UOBIČAJENI TRANSACT-SQL ZADACI

MILOŠ RADIVOJEVIĆ, MICROSOFT DATA PLATFORM MVP

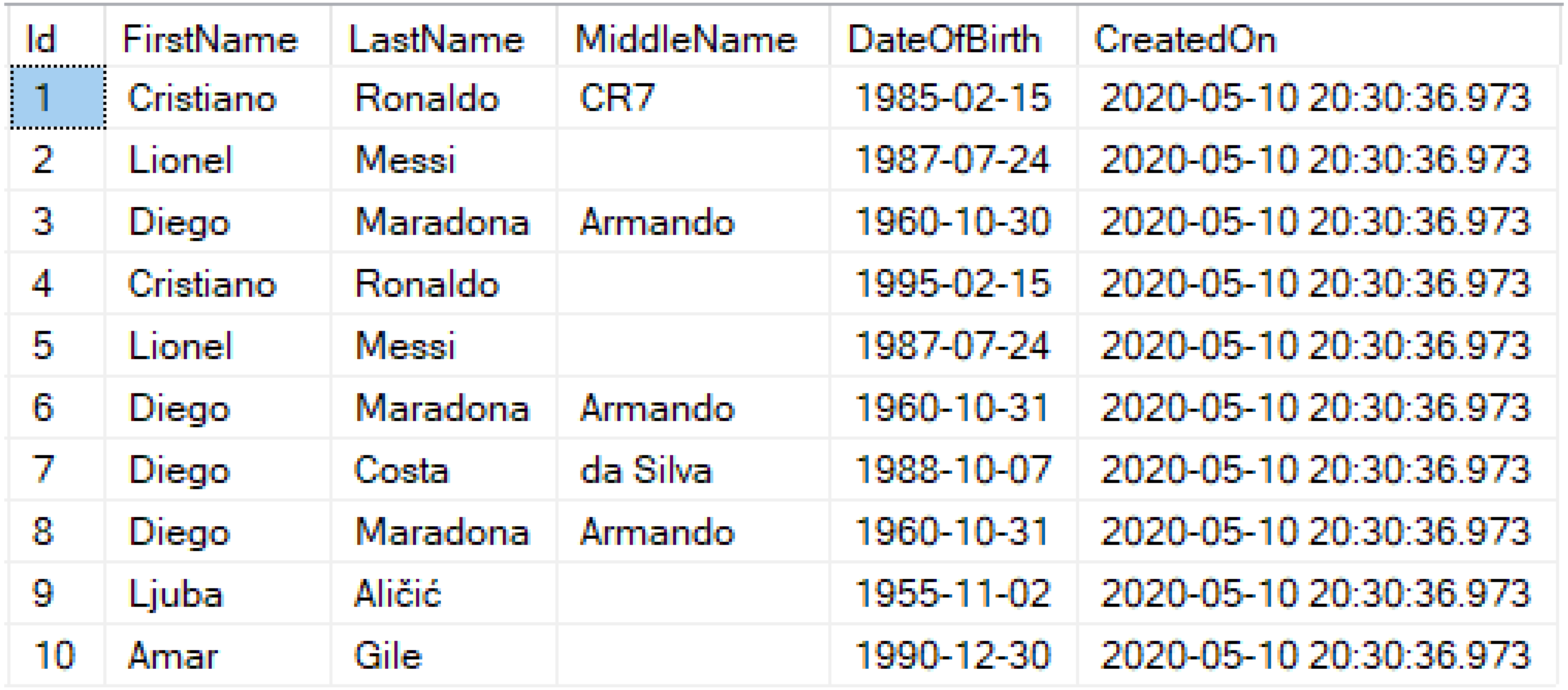
Agenda

- Uklanjanje duplikata
- Pronalaženje razlike između dva reda u tabeli sa mnogo kolona
- Računanje kumulativne sume (running total)
- Foreach petlja kao ubica performansi
- Vrednosti odvojene zarezima (CSL)

1 UKLANJANJE DUPLIKATA

Uklanjanje duplikata

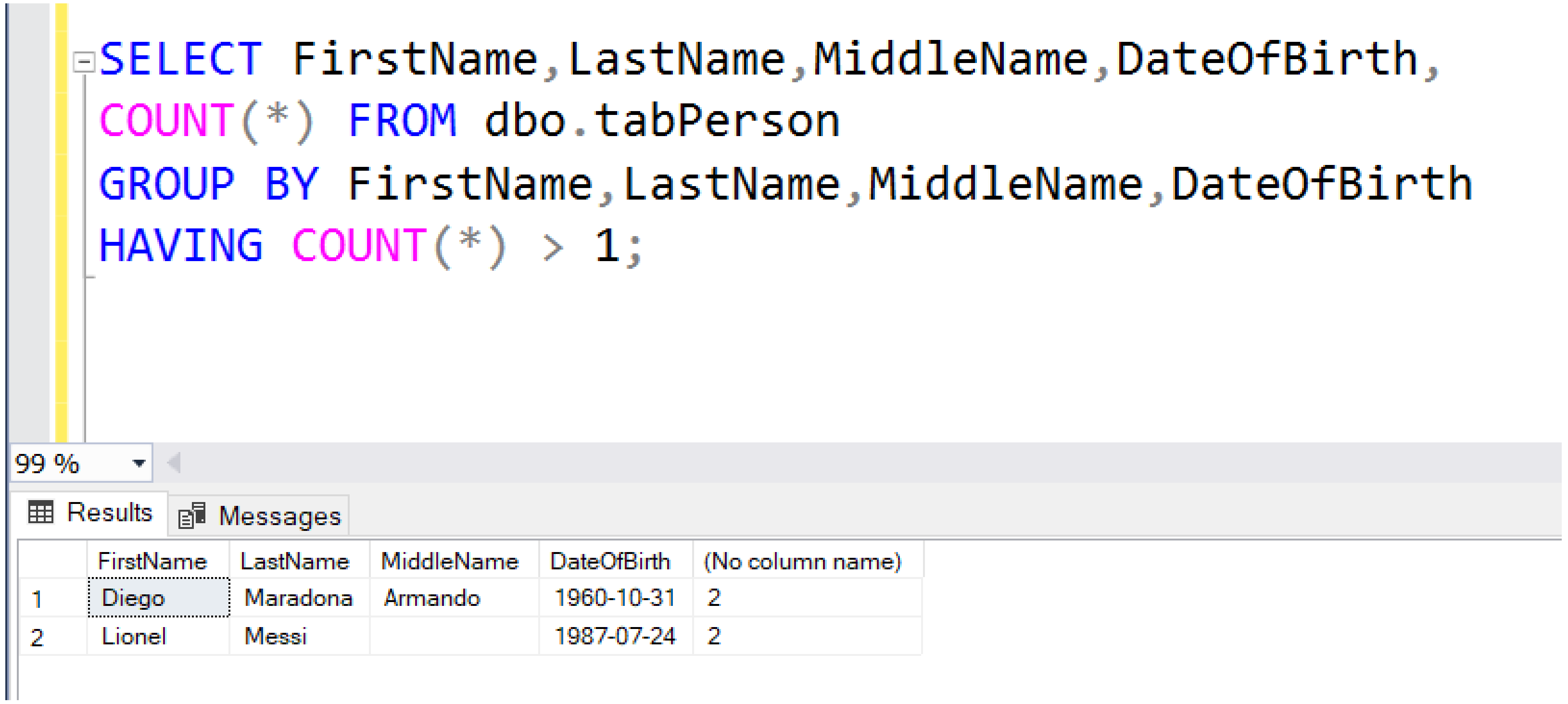
- Uklonite duplikate iz ovog skupa podataka!



Id	FirstName	LastName	MiddleName	DateOfBirth	CreatedOn
1	Cristiano	Ronaldo	CR7	1985-02-15	2020-05-10 20:30:36.973
2	Lionel	Messi		1987-07-24	2020-05-10 20:30:36.973
3	Diego	Maradona	Armando	1960-10-30	2020-05-10 20:30:36.973
4	Cristiano	Ronaldo		1995-02-15	2020-05-10 20:30:36.973
5	Lionel	Messi		1987-07-24	2020-05-10 20:30:36.973
6	Diego	Maradona	Armando	1960-10-31	2020-05-10 20:30:36.973
7	Diego	Costa	da Silva	1988-10-07	2020-05-10 20:30:36.973
8	Diego	Maradona	Armando	1960-10-31	2020-05-10 20:30:36.973
9	Ljuba	Aličić		1955-11-02	2020-05-10 20:30:36.973
10	Amar	Gile		1990-12-30	2020-05-10 20:30:36.973

Uklanjanje duplikata

- Ovaj upit identifikuje vrednosti koje se ponavljaju, ali ne i redove koje treba ukloniti



```
SELECT FirstName,LastName,MiddleName,DateOfBirth,
COUNT(*) FROM dbo.tabPerson
GROUP BY FirstName,LastName,MiddleName,DateOfBirth
HAVING COUNT(*) > 1;
```

	FirstName	LastName	MiddleName	DateOfBirth	(No column name)
1	Diego	Maradona	Armando	1960-10-31	2
2	Lionel	Messi		1987-07-24	2

ALI

Problem

- Pre nego što krenete sa pisanjem koda treba da nedvosmisleno utvrdite zahteve tj. koji se redovi smatraju duplikatima
  - Ono što je za Vas duplikat, za product ownera možda nije!
- Dve stvari moraju da budu napismeno definisane:
  - Kad su dva reda (ili više redova) smatraju identičnim
  - koji od pronađenih redova treba ostaviti

Preciziranje zahteva

- Dva ili više redova su identični, kada imaju istu vrednost za sve navedene atribute:

  FirstName, *LastName*, *MiddleName* and *DateOfBirth*

- Najnoviji red (red sa najvećim Id-em) se smatra redom koji treba da ostane u sistemu

Rešenje

- Window funkcija

Zajednički atributi

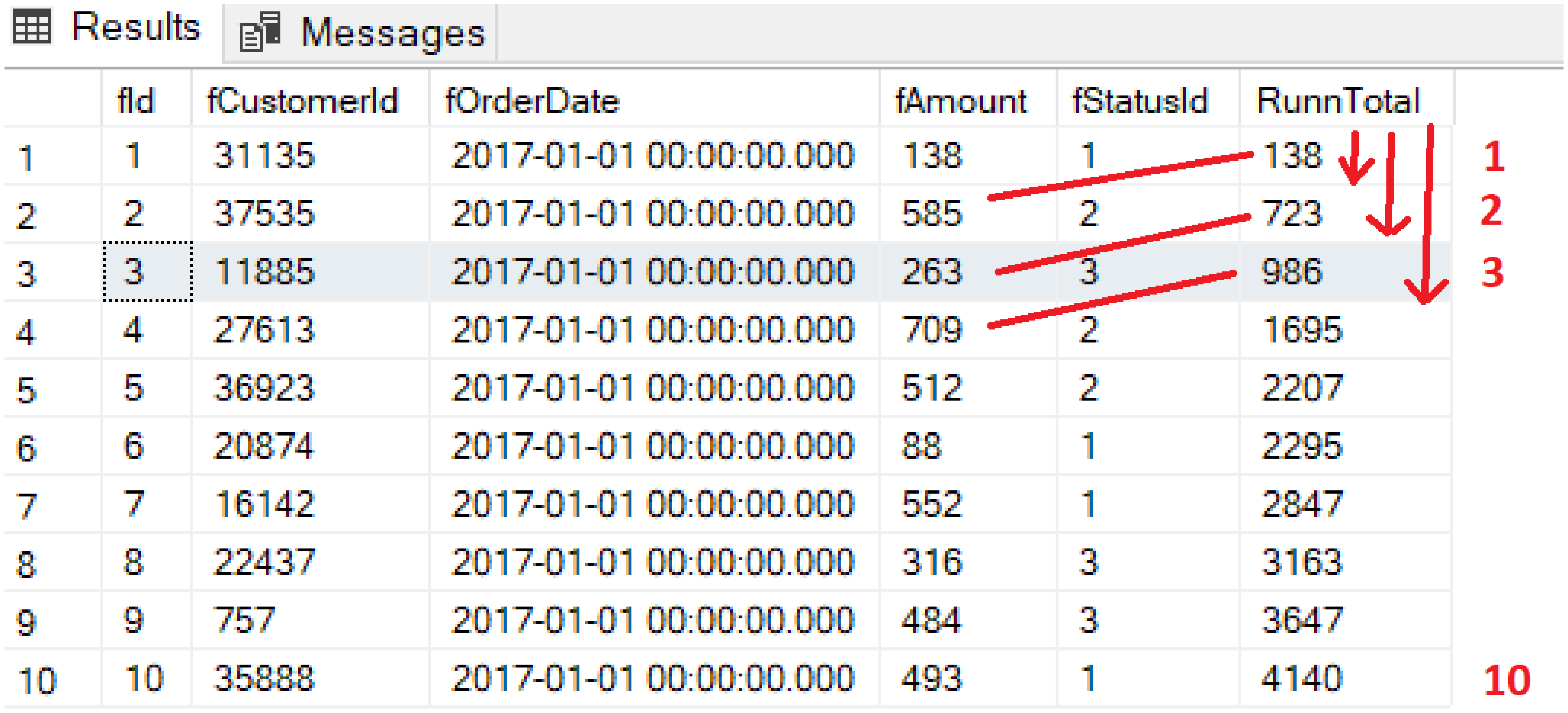
```
WITH cte AS(
SELECT *, ROW_NUMBER()
OVER(PARTITION BY FirstName, LastName, MiddleName, DateOfBirth
ORDER BY id DESC) rn
FROM dbo.tabPerson
)

DELETE FROM cte WHERE rn > 1;
```

original

2 RAZLIKA IZMEĐU DVA REDA NEKE TABELE

3 RUNNING TOTAL

Running total

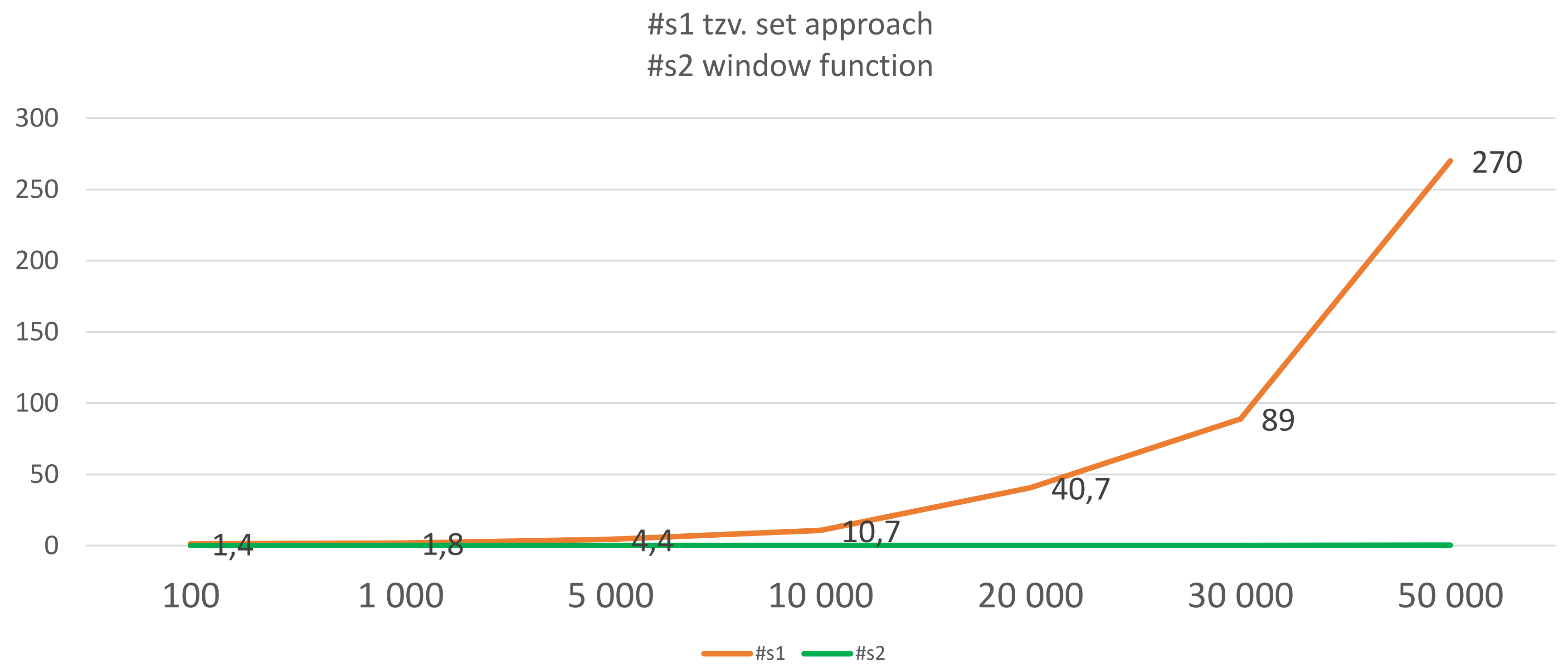


Results | Messages

	fId	fCustomerId	fOrderDate	fAmount	fStatusId	RunnTotal	
1	1	31135	2017-01-01 00:00:00.000	138	1	138	1
2	2	37535	2017-01-01 00:00:00.000	585	2	723	2
3	3	11885	2017-01-01 00:00:00.000	263	3	986	3
4	4	27613	2017-01-01 00:00:00.000	709	2	1695	
5	5	36923	2017-01-01 00:00:00.000	512	2	2207	
6	6	20874	2017-01-01 00:00:00.000	88	1	2295	
7	7	16142	2017-01-01 00:00:00.000	552	1	2847	
8	8	22437	2017-01-01 00:00:00.000	316	3	3163	
9	9	757	2017-01-01 00:00:00.000	484	3	3647	
10	10	35888	2017-01-01 00:00:00.000	493	1	4140	10

$$1 + 2 + 3 + \ldots + n = n^* (n+1)/2 \Rightarrow O(n^2)$$

Running total



```
SELECT fId, fAmount, SUM(fAmount)
OVER(ORDER BY fId ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) RunnTotal
FROM dbo.tabOrders o
ORDER BY o.fId;
```

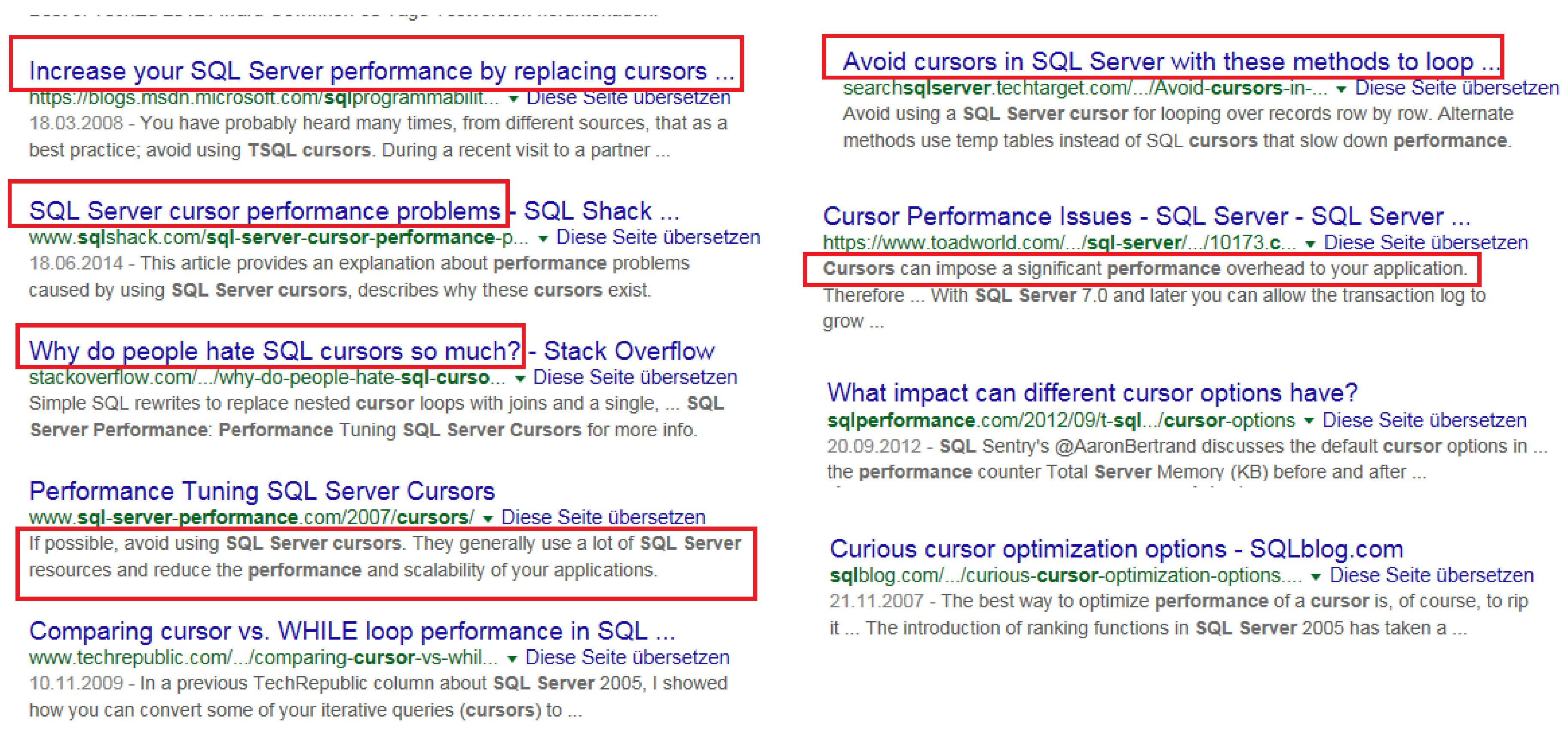
4 FOREACH PETLJA KAO UBICA PERFORMANSI

Tipični uzroci za nevolje sa performansama

- Loše indeksiranje
  - Ozbiljan problem, ali ne mora developer da ga rešava!
- ORM alati
  - Generišu kompleksne, suboptimalne upite
  - Korisnik ne vidi potrebu da uči Transact SQL
  - Ne možete da budete dobar developer bez znanja Transact-SQL-a!
- Row-by-Row Processing
  - Korišćenje kursora
  - Foreach petlja – skriveni ubica performansi i skalabilnosti

Row-by-Row Processing

- Kursori
  - Uživaju veoma lošu reputaciju i koriste se (i zbog toga) retko



Increase your SQL Server performance by replacing cursors ...
https://blogs.msdn.microsoft.com/sqlprogrammabilit... ▾ Diese Seite übersetzen
18.03.2008 - You have probably heard many times, from different sources, that as a best practice; avoid using **TSQL cursors**. During a recent visit to a partner ...

SQL Server cursor performance problems - SQL Shack ...
www.sqlshack.com/sql-server-cursor-performance-p... ▾ Diese Seite übersetzen
18.06.2014 - This article provides an explanation about **performance** problems caused by using **SQL Server cursors**, describes why these **cursors** exist.

Why do people hate SQL cursors so much? - Stack Overflow
stackoverflow.com/.../why-do-people-hate-sql-curso... ▾ Diese Seite übersetzen
Simple SQL rewrites to replace nested **cursor** loops with joins and a single, ... **SQL Server Performance**: **Performance** Tuning **SQL Server Cursors** for more info.

Performance Tuning SQL Server Cursors
www.sql-server-performance.com/2007/cursors/ ▾ Diese Seite übersetzen
If possible, avoid using **SQL Server cursors**. They generally use a lot of **SQL Server** resources and reduce the **performance** and scalability of your applications.

Comparing cursor vs. WHILE loop performance in SQL ...
www.techrepublic.com/.../comparing-cursor-vs-whil... ▾ Diese Seite übersetzen
10.11.2009 - In a previous TechRepublic column about **SQL Server** 2005, I showed how you can convert some of your iterative queries (**cursors**) to ...

Avoid cursors in SQL Server with these methods to loop ...
searchsqlserver.techtarget.com/.../Avoid-cursors-in-... ▾ Diese Seite übersetzen
Avoid using a **SQL Server cursor** for looping over records row by row. Alternate methods use temp tables instead of SQL **cursors** that slow down **performance**.

Cursor Performance Issues - SQL Server - SQL Server ...
https://www.toadworld.com/.../sql-server/.../10173.c... ▾ Diese Seite übersetzen
Cursors can impose a significant **performance** overhead to your application. Therefore ... With **SQL Server** 7.0 and later you can allow the transaction log to grow ...

What impact can different cursor options have?
sqlperformance.com/2012/09/t-sql.../cursor-options ▾ Diese Seite übersetzen
20.09.2012 - **SQL** Sentry's @AaronBertrand discusses the default **cursor** options in ... the **performance** counter Total **Server** Memory (KB) before and after ...

Curious cursor optimization options - SQLblog.com
sqlblog.com/.../curious-cursor-optimization-options.... ▾ Diese Seite übersetzen
21.11.2007 - The best way to optimize **performance** of a **cursor** is, of course, to rip it ... The introduction of ranking functions in **SQL Server** 2005 has taken a ...

Foreach Loop as Performance Killer

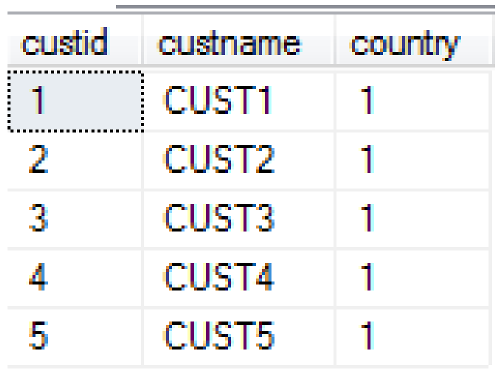
- Testne tabele Customers & Orders i broj redova u obe tabele za razne države:

- 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

Foreach Loop as Performance Killer

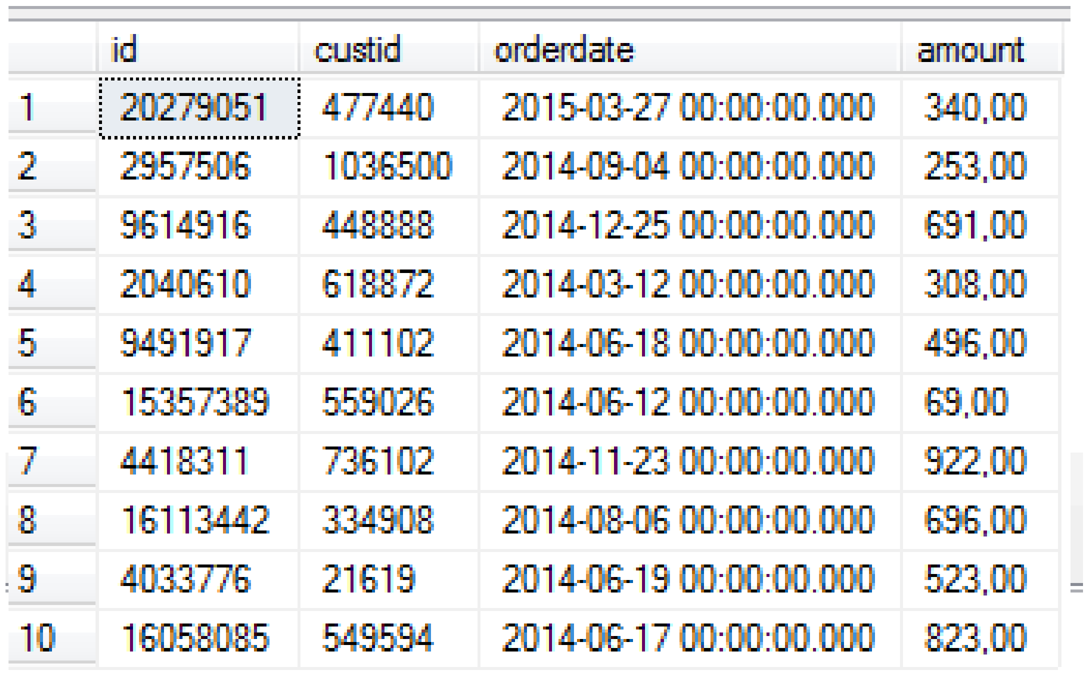
- Prikaži dva poslednja ordera za svakog klijenta ako su napravljena posle 1.1.2018. i ako je iznos narudžbe barem 1000 evra

