

# **Tabellenvariable, Temp. Tabellen, Views/Sichten**



## Problemstellung

- Viele Suchvorgänge wiederholen sich
- Table Scans notwendig, hohe Kosten der Suche

## ➤ Lösungsansätze

Sichten/Views

temporäre Tabellen: lokal, global

Tabellenvariable, Tabellentypen

CTE, Common Table Expressions

Tabellenwertige Inline-Funktionen

Indizierung, autom. Optimierung

## Einführung: View = Sicht

### Gespeicherter SELECT

- für differenzierte Rechtevergabe
- Basiert auf Sichten und/oder Tabellen
- Typen: Regulär, indiziert, partitioniert
- Teil der Datenbank (MDF)

## Syntax

- CREATE VIEW <name> [WITH ENCRYPTION] [WITH SCHEMABINDING] AS SELECT ... [WITH CHECK OPTION]
- Encryption: verschlüsselt in den Systemtabellen
- Check: INSERT/UPDATE ausschließlich gemäß WHERE-Klausel der View möglich
- Schemabinding: Struktur der Basistabellen/-sichten ist unveränderbar

## Beispiel

```
CREATE VIEW vwEinkauf
AS SELECT * FROM tblPersonal
WHERE PersAbtlg = 'EK'
WITH CHECK OPTION
```

Verwendung:

```
SELECT name FROM vwEinkauf
WHERE umsatz > 50000
```

- Ändern

- ◆ ALTER VIEW <name> ...
- ◆ Wie CREATE VIEW
- ◆ Alle Optionen neu setzen!!!!

- Löschen

- ◆ DROP VIEW <name>

- Die SELECT-Anweisung der View darf nicht enthalten:
  - ◆ ORDER BY
  - ◆ SELECT INTO
  - ◆ Verweis auf temporäre Tabellen

- Kein UPDATE/INSERT/DELETE möglich bei
  - ◆ Aggregatfunktionen
  - ◆ GROUP BY
  - ◆ TOP
  - ◆ UNION
  - ◆ DISTINCT
  - ◆ berechneten Spalten im SELECT

### Welche Views gibt es?

- `SELECT name FROM sysobjects WHERE xtype = 'V';`

### Wie ist eine View festgelegt?

- `SELECT OBJECT_DEFINITION  
(OBJECT_ID('<Viewname>'));`
- `EXEC sp_helptext '<Viewname>';`

## Temporäre Tabelle

- In tempDB
- reduziert Table Scans/Kosten der Suche
- indexierbar

**Lokal:** `CREATE TABLE #<name>(...)`

- Beschränkt auf Batch/Fenster/Sitzung, bzw. bis DB-Trennung

**Global:** `CREATE TABLE ##<name>(...)`

- Für alle Sitzungen/Fenster gültig
- Für alle User (kein REVOKE/DENY möglich)