-- Listing 3.18: An unparameterized query being sent to SQL Server.

cmd.CommandType = CommandType.Text;

cmd.CommandText = @"SELECT soh.SalesOrderNumber,

sod.ProductID

FROM Sales.SalesOrderHeader AS soh

INNER JOIN Sales.SalesOrderDetail AS sod

ON soh.SalesOrderID = sod.SalesOrderID

WHERE soh.SalesOrderNumber = '" + txtSalesOrderNo.Text + "'";

dtrSalesOrders = cmd.ExecuteReader();int retries = 4;

-- Listing 3.19: A parameterized query being sent to SQL Server.

dtrSalesOrders.Close();

cmd.CommandType = CommandType.Text;

cmd.CommandText = @"SELECT soh.SalesOrderNumber,

sod.ProductID

FROM Sales.SalesOrderHeader AS soh

INNER JOIN Sales.SalesOrderDetail AS sod

ON soh.SalesOrderID = sod.SalesOrderID

WHERE soh.SalesOrderNumber = @SalesOrderNo";

cmd.Parameters.Add("@SalesOrderNo", SqlDbType.NVarChar, 50);

cmd.Parameters["@SalesOrderNo"].Value = txtSalesOrderNo.Text;

dtrSalesOrders = cmd.ExecuteReader();

--Listing 7.17: TRY…CATCH handling of deadlock exceptions, in C#.

while (retries > 0)

{

try

{

// place sql code here

retries = 0;

}

catch (SqlException exception)

{

// exception is a deadlock

if (exception.Number == 1205)

{

// Delay processing to allow retry.

Thread.Sleep(500);

retries --;

}

// exception is not a deadlock

else

{

throw;

}

}

}