Jacob Dineen

IST659 DB Admin Lab #8

8/27/2017

1. Execute the following SQL against your database:

CREATE VIEW ProductCosts As

Select

Product.ProductName

, MIN(VendorProduct.Cost) AS min\_cost

, MAX(VendorProduct.Cost) AS max\_cost

, AVG(VendorProduct.Cost) AS avg\_cost

From Product

Join VendorProduct on VendorProduct.ProductID = Product.ProductID

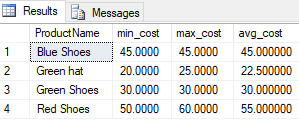
GROUP BY ProductName

1. Code and execute the SELECT statement to retrieve all values from this view. Copy and paste

your code to your lab document.

Select \* From ProductCosts

1. Paste a screenshot of the results of the query in b to your lab document.



4. Code and execute the SQL statement to create a view called ReorderProducts that shows all of

the products with a quantity on hand of less than 5. Copy and paste this code to your lab

document.

CREATE VIEW ReorderProducts As

SELECT

Product.ProductName

, Product.QtyOnHand

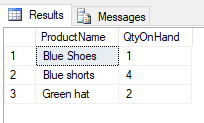
from Product

Where Product.QtyOnHand < 5

5. Code and execute the SELECT statement that retrieves all rows from this view. Copy and paste

this code into your lab document. Paste a screenshot of the results into your lab document.

select \* from ReorderProducts



6. Execute the following code against your database:

CREATE FUNCTION dbo.Profit (@ProductID int, @VendorID int, @SellPrice decimal(12,4))

RETURNS decimal (12,4)

AS

BEGIN

DECLARE @profit decimal (12,4)

SELECT @profit = @SellPrice - Cost FROM VendorProduct

WHERE ProductID = @ProductID and VendorID= @VendorID

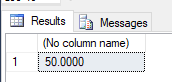
RETURN @profit

END

GO

SELECT dbo.Profit(1,1,100)

7. Paste a screenshot of the result of the SELECT statement to your lab document.



8. Code and execute the statement to create a function called PreferredVendor that accepts a

product id parameter as an input and returns the Vendor ID of the Vendor with the lowest cost

in VendorProduct (this may take some research). Copy and paste your code into your lab

document.

GO

ALTER FUNCTION dbo.PreferredVendor (@ProductID int)

RETURNS int

AS

BEGIN

RETURN

(

SELECT TOP 1 VendorID

FROM VendorProduct

WHERE ProductID = @ProductID

ORDER BY Cost

)

END

GO

SELECT dbo.PreferredVendor (6)

9. Code and execute the select statement that returns the ProductName, ProductDescription,

QtyOnHand, and the preferred vendorID (use your function). Copy and paste your code into

your lab document. Paste a screenshot of the results of this query into your lab document.

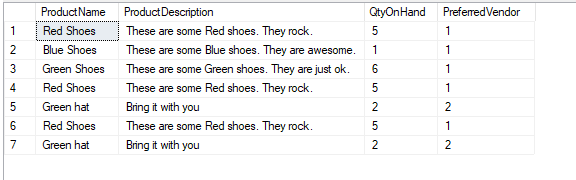
Select Product.ProductName

, Product.ProductDescription

, Product.QtyOnHand

, dbo.PreferredVendor(VendorProduct.ProductID) as PreferredVendor from VendorProduct

Join Product ON Product.ProductID = VendorProduct.ProductID



10. Execute the following code against your database:

GO

CREATE PROCEDURE AddProduct(@name char(30), @description varchar(255))

AS

BEGIN

DECLARE @ProductCount int

SELECT @ProductCount = Count(ProductID) From Product

Where ProductName= @name

If @ProductCount = 0

BEGIN

INSERT INTO Product (ProductName,ProductDescription)

VALUES (@name, @description)

END

RETURN @@IDENTITY

END

11. What does this procedure do (be descriptive). Answer in your lab document.

The Procedure searches the product table to find the count of product ids. If the count for a product is equal to 0, the sproc adds the product name and product description values specified into the table. The first part is basically running the if exists function on the table to make sure we aren’t replacing values that are there.

12. EXEC AddProduct(‘Red shirt’, ‘This is a red shirt’) will not add a row to the

database. Why? Answer in your lab document.

The syntax is different when running a sproc, as compared to a function. The following will add a row to the table:

EXEC AddProduct 'Red Shirt', "This is a red shirt"

13. Using AddProduct as a guide, code and execute a SQL command to create a stored procedure

called AddProductWithVendor that accepts a VendorID and a Cost (in addition to the

parameters in AddProduct) and not only adds the product, but also adds the VendorProduct

record with the appropriate. Hint: have a look at what @@identity does). Copy and paste your

sql code into your lab document.

Delete from Product where ProductName = 'Black hat'

GO

Alter PROCEDURE AddProductWithVendor(@name char(30), @description varchar(255), @vendorid int, @cost money)

AS

BEGIN

DECLARE @ProductCount int

SELECT @ProductCount = Count(ProductID) From Product

Where ProductName= @name

If @ProductCount = 0

BEGIN

INSERT INTO Product (ProductName,ProductDescription)

VALUES (@name, @description)

Insert INTO VendorProduct(ProductID, VendorID,Cost)

Values (@@identity, @vendorid,@cost)

END

RETURN @@IDENTITY

END

14. Code and execute the command to EXEC your stored procedure with the following values as

parameters. Copy and paste your code into your lab document.

a. Product Name: ‘Black hat’

b. Product Description: ‘This is a black hat’

c. Vendor ID: 1

d. Cost: 15

EXEC AddProductWithVendor 'Black hat', 'This is a black hat', '1', '15'

15. Select all values from your ProductCosts view and paste the results into your lab document

(make sure your results include a black hat).

