



链滴

postgre 函数整理

作者: [SummerWind](#)

原文链接: <https://ld246.com/article/1576759101351>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

1. **phis2zx_number** 返回int8类型的整型值, postgres中最大的整型值为numeric, 返回值可设为该类型

```
/*
 * 处理异常数据
 * 比如 传入的值为'|||
 */
CREATE OR REPLACE FUNCTION phis2zx_number(
    IN in_inputstr text          -- 传入的原始字符串
)
RETURNS int8
LANGUAGE plpgsql
AS $function$


/*
 * 函数说明：返回正常数字类型
 * 参数：按要求填写参数
 * * 返回：累加值
 */
declare
    -- 入参说明 '|||'/'|||'
    -- 程序变量说明
    v_result boolean default false;
begin
    -----将异常全部处理掉-----
    --需要匹配的字符串
    if(in_inputstr is null or in_inputstr = '')then
        return 0;
    end if;
    --只匹配数字类型
    select in_inputstr ~* '\d+' into v_result;
    --未匹配到 默认回0
    if(v_result is false) then
        return 0;
    end if;
    -----
    --正常数据
    return cast(substring(substring(in_inputstr from '\d+'),0,20) as numeric);
end
$function$;
```

2. **phis2zx_split_part_plus** 针对**split_part**函数封装 排除错误输入的text值

```
CREATE OR REPLACE FUNCTION phis2zx_split_part_plus(
```

```

IN in_inputstr text,          -- 传入的原始字符串
IN in_matchregex varchar(64), --需要匹配的字符串 \u0001 \u0027 |
in in_subindex integer --需要匹配的下标值
)

RETURNS TEXT
LANGUAGE plpgsql
AS $function$

/*
* 函数说明：返回第一个下标或者第二个下标memo值
* 参数：按要求填写参数
* * 返回：累加值
*/
declare
-- 入参说明 2\x01du\u0027le\u0027sou\u0027ru
-- 程序变量说明
v_bizopt record;      -- 选项记录集
v_othertext TEXT DEFAULT '';
v_matchcount int4 default 0;
begin

--将异常全部处理掉
--需要匹配的字符串
if(in_inputstr is null or in_inputstr = "")then
    if(in_subindex = 1) then
        return 1;
    else
        return ""; --不匹配 直接返回空字符串 (确实不知道是什么东西，有可能是垃圾数据)
    end if;
end if;
--没有需要匹配的
if(in_matchregex is null or in_matchregex = "")then
    if(in_subindex = 1) then
        return 1;
    else
        return ""; --不匹配 直接返回空字符串 (确实不知道是什么东西，有可能是垃圾数据)
    end if;
end if;

--判断当前需要截取的字符串中是否包含需要匹配的字符串 且为匹配值为2
select strpos(in_inputstr,in_matchregex) into v_matchcount;
if(v_matchcount = 0 or v_matchcount <> 2)then
    if(in_subindex = 1) then
        return 1;
    else
        return ""; --不匹配 直接返回空字符串 (确实不知道是什么东西，有可能是垃圾数据)
    end if;
end if;

-----
--正常数据
return split_part(in_inputstr,in_matchregex,in_subindex);

```

```
end
```

```
$function$
```

```
;
```

3. zx_transfer2zxfield 数据库json格式字段转换存储过程(特例)

```
/*
 * 該存儲過程對應數據庫兩個字段 分別是drugusage: 用藥情況、drugguide: 用藥指導（兩個字
使用的json格式一致）外部需要傳遞0,1 加以區分 得到自己想要的字段
*/
CREATE OR REPLACE FUNCTION zx_transfer2zxfield (
    IN in_json_field jsonb,          -- 选项JSON数组
    IN in_usagetype varchar(64)      -- 0用药情况, 1用药指导
)
RETURNS jsonb
LANGUAGE plpgsql
AS $function$
```

```
/*
 * 函数说明：将公衛用藥情況/用药指导的字段轉換為数据库字段（僅限用於JSON數組對象格式）
 * 参数：按要求填写参数
 * in_json_field :
 * [{"CmDrugID": "药物ID", "Drugs": "药物名称", "DailyTimes": "每日用药次数", "EachDose": "每次用
剂量值", "Remark": "每次用药剂量单位", "Remark1": "用药备注", "UsageType": "0用药情况, 1用药指
"}]
 * * 返回：累加值
 * [{"drugfreq": "用藥頻率", "drugmemo": "藥物備注", "drugname": "藥物", "drugdosage": "藥
用量", "drugunitcode": "計量單位編碼", "drugunitname": "計量單位名稱"}]
*/
declare
    -- 入参说明
    -- 程序变量说明
    v_bizopt record;      -- 选项记录集
    v_othertext TEXT DEFAULT '';
    v_rec record;
    v_jsonb array jsonb default '{}';
    v_json_field jsonb default null;
begin
```

```
    for v_rec in
        select
            drugs as drugname,
            dailytimes as drugfreq,
            eachdose as drugdosage,
            remark as drugunitname,
            '' as drugunitcode,
            remark1 as drugmemo,
            usagetype as drugtype -- 0用药情况, 1用药指导
        from (select * from jsonb_to_recordset(phis61.phis2field_lower(in_json_field)) AS
```

```

t(
    cmdrugid varchar(64), -- 药物ID
    drugs varchar(64), -- 药物名称
    dailytimes varchar(64), -- 每日用药次数
    eachdose varchar(64), -- 每次用药剂量值
    remark varchar(64), -- 每次用药剂量单位
    remark1 varchar(64), -- 用药备注
    usagetype varchar(64) --0用药情况, 1用药指导
) where t.usagetype = in_usagetype
) a
loop
    select jsonb_build_object (
        'drugfreq', v_rec.drugfreq,
        'drugmemo',v_rec.drugmemo,
        'drugname',v_rec.drugname,
        'druggdosage',v_rec.druggdosage,
        'drugunitcode',v_rec.drugunitcode,
        'drugunitname',v_rec.drugunitname
    ) into v_json_field;
    select jsonb_insert (v_jsonb_array,'{0}',v_json_field, false) into v_jsonb_array;
end loop;
RETURN v_jsonb_array;
end
$function$;

```

这个函数需要注意的是：phis61 phis2field_lower() postgreSQL的jsonb to recordset不能区分大小写，需要手动转换成固定的小写 才能获取到数据

4. phis2field_lower 转小写函数

```

/**
 * 本存储过程实现：
 * 将省公卫平台业务数据字段大小寫全部轉成小寫jsonb格式返回
 */

CREATE OR REPLACE FUNCTION phis2field_lower (
    IN in_jsonarray_field jsonb          -- 省公卫平台json格式數組
)
RETURNS jsonb
LANGUAGE plpgsql
AS $function$

/*
 * 函数说明：
 * 参数：按要求填写参数
 * in_jsonarray_field :
 * [{"CmDrugID":"药物ID","Drugs":"药物名称","DailyTimes":"每日用药次数","EachDose":"每次药剂量值","Remark":"每次用药剂量单位","Remark1":"用药备注","UsageType":"0用药情况, 1用药

```

```
导")]
*
* * 返回：平台统一格式的uuid（32位小写字符，无中间连字符）
*/
declare
    -- 入参说明
    -- 程序变量说明
    v_text text default '';
begin
    v_text := (in_jsonarray_field)::text;
    -- 如果选项无值，则直接返回 0
    IF (v_text IS NULL OR trim(v_text) = " ") THEN
        return -6;
    END IF;
    return (lower(replace(trim(v_text), '-', '')))::jsonb;
end

$function$;
```

如需转载 请注明本人。