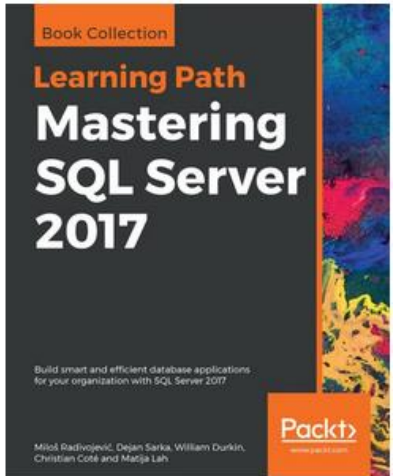
TRANSACT-SQL TIPS

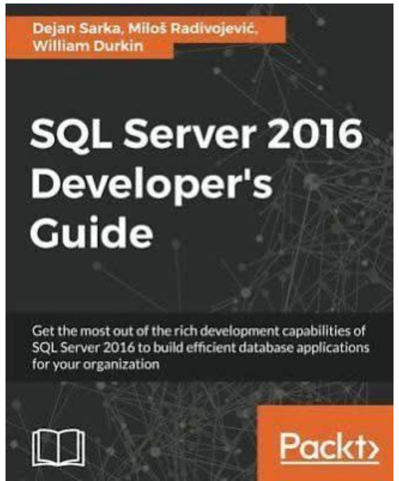
AUCKLAND SQL USER GROUP, APR 27TH 2021

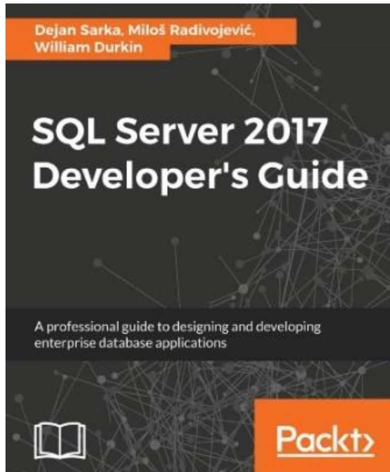
MILOŠ RADIVOJEVIĆ, MICROSOFT DATA PLATFORM MVP

ABOUT ME

- Data Platform MVP
- Principal Database Consultant at bwin, Vienna, Austria
- Co-Founder: SQL Pass Austria
- Conference Speaker, Book Author
- E: milos.radivojevic@chello.at
- W: https://milossql.wordpress.com


Book Collection
Learning Path
Mastering SQL Server 2017
Build smart and efficient database applications for your organization with SQL Server 2017
Packt
Miloš Radivojević, Dejan Sarka, William Durkin, Christian Coté and Matija Lah


Dejan Sarka, Miloš Radivojević, William Durkin
SQL Server 2016 Developer's Guide
Get the most out of the rich development capabilities of SQL Server 2016 to build efficient database applications for your organization
Packt


Dejan Sarka, Miloš Radivojević, William Durkin
SQL Server 2017 Developer's Guide
A professional guide to designing and developing enterprise database applications
Packt

