

15.3 Функции, дефинирани от потребителя (UDF)

Функциите, дефинирани от потребителя (UDF), са обекти за работа с бази от данни, които съпоставят на множество от входни данни множество от изходни данни. Например, такава функция може да приема някакви стойности, измерени в инчове и като резултат да връща стойностите в сантиметри.

В DB2 функции могат да се създават чрез SQL PL, C/C++, Java, CLR (Common Language Runtime) и OLE (Object Linking and Embedding). В тази книга ще се фокусираме върху функции, написани на SQL PL, заради тяхната простота, популярност и бързодействие.

Има четири типа функции: скалярни, върху таблици, върху редове и върху колони. В тази глава ще се фокусираме върху скалярните функции и функциите върху таблици.

15.3.1 Скалярни функции

Скалярните функции връщат единична стойност. Те не могат да включват команди, които променят състоянието на базата от данни, т.е. не са позволени командите INSERT, UPDATE, и DELETE. Някои вградени скалярни функции са SUM(), AVG(), DIGITS(), COALESCE(), и SUBSTR().

В DB2 е позволено сами да създадете специфична функция, която да енкапсулира често използвана логика. Например, да разгледаме мигрирането на вашата програма от Oracle към DB2. Във вашето приложение често използвате функцията от Oracle NVL(). Еквивалентната вградена функция в DB2 е COALESCE. Вместо да преименувате всички срещания на функцията NVL във вашата програма може да дефинирате функция в DB2 и да я наречете NVL. Тя ще извиква функцията COALESCE както е показано долу.

```
CREATE FUNCTION NVL (p_var1 VARCHAR(30),
                    p_var2 VARCHAR(30))
SPECIFIC nvlvarchar30
RETURNS VARCHAR(30)
RETURN COALESCE(p_var1, p_var2)
```

Функцията COALESCE връща първия аргумент, който не е Null.

Долу е показан друг пример за скалярна функция. Тя се нарича "deptname" и връща номера на отдела на всеки служител, използвайки идентификатора на служителя.

```
CREATE FUNCTION deptname(p_empid VARCHAR(6))
RETURNS VARCHAR(30)
SPECIFIC deptname
BEGIN ATOMIC
    DECLARE v_department_name VARCHAR(30);
    DECLARE v_err VARCHAR(70);
    SET v_department_name = (
        SELECT d.deptname FROM department d, employee e
        WHERE e.workdept=d.deptno AND e.empno= p_empid);
    SET v_err = 'Error: employee ' || p_empid || ' was not
```

```

found';
IF v_department_name IS NULL THEN
    SIGNAL SQLSTATE '80000' SET MESSAGE_TEXT=v_err;
END IF;
RETURN v_department_name;
END

```

За да тествате функцията, напишете следното в Command Window или Linux/UNIX shell:

```
db2 "values (deptname ('000300'))"
```

Извикване на скаларни функции

Скаларни потребителски функции могат да бъдат извиквани в SQL команди, където се очаква върнатата стойност да е скаларна, както и в клаузата VALUES. Дадени са два примера, които показват извикването на скаларната функция COALESCE:

```

SELECT DEPTNAME, COALESCE(MGRNO, 'ABSENT') FROM DEPARTMENT

VALUES COALESCE('A', 'B')

```

15.3.2 Функции върху таблици

Функциите върху таблици връщат като резултат таблица от редове. Те могат да се извикват в клаузата FROM на заявката. За разлика от скаларните функции, функциите върху таблици могат да променят състоянието на базата от данни, следователно могат да се използват командите INSERT, UPDATE и DELETE. Някои вградени функции върху таблици са SNAPSHOT_DYN_SQL() и MQREADALL(). Функциите върху таблици са подобни на изгледите, но са доста помощно средство, тъй като в тях могат да се използват командите за промяна на данните (INSERT, UPDATE и DELETE). Най-често тези функции се използват за връщане на таблица като резултат и записване на промените за проверка.

