

ARBORI – ALGORITM DE PARCURGERE INVERSA

LECT. DRD. HORIA DEMIAN

LECT. DRD. CARMEN POP

Universitatea din Oradea, Facultatea de Științe Economice, str Universitatii nr. 1, Tel 0259-432830

E-mail: horia_demian@yahoo.com; cpop@uoradea.ro

This paper presents the an algorithm for gathering nodes from a tree which came from bottom to top, using SQL Server 2000. This algorithm was used in conQuest application to solve a specific problem which is explained in this article.

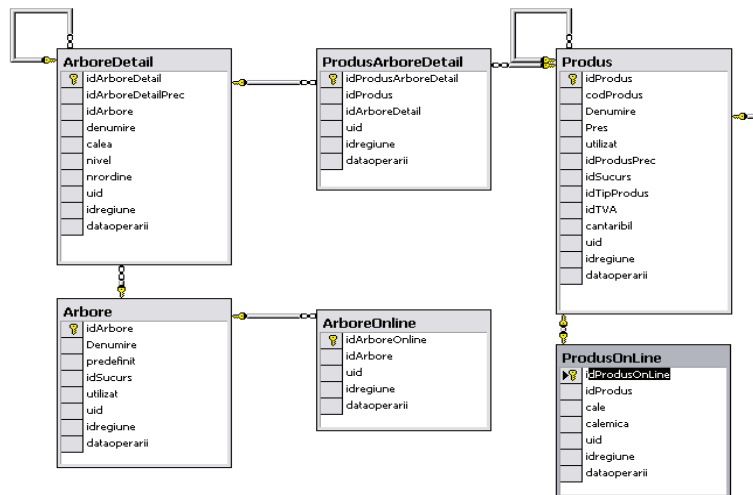
Obiectivul

Acest articol prezinta un algoritm de filtrare al nodurilor unui arbore pornand de la nodurile frunza spre cele parinte

Baza de date:

Baza de date este formata din urmatoarele tabele:

- Arbore - tabel utilizat la memorarea nodurilor radacina pentru arborii existenti.
- ArboreDetail – face descrierea fiecarui nod al arborelui precizand, denumirea, nivelul pe care se gaseste nodul parinte la care este legat, si pozitia in cadrul nodurilor de acelasi nivel
- Produs – tabel utilizat la memorarea produselor existente in catalogul de produse
- ProdusArboreDetail – tabel utilizat pentru legarea produselor la nodurile arborelui
- ProdusOnline – contine descrierea produselor ce sunt publicate online
- ArboreOnline – utilizat pentru precizarea arborilor ce vor fi publicati online



Obiectivul este acela de a filtra nodurile arborelui astfel incat vor apare doar nodurile la care sunt legate produse publicate, sau noduri ce au noduri copii cu produse publicate. De exemplu:

Reprezentarea in baza de date	Rezultatul
<ul style="list-style-type: none"> • Nod A <ul style="list-style-type: none"> ○ Nod A. 1. <ul style="list-style-type: none"> ▪ Nod A.1.1. <ul style="list-style-type: none"> • Produs 1 • Produs 2 ▪ Nod A.1.2. ○ Nod A.2. <ul style="list-style-type: none"> ▪ Nod A.2.1. ▪ Nod A.2.2. <ul style="list-style-type: none"> • Produs 3 	<ul style="list-style-type: none"> • Nod A <ul style="list-style-type: none"> ○ Nod A. 1. <ul style="list-style-type: none"> ▪ Nod A.1.1. <ul style="list-style-type: none"> • Produs 1 • Produs 2 ○ Nod A.2. <ul style="list-style-type: none"> ▪ Nod A.2.2. <ul style="list-style-type: none"> • Produs 3

Algoritmul:

Pentru rezolvarea acestei probleme in SQL Server 2000, s-a utilizat o functie, al carui rol este acela de a compune toate nodurile arborelui, ce au legata spre produse publicate.

```

CREATE FUNCTION dbo.FN_IERARHIEARBOREONLINE ( @idArbore UNIQUEIDENTIFIER)
RETURNS @IERARHIETEMP TABLE (
    ,IDARBOREDETAIL          UNIQUEIDENTIFIER
    ,IDARBOREDETAILPREC    UNIQUEIDENTIFIER
    ,IDARBORE              UNIQUEIDENTIFIER
    ,NIVEL                 INT
    ,NRORDINE              INT
)
AS
BEGIN
    DECLARE @IERARHIENOD TABLE (
        ,IDARBOREDETAIL          UNIQUEIDENTIFIER
        ,IDARBOREDETAILPREC    UNIQUEIDENTIFIER
        ,IDARBORE              UNIQUEIDENTIFIER
    )
    DECLARE @IERARHIEPREC TABLE (
        ,IDARBOREDETAIL          UNIQUEIDENTIFIER
        ,IDARBOREDETAILPREC    UNIQUEIDENTIFIER
        ,IDARBORE              UNIQUEIDENTIFIER
    )
    DECLARE @IERARHIEDEST TABLE (
        ,IDARBOREDETAIL          UNIQUEIDENTIFIER
        ,IDARBOREDETAILPREC    UNIQUEIDENTIFIER
        ,IDARBORE              UNIQUEIDENTIFIER
    )
    DECLARE @KOD              SMALLINT
    DECLARE @ERORI            INT
    DECLARE @ROWCOUNT        INT
    SET @ERORI                = 0
    SET @KOD                  = 0
    --INSERAREA NODURILOR DE PLECARE(cele de ultim nivel pentru care exista produse publicate)
    INSERT INTO @IERARHIENOD (
        ,IDARBOREDETAIL
        ,IDARBOREDETAILPREC
    
```

```

),IDARBORE
)
SELECT
    AD.IDARBOREDETAIL
    ,AD.IDARBOREDETAILPREC
    ,AD.IDARBORE
FROM
    ARBOREDETAIL AD
    inner Join ProbusArboreDetail PAD
        on AD.idArboreDetail = PAD.idArboreDetail
    Inner Join Probus P
        on PAD.idProbus = P.idProbus
    Inner Join ProbusOnline PO
        on p.idProbus = PO.idProbus
WHERE AD.IDARBORE=@idArbore
and not exists (Select * --conditia pentru a fi noduri de ultim nivel
                from ArboreDetail ADPREC
                where ADPREC.idArboreDetailPrec = AD.idArboreDetail
                and ADPREC.idArboreDETailprec <> ADPREC.idArboreDetail
                )
)

INSERT INTO @IERARHIEDEST --ACEASTA VA CONTINE DOAR parintii
SELECT
    IDARBOREDETAIL
    ,IDARBOREDETAILPREC
    ,IDARBORE
FROM @IERARHIENOD

WHILE (@KOD = 0 ) BEGIN
    --CAT TIMP MAI EXISTA NODURI CE FAC REFERINTA CATRE NODURI DIN PARINTE
    --CONTINUAM PARCURGAREA ARBORELUI
    DELETE FROM @IERARHIEPREC --GOLIREA BUFFERULUI DE NODURI COPII
    --Vom obtine lista nodurilor de nivel superior ce nu au mai fost inserate
    INSERT INTO @IERARHIEPREC
    SELECT DISTINCT
        A.IDARBOREDETAIL
        ,A.IDARBOREDETAILPREC
        ,A.IDARBORE
    FROM ARBOREDETAIL A INNER JOIN @IERARHIENOD S
    ON A.IDARBOREDETAIL = S.IDARBOREDETAILPREC
    AND S.IDARBOREDETAILPREC<>S.IDARBOREDETAIL
    WHERE NOT EXISTS (SELECT * FROM @IERARHIEDEST id
                      WHERE
                      A.IDARBOREDETAIL=id.IDARBOREDETAIL)

    IF @@ERROR<>0 OR @@ROWCOUNT=0 BEGIN
        SET @KOD = 1 --NU MAI SUNT COPII
    END
    INSERT INTO @IERARHIEDEST
    SELECT
        IDARBOREDETAIL
        ,IDARBOREDETAILPREC
        ,IDARBORE
    FROM @IERARHIEPREC
    --GOLIREA BUFFERULUI DE NODURI
    DELETE FROM @IERARHIENOD
    --COPIEREA COPIILOR IN NOD
    INSERT INTO @IERARHIENOD
    SELECT
        IDARBOREDETAIL
        ,IDARBOREDETAILPREC
        ,IDARBORE
    FROM @IERARHIEPREC

END
IF @ERORI = 0 BEGIN
    INSERT INTO @IERARHIETEMP

```

```

SELECT DISTINCT
    SD.IDARBOREDETAIL
    ,SD.IDARBOREDETAILPREC
    ,SD.IDARBORE
    ,AD.NIVEL
    ,AD.NRORDINE
FROM @IERARHIEDEST SD INNER JOIN ARBOREDETAIL AD
    ON SD.IDARBOREDETAIL=AD.IDARBOREDETAIL
ORDER BY AD.NIVEL,AD.NRORDINE
END
RETURN
END

```

Apelul acestei functii se poate face din interiorul comenzii SELECT , functia primind pe post de argument identificatorul arborelui ce se doreste a fi filtrat.

Exemplu de apel este urmatorul:

```

SELECT *
FROM
    dbo.fn_IerarhieArboreOnline(' D6F24737-5F95-4BFA-9597-405DE2EBCBD2')

```