AGENDA

Functions and Arithmetic Operators in the WHERE Clause

Local Variables and Performance

Data Type Conversions

Tips for Dealing with the OR Statement

ForEach as Performance Killer

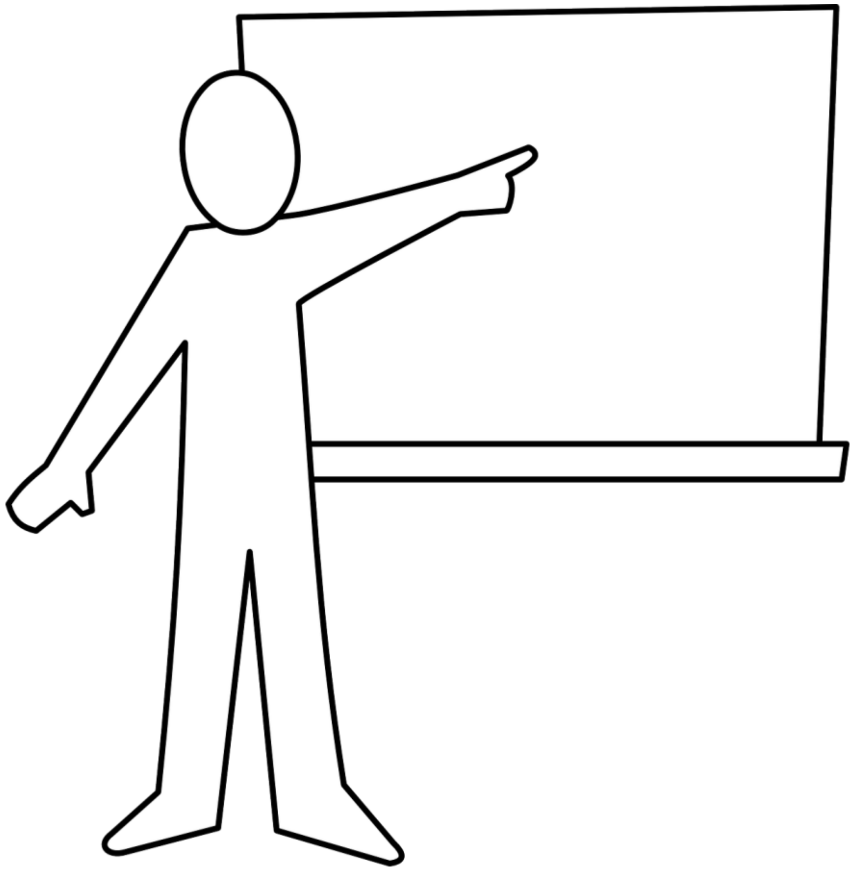
Window Functions vs. Apply

Database Constrains and Performance

Query and Table Hints (when to stop tuning)

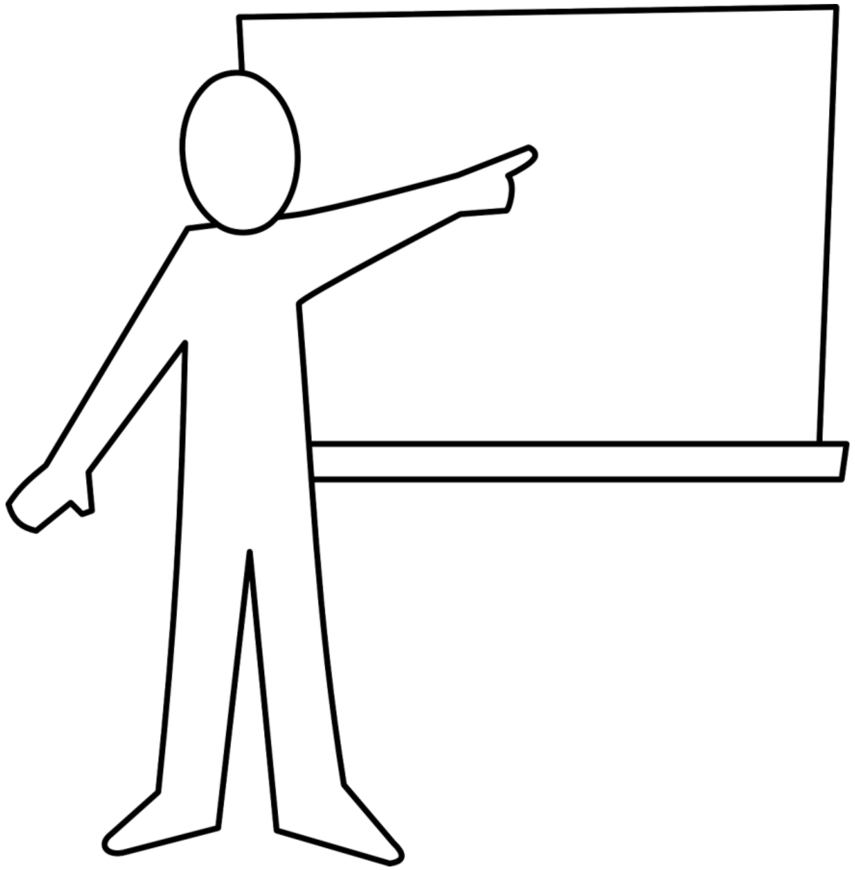
DEMO

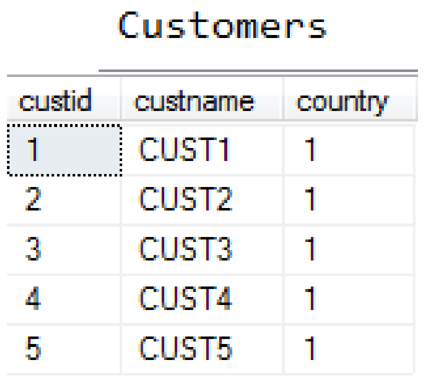
- FOR EACH LOOP – TOP PERFORMANCE KILLER



