# Microsoft

## 70-762 Exam

### Microsoft Developing SQL Databases Exam

Thank you for Downloading 70-762 exam PDF Demo

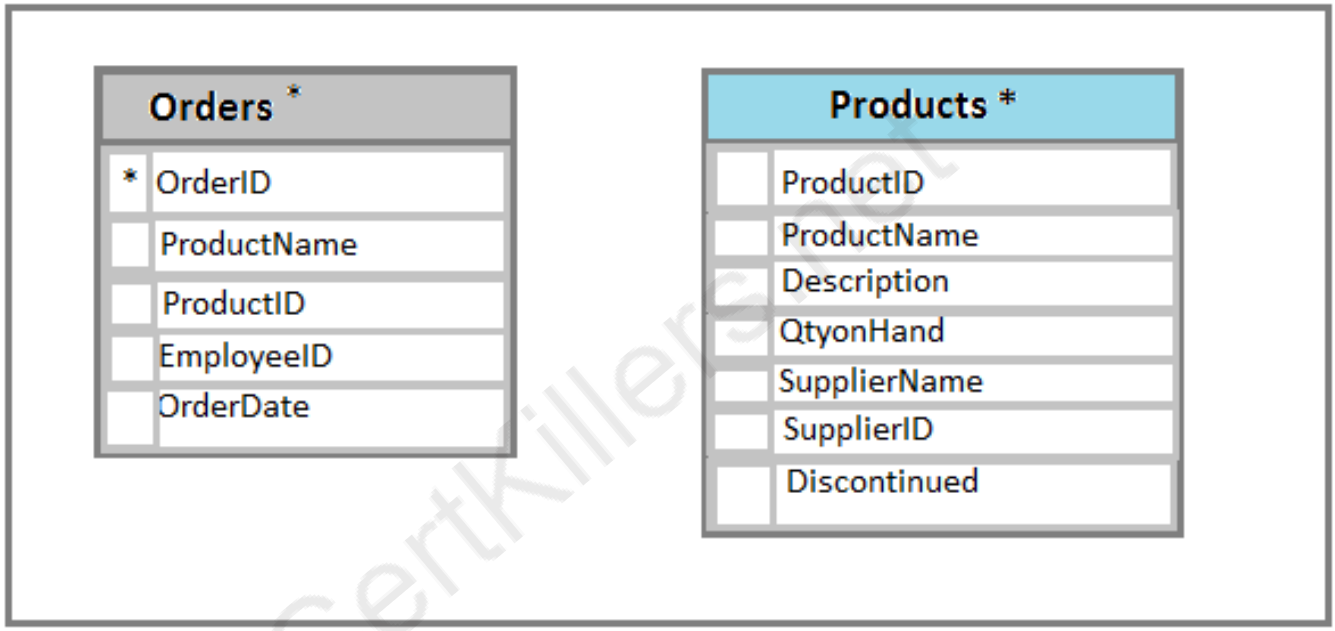
You can Buy Latest 70-762 Full Version Download

https://www.certkillers.net/Exam/70-762

# Version: 19.0

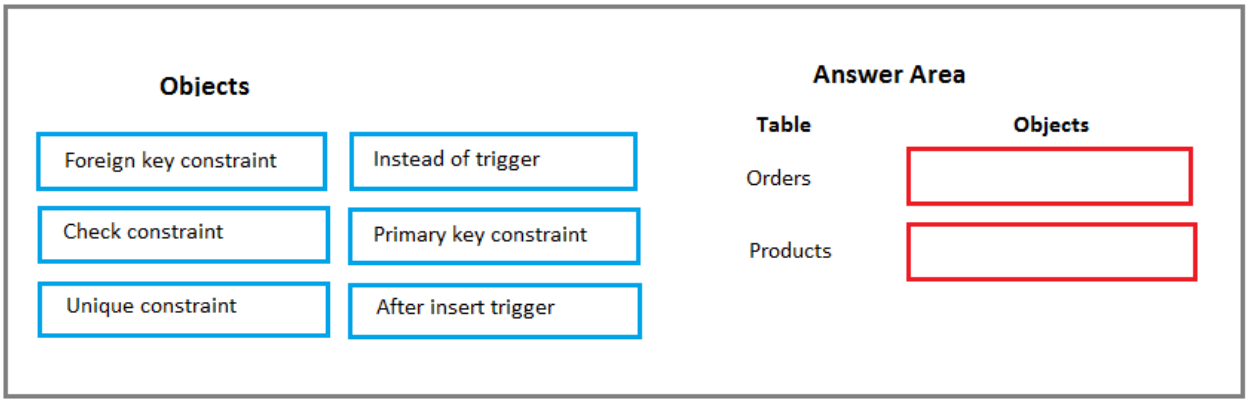
## Question: 1

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.