Следващият пример представя функция, която показва служителите от даден отдел:

```

CREATE FUNCTION getEnumEmployee(p_dept VARCHAR(3))
RETURNS TABLE
    (empno CHAR(6),
     lastname VARCHAR(15),
     firstnme VARCHAR(12))
SPECIFIC getEnumEmployee
RETURN
    SELECT e.empno, e.lastname, e.firstnme
    FROM employee e
    WHERE e.workdept=p_dept

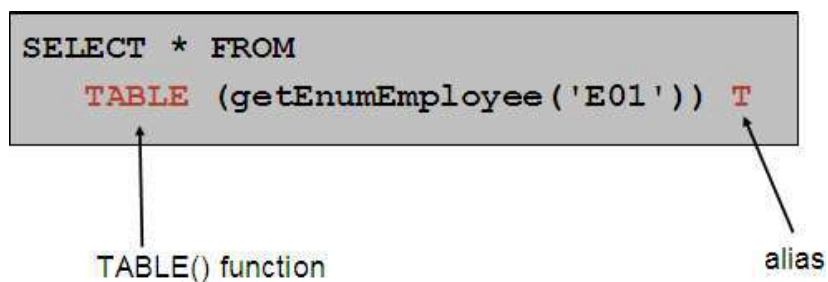
```

За да проверите горната функция, пробвайте:

```
db2 "SELECT * FROM table(getEnumEmployee('D11')) AS t"
```

Извикване на функции върху таблици

Функциите върху таблици трябва да бъдат извикани в клаузата FROM на SQL заявката. Трябва да се използва функцията TABLE() и на резултата да бъде даден псевдоним. На Фигура 15.2 е показано как да се извика функцията "getEnumEmployee", която пробвахме по-рано.



```
SELECT * FROM  
TABLE (getEnumEmployee('E01')) T
```

TABLE() function

alias

Фигура 15.2 – Извикване на функция върху таблица.

Лабораторно упражнение #13 – Създаване на потребителски функции в IBM Data Studio

Цел:

В това Лабораторно упражнение ще създадете скаларна потребителска функция в IBM Data Studio. Така ще придобиете повече опит с Data Studio и ще подобрите познанията си за функциите, дефинирани от потребителя в езика SQL PL.

Стъпки:

1. Отворете IBM Data Studio (Съвет: може да го намерите в Start menu).
2. От прозореца Data Project Explorer, изберете проекта, който създадохте в предишното лабораторно упражнение и изберете *Open Project*.
3. Натиснете с десен бутон върху папката *User-Defined Functions*. Изберете *New* от менюто. Изберете *SQL User-Defined Function* от менюто. Може да изберете и *User-Defined Function using Wizard* от менюто, ако искате да използвате помощника.
4. Ще се отвори изгледът *Editor* с примерна функция. Променете кода, както следва:

```
CREATE FUNCTION booktitle(p_bid INTEGER)
RETURNS VARCHAR(300)
-----
SQL UDF (Scalar)
-----
SPECIFIC booktitle
F1: BEGIN ATOMIC
DECLARE v_book_title VARCHAR(300);
DECLARE v_err VARCHAR(70);
SET v_book_title = (SELECT title FROM books WHERE p_bid =
book_id);
SET v_err = 'Error: The book with ID ' || CHAR(p_bid) || '
was not found.';
IF v_book_title IS NULL THEN SIGNAL SQLSTATE '80000' SET
MESSAGE_TEXT=v_err;
END IF;
RETURN v_book_title;
END
```

5. Създайте функцията като кликнете с десен бутон на функцията и дадете *Deploy*.
6. Стартирайте функцията, като натиснете бутона *Run* в лентата с инструменти.
7. Тъй като функцията приема един входен параметър, ще се отвори диалогов прозорец, който изисква въвеждането на стойност за параметъра.

Въведете стойността: 80002

Какъв е резултатът?

Пробвайте отново със стойността: 1002

А какъв е резултатът сега? (Съвет: Вижте в раздела *Messages* на изгледа *Output*).

8. След като свършите затворете IBM Data Studio.