

1 Création des fonctions de comparaisons à partir d'une table

Les programmes sont destinés à être utilisés avec PostgreSQL. Ils sont compatibles au moins avec les versions 11.10 et 12 (liste non exhaustive).

Pour créer les fonctions de comparaison à partir d'une table, vous devez charger le fichier `functionTable.sql` en entier.

Vous devez ensuite créer une table **functionTable**.

```
create table functionTable (name text, condition text,
result bit varying);
```

La colonne **name** doit contenir le nom des fonctions sans le premier « f », il sera rajouté lors de la création de la fonction. Dans l'exemple ci-dessous, la procédure créera une fonction **flevel** et **fgender**.

La colonne **condition** doit contenir une condition avec les paramètres.

Enfin, la colonne **result** doit contenir la chaîne de bits renvoyées lorsque la condition est respectée.

Un exemple de `functionTable` :

```
( 'level', 'x_numeric, y_numeric' );
( 'level', 'x=y or ((0<=x and x<2) and (0<=y and y<2))', '11' );
( 'level', '2<=x and x<5 and 2<=y and y<5 and x!=y', '10' );
( 'level', '0<=x and x<2 and 2<=y and y<5', '01' );
( 'level', '2<=x and x<5 and 0<=y and y<2', '01' );
( 'level', '', '00' );
( 'gender', 'x_char(1), y_char(1)' );
( 'gender', 'x=y', '1' );
( 'gender', 'x!=y', '0' );
```

Il faut un unique tuple par **name** avec un **result** null. La condition de ce tuple sera les paramètres de la fonction.

Vous pouvez si vous le souhaitez mettre un unique tuple avec une condition ''. Le **result** de ce tuple sera le résultat renvoyé si aucune condition n'est satisfaite.

Pour créer les fonctions de comparaisons, il vous reste à appeler la procédure `createfunctions()`.

Les fonctions de comparaisons créées par la `functionTable` ci-dessus seront donc :

```
CREATE OR REPLACE FUNCTION flevel(x numeric, y numeric)
RETURNS bit AS $$
SELECT
  CASE
```

```

        WHEN x=y or ((0<=x and x<2) and (0<=y and y<2)) THEN b'11'
        WHEN 2<=x and x<5 and 2<=y and y<5 and x!=y THEN b'10'
        WHEN 0<=x and x<2 and 2<=y and y<5 THEN b'01'
        WHEN 2<=x and x<5 and 0<=y and y<2<5 THEN b'01'
        ELSE b'00'
    END
$$ LANGUAGE sql;

CREATE OR REPLACE FUNCTION fgender(x char(1), y char(1))
RETURNS bit AS $$
SELECT
    CASE
        WHEN x=y THEN b'1'
        WHEN x!=y THEN b'0'
    END
$$ LANGUAGE sql;

```