SQL Server Neues und Altbewährtes, Best Practices und How-To‘s für Entwickler

tkansy@dotnetconsulting.eu

tkansy

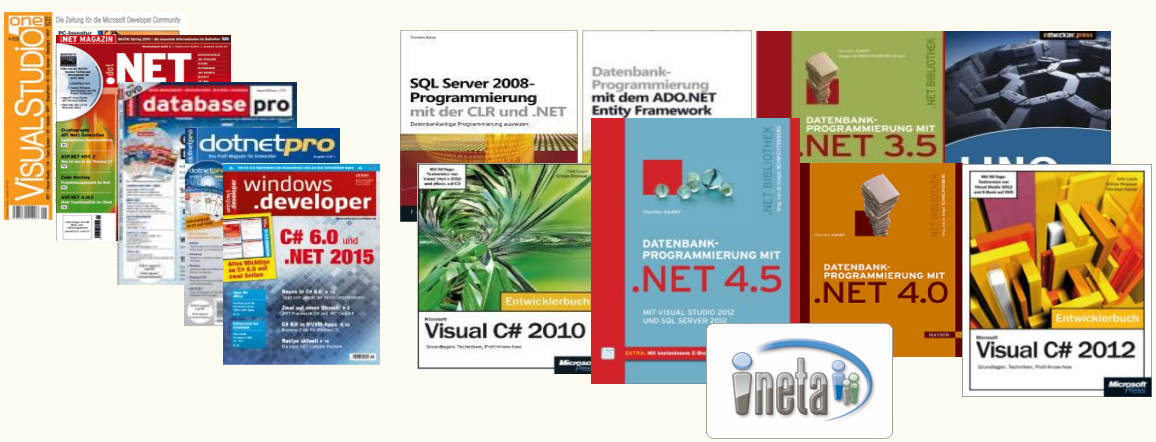
Thorsten Kansy | www.dotnetconsulting.eu

http://www.dotnetconsulting.eu

Über: Thorsten Kansy

Freier Consultant, Software Architekt,
Entwickler, Trainer & Fachautor

MCSE: Data Plattform & Business Intelligence
MCPD * MCTS * MCAD * MCSD
MCDBA * MCSE+I * MCT



http://www.dotnetconsulting.eu

DOTNETCONSULTING

Agenda

- Solider .NET Code
- Datenbankprojekte
- Unit Tests für Datenbankobjekte
- Effektive Sicherheit
- Performance
- Effizientes T-SQL
- Row Level Security
- Temporal Table (System Versioned Tables)
- GraphDb

Fertig? Los geht’s!

Solider .NET Code

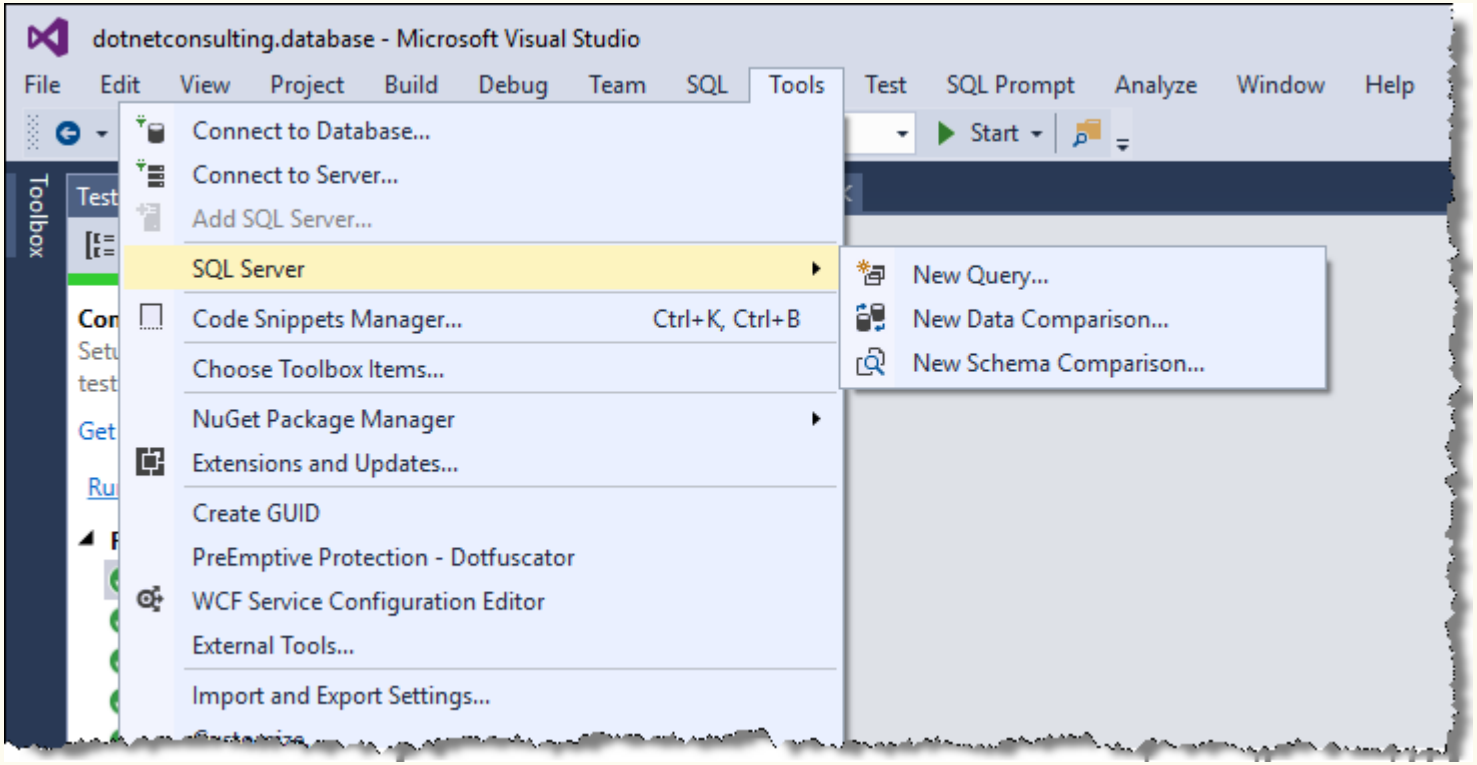
- `using`-Blöcke
- Parameter verwenden
  - Wo immer möglich!


