



# Team Developer 7.0

Teil 2

MD – Roadshow 2016

# Agenda

- Neue Funktionalitäten in der Objekt Orientierten Programmierung
- Table – Grid Vergleich
  - Funktionalität
  - Performance
  - Testanwendung

# Team Developer 7.0

- Objekt Orientierte Programmierung (OOP)
  - Seit vielen Jahren entwickelt sich die OOP ständig weiter
  - Es sind einige 'Techniken' entstanden, die Microsoft und andere Anbieter von Entwicklungsumgebungen vorangetrieben haben
    - Klassen
    - Methoden / Funktionen
    - Kapselung
    - Vererbung
    - Mehrfachvererbung

# Team Developer 7.0



- OOP im Team Developer
  - Funktionale & GUI Klassen
  - Instanz Variablen
  - Funktionen
  
- Vererbung
  - Sehr detaillierte Vererbungsmöglichkeiten
  - Multiple Vererbung
  - Late Bound Calls
  - Überschreiben von Funktionen

# Team Developer 7.0

- Seit einigen Versionen werden folgende Techniken unterstützt:
  - Constructors / Destructor
- Seit TD 7.0
  - Function Overload
  - Protected Functions / Variables

# Team Developer 7.0

- Constructor
  - ‘Vorbereiten’ einer Klasse: Initialisieren
    - Default Constructor
      - Ohne Parameter
    - Constructoren mit unterschiedlichen Parametern
    - Neuer Constructor/Destructor Node in der Outline
  - Wann / wie wird ein Constructor aufgerufen?
    - MyClass: obj1
    - MyClass: obj1 = OBJ\_Null
    - Set obj1 = New MyClass( )
    - Set obj1 = New myClass (10, “Foo”)

```
◆ Functional Class: Customer
  ◇ Description:
  ◇ Derived From
  ◇ Class Variables
  ◆ Instance Variables
  ◇ Number: CustomerID
  ◆ Constructor/Destructor
  ◆ Constructor: Customer|
  ◇ Description:
  ◆ Parameters
  ◇ Number: pCustId
  ◇ Local variables
  ◆ Actions
  ◇ Set CustomerID = pCustId
  ◇ ! Load customer details from DB
  ◆ Destructor: Customer
  ◇ Description:
  ◇ Local variables
  ◆ Actions
  ◇ ! Cleanup, close connections etc.
  ◇ Functions
```

# Team Developer 7.0

- Constructor
  - Sind sehr wichtig bei der Vererbung:
    - Alle Konstruktoren der Basisklassen werden durchlaufen
      - Kein extra Code nötig
    - Team Developer startet alle Default-Constructor aller beteiligten Klassen
    - Bei Mehrfachvererbung gilt die Reihenfolge in der Outline!

# Team Developer 7.0

- Destructor
  - Wie Constructors haben sie keinen Return Wert
  - Werden ausgelöst, wenn ein Objekt beendet wird (“destroyed”)
  - Kein seperater Aufruf nötig
  - Sinnvoll zum ‘Aufräumen’
    - Abbau eines File Handles
    - Abbau einer Datenbankverbindung
    - Referenzen zurücksetzen
    - ...

# Team Developer 7.0

- Function Overload (neu in TD 7.0)
  - Funktionen mit gleichen Namen, aber unterschiedlichen Parametern
  - Nicht unbedingt notwendig für OOP Programmierung, aber Overloading wird in vielen Sprachen unterstützt
    - Sinnvoll für Erweiterungen von Funktionen
      - *“Jetzt brauche ich einen Parameter mehr, aber die Funktionalität bleibt identisch”*

- Beispiele:

```
ShowText( String text )
```

```
ShowText( Number percent )
```

```
ShowText( String text, Number percent )
```

# Team Developer 7.0

- Function Overload (neu in TD 7.0)
  - Definierbar für interne Funktionen, Klassen Funktionen, Windows Funktionen
  - Duplikate werden nun mittels der Parameterliste der Funktion ermittelt
  - UDV werden als Parameter akzeptiert
  - Achtung mit Handles als Parametern:
    - String oder Datenfeld?
    - Number oder Checkbox?

```
♦ Function: Connect
  ♦ Description:
  ♦ Returns
  ♦ Parameters
    ♦ String: pConnectionString
    ♦ Number: pProviderType
    ♦ Boolean: pAutoCommit
  ♦ Static Variables
  ♦ Local variables
  ♦ Actions
    ♦ ! Connect to Database
♦ Function: Connect
  ♦ Description:
  ♦ Returns
  ♦ Parameters
    ♦ String: pConnectionString
    ♦ Number: pProviderType
  ♦ Static Variables
  ♦ Local variables
  ♦ Actions
    ♦ Call Connect(pConnectionString, pProviderType, FALSE)
♦ Function: Connect
  ♦ Description:
  ♦ Returns
  ♦ Parameters
    ♦ String: pConnectionString
  ♦ Static Variables
  ♦ Local variables
  ♦ Actions
    ♦ Call Connect(pConnectionString, DBP_PROVIDER_SQLBASE, FALSE)
```

# Team Developer 7.0

- Function Overload (neu in TD 7.0)
  - Was kann passieren?
    - ShowControlString(String str, WindowHandle hWnd) ??
    - ShowControlString(WindowHandle hWnd, String str) ??
    - Call ShowControlString(df1, df1) ??
  - Regel:
    - Zuerst der native Datentyp, dann der Handle
    - Links-nach-rechts Reihenfolge