Customers



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

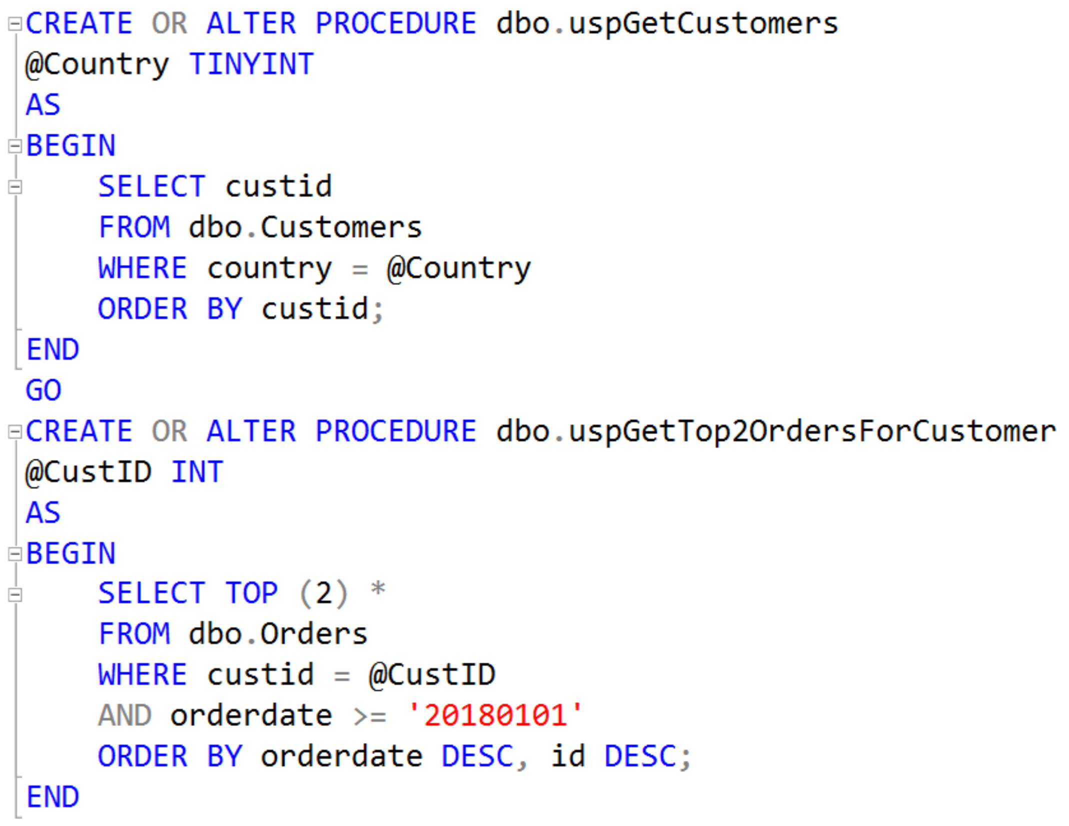
Orders



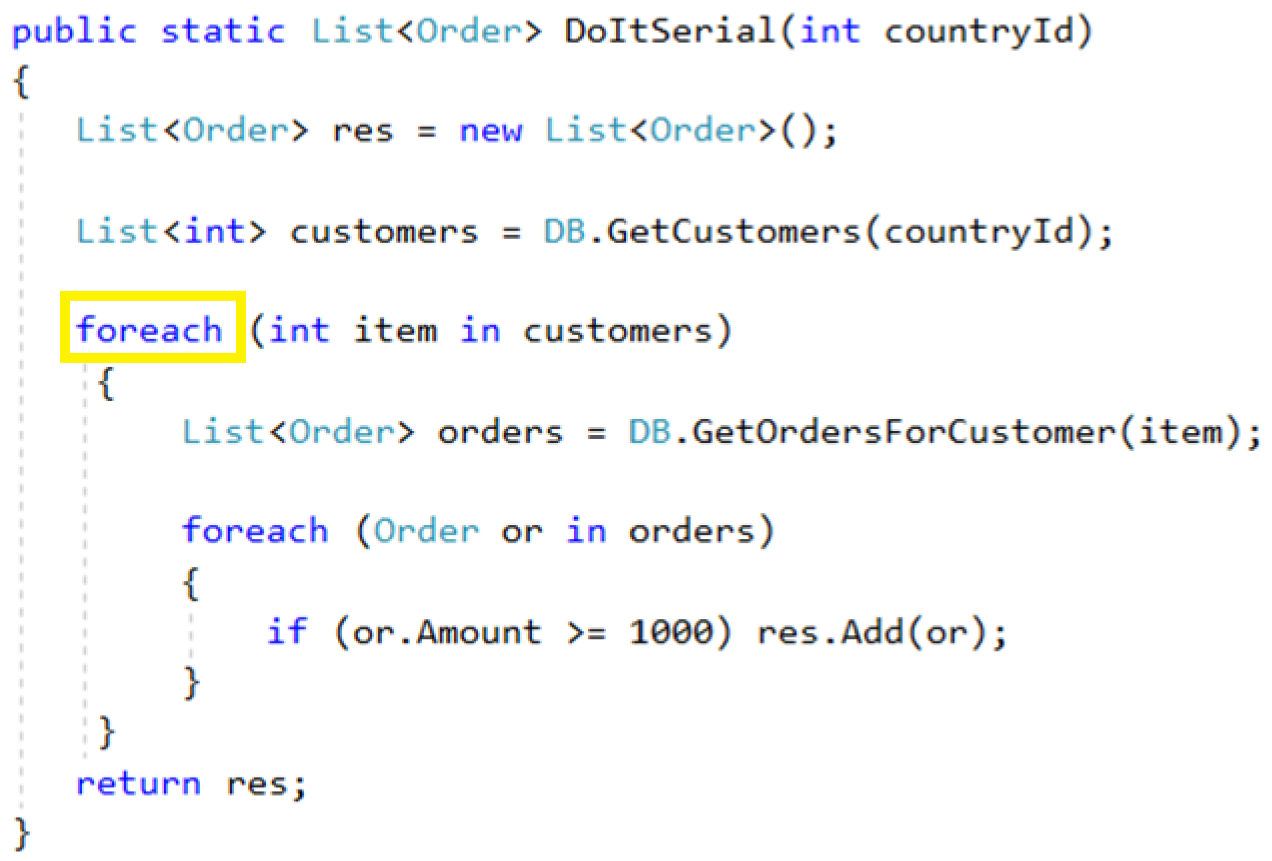
	id	custid	orderdate	amount
1	20279051	477440	2015-03-27 00:00:00.000	340,00
2	2957506	1036500	2014-09-04 00:00:00.000	253,00
3	9614916	448888	2014-12-25 00:00:00.000	691,00
4	2040610	618872	2014-03-12 00:00:00.000	308,00
5	9491917	411102	2014-06-18 00:00:00.000	496,00
6	15357389	559026	2014-06-12 00:00:00.000	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

