



Was Entwickler über SQL Server-Datenbanken wissen müssen



Thorsten Kansy (tkansy@dotnetconsulting.eu)

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
- Snapshots
- Best-of Features



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 (μ s)

1.000.000.000 Nanosekunden (ns)

RAM (ns) > SSD (μ s) > HDD(ms) > Server (ms)



A scenic landscape featuring a wide, unpaved dirt road that stretches from the foreground into the distance. The road is flanked by lush green grass. In the background, there are rolling green hills under a bright blue sky filled with large, white, fluffy clouds. A dark blue horizontal band is superimposed over the middle of the image, containing the title text.

Datentypen

Datentypen

CLR und SQL Server Datentypen
sind nicht 100% kompatibel

SQL Server	.NET
bigint	long
binary	byte[]
bit	bool
char	string
date	DateTime
datetime	DateTime
datetime2	DateTime
datetimeoffset	DateTimeOffset
decimal	decimal
filestream	byte[]
float	double
geography	Microsoft.SqlServer.Types.SqlGeography
geometry	Microsoft.SqlServer.Types.SqlGeometry
hierarchyid	Microsoft.SqlServer.Types.SqlHierarchyId
image	byte[]
int	int
money	decimal
nchar	string
ntext	string
numeric	decimal
nvarchar	string
real	Single
rowversion	byte[]
smalldatetime	DateTime
smallint	short
smallmoney	decimal
sql_variant	object
text	string
time	TimeSpan
timestamp	byte[]
tinyint	byte
uniqueidentifier	Guid
varbinary	byte[]
varchar	string
xml	string
else	object



