Keine Angst vor sperrigen Ausdrücken

1. Plötzlich ganz Einfach

```
/* bestimmt den den Drehwinkel der kürzesten Seite eines Rechtecks zur Nordrichtung */

   with_variable('geo', |
            with_variable( 'seg', segments_to_lines(boundary( oriented_bbox(  $geometry  )  )),
                case
                    when  length(geometry_n( @seg,1)) <  length(geometry_n( @seg,2))
                    then geometry_n( @seg,1)
                    else geometry_n( @seg,2)

                end
            )  ,
        (
            case
                when  y(start_point(  @geo )) > y(end_point(  @geo  ))
                then  degrees(
                            azimuth(
                                        start_point(  @geo  ),
                                        end_point(    @geo )
                                    )
                        )
                else  degrees(
                            azimuth(
                                        end_point(   @geo  ),
                                        start_point( @geo )
                                    )
                        )
            end
    )
)
        /* Neuerdings geht das aber auch einfach so: */
        main_angle($geometry) + 90
```

Keine Angst vor sperrigen Ausdrücken

2 Einrücken macht lesbar und Kommentare helfen

```
-- Ersetzen Kaskade zum ändern von Umlauten

replace(replace(replace(  "GEN" ,'ö','oe'),'ü','ue') ,'ä','ae')

oder

replace(
    replace(
        replace(  "GEN" ,'ö','oe'),
        'ü','ue')
,'ä','ae')
```

```
-- Flächenberechnung mit drei Stellen hinterm Komma
round(            -- round(wert,Stellen) ist die Rundungsfunktion
   ($area / 10000)   -- $area gibt die Fläche in Eiheiten des KBS aus (qm) geteilt durch 10000 ergibt ha
   ,3)
  /*#############################################################################
```

Keine Angst vor sperrigen Ausdrücken

3 Bedingungen

```
 Case
»    when "EWZ" => 500000 then 'Riesig'
»    when  "EWZ" => 100000 then 'groß'
»    when  "EWZ" => 50000 then 'mittel'
»    when  "EWZ" => 15000 then 'klein'
»    when  "EWZ" => 2000 then 'sehr klein'
»    when "EWZ" < 2000 then 'winzig'
»
»    Else 'unbekannt'
End
```

Keine Angst vor sperrigen Ausdrücken

4 Aggregieren

```
minimum( "EWZ") -- gibt die kleinste Einwonerzahl des Datenbestands aus und schreibt Sie in jede Zeile
maximum( "EWZ") -- gibt die größte Einwonerzahl des Datenbestands aus und schreibt Sie in jede Zeile

/* Es könne weitere Parameter angegeben werden, z.B zum gruppieren nach Werten einer weiteren Spalte
In der Hilfe des AE werden die Parameter aufgeführt. Sie sind entweder in der richtigen Reihenfolge kommagetrennt zu notieren oder
als benannte Parameter in der Form parametername:= ausdruck einzugenen*/

minimum( "EWZ" , "BEZ")  -- Gibt die Namen der Gemeinde mit geringster EWZ, gruppiert nach  Spalte "BEZ" also jeweils für Stadt und Gemeinde
aus
minimum( "EWZ" ,group_by:=  "BEZ")  --hier mit benanntem Paramter group:by

/* Die Verwendung benannter Parameter und Zeilenumbrüche machen komplexe Abfragen lesbarer
- Die Abfrage zur Ermittlung der Stadt mit der  der kleinsten Einwohnzahl in zwei verschiedenen Schreibweisen*/


-- In einer Zeile ohne benannte Parammeter
"EWZ" = minimum( "EWZ" ,"BEZ","BEZ" =  'Stadt' and "EWZ" > 0 )

-- Mit benannten Parametern
"EWZ" = minimum( "EWZ",group_by:= "BEZ", filter:=  "BEZ" = 'Stadt' and "EWZ" > 0 )

-- Mit Zeilenumbrüchen wird es lesbarer, ausserdem lassen sich Kommentare hinter die einzelnen Parameter schreiben

 "EWZ" =                    -- Der nachfolgende Ausdruck wird mit der EWZ jeder Zeile verglichen , wenn Übereinstimmung wird ausgewählt
    minimum( "EWZ" ,        -- Der minimale Wert des Einwohnerzahl im Datenestand
            group_by:=  "BEZ",      --Jeweils gruppiert nach Spalte BEZ, um jeweils für die Kategorie Stadt und der Kategorie Gemeinde zu
            ermitteln
            filter:=    "BEZ" =  'Stadt' and "EWZ" > 0  -- Ergebnis gefiltert auf Stadt und einer EWZ größer als 0
            )
```

