序列脚本

create SEQUENCE if not exists alf\_alertfacil\_p\_fid\_seq start 1

查询序列当前值

select last\_value from alf\_alertfacil\_p\_fid\_seq

更新序列当前值

alter sequence emergdata.alf\_alertfacil\_p\_fid\_seq restart with 222

序列迁移视图函数：

|  |
| --- |
| CREATE OR REPLACE FUNCTION generateSeq() RETURNS text  AS $$  #print\_strict\_params on  DECLARE  lastv text;  res text ;  seq text ;  BEGIN  res = '';  FOR seq IN SELECT sequencename FROM pg\_catalog.pg\_sequences where schemaname = 'emergdata' LOOP  EXECUTE **'select last\_value from ' || quote\_ident(seq)** INTO lastv;  res = res || 'create SEQUENCE if not exists ' || quote\_ident(seq) || ' start 1;';  res = res || 'alter sequence emergdata.alf\_alertfacil\_p\_fid\_seq restart with ' || lastv || ';';  RAISE NOTICE 'Refreshing materialized view %s ...', quote\_ident(res);  END LOOP;    -- select last\_value INTO STRICT lastv  -- from alf\_alertfacil\_p\_fid\_seq;  RETURN res;  END  $$ LANGUAGE plpgsql;  --  SELECT generateSeq(); |

**Ps：1、通过or replace 循环创建视图、表、函数、序列、**

**2、通过**RAISE NOTICE ‘log’ 关键字**打印日志（%s 字符串占位符）**

**3、拼接字符串 通过 ||**

**4、声明变量**DECLARE 关键字后声明，格式为 name type；需要被初始化否则打印null

5**、动态sql 格式为**

EXECUTE **sql**

[ INTO [STRICT] ***target*** ]

[ USING ***expression*** [, ... ] ];

**Sql ：执行sql 注意：普通参数试用$1 $2 … 分别对应using后的变量，表明等通过quote\_ident（v） 函数进行拼接。**

**Target：返回结果赋值变量**

**Using：动态参数列表 通过$ 进行调用；**

**6、循环查询结果**

FOR ***target*** IN ***query*** LOOP

***statements***

END LOOP

**Target：待赋值变量**

**Query ：查询sql**

**复杂对象声明：**mviews **RECORD**;

**调用属性可通过** mviews**.propertyname 进行**