

SQL Update

Paul Roy

Common Europe Luxembourg 24/03/2016

SQL UPDATE WORKSHOP

- Agenda
 - SQL review
 - CTE , Recursive, Hierarchical queries
 - SQL system access with SQL Services

SQL review

- Objects
 - Tables , views ,Index
- DDL
 - CREATE (OR REPLACE)
 - ALTER
 - DROP
- DML
 - INSERT
 - UPDATE
 - SELECT
- Embed in RPGLE or COBOL
 - CURSOR
 - FETCH
- Procedures
 - Language
- SQLCI
 - Client interface (ODBC, JDBC,...)

CTE:Common Table Expression

Pseudo view valid only in the current SELECT statement

WITH

```
cte1 AS (SELECT a,b,c FROM TAB1) ,
```

```
cte2 AS
```

```
  (SELECT a,sum(b)as sumb, sum(c)as sumc
   FROM cte1
   GROUP BY a)
```

```
SELECT cte1.*, sumb, sumc
```

```
FROM cte1 JOIN cte2 ON cte1.a = cte2.a
```

CTE:Common Table Expression

```

WITH
ta AS (SELECT COUNT(*) AS tota
      FROM AOFSQL_5/AOFDOCF ) ,
tp AS (SELECT dodonm, COUNT(*) AS totp
      FROM AOFSQL_5/AOFDOCF
      GROUP BY dodonm)
SELECT tp.dodonm, totp, tota,
CAST(totp*100.00/tota AS DECIMAL(5, 2)) as U
FROM tp, ta
ORDER by U DESC, tp.dodonm

```

Recursive SQL queries

- The key to the next is another column in the row
 - Parent →Child
- Examples
 - Program tree
 - Folders
- Hierarchical relationship (organigram, bom,..)
- Bi directional relationships (routes on map)

Recursive SQL query

- Initialization SELECT
 - Part of the CTE
 - Declare the starting point
 - Is only executed ONCE
- Recursive SELECT
 - Part of the CTE
 - UNION ALL to combine all rows
 - Merge the CTE with itself (ex Calling/Called)
- Final/Main SELECT
 - Process the CTE

SQL sample Context

DSPPGMREF PGM(COMMON/*ALL)

OUTPUT(*OUTFILE)

OBJTYPE(*PGM *SRVPGM)

OUTFILE(COMMON/PGMREF)

➔TABLE PGMREF

- WHPNAM : CALLING program
- WHFNAM: CALLED Program

Sample 1

- What is my most called program ?

```
SELECT COUNT(*) AS USAGES, WHFNAM
FROM PGMREF
GROUP BY WHFNAM
HAVING COUNT(*) =
  ( SELECT MAX(USAGES)
    FROM
      (SELECT COUNT(*) AS USAGES, WHFNAM
       FROM PGMREF
        GROUP BY WHFNAM) SSS)
```

VERB
FUNCTION
COLUMN NAME
TABLE NAME
INTERNAL SQL NAME

Sample 1 using CTE

- CTE (Common Table Expression)
 - Creation d'une table /vue temporaire
- ```
WITH
TMPVIEW (USAGES, PROGRAMME)
AS
 (SELECT COUNT(*) AS USAGES, WHFNAM
 FROM PGMREF
 GROUP BY WHFNAM)
SELECT USAGES, PROGRAMME
FROM TMPVIEW
WHERE USAGES =
 (SELECT MAX(USAGES)
 FROM TMPVIEW)
```

## Sample 1 using 2 CTE's

- You can use as many CTE as you want ...

```

WITH
 TMPVIEWCOUNT (USAGES, PROGRAMME)
AS (SELECT COUNT(*) AS USAGES, WHFNAM
 FROM PGMREF
 GROUP BY WHFNAM)
, TMPVIEWMAX (MAXUSE)
AS (SELECT MAX(USAGES)
 FROM TMPVIEWCOUNT)
SELECT USAGES, PROGRAMME
FROM TMPVIEWCOUNT
INNER JOIN TMPVIEWMAX
ON USAGES = MAXUSE

```

## Sample 2

- Hierarchy/recursion
  - CTE Starting point request
  - UNION ALL ho to go to next
  - Main SELEECT
- Build the program tree
  - WHPNAM Calling program CMNU01
  - WHFNAM Called program

## Hierarchy

- Starting point
  - SELECT WHPNAM, WHFNAM FROM PGMREF WHERE WHPNAM = 'CMNU01'
- Destination
  - SELECT WHPNAM, WHFNAM FROM PGMREF
- UNION ALL
  - SELECT WHPNAM, WHFNAM FROM PGMREF WHERE WHPNAM = 'CMNU01'
  - UNION ALL SELECT WHPNAM, WHFNAM FROM PGMREF

## Combining the CTE

```

WITH
 PGMTREE (CALLING, CALLED)
AS (
 SELECT WHPNAM, WHFNAM
 FROM PGMREF
 WHERE WHPNAM = 'CMNU01'
 UNION ALL
 SELECT WHPNAM, WHFNAM
 FROM PGMREF
 | INNER JOIN PGMTREE
 ON CALLED= WHPNAM)
SELECT * FROM PGMTREE

```

## Hierarchical indendation

- Ordering by calling, called
- Create the order string

```

WITH
 PGMTREE (CALLING, CALLED, path)
AS (
 SELECT WHPNAM, WHFNAM ,
 cast(concat(whpnam , whfnam)as varchar(100)) as path
 FROM COMMON.PGMREF
 WHERE WHPNAM = 'CMNU01'
 UNION ALL
 SELECT WHPNAM, WHFNAM , concat(t.path , x.whfnam)
 FROM COMMON.PGMREF x
 INNER JOIN PGMTREE T
 ON CALLED= x.WHPNAM)

SELECT * FROM PGMTREE
order by path

```

## Hierarchical indendation

- Add the level

```

WITH
 PGMTREE (CALLING, CALLED, path ,level)
AS (
 SELECT WHPNAM, WHFNAM , cast(concat(whpnam , whfnam)as
 varchar(100)) as path , 1
 FROM COMMON.PGMREF
 WHERE WHPNAM = 'CMNU01'
 UNION ALL
 SELECT WHPNAM, WHFNAM , concat(t.path , x.whfnam) , level+1
 FROM COMMON.PGMREF x
 INNER JOIN PGMTREE T
 ON CALLED= x.WHPNAM)

SELECT * FROM PGMTREE
order by path

```



## Hierarchical indendation

- Display indentation

```

WITH
 PGMTREE (CALLING, CALLED, path ,level)
AS (
 SELECT WHPNAM, WHFNAM ,
 cast(concat(whpnam , whfnam)as varchar(100)) as path,0
 FROM COMMON.PGMREF
 WHERE WHPNAM = 'CMNU01' AND WHOBJT = 'P'
 UNION ALL
 SELECT WHPNAM, WHFNAM , concat(t.path , x.whfnam) , level+1
 FROM COMMON.PGMREF x
 INNER JOIN PGMTREE T
 ON CALLED= x.WHPNAM)
SELECT CONCAT(concat(SPACE(LEVEL*3), CALLING) , called)
FROM PGMTREE
ORDER BY path

```

## VALUES Statement

- Run SQL without a table
- VALUES can return
  - a single value
    - VALUES(sqlexpr1)
  - A table of values
    - VALUES(sqlexpr1,sqlexpr2,sqlexpr3),
   
          (sqlexpre4,sqlexpre5,sqlexpr6),
   
          (sqlexpr7, sqlexpr8,sqlexpr9)

## VALUES samples

- Values(WEEK\_ISO(Current\_Date))
- VALUES((Current\_date),  
(DAYNAME(Current\_Date)))
- VALUES('Date ISO', char(Current\_date,ISO)),  
( 'Date EUR' , char(Current\_date, EUR)),  
( 'Date USA' , char(Current\_date, USA))

## VALUES example

```
select * from
 (VALUES('Date ISO', char(Current_date,ISO)),
 ('Date EUR' , char(Current_date, EUR)),
 ('Date USA' , char(Current_date, USA))) Tb (Format,
sample)
```

## Recursive SQL examples

- Generate running numbers

```
WITH counter (N) as (select 0 from sysibm.sysdummy1
 UNION ALL
 Select n+2 from counter
 where n < 20)
select * from Counter
```

- LIST 5 next years

```
WITH years (ANNEE)
 as (VALUES (Year(Current_Date))
 UNION ALL
 Select ANNEE+1 from years
 where annee < Year(Current_Date) +5)
select * from years
```

## Recursive SQL examples

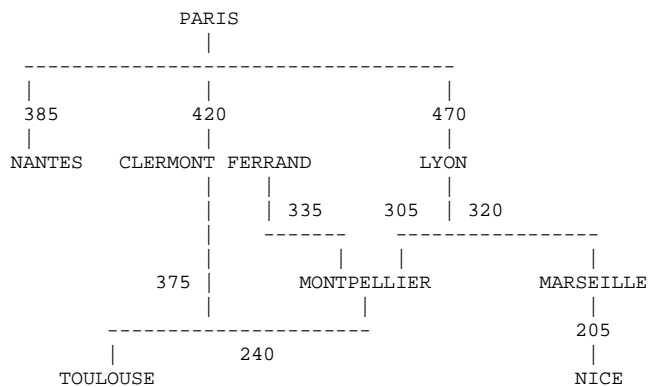
- Calendar

```
WITH years (ANNEE)
 as (VALUES (Year(Current_Date) -15)
 UNION ALL
 Select ANNEE+1 from years
 where annee < Year(Current_Date)),
 num as (select count(*) as cnt,
 year(arardt) as aryear from
 AOFSQL_8.aofarcf group by year(arardt))

SELECT ANNEE, COALESCE(cnt,0) cnt FROM years
LEFT OUTER JOIN num ON aryear = annee
```

## Navigation

- Réseau autoroutier.



## PARIS-TOULOUSE

```

WITH journey (TO_TOWN, STEPS, DISTANCE, WAY)
AS
 (SELECT DISTINCT JNY_FROM_TOWN, 0, 0, CAST('PARIS' AS VARCHAR(200))
 FROM T_JOURNEY
 WHERE JNY_FROM_TOWN = 'PARIS'
 UNION ALL
 SELECT JNY_TO_TOWN, departure.STEPS + 1,
 departure.DISTANCE + arrival.JNY_KM,
 concat(concat(departure.WAY, ','), arrival.JNY_TO_TOWN)
 FROM T_JOURNEY AS arrival
 INNER JOIN journey AS departure
 ON departure.TO_TOWN = arrival.JNY_FROM_TOWN
 WHERE departure.WAY NOT LIKE concat(concat('%',
arrival.JNY_TO_TOWN), '%'))
 SELECT *
 FROM journey
 WHERE TO_TOWN = 'TOULOUSE'

```