**Problem #1 – TSQL – Basic queries**

/\* Use the temp tables below and any other SQL capabilities to

1) Display a list of Customers that did not order anything

2) Display a list of Orders that do not have a valid customer association

3) Display a list of Orders with product info and total cost

4) Display a list of Products that were not ordered

\*/

SET NOCOUNT ON;

DECLARE @customer TABLE (customerID int, customerName varchar(50))

DECLARE @orders TABLE (orderID int,customerID int, productID int, orderDate datetime, qty float)

DECLARE @product TABLE (productID int, productName varchar(250), pricePer float)

INSERT @customer VALUES(1,'Superman')

INSERT @customer VALUES(2,'Batman')

INSERT @customer VALUES(3,'Flash')

INSERT @customer VALUES(4,'Hulk')

INSERT @customer VALUES(5,'Thor')

INSERT INTO @orders VALUES(1,1,1,'10/2/2009',25)

INSERT INTO @orders VALUES(2,2,2,'10/15/2009',5)

INSERT INTO @orders VALUES(3,1,1,'10/25/2009',35)

INSERT INTO @orders VALUES(4,2,3,'10/16/2009',10)

INSERT INTO @orders VALUES(5,5,1,'10/2/2009',2)

INSERT INTO @orders VALUES(6,5,1,'10/3/2009',1)

INSERT INTO @orders VALUES(6,6,1,'10/3/2009',3)

INSERT INTO @orders VALUES(6,6,1,'10/3/2009',1)

INSERT INTO @product VALUES(1,'Glasses',250.00)

INSERT INTO @product VALUES(2,'Belt',850.00)

INSERT INTO @product VALUES(3,'Hook',250.00)

INSERT INTO @product VALUES(4,'Candy',50.00)

INSERT INTO @product VALUES(5,'Hotel',12500050.00)

INSERT INTO @product VALUES(6,'Food',50.00)

SELECT \*

FROM @customer a

SELECT \*

FROM @orders a

SELECT \*

FROM @product a

Answer SQL Script:

--1) Display a list of Customers that did not order anything

SELECT \*

FROM @customer a WHERE customerID NOT IN (SELECT customerID FROM @orders )

--2) Display a list of Orders that do not have a valid customer association

SELECT \*

FROM @orders a WHERE customerID NOT IN (SELECT customerID FROM @customer )

--3) Display a list of Orders with product info and total cost

SELECT a.\*, p.\*, a.qty\*, p.pricePer AS 'total cost'

FROM @orders a

INNER JOIN @product p ON a.productID = p.productID

--4) Display a list of Products that were not ordered

SELECT \*

FROM @product a WHERE a.productID NOT IN (SELECT productID FROM @orders )

**Problem #2 – TSQL - conditional display**

Code:

/\*Pulling data based on conditions and display warnings

You have a master table with date sensitive records.

You have 3 other child tables with associated records and volumes.

The three child tables house the same information but T1 is considered more accurate than T2 and it is considered more accurate than T3.

These tables may or may not have information for a given day.

Your task:

1. To pull all information from the master table

2. Pull the most accurate volume in as a column (vol)

3. Create a warning flag for all three child tables if the volume is over 50 or below -50

4. Do this with the minimal amount of code

5. No loops

6. No sub queries

7. No temp tables

\*/

SET NOCOUNT ON;

DECLARE @tbl\_mstr TABLE (id int, nm varchar(50), dt datetime)

DECLARE @tbl\_1 TABLE (id int, dt datetime, vol float)

DECLARE @tbl\_2 TABLE (id int, dt datetime, vol float)

DECLARE @tbl\_3 TABLE (id int, dt datetime, vol float)

INSERT @tbl\_mstr VALUES(1,'Helga','10/1/2009')

INSERT @tbl\_mstr VALUES(1,'Helga','10/2/2009')

INSERT @tbl\_mstr VALUES(1,'Helga','10/3/2009')

INSERT @tbl\_mstr VALUES(1,'Helga','10/4/2009')

INSERT @tbl\_mstr VALUES(1,'Helga','10/5/2009')