DEMO

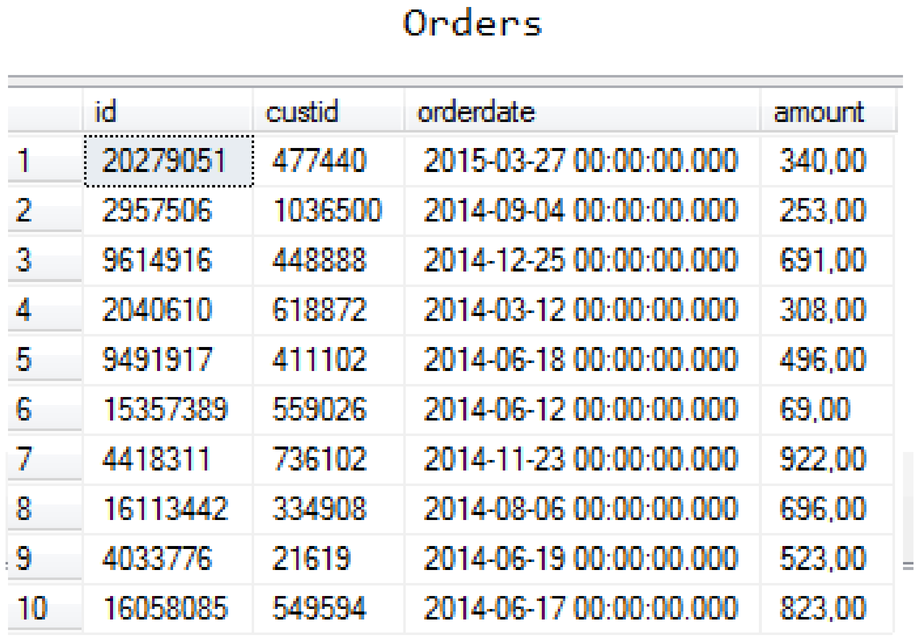
- RETURN TWO MOST RECENT ORDERS FOR EACH CUSTOMER IN THE YEAR 2019 IF THE AMOUNT IS AT LEAST 1000 €





Customers

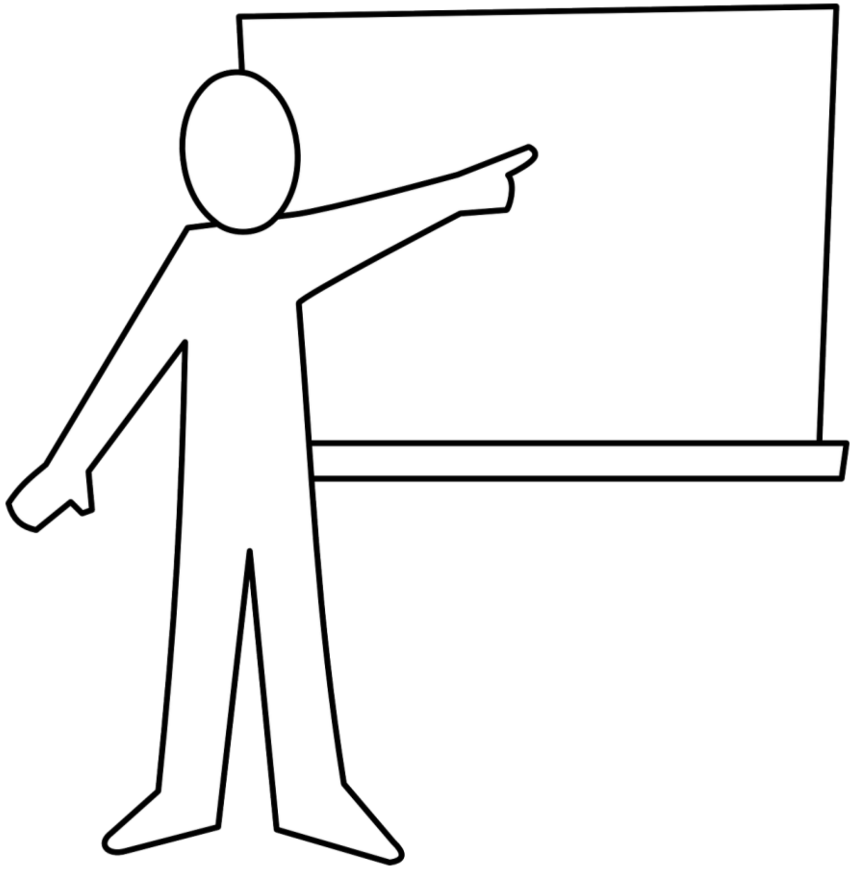
custid	custname	country
1	CUST1	1
2	CUST2	1
3	CUST3	1
4	CUST4	1
5	CUST5	1