Demo

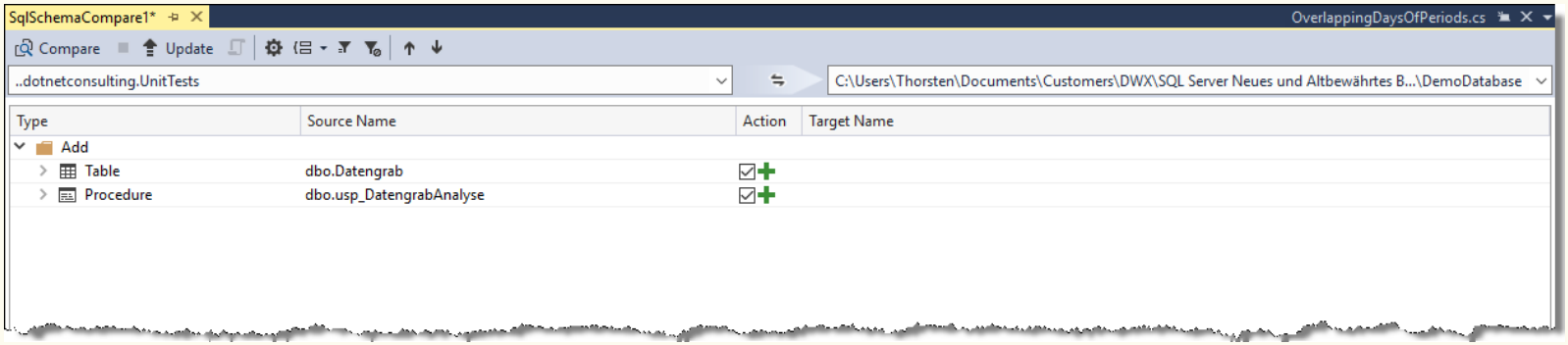
Datenbankprojekte

- Refactoring
- Aufspüren von Inkonsistenzen
- Deployment
  - DACPAC (Data-tier Application)
- Schema Compare
- Data Compare
- T4-Skripte
  - Automatische Objekterstellung

