



# Was EF Core Entwickler über SQL Server-Datenbanken wissen sollten



# Meine Person- Thorsten Kansy

Freier Consultant, Software Architekt,  
Entwickler, Trainer & Fachautor



Azure Cosmos DB



# Mein Service- Ihr Benefit

- Individuelle Trainings
- Projektbegleitung
- Beratung
  - Problemanalyse und Lösungen
  - Technologieentscheidungen



# Agenda

- Allgemeines
- Der große Abgrund
- Datentypen
- Collation
- Wie Daten organisiert werden
- Sichten, Prozeduren & Co
- Indizes-Mythen



# Agenda II

- Das Multi-User-Dilemma
- Das Abfrage-Dilemma
- Das Index-Dilemma



A photograph of the Mausoleum of the Prophet in Medina, featuring a large dome and an arched entrance, with a crowd of people in the foreground. The building is constructed from light-colored stone blocks. The central entrance is a large, ornate archway. A large, light-colored dome sits atop the central part of the building. In the foreground, a large group of people, including men in traditional white thobes and women in headscarves, are gathered on a paved plaza with a geometric tile pattern. The scene is set outdoors with trees visible on the left and right sides.

# Allgemeines

# Allgemeines

Server und Client haben nur sporadische Verbindung.

Große Datenbanken sind nicht automatisch langsam,  
kleine nicht schnell.

Bessere Hardware bedeutet nicht automatisch bessere Performance.

Optimierung ist nicht unendlich möglich.

Der (SQL) Server ist sprach-neutral.

Und nein, an schlechter Performance ist *nicht* immer der SQL Server Schuld





# Der große Abgrund



# Der große Abgrund

Zugriffe auf den Server sind langsam!

1.000 Milisekunden (ms)

1.000.000 Mikrosekunden ( $\mu$ s)

1.000.000.000 Nanosekunden (ns)

RAM (ns) > SSD ( $\mu$ s) > HDD(ms) > Server (ms)





# Temporal Tables



# Temporal Tables- Automatische Historisierung

```
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'];
```



# IsTemporal (Fluent API)

- Temporal table (SQL Server)
- Standard: -

```
modelBuilder.Entity<MyTemporalTable>()  
    .ToTable(t => t.IsTemporal(hist =>  
        {  
            hist.UseHistoryTable("SessionHistory");  
        }  
    );
```

```
dbContext.MyTemporalTable  
    .TemporalAsOf(<Date>).Where(w => ...);
```

# Temporal AsOf & Co

TemporalAll()	Alle Daten aus der Tabelle und der historischen Tabelle.
TemporalAsOf(...)	Daten zu einem bestimmten Zeitpunkt.
TemporalFromTo(...)	Daten aus der Tabelle plus der historischen Tabelle exk. Einschluss der oberen Grenze.
TemporalBetween(...)	Daten aus der Tabelle plus der historischen Tabelle inkl. Einschluss der oberen Grenze.
TemporalContainedIn(...)	Nur Daten aus den der historischen Tabelle aus dem Zeitraum.



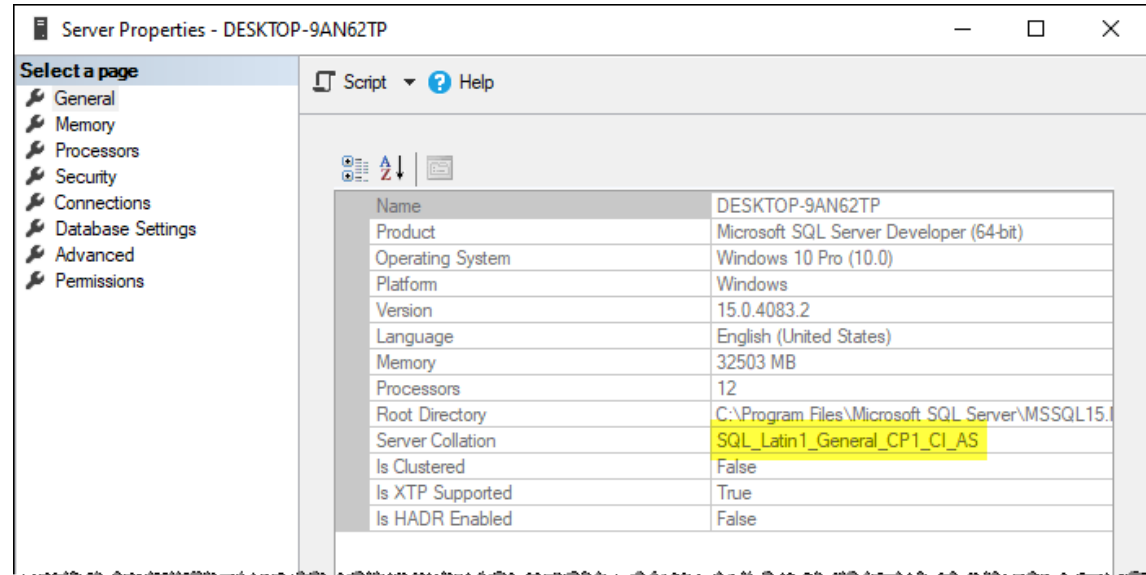
Collation



# Collation

Groß-/ Kleinschreibung

Sortierung



Server Properties - DESKTOP-9AN62TP

Select a page

- General
- Memory
- Processors
- Security
- Connections
- Database Settings
- Advanced
- Permissions

