

**Grundsätzliches:** Trigger sind SQL-Codeelemente, welche bei Veränderungen des Datenbestandes auf Tabellen automatisch ausgeführt werden. Veränderungen von Datenbeständen laufen über eines dieser drei Statements:

- INSERT (bzw. LOAD DATA)
- UPDATE
- DELETE

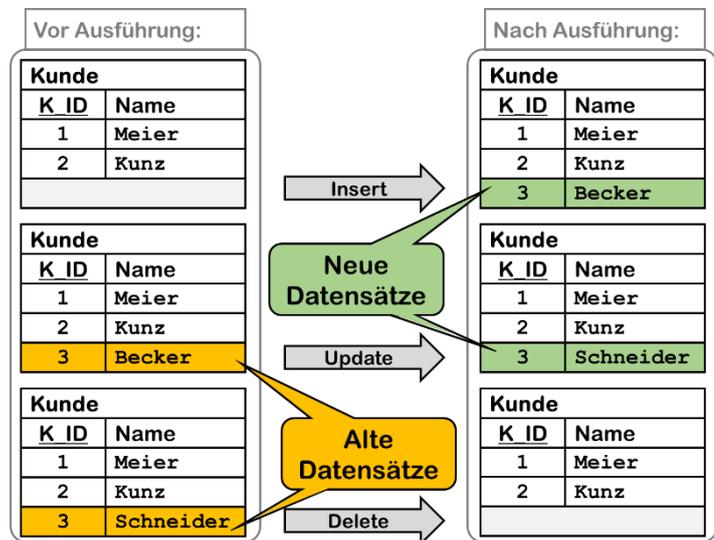
Anmerkung: REPLACE ist ein DELETE mit einem anschließendem INSERT.

Bei Datenänderungen können wir den Zustand der Tabelle vor und nach der Ausführung definieren.

Auch können wir neue (oder veränderte) bzw. alte Datensätze identifizieren.

Ein Trigger muss also wissen:

- \_\_\_\_\_
- \_\_\_\_\_
- \_\_\_\_\_



Bezüglich dieser drei Eigenschaften müssen Trigger eindeutig sein. Es darf also keine 2 Trigger geben, welche bspw. „vor“ einer „Änderung“ auf der „Kundentabelle“ agiert.

Hier ein beispielhafter Code für einen Trigger. Alles, was fett gedruckt wurde, ist zwingender Code. Der innere Teil des Triggers ist „optional“ – da er den im Triggerfall auszuführenden Code beinhaltet.

```

01 DELIMITER $$
02 CREATE TRIGGER MyTrigger AFTER UPDATE ON Kunde
03   FOR EACH ROW BEGIN
04     IF (SELECT VIP FROM LevTab l WHERE l.K_ID = NEW.K_ID) = 1 THEN
05       INSERT INTO KundeLog SET
06         K_ID = OLD.K_ID,
07         NameVon = OLD.Name,
08         NameNach = NEW.Name;
09       INSERT INTO ChangeLog SET
10         ChangeID = OLD.K_ID,
11         ChangeTAB = "Kunde";
12     END IF;
13 END $$
14 DELIMITER ;

```

Analysieren Sie nun den Code und notieren Sie, was die einzelnen Codezeilen machen:

Z.:	Erklärung:
01 -	
02 -	
03 -	
04 -	
05 -	
06 -	
08 -	
09 -	
12 -	
13 -	
14	

Dieser Trigger wird also nach einer Änderung eines, oder mehrerer Datensätze der Tabelle „Kunde“ für jeden Datensatz je einen neuen Eintrag in der KundeLog und der ChangeLog Tabelle durchführen, sofern der die KundenID in der LevTab Tabelle gefunden wird und den Wert 1 aufweist.

*Anmerkung: Die Nutzung von Trigger macht eine Datenbank unübersichtlicher. Es ist auch so, dass ein SQL Fehler innerhalb eines Triggers bei der Ausführung des triggernden Statements auftritt. Ist bspw. im oberen Beispiel in Zeile 08 ein Syntaxfehler, so würde bei einem UPDATE auf die Kundentabelle eine Fehlermeldung erzeugt werden, dass ein INSERT auf die KndeLog Tabelle fehlgeschlagen ist.*

**Wichtig:** Trigger sollten immer nur Werte verändern, welche nichts mit den Businessdaten zu tun haben. Somit können wir Trigger für Loggingeinträge verwenden, oder auch für Performanceoptimierungen – bspw. das Updaten von materialisierten Views.

#### Nützliches:

Anzeigen aller Trigger einer Datenbank:

```
SHOW TRIGGERS;
```

Anzeigen der Definition eines Triggers:

```
SHOW CREATE TRIGGER MyTrigger;
```

Löschen eines Triggers:

```
DROP TRIGGER MyTrigger;
```