Orders

	id	custid	orderdate	amount
1	20279051	477440	2015-03-27 00:00:00.000	340,00
2	2957506	1036500	2014-09-04 00:00:00.000	253,00
3	9614916	448888	2014-12-25 00:00:00.000	691,00
4	2040610	618872	2014-03-12 00:00:00.000	308,00
5	9491917	411102	2014-06-18 00:00:00.000	496,00
6	15357389	559026	2014-06-12 00:00:00.000	69,00
7	4418311	736102	2014-11-23 00:00:00.000	922,00
8	16113442	334908	2014-08-06 00:00:00.000	696,00
9	4033776	21619	2014-06-19 00:00:00.000	523,00
10	16058085	549594	2014-06-17 00:00:00.000	823,00

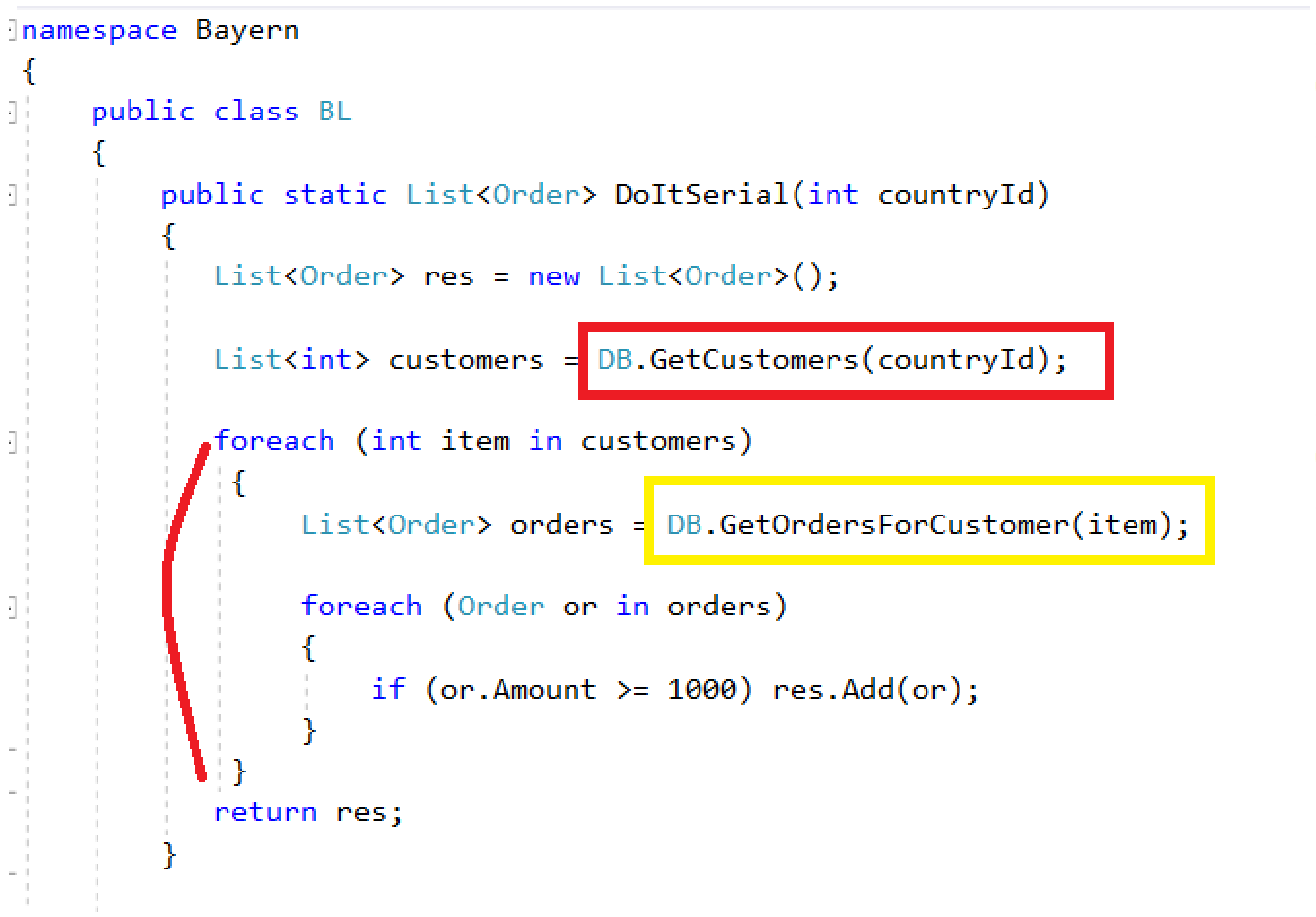
DEMO



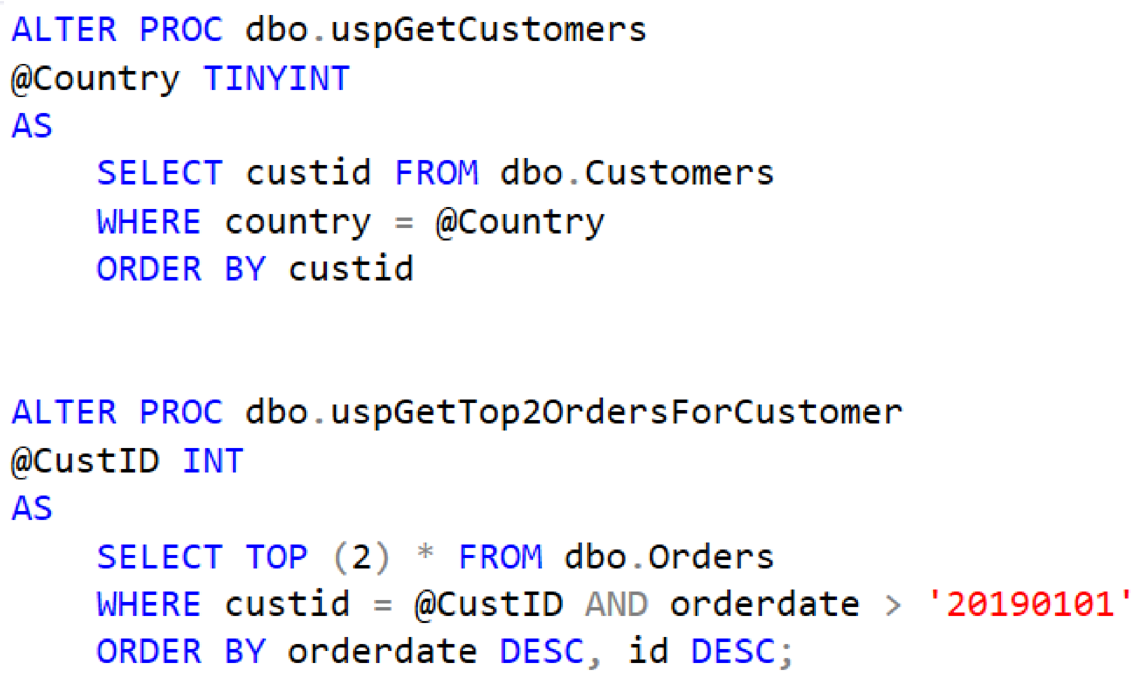
- CUSTOMERS & ORDERS PER COUNTRY:

- LUX - 100 => 1.923
- SWI - 1.000 => 18.721