Script Help

Name	Value
Name	DESKTOP-9AN62TP
Product	Microsoft SQL Server Developer (64-bit)
Operating System	Windows 10 Pro (10.0)
Platform	Windows
Version	15.0.4083.2
Language	English (United States)
Memory	32503 MB
Processors	12
Root Directory	C:\Program Files\Microsoft SQL Server\MSSQL15.I
Server Collation	SQL_Latin1_General_CP1_CI_AS
Is Clustered	False
Is XTP Supported	True
Is HADR Enabled	False

CI Case insensitive / CS Case sensitive

AI Accent insensitive / AS Accent sensitive





# Sichten, Prozeduren & Co.



# Sichten, Prozeduren & Co.

SQL Server kann viel mehr als nur Tabellen

- Sichten
- Stored Procedures
- Funktionen
- Trigger
- Volltextsuche
- ...





# Das Multi-User-Dilemma

# Das Multi-User-Dilemma

Viele “chaotische” Zugriffe geschehen parallel und müssen vor anderen Zugriffen abgegrenzt (“geschützt”) werden. Dies erfordert Sperren und verursacht längere Zugriffszeiten.

## SQL Server

- blockt Ressourcen während der Zugriffe
- Ressourcen (Row, Pages, Extends, Tabellen, Datenbanken, ...)
- Zugriffe (Exklusiv, Shared, ...)
- Zugriffe können koexistieren oder auch eben nicht (=> Locks)



# Optimistic Concurrency

Es gibt keinen Mechanismus, um Daten zu “sperrern” in Form einer Zuweisungen an einen User (wie z.B. das Auschecken in SharePoint)

Der Client kennt nie mit 100% Sicherheit, wie die Daten auf dem SQL Server aussehen.

Und Daten können “gleichzeitig” verändert werden.





# Indizes-Mythen

# Indizes-Mythen

Indizes sind optional.

Je mehr Indizes, desto besser (“Viel hilft viel”)

Indizes sind nicht statisch- einmal konzipieren reicht nicht!

...



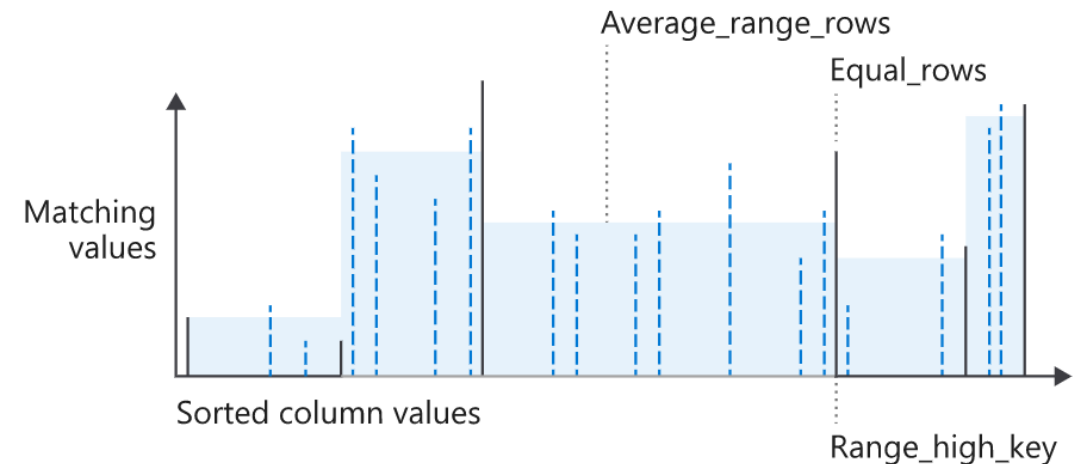


# Das Index-Dilemma

# Das Index-Dilemma

Der SQL Server muss möglichst schnell entscheiden, ob und welchen Index er verwendet und das kostet Zeit.

Histogramme geben Auskunft über die Verteilung der Daten





# Fragen? Jetzt oder später!



[www.dotnetconsulting.eu](http://www.dotnetconsulting.eu)



# Links



[www.dotnetconsulting.eu](http://www.dotnetconsulting.eu)



[@Tkansy](https://twitter.com/Tkansy)



[tkansy@dotnetconsulting.eu](mailto:tkansy@dotnetconsulting.eu)