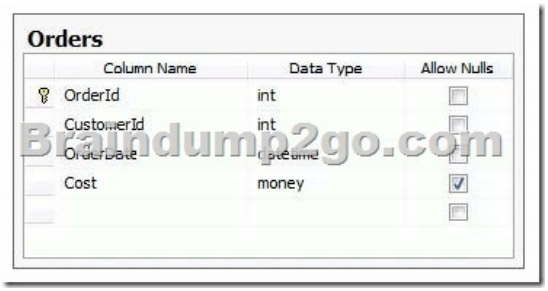
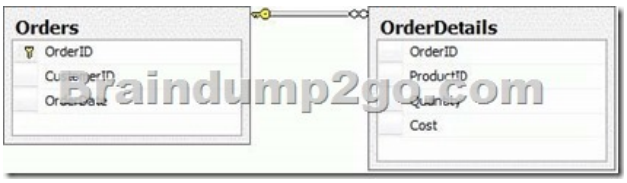
Official New Updated 70-433 Exam Questions and Answers Shared By Braindump2go For Free Download Today! (111-120)

Do you want to pass Microsoft 70-433 Exam ? If you answered YES, then look no further. Braindump2go offers you the best 70-433 exam questions which cover all core test topics and certification requirements.All REAL questions and answers from Microsoft Exam Center will help you be a 70-433 certified! Exam Code: 70-433Exam Name: TS: Microsoft SQL Server 2008, Database DevelopmentCertification Provider: MicrosoftKeywords: 70-433 Exam Dumps,70-433 Practice Tests,70-433 Practice Exams,70-433 Exam Questions,70-433 PDF,70-433 VCE Free,70-433 Book,70-433 E-Book,70-433 Study Guide,70-433 Braindump,70-433 Prep Guide

QUESTION 111You have a table named Orders. OrderID is defined as an IDENTITY(1,1). OrderDate has a default value of 1. You need to write a query to insert a new order into the Orders table for CustomerID 45 with today's date and a cost of 89.00. Which statement should you use?



A. INSERT INTO Orders (CustomerId, OrderDate, Cost) VALUES (45, DEFAULT, 89.00); B. INSERT INTO Orders (OrderID, CustomerId, OrderDate, Cost) VALUES (1, 45, DEFAULT, 89.00); C. INSERT INTO Orders (CustomerId, OrderDate, Cost) VALUES (45, CURRENT_TIMESTAMP, 89.00); D. INSERT INTO Orders (OrderID, CustomerId, OrderDate, Cost) VALUES (1, 45, CURRENT_TIMESTAMP, 89.00); Answer: C QUESTION 112You have the following two tables. The foreign key relationship between these tables has CASCADE DELETE enabled. You need to remove all records from the Orders table. Which Transact-SQL statement should you use?



A. DROP TABLE Orders B. DELETE FROM Orders C. TRUNCATE TABLE Orders D. DELETE FROM OrderDetails Answer: B QUESTION 113You have a SQL Server database. The database contains two schemas named Marketing and Sales. The Marketing schema is owned by a user named MarketingManager. The Sales schema is owned by a user named SalesManager. A user named John must be able to access the Sales.Orders table by using a stored procedure named Marketing.GetSalesSummary. John is not granted a SELECT permission on the Sales.Orders table. A user named SalesUser does have SELECT permission on the Sales.Orders table. You need to implement appropriate permissions for John and the stored procedure Marketing.GetSalesSummary. What should you do? A. Marketing.GetSalesSummary should be created by using the EXECUTE AS 'SalesUser' clause. John should be granted EXECUTE permission on Marketing.GetSalesSummary. B. Marketing.GetSalesSummary should be created by using the EXECUTE AS OWNER clause. John should be granted EXECUTE WITH GRANT OPTION on Marketing.GetSalesSummary. C. Marketing.GetSalesSummary should be created by using the EXECUTE AS CALLER clause. John should be granted IMPERSONATE permission for the user named SalesUser. D. Marketing.GetSalesSummary should be created without an EXECUTE AS clause. John should be granted SELECT permission on the Sales.Orders table. Answer: A