- AUT - 10.000 => 188.960
- GER - 100.000 => 1.889.010
- USA - 1.000.000 => 18.901.386

FOR EACH – ITERATIVE APPROACH

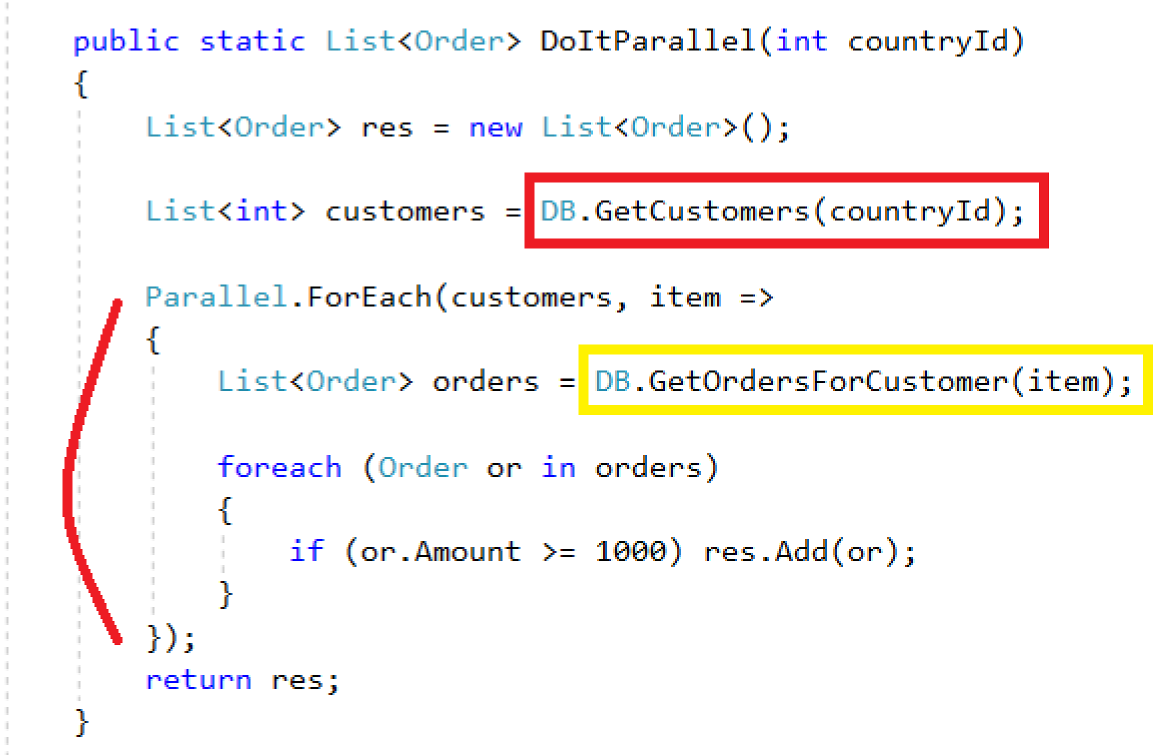


```
namespace Bayern
{
    public class BL
    {
        public static List<Order> DoItSerial(int countryId)
        {
            List<Order> res = new List<Order>();

            List<int> customers = DB.GetCustomers(countryId);

            foreach (int item in customers)
            {
                List<Order> orders = DB.GetOrdersForCustomer(item);

                foreach (Order or in orders)
                {
                    if (or.Amount >= 1000) res.Add(or);
                }
            }
            return res;
        }
```