Foreach – iterativni pristup

Two queries (stored procs) and a loop



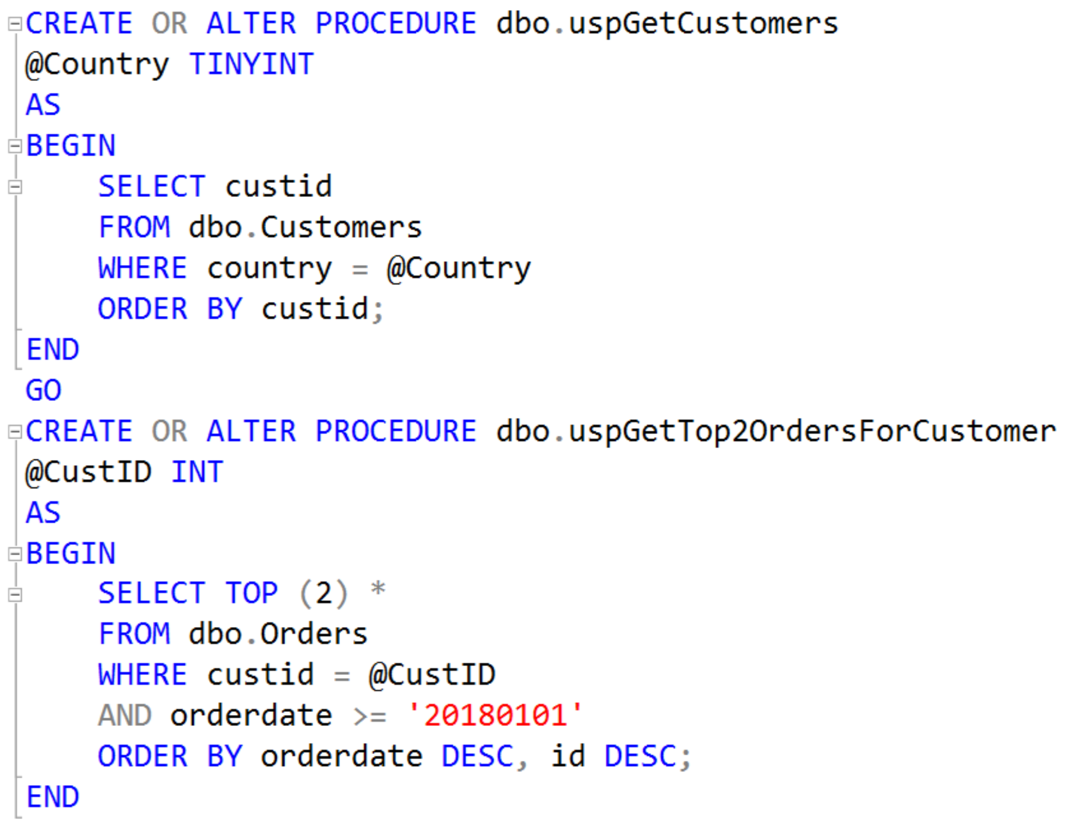
```sql
CREATE OR ALTER PROCEDURE dbo.uspGetCustomers
@Country TINYINT
AS
BEGIN
    SELECT custid
    FROM dbo.Customers
    WHERE country = @Country
    ORDER BY custid;
END
GO
CREATE OR ALTER PROCEDURE dbo.uspGetTop2OrdersForCustomer
@CustID INT
AS
BEGIN
    SELECT TOP (2) *
    FROM dbo.Orders
    WHERE custid = @CustID
    AND orderdate >= '20180101'
    ORDER BY orderdate DESC, id DESC;
END
```



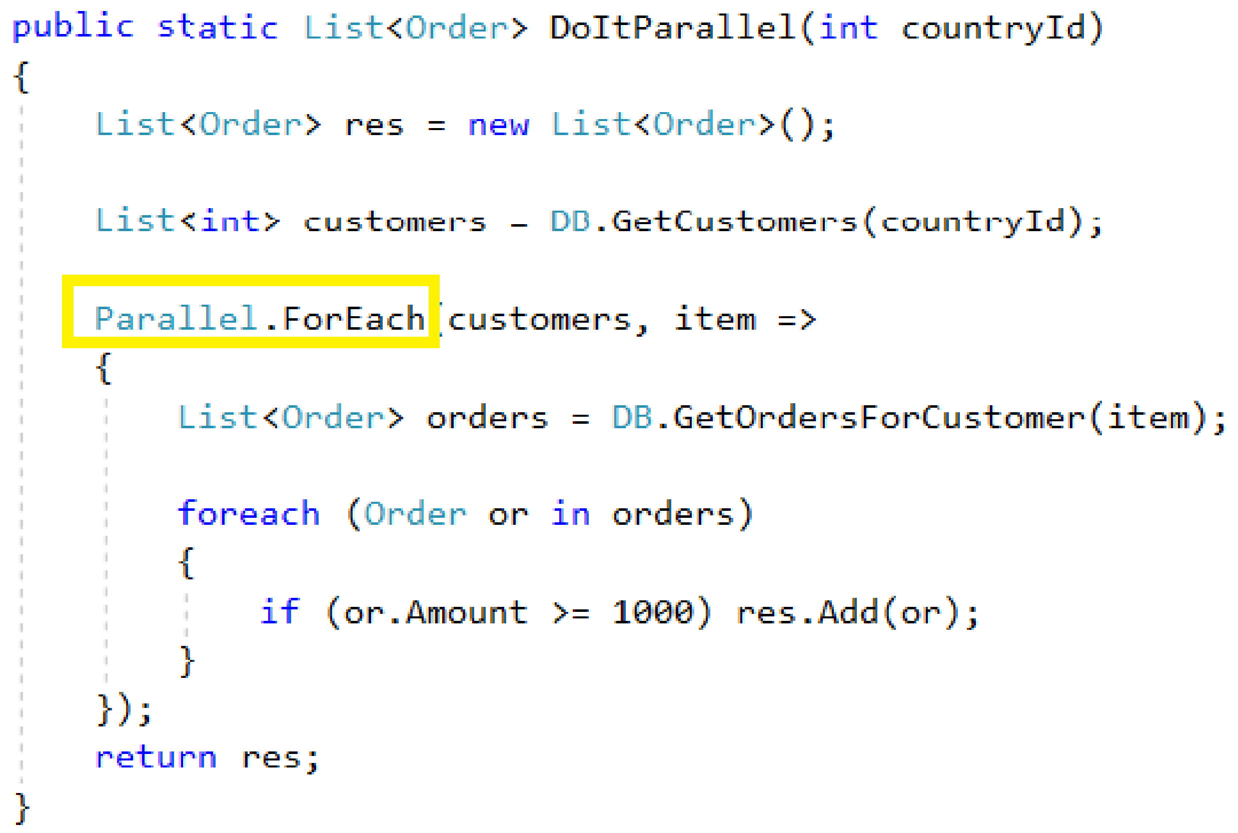
```csharp
public static List<Order> DoItSerial(int countryId)
{
    List<Order> res = new List<Order>();

    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;
}
```

