Table of Contents

Die Anzeige von Daten aus einer Datenbank Tabelle ist Grundvoraussetzung für Datenbank Applikationen. Eine weitere Grundvoraussetzung ist die Einschränkung der Datenmenge. Dies kann Clientseitig durchgeführt werden und auch Datenbankseitig.

Am Client wird die Filterung im Hauptspeicher durchgeführt, das setzt allerdings voraus das alle Daten vom Server/Datenbank zum Client übertragen werden. Bei geringen Datenmengen sollte das keine Probleme bereiten und reduziert die Kommunikation. Dabei ist allerdings zu beachten, daß die Daten manuell aktualisiert werden müssen!

Die Datenbankseitige Filterung ist die Standardeinstellung und in den meisten Fällen die bessere Wahl. Dabei wird das SQL Kommando um die Filterbedingung erweitert und die Datenmenge wird dadurch bereits von der Datenbank eingeschränkt. Für den Client wird dann nur noch die eingeschränkte Datenmenge aufbereitet und aufgrund von Load-on-Demand wird auch hier nur die tatsächlich benötigte Menge übertragen. Außerdem wird immer auf die Echtdaten zugegriffen.

Anwendungsbeispiel

Unsere Applikation verwaltet Kontaktdaten (Personen, Adresse, usw.). In einer unserer Masken existiert ein Feld in dem ein Suchbegriff eingegeben werden kann. Durch einen Button wird eine Wildcard Suche nach Vorname, Nachname, Straße, Ort ausgelöst.

Nachfolgende Client Action führt die gewünschte Filterung durch:

```
/**
 * Searches the contacts with the search text.
 *
 * @throws ModelException if the search fails
 */
public void doFilter() throws ModelException
{
   String sText = (String)drSearch.getValue("SEARCH");

   if (sText == null)
   {
      //reset the filter: show all rows
      rdbContacts.setFilter(null);
   }
   else
   {
      //set the filter: show only found rows
      ICondition filter = new LikeIgnoreCase("FIRSTNAME", "*" + sText +
"*").or(
                     new LikeIgnoreCase("LASTNAME", "*" + sText + "*").or(
                     new LikeIgnoreCase("STREET", "*" + sText + "*").or(
                     new LikeIgnoreCase("TOWN", "*" + sText + "*"))));
      rdbContacts.setFilter(filter);
   }
}
```

Der Filter bzw. die Condition wird auf ein RemoteDataBook angewandt und wie von SQL bekannt zusammengesetzt. Unsere Condition sucht nach dem Vorkommen des eingegebenen Textes,

entweder im Vornamen, dem Nachnamen, der Straße oder der Stadt.

Für die Filterung ausschließlich am Client, müsste das RemoteDataBook wie folgt konfiguriert werden:

```
rdbContacts.setMemFilter(true);
```

Folgende Filter Klassen stehen zur Verfügung:

- ContainsIgnoreCase
- EndsWithIgnoreCase
- Equals
- Greater
- GreaterEquals
- Less
- LessEquals
- Like
- LikeIgnoreCase
- LikeReverse
- LikeReverseIgnoreCase
- StartsWithIgnoreCase
- Not
- Or
- And

Es gilt zu Beachten, daß ein Filter nach Veränderung immer neu zugewiesen werden muss, z.B.:

```
filter = filter.and(new Equals("CODE", "A-123").and(new Equals("REF",
"re133")));
```

Ohne die Zuweisung von `filter = filter…` wird der Filter nicht verändert.

Details, siehe javadoc.