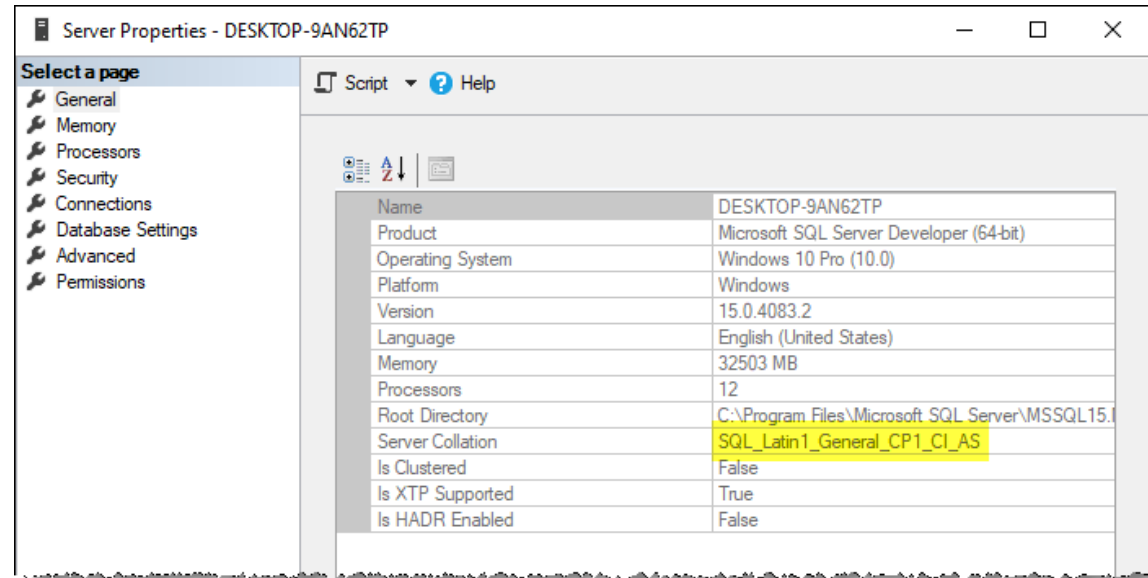


Collation



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





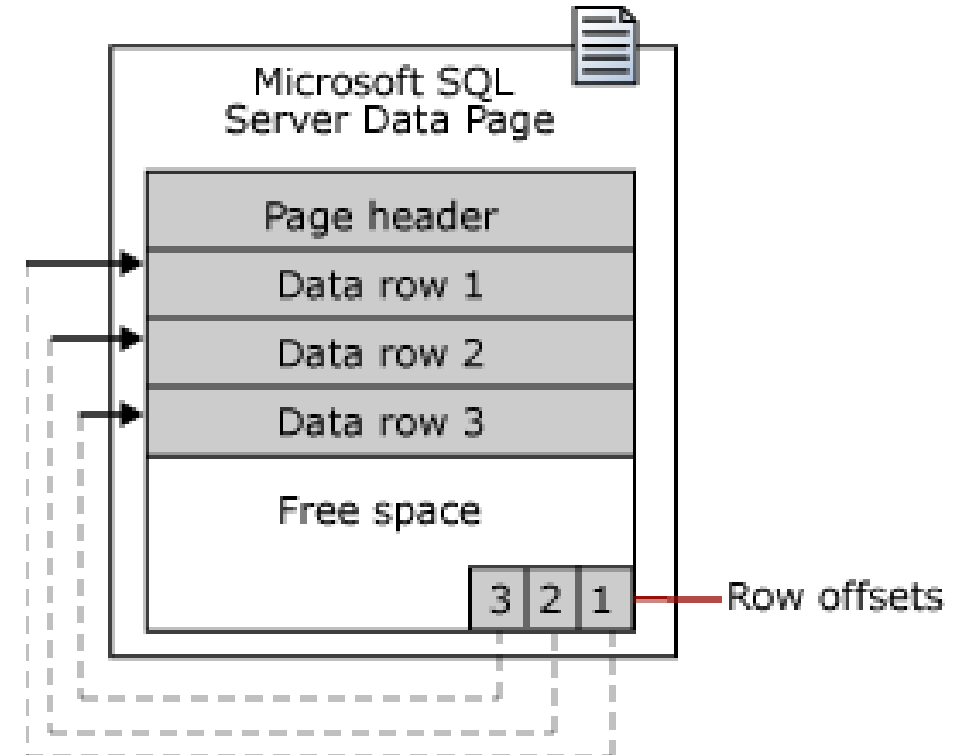
Wie Daten organisiert werden



Wie Daten organisiert werden

Pages (8k) -> Extents (64kb)

8.060 Bytes Nutzlast pro Page





Sichten, Prozeduren & Co.



Sichten, Prozeduren & Co.

SQL Server kann viel mehr als nur Tabellen

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



Indizes-Mythen

A photograph of a large, white, classical-style building with a central dome and clock tower, surrounded by a green lawn and trees. A blue semi-transparent banner is overlaid across the middle of the image, containing the text 'Indizes-Mythen'.

Indizes-Mythen

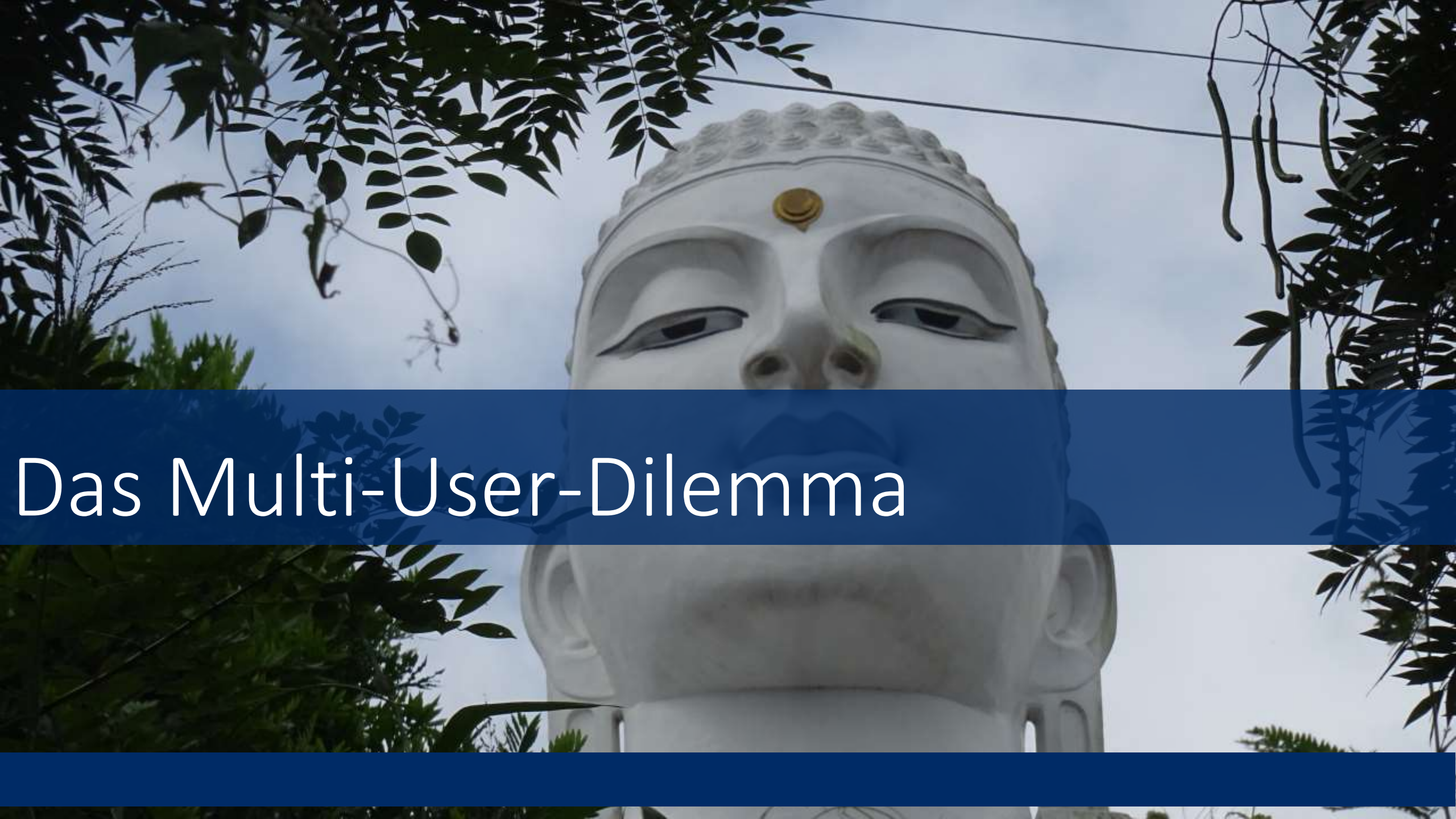
Indizes sind optional.