DbProjekte: Schema Compare



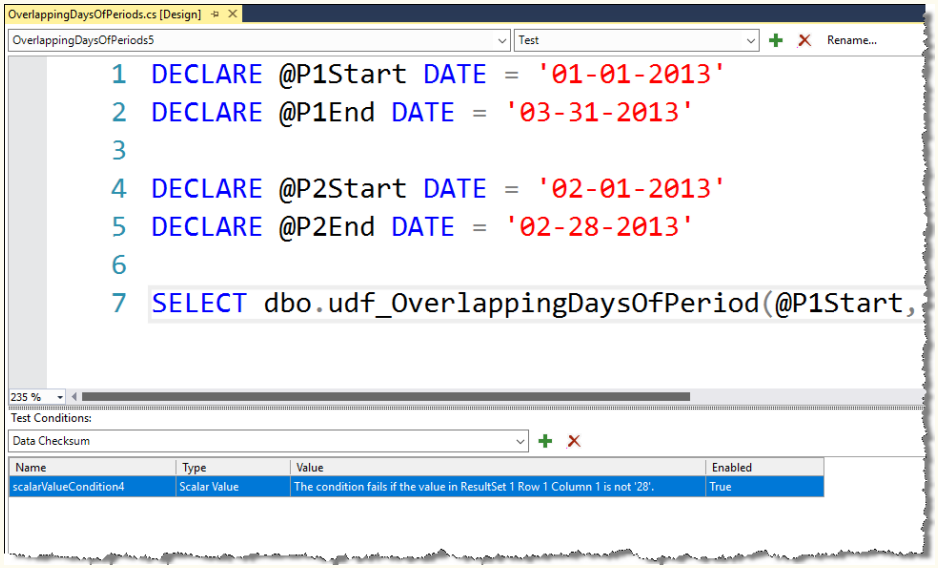
DbProjekte: Schema Compare

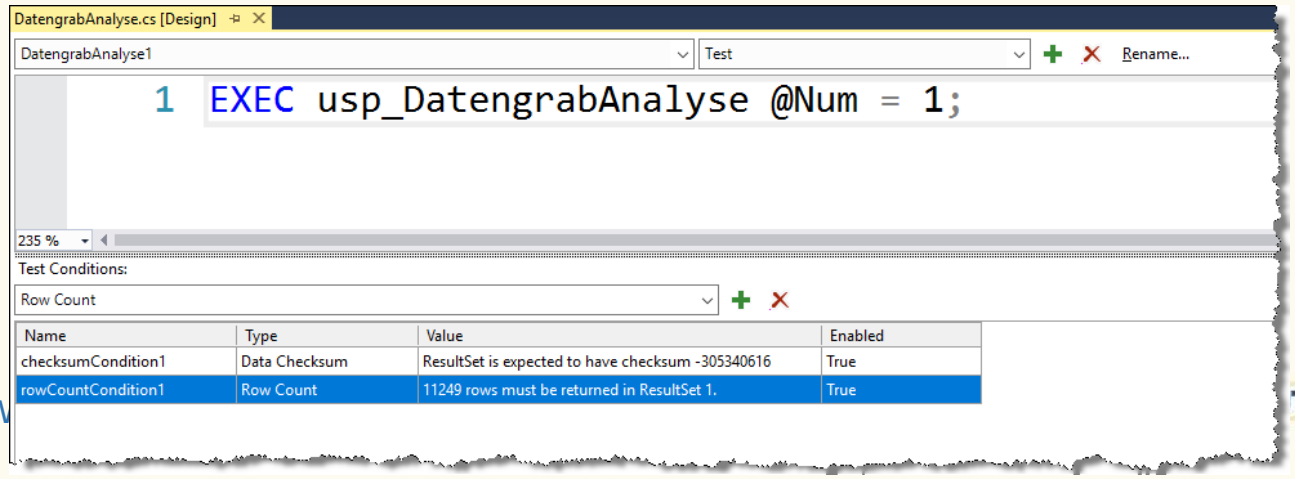


DOTNETCONSULTING


Demo

Unit Tests für Datenbankobjekte



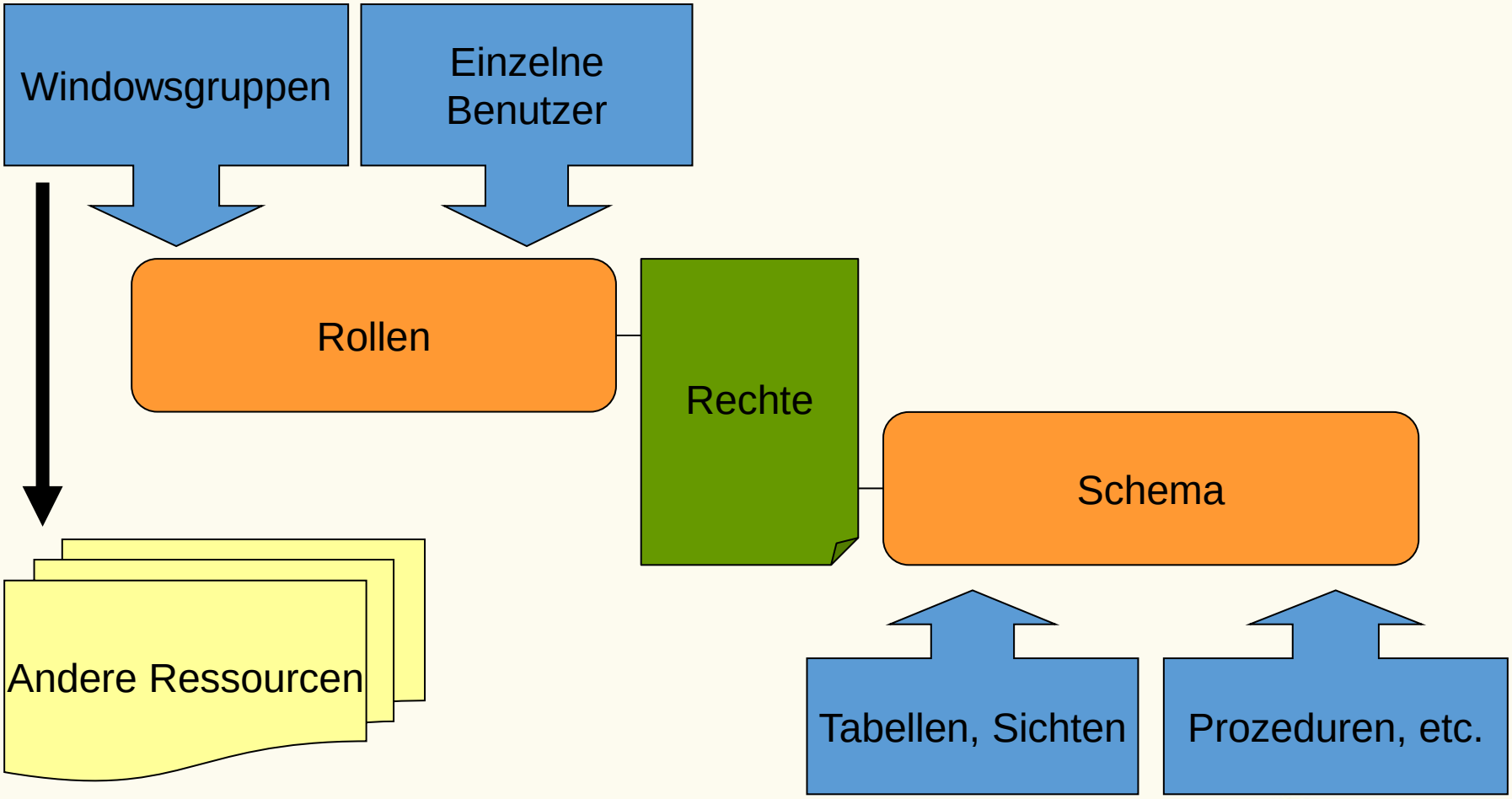