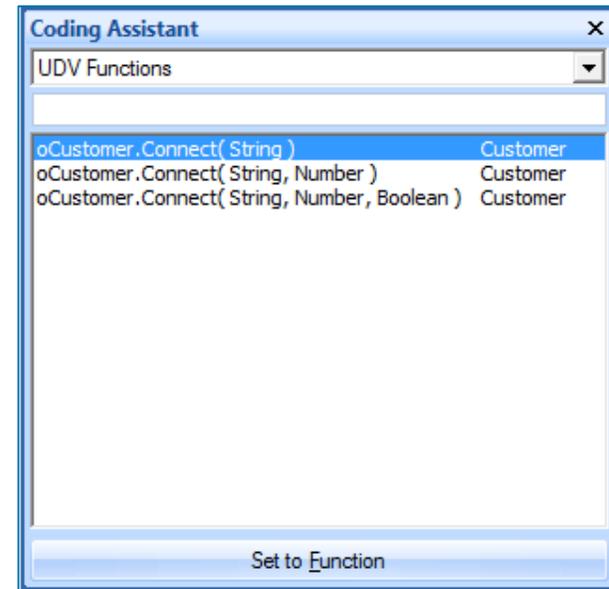


# Team Developer 7.0



- Function Overload (neu in TD 7.0)
  - ACA und Coding Assistant unterstützen Overloaded Functions

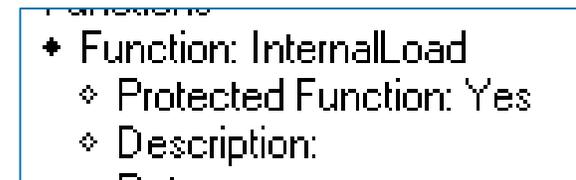
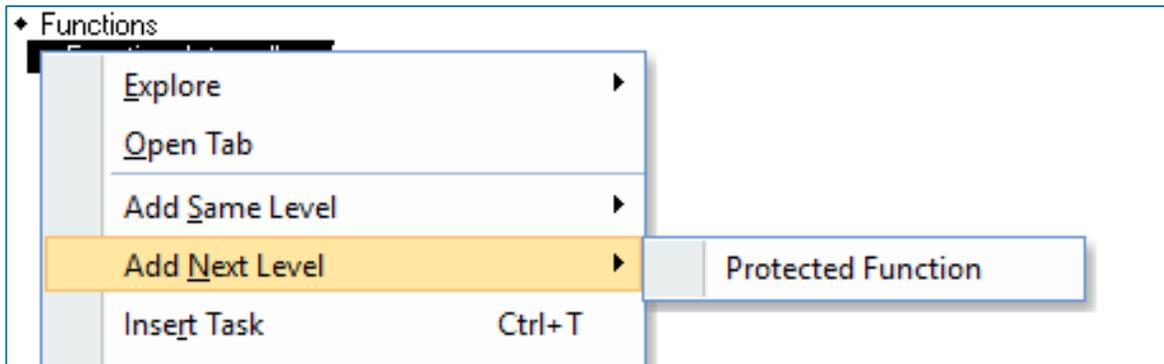
```
* On SAM_CreateComplete
  ◊ Call oCustomer.
    f0 Connect( String )
    f0 Connect( String, Number )
    f0 Connect( String, Number, Boolean )
    CustomerID
```



# Team Developer 7.0



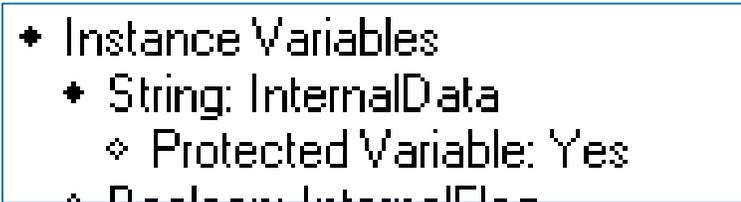
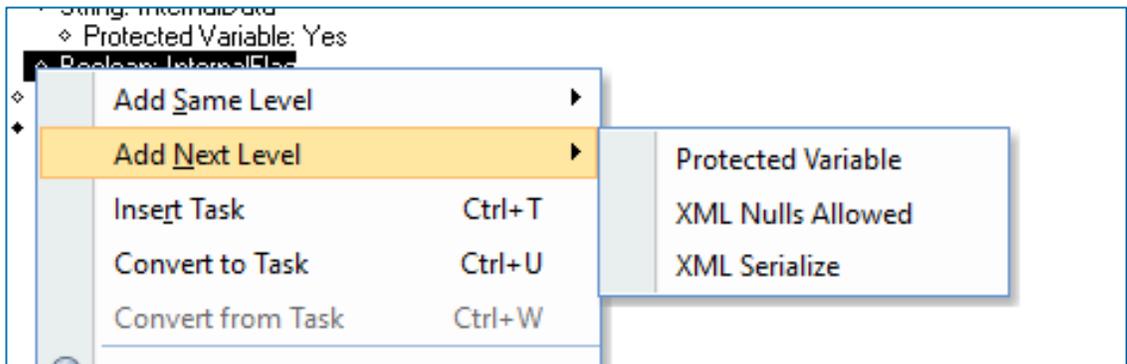
- Protected Functions & Variables (neu in TD 7.0)
  - Funktionen und Variablen als 'geschützt' markieren
  - Diese sind nur innerhalb der Klasse bzw. Sub-Klassen zur Verwendung freigegeben
  - Funktionen schützen:



# Team Developer 7.0



- Protected Functions & Variables (neu in TD 7.0)
  - Variablen schützen
    - Nur Instanz- & Klassenvariablen und Funktionen können geschützt werden
    - Kein Support für Windows Variablen und Funktionen
      - Support von GUI Klassen!
    - Der Coding Assistant zeigt, je nach Scope, die geschützten Variablen & Funktionen



# Team Developer 7.0

- Table <> Grid
  - Grid hat eine Vielzahl von neuen Funktionen:
    - Summary Bar
    - Grouping / Sorting / Filtering
    - Sub-Totals in Gruppen
    - Eingebautes Drucken
    - Eine Vielzahl von Spaltentypen
    - Eingabe / Ausgabe von Daten nach (XLSX, CSV, XML, Text)
    - Viele neue SalGrid API Funktionen

# Team Developer 7.0



- Table <> Grid
  - Aber:

“Das Grid ist ja so fürchterlich langsam!”

