

# Datenquelle erstellen im iReport-Designer

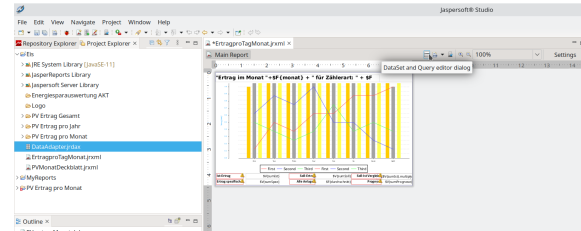
Vorgehensweise:

- Schritt 1
- Schritt 2
- Schritt 3
- Schritt 4

Nachdem die Verbindung eingerichtet wurde, ist es möglich eine Datenquelle zu erstellen. Diese Datenquelle muß später in den Nucleos Datenquelleneditor übertragen werden. Der Vorteil der Erstellung im Designer besteht darin., daß die Attribute(Daten) einer Abfrage nach der Bestätigung sofort als Felder im Formular zur Verfügung stehen.

## Schritt 1

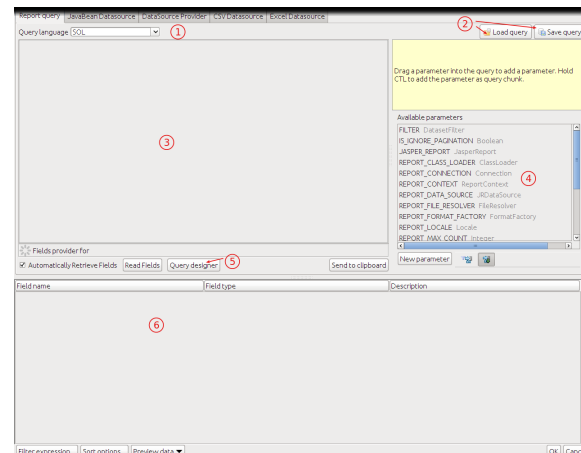
Öffnen von DataSet and Query editor Dialog



## Schritt 2

Nachfolgend werden kurz die Bereiche beschrieben, die für die Erstellung der Datenquelle getestet wurden.

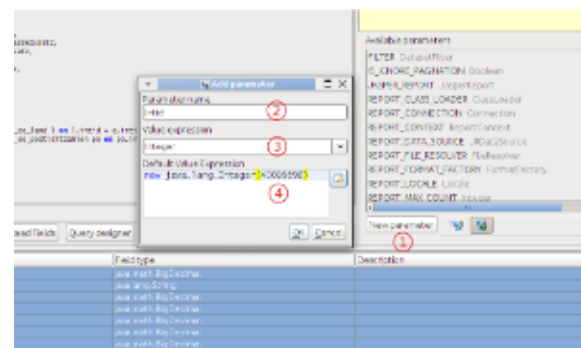
1. Auswahl der Abfragesprache des verwendeten Datenbankmanagementsystems, hier SQL.
2. Es können Abfragen gespeichert und geladen werden. Die Dateien müssen vom Typ .sql oder .txt sein, wie sie möglicher Weise mit einem anderen Datenbankadministrationstool (Pg-Admin "Postgres") erstellt werden.
3. Hier wird die Abfrage entwickelt bzw. editiert oder hochgeladen.
4. Ein Datensatz wird beispielsweise für ein Formular nach einem Parameter gefiltert.(Es wird nur ein Auftrag mit einer speziellen Intid gefiltert). Für dieses Beispiel wird ein Parameter "Intid" vom Typ Integer erstellt. (nächster Schritt)
5. Zum Erstellen der Datenquelle kann auch der eingebaute Query designer verwendet werden. Er gibt die Möglichkeit mit wenig Kenntnissen über SQL, Attribute aus Tabellen auszuwählen und zu einer Abfrage zusammenzufügen. (In diesem Beispiel wird er jedoch nicht verwendet)
6. Ist die Abfrage syntaktisch korrekt wird die Ergebnismenge oder Fehler hier ausgegeben.