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

Indizes sind nicht statisch- einmal konzipieren reicht nicht!

...





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)



The image is a wide landscape photograph. The top half shows a bright blue sky with scattered white and grey clouds. Below the sky is a dark, calm body of water, likely the sea, which stretches to a flat horizon. On the horizon, there is a small, dark, rectangular object that looks like a boat or a small structure, and a tiny human figure standing nearby. The bottom half of the image shows a field of dry, brownish-yellow grass with patches of green. A solid dark blue horizontal bar is at the very bottom of the frame.

Das Abfrage-Dilemma

Das Abfrage-Dilemma

“Starre”, immer wiederkehrende Abfragen sind für die Optimierung optimal. Viele Anwendungen möchte aber eher “flexible” Abfragen durchführen

Flexibel in

- Zeit
- Inhalt



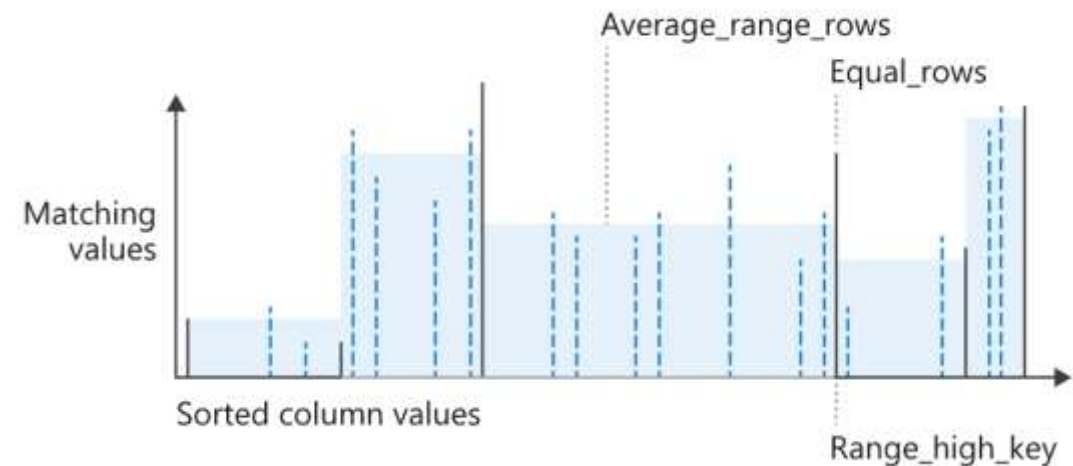
The background image shows a Jain temple complex, likely the Dilwara Temples in Rajasthan, India. It features a series of ornate, white marble temples with intricate carvings and domes, situated on a hill. A prominent temple with a tiered roof is visible on the right. The foreground is a dry, rocky slope. A semi-transparent blue horizontal band is overlaid across the middle of the image, containing the title text.