http://w

TNETCONSULTING


Demo

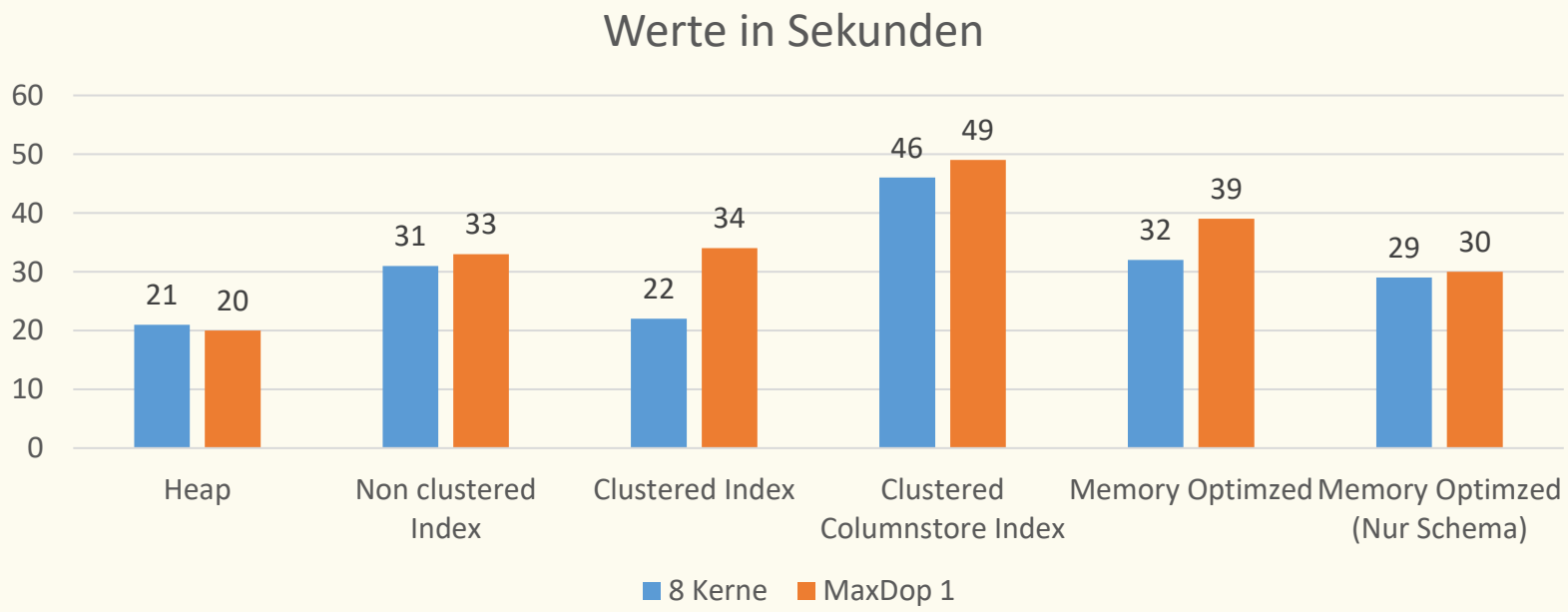
Effektive Sicherheit



Performance: Indizes

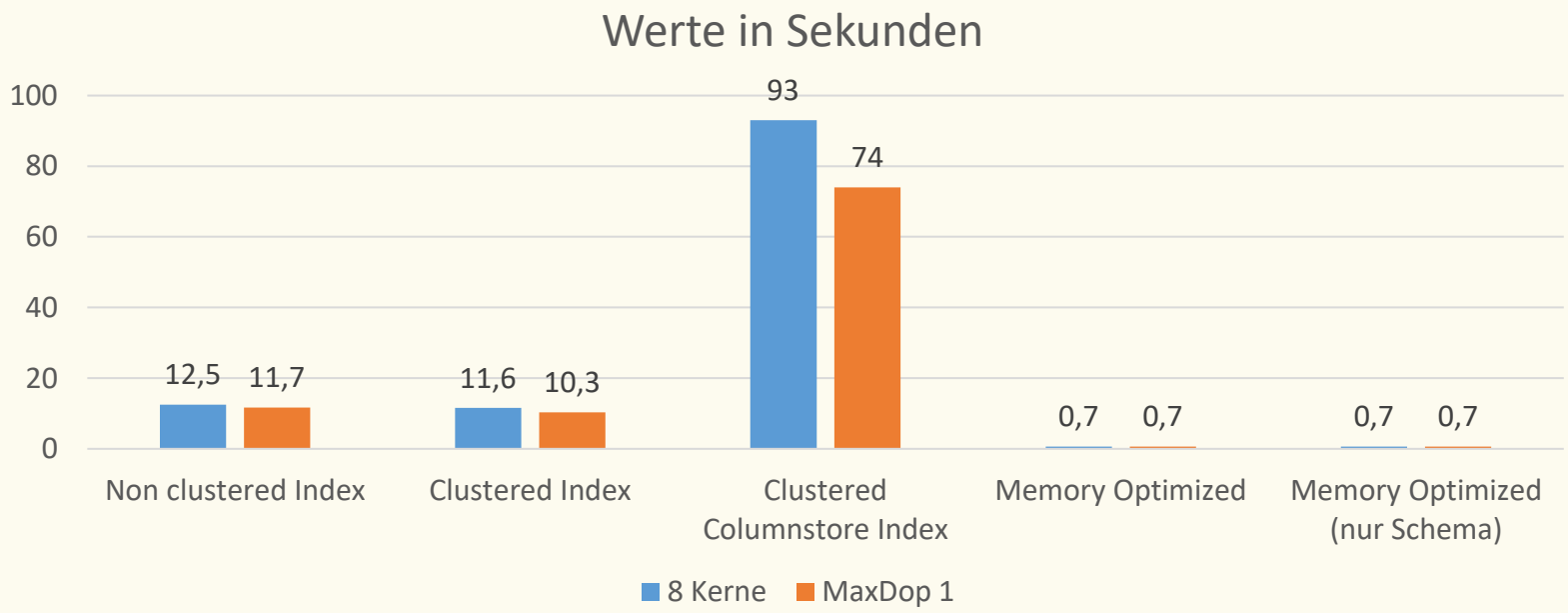
- Clustered Index
  Häufigster Zugriff (Primary Key)
- Non Clustered Index
  Häufige Zugriffe (Suchen)
- Clustered Columnstore Index
  Große Zeilenanzahl und statische Daten
- Memory Optimized (Schema Only)
  Schnelle Erfassung/Zwischenergebnisse
  Bei viel RAM eine Option
- Zu viele Indizes schaden der Performance

