# Modul 09: temporäre Tabellen, Views (Sichten), Prozeduren: ANGABEN

*Systemvoraussetzungen: eine beliebige Version von SQL-Server und SSMS (Microsoft SQL Server Management Studio)*

*Tools: SQL-Server; SSMS (Microsoft SQL Server Management Studio)*

*Autor: Leonard Hlavin, Letzte Änderung: 19.04.2021*

Temporäre Tabellen ermöglichen uns, Informationen zwischenzuspeichern. Das können Inhalte ganzer Tabellen sein oder das von einem SELECT erzeugte Ergebnis, mit dem wir noch weiterarbeiten möchten.

Es gibt lokale und globale temporäre Tabellen; lokale temporäre Tabellen existieren nur in der Session, in der sie erstellt wurden, auf globale temporäre Tabellen können wir auch aus anderen Sessions zugreifen.

Beide existieren nur so lange, wie die Session, in der sie erstellt wurden, aufrecht ist.

Bei einer lokalen temporären Tabelle stellen wir ein #-Zeichen vor den Namen (#tablename), bei einer globalen temporären Tabelle zwei #-Zeichen (##tablename).

Views werden verwendet, um komplexere Abfragen, die häufiger gebraucht werden, nicht immer neu schreiben zu müssen, aber auch aus Sicherheitsgründen und um beispielsweise nur bestimmte Spalten der beteiligten Tabellen zur Verfügung zu stellen (Berechtigungen).

Eine View wird als virtuelle Tabelle verwendet; die Daten darin sind aktuell, denn bei Verwendung der View wird die dahinterliegende Abfrage ausgeführt.

Mit der View können wir wie mit einer Tabelle arbeiten; daraus ergeben sich sowohl Vor-, als auch Nachteile.

Eine Prozedur ist entfernt vergleichbar mit den Funktionen bei den Programmierern. Auch mit der Prozedur können wir Funktionen ausführen (Serverfunktionen oder UDF), auch die Prozedur rufen wir auf, so ähnlich wie beim function call beim Programmieren, und wie einer Funktion können wir auch der Prozedur Variablen übergeben.

## 1. Übung: Erstellung einer VIEW

Ziel: Erstellen einer View (Sicht); JOINS

Aufgabenstellung:

* Erstellen Sie eine View v\_Rechnungsdaten.
* Wählen Sie sinnvolle Spalten aus mehreren Tabellen, die für die Ausgabe einer Rechnung notwendig wären.
* Führen Sie testweise ein SELECT \* FROM v\_Rechnungsdaten aus.

Ergebnis: 2155 Zeilen

## 2. Übung: Erstellen einer Prozedur

Ziel: Erstellen einer Prozedur

Aufgabenstellung:

* Erstellen Sie eine Prozedur p\_Customers\_Cities.
* Beim Ausführen der Prozedur kann der Name einer Stadt übergeben werden, so dass nur die Kunden- und Bestelldaten der Kunden in dieser Stadt ausgegeben werden.
* Führen Sie die Prozedur für „Buenos Aires“ aus.
* Fortgeschrittenen-Zusatzaufgabe: Erstellen Sie eine temporäre Tabelle, fügen Sie das Ergebnis der Prozedur dort ein und ordnen Sie das Ergebnis nach OrderID

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ergebnis: 16 Zeilen

## 3. Bonusübung: Prozedur und CASE

**Bonusübung: Hierfür benötigen Sie auch das Modul 12 – CREATE,… und das Bonusmodul 13 CASE.**

**Wir verwenden die Tabelle, die wir im Modul 12 erstellen!**

Ziel: Erstellen einer Prozedur; Üben von CASE

Aufgabenstellung:

* Erstellen Sie eine Stored Procedure p\_Raise, die beim Ausführen das Gehalt aller Mitarbeiter, deren aktuelles Gehalt unter dem Durchschnitt liegt, um 20% erhöht, das der Mitarbeiter, deren aktuelles Gehalt über dem Durchschnitt liegt, um 10% erhöht, und das der Mitarbeiter, deren Gehalt 10.000 übersteigt, unverändert lässt.
* Führen Sie die Prozedur mehrmals aus und überprüfen Sie jedes Mal das Ergebnis, indem Sie das Gehalt der Mitarbeiter abfragen.

Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung

Ergebnis: 9 Zeilen (so viele, wie Mitarbeiter). Abhängig vom ursprünglichen Gehalt, das Sie vergeben haben, wird sich das Gehalt entsprechend ändern. Die Gehaltserhöhung ist deshalb so hoch, damit wir zum Testen rascher die 10.000er-Grenze erreichen können.