PRIMARY KEY,
    Name varchar (1000) NOT NULL,
    SalesID Money
)
```



You must modify the ProductReview Table to meet the following requirements:

- 1.The table must reference the ProductID column in the Product table
- 2.Existing records in the ProductReview table must not be validated with the Product table.
- 3.Deleting records in the Product table must not be allowed if records are referenced by the ProductReview table.
- 4.Changes to records in the Product table must propagate to the ProductReview table.

You also have the following database tables: Order, ProductTypes, and SalesHistory, The transact-SQL statements for these tables are not available.

You must modify the Orders table to meet the following requirements:

- 1.Create new rows in the table without granting INSERT permissions to the table.
- 2.Notify the sales person who places an order whether or not the order was completed.

You must add the following constraints to the SalesHistory table:

- a constraint on the SaleID column that allows the field to be used as a record identifier
- a constant that uses the ProductID column to reference the Product column of the ProductTypes table
- a constraint on the CategoryID column that allows one row with a null value in the column
- a constraint that limits the SalePrice column to values greater than four

Finance department users must be able to retrieve data from the SalesHistory table for sales persons where the value of the SalesYTD column is above a certain

threshold.

You plan to create a memory-optimized table named SalesOrder. The table must meet the following requirements:

- The table must hold 10 million unique sales orders.
- The table must use checkpoints to minimize I/O operations and must not use transaction logging.
- Data loss is acceptable.

Performance for queries against the SalesOrder table that use Where clauses with exact equality operations must be optimized.

You need to create a stored procedure named spDeleteCategory to delete records in the database. The stored procedure must meet the following requirements:

- 1.Delete records in both the BlogEntry and BlogCategory tables where CategoryId equals parameter @CategoryId.
- 2.Avoid locking the entire table when deleting records from the BlogCategory table.
- 3.If an error occurs during a delete operation on either table, all changes must be rolled back, otherwise all changes should be committed.

How should you complete the procedure? To answer, select the appropriate Transact-SQL segments in the answer area.

**Hot Area:**

**Answer Area**

```

CREATE PROCEDURE spDeleteCategory
    (@CategoryID int)
AS
BEGIN
    SET NOCOUNT ON;

    SET IMPLICIT_TRANSACTIONS ON
    SET IMPLICIT_TRANSACTIONS OFF
    SET TRANSACTION ISOLATION LEVEL READ COMMITTED
    SET TRANSACTION ISOLATION LEVEL SNAPSHOT

    BEGIN TRY
        DELETE FROM BlogEntry WHERE CategoryID = @CategoryID;
        ...
        DELETE FROM BlogCategory
        WITH (
            ROWLOCK
            TABLOCKX
        ) WHERE CategoryId = @CategoryID;

        IF @@TRANCOUNT = 0
        BEGIN
            BEGIN TRANSACTION;
            COMMIT;
        END

    END TRY
    BEGIN CATCH
        IF @@TRANCOUNT > 0
        BEGIN
            BEGIN TRANSACTION;
            COMMIT;
            ROLLBACK;
        END
    END
END
    
```

**Correct Answer:**

**Answer Area**

```

CREATE PROCEDURE spDeleteCategory
(@CategoryID int)
AS
BEGIN
    SET NOCOUNT ON;

    SET IMPLICIT_TRANSACTIONS ON
    SET IMPLICIT_TRANSACTIONS OFF
    SET TRANSACTION ISOLATION LEVEL READ COMMITTED
    SET TRANSACTION ISOLATION LEVEL SNAPSHOT

    BEGIN TRY
        DELETE FROM BlogEntry WHERE CategoryID = @CategoryID;
        ...
        DELETE FROM BlogCategory
        WITH (ROWLOCK, TABLOCKX) WHERE CategoryId = @CategoryID;

        IF @@TRANCOUNT = 0
        BEGIN
            COMMIT
        END

    END TRY
    BEGIN CATCH
        IF @@TRANCOUNT > 0
        BEGIN
            COMMIT
            ROLLBACK
        END
    END
END
    
