Gespeicherte Funktionen

Funktionen

- 3 Arten von Funktionen
 - > Skalarfunktionen (liefern einzelnen Wert zurück)
 - > Tabellenwertfunktion (liefern Tabelle zurück)
 - ➤ Inline-Funktion (Spezialfall einer Tabellenwert-Funktion)
- Funktionen liefern immer einen Wert zurück
- Funktionen können in SQL-Anweisungen verwendet werden
- In Funktionen können viele Anweisungen nicht genutzt werden
 - > z.B. Definitionsvorgänge (CREATE, ALTER...), Datenänderungen

Skalarfunktion

- Geben einen einzelnen Wert zurück
 - Skalar = mathematische Größe, die allein durch die Angabe eines Einheitswertes charakterisiert ist (einzelner Zahlenwert, einzelne Zeichenkette usw.)

Syntax:

```
CREATE FUNCTION schema.name ([@Parameter <Typ> [,...]])
                                                                         <u>Funktionsdefinition</u> mit optionaler
RETURNS <Typ>
                                                                         Parameterliste, notwendigem
[WITH ENCRYPTION | SCHEMABINDING | EXECUTE AS]
                                                                         Rückgabetyp und optionalen
AS
                                                                         Eigenschaften
BEGIN
                                                                         Funktionsrumpf durch BEGIN und END
        <Anweisungen>
                                                                         begrenzt, mit den Anweisungen und
       <Rückgabewert>
RETURN
                                                                         dem notwendigen RETURN-Wert
END
```

Skalarfunktion Eigenschaften

ENCRYPTION

> Text der Funktion wird in ein verborgenes Format konvertiert

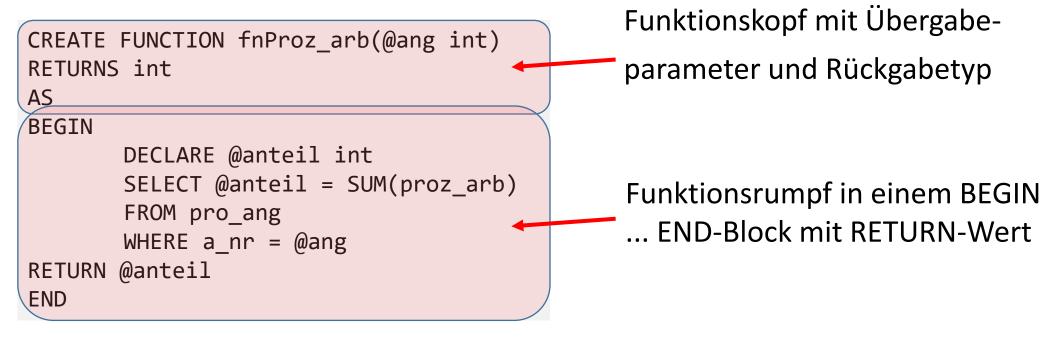
SCHEMABINDING

Von der Funktion verwendete Tabellen k\u00f6nnen nicht mehr "gegen die Funktionsdefinition" ge\u00e4ndert werden

EXECUTE AS <Benutzer>

Funktion wird innerhalb des Sicherheitskontextes des angegebenen Benutzers ausgeführt

Beispiel



Aufruf mit:

Inline-Funktion

- Eigenschaften
 - Gibt eine Tabelle zurück
 - Enthält nur eine SELECT-Anweisung
 - Es gibt keinen Funktionsrumpf
 - RETRUNS TABLE in RETURNS-Klausel
 - RETURN-Klausel enthält SELECT-Anweisung

Beispiel

```
CREATE FUNCTION fnAngAbt(@abteilung int)

RETURNS TABLE

AS

RETURN SELECT ang.name,
    abt.abt_name
    FROM ang
    INNER JOIN abt
    ON ang.abt_nr = abt.abt_nr
    WHERE abt.abt_nr = @abteilung;
```

RETURNS TABLE ist Pflicht bei Inline-Funktion

Es gibt keinen durch BEGIN...END gekennzeichneten Funktionsrumpf; Es gibt nur ein RETURN SELECT (...)

Aufruf mit:

```
SELECT * FROM dbo.fnAngAbt(2);
```

Tabellenwertfunktion

```
Beispiel
```

```
Definition der
                                                                    Rückgabetabelle
CREATE FUNCTION fnAngData(@anr int)
RETURNS @tabelle TABLE (Name nvarchar(max),
               Vorgesetzter nvarchar(max),
               Untergebener nvarchar(max))
AS
                                                                    Befüllen der Rückgabetabelle
BEGIN
       INSERT INTO @tabelle
               SELECT ang.name, a.name, b.name
               FROM ang
                                                                    SELECT-Statement
               LEFT JOIN ang as a ON ang.vorg = a.a nr
               LEFT JOIN ang AS b on ang.a_nr = b.vorg
               WHERE ang.a_nr = @anr
RETURN
                                                                     RETURN Rückgabe
END
SELECT * FROM dbo.fnAngData(205);
                                                      Funktionsrumpf
```

Deterministische Funktionen

- Geben bei gleichen Eingabewerten und gleichem Datenbankstatus immer auch den gleichen Wert zurück
- ➤ Integrierte Aggregat- und Zeichenfolgen-Funktionen
- > Ergebnisse können indiziert werden

Nicht Deterministische Funktionen

- Können bei gleichen Eingabewerten und gleichem Datenbankstatus unterschiedliche Werte zurückgeben
- Ergebnisse können nicht indiziert werden
- Integrierte Konfigurations-, Cursor-, Metadaten-, Sicherheits- und statistische Systemfunktionen