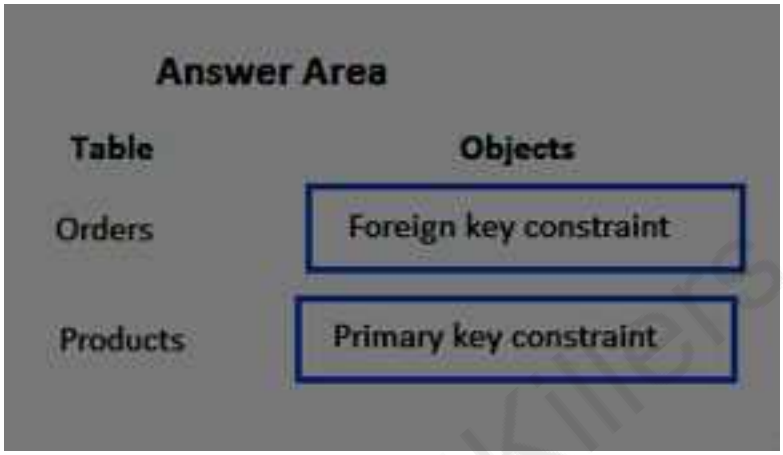





The customer table includes a column that stores the data for the last order that the customer placed.
You plan to create a table named Leads. The Leads table is expected to contain approximately 20,000 records. Storage requirements for the Leads table must be minimized.
Changes to the price of any product 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.



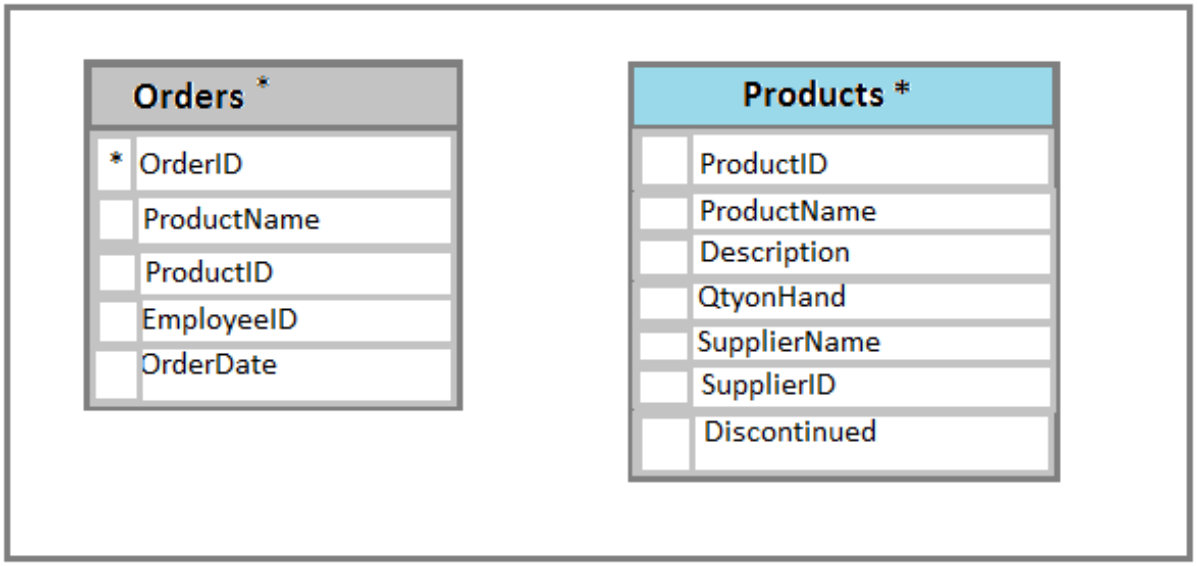

**Answer:**




Explanation:
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.

**Answer Area**

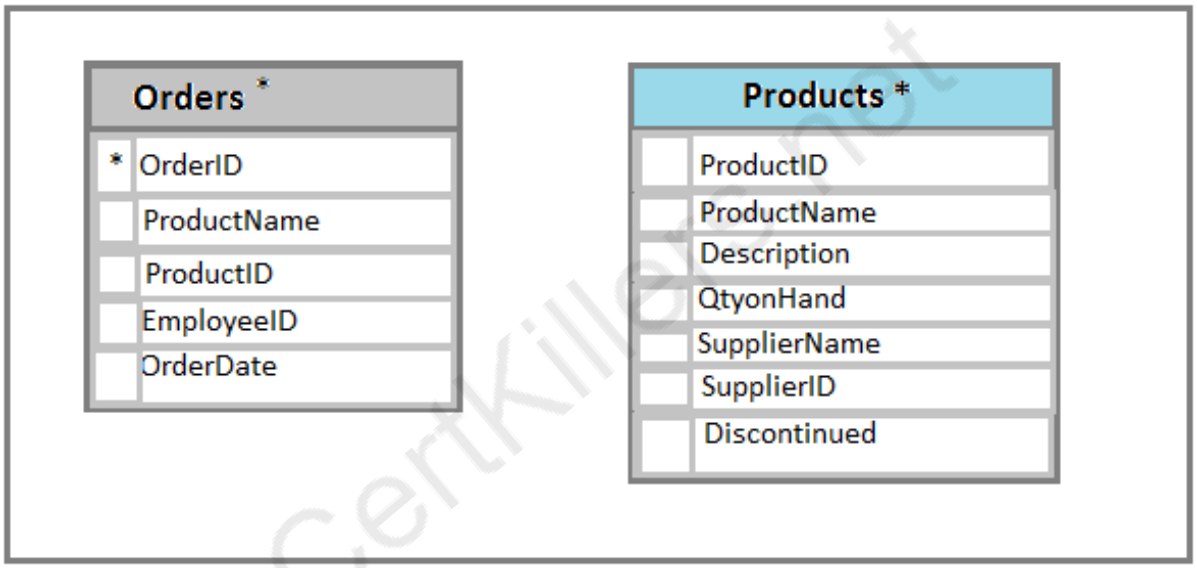
| Requirement | Transact-SQL segment |
|---|---|
| Handle errors | Try/Parse Select @@error<br>Begin Tran/Rollback Tran<br>Try/Catch* |
| Display error message | ERROR_MESSAGE()<br>PRINT<br>RAISERROR<br>RETURN |

**Answer:**

Try/Catch*
RAISERROR
Explanation:
https://docs.microsoft.com/en-us/sql/t-sql/language-elements/raiserror-transact-sql?view=sql-server-2017
References: 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.

**Answer Area**

| Trigger type | Provide custom | Update Customer table |
|---|---|---|
| AFTER INSERT trigger | ● | ● |
| INSTEAD OF INSERT trigger | ● | ● |
| AFTER UPDATE trigger | ● | ● |
| INSTEAD OF UPDATE trigger | ● | ● |