```

Section: (none)  
Explanation

Explanation/Reference:

# QUESTION 11

## DRAG DROP

You are analyzing the performance of a database environment.

Applications that access the database are experiencing locks that are held for a large amount of time. You are experiencing isolation phenomena such as dirty, nonrepeatable and phantom reads.

You need to identify the impact of specific transaction isolation levels on the concurrency and consistency of data.

What are the consistency and concurrency implications of each transaction isolation level? To answer, drag the appropriate isolation levels to the correct locations.

Each isolation level 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:

Isolation levels

read committed

serializable

read uncommitted

repeatable read

Concurency

Isolation Level

Isolation level

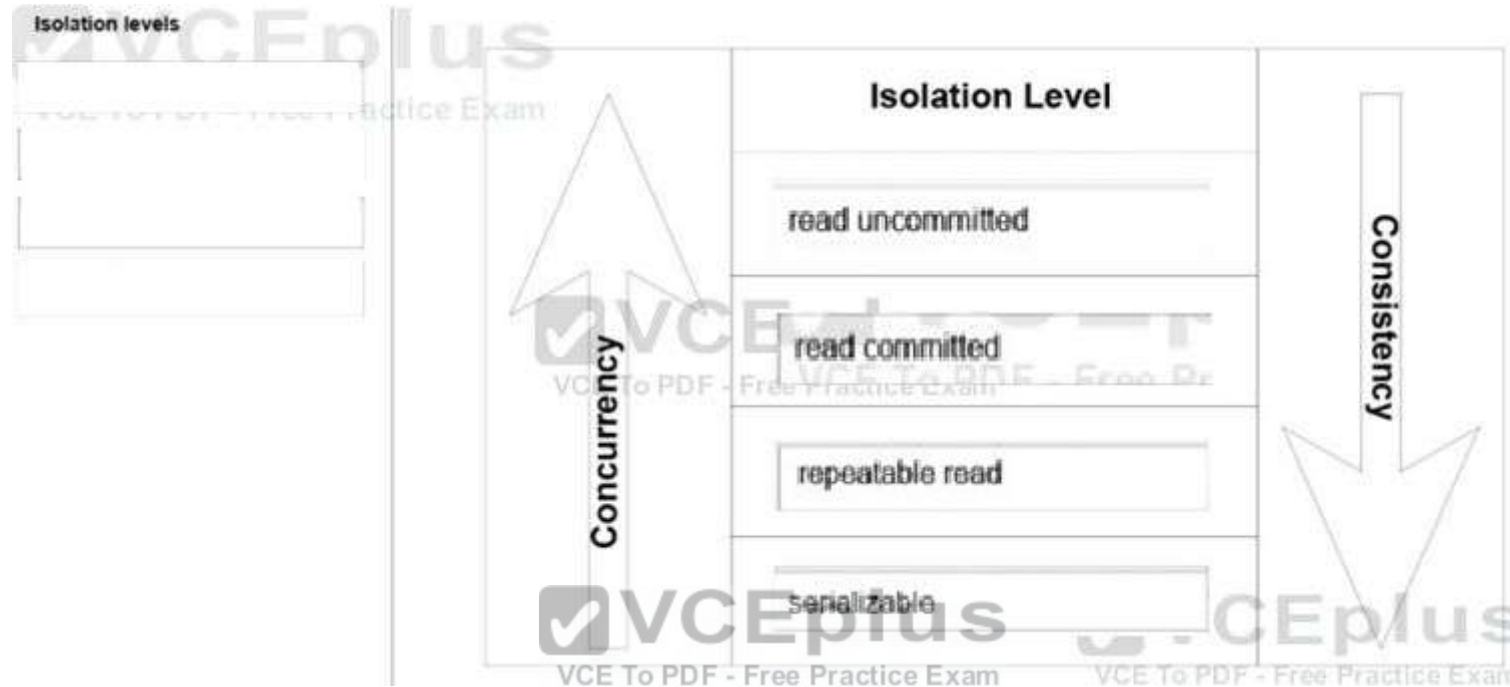
Isolation level

Isolation level

Isolation level

Consistency

Correct Answer:



**Section: (none)**

**Explanation**

**Explanation/Reference:**

**Read Uncommitted (aka dirty read):** A transaction T1 executing under this isolation level can access data changed by concurrent transaction(s).

**Pros:** No read locks needed to read data (i.e. no reader/writer blocking). Note, T1 still takes transaction duration locks for any data modified.

**Cons:** Data is not guaranteed to be transactionally consistent.

**Read Committed:** A transaction T1 executing under this isolation level can only access committed data.

**Pros:** Good compromise between concurrency and consistency.

**Cons:** Locking and blocking. The data can change when accessed multiple times within the same transaction.

**Repeatable Read:** A transaction T1 executing under this isolation level can only access committed data with an additional guarantee that any data read cannot change (i.e. it is repeatable) for the duration of the transaction.

**Pros:** Higher data consistency.

**Cons:** Locking and blocking. The S locks are held for the duration of the transaction that can lower the concurrency. It does not protect against phantom rows.

**Serializable:** A transaction T1 executing under this isolation level provides the highest data consistency including elimination of phantoms but at the cost of reduced concurrency. It prevents phantoms by taking a range lock or table level lock if range lock can't be acquired (i.e. no index on the predicate column) for the duration of the transaction.

**Pros:** Full data consistency including phantom protection.



Cons: Locking and blocking. The S locks are held for the duration of the transaction that can lower the concurrency.  
<https://blogs.msdn.microsoft.com/sqlcat/2011/02/20/concurrency-series-basics-of-transaction-isolation-levels/>

## QUESTION 12

### DRAG DROP

You are evaluating the performance of a database environment

You must avoid unnecessary locks and ensure that lost updates do not occur.

You need to choose the transaction isolation level for each data scenario.

Which isolation level should you use for each scenario? To answer, drag the appropriate isolation levels to the correct scenarios. Each isolation 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:

Isolation levels	Scenario	Isolation levels
read committed	Reading accurate data is top priority. Select statements will wait until any transaction that currently owns the data has been committed or rolled back before returning the value	Isolation level
serializable	Performance is top priority. The work and memory required by the Microsoft SQL Server lock manager is reduced	Isolation level
read uncommitted	The same select statement is issued multiple times within a transaction and the same result are returned. New records are allowed to be inserted into the table referenced by the Select statement	Isolation level
repeatable read		

Correct Answer:



### Isolation levels

- read committed
- serializable
- read uncommitted
- repeatable read

### Answer area

#### Scenario

Reading accurate data is top priority. Select statements will wait until any transaction that currently owns the data has been committed or rolled back before returning the value

Performance is top priority. The work and memory required by the Microsoft SQL Server lock manager is reduced

The same select statement is issued multiple times within a transaction and the same result are returned. New records are allowed to be inserted into the table referenced by the Select statement

#### Isolation levels

read committed

read committed

serializable

**Section: (none)**  
**Explanation**

**Explanation/Reference:**

#### QUESTION 13

##### DRAG DROP

You have two database tables. Table1 is a partitioned table and Table 2 is a nonpartitioned table.

Users report that queries take a long time to complete. You monitor queries by using Microsoft SQL Server Profiler. You observe lock escalation for Table1 and Table 2.

You need to allow escalation of Table1 locks to the partition level and prevent all lock escalation for Table2.

Which Transact-SQL statement should you run for each table? To answer, drag the appropriate Transact-SQL statements to the correct tables. Each command 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 statements**

DBCC TRACEON(1211, -1)

DBCC TRACEON(1224, -1)

ALTER TABLE Table1 SET (LOCK\_ESCALATION = DISABLE)

ALTER TABLE Table2 SET (LOCK\_ESCALATION = DISABLE)

ALTER TABLE Table1 SET (LOCK\_ESCALATION = AUTO)

ALTER TABLE Table2 SET (LOCK\_ESCALATION = AUTO)

ALTER TABLE Table1 SET (LOCK\_ESCALATION = TABLE)

ALTER TABLE Table2 SET (LOCK\_ESCALATION = TABLE)

**Answer Area**

**Table**

Table1

Table2

**Transaction**

Transact-SQL statement

Transact-SQL statement

**Correct Answer:**

**Transact-SQL statements**

DBCC TRACEON(1211, -1)

DBCC TRACEON(1224, -1)

ALTER TABLE Table1 SET (LOCK\_ESCALATION = DISABLE)

ALTER TABLE Table2 SET (LOCK\_ESCALATION = AUTO)

ALTER TABLE Table1 SET (LOCK\_ESCALATION = TABLE)

ALTER TABLE Table2 SET (LOCK\_ESCALATION = TABLE)

**Answer Area**

Table	Transaction
ALTER TABLE Table2 SET (LOCK_ESCALATION = DISABLE)	
ALTER TABLE Table1 SET (LOCK_ESCALATION = AUTO)	

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Since SQL Server 2008 you can also control how SQL Server performs the Lock Escalation - through the ALTER TABLE statement and the property LOCK\_ESCALATION. There are 3 different options available:

TABLE  
AUTO  
DISABLE

Box 1: Table, Auto

The default option is TABLE, means that SQL Server \*always\* performs the Lock Escalation to the table level -even when the table is partitioned. If you have your table partitioned, and you want to have a Partition Level Lock Escalation (because you have tested your data access pattern, and you don't cause deadlocks with it), then you can change the option to AUTO. AUTO means that the Lock Escalation is performed to the partition level, if the table is partitioned, and otherwise to the table level.

Box 2: Table 2, DISABLE

With the option DISABLE you can completely disable the Lock Escalation for that specific table.

For partitioned tables, use the LOCK\_ESCALATION option of ALTER TABLE to escalate locks to the HoBT level instead of the table or to disable lock escalation.  
<http://www.sqlpassion.at/archive/2014/02/25/lock-escalations/>

#### **QUESTION 14**

##### **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 them in the correct order.

**Select and Place:**

Transact-SQL segments

```
BEGIN CATCH
IF (XACT_STATE() != 0)
    ROLLBACK TRANSACTION
END CATCH
```

```
IF (@TRANCOUNT > 0)
    ROLLBACK TRANSACTION
```

```
BEGIN TRAN
```

```
EXEC dbo.Proc1
EXEC dbo.Proc2
EXEC dbo.Proc3
```

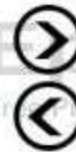
```
SET IMPLICIT_TRANSACTION OFF
```

```
SET IMPLICIT_TRANSACTION ON
```

```
COMMIT TRANSACTION
```

```
BEGIN TRY
    EXEC dbo.Proc1
    EXEC dbo.Proc2
    EXEC dbo.Proc3
    IF (XACT_STATE() = 1)
        COMMIT TRANSACTION;
END TRY
```

Answer Area



Correct Answer:

Transact-SQL segments	Answer Area
	SET IMPLICIT_TRANSACTIONS ON
	BEGIN TRAN
IF (@TRANCOUNT > 0) ROLLBACK TRANSACTION	BEGIN TRY EXEC dbo.Proc1 EXEC dbo.Proc2 EXEC dbo.Proc3 IF (XACT_STATE() = 1) COMMIT TRANSACTION; END TRY
EXEC dbo.Proc1 EXEC dbo.Proc2 EXEC dbo.Proc3	BEGIN CATCH IF (XACT_STATE() != 0) ROLLBACK TRANSACTION END CATCH
SET IMPLICIT_TRANSACTIONS OFF	
COMMIT TRANSACTION	

**Section: (none)**

**Explanation**

**Explanation/Reference:**

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, all individual 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.

[https://technet.microsoft.com/en-us/library/ms187807\(v=sql.105\).aspx](https://technet.microsoft.com/en-us/library/ms187807(v=sql.105).aspx)

[https://technet.microsoft.com/en-us/library/ms189797\(v=sql.110\).aspx](https://technet.microsoft.com/en-us/library/ms189797(v=sql.110).aspx)

## QUESTION 15

### HOTSPOT

You are profiling a frequently used database table named UserEvents. The READ\_COMMITTED\_SNAPSHOT database option is set to OFF.

In the trace results, you observe that lock escalation occurred for one stored procedure even though the number of locks in the database did not exceed memory or configuration thresholds. Events details are provided in the following table:

Attribute	Value
EventClass	LockEscalation
Mode	5 - X (X represents exclusive)
ObjectID	274100017 (objectID of table UserEvents)
Type	5 - OBJECT
TextData	<pre>CREATE PROCEDURE uspDeleteEvents (@EventType tinyint) AS BEGIN     SET NOCOUNT ON;     SET TRAN ISOLATION LEVEL READ COMMITTED;     LABEL_DELETE: DELETE FROM UserEvents WITH (ROWLOCK) WHERE EventType = @EventType; END;</pre>

You need to modify the uspDeleteEvents stored procedure to avoid lock escalation.

How should you modify the stored procedure? To answer, select the appropriate Transact-SQL segments in the answer area.

**Hot Area:**

**Answer Area**

```
CREATE PROCEDURE uspDeleteEvents (@EventType tinyint)
AS
BEGIN
    SET NOCOUNT ON;
    SET TRAN ISOLATION LEVEL READ COMMITTED;

    LABEL_DELETE_DELETE TOP(4000) FROM UserEvents WITH (ROWLOCK) WHERE EventType = @EventType
    LABEL_DELETE_DELETE FROM UserEvents WITH (PAGELOCK) WHERE EventType = @EventType
    LABEL_DELETE_DELETE TOP(400000) FROM UserEvents WITH (ROWLOCK) WHERE EventType = @EventType

    IF @@TRANCOUNT > 0 COMMIT TRAN
    WHILE @@ROWCOUNT > 0 GOTO LABEL_DELETE
    IF @@TRANCOUNT > 4000 COMMIT TRAN
    SET ROWCOUNT 4000
END;
```

Correct Answer:

**Answer Area**

```
CREATE PROCEDURE uspDeleteEvents (@EventType tinyint)
AS
BEGIN
    SET NOCOUNT ON;
    SET TRAN ISOLATION LEVEL READ COMMITTED;

    LABEL_DELETE_DELETE TOP(4000) FROM UserEvents WITH (ROWLOCK) WHERE EventType = @EventType
    LABEL_DELETE_DELETE FROM UserEvents WITH (PAGELOCK) WHERE EventType = @EventType
    LABEL_DELETE_DELETE TOP(400000) FROM UserEvents WITH (ROWLOCK) WHERE EventType = @EventType

    IF @@TRANCOUNT > 0 COMMIT TRAN
    WHILE @@ROWCOUNT > 0 GOTO LABEL_DELETE
    IF @@TRANCOUNT > 4000 COMMIT TRAN
    SET ROWCOUNT 4000
END;
```

Section: (none)

Explanation

**Explanation/Reference:**

Delete up to 4000 rows at a time. Keep doing it until all rows have been deleted  
Note that @@ROWCOUNT returns the number of rows affected by the last statement  
<https://msdn.microsoft.com/en-us/library/ms187316.aspx>

QUESTION 16



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 that contains the following tables: BlogCategory, BlogEntry, ProductReview, Product, and Salesperson. The tables were created using the following Transact SQL statements:

```
CREATE TABLE BlogCategory
(
    CategoryID int NOT NULL PRIMARY KEY,
    CategoryName nvarchar (20)
);

CREATE TABLE BlogEntry
(
    Entry int NOT PRIMARY KEY,
    Entrytitle nvarchar (50),
    Category int NOT NULL FOREIGN KEY REFERENCES BlogCategory
(CategoryID)
);

CREATE TABLE dbo.ProductReview
(
    ProductReviewID IDENTITY(1,1) PRIMARY KEY,
    Product int NOT NULL,
    Review varchar (1000) NOT NULL
);

CREATE TABLE dbo.Product
(
    ProductID int Identity(1,1) PRIMARY KEY,
    Name varchar(1000) NOT NULL
);