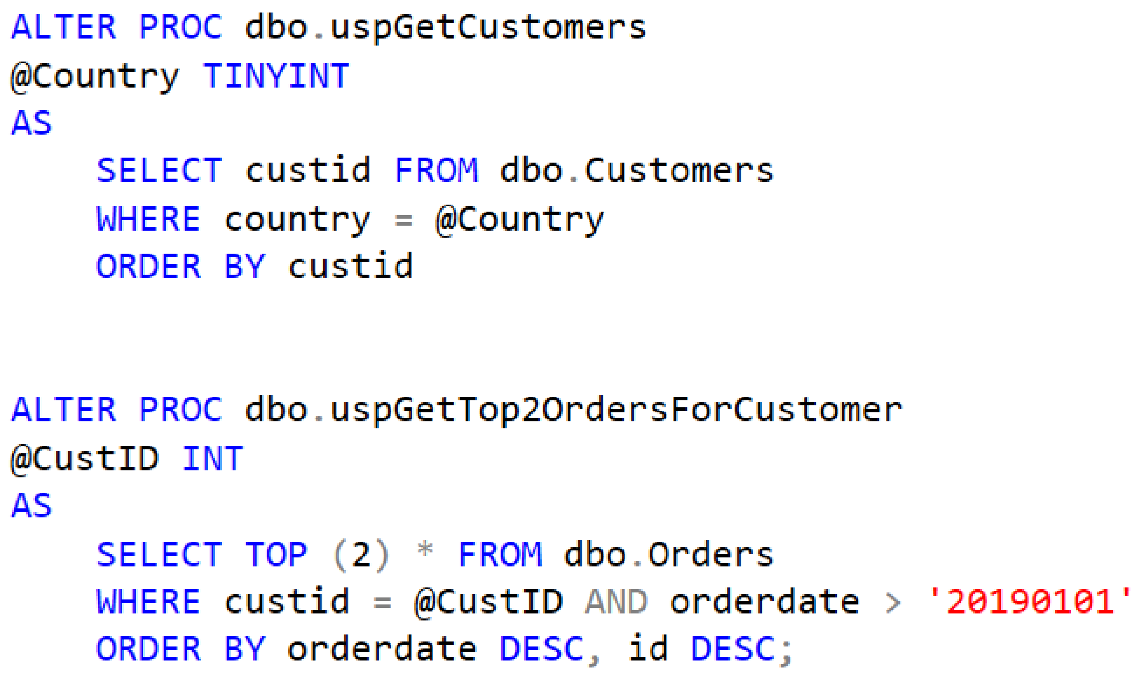


```
ALTER PROC dbo.uspGetCustomers
@Country TINYINT
AS
    SELECT custid FROM dbo.Customers
    WHERE country = @Country
    ORDER BY custid


ALTER PROC dbo.uspGetTop2OrdersForCustomer
@CustID INT
AS
    SELECT TOP (2) * FROM dbo.Orders
    WHERE custid = @CustID AND orderdate > '20190101'
    ORDER BY orderdate DESC, id DESC;
```

FOR EACH – PARALLEL APPROACH



```
public static List<Order> DoItParallel(int countryId)
{
    List<Order> res = new List<Order>();

    List<int> customers = DB.GetCustomers(countryId);

    Parallel.ForEach(customers, item =>
    {
        List<Order> orders = DB.GetOrdersForCustomer(item);

        foreach (Order or in orders)
        {
            if (or.Amount >= 1000) res.Add(or);
        }
    });
    return res;
}
```