Foreach – iterativni pristup

Foreach petlja i paralelno procesiranje



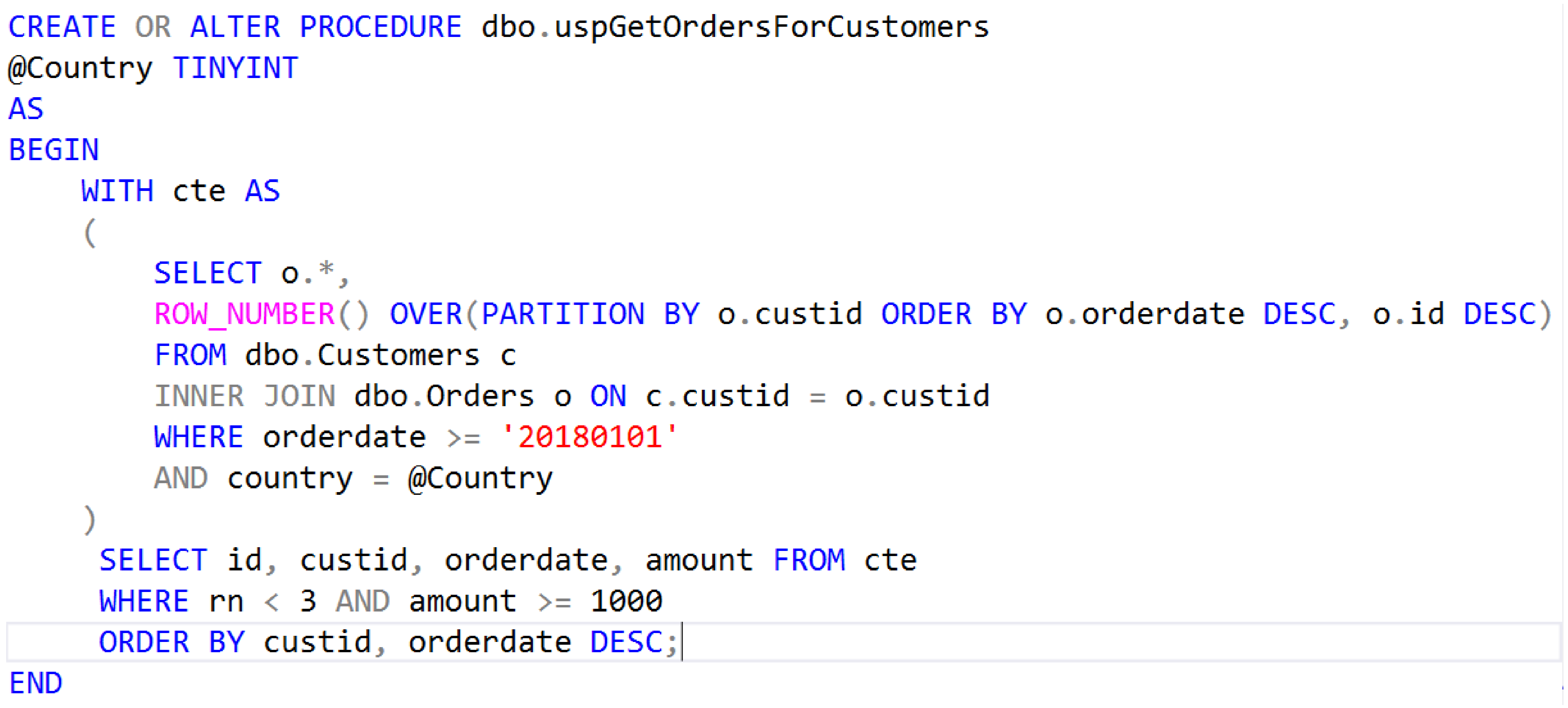
```sql
CREATE OR ALTER PROCEDURE dbo.uspGetCustomers
@Country TINYINT
AS
BEGIN
    SELECT custid
    FROM dbo.Customers
    WHERE country = @Country
    ORDER BY custid;
END
GO
CREATE OR ALTER PROCEDURE dbo.uspGetTop2OrdersForCustomer
@CustID INT
AS
BEGIN
    SELECT TOP (2) *
    FROM dbo.Orders
    WHERE custid = @CustID
    AND orderdate >= '20180101'
    ORDER BY orderdate DESC, id DESC;
END
```



```csharp
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;
}
```

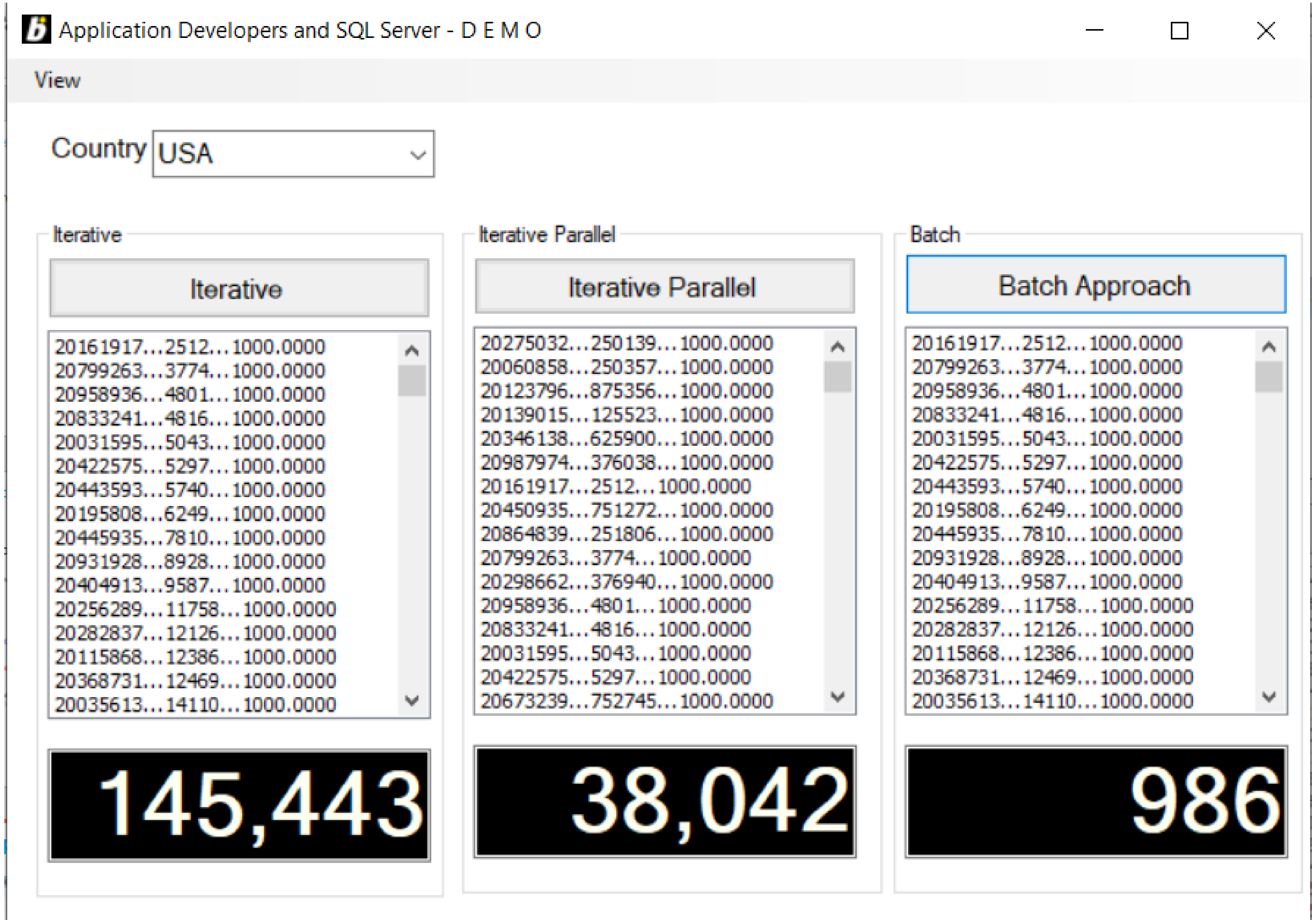
Foreach – „batch“ pristup

A single query (stored proc)



```
CREATE OR ALTER PROCEDURE dbo.uspGetOrdersForCustomers
@Country TINYINT
AS
BEGIN
    WITH cte AS
    (
        SELECT o.*,
        ROW_NUMBER() OVER(PARTITION BY o.custid ORDER BY o.orderdate DESC, o.id DESC)
        FROM dbo.Customers c
        INNER JOIN dbo.Orders o ON c.custid = o.custid
        WHERE orderdate >= '20180101'
        AND country = @Country
    )
     SELECT id, custid, orderdate, amount FROM cte
     WHERE rn < 3 AND amount >= 1000
     ORDER BY custid, orderdate DESC;
END
```

Foreach Loop as Performance Killer



```
20161917...2512...1000.0000
20799263...3774...1000.0000
20958936...4801...1000.0000
20833241...4816...1000.0000
20031595...5043...1000.0000
20422575...5297...1000.0000
20443593...5740...1000.0000
20195808...6249...1000.0000
20445935...7810...1000.0000
20931928...8928...1000.0000
20404913...9587...1000.0000
20256289...11758...1000.0000
20282837...12126...1000.0000
20115868...12386...1000.0000
20368731...12469...1000.0000
20035613...14110...1000.0000
```

145,443

```
20275032...250139...1000.0000
20060858...250357...1000.0000
20123796...875356...1000.0000
20139015...125523...1000.0000
20346138...625900...1000.0000
20987974...376038...1000.0000
20161917...2512...1000.0000
20450935...751272...1000.0000
20864839...251806...1000.0000
20799263...3774...1000.0000
20298662...376940...1000.0000
20958936...4801...1000.0000
20833241...4816...1000.0000
20031595...5043...1000.0000
20422575...5297...1000.0000
20673239...752745...1000.0000
```

38,042

```
20161917...2512...1000.0000
20799263...3774...1000.0000
20958936...4801...1000.0000
20833241...4816...1000.0000
20031595...5043...1000.0000
20422575...5297...1000.0000
20443593...5740...1000.0000
20195808...6249...1000.0000
20445935...7810...1000.0000
20931928...8928...1000.0000
20404913...9587...1000.0000
20256289...11758...1000.0000
20282837...12126...1000.0000
20115868...12386...1000.0000
20368731...12469...1000.0000
20035613...14110...1000.0000
```

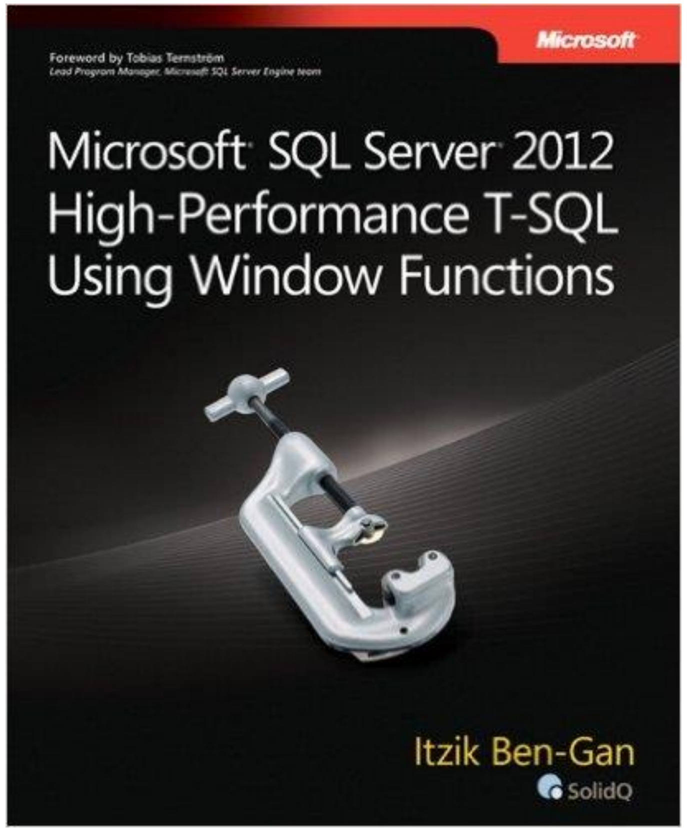
986

Foreach Loop as Performance Killer

- Zašto se ovo radi ovako. Šta kažu ti koji rade?
  - Business logic belongs to business layer
- Zašto to kažu?
  - Najčešće da bi pokrili to što ne znaju da napišu jedan upit koji to rešava
- Iterativni pristup je spor i ne skalira
- Još pride učitavaju (i prenose) mnogo više podataka nego što je potrebno. Različiti izgovori:
  - Moramo da budemo fleksibilni, možda nam treba za kasnije
- Ovaj problem ne može da se reši bez ponovnog pisanja aplikativnog koda!
- Transact-SQL znanje pravi razliku!

Transact-SQL Knowledge

- CTE
- Window Functions
- APPLY Operator


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

- 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

5 LISTE ODVOJENE ZAREZIMA