Trigger

Wechseln zu:Navigation, Suche

Dieser Artikel sollte auf Korrektheit, Quellenangaben und GlossarWiki-Konformität hin überprüft werden.

Inhaltsverzeichnis

1 Definition

In Datenbanken ist es neben der Speicherung und Abfrage von Daten auch möglich Logik selbst zu implementieren. Trigger gehören zum Bereich Logik und werden automatisch bei bestimmten Ereignissen, wie UPDATE, INSERT oder DELETE von Daten, ausgelöst.

2 Vorteile

Trigger lösen automatisch aus; Der Benutzer muss sich nicht über weitere Vorgänge des Updates Gedanken machen (Fehlervermeidung)
Trigger können Bedingungen bei INSERT, UPDATE oder DELETE Befehlen überprüfen
Trigger werden einmal für eine bestimmte Tabelle und Aufgabe implementiert
Trigger können in sich wiederum Trigger auslösen.
Trigger werden direkt im Datenbanksystemaufgerufen
Trigger können den Client-Serververkehr entlasten

3 Nachteile & Warnungen

Trigger können in sich wiederum Trigger auslösen (Belastung des Datenbanksystems)
Debuggen wird erschwert, weil Statements in Triggern nicht beim Debuggen aufgeführt werden
Mächtigkeit der Trigger fordert eine sehr gute Implementierung.

Fehlerhafte Trigger können Datenbestände beschädigen oder unbrauchbar machen
Festlegung, wie oft ein Trigger ausgeführt werden soll (Entlastung des Datenbanksystems)

4 Möglichkeiten von Triggern

Oft ist es notwendig, bei einer Löschung eines Kunden oder Users weitere Löschvorgänge auszuführen
Protokollerstellung bei einem Löschvorgang
Sicherung von Daten in eine Sicherungstabelle (s. Beispiel)
Auslösung von Vorgängen bei einem bestimmten Fall (z.B.: Mitgliederzahl über 100)

5 Implementierung von Triggern

5.1 Allgemein

Für die Erstellung von Triggern werden spezielle Programmiersprachen (PL/SQL oder SQL PL) verwendet. Außerdem ist es in einigen Datenbanksystemen möglich, in Triggern auch Stored Procedures aufzurufen. Dadurch entsteht die Möglichkeit ein Programm aus einer anderen Programmiersprache zu verwenden,

5.2 Syntax

Anlegen eines Triggers

```
CREATE TRIGGER triggername
[BEFORE | AFTER] [ INSERT | DELETE | UPDATE [OR ...] ]
ON tablename FOR EACH ROW
EXECUTE PROCEDURE functionname [(args)];
```

Löschen eines Triggers

```
DROP TRIGGER triggername
```

6 Beispiele

7 Automatische Sicherung von Inhalten

einer Tabelle in eine zweite Tabelle

1. Schritt: Erstellung einer Tabelle

```
CREATE TABLE tabelle1
(
        id          integer primary key,
        name        varchar(40) not null,
        telefon     varchar(10),
        umsatz      decimal(10,2)
);
```

Erklärung: Anlegen der Tabelle.

2. Schritt: Erstellung der Sicherungstabelle

```
CREATE TABLE tabelle1_backup
(
        id              integer primary key,
        name            varchar(40) not null,
        telefon         varchar(10),
        umsatz          decimal(10,2)
        user_changed    varchar(40),
        date_changed    date,
        operation       varchar(15)
);
```

Erklärung: Anlegen der Sicherungstabelle mit den gleichen Spalten aus der Haupttabelle und zusätzlichen Spalten, die das Datum, die vorgenommene Änderung und User speichern.

3. Schritt: Erstellung einer Funktion für die automatische Sicherung

```
CREATE FUNCTION backup_tabelle1() RETURNS OPAQUE AS ‘
BEGIN
INSERT INTO tabelle1_backup
VALUES
(
OLD.id,
OLD.name,
OLD.telefon,
OLD.umsatz,
CURRENT_USER,
now(),
TG_OP
);
RETURN NULL;
END;
‘ LANGUAGE ‘plpgsql’
```

Erklärung: Die Funktion backup_tabelle1() nimmt die nicht mehr aktuellen Werte aus tabelle1 und speichert sie in der tabelle1_backup als eine neue Zeile.

4. Erstellung des Triggers

```
CREATE TRIGGER backup_tabelle1
AFTER DELETE OR UPDATE
ON tabelle1
FOR EACH ROW
EXECUTE PROCEDURE backup_tabelle1();
```

Erklärung: Der Trigger wird erstellt und nach jedem DELETE oder UPDATE Befehl der auf die tabelle1 ausgeführt wird, ruft er die Funktion backup_tabelle1() auf,

8 Quellen

Throll, M.; Bartosch, O. (2007): Einstieg in SQL
Eisentraut, P. (2003): PostgreSQL - Das offizielle Handbuch
Douglas, K.; Douglas S (2003): PostgreSQL
[1]
[2]

Kategorien:
SQL
Glossar