CREATE TABLE dbo.SalesPerson
(
    SalesPersonID int IDENTITY(1,1) PRIMARY KEY,
    Name varchar (1000) NOT NULL,
    SalesID Money
)
```



You must modify the ProductReview Table to meet the following requirements:

- 1The table must reference the ProductID column in the Product table
- 2Existing records in the ProductReview table must not be validated with the Product table.
- 3,Deleting records in the Product table must not be allowed if records are referenced by the ProductReview table.
- 4,Changes to records in the Product table must propagate to the ProductReview table.

You also have the following database tables: Order, ProductTypes, and SalesHistory, The transact-SQL statements for these tables are not available.

You must modify the Orders table to meet the following requirements:

- 1.Create new rows in the table without granting INSERT permissions to the table.
- 2,Notify the sales person who places an order whether or not the order was completed

You must add the following constraints to the SalesHistory table:

- a constraint on the SaleID column that allows the field to be used as a record identifier
- a constant that uses the ProductID column to reference the Product column of the ProductTypes table
- a constraint on the CategoryID column that allows one row with a null value in the column

-a constraint that limits the SalePrice column to values greater than four

Finance department users must be able to retrieve data from the SalesHistory table for sales persons where the value of the SalesYTD column is above a certain threshold.

You plan to create a memory-optimized table named SalesOrder. The table must meet the following requirements:

-The table must hold 10 million unique sales orders.

-The table must use checkpoints to minimize I/O operations and must not use transaction logging.

-Data loss is acceptable.

Performance for queries against the SalesOrder table that use Where clauses with exact equality operations must be optimized. You need to enable referential integrity for the ProductReview table.

How should you complete the relevant Transact-SQL statement? To answer, select the appropriate Transact-SQL segments in the answer area.

Answer Area

Alter Table dbo.ProductReview

WITH CHECK

ADD CONSTRAINT FK\_ProductReview\_Product FOREIGN KEY (ProductID)

REFERENCES Product (productID)

ON DELETE NO ACTION ON UPDATE CASCADE

Select two alternatives.

- A. For the first selection select: WITH CHECK
- B. For the first selection select: WITH NOCHECK
- C. For the second selection select: ON DELETE NO ACTION ON UPDATE CASCADE
- D. For the second selection select: ON DELETE CASCADE ON UPDATE CASCADE
- E. For the second selection select: ON DELETE NO ACTION ON UPDATE NO ACTION
- F. For the second selection select: ON DELETE CASCADE ON UPDATE NO ACTION

**Correct Answer:** BC

**Section:** (none)

**Explanation**

**Explanation/Reference:**

B: We should use WITH NOCHECK as existing records in the ProductReview table must not be validated with the Product table.

C: Deletes should not be allowed, so we use ON DELETE NO ACTION.

Updates should be allowed, so we use ON DELETE NO CASCADE.

NO ACTION: the Database Engine raises an error, and the update action on the row in the parent table is rolled back.

CASCADE: corresponding rows are updated in the referencing table when that row is updated in the parent table.

Note: ON DELETE { NO ACTION | CASCADE | SET NULL | SET DEFAULT }

Specifies what action happens to rows in the table that is altered, if those rows have a referential relationship and the referenced row is deleted from the parent table. The default is NO ACTION.

ON UPDATE { NO ACTION | CASCADE | SET NULL | SET DEFAULT }

Specifies what action happens to rows in the table altered when those rows have a referential relationship and the referenced row is updated in the parent table. The default is NO ACTION.

Note: You must modify the ProductReview Table to meet the following requirements:

1. The table must reference the ProductID column in the Product table
2. Existing records in the ProductReview table must not be validated with the Product table.
3. Deleting records in the Product table must not be allowed if records are referenced by the ProductReview table.
4. Changes to records in the Product table must propagate to the ProductReview table

<https://msdn.microsoft.com/en-us/library/ms190273.aspx>

<https://msdn.microsoft.com/en-us/library/ms188066.aspx>

### QUESTION 17

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 that contains the following tables: BlogCategory, BlogEntry, ProductReview, Product, and Salesperson. The tables were created using the following Transact SQL statements:

```
CREATE TABLE BlogCategory
(
    CategoryID int NOT NULL PRIMARY KEY,
    CategoryName nvarchar (20)
);

CREATE TABLE BlogEntry
(
    Entry int NOT PRIMARY KEY,
    Entrytitle nvarchar (50),
    Category int NOT NULL FOREIGN KEY REFERENCES BlogCategory
(CategoryID)
);

CREATE TABLE dbo.ProductReview
(
    ProductReviewID IDENTITY(1,1) PRIMARY KEY,
    Product int NOT NULL,
    Review varchar (1000) NOT NULL
);

CREATE TABLE dbo.Product
(
    ProductID int Identity(1,1) PRIMARY KEY,
    Name varchar(1000) NOT NULL
);

CREATE TABLE dbo.SalesPerson
(
    SalesPersonID int IDENTITY(1,1) PRIMARY KEY,
    Name varchar (1000) NOT NULL,
    SalesID Money
);
```



You must modify the ProductReview Table to meet the following requirements:

1. The table must reference the ProductID column in the Product table
2. Existing records in the ProductReview table must not be validated with the Product table.
3. Deleting records in the Product table must not be allowed if records are referenced by the ProductReview table.
4. Changes to records in the Product table must propagate to the ProductReview table.

You also have the following database tables: Order, ProductTypes, and SalesHistory. The Transact-SQL statements for these tables are not available.

You must modify the Orders table to meet the following requirements:

1. Create new rows in the table without granting INSERT permissions to the table.
2. Notify the sales person who places an order whether or not the order was completed.

You must add the following constraints to the SalesHistory table:

- a constraint on the SaleID column that allows the field to be used as a record identifier
- a constraint that uses the ProductID column to reference the Product column of the ProductTypes table
- a constraint on the CategoryID column that allows one row with a null value in the column
- a constraint that limits the SalePrice column to values greater than four

Finance department users must be able to retrieve data from the SalesHistory table for sales persons where the value of the SalesYTD column is above a certain threshold.

You plan to create a memory-optimized table named SalesOrder. The table must meet the following requirements:

- The table must hold 10 million unique sales orders.
- The table must use checkpoints to minimize I/O operations and must not use transaction logging.
- Data loss is acceptable.

Performance for queries against the SalesOrder table that use Where clauses with exact equality operations must be optimized.

You need to modify the design of the Orders table.

What should you create?

- A. a stored procedure with the RETURN statement
- B. a FOR UPDATE trigger
- C. an AFTER UPDATE trigger
- D. a user defined function



**Correct Answer: D**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Requirements: You must modify the Orders table to meet the following requirements:

1. Create new rows in the table without granting INSERT permissions to the table.
2. Notify the sales person who places an order whether or not the order was completed.

<https://msdn.microsoft.com/en-us/library/ms186755.aspx>

#### **QUESTION 18**

You have a database that is experiencing deadlock issues when users run queries.

You need to ensure that all deadlocks are recorded in XML format.

What should you do?

- A. Create a Microsoft SQL Server Integration Services package that uses sys.dm\_tranJocks.
- B. Enable trace flag 1224 by using the Database Consistency Checker (BDCC).

- C. Enable trace flag 1222 in the startup options for Microsoft SQL Server.
- D. Use the Microsoft SQL Server Profiler Lock:Deadlock event class.

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

When deadlocks occur, trace flag 1204 and trace flag 1222 return information that is captured in the SQL Server error log. Trace flag 1204 reports deadlock information formatted by each node involved in the deadlock. Trace flag 1222 formats deadlock information, first by processes and then by resources. The output format for Trace Flag 1222 only returns information in an XML-like format.  
[https://technet.microsoft.com/en-us/library/msl78104\(v=sql.105\).aspx](https://technet.microsoft.com/en-us/library/msl78104(v=sql.105).aspx)

**QUESTION 19**

You are developing an application that connects to a database. The application runs the following jobs:

Job	Transact-SQL statement	Description
JobA	Exec uspDeletePrevRecords	The stored procedure deletes all records from a table named tblBalanceTransactions that were created before the current month by using a single DELETE statement. Approximately 10 million records are deleted each time you run this stored procedure.
JobB	Exec uspUpdateCurRecords	This stored procedure updates records in the tblBalanceTransaction table that were created in the current month. Only a few hundred records are updated each time you run this stored procedure.

The READ\_COMMITTED\_SNAPSHOT database option is set to OFF, and auto-commit is set to ON. Within the stored procedures, no explicit transactions are defined.

If JobB starts before JobA, it can finish in seconds. If JobA starts first, JobB takes a long time to complete.

You need to use Microsoft SQL Server Profiler to determine whether the blocking that you observe in JobB is caused by locks acquired by JobA.

Which trace event class in the Locks event category should you use?

- A. LockAcquired
- B. LockCancel
- C. LockDeadlock
- D. LockEscalation

**Correct Answer:** A

**Section:** (none)

## Explanation

### Explanation/Reference:

The Lock:Acquiredevent class indicates that acquisition of a lock on a resource, such as a data page, has been achieved.

The Lock:Acquired and Lock:Released event classes can be used to monitor when objects are being locked, the type of locks taken, and for how long the locks were retained. Locks retained for long periods of time may cause contention issues and should be investigated.

### QUESTION 20

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:

```
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)
);
```



The value of the CustomerStatus column is equal to one for active customers. The value of the Account1 Status and Account2Status columns are equal to one for active accounts. The following table displays selected columns and rows from the Customer table.

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:

1. Allow users access to the CustomerName and CustomerNumber columns for active customers.
2. Allow changes to the columns that the view references. Modified data must be visible through the view.
3. Prevent the view from being published as part of Microsoft SQL Server replication.

Sales.Female.Customers must meet the following requirements:



1. Allow users access to the CustomerName, Address, City, State and PostalCode columns.
2. Prevent changes to the columns that the view references.
3. 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 spUpdateCustomerSummary stored procedure to make changes to customer account summaries. Other stored procedures call the spDeleteCustAcctRelationship to delete records from the CustomerToAccountBridge table.

You must update the design of the Customer table to meet the following requirements.

1. You must be able to store up to 50 accounts for each customer.
2. Users must be able to retrieve customer information by supplying an account number.
3. Users must be able to retrieve an account number by supplying customer information.

You need to implement the design changes while minimizing data redundancy.

What should you do?

- A. Split the table into three separate tables. Include the AccountNumber and CustomerID columns in the first table. Include the CustomerName and Gender columns in the second table. Include the AccountStatus column in the third table.
- B. Split the table into two separate tables. Include AccountNumber, CustomerID, CustomerName and Gender columns in the first table. Include the AccountNumber and AccountStatus columns in the second table.
- C. Split the table into two separate tables, Include the CustomerID and AccountNumber columns in the first table. Include the AccountNumber, AccountStatus, CustomerName and Gender columns in the second table.
- D. Split the table into two separate tables, Include the CustomerID, CustomerName and Gender columns in the first table. Include AccountNumber, AccountStatus and CustomerID columns in the second table.

**Correct Answer: D**

**Section: (none)**

**Explanation**

## Explanation/Reference:

### QUESTION 21

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 Account1 Status and Account2Status columns are equal to one for active accounts. The following table displays selected columns and rows from the Customer table.

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:

- 1.Allow users access to the CustomerName and CustomerNumber columns for active customers.
- 2.Allow changes to the columns that the view references. Modified data must be visible through the view.
- 3.Prevent the view from being published as part of Microsoft SQL Server replication.

Sales.Female.Customers must meet the following requirements:

- 1.Allow users access to the CustomerName, Address, City, State and PostalCode columns.
- 2.Prevent changes to the columns that the view references.
- 3.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 Transact-SQL statement:

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. The value of the TotalAccountCount column decreased.
- B. The value of the TotalDepositAccountCount column is decreased.
- C. The statement that modifies TotalDepositAccountCount is excluded from the transaction.
- D. The value of the TotalAccountCount column is not changed.

**Correct Answer: D**

**Section: (none)**

**Explanation**



## Explanation/Reference:

### QUESTION 22

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 on 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 work on an OLTP database that has no memory-optimized file group defined.

You have a table names tblTransaction that is persisted on disk and contains the information described in the following table:

Item	Name	Data Type	Nullable	Notes
Column	TransactionDate	Date	No	For each transaction date, there are only about 100,000 records. The table contains over one billion records in total.
Column	SequenceNo	bigint	No	Uniquely identifies a transaction record within a date
Column	AccountId	int	No	
Column	ValueType	char(3)	No	
Column	Amount	decimal(20,2)	Yes	
	IX_ValueType			Nonclustered columnstore index on the ValueType column.

Users report that the following query takes a long time to complete

