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 : **70-229**

Title : Designing and Implementing Databases with Microsoft SQL Server 2000, Enterprise Edition

Version : DEMO

1. You are designing an inventory and shipping database for Contoso.Ltd. You create the logical database design shown in the exhibit.

You must ensure that the referential integrity of the database is maintained. which three types of constraints should you apply to your design?

A. Create a FOREIGN KEY constraint on the Products table that references the Order Details table

B. Create a FOREIGN KEY constraint on the Products table that references the Suppliers table

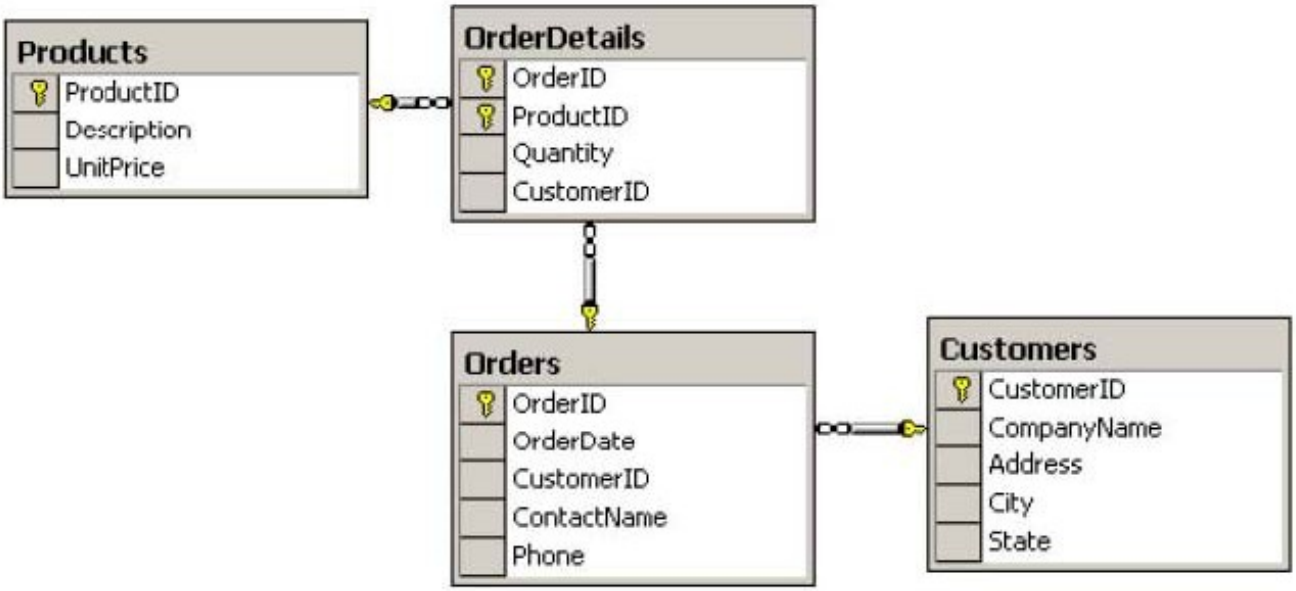
C. Create a FOREIGN KEY constraint on the Orders table that references the Order Details table

D. Create a FOREIGN KEY constraint on the Order Details table that references the Orders table

E. Create a FOREIGN KEY constraint on the Order Details table that references the Products table

F. Create a FOREIGN KEY constraint on the Suppliers table that references the Products table
Answer: BDE

2. You are designing a database for Tailspin Toys. You review the database design. which is shown in the exhibit



You want to promote quick response times for queries and minimize redundant data. what should you do?

A. Create a new table named CustomerContact. Add CustomerID,ContactName,and Phone columns to this table

B. Create a new composite PK constraint on the Orderdetails table Include the OrderID,ProductId and CustomerIDcomlumns in the contraint

C. Remove the PK constraint from the OrderDetails table Use and IDENTITY column to create a surrogate key for theOrderDetails table

D. Remove the CustomerID column from the OrderDetails table

E. Remove the Quantity column from the OrderDetails table Add a Quantity column to the Orders table
Answer: D

3. You are a database developer for an insurance company. You create a table named Insured. which will contain information about persons covered by insurance policies. You use the script shown in the exhibit to create this table. Create table dbo.insured (InsuredID int identity(1,1) not null, Policyid int not null, Insurename char(30) not null, Insuredbirthdate datatime not null, Constraint pk_insured primary key clustered (insuredID), constraint fk_insured_policy foreign key(policyID) references dbo.policy(policyID)) A person covered by an insurance policy is uniquely identified by his or her name and birth data. An insurance policycan cover more than one person. A person cannot be covered more than once by the same insurance policy. You must ensure that the database correctly enforces the relationship between insurance policies and the persons coveredby insurance policies.what should you do?

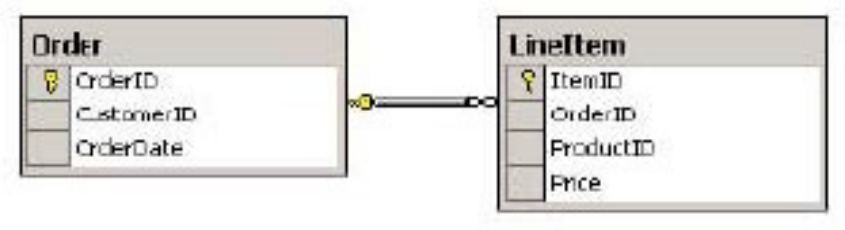
A. Add the PolicyID.InsuredName,and InsuredBirthDate columns to the primary key

B. Add a UNIQUE constraint to enforce the uniqueness of the combination of the PolicyID, InsuredName,and InsuredBirthDate columns

C. Add a CHECK constraint to enforce the uniqueness of the combination of the PolicyID.InsuredName,and InsuredBirthDate columns.

D. Create a clustered index on the PolicyID.InsuredName,and InsuredBirthDate columns
Answer: B

4. You are a database developer for Wingtip Toys. You have created an order entry database that includes two tables, as shown in the exhibit.



Users enter orders into an order entry application. when a new order is entered, the data is saved to the Order andLineItem tables in the order entry database. You must ensure that the entire order is saved successfully. which script should you use?

A. BEGIN TRANSACTION Order INSERT INTO Order VALUES(@ID,@CustomerID,@OrderDate) INSERT INTOLineItem VALUES(@ItemID,@ID,@ProductID,@Price) SAVE TRANSACTION Order

B. INSERT INTO Order VALUES(@ID,@CustomerID,@OrderDate) INSERT INTO LineItemVALUES(@ItemID,@ID,@ProductID,@Price) IF (@@Error = 0) COMMIT TRANSACTION ELSE ROLLBACK TRANSACTION

C. BEGIN TRANSACTION INSERT INTO Order VALUES(@ID,@CustomerID,@OrderDate) IF (@@Error = 0) BEGIN INSERT INTO LineItemVALUES (@ItemID,@ID,@ProductID,@Price) IF(@@Error = 0) COMMITTRANSACTION ELSE ROLLBACK TRANSACTION END ELSE ROLLBACK TRANSACTION

D. BEGIN TRANSACTION INSERT INTO Order VALUES(@ID,@CustomerID,@OrderDate) IF (@@Error = 0) COMMIT TRANSACTION ELSE ROLLBACK TRANSACTION BEGIN TRANSACTION INSERT INTO LineItemVALUES(@ItemID,@ID,@ProductID,@Price) IF (@@Error = 0) COMMIT TRANSACTION ELSE ROLLBACK TRANSACTION
Answer: C

5. You are a database developer for Proseware,Inc. The company has a database contains information about companieslocated within specific postal codes. This information is contained in the Company table within this database. Currently, the database contains company data for five different postal codes. The

number of companies in a specific postal codecurrently ranges from 10 to 5,000.More companies and postal codes will be added to the database over time. You are creating a query to retrieve information from the database. You need to accommodate new data by making onlyminimal changes to the database. the performance of your query must not be affected by the number of companiesreturned. You want to create a query that performs consistently and minimizes future maintenance. what should you do?

A. Create a stored procedure that requires a postal code as a parameter Include the WITH RECOMPILE option when the procedure is created

B. Create one stored procedure for each postal code

C. Create one view for each postal code

D. Split the Company table into multiple tables so that each table contains one postal code. Build a portioned view on thetables so that the data can still be viewed as a single table
Answer: A

6. You are a database developer for a vacuum sales company. The company has a database named Sales that contains tables named VacuumSales and Employee. Sales information is stored in the VacuumSales table. Employee informationis stored in the EmployeeID that uniquely identifies each employee. All sales entered into the VacuumSales table mustcontain an employee ID of a currently employed employee. How should you enforce this requirement?

A. Use the MSDTC to enlist the Employee table in a distributed transaction that will roll back the entire transaction if the employee ID is not active

B. Add a CHECK constraint on the EmployeeID column of the VacuumSales table

C. Add a FK constraint on the EmployeeID column of the VacuumSales table that references the EmployeeID column in the Employee table

D. Add a FOR INSERT trigger on the VacuumSales table In the trigger,join the Employee table with the inserted tablebased on the EmployeeID column,and test the IsActive column
Answer: D

7. You are a database developer for a large brewery, Information about each of the brewery??s plants and the equipment located at each plant is stored in a database named Equipment. The plant information is stored in a table named Location,and the equipment information is stored in a table named Parts. The scripts that were used to create these tables areshown in the location and Parts Scripts exhibit. CREATE TABLE Location(LocationID int NOT NULL, LocationName char (30) NOT NULL UNIQUE, CONSTRAINT PK_location PRIMARY KEY (LocationID)) CREATE TABLE Parts(PartID int NOT NULL,

LocationID int NOT NULL, PartName char (30) NOT NULL, CONSTRAINT PK_Parts PRIMARY KEY (PartID), CONSTRAINT FK_Partslocation FOREIGN KEY (LocationID) REFERENCES location (LocationID)) the brewery is in the process of closing several existing plants and opening several new

plants. when a plant is closed, the information about the plant and all of the equipment at that must be deleted from the database. you have created a stored procedure to perform this operation. the stored procedure is shown in the Script for sp_Deletlocation exhibit. CREATE PROCEDURE sp_deletelocation @locName char(30) AS BEGIN DECLARE @PartID int DECLARE crs_Parts, CURSOR FOR SELECT p.PartID FROM Parts AS p INNER JOIN Location AS l ON p.LocationID = l.LocationID WHERE l.LocationName = @locName OPEN crs_parts FETCH NEXT FROM crs_Parts INTO @PartID WHILE (@@FETCH STATUS <> 1) BEGIN DELETE Parts WHERE CURRENT OF crs_Parts FETCH NEXT FROM = Part, INTO @PartID END CLOSE crs_Parts DEALLOCATE crs_Parts DELETE Location WHERE LocationName = @locName END This procedure is taking longer than expected to execute. You need to reduce the execution time of the procedure. What should you do?

A. Add the WITH RECOMPILE option to the procedure definition

B. Replace the cursor operation with a single DELETE statement

C. Add a BEGIN TRAN statement to the beginning of the procedure, and add a COMMIT TRAN statement to the end of the procedure

D. Set the transaction isolation level to READ UNCOMMITTED for the procedure

E. Add a nonclustered index on the PartID column of the Parts table

Answer: B

8. You are a database developer for a travel agency. A table named FlightTimes in the Airlines database contains flight information for all airlines. The travel agency uses an intranet-based application to manage travel reservations. thisapplication retrieves flight information for each airline from the FlightTimes table. Your company primarily works withone particular airline. In the airlines database, the unique identifier for this airline is 101. The application must be able to request flight times without having to specify a value for the airline. The application should be required to specify a value for the airline only if a different airline??s flight times are needed?What should you do?

A. Create two stored procedures,and specify that one of the stored procedures should accept a parameter and that the other should not

B. Create a user-defined function that accepts a parameter with a default value of 101

C. Create a stored procedure that accepts a parameter with a default value of 101

D. Create a view that filters the FlightTimes table on a value of 101

E. Create a default of 101 on the FlightTimes table

Answer: C

9. You are a database developer for a company that provides consulting services. The company maintains data about itsemployees in a table named Employee. the script that was used to create the Employee table is shown in the exhibit.CREATE TABLE Employee(EmployeeID int NOT NULL, EmpType char(1) NOT NULL, EmployeeName char(50) NOT NULL, Address char(50) NULL, Phone char(20) NULL, CONSTRAINT PK Employee PRIMARY KEY (EmployeeID)) The EmpType column in this table is used to identify employees as executive, administrative, or consultants. You need toensure that the administrative employees can add, update, or delete data for non-executive employees only. What should you do?

A. Create a view ,and include the WITH ENCRYPTION clause

B. Create a view ,and include the WITH CHECK OPTION clause

C. Create a view ,and include the SCHEMABINDING clause

D. Create a view ,and build a covering index on the view

E. Create a user-defined function that returns a table containing the non-executive employees.

Answer: B

10. You are a database developer for a company that leases trucks. The company has created a Web site that customers can use to reserve trucks. You are designing the SQL Server 2000 database to support the Web site.New truck reservations are inserted into a table named Reservations. Customers who have reserved a truck can return tothe Web site and update their reservation. When a reservation is updated, the entire existing reservation must be copied toa table named History.Occasionally, customers will save an existing reservation without actually changing any of the information about thereservation, In this case, the existing reservation should not be copied to the History table. You need to develop a way to create the appropriate entries in the History table. what should you do?

A. Create a trigger on the reservation table to create History table entries

B. Create a cascading referential integrity constraint on the reservation table to create History table entries

C. Create a view on the reservation table Include the WITH SCHEMABINDING option in the view definition

D. Create a view on the reservation table Include the WITH CHECK OPTION clause in the view definition

Answer: A

11. You are a database developer for a clothing retailer. The company has database named Sales. This database contains a table named Inventory. The Inventory table contains the list of items for sale and the

quantity available for each of thoseitems. When sales information is inserted into the database, this table is updated. The stored procedure that updates the Inventory table is shown in the exhibit.

```
CREATE PROCEDURE UpdateInventory @IntID int
AS
BEGIN

DECLARE @Count int

BEGIN TRAN

SELECT @Count = Available
FROM Inventory WITH (HOLDLOCK)
WHERE InventoryID = @IntID

IF (@Count > 0)
UPDATE Inventory SET Available = @Count - 1
WHERE InventoryID = @IntID

COMMIT TRAN

END
```

When this procedure executes, the database server occasionally returns the following error message: Transaction(Process ID 53) was deadlocked on (lock)resources with another process and has been chosen as the deadlock victim. return the transaction. You need to prevent the error message from occurring while maintaining data integrity. what should you do?

A. Remove the table hint.

B. Change the table hint to UPDLOCK

C. Change the table hint to REPEATABLEREAD

D. Set the transaction isolation level to SERIALIZABLE

E. Set the transaction isolation level to REPEATABLEREAD
Answer: B

12. You are a database developer for an online brokerage firm. The prices of the stocks owned by customers are maintained in a SQL Server 2000 database. To allow tracking of the stock price history, all updates of stock prices must be logged. To help correct problems regarding price updates, and errors that occur during an update must also be logged. When errors are logged, a messagethat identifies stock producting the error must be returned to the client application. You must ensure that the appropriate

conditions are logged and that the appropriate messages are generated.Which procedure should you use?

