Programming and Stored Procedure Problem Set

Use the AdventureWorks2014 database for these questions.

1. Write a script that saves the count of rows in the SalesOrderHeader table in a variable. Then set up an IF block that prints “The count of sales is over 50,000” or “The count of sales is under 50,000” based on the count.

2. Write a loop that runs 100 times. If the counter is divisible by 2 (look up the modulo function %) then print “The value is even”. If it is odd, print “The value is odd.”

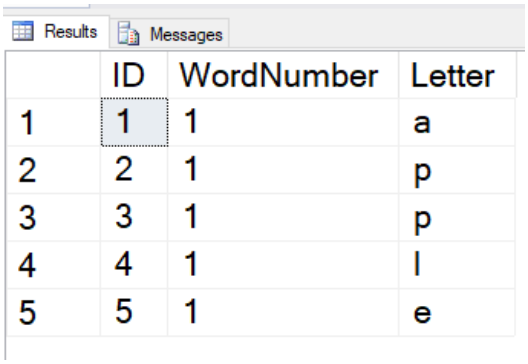
3. Write a script that contains nested IF blocks. The outer block should check to see whether the month is May or June. If it’s one of those months, then print “The month is “ plus the month. The inner block should check to see if the year is even. Print that information.

4. Save the production.product table into a temp table. Set up a loop to delete one row at a time.

5. Create a table called WordToLetters that has an identity column called ID, an Integer column called WordNumber and a CHAR(1) column called Letter. Create a stored procedure called SplitWord that takes a parameter called @Word varchar(50). This stored procedure should enter one row in the table for each letter in the word. Be sure to review functions that can help you write this proc. Here is an example:

   Exec dbo.SplitWord @word = ‘apple’;

   SELECT * FROM WordToLetters;



	ID	WordNumber	Letter
1	1	1	a
2	2	1	p
3	3	1	p
4	4	1	l
5	5	1	e

Programming and Stored Procedure Problem Set Answers

Use the AdventureWorks2014 database for these questions. There is not one right answer to these problems. Here are suggested solutions.

1. Write a script that saves the count of rows in the SalesOrderHeader table in a variable. Then set up an IF block that prints “The count of sales is over 50,000” or “The count of sales is less than or equal to 50,000” based on the count.

```
DECLARE @RowCount INT;

SELECT @RowCount = COUNT(*)
FROM Sales.SalesOrderHeader;

IF @RowCount > 50000 BEGIN
    PRINT 'The count of sales is over 50,000.';
END
ELSE BEGIN
    PRINT 'The count of sales is less than or equal to 50,000';
END;
```

2. Write a loop that runs 100 times. Print the current loop count. If the counter is divisible by 2 (look up the modulo function %) then print “The value is even”. If it is odd, print “The value is odd.”

```
DECLARE @Count INT = 1;

WHILE @Count <= 100 BEGIN
        PRINT @Count
        IF @Count % 2 = 0 BEGIN
                PRINT 'The value is even.';
        END
        ELSE BEGIN
                PRINT 'The value is odd.';
        END
        SET @Count += 1;
END;
```

3. Write a script that contains nested IF blocks. The outer block should check to see whether the month is May or June. If it's one of those months, then print "The month is " plus the month. The inner block should check to see if the year is even. Print that information.

```
IF DATENAME(month,GETDATE()) IN ('June','May') BEGIN
	PRINT 'The month is ' + DATENAME(month,GETDATE());
	IF YEAR(GETDATE()) % 2 = 0 BEGIN
		PRINT 'The year is even';
	END
	ELSE BEGIN
		PRINT 'The year is odd';
	END
END;
```

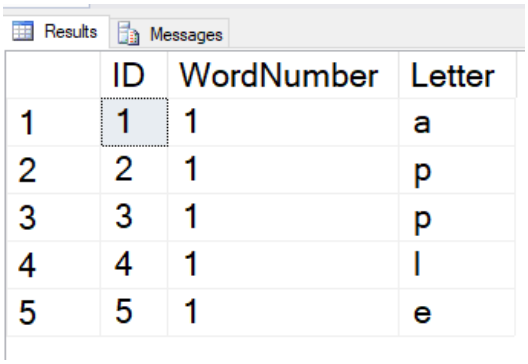
4. Save the production.product table into a temp table. Set up a loop to delete one row at a time. Print the name of the product that is being deleted.

```
SELECT *
INTO #Products
FROM Production.Product;

DECLARE @ProductName NVARCHAR(50);
WHILE EXISTS(SELECT * FROM #Products) BEGIN
	SELECT @ProductName = Name
	FROM #Products;

	PRINT @ProductName;

	DELETE FROM #Products
	WHERE Name = @ProductName;
END;
```

5. Create a table called WordToLetters that has an identity column called ID, an Integer column called WordNumber and a CHAR(1) column called Letter. Create a stored procedure called SplitWord that takes a parameter called @Word varchar(50). This stored procedure should enter one row in the table for each letter in the word. Be sure to review functions that can help you write this proc. Here is an example:

Exec dbo.SplitWord @word = 'apple';

SELECT * FROM WordToLetters;



Results Messages

	ID	WordNumber	Letter
1	1	1	a
2	2	1	p
3	3	1	p
4	4	1	l
5	5	1	e

```
DROP TABLE IF EXISTS WordToLetter;
GO
CREATE TABLE WordToLetter (
        ID INT NOT NULL IDENTITY,
        WordNumber INT NOT NULL,
        Letter CHAR(1));
GO
DROP PROC IF EXISTS dbo.SplitWord;
GO
CREATE PROC dbo.SplitWord @Word VARCHAR(50) AS

        DECLARE @WordNumber INT;
        DECLARE @Count INT = 1;
        DECLARE @Letter CHAR(1);

        --Find the highest word number and add 1
        SELECT @WordNumber = MAX(WordNumber) FROM WordToLetter;
        SET @WordNumber = ISNULL(@WordNumber,0) + 1;

        --Use a loop to look at one letter at a time
        WHILE @Count <= LEN(@Word) BEGIN
                SET @Letter = SUBSTRING(@Word,@Count,1);
                INSERT INTO WordToLetter(WordNumber,Letter)
```

```
                VALUES(@WordNumber, @Letter);
                SET @Count += 1;
        END;
GO

EXEC dbo.SplitWord @Word = 'apple';

SELECT * FROM WordToLetter
WHERE WordNumber = 1
ORDER BY ID;

EXEC dbo.SplitWord @Word = 'grapes';


SELECT * FROM WordToLetter
WHERE WordNumber = 2
ORDER BY ID;
```