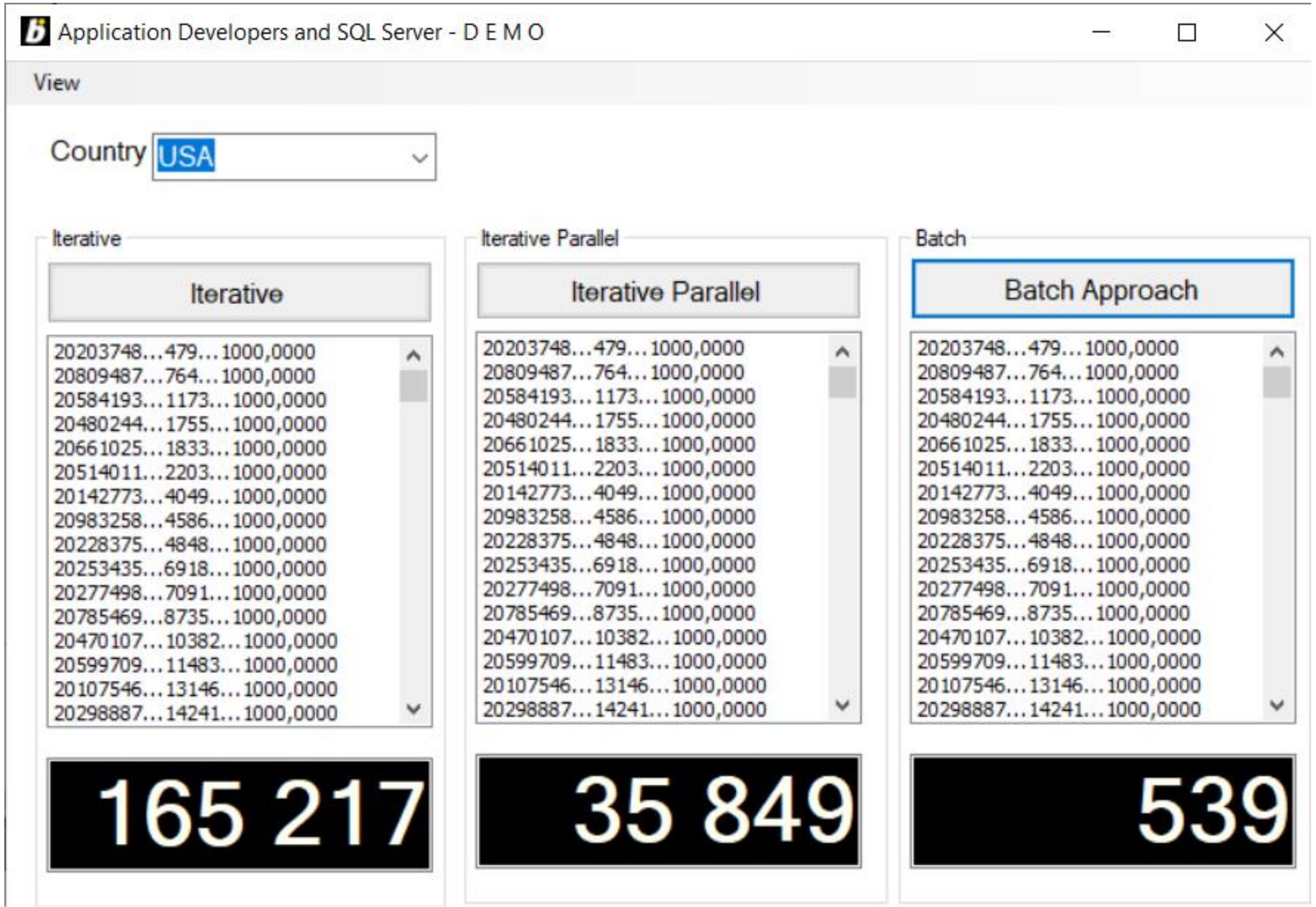


```
ALTER PROC dbo.uspGetCustomers
@Country TINYINT
AS
    SELECT custid FROM dbo.Customers
    WHERE country = @Country
    ORDER BY custid


ALTER PROC dbo.uspGetTop2OrdersForCustomer
@CustID INT
AS
    SELECT TOP (2) * FROM dbo.Orders
    WHERE custid = @CustID AND orderdate > '20190101'
    ORDER BY orderdate DESC, id DESC;
```

BATCH APPROACH

```sql
ALTER PROC dbo.uspGetOrdersForCustomers
@Country TINYINT
AS
    WITH cte AS(
    SELECT o.*,
        ROW_NUMBER() OVER(PARTITION BY o.custid ORDER BY o.orderdate DESC, o.id DESC) rn
    FROM dbo.Customers c
    INNER JOIN dbo.Orders o ON c.custid = o.custid
    WHERE orderdate > '20190101'
    AND country = @Country
    )
    SELECT id, custid, orderdate, amount FROM cte
    WHERE rn < 3 AND amount >= 1000
    ORDER BY custid;
```