A. CREATE PROCEDURE UpdateStockPrice @StockID int,@Price decimal AS BEGIN DECLARE @Msg

varchar(50) UPDATE Stocks SET CurrentPrice = @Price WHERE StockID = @StockID AND CurrentPrice <>@Price IF @@ERROR <> 0 RAISERROR (??Error %d occurred updating Stock %d.??,10,1 @@ERROR, @StockID) WITH LOG IF @@ROWCOUNT>0 BEGIN SELECT @Msg = ??Stock??+STR(@StockID)+??updated to??+STR(@Price)+??.?? EXEC master..xp_logevent 50001 END END ,@Msg

B. CREATE PROCEDURE UpdateStockPrice @StockID int,@Price decimal AS BEGIN UPDATE Stocks SETCurrentPrice = @Price WHERE StockID = @StockID AND CurrentPrice <>@Price IF @@ERROR <> 0 PRINT ??Error ??+ STR(@StockID)+??occurred updating Stock??+STR(@StockID)+??.?? IF @@ROWCOUNT>0BEGIN PRINT ??Stock??+ STR(@StockID)+??update to?? +STR(@Price)+??.?? END

C. CREATE PROCEDURE UpdateStockPrice @StockID int,@Price decimal AS BEGIN DECLARE @Errint,@Rcount int,@Msg varchar(50) UPDATE Stocks SET CurrentPrice = @Price WHERE StockID = @StockID ANDCurrentPrice <>@Price SELECT @Err =@@ERROR,@Rcount=@@ROWCOUNT IF @Err <> 0 BEGIN SELECT@Msg = ??Error??+STR(@Err)+??occurred updating Stock??+STR(@StockID)+??.?? EXEC master..xp_logevent 50001,@Msg END IF @RCount <> 0 BEGIN SELECT @Msg = ??Stock??+STR(@StockID)+??updated to??+STR(@Price)+??.?? EXEC master..xp_logevent 50001,@Msg END END

D. CREATE PROCEDURE UpdateStockPrice @StockID int,@Price decimal AS BEGIN DECLARE @Err int,@Rcount int,@Msg varchar(50) UPDATE Stocks SET CurrentPrice = @Price WHERE StockID = @StockID ANDCurrentPrice <>@Price SELECT @Err =@@ERROR,@Rcount=@@ROWCOUNT IF @Err <> 0 BEGINRAISERROR (??Error %d occurred updating Stock %d.??,10,1 @@ERROR, @StockID) WITH LOG IF @@RCount>0 BEGIN SELECT @Msg = ??Stock??+STR(@StockID)+??updated to??+STR(@Price)+??.?? EXECmaster..xp_logevent 50001,@Msg END END
Answer: D

13. You are a database developer for a sales organization. Your database has a table named Sales that contains summary information regarding the sales orders from salespeople. The sales manager asks you to create a report of the salespeople who had the 20 highest total sales. Which query should you use to accomplish this?

A. SELECT TOP 20 PERCENT LastName,FirstName,SUM(OrderAmount) AS ytd FROM Sales GROUP BYLastName,FirstName ORDER BY 3 DESC

B. SELECT TOP 20 LastName,FirstName,COUNT(*) AS Sales FROM Sales GROUP BY LastName,FirstNameHAVING COUNT(*)>30 ORDER BY 3 DESC

C. SELECT TOP 20 LastName,FirstName,MAX(OrderAmount) FROM Sales GROUP BY LastName,FirstNameORDER BY 3 DESC

D. SELECT TOP 20 LastName,FirstName, SUM(OrderAmount) AS ytd FROM Sales GROUP BY LastName,FirstNameORDER BY 3 DESC

E. SELECT TOP 20 WITH TIES LastName,FirstName, SUM(OrderAmount) AS ytd FROM Sales GROUP BYLastName,FirstName ORDER BY 3 DESC
Answer: E

14. You are a database developer for Woodgrove Bank. You are implementing a process that loads data into a SQL Server2000 database. As a part of this process, data is temporarily loaded into a table named Staging. When the data loadprocess is complete, the data is deleted from this table. You will never need to recover this deleted data. You need to ensure that the data from the Staging table is deleted as quickly as possible. What should you do?

A. Use a DELETE statement to remove the data from the table

B. Use a TRUNCATE TABLE statement to remove the data from the table

C. Use a DROP TABLE statement to remove the data from the table

D. Use an updatable cursor to access and remove each row of data from the table
Answer: B

15. You are a database developer for Adventure Works. You are designing a script for the human resources departmentthat will report yearly wage information. There are three types of employees. Some employees earn an hourly wage, some are salaried, and some are paid commission on each sale that they make.This data is recorded in a table named Wages, which was created by using the following script: CREATE TABLE wages (emp_id tinyint identity, hourly_wage decimal NULL, salary decimal NULL,commission decimal NULL, num_sales tinyint NULL) An employee can have only one type of wage information. You must correctly report each employee??s yearly wage information. Which script should you use?

A. SELECT CAST(COALESCE(hourly_wage * 40 * 52 + Salary+commission * num_sales) AS money) AS YearlyWages FROM wages

B. SELECT CAST(COALESCE(hourly_wage * 40 * 52 , Salary,commission * num_sales) AS money) AS YearlyWages

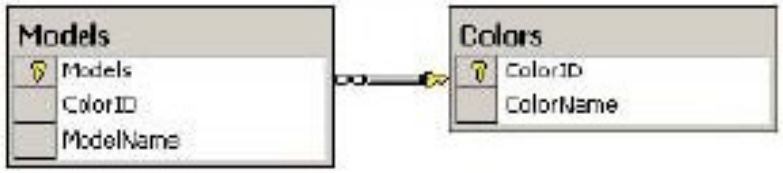
FROM wages

C. SELECT CAST(CASE WHEN ((hourly_wage,)IS NOTNULL) THEN hourly_wage* 40 * 52 WHEN (NULLIF(salary,NULL) IS NULL)THEN salary ELSE commission * num_sales ENDAS MONEY) AS YearlyWages FROM Wages

D. SELECT CAST(CASE WHEN ((hourly_wage,)IS NULL) THEN salary WHEN (salary IS NULL)THEN commission

* num_sales ELSE commission * num_sales END AS MONEY) AS YearlyWages FROM Wages
Answer: B

16. You are a database developer for an automobile dealership. You are designing a database to support a Web site that will be used for purchasing automobiles. A person purchasing an automobile from the Web site will be able to customize his or her order by selecting the model and color. The manufacturer makes four different models of automobiles. The models can be ordered in any one of five colors. A default color is assigned to each model. The model are stored in a table named Models, and the colors are stored in a table named Colors. These tables are shown in the exhibit.



You need to create a list of all possible model and color combinations. which script should you use?

A. SELECT m.ModelName,c.ColorName FROM Colors AS c FULL OUTER JOIN Models AS m ON c.ColorID = m.ColorID ORDER BY m.ModelName,c.ColorName

B. SELECT m.ModelName,c.ColorName FROM Colors AS c CROSS JOIN Models AS m ORDER BY m.ModelName,c.ColorName

C. SELECT m.ModelName,c.ColorName FROM Colors AS c INNER JOIN Models AS m ON c.ColorID = m.ColorID ORDER BY m.ModelName,c.ColorName

D. SELECT m.ModelName,c.ColorName FROM Colors AS c LEFT OUTER JOIN Models AS m ON c.ColorID = m.ColorID UNION SELECT m.ModelName,c.ColorName FROM Colors AS c RIGHT OUTER JOIN Models AS m ON c.ColorID = m.ColorID ORDER BY m.ModelName,c.ColorName

E. E. SELECT m.ModelName FROM Models AS m UNION SELECT c.ColorName FROM Colors AS c ORDER BY m.ModelName
Answer: B

17. You are a database developer for an insurance company. The company has a database named Policies. You havedesigned stored procedures for this database that will use cursors to process large result sets. Analysts who use the storedprocedures report that there is a long initial delay before data is displayed to them. After the delay, performance isadequate. Only data analysts, who perform data analysis, use the Policies database.You want to improve the performance of the stored procedures. Which script should you use?