```
SELECT TransactionDate, COUNT(*) AS TotalCount FROM tblTransaction
WHERE TransactionDate - DATEADD(D, -1, CONVERT (DATE, CONVERT (VARCHAR(8),
GETDATE(),112)112))
GROUP BY TransactionDate;
```

You need to create an index that:

- improves the query performance
- does not impact the existing index
- minimizes storage size of the table (inclusive of index pages)

What should you do?

- Create a clustered index on the table
- Create a nonclustered index on the table.
- Create a nonclustered filtered index on the table.
- Create a clustered columnstore index on the table.
- Create a nonclustered columnstore index on the table.

F. Create a hashindex on the table.

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

A filtered index is an optimized nonclustered index, especially suited to cover queries that select from a well-defined subset of data. It uses a filter predicate to index a portion of rows in the table. A well-designed filtered index can improve query performance, reduce index maintenance costs, and reduce index storage costs compared with full-table indexes.

**QUESTION 23**

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 is independent of the other questions in this series. Information and details provided in a question apply only to that question.

You have a database named DB1. There is no memory-optimized filegroup in the database.

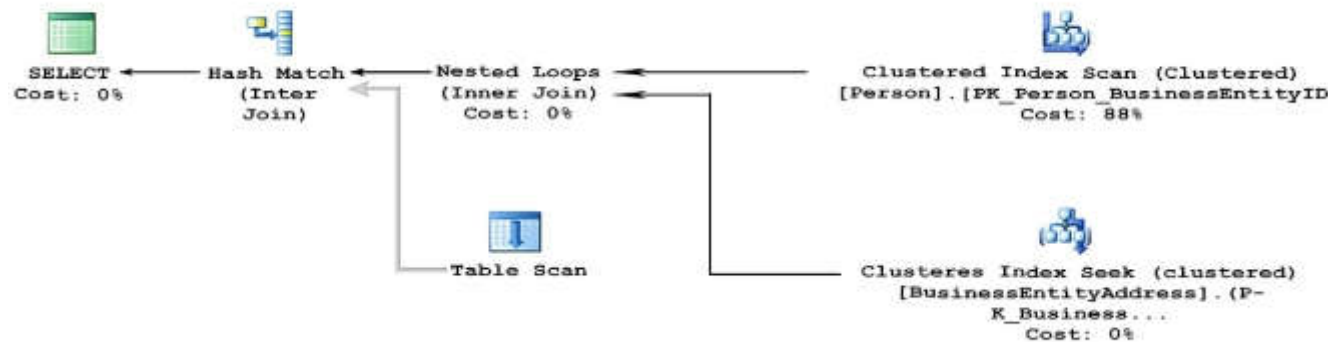
You run the following query:

The following image displays the execution plan the query optimizer generates for this query

```
DECLARE @LastName varchar(20) = "contoso";
SELECT P.FirstName, Addr.*
FROM Person.Person P
JOIN Person.BusinessEntityAddress ON P.BusinessEntityID =
BusinessEntityAddress.BusinessEntityID
JOIN Person.[Address] ADDR ON BusinessEntityAddress.AddressID =
Addr.AddressID
WHERE P.LastName = @LastName;
```



Users frequently run the same query with different values for the local variable @lastName. The table named Person is persisted on disk. You need to create an index on the Person.Person table that meets the following requirements:



1.All users must be able to benefit from the index.  
2.FirstName must be added to the index as an included column.  
What should you do?

- A. Create a clustered index on the table.
- B. Create a nonclustered index on the table.
- C. Create a nonclustered filtered index on the table.
- D. Create a nonclustered columnstore index on the table.
- E. Create a hash index on the table
- F. Create a clustered columnstore index on the table

**Correct Answer:** B

**Section:** (none)

**Explanation**

**Explanation/Reference:**

By including nonkey columns, you can create nonclustered indexes that cover more queries. This is because the nonkeycolumns have the following benefits:  
They can be data types not allowed as index key columns.  
They are not considered by the Database Engine when calculating the number of index key columns or index key size

**QUESTION 24**

Note: The 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 is independent of the other question in the series. Information and details provided in a question apply only to that question.

You have a reporting database that includes a non-partitioned fact table named Fact\_Sales. The table is persisted on disk. Users report that their queries take a

long time to complete. The system administrator reports that the table takes too much space in the database. You observe that there are no indexes defined on the table, and many columns have repeating values. You need to create the most efficient index on the table, minimize disk storage and improve reporting query performance.

What should you do?

- A. Create a clustered index on the table.
- B. Create a nonclustered index on the table.
- C. Create a nonclustered filtered index on the table.
- D. Create a clustered columnstore index on the table.
- E. Create a nonclustered columnstore index on the table.
- F. Create a hash index on the table.

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

The columnstore index is 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, and up to 10x data compression over the uncompressed data size. A clustered columnstore index is the physical storage for the entire table.

**QUESTION 25**

Note: The 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 is independent of the other question in the series. Information and details provided in a question apply only to that question.

You have a database named DB1. The database does not use a memory-optimized filegroup. The database contains a table named Table1. The table must support the following workloads:

Workload	Type	Description
Reporting	Existing	The reporting workload must scan most of the records in the table to aggregate on a number of columns. A clustered columnstore index is already created on the table to support this workload.
OLTP	New	The OLTP workload must support 3,000 transactions per second. Rows are identified by using two columns. The filter is variant on one of the two columns while constant on the other. Only a small number of records with a few columns are returned by the query.

You need to add the most efficient index to support the new OLTP workload, while not deteriorating the existing Reporting query performance.

What should you do?

- A. Create a clustered index on the table
- B. Create a nonclustered index on the table,

- C. Create a nonclustered filtered index on the table.
- D. Create a clustered columnstore index on the table,
- E. Create a nonclustered columnstore index on the table
- F. Create a hash index on the table,

**Correct Answer: C**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

A filtered index is an optimized nonclustered index, especially suited to cover queries that select from a well-defined subset of data. It uses a filter predicate to index a portion of rows in the table. A well-designed filtered index can improve query performance, reduce index maintenance costs, and reduce index storage costs compared with full-table indexes.

[https://technet.microsoft.com/en-us/library/cc280372\(v=sql.105\).aspx](https://technet.microsoft.com/en-us/library/cc280372(v=sql.105).aspx)

**QUESTION 26**

Note: The 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 is independent of the other question in the series. Information and details provided in a question apply only to that question.

You have a database named DB1. The database does not have a memory optimized filegroup.

You create a table by running the following Transact-SQL statement:

```
CREATE TABLE tblTransaction(  
    [TransactionID] [int] NOT NULL PRIMARY KEY,  
    [TransactionDate] [date] NOT NULL,  
    [AccountId] [int] NOT NULL,  
    [ValueType] [char](3) NOT NULL,  
    [Amount] [decimal](20,2) NULL  
);
```

The table is currently used for OLTP workloads. The analytics user group needs to perform real-time operational analytics that scan most of the records in the table to aggregate on a number of columns.

You need to add the most efficient index to support the analytics workload without changing the OLTP application.

What should you do?

- A. Create a clustered index on the table.
- B. Create a nonclustered index on the table.
- C. Create a nonclustered filtered index on the table
- D. Create a clustered columnstore index on the table.
- E. Create a nonclustered columnstore index on the table.

F. Create a hash index on the table

**Correct Answer:** E

**Section:** (none)

**Explanation**

**Explanation/Reference:**

A nonclustered columnstore index enables real-time operational analytics in which the OLTP workload uses the underlying clustered index, while analytics run concurrently on the columnstore index.

Columnstore indexes can achieve up to 100xbetter performance on analytics and data warehousing workloads and up to 10x better data compression than traditional rowstore indexes. These recommendations will help your queries achieve the very fast query performance that columnstore indexes are designed to provide.

<https://msdn.microsoft.com/en-us/library/gg492088.aspx>

#### QUESTION 27

You use Microsoft SQL Server Profiler to evaluate a query named Query1. The Profiler report indicates the following issues:

-At each level of the query plan, a low total number of rows are processed.

-The query uses many operations. This results in a high overall cost for the query.

You need to identify the information that will be useful for the optimizer. What should you do?

- A. Start a SQL Server Profiler trace for the event class Auto Stats in the Performance event category.
- B. Create one Extended Events session with the sqlserver.missing\_column\_statistics event added.
- C. Start a SQL Server Profiler trace for the event class Soft Warnings in the Errors and Warnings event category.
- D. Create one Extended Events session with the sqlserver.missing\_join\_predicate event added.

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

The Missing Join Predicate event class indicates that a query is being executed that has no join predicate. This could result in a long-running query.

#### QUESTION 28

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 dynamicmanagement view query

D. an Activity Monitor session in Microsoft SQL Management Studio.

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

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 29

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 is independent of the other questions in the series. Information and details provided in a question apply only to that question.

You are developing an application to track customer sales.

You need to create a database object that meets the following requirements:

-Return a value of 0 if data inserted successfully into the Customers table.

-Return a value of 1 if data is not inserted successfully into the Customers table.

-Support logic that is written by using managed code.

What should you create?

- 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:** B

**Section:** (none)

**Explanation**

**Explanation/Reference:**

DML triggers is a special type of stored procedure that automatically takes effect when a data manipulation language (DML) event takes place that affects the table or view defined in the trigger. DML events include INSERT, UPDATE, or DELETE statements. DML triggers can be used to enforce business rules and data integrity, query other tables, and include complex Transact-SQL statements.



A CLR trigger is a type of DDL trigger. A CLR Trigger can be either an AFTER or INSTEAD OF trigger. A CLR trigger can also be a DDL trigger. Instead of executing a Transact-SQL stored procedure, a CLR trigger executes one or more methods written in managed code that are members of an assembly created in the .NET Framework and uploaded in SQL Server.  
<https://msdn.microsoft.com/en-us/library/ms178110.aspx>

### QUESTION 30

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 is independent of the other questions in the series. Information and details provided in a question apply only to that question.

You are developing an application to track customer sales.

You need to create a database object that meets the following requirements:

- Return a value of 0 if data is inserted successfully into the Customers table.
- Return a value of 1 if data is not inserted successfully into the Customers table.
- Support TRY . CATCH error handling
- Be written by using Transact-SQL statements.

What should you create?

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

**Correct Answer:** D

**Section:** (none)

**Explanation**

#### **Explanation/Reference:**

DML triggers is a special type of stored procedure that automatically takes effect when a data manipulation language (DML) event takes place that affects the table or view defined in the trigger. DML events include INSERT, UPDATE, or DELETE statements. DML triggers can be used to enforce business rules and data integrity, query other tables, and include complex Transact-SQL statements.  
<https://msdn.microsoft.com/en-us/library/ms178110.aspx>

### QUESTION 31

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 is independent of the other questions in the series. Information and details provided in a question apply only to that question.

You are developing an application to track customer sales.

You need to create a database object that meets the following requirements:

- Launch when table data is modified.
- Evaluate the state a table before and after a data modification and take action based on the difference.
- Prevent malicious or incorrect table data operations.