Keine Angst vor sperrigen Ausdrücken

5 Arrays kommen ins Spiel: Die 10 größten Städte suchen

```
/*##########################################
PDie 10 größten Städte suchen
##################################
*/


array_to_string(   » »    -- Rückumwandlung in Textfeld
»   array_slice( »   »    -- Einzelne Stellen abfragen
»      »   string_to_array(    -- umwandeln in Listenfeld (Array)
»      »      »      »
»      »      »   concatenate(  "GEN",»    --Auflistung aller gemeindenamen
»      »      »      »      »      »   filter:=  "EWZ" > 0,
»      »      »      »      »      »   order_by:= 1/"EWZ",  --sortiert absteigend
»      »      »      »      »      »   concatenator:=','
»      »      »      »      »      »   )
»      »      »      »   )»   »    »    --umgewandelt in Listenfeld
»      »      »      »   ,0,9)»   »    -- Stellen 0-9 werden abgefragt
»      »   )»   »    »    »    »    -- in text umgewandelt
»
»   like »   »    -- Vergleich mit jeder Zeile.
»   '%' ||"gen"|| '%'  -- Entspricht ein  Teil der Liste der gemeinde

»   -- Geht auch über
    array_to_string(array_agg
```

Keine Angst vor sperrigen Ausdrücken

6 Wirre Zeichenfolgen: Suchen mit regulären Ausdrücken

```
»   case
»   »   when  "leistung"  ilike '%kw%'  -- Testen ob kw in der Spalte vorkommz
»   »   then
»   »   »
»   »   »   regexp_substr(  replace(   "leistung"  ,',','.') ,'[0-9.]+') / 1000  -- Zahlenwerte und . herausholen
            sowie durch 1000 dividieren
»   »
»   »   when   "leistung"  ilike '%mw%'
»   »   then
»   »   »    regexp_substr(  replace(   "leistung"  ,',','.') ,'[0-9.]+')
»   end

/* Hausnummern extrahieren '*/

/*Wenn garantiert kein Leerzeichen in der Im Straßenname ist, lässt das Ganze auch mit substr und strpos lösen */

substr( 'Hauptstraße 32 ab',
»   »   »   »   »   »   »   strpos('Hauptstraße 32 ab',' ') +1)
-- Der Ausdruck holt sämtliche Zeichen nach dem ersten Leerzeichen

-- Das ist unflexibel. Mit regexp_substr ist es präziser
regexp_substr( 'Hauptstraße 32 ab' ,'\\d{1,4}[ a-z]{0,3}')       32 ab  -- 1-4 Ziffern dann aus der Menge
```

Keine Angst vor sperrigen Ausdrücken

7 Datendefinierte Überschreibung: Schriftgröße und Platzierung

```
/* Platzierung der Beschriftung */
-- z.B. Unter
Layer > Eigenschaften > Beschriftung >

Dort:
if("xc" is not NULL, make_point("xc","yc")
»    ,»
»    case
»    »    when "platz_typ" = 1 then   pole_of_inaccessibility($geometry,0.1)
»    »    when "platz_typ" = 2 then    point_on_surface( $geometry)
»    »    else centroid($geometry)
»    end
»    )
»    /* Anhand der Wurzel der  Flächengrößen */
     if("schr_gr"  is NULL,
»    case
»    »    when sqrt($area ) /  length( "gen") > 1500 then 1500
»    »    when sqrt($area ) / length( "gen")  < 500 then 500
»    »    else sqrt($area ) / length( "gen")
»    end
     ,"schr_gr"
     )
```

Keine Angst vor sperrigen Ausdrücken

8 Räumliche Abfragen und Attributübertragungen

```
/* Leistung der WKA aummieren */

 coalesce(
 array_sum(
    overlay_intersects('kraftwerke' ,
                       expression:= "mw",
                       filter:=  "quelle" = 'wind'
                       )
                    ),0
                    )

                    --oder
coalesce(
aggregate(
      aggregate:=  'sum',
      layer:=      'kraftwerke',
      expression:= "mw",
      filter:=     "quelle" = 'wind'
                   and
                   intersects( $geometry, geometry(@parent))
                   ),0
                   )
```

9 Filterung im DropDowmenü

Keine Angst vor sperrigen Ausdrücken

10 Kreisbogen für Beschriftung im Geometriegenerator

```
-- Kreisbögen generieren im Geometriegenrator
translate(
        difference(
            boundary(
                difference(
                    (buffer( $geometry,1200,45)),
                        make_polygon(
                            geom_from_wkt( 'Linestring('      || ($x +1600) ||' '||   ($y +800) ||
                                         ', '  || ($x +1600) ||' '||   ($y -1600) ||
                                         ', '  || ($x -1600) ||' '||   ($y -1600) ||
                                         ', '  || ($x -1600) ||' '||   ($y +800) ||
                                         ', '  || ($x +1600) ||' '||  ($y + 800) || ')'
                            )
                        )
                    )
                ),
            geom_from_wkt('Linestring('  || ($x +1600) ||' '||   ($y +800) || ' '  || ($x - 1600) ||' '||   ($y +800)  || ')')
                    )
     ,0,-700)
```