INSERT @tbl\_mstr VALUES(1,'Helga','10/6/2009')

INSERT @tbl\_mstr VALUES(1,'Helga','10/7/2009')

INSERT @tbl\_mstr VALUES(1,'Helga','10/8/2009')

INSERT @tbl\_mstr VALUES(1,'Helga','10/9/2009')

INSERT @tbl\_mstr VALUES(1,'Helga','10/10/2009')

INSERT INTO @tbl\_1 VALUES(1,'10/2/2009',25)

INSERT INTO @tbl\_1 VALUES(1,'10/8/2009',42)

INSERT INTO @tbl\_1 VALUES(1,'10/9/2009',38)

INSERT INTO @tbl\_2 VALUES(1,'10/1/2009',-55)

INSERT INTO @tbl\_2 VALUES(1,'10/3/2009',69)

INSERT INTO @tbl\_2 VALUES(1,'10/8/2009',74)

INSERT INTO @tbl\_2 VALUES(1,'10/10/2009',16)

INSERT INTO @tbl\_3 VALUES(1,'10/1/2009',08)

INSERT INTO @tbl\_3 VALUES(1,'10/4/2009',37)

INSERT INTO @tbl\_3 VALUES(1,'10/5/2009',75)

INSERT INTO @tbl\_3 VALUES(1,'10/6/2009',-22)

INSERT INTO @tbl\_3 VALUES(1,'10/7/2009',-64)

Answer SQL Script:

SELECT tm.\*,

--Use with Azure SQL

--LEAST(t1.vol, t2.vol, t3.vol) as ValMin,

--GREATEST(t1.vol, t2.vol, t3.vol) as ValMax,

(select max(i) from (values (t1.vol), (t2.vol), (t3.vol)) AS T(i)) [ValMax],

(select min(i) from (values (t1.vol), (t2.vol), (t3.vol)) AS T(i)) [ValMin],

(CASE

WHEN t1.vol > 50

OR t1.vol < -50 THEN

'T1 Wrng'

ELSE

'OK'

END

) AS 'T1 Flag',

(CASE

WHEN t2.vol > 50

OR t2.vol < -50 THEN

'T2 Wrng'

ELSE

'OK'

END

) AS 'T2 Flag',

(CASE

WHEN t3.vol > 50

OR t3.vol < -50 THEN

'T3 Wrng'

ELSE

'OK'

END

) AS 'T3 Flag'

FROM @tbl\_mstr tm

LEFT JOIN @tbl\_1 t1

ON t1.id = tm.id

AND tm.dt = t1.dt

LEFT JOIN @tbl\_2 t2

ON t2.id = tm.id

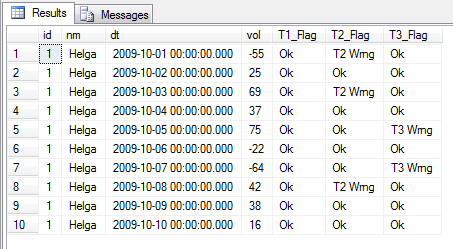
AND tm.dt = t2.dt

LEFT JOIN @tbl\_3 t3

ON t3.id = tm.id

AND tm.dt = t3.dt

Results:



**Problem #3 – TSQL - summarization and capitalization**

Code:

/\*Pulling data and display with appropriate format and show summarized data

You have a simple table of data.

No additional table variables or temp tables should be used

Your task in your sql output:

3. Capitalize the state names

2. Convert the county names to camel (meaning first letter is capitalized)

3. Show the city data along with county and state totals

4. Sort by state, county, city, then totals

\*/

SET NOCOUNT ON;

/\*play data\*/

--\*\*--\*\*--

DECLARE @tbl TABLE (State varchar(2), County varchar(40), City varchar(40), WellCount int)

INSERT @tbl VALUES ('ok','la flore','Mcalister',5)

INSERT @tbl VALUES ('ok','la flore','Savannah',2)

INSERT @tbl VALUES ('ok','hughes','Dustin',9)

INSERT @tbl VALUES ('tx','tarrant','Fort Worth',51)

INSERT @tbl VALUES ('tx','tarrant','Burleson',6)

