**Advanced SQL – Week 11**

* The **USE** statement sets the current database
* You must **DECLARE** variables before you use them, you can define one or more at the same time.

DECLARE @variablename variabletype = value

SET @TotalCost = 10

SET @TotalCost = @UnitCost \* 1.1

DECLARE @Test money

SET @Test = (SELECT MAX(Sal) FROM Emp)

SELECT @Test

PRINT 'The maximum salary is ' +CONVERT(varchar,@Test,1)

go

* There are a number of System Variables such as: @@ERROR, @@IDENTITY, @@ROWCOUNT

* To separate a script into multiple batches, we use the **GO** statement. Using it causes all statements ince the beginning of the script or the last GO statement (whichever is closer) to be compiled into one execution plan and sent to the server independently of any other batches.

* Variables declared in one batch are not available to other batches

EXAMPLES

SELECT convert(varchar,table\_catalog,20) as "Database Name", table\_name

FROM information\_schema.tables

create table Billings (

BankerID INTEGER,

BillingNumber INTEGER,

BillingDate datetime,

BillingTotal INTEGER,

TermsID INTEGER,

BillingDueDate datetime ,

PaymentTotal INTEGER,

CreditTotal INTEGER);

GO

INSERT INTO Billings VALUES (5, 1, '2005-01-22', 110, 1,'2005-04-22',123,321);

GO

INSERT INTO Billings VALUES (2, 2, '2001-02-21', 165, 1,'2002-02-22',123,321.);

GO

INSERT INTO Billings VALUES (5, 3, '2003-05-02', 190, 1,'2005-04-12',123,321);

GO

INSERT INTO Billings VALUES (4, 4, '1999-03-12', 165, 1,'2005-04-18',123,321);

GO

INSERT INTO Billings VALUES (5, 5, '2000-04-23', 165, 1,'2005-04-17',123,321);

GO

INSERT INTO Billings VALUES (6, 6, '2001-06-14', 165, 1,'2005-04-18',123,321);

GO

INSERT INTO Billings VALUES (5, 7, '2002-07-15', 120, 1,'2005-04-19',123,321);

GO

INSERT INTO Billings VALUES (8, 8, '2003-08-16', 165, 1,'2005-04-20',123,321);

GO

INSERT INTO Billings VALUES (9, 9, '2004-09-17', 165, 1,'2005-04-21',123,321);

GO

INSERT INTO Billings VALUES (5, 0, '2005-10-18', 145, 1,'2005-04-22',123,321);

GO

DECLARE @MaxBilling money, @MinBilling money

DECLARE @PercentDifference decimal(8,2)

DECLARE @BillingCount int, @BankerIDVar int

SET @BankerIDVar = 5

SET @MaxBilling = (SELECT MAX(BillingTotal) FROM Billings

WHERE BankerID = @BankerIDVar)

SELECT @MinBilling = MIN(BillingTotal), @BillingCount = COUNT(\*)

FROM Billings

WHERE BankerID = @BankerIDVar

SET @PercentDifference = (@MaxBilling - @MinBilling) / @MinBilling \* 100

PRINT 'Maximum Billing is $' + CONVERT(varchar,@MaxBilling,1) + '.'

PRINT 'Minimum Billing is $' + CONVERT(varchar,@MinBilling,1) + '.'

PRINT 'Maximum is ' + CONVERT(varchar,@PercentDifference) +

'% more than minimum.'

PRINT 'Number of Billings: ' + CONVERT(varchar,@BillingCount) + '.'

* Control of Flow Statements: IF … ELSE, CASE, WHILE

IF <Boolean Expression>

<SQL statement> | BEGIN <code series> END

[ELSE

<SQL statement> | BEGIN <code series> END]

Variable = CASE <input expression>

WHEN <when expression> THEN <result expression>

[…n]

[ELSE <result expression>]

END

WHILE <Boolean expression>

<sql statement> |

[BEGIN

<statement block>

[BREAK]

<sql statement> | <statement block>

[CONTINUE]

**Transact-SQL Variables** <http://msdn.microsoft.com/en-us/library/ms187953.aspx>

**T-SQL Loop Statement** <http://www.databasejournal.com/features/mssql/article.php/3100621/T-SQL-Programming-Part-2---Building-a-T-SQL-Loop.htm>

**Looping Statements** <http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp?topic=/com.ibm.db2.udb.apdv.sql.doc/doc/c0024350.htm>