Advanced SQL in Oracle and SQL Server

The WITH Clause – PART I (Non-Recursive)

Scott L. Hecht
http://www.sheepsqueezers.com
@sheepsqueezers

Module Contents

Introduction

- **Why Learn the WITH Clause?**
    - Simple (Non-Recursive)
        - Neatens up large SQL query
        - Prevents repeated subqueries from being processed repeatedly
        - Can refer to previously defined WITH Clauses in newly defined WITH Clauses
    - Availability:
        - Oracle: 9i/R1 (Subquery Factoring Clause)
        - SQL Server: 2005 (Common Table Expressions)

Data Used in Module

- **Table**
  - CHILDSTAT
- **Columns**
  - FIRSTNAME – child's first name
  - GENDER – child's gender (M=Male, F=Female)
  - BIRTHDATE – child's date of birth
  - HEIGHT – child's height (inches)
  - WEIGHT – child's weight (pounds)
- **Data**

FIRSTNAME	GENDER	BIRTHDATE	HEIGHT	WEIGHT
LAUREN	F	10-JUN-00	54	876
ROSEMARY	F	08-MAY-00	35	123
ALBERT	M	02-AUG-00	45	150
BUDDY	M	02-OCT-98	45	189
FARQUAR	M	05-NOV-98	76	198
SIMON	M	03-JAN-99	87	256
TOMMY	M	11-DEC-98	78	167

Non-Recursive WITH Clause

- **SELECT and the WITH Clause**
  - Face it, we've all produced our share of large, complicated SQL queries!
  - Difficult to read and modify!

```
SELECT A.FIRSTNAME AS FIRSTNAME_MALE,
       A.WEIGHT AS WEIGHT_MALE,
       B.FIRSTNAME AS FIRSTNAME_FEMALE,
       B.WEIGHT AS WEIGHT_FEMALE
 FROM (SELECT FIRSTNAME,WEIGHT,
              ROW_NUMBER() OVER (ORDER BY WEIGHT DESC) AS WT_RANK_MALE
        FROM CHILDSTAT
        WHERE GENDER='M') A
      INNER JOIN
      (SELECT FIRSTNAME,WEIGHT,
              ROW_NUMBER() OVER (ORDER BY WEIGHT DESC) AS WT_RANK_FEMALE
        FROM CHILDSTAT
        WHERE GENDER='F') B
 ON A.WT_RANK_MALE=B.WT_RANK_FEMALE
```

Non-Recursive WITH Clause

- **SELECT and the WITH Clause**
  - Use WITH Clause to neaten up code!

```
WITH WT_M AS (SELECT FIRSTNAME,WEIGHT,
                     ROW_NUMBER() OVER (ORDER BY WEIGHT DESC)
                                                  AS WT_RANK_MALE
               FROM CHILDSTAT
               WHERE GENDER='M'),
     WT_F AS (SELECT FIRSTNAME,WEIGHT,
                     ROW_NUMBER() OVER (ORDER BY WEIGHT DESC)
                                                  AS WT_RANK_FEMALE
               FROM CHILDSTAT
               WHERE GENDER='F')
SELECT A.FIRSTNAME AS FIRSTNAME_MALE,
       A.WEIGHT AS WEIGHT_MALE,
       B.FIRSTNAME AS FIRSTNAME_FEMALE,
       B.WEIGHT AS WEIGHT_FEMALE
 FROM WT_M A INNER JOIN WT_F B
 ON A.WT_RANK_MALE=B.WT_RANK_FEMALE
```

Non-Recursive WITH Clause

- **SELECT and the WITH Clause**
  - Allows database to treat query as inline view/temporary table
  - Database doesn't have to run same query multiple times

```
WITH WT_AVG AS (SELECT AVG(WEIGHT) AS AVG_WT
                 FROM CHILDSTAT)
SELECT A.FIRSTNAME,A.GENDER,A.HEIGHT,A.WEIGHT
 FROM CHILDSTAT A
 WHERE A.GENDER='M'
       AND A.HEIGHT>50
       AND A.WEIGHT<=(SELECT AVG_WT FROM WT_AVG)
UNION ALL
SELECT B.FIRSTNAME,B.GENDER,B.HEIGHT,B.WEIGHT
 FROM CHILDSTAT B
 WHERE B.GENDER='F'
       AND B.HEIGHT>=40
       AND B.WEIGHT>=(SELECT AVG_WT FROM WT_AVG)
```

Non-Recursive WITH Clause

- **SELECT and the WITH Clause**
  - Can use WITH Clause within subqueries/inline views
  - Works on **Oracle** only!
  - Recommendation: **DON'T DO THIS!**

```
SELECT A.FIRSTNAME,A.GENDER,A.HEIGHT,A.WEIGHT
 FROM (
       WITH HT_AVG AS (SELECT AVG(HEIGHT) AS AVG_HT
                        FROM CHILDSTAT)
       SELECT *
        FROM CHILDSTAT
        WHERE HEIGHT>=(SELECT AVG_HT FROM HT_AVG)
      ) A
 WHERE A.WEIGHT>=50
```

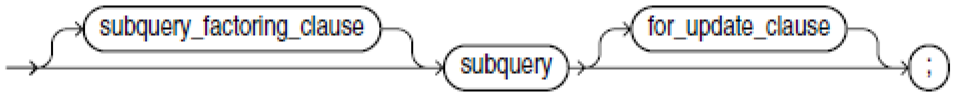
```
SELECT A.FIRSTNAME,A.GENDER,A.HEIGHT,A.WEIGHT
 FROM CHILDSTAT A
 WHERE A.WEIGHT>=(WITH WT_AVG AS (SELECT AVG(WEIGHT) AS AVG_WT
                                  FROM CHILDSTAT)
                  SELECT AVG_WT
                   FROM WT_AVG)
```

Non-Recursive WITH Clause Syntax

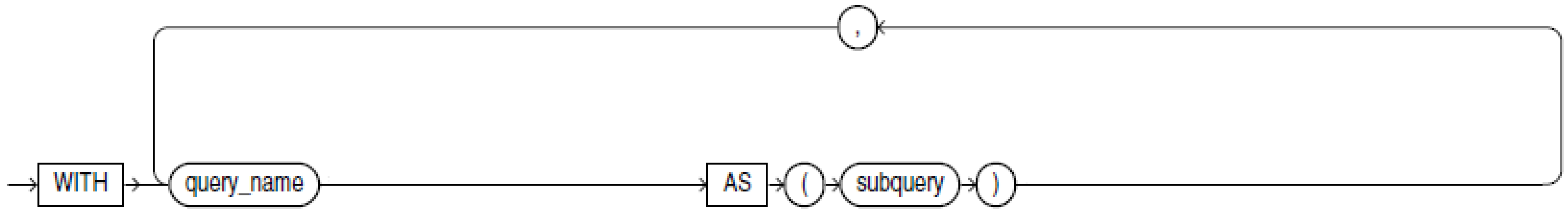
- **Non-Recursive WITH Clause Syntax**
  - Below are the Oracle syntax charts for the WITH Clause
  - Shows only non-recursive syntax

Syntax

***select*::=**



***subquery_factoring_clause*::=**



  - Some items removed from syntax above

Non-Recursive WITH Clause Syntax

- **CREATE TABLE and the WITH Clause Syntax**
  - The placement of CREATE TABLE is **above** the WITH Clause
  - SQL Server uses INTO in its normal place.

```
CREATE TABLE CHILDSTAT_REDUCED AS
 WITH WT_AVG AS (SELECT AVG(WEIGHT) AS AVG_WT
                  FROM CHILDSTAT)
 SELECT A.FIRSTNAME,A.GENDER,A.HEIGHT,A.WEIGHT
  FROM CHILDSTAT A
  WHERE A.GENDER='M'
        AND A.HEIGHT>50
        AND A.WEIGHT<=(SELECT AVG_WT FROM WT_AVG)
 UNION ALL
 SELECT A.FIRSTNAME,A.GENDER,A.HEIGHT,A.WEIGHT
  FROM CHILDSTAT A
  WHERE A.GENDER='F'
        AND A.HEIGHT>=40
        AND A.WEIGHT>=(SELECT AVG_WT FROM WT_AVG)
```

Non-Recursive WITH Clause Syntax

- **Using Previously Defined WITH Clauses**
  - Can refer to WITH Clauses previously defined
  - Cannot refer to those defined afterwards

```
WITH AVG_WT_GENDER AS (SELECT GENDER,AVG(WEIGHT) AS AVG_WT_SEX
                        FROM CHILDSTAT
                        GROUP BY GENDER),
     AVG_WT_OVERALL AS (SELECT AVG(AVG_WT_SEX) AS AVG_WT_ALL
                         FROM AVG_WT_GENDER)
 SELECT A.*
  FROM CHILDSTAT A
  WHERE A.WEIGHT>=(SELECT AVG_WT_ALL
                   FROM AVG_WT_OVERALL)
```

- **Finally…**
  - Once the SQL query completes, the WITH Clause's queries are destroyed
  - Cannot access them unless added to next SQL query

Summary

- Simplify hard-to-read SQL code
- Duplicated subqueries not processed multiple times
- Can access previously defined WITH Clauses