INSERT @tbl VALUES ('tx','parKer','Weatherford',7)

INSERT @tbl VALUES ('ar','bryaNnt','Little Rock',12)

INSERT @tbl VALUES ('ar','bryaNnt','Ozark',12)

INSERT @tbl VALUES ('ar','reeD','Van Buren',46)

INSERT @tbl VALUES ('nm','saN Jaun','Farmington',3)

INSERT @tbl VALUES ('nm','saN Jaun',' Bloomfield',3)

INSERT @tbl VALUES ('nm','rio arriba','Durango',104)

--\*\*--\*\*--

Answer SQL Script:

SELECT UPPER( CASE

WHEN [State] IS NULL THEN

'Total:'

ELSE

[State]

END

) As [State],

(

SELECT STRING\_AGG(UPPER(left(value, 1)) + SUBSTRING(value, 2, 999), ' ')

from STRING\_SPLIT(LOWER( CASE

WHEN County IS NULL

AND [State] IS NOT NULL THEN

'State Total:'

ELSE

County

End

), ' ')

) AS County,

(

SELECT STRING\_AGG(UPPER(left(value, 1)) + SUBSTRING(value, 2, 999), ' ')

from STRING\_SPLIT(LOWER( CASE

WHEN City IS NULL

AND County IS NOT NULL

AND [State] IS NOT NULL THEN

'Country Total:'

ELSE

City

End

), ' ')

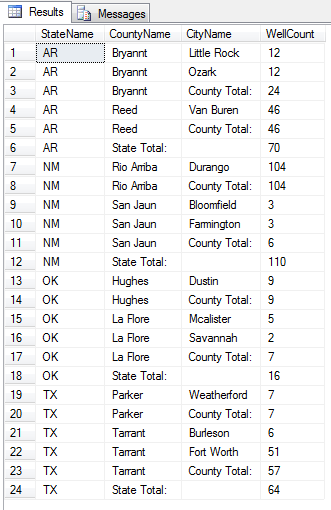
) AS City,

SUM(WellCount) AS WellCount

FROM @tbl

GROUP BY ROLLUP ([State], County, City)

Results:



**Problem #4 – TSQL - looping and manipulating tables**

Code:

/\*Looping and manipulating tables

Have auto commit set in your SQL Server Management Studio settings

You have 1 simple table of data.

Based on the date range given, display the data in a horizontal format for each unload event

Looping is recommended, and 1 temp table is recommended. You may also need another temp or table variable.

Working variables are also recommended.

Your task in your sql output:

1. Catch all unload events for a date range

2. Sequence the time events in numerical order incrementing by 1

3. Display the sequence in one column

4. Have another column for each unload serial number - start date and display all unload density values per sequence

5. The column creation for the serial numbers should be dynamic requiring looping

6. Create the script so that you can feed it a start date and end date

\*/

SET NOCOUNT ON;

/\*declare initial data holding table\*/

CREATE TABLE ##data (recorddatetime datetime, unload\_density float, loadserialnumber int);

/\*load initial data to work with\*/

INSERT INTO ##data VALUES('2010-10-09 13:26:36.000',1.01059,252);

INSERT INTO ##data VALUES('2010-10-09 13:26:41.000',1.01068,252);

INSERT INTO ##data VALUES('2010-10-09 13:26:46.000',1.02589,252);

INSERT INTO ##data VALUES('2010-10-10 08:15:02.000',1.03333,266);

INSERT INTO ##data VALUES('2010-10-10 08:15:07.000',1.04444,266);

INSERT INTO ##data VALUES('2010-10-10 08:15:12.000',1.01561,266);

INSERT INTO ##data VALUES('2010-10-10 08:15:17.000',1.03581,266);

INSERT INTO ##data VALUES('2010-10-10 08:15:22.000',1.03554,266);

INSERT INTO ##data VALUES('2010-10-08 18:47:45.000',1.04983,302);

INSERT INTO ##data VALUES('2010-10-08 18:47:50.000',1.09813,302);

INSERT INTO ##data VALUES('2010-10-08 18:47:56.000',1.04576,302);

