Créez vos fonctions SQL.

- Les articles -

Date de mise en ligne : vendredi 3 décembre 2004

Description :

Vous pouvez créer dans une base DB2, des fonctions spécifiques à vos besoins. C'est le sujet de cet article.

Environnement iSeries

Vous pouvez créer dans une base DB2, des fonctions spécifiques à vos besoins (user defined function). C'est le sujet de cet article.

Il est basé sur un exemple rudimentaire, juste pour aborder le sujet. Les liens en fin d'article vous permetront d'aller plus avant.

Rappel

Une expression est :

- un ensemble de variables (contenu d'une colonne),
- de constantes
- de fonctions combinées au moyen d'opérateurs.

Les expressions peuvent figurer :

- en tant que colonne résultat d'un SELECT,
- dans une clause WHERE,
- dans une clause ORDER BY.

Une fonction :

- retourne une valeur dépendant de ses arguments qui peuvent être eux-mêmes des expressions.

SQL offre un grand nombre de fonctions :

- arithmétiques
- chaines de caractères
- dates
- conversions
- autres

Procédons via un exemple concret.

Données servant à l'exemple

- Une table t_article contient les articles et l'unité de vente utilisée. Par exemple l'article 0000001 représente un tissu.
- Une table t_prix contient le prix des articles dans leur unité de vente par défaut. Par exemple l'article 0000001 (du

tissu) est vendu 20 Euros le mètre linéaire.

- Une table t_unite contient les unités de vente possibles pour un article. Par exemple l'article 0000001 (du tissu) peut être vendu au mètre linéaire, ou au rouleau, avec le coefficient arithmétique permettant de convertir une unité en une autre.
- Le prix au mètre linéaire est de 20 Euros (stocké dans t_prix)
- Un rouleau représente 50 mètres linéaires (le coefficient 50 est stocké dans t_unité)

Problème posé

- Editer une liste des articles et des prix de vente au rouleau.
- Si le prix est stocké au rouleau, il suffit de le récupérer dans t_prix. Si le prix est stocké au mètre linéaire, il faut le récupérer dans t_prix et y appliquer une règle de trois pour passer du mètre linéaire au rouleau.

1ère solution

- Déclarer un curseur qui lira les articles qui peuvent êtres vendus au rouleau, l'ouvrir, le lire
- Pour chaque article, traiter les unités pour appliquer la règle de trois.

2nde solution

- Créer une fonction SQL qui sera chargée de calculer le prix après règle de 3, en fonction de l'article, et de l'unité de vente passée en paramètre.
- Utiliser la fonction SQL pour construire votre requête.

Vous l'avez deviné (mais ce n'était pas difficile), c'est cette seconde solution qui est traitée dans cet article.

Le code source de la fonction SQL

Commencez par créer une bibliothèque sur votre système AS400. Appelez la par exemple BIBSQLFUN.

```
CRTLIB LIB(BIBSQLFUN) TEXT('Bib de fonctions SQL pour mon applicatif')
```

C'est dans cette bibliothèque que le système créera le programme de service correspondant à votre fonction.

Créer la fonction REGLE3

- 1er paramètre : le code article.
- 2nd paramètre : l'unité de vente souhaitée.
- recherche du prix de l'article et de l'unité de vente par défaut dans t_prix
- si l'unité de vente souhaitée est égale à l'unité de vente par défaut, le coefficient appliqué par la règle de trois sera de 1.
- si l'unité de vente souhaitée est différente de l'unité de vente par défaut, le coefficient à appliquer est cherché dans t_unite.
- la fonction retourne le prix de l'article, dans l'unité de vente passée en paramètre.
- Je vous conseille d'intégrer la création de la fonction dans un script SQL que vous stockerez dans un membre source ou dans un programme de type rpg ile.

```
CREATE FUNCTION BIBSQLFUN/REGLE3 (
VARTICLE CHAR(7)  ,
VUNITEVTE CHAR(2) ,
)
RETURNS DECIMAL(20, 9)
LANGUAGE SQL
MODIFIES SQL DATA
BEGIN
DECLARE VPRIX DECIMAL(20, 9);
DECLARE VCOEFF DECIMAL(20, 9);
DECLARE VUNITE CHAR(2);

SET VPRIX   = 0 ;
SET VCOEFF = 0 ;

SELECT prix, unite
INTO VPRIX , VUNITE
FROM t_prix
WHERE article = VARTICLE ;
END IF ;

IF VUNITEVTE = VUNITE THEN
SET VCOEFF = 1 ;
ELSE
SELECT coeff
INTO VCOEFF
FROM t_unite
WHERE article = VARTICLE
AND unite = VUNITEVTE ;
END IF ;

SET VPRIX = VPRIX * VCOEFF ;

RETURN VPRIX ;

END
```

Utiliser la fonction REGLE3

- Il vous est impossible de passer une constante en paramètre de la fonction. Les paramètres sont le contenu d'une colonne ou une variable hôte.
- Vous devez indiquez à SQL le schéma (ici BIBSQLFUN) où aller chercher la fonction REGLE3.
- l'exemple ci-dessous récupère le prix arrondi à deux décimales, dans la variable hôte prix pour l'article 0000001 (variable hôte article) et l'unité rouleau (variable hôte unite).

```
EXEC SQL
SET OPTION SQLPATH = 'BIBSQLFUN'
END-EXEC
unite = 'RL'
article = '0000001'
prix = 0
EXEC SQL
```

```
SELECT ROUND(REGLE3(a.article , :unite) , 2 )
INTO :prix
FROM t_article a
WHERE a.article = :article
END-EXEC
```

Comment voir les paramètres de la fonction créée

```
SELECT ROUTINE_NAME,SPECIFIC_NAME,SPECIFIC_SCHEMA,
IN_PARMS, OUT_PARMS, INOUT_PARMS
FROM QSYS2/SYSROUTINES WHERE ROUTINE_NAME LIKE 'REGLE3%'
```

Comment supprimer la fonction

```
DROP SPECIFIC FUNCTION BIBSQLFUN/REGLE3
```

Vous n'arrivez pas à tester la variable SQLSTATE ?

SQLSTATE n'est pas une variable automatiquement déclarée. Vous devez la déclarer sur une longueur de 5 charactères :

```
DECLARE SQLSTATE CHAR(5);
```

Pour tester si une lecture à aboutie par exemple, vous pouvez utiliser SQLSTATE comme suite :

```
IF SQLSTATE <> '02000' THEN
SET FL1 = 'O' ;
END IF;
```

Les documentations utiles

- User defined function
- Create function
- DB2 UDB for AS/400 Visual Explain
- SQL Reference volume 2 version 8
- SQL Getting started version 7