## Schritt 3

Um nach einem bestimmten Datensatz zu filtern wird ein Parameter benötigt, der in der Where-Bedingung dafür sorgt, dass die Abfrage eingeschränkt wird. Je nach Anforderung können verschiedenste Parameter erstellt werden(Integer, Date, String,...)

1. "New Parameter" betätigen.
2. Bezeichnung des Parameters eingeben hier "Intid"
3. Typ des Parameters auswählen "Integer"
4. Der Parameter kann mit einem Wert vorbelegt werden. Über ein das Admintool zum DBMS ermittelt man eine intid zu einem bestehenden Datensatz. In diesem Beispiel wird nach einer Intid gefiltert. Jede Datenbank-Tabelle besitzt eine Spalte "Intid"



|   | intid [PK] numeric(20,0) | datchange timestamp | strchange character | datcreated timestamp | strcreated timestamp |
|---|--------------------------|---------------------|---------------------|----------------------|----------------------|
| 1 | 40006408                 | 2012-09-nuclos      |                     | 2012-07-nuclos       |                      |

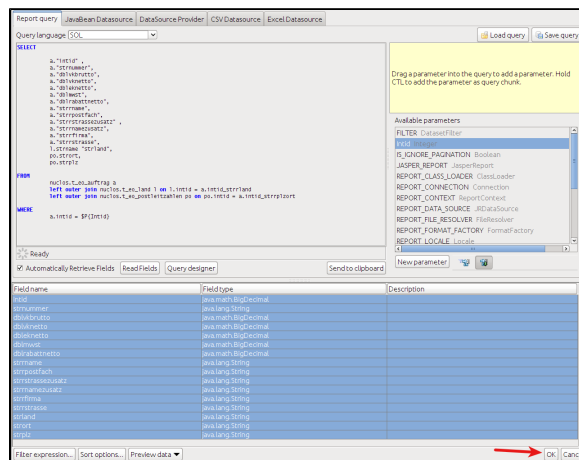
## Schritt 4

In der Abfrage sind unterhalb des SELECT-Statement alle gewünschten Spalten der Tabelle Auftrag, Land und Postleitzahlen aufgelistet.

Die Tabelle Auftrag besitzt Verbindungen(Referenzen) zu anderen Tabellen, welche in der FROM-Bedingung referenziert(Join) werden. Dabei werden auch nur die Datensätze herangezogen die der intid entsprechen(intid\_strland,...)

Die WHERE-Bedingung ermittelt mit Hilfe des erstellten Parameters (\$P {Intid}) den richtigen Auftrag.

SQL-Statements können beliebig groß und umfangreich werden.



### Beispiel Auftrag

```
SELECT

    a."intid" ,
    a."strnummer",
    a."dblvkbrutto",
    a."dblvknetto",
    a."dbleknetto",
    a."dblmwst",
    a."dblrattnetto",
    a."strname",
    a."strrpostfach",
    a."strrstrassezusatz",
    a."strnamezusatz",
    a."strrfirma",
    a."strrstrasse" ,
    l."strname" strland,
    po."strort",
    po."strplz"

FROM

    nuclos.t_eo_auftrag a
    left outer join nuclos.t_eo_land l on l.intid
= a.intid_strland
    left outer join nuclos.t_eo_postleitzahlen po
on po.intid = a.intid_strrplzort

WHERE

    a.intid = $P{Intid} //z.B. 40006408
```

War die Abfrage erfolgreich kann mit "OK" bestätigt werden.

Alle Attribute aus dem SELECT-Statement werden automatisch mit ihren Namen bzw. Aliasnamen angelegt.

**Tipp:** Wenn die Namen der Attribute mit "str" beginnen so bedeutet das der Datentyp ein String ist.(dbl = Double,int =Integer, bln = BOOLEAN). Zur Fehleranalyse und zur weiteren Berechnung läßt sich schnell auf den Datentypen schließen.

Nummerische Sql-typen konvertiert der Designer in java.lang.BigDecimal. Für die Darstellung im Report muss keine Konvertierung vorgenommen werden.