INSERT INTO ##data VALUES('2010-10-08 18:48:01.000',1.03485,302);

Answer SQL Script:

CREATE TABLE #MyData (Record INT IDENTITY(1,1));

DECLARE @ListDates TABLE(Record INT IDENTITY(1,1), RecordDate DateTime, RecordTime varchar(20), SerialNumber int)

INSERT INTO @ListDates

SELECT Distinct

CONVERT(date, recorddatetime) AS recorddate,

FIRST\_VALUE(CONVERT(VARCHAR(20), recorddatetime, 8)) OVER (PARTITION BY CONVERT(date, recorddatetime),

loadserialnumber

ORDER BY loadserialnumber ASC

),

loadserialnumber

FROM ##data

ORDER BY loadserialnumber

DECLARE @counter int = 1

DECLARE @Rowcount int = (SELECT COUNT(\*) FROM @ListDates)

DECLARE @ColumnName NVARCHAR(200) = ''

DECLARE @SelecColumnName NVARCHAR(200) = ''

DECLARE @TempColumnName NVARCHAR(200) = ''

DECLARE @serialnumber NVARCHAR(10) = ''

DECLARE @InnerColumnText NVARCHAR(MAX) = ''

WHILE @counter <= @Rowcount

BEGIN

IF @ColumnName <> '' BEGIN

SET @ColumnName += ', '

SET @SelecColumnName += ', '

END

SELECT @serialnumber = CAST(SerialNumber AS VARCHAR(20)),

@TempColumnName

= '[Load' + CAST(SerialNumber AS VARCHAR(20)) + '\_' + CONVERT(VARCHAR(20), RecordDate, 101) + '\_'

+ RecordTime + ']',

@SelecColumnName += @TempColumnName + '.unload\_density',

@ColumnName += @TempColumnName + 'float NULL'

FROM @ListDates

WHERE Record = @counter

SET @InnerColumnText += FORMATMESSAGE('LEFT JOIN (SELECT ROW\_NUMBER() OVER (ORDER BY recorddatetime) as rn, unload\_density

FROM ##data WHERE loadserialnumber=%s) %s ON %s.rn =d1.rn ', @serialnumber, @TempColumnName, @TempColumnName)

SET @counter += 1

END

SET @ColumnName = 'ALTER TABLE #MyData ADD ' + @ColumnName

Exec sp\_executesql @ColumnName

SET @InnerColumnText

= FORMATMESSAGE(

'INSERT INTO #MyData

SELECT %s FROM

(SELECT ROW\_NUMBER() OVER (ORDER BY recorddatetime) as rn

FROM ##data) d1 %s

WHERE COALESCE (%s ) IS NOT NULL; ',

@SelecColumnName,

@InnerColumnText,

@SelecColumnName

)

Exec sp\_executesql @InnerColumnText

SELECT \* FROM #MyData

DROP TABLE ##data

DROP TABLE #MyData

Independent:

--SELECT t1.unload\_density,

-- t2.unload\_density,

-- t3.unload\_density

--FROM

--(

-- SELECT ROW\_NUMBER() OVER (ORDER BY recorddatetime) as rn

-- FROM ##data

--) d1

-- LEFT JOIN

-- (

-- SELECT ROW\_NUMBER() OVER (ORDER BY recorddatetime) as rn,

-- unload\_density

-- FROM ##data

-- WHERE loadserialnumber = 252

-- ) t1

-- ON t1.rn = d1.rn

-- LEFT JOIN

-- (

-- SELECT ROW\_NUMBER() OVER (ORDER BY recorddatetime) as rn,

-- unload\_density

-- FROM ##data

-- WHERE loadserialnumber = 266

-- ) t2

-- ON t2.rn = d1.rn

-- LEFT JOIN

-- (

-- SELECT ROW\_NUMBER() OVER (ORDER BY recorddatetime) as rn,

-- unload\_density

-- FROM ##data

-- WHERE loadserialnumber = 302

-- ) t3

-- ON t3.rn = d1.rn

--WHERE COALESCE(t1.unload\_density, t2.unload\_density, t3.unload\_density) IS NOT NULL;

Results:

