

Triggern.

# Triggers

- Was sind Triggers?
  - Eine bestimmte Art von gespeicherter Prozedur, die automatisch ausgeführt wird wenn eine DML oder DDL Anweisung ausgeführt wird
  - Eine Menge von Aktionen, die durch das Eintreten eines Ereignisses angestoßen werden
- Beispiel von ausgelösten Aktionen (Effekt des Triggers)
  - Änderungen an anderen Tabellen
  - Erzeugung bzw. Transformation von Werten für eingefügte, bzw. geänderte Werte
  - Aufrufen von Datenbankfunktionen

# Triggers

- Wann werden Triggers benutzt?
  - Datenbankzusicherungen (constraints) sichern einen Zustand einer Datenbank
  - Datenbanktrigger werden benutzt, um Aktionen auszuführen, falls bestimmte Bedingungen bzgl. des Zustand einer Datenbank zutreffen
- Ereignisse, die Trigger feuern
  - DML Anweisungen: INSERT, UPDATE, DELETE
  - DDL Anweisungen: CREATE DATABASE, DROP LOGIN, UPDATE STATISTICS, DROP TRIGGER, ALTER TABLE

# Triggers

- Trigger-Konzept
  - ECA-Regeln: Event-Condition-Action Rules
- Event
  - Auslösendes Ereignis einer Aktion
- Condition
  - Bedingung zum Auslösen der Aktion
- Action
  - Angabe der Aktion (Folge von Datenbankoperationen)

# Anlegen von Triggern

```
CREATE TRIGGER <trigger-name>
ON {table | view}
[WITH <dml_trigger_option> [,...n] ]
{FOR | AFTER | INSTEAD OF}
{ [INSERT] [,] [UPDATE] [,] [DELETE] }
[ WITH APPEND ] [NOT FOR REPLICATION ]
AS
    {sql_statement [;] [,...n] |
EXTERNAL NAME <method specifier [;] > }
```

Auslösezeitpunkt

Auslöser

Aktion

# Triggern

- Auslösezeitpunkt
  - FOR / AFTER
  - INSTEAD OF
- Wenn mehrere Trigger für dieselbe Operation definiert werden, dann werden diese in beliebige Reihenfolge ausgeführt
- Wenn ein Trigger ausgeführt wird, stehen 2 spezielle Tabellen zur verfügung: `inserted` und `deleted`

# Trigger

```
CREATE TRIGGER [dbo].[On_Product_Insert]
    ON [dbo].[Products]
    AFTER INSERT
AS
BEGIN
    SET NOCOUNT ON;

    insert into LogBuys (Name, Date, Quantity)
    select Name, GETDATE(), Quantity
    from inserted
END
```

# Trigger

```
CREATE TRIGGER [dbo].[On_Product_Delete]
    ON [dbo].[Products]
    AFTER DELETE
AS
BEGIN
    SET NOCOUNT ON;

    insert into LogSells (Name, Date, Quantity)
    select Name, GETDATE(), Quantity
    from deleted
END
```



# Triggers

```
ALTER TRIGGER [dbo].[On_Product_Update]
    ON [dbo].[Products]
    AFTER UPDATE
AS
BEGIN
    SET NOCOUNT ON;
    insert into LogSells (Name, Date, Quantity)
    select d.Name, GETDATE(), d.Quantity - i.Quantity
    from deleted d inner join inserted i on d.ID=i.ID
    where i.Quantity < d.Quantity

    insert into LogBuys (Name, Date, Quantity)
    select i.Name, GETDATE(), i.Quantity - d.Quantity
    from deleted d inner join inserted i on d.ID=i.ID
    where i.Quantity > d.Quantity
END
```

# Triggers

- SET NOCOUNT ON/OFF
  - ON – die Anzahl der betroffenen Zeilen wird nicht als Teil des Resultsets zurückgegeben
  - OFF – die Anzahl wird zurückgegeben
- @@ROWCOUNT
  - Anzahl der Zeilen auf die sich die letzte Anweisung ausgewirkt hat
  - wird immer aktualisiert (auch wenn NOCOUNT ON ist)

# Triggers

- Entfernen von Triggern

```
DROP TRIGGER <trigger-name>
```

# Anwendung von Triggern

- Definition und Umsetzung von Anwendungslogik („business rules“) → **AFTER-Trigger**
  - Bsp. Erhöhung der Gehälter um maximal 10%
  - Änderung der Anwendungslogik ohne Änderung der Anwendung
  - Wird dazu verwendet, Änderungen an der Datenbasis vorzunehmen bzw. Externe Funktionen (zB Fehlermeldungen) auszuführen
- Erweiterte Integritätsprüfung
  - Überprüfung der Eingabedaten auf Gültigkeit
  - Automatisches Generieren von Werten für neu eingefügte Tupel
  - Lesen von anderen Tabellen zur Evaluierung von Querverweisen

# Aufgabe

- Für die Tabelle **Studenten**(MatrNr, Name, Vorname, Geburtsdatum, Email) wollen wir bei dem Einfügen Folgendes überprüfen:
  - '1990-1-1' <= Geburtsdatum <= '2002-1-1'
- Implementiere das einmal als **Integritätsregel** und einmal als **Trigger**, sodass Tupeln, welche die Regel nicht einhalten, nicht eingefügt werden

```
ALTER TABLE [NameT]  
ADD CONSTRAINT [NameC]  
CHECK ...
```