**Answer:**

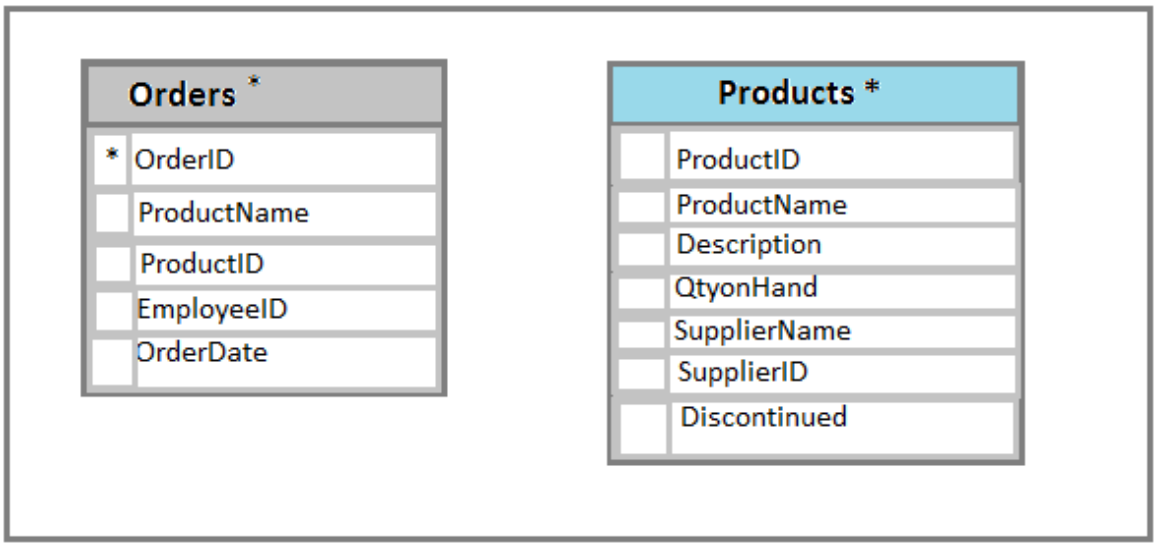
**Answer Area**

| Trigger type | Provide custom | Update Customer table |
|---|---|---|
| AFTER INSERT trigger | ● | ● |
| INSTEAD OF INSERT trigger | ● | ● |
| AFTER UPDATE trigger | ● | ● |
| INSTEAD OF UPDATE trigger | ● | ● |

Explanation:
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.
References: 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 vale 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.

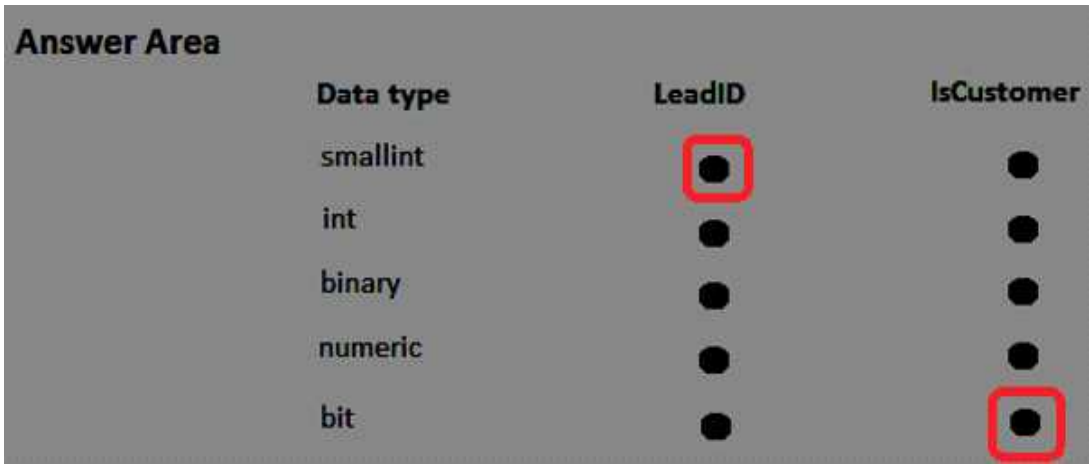
In the table below, identify the data type that must be used for each table column.

NOTE: Make only one selection in each column.

**Answer Area**

| Data type | LeadID | IsCustomer |
|---|---|---|
| smallint | ● | ● |
| int | ● | ● |
| binary | ● | ● |
| numeric | ● | ● |
| bit | ● | ● |

**Answer:**



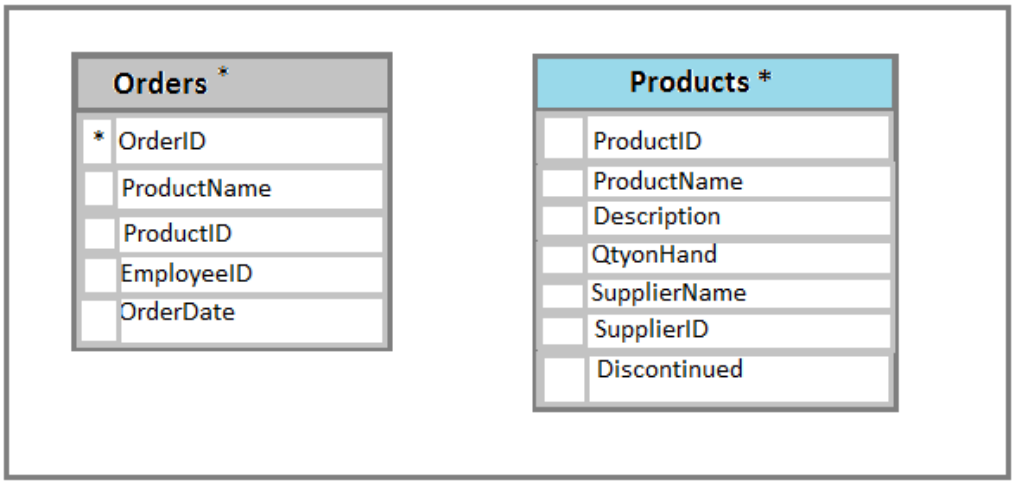

Explanation:
Bit is a Transact-SQL integer data type that can take a value of 1, 0, or NULL.
Smallint is aT ransact-SQL integer data type that can take a value in the range from -32,768 to 32,767.
int, bigint, smallint, and tinyint (Transact-SQL)
Exact-number data types that use integer data.

| Data type | Range | Storage |
|---|---|---|
| **bigint** | -2^63 (-9,223,372,036,854,775,808) to 2^63-1 (9,223,372,036,854,775,807) | 8 Bytes |
| **int** | -2^31 (-2,147,483,648) to 2^31-1 (2,147,483,647) | 4 Bytes |
| **smallint** | -2^15 (-32,768) to 2^15-1 (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.

**Answer Area**

| Constraint | Orders table | Products table |
|---|---|---|
| Check constraint on **OrderID** | ● | ● |
| Foreign key constraint on **ProductID** | ● | ● |
| Check constraint on **ProductID** | ● | ● |
| Foreign key constraint on **OrderID** | ● | ● |

**Answer:**

**Answer Area**

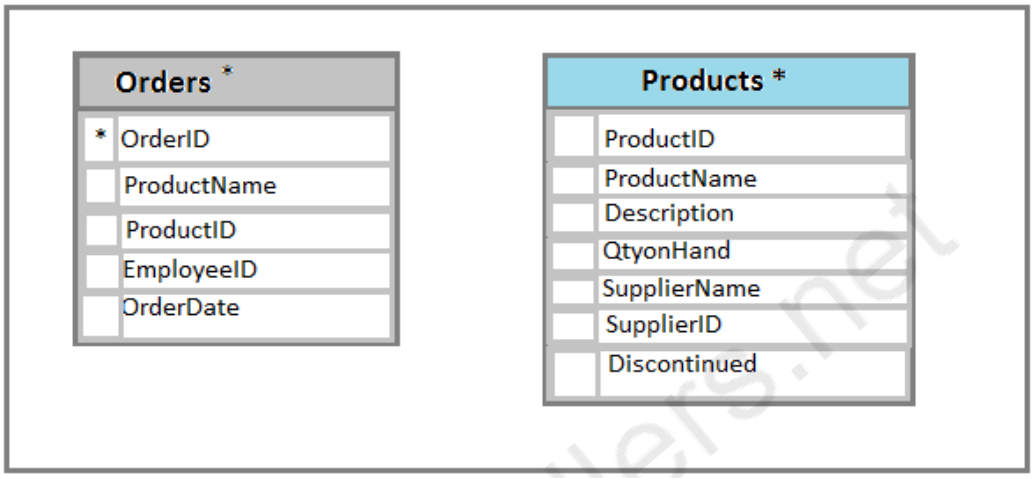
| Constraint | Orders table | Products table |
|---|---|---|
| Check constraint on **OrderID** | ● | ● |
| Foreign key constraint on **ProductID** | ● | ● |
| Check constraint on **ProductID** | ● | ● |
| Foreign key constraint on **OrderID** | ● | ● |

Explanation:
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.

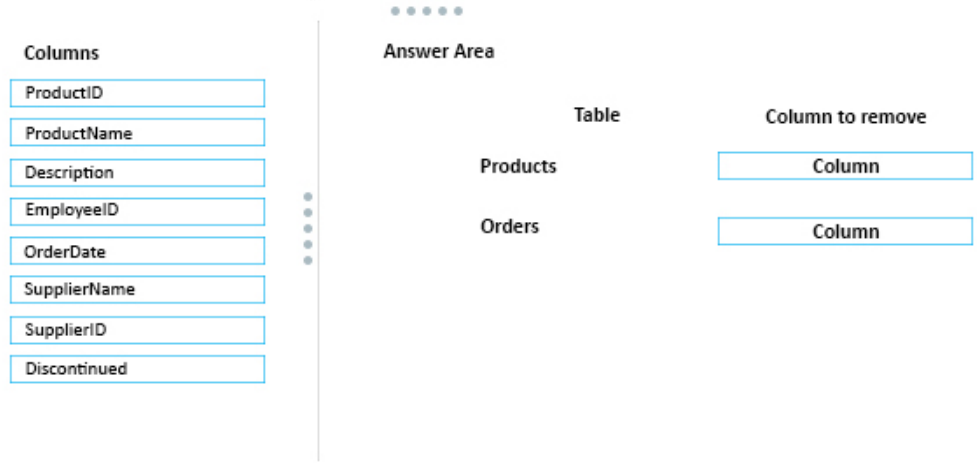
With a check constraint on the ProductID we can ensure that the Products table contains only unique rows.
References: http://www.w3schools.com/sql/sql_foreignkey.asp

## 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.
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.




**Answer:**

Answer Area

| Table | Column to remove |
|---|---|
| Products | SupplierName |
| Orders | ProductName |

Explanation:
In the Products table the SupplierName is dependent on the SupplierID, not on the ProductID.
In the Orders table the ProductName is dependent 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 non-primary fields are dependent on the primary key.
Second normal form states that it should meet all the rules for First 1Normnal Form and there must be no partial dependencies 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.
References: https://www.tutorialspoint.com/sql/third-normal-form.htm

## Question: 7

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:
* The table must reference the ProductID column in the Product table
* Existing records in the ProductReview table must not be validated with the Product table.
* Deleting records in the Product table must not be allowed if records are referenced by the ProductReview table.
* 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:
* Create new rows in the table without granting INSERT permissions to the table.
* 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.
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

**Answer: B,C**

Explanation:
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.
References: https://msdn.microsoft.com/en-us/library/ms190273.aspx
https://msdn.microsoft.com/en-us/library/ms188066.aspx