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





Best-of Features

Snapshot

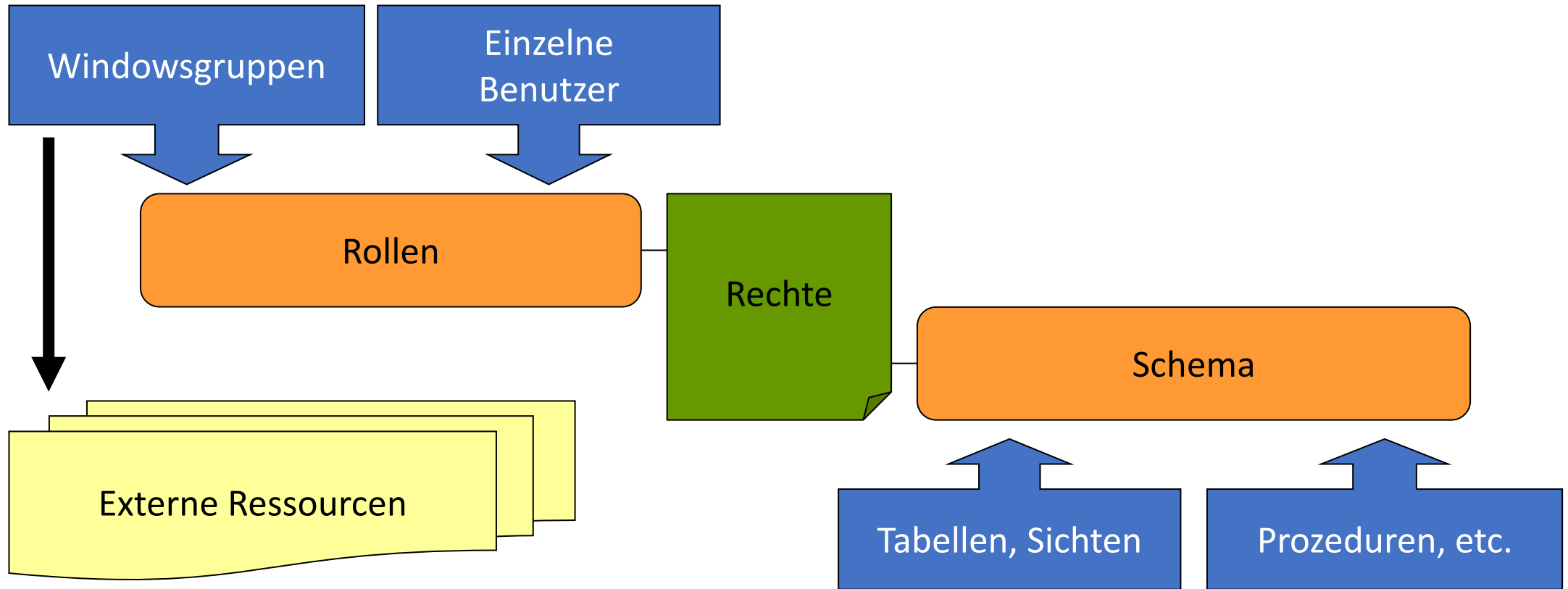
Datenbanken „einfrieren“ und wiederherstellen- schneller als Backups, aber nicht eigenständig.

```
CREATE DATABASE AdventureWorksSnapshot ON ( ... )  
AS SNAPSHOT OF AdventureWorks;
```

```
ALTER DATABASE AdventureWorks SET SINGLE_USER WITH ROLLBACK IMMEDIATE;  
RESTORE DATABASE AdventureWorks FROM DATABASE_SNAPSHOT =  
'AdventureWorksSnapshot';
```



Sicherheit - Best Practices



Row Level Security

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;
```



Temporal Table

Automatisch 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'];
```



Fragen? Jetzt oder später!



www.dotnetconsulting.eu



Links



www.dotnetconsulting.eu



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



tkansy@dotnetconsulting.eu