Explanation:1. When the module is executed, the Database Engine first verifies that the user executing the module has EXECUTE permission on the module. So John should be granted EXECUTE permission on Marketing.GetSalesSummary stored procedure.2. Additional permissions checks on objects that are accessed by the module are performed against the user account specified in the EXECUTE AS clause. The user executing the module is, in effect, impersonating the specified user. Because John is not granted a SELECT permission on the Sales.Orders table which is referenced by the stored procedure, EXECUTE AS CALLER is not suitable. (CALLER specifies the statements inside the module are executed in the context of the caller of the module. The user executing the module must have appropriate permissions not only on the module itself, but also on any database objects that are referenced by the module.) Because the user named SalesUser DOES have SELECT permission on the Sales.Orders table, he can be specified in EXECUTE AS clause. It means that Marketing.GetSalesSummary stored procedure should be created by using the EXECUTE AS 'SalesUser' clause. QUESTION 114You have tables named Sales.SalesOrderDetails and Sales.SalesOrderHeader. You have been tasked to update the discount amounts for the sales of a particular salesperson. You need to set UnitPriceDiscount to 0.1 for all entries in Sales.SalesOrderDetail that only correspond to SalesPersonID 290. Which Transact-SQL statement should you use? A. UPDATE d SET UnitPriceDiscount = .1 FROM Sales.SalesOrderDetail d INNER JOIN Sales.SalesOrderHeader h ON h.SalesOrderID = d.SalesOrderID WHERE h.SalesPersonID = 290; B. UPDATE Sales.SalesOrderDetail SET UnitPriceDiscount = .1 FROM Sales.SalesOrderHeader h WHERE h.SalesPersonID = 290; C. UPDATE Sales.SalesOrderDetail SET UnitPriceDiscount = .1 WHERE EXISTS (SELECT * FROM Sales.SalesOrderHeader h WHERE h.SalesPersonID = 290); D. UPDATE Sales.SalesOrderDetail SET UnitPriceDiscount = .1 FROM Sales.SalesOrderDetail dWHERE EXISTS (SELECT * FROM Sales.SalesOrderHeader h WHERE h.SalesPersonID = 290); Answer: A QUESTION 115You have a table named Product. You need to increase product prices for only the vendor named Coho Winery by 10 percent and then return a list of the products and updated prices. Which code segment should you use? A. UPDATE Product SET Price = Price * 1.10, ProductName = ProductName WHERE Product.VendorName = 'Coho Winery' B. UPDATE Product SET Price = Price * 1.10 OUTPUT inserted.ProductName, deleted.Price WHERE Product.VendorName = 'Coho Winery' C. UPDATE Product SET Price = Price * 1.10 OUTPUT inserted.ProductName, inserted.Price WHERE Product.VendorName = 'Coho Winery' D. UPDATE Product SET Price = Price * 1.10, VendorName = 'Coho Winery' OUTPUT inserted.ProductName, inserted.Price Answer: C QUESTION 116 You have two tables named dbo.Products and dbo.PriceChange. Table dbo.Products contains ten products. Five products are priced at $20 per unit and have PriceIncrease set to 1. The other five products are priced at $10 per unit and have PriceIncrease set to 0. You have the following query: INSERT dbo.PriceChange (ProductID, Change, ChangeDate) SELECT ProductID, inPrice -delPrice, SYSDATETIME() FROM (UPDATE dbo.Products SET Price *= 1.1 OUTPUT inserted.ProductID, inserted.Price, deleted.Price WHERE PriceIncrease = 1) p (ProductID, inPrice, delPrice); You need to predict the results of the query. Which results should the query produce? A. Five rows are updated in dbo.Products. Five rows are inserted into dbo.PriceChange. B. Five rows are updated in dbo.Products. No rows are inserted into dbo.PriceChange. C. No rows are updated in dbo.Products. Five rows are inserted into dbo.PriceChange. D. No rows are updated in dbo.Products. No rows are inserted into dbo.PriceChange. Answer: A QUESTION 117You have two tables named MainTable and ArchiveTable. You need to move data older than 30 days from MainTable into ArchiveTable. Which code segment should you use? A. DELETE FROM MainTable OUTPUT deleted.* WHERE RecordDate < DATEADD(D,-30,GETDATE()) B. DELETE FROM MainTable OUTPUT DELETED.* INTO ArchiveTable WHERE RecordDate < DATEADD(D,-30,GETDATE()) C. INSERT INTO ArchiveTable SELECT * FROM MainTable WHERE RecordDate < DATEADD(D,-30,GETDATE()) D. INSERT INTO ArchiveTable SELECT * FROM MainTable WHERE RecordDate < DATEADD(D,-30,GETDATE()) DELETE FROM MainTable Answer: B QUESTION 118You have been tasked with creating a table named dbo.Widgets. You need to insert five rows into the dbo.Widgets table and return WidgetID for each of the five rows that have been inserted. Which Transact-SQL batch should you use? A. CREATE TABLE dbo.Widgets (WidgetID INT IDENTITY PRIMARY KEY, WidgetName VARCHAR(25)); GO INSERT dbo.Widgets (WidgetName) OUTPUT inserted.WidgetID, inserted.WidgetName VALUES ('WidgetOne'),('WidgetTwo'),('WidgetThree'),('WidgetFour'),('WidgetFive'); B. CREATE TABLE dbo.Widgets (WidgetID INT IDENTITY PRIMARY KEY, WidgetName VARCHAR(25)); GO INSERT dbo.Widgets (WidgetName) VALUES ('WidgetOne'),('WidgetTwo'),('WidgetThree'),('WidgetFour'),('WidgetFive'); SELECT SCOPE_IDENTITY(); C. CREATE TABLE dbo.Widgets (WidgetID UNIQUEIDENTIFIER PRIMARY KEY, WidgetName VARCHAR(25)); GO INSERT dbo.Widgets (WidgetName) VALUES ('WidgetOne'),('WidgetTwo'),('WidgetThree'),('WidgetFour'),('WidgetFive'); SELECT SCOPE_IDENTITY(); D. CREATE TABLE dbo.Widgets (WidgetID UNIQUEIDENTIFIER PRIMARY KEY, WidgetName VARCHAR(25)); GO INSERT dbo.Widgets (WidgetName) OUTPUT inserted.WidgetID, inserted.WidgetName VALUES ('WidgetOne'),('WidgetTwo'),('WidgetThree'),('WidgetFour'),('WidgetFive'); Answer: A QUESTION 119You have the following