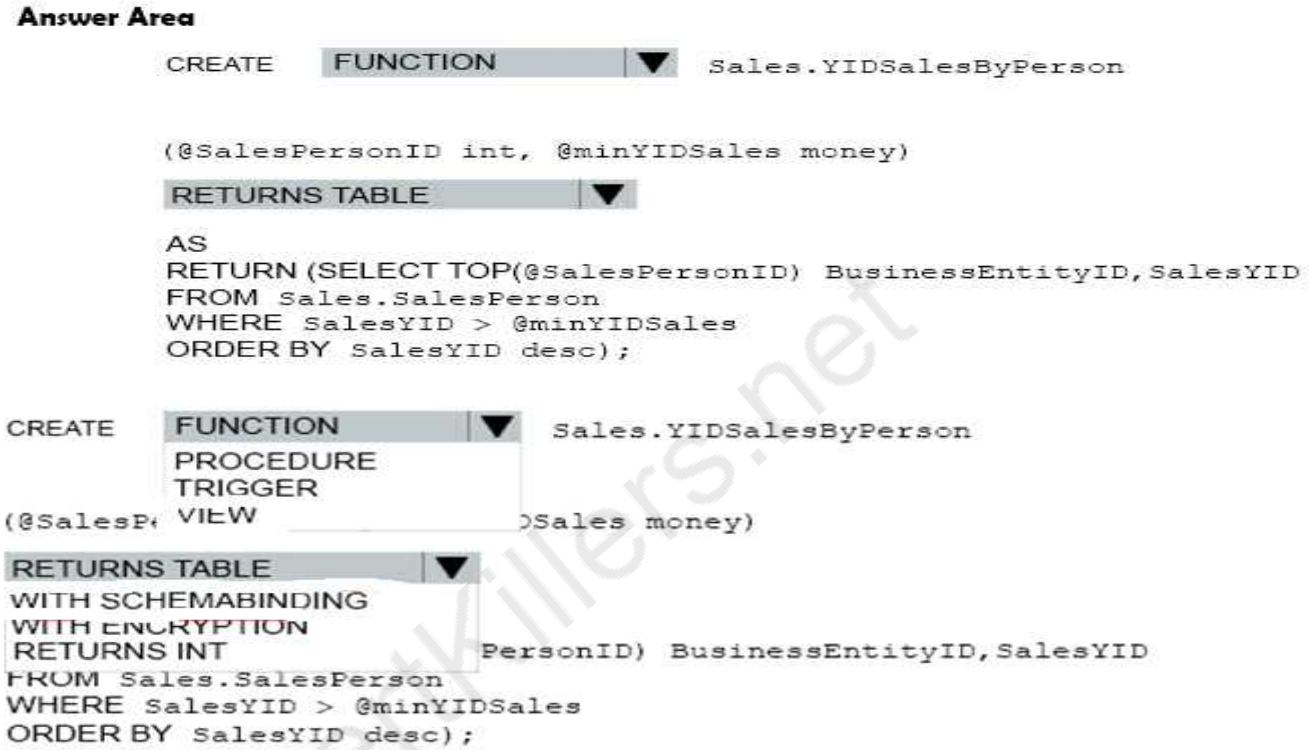
## Question: 8

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:

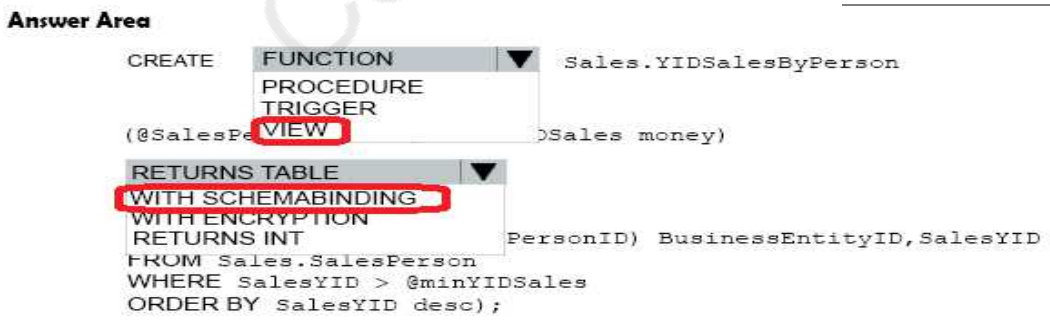
```
CREATE TABLE BlogCategory
(
      CategoryID int NOT NULL PRIMARY KEY,
      CategoryName nvarchar (20)
);

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:
* The table must reference the ProductID column in the Product table
* Existing records in the ProductReview table must not be validated with the Product table.
* Deleting records in the Product table must not be allowed if records are referenced by the ProductReview table.
* 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:
* Create new rows in the table without granting INSERT permissions to the table.
* 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.
How should you complete the Transact-SQL statements? To answer, select the appropriate Transact-SQL segments in the answer area.




**Answer:**




Explanation:
From question: Finance department users must be able to retrieve data from the SalesHistory table fo 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.

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

Question: 9
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:
* The table must reference the ProductID column in the Product table
* Existing records in the ProductReview table must not be validated with the Product table.
* Deleting records in the Product table must not be allowed if records are referenced by the ProductReview table.
* 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:
* Create new rows in the table without granting INSERT permissions to the table.
* 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 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

**Answer: D**

Explanation:

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.

References: https://msdn.microsoft.com/en-us/library/ms186755.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
)
```