Insert 10 Mio. Zeilen



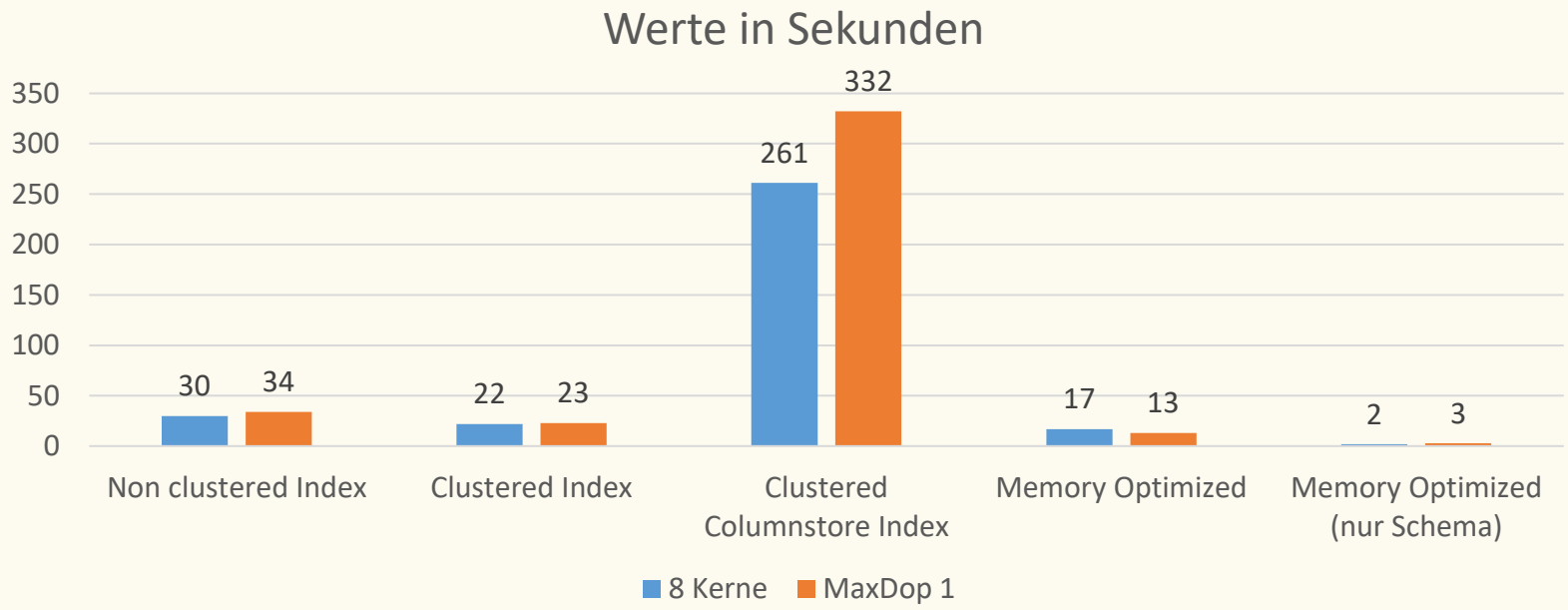
DOTNET CONSULTING

Select 100.000 Zeilen



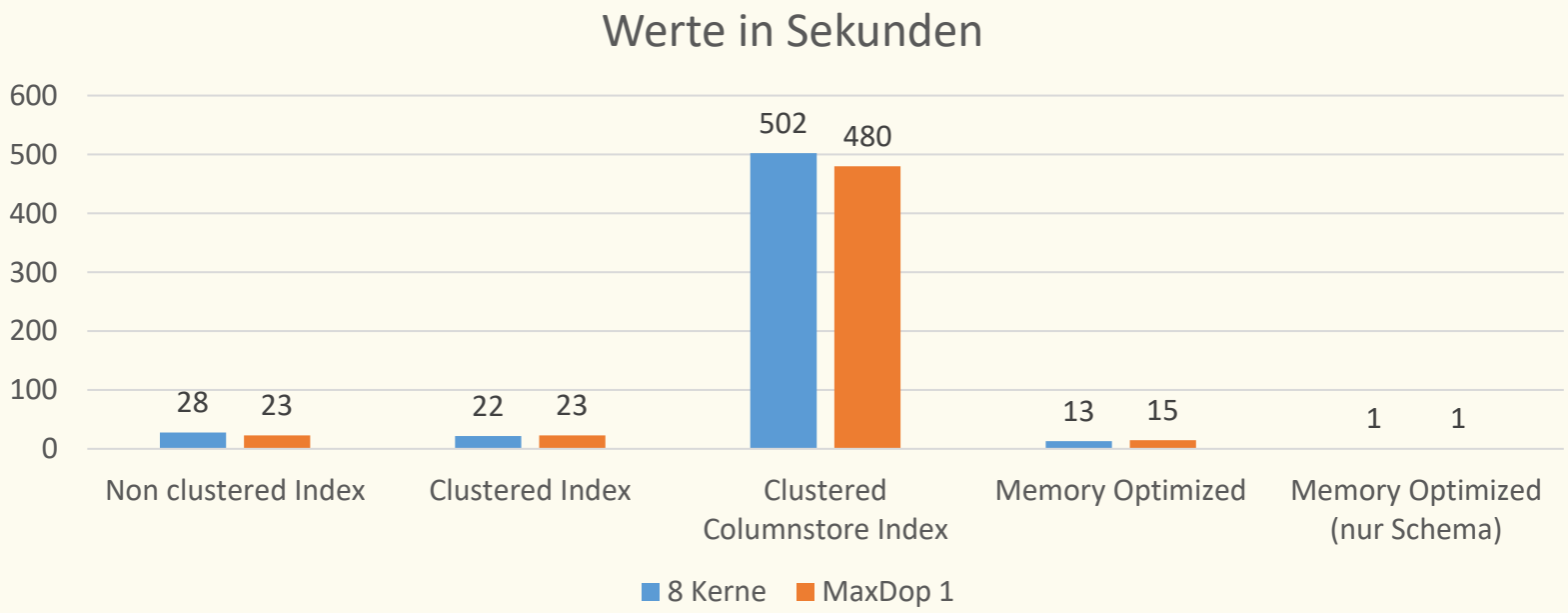
DOTNET CONSULTING

Update 100.000 Zeilen

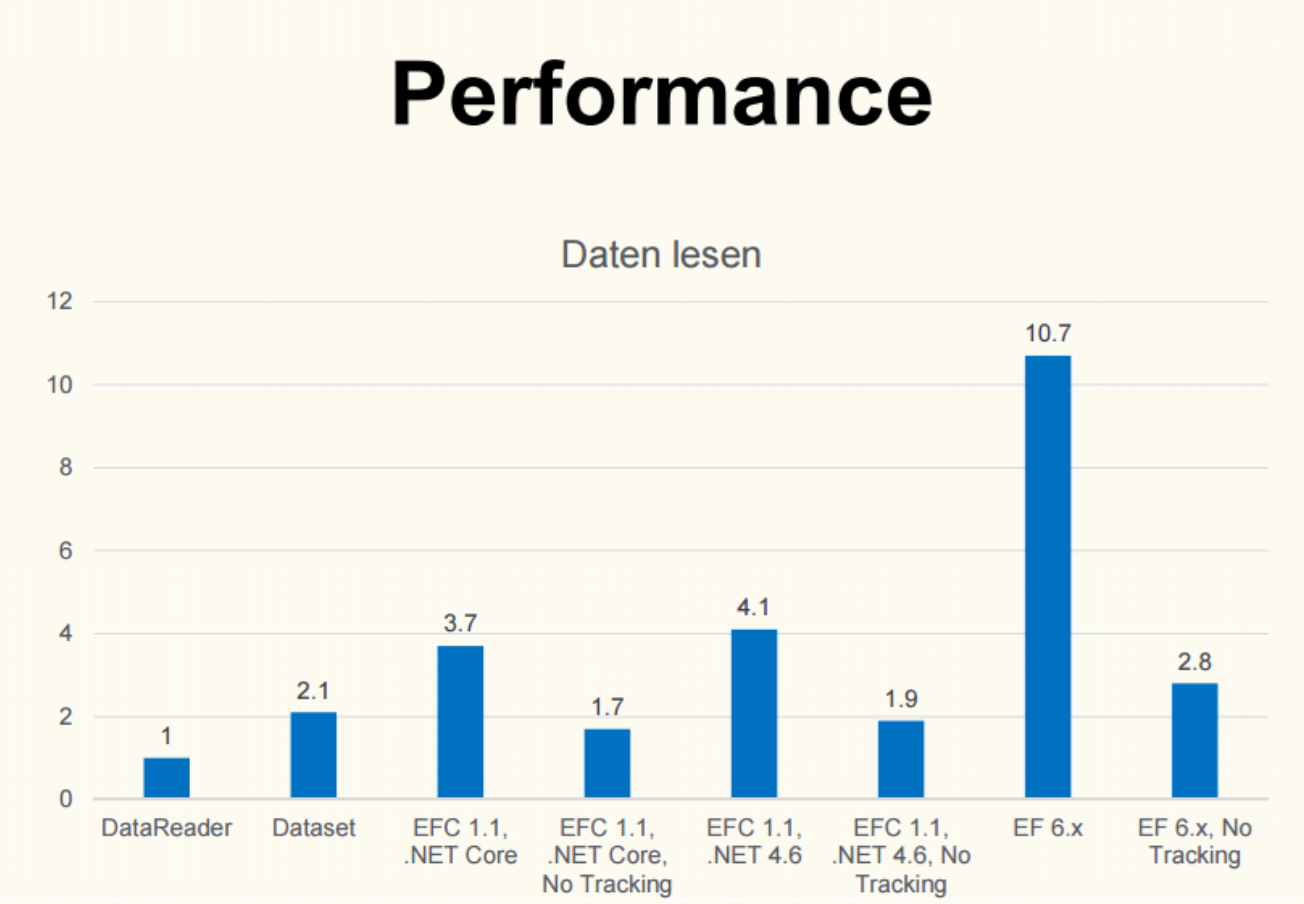


DOTNET CONSULTING

Delete 100.000 Zeilen



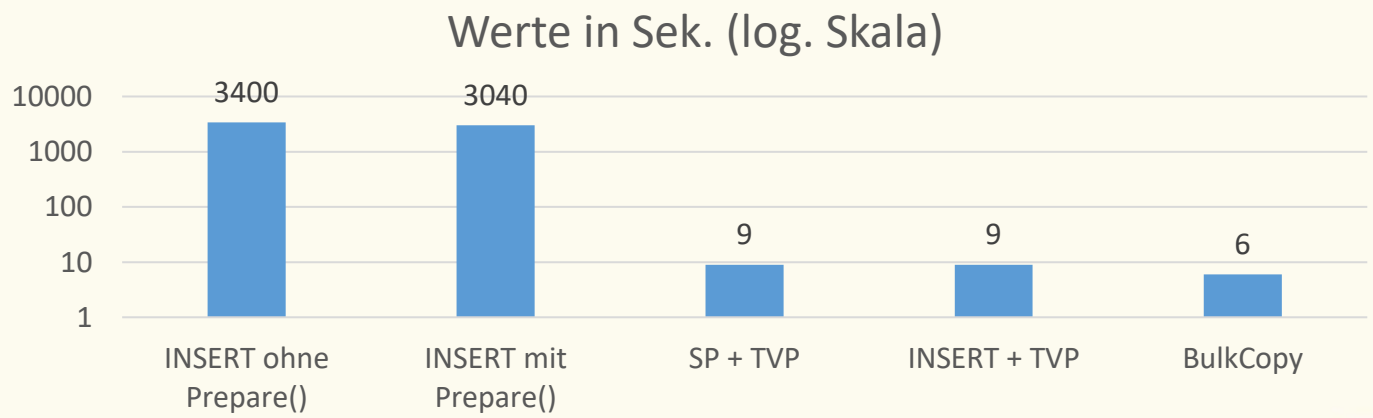
Performance Vergleich (EF)



DOTNET CONSULTING

Performance Vergleich (1 Mio Zeilen)

Methode	1.000.000 Zeilen Import
INSERT ohne Prepare()	3400 s
INSERT mit Prepare()	3040 s
Stored Procedure + TVP	9 s
INSERT + TVP	9 s
BulkCopy	6 s



DOTNET CONSULTING

Effizientes T-SQL

- TVPs
- OUTPUT
- SQLCMD-Mode

- Stored Procedures >
  - Views >
    - TVF (Inline) >
      - TVF (Multistatement)

Temporal Table (2016)

