T-SQL Scripting

```sql
-- Scalar Variables
USE AccountPayables;
GO

DECLARE @MaxInvoice money, @MinInvoice money;
DECLARE @PercentDifference decimal(8,2);
DECLARE @InvoiceCount int, @VendorIDVar int;

SET @VendorIDVar = 95;
SET @MaxInvoice = (SELECT MAX(InvoiceTotal) FROM Invoices
    WHERE VendorID = @VendorIDVar);
SELECT @MinInvoice = MIN(InvoiceTotal), @InvoiceCount = COUNT(*)
FROM Invoices
WHERE VendorID = @VendorIDVar;
SET @PercentDifference = (@MaxInvoice - @MinInvoice) / @MinInvoice * 100;

PRINT 'Maximum invoice is $' + CONVERT(varchar,@MaxInvoice,1) + '.';
PRINT 'Minimum invoice is $' + CONVERT(varchar,@MinInvoice,1) + '.';
PRINT 'Maximum is ' + CONVERT(varchar,@PercentDifference) +
    '% more than minimum.';
PRINT 'Number of invoices: ' + CONVERT(varchar,@InvoiceCount) + '.';

GO


-- Table Variables
USE AccountPayables;

DECLARE @BigVendors table
(VendorID int,
VendorName varchar(50));

INSERT @BigVendors
SELECT VendorID, VendorName
FROM Vendors
WHERE VendorID IN (SELECT VendorID FROM Invoices WHERE InvoiceTotal > 5000);

SELECT * FROM @BigVendors;
GO

--Local Temporary Tables
USE AccountPayables;

SELECT TOP 1 VendorID, AVG(InvoiceTotal) AS AvgInvoice
INTO #TopVendors
FROM Invoices
GROUP BY VendorID
ORDER BY AvgInvoice DESC;

SELECT Invoices.VendorID, MAX(InvoiceDate) AS LatestInv
FROM Invoices JOIN #TopVendors
    ON Invoices.VendorID = #TopVendors.VendorID
GROUP BY Invoices.VendorID;
GO

--Global Temporary Tables
```

```sql
CREATE TABLE ##RandomSSNs
(SSN_ID int IDENTITY,
SSN char(9) DEFAULT
    LEFT(CAST(CAST(CEILING(RAND()*10000000000)AS bigint)AS varchar),9));

INSERT ##RandomSSNs VALUES (DEFAULT);
INSERT ##RandomSSNs VALUES (DEFAULT);

SELECT * FROM ##RandomSSNs;
GO

-- IF ... ELSE
USE AccountPayables;
DECLARE @MinInvoiceDue money, @MaxInvoiceDue money;
DECLARE @EarliestInvoiceDue smalldatetime, @LatestInvoiceDue smalldatetime;
SELECT @MinInvoiceDue = MIN(InvoiceTotal - PaymentTotal - CreditTotal),
    @MaxInvoiceDue = MAX(InvoiceTotal - PaymentTotal - CreditTotal),
    @EarliestInvoiceDue = MIN(InvoiceDueDate),
    @LatestInvoiceDue = MAX(InvoiceDueDate)
FROM Invoices
WHERE InvoiceTotal - PaymentTotal - CreditTotal > 0;
IF @EarliestInvoiceDue < GETDATE()
    BEGIN
        PRINT 'Outstanding invoices overdue!';
        PRINT 'Dated ' + CONVERT(varchar,@EarliestInvoiceDue,1) +
            ' through ' + CONVERT(varchar,@LatestInvoiceDue,1) + '.';
        PRINT 'Amounting from $' + CONVERT(varchar,@MinInvoiceDue,1) +
            ' to $' + CONVERT(varchar,@MaxInvoiceDue,1) + '.';
    END;
ELSE --@EarliestInvoiceDue >= GETDATE()
    PRINT 'No overdue invoices.';
GO

-- WHILE
USE AccountPayables;
IF OBJECT_ID('tempdb..#InvoiceCopy') IS NOT NULL
    DROP TABLE #InvoiceCopy;

SELECT * INTO #InvoiceCopy FROM Invoices
WHERE InvoiceTotal - CreditTotal - PaymentTotal > 0;

WHILE (SELECT SUM(InvoiceTotal - CreditTotal - PaymentTotal)
        FROM #InvoiceCopy) >= 20000
    BEGIN
        UPDATE #InvoiceCopy
        SET CreditTotal = CreditTotal + .05
        WHERE InvoiceTotal - CreditTotal - PaymentTotal > 0;

        IF (SELECT MAX(CreditTotal) FROM #InvoiceCopy) > 3000
            BREAK;
        ELSE --(SELECT MAX(CreditTotal) FROM #InvoiceCopy) <= 3000
            CONTINUE;
    END;

SELECT InvoiceDate, InvoiceTotal, CreditTotal
FROM #InvoiceCopy;
GO
-- Cursors
```

```
USE AccountPayables;

DECLARE @InvoiceIDVar int, @InvoiceTotalVar money, @UpdateCount int;
SET @UpdateCount = 0;

DECLARE Invoices_Cursor CURSOR
FOR
    SELECT InvoiceID, InvoiceTotal  FROM Invoices
    WHERE InvoiceTotal - PaymentTotal - CreditTotal > 0;

OPEN Invoices_Cursor;

FETCH NEXT FROM Invoices_Cursor INTO @InvoiceIDVar, @InvoiceTotalVar;
WHILE @@FETCH_STATUS <> -1
    BEGIN
        IF @InvoiceTotalVar > 1000
        BEGIN
            UPDATE Invoices
            SET CreditTotal = CreditTotal + (InvoiceTotal * .1)
            WHERE InvoiceID = @InvoiceIDVar;

            SET @UpdateCount = @UpdateCount + 1;
        END;
        FETCH NEXT FROM Invoices_Cursor INTO @InvoiceIDVar, @InvoiceTotalVar;
    END;

CLOSE Invoices_Cursor;
DEALLOCATE Invoices_Cursor;

PRINT '';
PRINT CONVERT(varchar, @UpdateCount) + ' row(s) updated.';

GO

--Error handling
USE AccountPayables;
BEGIN TRY
    INSERT Invoices
    VALUES (799, 'ZXK-799', '2016-05-07', 299.95, 0, 0,
            1, '2016-06-06', NULL);
    PRINT 'SUCCESS: Record was inserted.';
END TRY
BEGIN CATCH
    PRINT 'FAILURE: Record was not inserted.';
    PRINT 'Error ' + CONVERT(varchar, ERROR_NUMBER(), 1)
        + ': ' + ERROR_MESSAGE();
END CATCH;

GO

--@@IDENTITY
USE AccountPayables;
DECLARE @MyIdentity int, @MyRowCount int;

INSERT Vendors (VendorName, VendorAddress1, VendorCity, VendorState,
    VendorZipCode, VendorPhone, DefaultTermsID, DefaultAccountNo)
VALUES ('Peerless Binding', '1112 S Windsor St', 'Hallowell', 'ME',
    '04347', '(207) 555-1555', 4, 400);
```

```
SET @MyIdentity = @@IDENTITY;
SET @MyRowCount = @@ROWCOUNT;

IF @MyRowCount = 1
    INSERT Invoices
    VALUES (@MyIdentity, 'BA-0199', '2012-05-01', 4598.23,
        0, 0, 4, '2016-06-30', NULL);
```