“Table: 10 Sekunden ↔ Grid: 3 Minuten”

# Team Developer 7.0



- Table <> Grid
  - Einige Änderungen in der Implementation des Grids wurden vorgenommen
  - Interner Code wurde optimiert
- Eine Testanwendung wurde erstellt

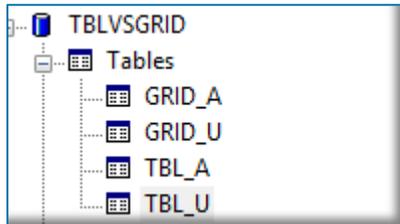
# Team Developer 7.0

- Testumgebung:
  - SQLBase 12.0 SP1, 32 Bit (64 Bit), nativer Router
  - 4 Tabellen
    - 2 Tabellen mit ANSI varchar Spalten
    - 2 Tabellen mit UNICODE nvarchar Spalten
    - 50.000 Datensätze je Tabelle, automatisch generiert
  - Team Developer
    - Version 6.3 SP2 Enterprise (auch mit 7.0 64-Bit getestet)
    - Test - Anwendung nutzt den 'RL' Isolation Level

# Team Developer 7.0



- Datenbank – Command Center (Grid)

A screenshot of the table structure for GRID\_A. The table is selected in the workspace tree. The table structure is shown in a grid with the following data:

Name	Data Type	Length	Scale	Nulls
ID	INTEGER	4	0	No
CHARDATAA	VARCHAR	55	0	No
CHARDATAB	VARCHAR	35	0	No
CHARDATAC	VARCHAR	15	0	No

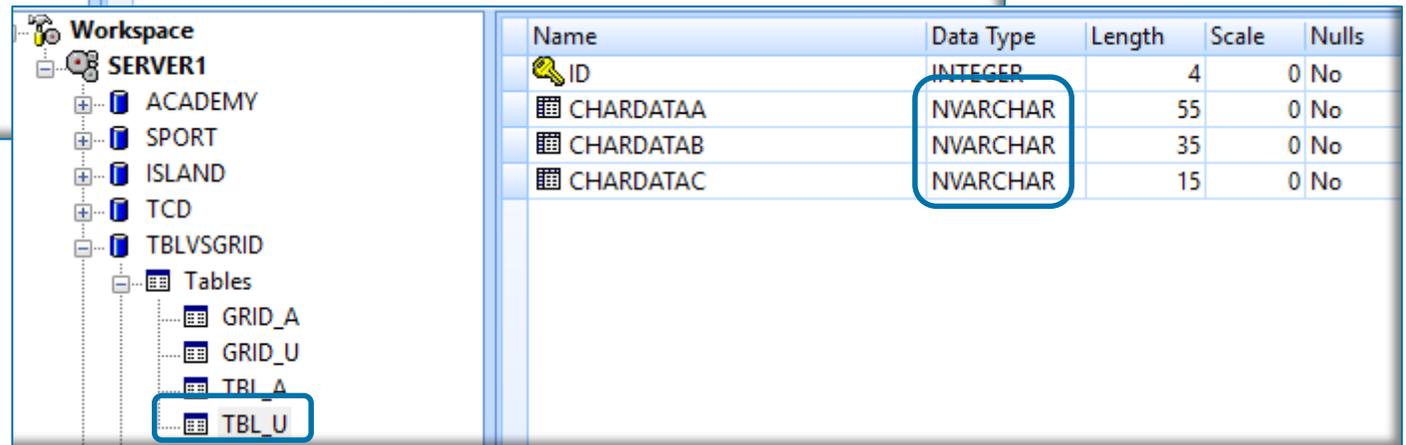
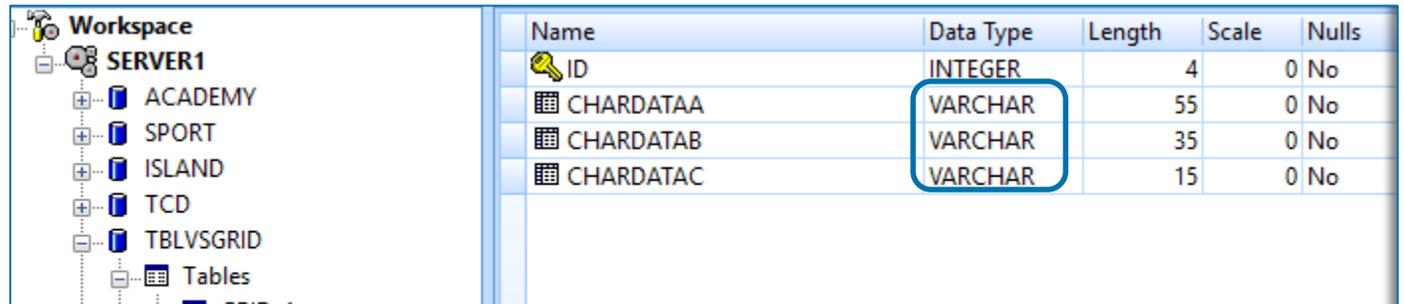
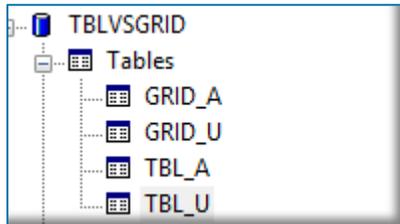
A screenshot of the table structure for GRID\_U. The table is selected in the workspace tree. The table structure is shown in a grid with the following data:

Name	Data Type	Length	Scale	Nulls
ID	INTEGER	4	0	No
CHARDATAA	NVARCHAR	55	0	No
CHARDATAB	NVARCHAR	35	0	No
CHARDATAC	NVARCHAR	15	0	No

# Team Developer 7.0



- Datenbank – Command Center (Table)