Automatische Historisierung von Daten

```
CREATE TABLE dbo.Werte
(
       ID INT IDENTITY(1,1) NOT NULL,
       Wert1 NVARCHAR(10) NULL,
       Wert2 NVARCHAR(10) NULL,
       StartTime datetime2(7) GENERATED ALWAYS AS ROW START HIDDEN NOT NULL,
       EndTime datetime2(7) GENERATED ALWAYS AS ROW END HIDDEN NOT NULL,
       PERIOD FOR SYSTEM_TIME(StartTime, EndTime),
       CONSTRAINT Werte_PK PRIMARY KEY (ID)
)
WITH
(SYSTEM_VERSIONING = ON (HISTORY_TABLE = dbo.Werte_History));


SELECT * FROM dbo.Werte
  [FOR SYSTEM_TIME AS OF '2016-01-25 18:21:24.0738473'];
```


Demo

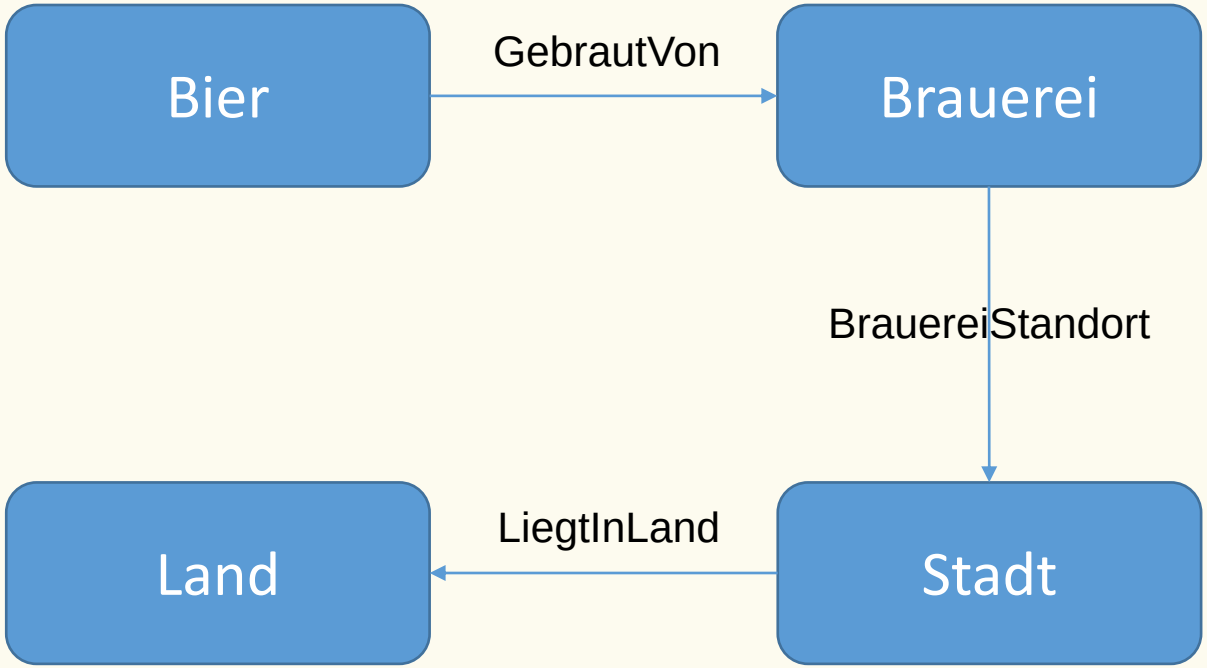
Row Level Security (2016)

Zeilenweiser, funktionsbasierter Filter

```
CREATE SECURITY POLICY SecurityFilter
  ADD FILTER PREDICATE dbo.fn_AccessFilter([SecurityDescriptor]) ON
    [dbo].[Mitarbeiter],
  ADD BLOCK PREDICATE dbo.fn_AccessFilter([SecurityDescriptor]) ON
    [dbo].[Mitarbeiter] AFTER INSERT,
  ADD BLOCK PREDICATE dbo.fn_AccessFilter([SecurityDescriptor]) ON
    [dbo].[Mitarbeiter] BEFORE DELETE,
  ADD BLOCK PREDICATE dbo.fn_AccessFilter([SecurityDescriptor]) ON
    [dbo].[Mitarbeiter] BEFORE UPDATE;
```


Demo

GraphDb (2017)



GraphDb (2017)

```
SELECT    *
FROM [dbo].[Biere],
     [dbo].[GebrautVon],
     [dbo].[Brauereien],
     [dbo].[Brauereistandort],
     [dbo].[Staedte]
WHERE
MATCH (Biere-(GebrautVon)->Brauereien-
(Brauereistandort)->Staedte)
AND [Staedte].[Name] = 'Stadt #7';
```


Demo

Und sonst so?

Deathstar *

Common Table Expressions

Volltextsuche

File Tables

CLR Integration

Table Types

Transaktionen

Fragen?

Support für Ihr Projekt

http://www.dotnetconsulting.eu

DOTNETCONSULTING

Links


http://dotnetconsulting.eu/blog/


@Tkansy


tkansy@dotnetconsulting.eu


www.dotnetconsulting.eu