A. EXEC sp_configure ??cursor threshold??,0

B. EXEC sp_dboption ??Policies?? SET CURSOR_CLOSE_ON_COMMIT ON

C. SET TRANSACTION ISOLATION LEVEL SERIALIZABLE

D. ALTER DATABASE Policies SET CURSOR_DEFAULT LOCAL
Answer: A

18. You are a database developer for Wide World importers. The company tracks its order information in a SQLServer2000 database. The database includes two tables that contain order details. The tables are named Order and LineItem. The script that was used to create these tables is shown in the exhibit. Create table dbo.orde(Orderid int not null,Customerid int not null, Orderdate datetime not null,Constraint df_order_orderdate default (getdate()) for orderdate, Constraint pk_order primary key clustered(orderid)) Create table dbo.lineitem(Itemid int not null, Orderid int not null, Productid int not null, Price money not null, Constraint pk_lineitem primary key clustered(itemid), Constraint fk_lineitem_order foreign key(orderid) References dbo.order(orderid)) the company??s auditors have discovered that every item that was ordered on June 1,2000,was entered with a price that was $10 more than its actual price. You need to correct the data in the database as quickly as possible. Which script should you use?

A. UPDATE 1 SET Price = Price ?????¡ì??C10 FROM LineItem AS l INNER JOIN [Order] AS o ON l.OrderID = o.OrderID WHERE o.OrderDate >= ??6/1/2000?? AND o.OrderDate < ??6/2/2000??

B. UPDATE 1 SET Price = Price ?????¡ì??C1 FROM LineItem AS l INNER JOIN [Order] AS o ON l.OrderID =

o.OrderID WHERE o.OrderDate >= ??6/1/2000??

C. DECLARE @ItemID int DECLARE items_cursor CURSOR FOR SELECT l.ItemID FROM LineItem AS l INNER JOIN [Order] AS o ON l.OrderID = o.OrderID WHERE o.OrderDate >= ??6/1/2000?? AND o.OrderDate < ??6/2/2000?? FOR UPDATE OPEN items_cursor FETCH NEXT FROM items_cursor INTO @ItemID WHILE @@FETCH_STATUS = 0 BEGIN UPDATE LineItem SET Price = Price ?????¡ì??C10 WHERE CURRENT OF items_cursor FETCH NEXT FROM items_curor INTO @ItemID END CLOSE items_cursor DEALLOCATE items_cursor

D. DECLARE @ItemID int DECLARE order_cursor CURSOR FOR SELECT OrderID FROM [Order] WHERE o.OrderDate = ??6/1/2000?? OPEN order_cursor FETCH NEXT FROM order_cursor INTO @OrderID WHILE @@FETCH_STATUS = 0 BEGIN UPDATE LineItem SET Price = Price ?????¡ì??C10 WHERE CURRENT OF items_cursor FETCH NEXT FROM order_cursor r INTO @OrderID END CLOSE order_cursor DEALLOCATE i order_cursor
Answer: A

19. You are a database developer for a bookstore. Each month, you receive new supply information form your vendors in the form of an XML document. The XML document is shown in the XML document exhibit.

XML Document

```
<ROOT>
<CategoryID= "2" CategoryName= "Videos">
      <Product ProductID= "80248" Description= "7 Minute Abs">
      </Product>
</Category>
<Category CategoryID= "3" CategoryName= "Computer Books">
  <Product ProductID= "12345" Description= "Inside SQL Server 2000">
     </Product>
</Category>
<Category CategoryID= "3" CategoryName= "Computer Books">
  <Product ProductID= "22345" Description= "Analysis Services with SQL
Server 2000">
    </Product>
</Category>
<ROOT>
```

You are designing a stored procedure to read the XML document and to insert the data to a table named Products. The Products table is show in the Products table exhibit.

Which script should you use to create this stored procedure?

