

IT 认证电子书



质 量 更 高 服 务 更 好

半年免费升级服务

<http://www.itrenzheng.com>

Exam : 70-762

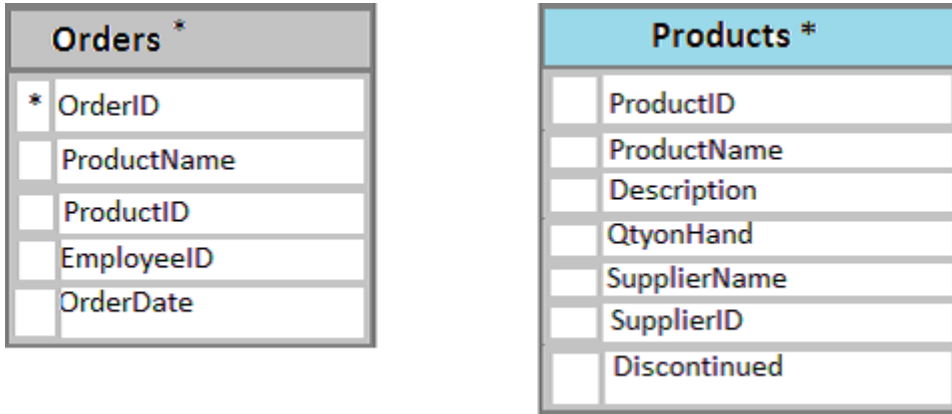
Title : Developing SQL Databases

Version : DEMO

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.

Objects

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

Answer Area

Table	Objects
Orders	
Products	

Answer:

Objects

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

Answer Area

Table	Objects
Orders	Foreign key constraint
Products	Primary key constraint

Explanation:

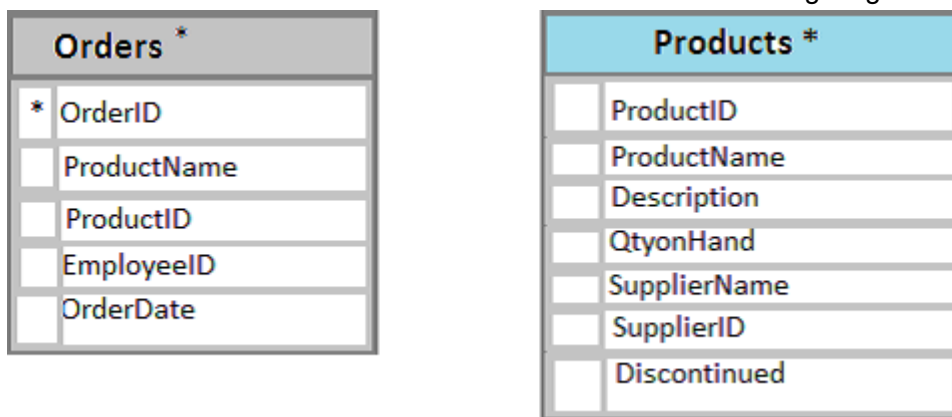
The Products table needs a primary key constraint on the ProductID field. The Orders table needs a foreign key constraint on the productID field, with a reference to the ProductID field in the Products table.

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	<table border="1"> <tr><td>Try/Parse</td></tr> <tr><td>Select @@error</td></tr> <tr><td>Begin Tran/Rollback Tran</td></tr> <tr><td>Try/Catch*</td></tr> </table>	Try/Parse	Select @@error	Begin Tran/Rollback Tran	Try/Catch*
Try/Parse					
Select @@error					
Begin Tran/Rollback Tran					
Try/Catch*					
Display error message	<table border="1"> <tr><td>ERROR MESSAGE()</td></tr> <tr><td>PRINT</td></tr> <tr><td>RAISERROR</td></tr> <tr><td>RETURN</td></tr> </table>	ERROR MESSAGE()	PRINT	RAISERROR	RETURN
ERROR MESSAGE()					
PRINT					
RAISERROR					
RETURN					

Answer:

Answer Area

Requirement	Transact-SQL segment				
Handle errors	<table border="1"> <tr><td>Try/Parse</td></tr> <tr><td>Select @@error</td></tr> <tr><td>Begin Tran/Rollback Tran</td></tr> <tr style="border: 2px solid red;"><td>Try/Catch*</td></tr> </table>	Try/Parse	Select @@error	Begin Tran/Rollback Tran	Try/Catch*
Try/Parse					
Select @@error					
Begin Tran/Rollback Tran					
Try/Catch*					
Display error message	<table border="1"> <tr style="border: 2px solid red;"><td>ERROR MESSAGE()</td></tr> <tr><td>PRINT</td></tr> <tr><td>RAISERROR</td></tr> <tr><td>RETURN</td></tr> </table>	ERROR MESSAGE()	PRINT	RAISERROR	RETURN
ERROR MESSAGE()					
PRINT					
RAISERROR					
RETURN					

Explanation:

Using TRY...CATCH in Transact-SQL

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

TRY...CATCH can use the following error function to capture error information: ERROR_MESSAGE () returns the complete text of the error message. The text includes the values supplied for any substitutable parameters such as lengths, object names, or times.

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

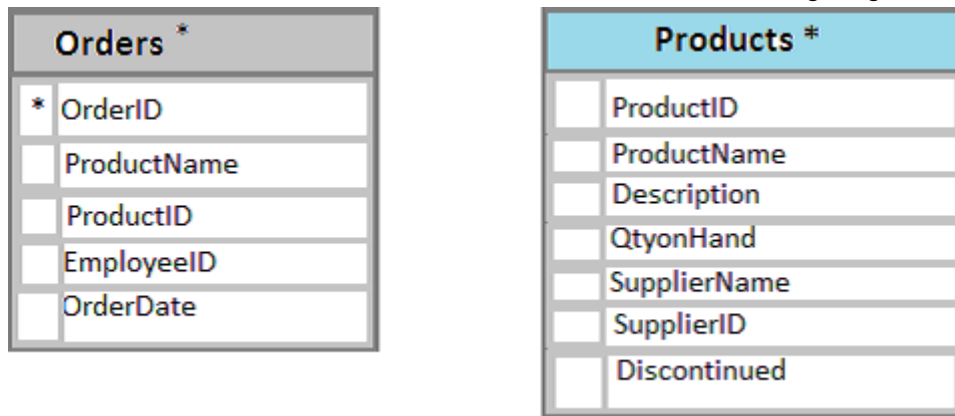
3.HOTSPOT

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	<input type="radio"/>	<input type="radio"/>
INSTEAD OF INSERT trigger	<input type="radio"/>	<input type="radio"/>
AFTER UPDATE trigger	<input type="radio"/>	<input type="radio"/>
INSTEAD OF UPDATE trigger	<input type="radio"/>	<input type="radio"/>

Answer:

Answer Area

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

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.

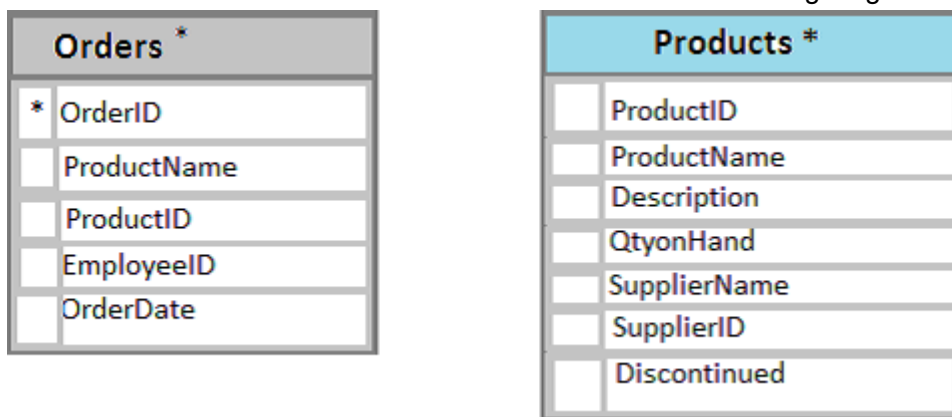
References: [https://technet.microsoft.com/en-us/library/ms175089\(v=sql.105\).aspx](https://technet.microsoft.com/en-us/library/ms175089(v=sql.105).aspx)

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.

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"/>

Answer:

Answer Area

Data type	LeadID	IsCustomer
smallint	<input checked="" 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 checked="" type="radio"/>

Explanation:

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^{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>

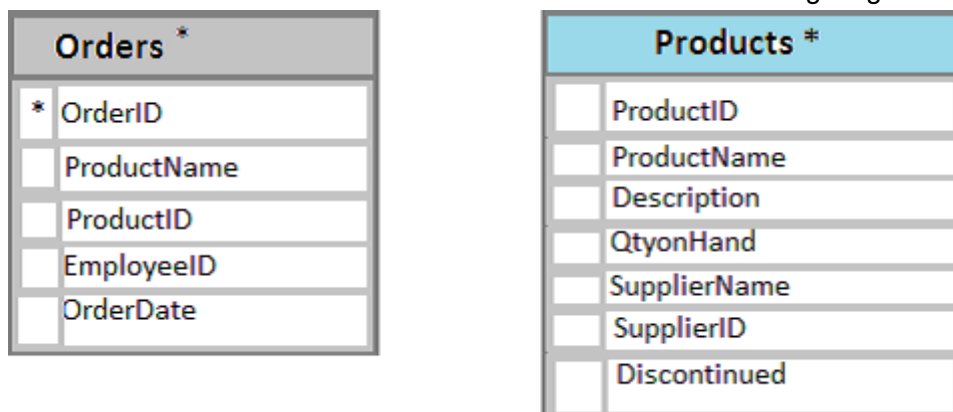
<https://msdn.microsoft.com/en-us/library/ms177603.aspx>

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	<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"/>

Answer:

Answer Area

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"/>

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