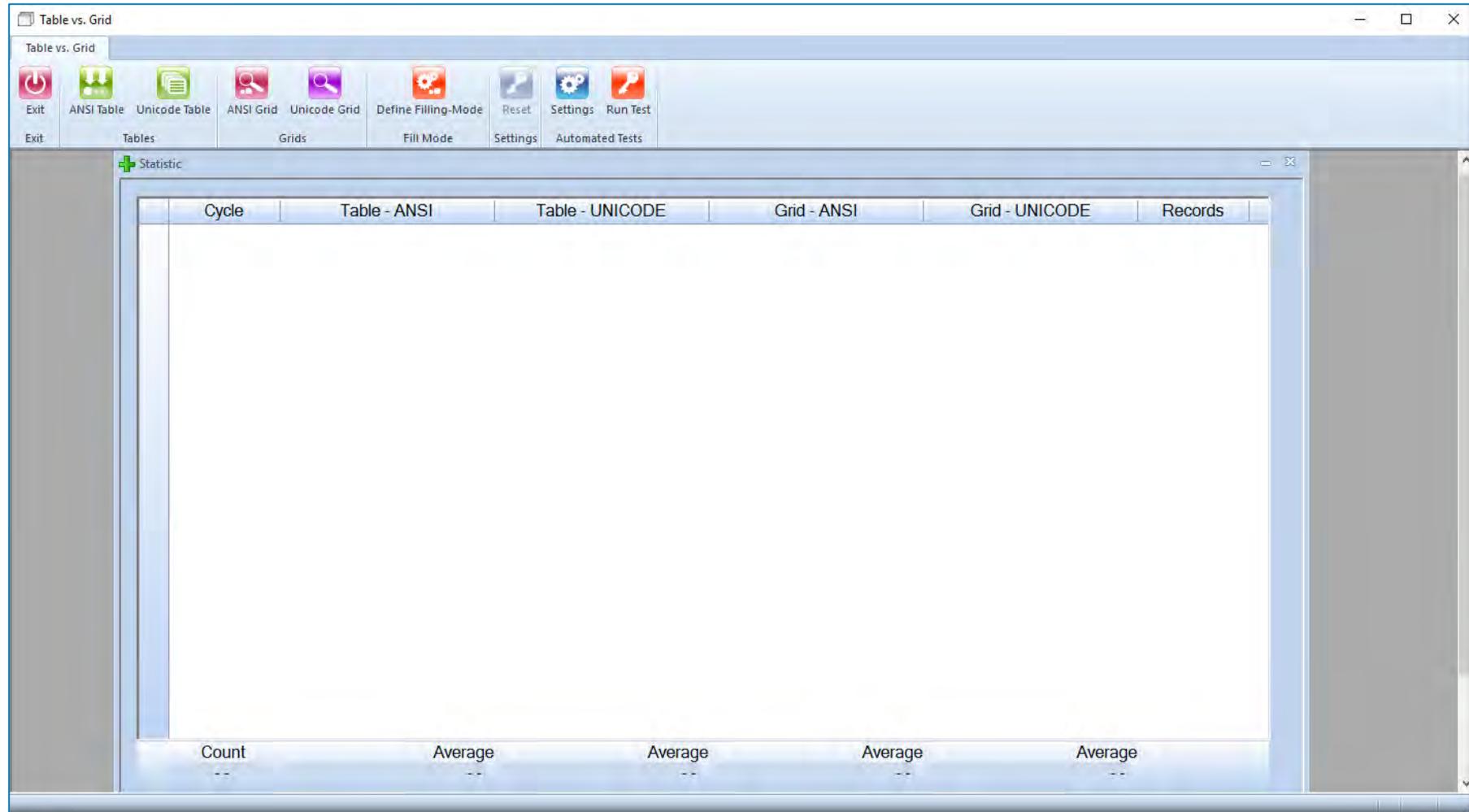
# Team Developer 7.0



- Datenbank – SQLTalk: Daten
  - Randomized generierte Daten (50.000 Records)
    - Separat für ANSI & UNICODE

ID	CHARDATAA	CHARDATAB	CHARDATAC
1	Zf\·^1ê Xóó&äªJà/J\~@y47[c€+~.<éÿ^-Mα1J@ñsÿe©SñP	óó&äªJà/J\~@y47[c€+~.<éÿ^-Mα1J	@ñsÿe©SñP
2	·påš' fll/æSqš'T2_lyAXt[P~D%%'·,*l@åmn,S?8z\wöR'ôä	Sqš'T2_lyAXt[P~D%%'·,*l@åmn,S	?8z\wöR'ôä
3	là-Y`æK>IàB2~0G< >e \l[ `µ;llñ-·1š`zp`< có< ö>{ž@{€f	B2~0G< >e \l[ `µ;llñ-·1š`zp`< có<	ö>{ž@{€f
4	.âêSc¶<8z¥è,ijá};©sn¥c;¶ê' >«)D£¢=-0~o-ç\$Aĵg;åwF	è,ijá};©sn¥c;¶ê' >«)D£¢=-0~o-ç	\$Aĵg;åwF
5	v·_AC·g>J,«zòe6â·èDm,αš" ñ"ôă@o2àat(; /TYCB>_mâ	«zòe6â·èDm,αš" ñ"ôă@o2àat(; /TYCB>_mâ	
6	~ ç' a90t,~lzvè-ò' Zh«r¥+·žjVô>™\k`ñ}·ªx~f.{š\;ll;nOô	lzvè-ò' Zh«r¥+·žjVô>™\k`ñ}·ªx~f	·{š\;ll;nOô

# Team Developer 7.0



# Team Developer 7.0

- Application
  - Nur eine Funktion für alle Datenbank Aktionen
  - Stateless connection zur Datenbank

- ◆ Function: fGetData
  - ◇ Description:
  - ◇ Returns
  - ◆ Parameters
    - ◇ Window Handle: hTable
    - ◇ Receive Number: nCount
    - ◇ Receive Date/Time: dStart
    - ◇ Receive Date/Time: dEnd
    - ◇ Receive Number: nDiff
  - ◇ Static Variables
  - ◆ Local variables
  - ◆ Actions

