

## DB – Praktikum 13

---

### *Integritätssicherung, Datenbankprogrammierung*

#### **Vorbemerkungen**

Gespeicherte Prozeduren („Stored Procedures“) sind Routinen, die in der Datenbank aufbewahrt werden und Code kapseln. In PostgreSQL ist der Begriff „Stored Procedure“ allerdings mehrdeutig: PostgreSQL <= V10 unterstützt nur „Functions“, die entsprechend in vielen Dokumenten auch als „Stored Procedures“ bezeichnet werden. PostgreSQL >= V11 führt zudem auch „Procedures“ ein. Der Unterschied ist aber für unsere Zwecke nicht interessant.

„Functions“ können in einer Reihe von Sprachen geschrieben werden: SQL, C, aber auch „Prozedurale Sprachen“, die als Erweiterung von PostgreSQL realisiert sind. Üblicherweise werden bei einer Standardinstallation bereits „PL/pgSQL“, „PL/Perl“, „PL/Python“ und „PL/Tcl“ mitgeliefert. Wir beschränken uns im Folgenden auf SQL und PL/pgSQL.

Functions nehmen Parameter entgegen und geben Skalarwerte oder Tabellen zurück.

Ein Trigger ist mit ausgewählten Ereignissen (meist DML-Anweisungen, es gibt aber auch DDL-Trigger) in einer Tabelle oder View verknüpft. Immer, wenn dieses Ereignis auftritt, wird der Trigger ausgelöst und sein Code ausgeführt. Eine explizite Ausführung ist nicht möglich. Typischerweise wird in PostgreSQL eine Function aufgerufen.

Wir arbeiten im Praktikum 13 mit der Datenbank IMDB. Diese haben Sie ja schon im Praktikum 1 installiert.

Nutzen Sie für die Aufgaben auch die offizielle PostgreSQL-Dokumentation.

#### **Aufgabe 1: Referenzdaten beschaffen**

In der Aufgabe 1 geht es darum, Referenzdaten (Postleitzahlen der Schweiz und ISO-Ländercodes) bereitzustellen, die anschliessend zur Integritätssicherung verwendet werden sollen.

Unter der Adresse <https://postleitzahlenschweiz.ch/tabelle> steht eine Liste von Postleitzahlen für die Schweiz bzw. Liechtenstein zur Verfügung.

Unter der Adresse: [https://datahub.io/core/country-list#resource-country-list\\_zip](https://datahub.io/core/country-list#resource-country-list_zip) können Daten zu ISO-Ländercodes bezogen werden.

Bevor diese „Rohdaten“ aber benutzt werden können, müssen einige vorbereitende Schritte ausgeführt werden (Spalten löschen bzw. umbenennen, Daten in UTF-8 formatieren). Um Zeit zu sparen, finden Sie die bereits bereinigten Referenzdaten auf Moodle unter `D_Praktikum_13.zip`.

Laden Sie diese Datei herunter und entpacken Sie sie.

## Aufgabe 2: Referenzdaten bereitstellen

Studieren Sie das Skript `S_Praktikum_13.sql` und passen es dann an ihre eigene Umgebung an (Pfade zu Ausgangsdaten). Lassen Sie es dann laufen. Damit werden die Referenzdaten in die Datenbank geladen.

Überprüfen Sie anschliessend mit Hilfe von SQL-Abfragen kurz die PLZ-Daten (doppelte PLZ, doppelte Orte, Kleinste, Grösste, Anzahl verschiedener PLZ) um die Daten besser zu „verstehen“.

## Aufgabe 3: Adresstabelle vorbereiten

Erstellen Sie nun mit Hilfe einer DDL-Anweisung eine neue Tabelle mit der Bezeichnung `Adressen`. Diese soll folgende Attribute umfassen:

- `AdrNr` integer, primary key
- `Nachname` varchar(50)
- `Vorname` varchar(50)
- `Strasse` varchar(50)
- `LandCode` char(2)
- `PLZ` varchar(10)
- `Ort` varchar(50)
- `Land` varchar(50)

Dabei sollen zusätzlich folgende Konsistenzbedingungen sichergestellt werden:

- Alle Attribute brauchen immer einen Wert.
- Wenn nichts geliefert wird, so sollen `LandCode` bzw. `Land` die Werte „CH“ bzw. „Switzerland“ erhalten.
- PLZ müssen mindestens vier Zeichen lang sein.
- Adressnummern müssen positiv sein und sollen automatisch vom System vergeben werden.

Testen Sie Ihre Lösung durch Einfügen einiger Datensätze.

## Aufgabe 4: Prüfung einer schweizerischen PLZ auf Gültigkeit

Das Ziel ist nun, anhand der beiden Referenztabellen sicherzustellen, dass keine falschen Daten in die Adresstabelle eingegeben werden können. Als erstes sollen schweizerische PLZ (d.h. wenn der `LandCode` = ‚CH‘ ist) daraufhin überprüft werden, dass sie vierstellig, numerisch und in der Referenz-tabelle `PLZSchweiz` vorhanden sind.

Schreiben Sie dafür eine Funktion `IsCHPLZ` die als Argument eine zu überprüfende PLZ erhält und die als Resultat entweder 0 (= Fehler) oder 1 (= korrekte PLZ) zurückliefert. Verwenden Sie für den Test auf „vierstellig numerisch“, einen regulären Ausdruck.

## Aufgabe 5: Prüfung von Adressen

Die Funktion von Aufgabe 4 soll nun eingesetzt werden um bei jeder Einfüge- bzw. Änderungsoperation bei der Tabelle Adressen von Aufgabe 3 automatisch zu prüfen, dass die PLZ korrekt ist (ansonsten soll keine Mutation durchgeführt werden).

Erstellen Sie dazu einen Trigger auf der Tabelle Adressen, der bei Einfüge- oder Änderungsoperationen von schweizerischen oder liechtensteinischen Adressen prüft, ob die PLZ korrekt ist.

Testen Sie die Funktion des Triggers durch Ausführen verschiedener Einfüge- bzw. Änderungsanweisungen.

## Aufgabe 6: Ausbau der Integritätsprüfung

Die bis jetzt implementierte Konsistenzprüfung ist nur ein erster Schritt. Weitere Möglichkeiten wären z.B. folgende:

- Die Kombination PLZ / Ort muss in der Referenztabelle vorhanden sein.
- Wenn nur der LandCode aber kein Land geliefert wird, so liesse sich das ergänzen anhand der ISOLaenderCodes-Tabelle.
- Ländercodes müssen in der Referenztabelle vorhanden sein.
- In einzelnen Fällen könnte ein fehlender Ort aus der PLZ abgeleitet werden (Frage: Warum geht das nicht immer?).
- ...

Überlegen Sie sich weitere mögliche Prüfungen.