

3. Problem – Die Struktur der Datenbank ändern

2 Wochen

Theorie

Syntax:

```
CREATE PROCEDURE <Name> [@param1 type1, ...] AS
    -- Sequenz von SQL Befehle
GO
```

Eine gespeicherte Prozedur wird **mit EXEC ausgeführt**:

```
EXEC <Name>
```

Eine Prozedur ändern:

```
ALTER PROCEDURE .....
```

Prozedur mit Output Parameter:

```
CREATE PROCEDURE getKursTitel(@Kredite int, @Number int
output)
AS
    SELECT @Number = COUNT(*)
    FROM Kurse
    WHERE ECTS = @Kredite
GO
```

Prozedur mit Output ausführen:

```
DECLARE @Nr int
SET @Nr = 0
exec getKursTitel 6, @Number=@Nr output
print @Nr
```

Dynamische Execution

Syntax:

```
EXEC (command)
```

Beispiel:

- EXEC ('SELECT * FROM Studenten WHERE age>20')
- ```
GO
```

oder

- DECLARE @query as varchar(MAX)  
SET @query = 'SELECT \* FROM Studenten WHERE age>20'  
EXEC(@query)  
GO

Nachteile von EXEC:

- kann schlecht für Performance sein (es kann sein dass SQL Server für dynamisches SQL den Ausführungsplan jedesmal neu erstellen muss)
- potentielle Sicherheit Probleme (SQL Injection)

Alternative für EXEC → gespeicherte Prozedur **sp\_executesql**:

- manchmal viel schneller als EXEC
- verhindert SQL Injection
- sie können Parameter nur dort verwenden, an denen die SQL Syntax dies auch zulässt  
→ es dürfen keine Parameter für Spalten oder Tabellennamen verwenden können
- wenn dynamisches SQL regelmäßig verwendet wird, ist **sp\_executesql** die bessere Wahl, da der Abfrageplan/ Ausführungsplan wiederverwendet werden kann

```
DECLARE @query as nchar(50)
SET @query = 'SELECT * FROM Studenten WHERE age>@Nr'
EXECUTE sp_executesql @query, N'@Nr int', @Nr=20;
```

### Bemerkung!

1. Wenn Parameter auch für Spalten oder Tabellennamen verwendet werden müssen, funktioniert das nur mit EXEC
2. Wenn die Abfrage als Verkettung von Strings aufgebaut wird, passt auf wo man Leerzeichen braucht (mit **PRINT** die Abfrage überprüfen)

### Beispiel von einer Prozedur mit dynamischer Execution

Eine Prozedur, die eine Indexstruktur löscht:

```
create procedure deleteIndex (@tableName varchar(10),
 @indexName varchar(10))
as
begin
 declare @sqlQuery as varchar(MAX)
 set @sqlQuery = 'drop index ' + @tableName + '.' + @indexName

 print(@sqlQuery)
 exec(@sqlQuery)
end
go
```

Die Prozedur ausführen:

```
exec deleteIndex 'table1', 'index1'
```

## Aufgabe

Manchmal müssen wir die Struktur der Datenbank, nachdem diese erstellt wurde, ändern. Aber solche Änderungen sind nicht immer korrekt, also es muss eine Möglichkeit geben zu einer vorherigen Version zurückzukehren. Eure Aufgabe ist ein Versioning Mechanismus zu erstellen, der einem Benutzer erlaubt von einer Version zu einer anderen zu wechseln.

Schreibe gespeicherte Prozeduren (wenn die SQL Anweisungen generiert wurden müssen diese genau erklärt werden können), die folgende Operationen implementieren:

- den Typ einer Spalte (Attribut) ändert (modify type of column)
- ein default Constraint erstellt
- eine neue Tabelle erstellt (create table)
- eine neue Spalte für eine Tabelle erstellt (add a column)
- eine Referenz-Integritätsregel erstellt (foreign key constraint)

Für jede dieser Operationen schreibe eine neue Operation die den Rollback/Revert implementiert (das umgekehrte der Operation).

Erstelle eine neue Tabelle, die nur die aktuelle Version der Datenbank speichert. (eine Spalte mit einer einzigen Nummer = aktuelle Version)

Schreibe eine gespeicherte Prozedur, die eine Versionsnummer als Parameter kriegt und die Datenbank zu dieser Version bringt.

Tipp für die Lsg.

- ihr könnt eine zusätzliche Tabelle benutzen um für jede neue Version das folgende zu speichern:
  - o welche neue Prozedur wurde ausgeführt (um zu dieser Version zu gelangen)
  - o mit welchen Parametern (wenn nötig)

Für mehrere Informationen über Transact-SQL (declare, begin, end, set, usw.) und gespeicherte Prozeduren:

- Seminar 3
- <https://msdn.microsoft.com/de-de/library/ms190487.aspx>
- <https://msdn.microsoft.com/de-de/library/ms188927.aspx>
- <https://msdn.microsoft.com/de-de/library/ms189484.aspx>
- <https://msdn.microsoft.com/de-de/library/ms178642.aspx>
- <https://msdn.microsoft.com/de-de/library/ms188001.aspx>
- <https://msdn.microsoft.com/de-de/library/ms190782.aspx>