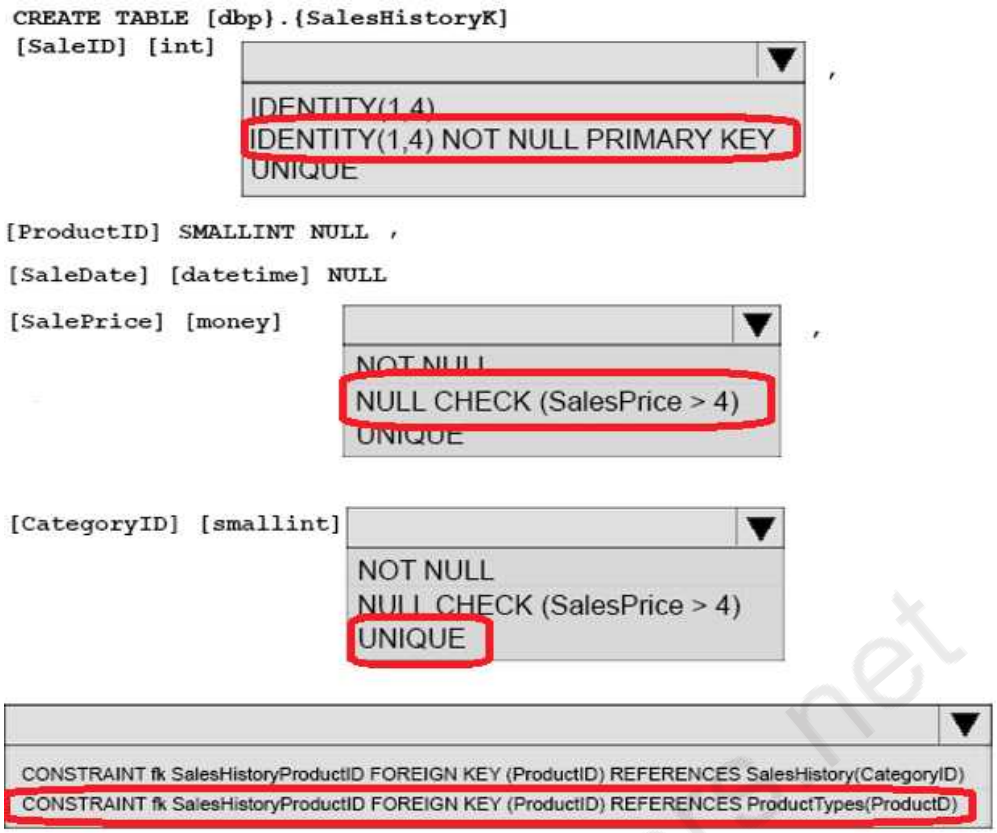
You must modify the ProductReview Table to meet the following requirements:
* The table must reference the ProductID column in the Product table
* Existing records in the ProductReview table must not be validated with the Product table.
* Deleting records in the Product table must not be allowed if records are referenced by the ProductReview table.
* 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:
* Create new rows in the table without granting INSERT permissions to the table.
* 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.

**Answer Area**

```
IF OBJECT_id(*SalesHistory*)>0 DROP TABLE SalesHistory
GO
IF OBJECT_ID(*ProductTypes*)>0 DROP TABLE  ProductTypes
GO
CREATE TABLE ProductTypes
(
    ProductID SMALLINT,
    ProductDescripltion VARCHAR(255),
    CONSTRANT pk_ProductID PRIMARY KEY (ProductID)
)
GO
CREATE TABLE [dbp].{SalesHistoryK]
[SaleID] [int]  [ IDENTITY(1,4) | IDENTITY(1,4) NOT NULL PRIMARY KEY | UNIQUE ] ,
[ProductID] SMALLINT NULL ,
[SaleDate] [datetime] NULL
[SalePrice] [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(ProductD) ]
)
GO
```

**Answer:**




Explanation:
Box 1:
SaleID must be the primary key, as a constraint on the SaleID column that allows the field to be used as a record identifier is required.
Box2:
A constraint that limits the SalePrice column to values greater than four.
Box 3: UNIQUE
A constraint on the CategoryID column that allows one row with a null value in the column.
Box 4:
A foreign key constraint must be put on the productID referencing the ProductTypes table, as a constraint that uses the ProductID column to reference the Product column of the ProductTypes table is required.
Note: Requirements are:
You must add the following constraints to the SalesHistory table:

# Thank You for trying 70-762 PDF Demo

To Buy Latest 70-762 Full Version Download visit link below

https://www.certkillers.net/Exam/70-762

# Start Your 70-762 Preparation