- ◇ ! Connect to the database
- ◆ If SqlConnection( hSQL )
  - ◇ ! Do something with DB
  - ◇ Call SqlSetIsolationLevel( hSQL, 'RL' )
  - ◆ If SqlPrepareAndExecute( hSQL, sSQL )
  - ◇ Call SqlDisconnect( hSQL )
- ◇ ! Disconnect

# Team Developer 7.0

- SQL-Statements innerhalb der Funktion:
  - Randomized 'Order by'

```
◆ If nMode = 4
  ◆ If nRandomOrder = 0
    ◇ Set sSQL = 'select ID, CharDataA, CharDataB, CharDataC from Grid_U
                into :nCompID, :sCompany_Name, :sAddress , :sCity '
  ◆ Else
    ◆ If nOrder = 1
      ◇ Set sSQL = 'select ID, CharDataA, CharDataB, CharDataC from Grid_U
                  into :nCompID, :sCompany_Name, :sAddress , :sCity order by CharDataC'
    ◆ If nOrder = 2
      ◇ Set sSQL = 'select ID, CharDataA, CharDataB, CharDataC from Grid_U
                  into :nCompID, :sCompany_Name, :sAddress , :sCity order by CharDataB, CharDataC'
```

# Team Developer 7.0

- SQL-Statement in der Funktion
  - Schrittweises auslesen des Result-Set:

```
◇ ! Connect to the database
◆ If SqlConnect( hSQL )
  ◇ ! Do something with DB
  ◆ If SqlPrepareAndExecute( hSQL, sSQL )
    ◆ While SqlFetchNext( hSQL, nRet )
      ◇ Call SalTblInsertRow( hTable, TBL_MaxRow )
      ◇ Set hTable.colID = nCompID
      ◇ Set hTable.colName = sCompany_Name
      ◇ Set hTable.colAddress= sAddress
      ◇ Set hTable.colCity= sCity
      ◇ Set nCount=nCount+1
      ◆ If nRec = nCount
        ◇ Break
      ◇ Call SqlDisconnect( hSQL )
  ◇ ! Disconnect
```

# Team Developer 7.0



- Anzeige der Daten

- Table mit ANSI Daten und 'Order by'

Table with ANSI Data

Go No of Records: 50 Start 14:23:01:520000 End 14:23:01:890000 Diff: 0,42824074074 Close

ID	Name	Address	City
31550	-v1&px~gl,¶,D\DTTádµ™8¥ }},NŸiVO9^\$\$€	@~i,¶,D\DTTádµ™8¥ }},NŸiVO9^	\$\$8lròq%ä,,
13099	†SU{T=X@ié{&"w"[náŸ\$Ne~]7\$Hžghò)q@	{&"w"[náŸ\$Ne~]7\$Hžghò)q@š-»	\$\$gòâµZP.¢
7078	Wê,@Ž«Fé9/á@†Š/¢Bæ³<áv %,,` 8°œ=llk(	/á@†Š/¢Bæ³<áv %,,` 8°œ=llkQ>u0	\$\$n)qD-Dµ;
36997	^GXqŸ,Fd™æ8mSdâ)Ufçã†b(¢Ud>-N@Š	æ8mSdâ)Ufçã†b(¢Ud>-N@Š1@PŽ3«	\$\$¥=²žò\$€
23115	êP.çŸ"wâl*†@L@J"©¢¢éâjs"æ¢l8àð71O,,c	†@L@J"©¢¢éâjs"æ¢l8àð71O,,c'n)c	%%éæp\$)eG
40877	Ž->{aàaj8)Sâ&U¥\$AH'œw'õ'qé<=?æ+†m:	)Sâ&U¥\$AH'œw'õ'qé<=?æ+†m=sEð	\$'©sp1N«Ÿæ
49653	—ð8_0KxŸr_â†æœ«o¶Ÿ*ikö7V,«"„_fh„l¶¶	Ÿr_â†æœ«o¶Ÿ*ikö7V,«"„_fh„l¶¶	\$'-F,,ÇeyWyQ
4752	r"«J7-XšŽ¥mQz_„†EK„,œYª@ð{áW[äim	Ž¥mQz_„†EK„,œYª@ð{áW[äim—lP†	\$(DX£=cFJ
26443	w1NŽ«/«@C0«,CE>CE«COeFKönváS_2ct	«,CE>CE«COeFKönváS_2cbã'ð{h[r?Ÿ	)\$äv3j Wðá
18634	8µ-6Ž%jäl,âb«uñoH?<énD^%š=é+-ò5u4	läL,âb«uñoH?<énD^%š=é+-ò5u4	\$*"Kq",g3V
24439	EhO'F9tê+,Jb}dzgâ567@,«2Q"â- v9æYçl	Jb}dzgâ567@,«2Q"â- v9æYçlA â	\$*@,,1*Wð
30670	fN<.PA_£Gq'O-hcBFhGðyàNrBN\$.ŽT6Le :	'O-hcBFhGðyàNrBN\$.ŽT6Le æiŽY	+\$œlŽX)•l%
16072	EñpW)GcCAKµç9*¶-ž%-Q):'4'ft,Jk<ò@(\$	µç9*¶-ž%-Q):'4'ft,Jk<ò@(\$,+	,\$,éòáO«Ž-
13728	00.>æ0@Wð A†=ðáæKb:-^^ 8O..bâçfZê>	A†=ðáæKb:-^^ 8O..bâçfZê>2ñ óà\	\$.á8â-,>
31757	`A,'k-9èF«†H©V9?3*g1`=f-†+ŠéULð'&*U	H©V9?3*g1`=f-†+ŠéULð'&*UMDx	\$1éàòèŸHd
38560	NwbdftIT<Hkæ-"š. ?6[-(â³@4>h: UŽ{u^-M>»	Hkæ-"š. ?6[-(â³@4>h: UŽ{u^-M>»o	\$2P~^,†f(
24625	8ó[M£ofBqSG""ž,r.]hu¥œ-céqll:2HrlQ="è	SG""ž,r.]hu¥œ-céqll:2HrlQ="èšN	\$2t6oGž,-
30108	©²âc†*+d*]èðOäæ,à†µž-Rt~tty,ðášw)Eg,	èðOäæ,à†µž-Rt~tty,ðášw)EgAâ?ç	\$3V_áµW=
41562	,,mœ4C,l£ð5bG@«(x\$èqeVZU8ycfb,è4¶	ð5bG@«(x\$èqeVZU8ycfb,è4¶rWbæ-	\$3oá²àZ0µ
5119	L³nB²â!Tœç1â³d;"™BLCr®•kD'HAT†0è, 6	1â³d;"™BLCr®•kD'HAT†0è, 6Fh	\$3á9y«r)k/
18531	â é}3f8SPzDa8QêfVA6^êF5_„ä„Ž'+•QKW	zDa8QêfVA6^êF5_„ä„Ž'+•QKWz9Y_	\$4>/£l3çX
45327	6-df*Nð2KY:}•xj81«µ/uq&ð;=•6ltJ+8l*\$4	Y:}•xj81«µ/uq&ð;=•6ltJ+8l*	\$4Hc;«òCE â
21635	j-Tj/A—çnOh{£P(µrñOŠm„)%Cæ{ázð@<8%*	çnOh{£P(µrñOŠm„)%Cæ{ázð@<8%*	\$4éŸ>âM6(4
43526	gGl†aMbZ*)Ÿeó£S,-)F,,çJAt 'µf f,š>Aâ*é(Š	Ÿeó£S,-)F,,çJAt 'µf f,š>Aâ*é(Š	\$6@qæ,,Gá<



# Team Developer 7.0

OpenText  
Roadshow  
2016

- Statistik:

Cycle	Table - ANSI	Table - UNICODE	Grid - ANSI	Grid - UNICODE	Records
1	0,4282	0,3241	0,1273	0,3009	50

Count	Average	Average	Average	Average
1	0,4282	0,3241	0,1273	0,3009

Export to Excel   Print   Chart   Reset Grid   Load Sample Data

# Test Environment



- Einstellungen für den automatisierten Test:

The 'Setup Automated Test' dialog box contains the following settings:

- Set Test Parameter**
  - Number of records:** 50 Records (selected), 500 Records, 5000 Records
  - Number of Cycles:** 10 Cycles (selected), 50 Cycles, 100 Cycles
  - Randommized Sequenz:**  Randomized
  - Randomized 'Order by':**  Random 'Order by'
  - Test for:**  Table with ANSI,  Table with UNICODE,  Grid with ANSI,  Grid with UNICODE
  - Timing:** No of seconds between each sequence: 5
  - Save Results:**  Save as XLS file, 4143433.xls,

The 'Fill Mode' dialog box contains the following settings:

- Table Filling Mode**
  - TBL\_FillAll
  - TBL\_FillAllBackground
  - TBL\_FillNormal

# Team Developer 7.0

- Automatisierter Test:
  - 1ste Ergebnisse:

Automated Test

**Parameter**

Number of Records in Table/Grid:	50	Save Results:	Yes
Number of Cycles:	10	Excel Document:	4114913.xls
Randomized Sequenz:	Yes	Table / Grid?:	TA-TU-GA-GU
Randomized 'Order By':	Yes	Seconds between run:	5

**Actual Test run**

Start	14:42:26:330000	Action:	Batch Startet
End:	14:43:04:500000		With randomized Sequenz: Yes
Diff:	44,1782		With >order by<: Yes
			Random: TU
			Random: TA
			Random: TU
			Random: GU
			Random: GA
			Random: GU

Start Test

Close

# Team Developer 7.0



Statistic

	Cycle	Table - ANSI	Table - UNICODE	Grid - ANSI	Grid - UNICODE	Records
➔	1	0,1389	0,3241	0,1273	0,3125	50
➔	2	0,1736	0,3704	0,1620	0,3588	50
➔	3	0,1389	0,3241	0,1505	0,3125	50
➔	4	0,1505	0,3356	0,1273	0,3125	50
➔	5	0,1273	0,3241	0,1389	0,3125	50
➔	6	0,1389	0,3125	0,1273	0,3009	50
➔	7	0,1736	0,3125	0,1389	0,3125	50
➔	8	0,1505	0,3241	0,1273	0,3009	50
➔	9	0,1389	0,3819	0,1273	0,3241	50
➔	10	0,1389	0,3125	0,1273	0,3125	50

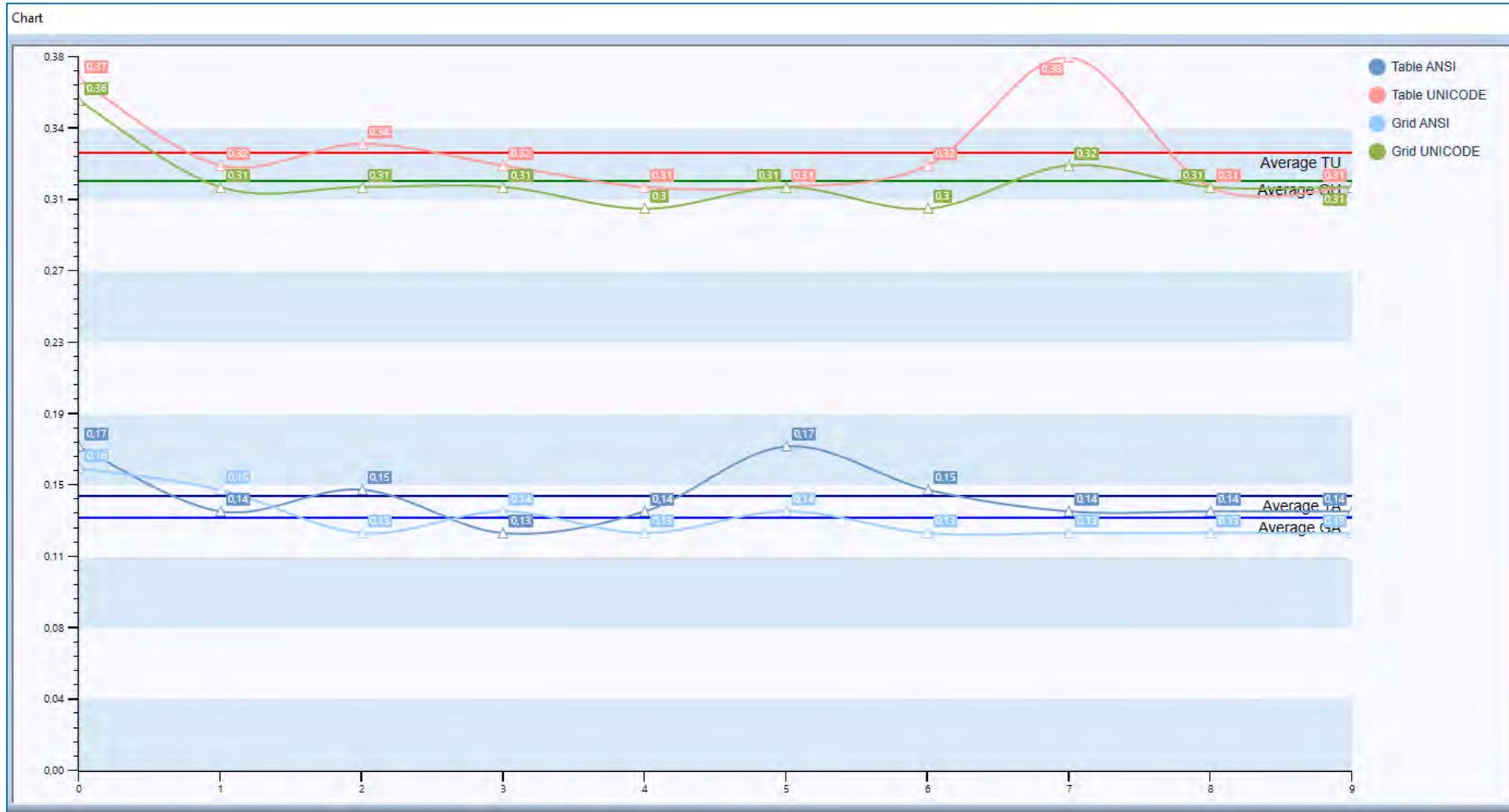
  

Count	Average	Average	Average	Average
10	0,1470	0,3322	0,1354	0,3160

Export to Excel   Print   Chart   Reset Grid   Load Sample Data

# Team Developer 7.0

OpenText  
Roadshow  
2016



# Test Result (5000 Records)



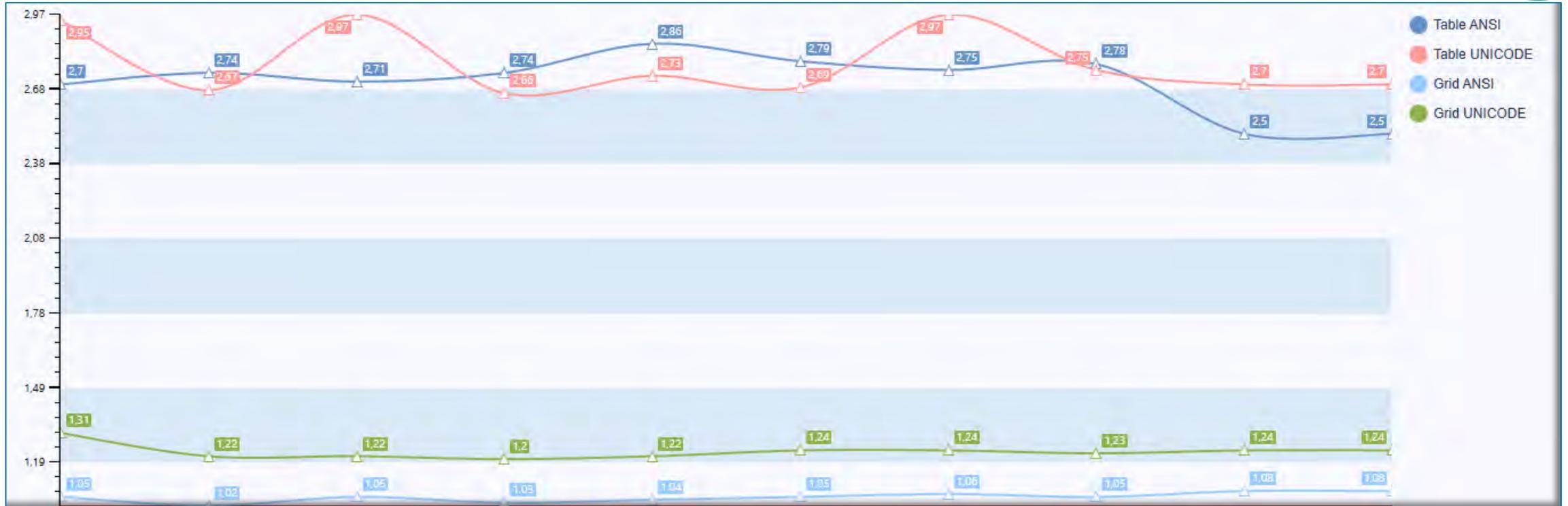
	Cycle	Table - ANSI	Table - UNICODE	Grid - ANSI	Grid - UNICODE	Records
→	1	2,7199	2,9861	1,0185	1,2269	5000
→	2	2,6968	2,9514	1,0532	1,3079	5000
→	3	2,7431	2,6736	1,0185	1,2153	5000
→	4	2,7083	2,9745	1,0532	1,2153	5000
→	5	2,7431	2,6620	1,0301	1,2037	5000
→	6	2,8588	2,7315	1,0417	1,2153	5000
→	7	2,7894	2,6852	1,0532	1,2384	5000
→	8	2,7546	2,9745	1,0648	1,2384	5000
→	9	2,7778	2,7546	1,0532	1,2269	5000
→	10	2,5000	2,6968	1,0764	1,2384	5000

Count	Average	Average	Average	Average
10	2,7292	2,8090	1,0463	1,2327

Export to Excel   Print   Chart   Reset Grid   Load Sample Data

# Test Result (5000 Records)



# Auswertung

- Nahezu gleiche Performance
  - SQL – Statements haben die gleiche Komplexität
  - SQL – Statements werden von der Datenbank abgearbeitet
    - Es wird kein Buffer genutzt!
  - Gleiche Funktion für alle Tests
  - Nur die Differenz der Werte zwischen Table und Grid dürfen berücksichtigt werden!
  - Wichtig:
    - Das Result-Set wird in einer Schleife abgearbeitet
    - Spaltentypen (Bilder)
    - Anzahl der Spalten

# Auswertung



- Unterschiede: Table & Grid:

Records	Table ANSI	Grid ANSI	Diff	Table UNICODE	Grid UNICODE	Diff
50	0,1459	0,1261	0,0198	0,3264	0,3091	0,0174
500	0,3912	0,2187	0,1725	0,5625	0,3981	0,1644
5000	2,7292	1,0463	1,6829	2,8091	1,2327	1,5763
			Numbers in Seconds			

- Die Netto-Performance des Grid ist besser als die des Tables!
  - Für diesen Test!!

# Auswertung



- Die Performance ist immer noch 'schlecht'?
  - SQLBase: 'RCx' Isolation Level ausprobieren (wenn möglich)
  - Check SQL Statements / Indexes
  - Suche nach dem Flaschenhals in der Anwendung
    - Was passiert beim fetchen durch den Result-Set?
  - Gleicher Code wie beim Table?
  - Werden die Testergebnisse richtig interpretiert?
  - Letztes Service Pack von TD6.3 installiert?
  - OpenText PS & MD bietet Consulting an, um die 'letzten Sekunden' zu finden
  - Gibt es einen Grund um 1000+ Zeilen im Table/Grid darzustellen?
  - Anzahl der (non numeric) Spalten ist ebenfalls ausschlaggebend!

# Anderes Beispiel



- Anzeige in Millisekunden (10 Zeilen; sehr viele DB Zugriffe)

2007/2008	9,01	645	50	204	291	116	24	8	1	0	1	0	0	135,2	270,3	31,6%
2008/2009	9,29	930	45	447	337	121	22	2	0	1	0	0	0	139,4	278,7	48,1%
2009/2010	9,40	1315	60	670	516	114	15	0	0	0	0	0	0	141,0	282,0	51,0%
2010/2011	9,58	1312	68	819	455	36	0	0	0	0	0	0	0	143,7	287,4	62,4%
2011/2012	9,68	1435	65	993	426	16	0	0	0	0	0	0	0	145,2	290,4	69,2%
2012/2013	9,76	1285	66	980	298	7	0	0	0	0	0	0	0	146,4	292,8	76,3%
2013/2014	9,76	1450	66	1117	323	9	0	0	0	0	0	0	0	146,4	292,8	77,0%
2014/2015	9,79	840	49	665	173	2	0	0	0	0	0	0	0	146,9	293,7	79,2%
2015/2016	9,79	1245	69	990	244	11	0	0	0	0	0	0	0	146,9	293,7	79,5%
2016/2017	9,76	405	20	311	92	2	0	0	0	0	0	0	0	146,4	292,8	76,8%

Viele Berechnungen für die Zellen & Farbänderungen

# Anderes Beispiel



- Angezeigt in 5 Sekunden (20 Zeilen, nur 1 DB Zugriff):

670	20-10-2016	1	294	30	9,80	98,00	●	0	Anschütz 8002 S2	8
671	20-10-2016	2	148	15	9,87	98,67	●	0	Anschütz 8002 S2	8
668	06-10-2016	1	287	30	9,57	95,67	●	0	Anschütz 8002 S2	8
669	06-10-2016	2	145	15	9,67	96,67	●	0	Anschütz 8002 S2	8
666	04-10-2016	1	296	30	9,87	98,67	●	0	Anschütz 8002 S2	8
667	04-10-2016	2	147	15	9,80	98,00	●	0	Anschütz 8002 S2	8
664	27-09-2016	1	292	30	9,73	97,33	●	0	Anschütz 8002 S2	8
665	27-09-2016	2	148	15	9,87	98,67	●	0	Anschütz 8002 S2	8
662	22-09-2016	1	292	30	9,73	97,33	●	0	Anschütz 8002 S2	8
663	22-09-2016	2	144	15	9,60	96,00	●	0	Anschütz 8002 S2	8
660	20-09-2016	1	291	30	9,70	97,00	●	0	Anschütz 8002 S2	8
661	20-09-2016	2	146	15	9,73	97,33	●	0	Anschütz 8002 S2	8
658	13-09-2016	1	290	30	9,67	96,67	●	0	Anschütz 8002 S2	8
659	13-09-2016	2	146	15	9,73	97,33	●	0	Anschütz 8002 S2	8
656	30-08-2016	1	292	30	9,73	97,33	●	0	Anschütz 8002 S2	7
657	30-08-2016	2	148	15	9,87	98,67	●	0	Anschütz 8002 S2	7
654	23-08-2016	1	294	30	9,80	98,00	●	0	Anschütz 8002 S2	8

File IO !!!

# Team Developer 7.0



- Änderung seit der DevCon:
  - Aber!
    - “Alternate Background Color macht das Grid langsam”
    - “Schließen des Grids dauert ewig”
  - Die Testanwendung wurde entsprechend erweitert.
  - Keine Performance - Einbußen zu ermitteln!



# OPENTEXT™

[www.opentext.com](http://www.opentext.com)



[twitter.com/opentext](https://twitter.com/opentext)



[facebook.com/opentext](https://facebook.com/opentext)



[linkedin.com/company/opentext](https://linkedin.com/company/opentext)