

Praktikumsaufgabe 3

Lernziele

Einblick in die serverseitige Programmierung. Kennenlernen der Nützlichkeit von Triggern an einem typischen Anwendungsgebiet.

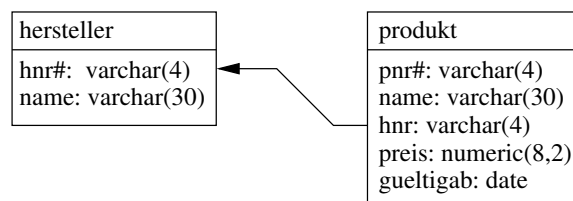
Vorbereitung

Folgende Informationen müssen Sie zum *Antestat* bereithalten:

- Was ist ein Trigger und auf welche Weise wird er in PostgreSQL angelegt? ([1] 3.2.2 & 3.2.3, [2] Kap. 5.15 und 8.2.4, [3] Kap. “Trigger Procedures”)
- Was ist ein *View* und wie legt man ihn an? ([1] Kap. 3.1.3, [2] Kap. 3.3, [3] Kap. 3.2 “Views” und SQL-Doku “Create View”)
- Erläutern Sie, wie Sie die Historie implementieren wollen (Struktur der Historientabelle, was macht Ihr Trigger). Was ist der Primary Key der Historientabelle?
- Legen Sie ein SQL-Statement vor, mit dem im Rahmen Ihres Lösungsansatzes der Preis eines bestimmten Produkts (*pnr*) zu einem bestimmten Datum ermittelt werden kann.
Hinweis: Sie können sich die Arbeit erleichtern durch die Definition eines Views.

Aufgabe

Das folgende Datenbankschema soll um eine Preishistorie für die Produkte erweitert werden:



Die Tabelle *produkt* enthält nur den jeweils aktuell gültigen Preis (dieser Preis ist gültig seit *gueltigab*). Die neu einzuführende Historientabelle *preishist* soll bei Preisänderungen automatisch über Trigger gefüllt werden. Dabei gelten folgende Randbedingungen:

- Aus der Historientabelle muss ersichtlich sein, wann vergangene Preiserhöhungen erfolgt sind und was die damals gültigen Preise waren.
- Eine Änderung an *produkt* ist nur historienrelevant, wenn eine Preisänderung stattfindet und sich gleichzeitig *gueltigab* erhöht. Wenn sich nur der Preis ändert und nicht *gueltigab*, erfolgt ein normales Update auf *produkt*.
- Eine Änderung von *gueltigab* in *produkt* auf ein Datum vor der letzten Preisänderung laut *preishist* muss unterbunden werden (sonst würde die zeitliche Abfolge durcheinandergebracht).

Demonstrieren Sie mit Beispielupdates nebst anschliessenden Abfragen, dass Ihre Implementierung funktioniert.

Referenzen

- [1] Dalitz: *Datenbanksysteme*. Vorlesung an der Hochschule Niederrhein, Krefeld (2013)
- [2] Hartwig: *PostgreSQL Professionell und Praxisnah*. Semesterapparat (TWY Hart)
Achtung: ab Version 7.3 muss der Returnwert einer Triggerfunktion *trigger* statt *opaque* sein.
- [3] The PostgreSQL Global Development Group:
PostgreSQL 8.x.y Dokumentation. <http://www.postgresql.org/docs/> (2008-2011)