A. CREATE PROCEDURE spAddCataLogItems (@xmlDocument varchar(8000)) AS BEGIN DECLARE @docHandle int EXEC sp_xml_preparedocument @docHandle OUTPUT, @xmlDocument INSERT INTO ProductsSELECT * FROM OPENXML (@docHandle,??/ROOT/Category/Product??,1) WITH Products EXEC sp_xml_removedocument @docHandle END

B. CREATE PROCEDURE spAddCataLogItems (@xmlDocument varchar(8000)) AS BEGIN DECLARE@docHandle int EXEC sp_xml_preparedocument @docHandle OUTPUT, @xmlDocument INSERT INTO Products SELECT * FROM OPENXML (@docHandle,??/ROOT/Category/Product??,1) WITH (ProductID int ??./@ProductID,CategoryID int ,../@CategoryID. [Description] varchar (100)??./@Description??) EXEC sp_xml_removedocument@docHandle END

C. CREATE PROCEDURE spAddCataLogItems (@xmlDocument varchar(8000)) AS BEGIN INSERT INTOProducts SELECT * FROM OPENXML (@docHandle,??/ROOT/Category/Product??,1) WITH(ProductIDint,Description varchar(50)) END

D. CREATE PROCEDURE spAddCataLogItems (@xmlDocument varchar(8000)) AS BEGIN INSERT INTOProducts SELECT * FROM OPENXML (@xmlDocument,??/ROOT/Category/Product??,1) WITH

Products END
Answer: B

20. You are a database developer for a bookstore. You are designing a stored procedure to process XML documents. Youuse the following script to create the stored procedure: CREATE PROCEDURE spParseXML (@xmlDocument varchar(1000)) ASDECLARE @docHandle int EXEC sp_xml_preparedocument @docHandle OUTPUT, @xmlDocument SELECT * FROM OPENXML (@docHandle,??/ROOT/Category/Product??,2)WITH(ProductID int,CategoryID int, CategoryName varchar(50)

[Description] varchar(50))EXEC sp_xml_removedocument @docHandle You execute this stored procedure and use an XML document as the input document. the XML document is shown in theXML document exhibit

XML Document

```
<ROOT>
<Category CategoryID= "1" CategoryName= "General Books">
     <Product  ProductID="10248"  Description="Cooking  for
you">
     </Product>
</Category>
<Category CategoryID= "2" CategoryName= "Videos">
  <Product ProductID= "80248" Description= "7 Minute Abs">
     </Product>
</Category>
<Category CategoryID= "3" CategoryName= "Computer Books">
  <Product  ProductID=  "12345"  Description=  "Inside  SQL
Server 2000">
    </Product>
    <Product  ProductID=  "22345"  Description=  "Analysis
Services with SQL Server 2000">
     </Product>
</Category>
<ROOT>
```

You receive the output shown in the Output exhibit.

Output

ProductID	CategoryID	CategoryName	Description
NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL

(4 row(s) affected)

You need to replace the body of the stored procedure. which script should you use?

A. SELECT * FROM OPENXML (@docHandle,??/ROOT/Category/Product??,1) WITH(ProductID int, CategoryID int, CategoryName varchar(50) [Description] varchar(50))

B. SELECT * FROM OPENXML (@docHandle,??/ROOT/Category/Product??,8) WITH(ProductID int, CategoryID int, CategoryName varchar(50) [Description] varchar(50))

C. SELECT * FROM OPENXML (@docHandle,??/ROOT/Category/Product??,1) WITH(ProductID int, CategoryID int ??@CategoryID??, CategoryName varchar(50) ??@CategoryID??, [Description] varchar(50))

D. SELECT * FROM OPENXML (@docHandle,??/ROOT/Category/Product??,1) WITH(ProductID int, CategoryID int ??../@CategoryID??, CategoryName varchar(50) ??../@CategoryID??, [Description] varchar(50))
Answer: D

This document was created with Win2PDF available at http://www.win2pdf.com.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.
This page will not be added after purchasing Win2PDF.