FOR EACH — TOP PERFORMANCE KILLER



Iterative:

```
20203748...479...1000,0000
20809487...764...1000,0000
20584193...1173...1000,0000
20480244...1755...1000,0000
20661025...1833...1000,0000
20514011...2203...1000,0000
20142773...4049...1000,0000
20983258...4586...1000,0000
20228375...4848...1000,0000
20253435...6918...1000,0000
20277498...7091...1000,0000
20785469...8735...1000,0000
20470107...10382...1000,0000
20599709...11483...1000,0000
20107546...13146...1000,0000
20298887...14241...1000,0000
```

```
165 217
```

Iterative Parallel:

```
20203748...479...1000,0000
20809487...764...1000,0000
20584193...1173...1000,0000
20480244...1755...1000,0000
20661025...1833...1000,0000
20514011...2203...1000,0000
20142773...4049...1000,0000
20983258...4586...1000,0000
20228375...4848...1000,0000
20253435...6918...1000,0000
20277498...7091...1000,0000
20785469...8735...1000,0000
20470107...10382...1000,0000
20599709...11483...1000,0000
20107546...13146...1000,0000
20298887...14241...1000,0000
```

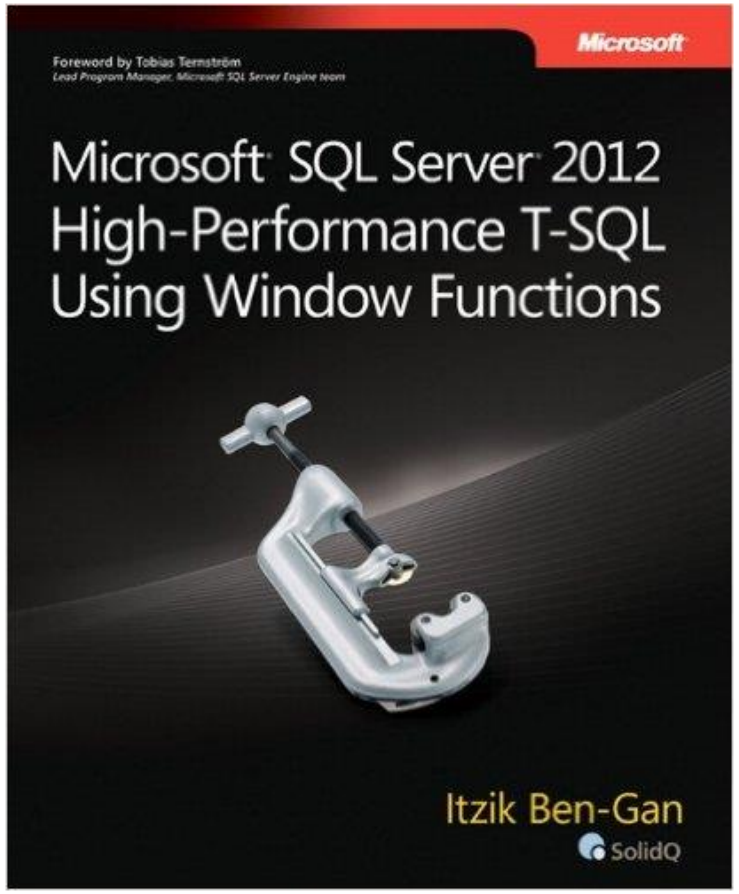
```
35 849
```

Batch:

```
20203748...479...1000,0000
20809487...764...1000,0000
20584193...1173...1000,0000
20480244...1755...1000,0000
20661025...1833...1000,0000
20514011...2203...1000,0000
20142773...4049...1000,0000
20983258...4586...1000,0000
20228375...4848...1000,0000
20253435...6918...1000,0000
20277498...7091...1000,0000
20785469...8735...1000,0000
20470107...10382...1000,0000
20599709...11483...1000,0000
20107546...13146...1000,0000
20298887...14241...1000,0000
```

```
539
```

TRANSACT-SQL KNOWLEDGE


Microsoft

Foreword by Tobias Ternström
Lead Program Manager, Microsoft SQL Server Engine team

Microsoft SQL Server 2012
High-Performance T-SQL
Using Window Functions

Itzik Ben-Gan
SolidQ

- CTE
- Window Functions
- APPLY Operator
- Paging

- Efficient finding or removing duplicates
- Compare previous and current values (i.e., previous and actual order for specific customer)
- Find most recent N items for an outer record