two tables. Products ProductID ProductName VendorID 1 Product1 0 2 Product2 1 3 Product3 1 4 Product4 0 ProductChanges ProductID ProductName VendorID 1 Product1 1 2 Product2 1 3 NewProduct3 2 5 Product5 1 You execute the following statement. MERGE Products USING ProductChanges ON (Products.ProductID = ProductChanges.ProductID) WHEN MATCHED AND Products.VendorID = 0 THEN DELETE WHEN MATCHED THEN UPDATE SET Products.ProductName = ProductChanges.ProductName Products.VendorID = ProductChanges.VendorID; You need to identify the rows that will be displayed in the Products table. Which rows will be displayed? A. ProductID ProductName VendorID 2 Product2 1 3 NewProduct3 2 B. ProductID ProductName VendorID 2 Product2 1 3 NewProduct3 2 4 Product4 0 C. ProductID ProductName VendorID 1 Product1 1 2 Product2 1 3 NewProduct3 2 5 Product5 1 D. ProductID ProductName VendorID 1 Product1 1 2 Product2 1 3 NewProduct3 2 4 Product4 0 5 Product5 1 Answer: B QUESTION 120You have two tables. A table named Student.CurrentStudents contains the names of all students enrolled for the current year. Another table named Student.NewYearRoster contains the names of students who have enrolled for the upcoming year. You have been tasked to write a MERGE statement to: Insert into Student.CurrentStudents the names of students who are enrolled for the upcoming year but not for the current year. Update information in Student.CurrentStudents for students who are enrolled both in the current year and in the upcoming year. Delete from Student.CurrentStudents the names of students who are not enrolled for the upcoming year. You need to write the appropriate MERGE statement. Which Transact-SQL statement should you use? A. MERGE Student.CurrentStudents AS T USING Student.NewYearRoster AS S ON S.LastName = T.LastName AND S.FirstName = T.FirstNameWHEN MATCHED THEN UPDATE SET Address = S.Address, Age = S.Age WHEN NOT MATCHED BY TARGET THEN INSERT (LastName, FirstName, Address, Age) VALUES (S.LastName, S.FirstName, S.Address, S.Age) WHEN NOT MATCHED BY SOURCE THEN DELETE; B. MERGE Student.CurrentStudents AS T USING Student.NewYearRoster AS S ON S.LastName = T.LastName AND S.FirstName = T.FirstName WHEN MATCHED THEN DELETE WHEN NOT MATCHED THEN INSERT (LastName, FirstName, Address, Age) VALUES (S.LastName, S.FirstName, S.Address, S.Age) WHEN NOT MATCHED BY SOURCE THEN UPDATE SET Address = T.Address, Age = T.Age; C. MERGE Student.CurrentStudents AS T USING Student.NewYearRoster AS S ON S.LastName = T.LastName AND S.FirstName = T.FirstName WHEN MATCHED AND NOT T.Address = S.Address OR NOT T.Age = S.Age THEN UPDATE SET T.Address = S.Address, T.Age = S.Age WHEN NOT MATCHED THEN INSERT (LastName, FirstName, Address, Age) VALUES (S.LastName, S.FirstName, S.Address, S.Age) WHEN MATCHED THEN DELETE; D. MERGE Student.CurrentStudents AS T USING Student.NewYearRoster AS S ON S.LastName = T.LastName AND S.FirstName = T.FirstName WHEN MATCHED AND NOT T.Address = S.Address AND NOT T.Age = S.Age THEN UPDATE SET T.Age = S.Age, T.Address = S.Address WHEN NOT MATCHED BY TARGET THEN INSERT (LastName, FirstName, Address, Age) VALUES (S.LastName, S.FirstName, S.Address, S.Age) WHEN NOT MATCHED BY SOURCE THEN DELETE; Answer: A Braindump2go New Updated 70-433 Exam Dumps are Complete Microsoft 70-433 Course Coverage! 100% Real Questions and Correct Answers Guaranteed! Updated 70-433 Preparation Material with Questions and Answers PDF Instant Download: