Advanced SQL in Oracle and SQL Server

The MERGE Statement

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

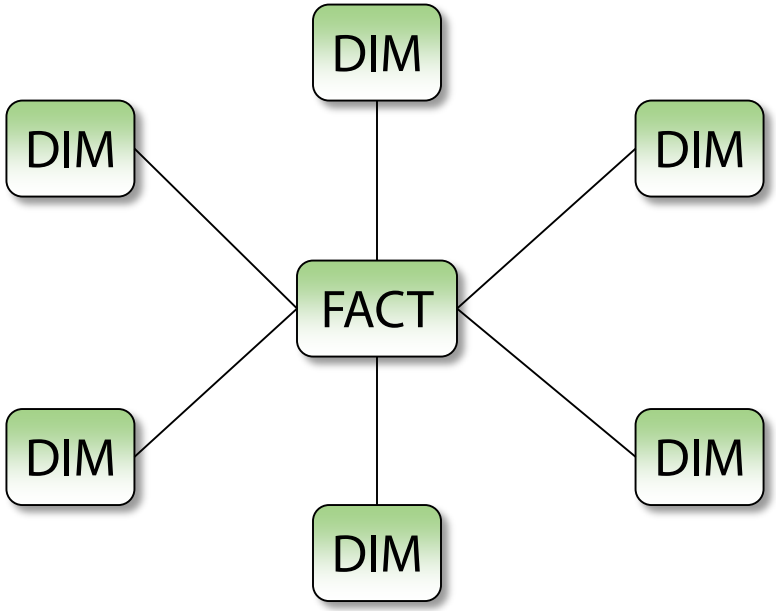
pluralsight
hardcore developer training

Module Contents

- **The MERGE Statement**
  - Introduction
  - Data used in Module
  - Reminder of Old Friends (INSERT, UPDATE and DELETE)
  - Without the MERGE Statement (Using INSERT, UPDATE and DELETE)
  - MERGE Syntax
    - WHEN MATCHED THEN
    - WHEN NOT MATCHED THEN
    - WHEN NOT MATCHED BY SOURCE THEN (SQL Server-Specific Syntax)
  - Using Additional Conditions with AND/WHERE
  - Deleting Rows with DELETE
  - Summary

Introduction

- **Why learn the MERGE Statement?**
  - Allows you to perform INSERTs, UPDATEs and DELETEs in one SQL statement
  - Can add additional criteria using AND or WHERE to ensure appropriate data is being affected
  - Useful for DBAs
  - Useful for Master/Reference Data Management



Introduction

- **Why learn the MERGE Statement? (*continued*)**
  - Requires two tables:
    - TARGET table – contains master/reference data requiring changes
    - SOURCE table – contains changes to be applied to TARGET
- **SOURCE table can contain rows:**
  - to be inserted into TARGET
  - to be updated in TARGET
- **Rows in TARGET not appearing in SOURCE can be deleted, if desired.**
- **Must have appropriate permissions on the underlying table(s)**
  - If necessary, contact your Database Administrator (DBA)
- **May need to COMMIT to commit changes!**
- **Availability:**
  - Oracle: 9i/R1
  - SQL Server: 2008

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

Data Used in Module

- **Table**
  - CHANGES
- **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
BOB	M	12-JUN-10	55	125
LAUREN				85

Reminder of Old Friends

- **The INSERT Statement**
  - Allows you to insert one or more rows of data into a table
  - *Syntax – Hard-coded values*

```
INSERT INTO target(Tcol1,Tcol2,...)
 VALUES(val1,val2,…)
```

  - *Syntax – Values from Existing Source Table*

```
INSERT INTO target(Tcol1,Tcol2,...)
 SELECT Scol1,Scol2,…
  FROM source
```

Reminder of Old Friends

- **The UPDATE Statement**
  - Allows you to update one or more rows of data in table
  - *Syntax – Hard-coded values*

```
UPDATE target
  SET Tcol1=val1,…
  WHERE …
```

  - *Syntax – Values from Existing Source Table*

```
UPDATE target
  SET Tcol1=Scol1,…
  FROM source…
  WHERE …
```

Reminder of Old Friends

- **The DELETE Statement**
  - Allows you to delete one or more rows of data in table
  - *Syntax – Hard-coded values*

```
DELETE FROM target
 WHERE …
```

  - If removing all rows, use TRUNCATE instead

Without the MERGE Statement

- **Update Target Table without the MERGE Statement**
  - Insert rows into *target*
  - Update *target*
  - Delete rows from *target*
- **Manual SQL:**

```
/* INSERT new row into CHILDSTAT */
INSERT INTO CHILDSTAT(FIRSTNAME,GENDER,BIRTHDATE,HEIGHT,WEIGHT)
 VALUES('BOB','M',DATE '2010-06-12',55,125);

/* UPDATE Lauren's weight */
UPDATE CHILDSTAT
 SET WEIGHT=85
 WHERE FIRSTNAME='LAUREN';

/* DELETE Simon because he's dead */
DELETE
 FROM CHILDSTAT
 WHERE FIRSTNAME='SIMON';
```

MERGE Syntax

- **MERGE Requirements**
  - A target table – table to be modified (e.g., CHILDSTAT)
  - A source table – table containing modifications (e.g., CHANGES)
  - How many different ways can these two tables be brought together?

WHEN MATCHED

WHEN NOT MATCHED

WHEN NOT MATCHED BY SOURCE

MERGE Syntax

- Syntax

```
MERGE
  INTO target T
  USING source S
  ON (join-criteria)
  WHEN MATCHED THEN
   update-clause
  WHEN NOT MATCHED THEN
   insert-clause
  WHEN NOT MATCHED BY SOURCE THEN
   update-clause | delete-clause
```

Example #1

- **Task: Update CHILDSTAT using CHANGES.**

```
MERGE
 INTO CHILDSTAT A
 USING CHANGES B
 ON (A.FIRSTNAME=B.FIRSTNAME)
 WHEN MATCHED THEN
  UPDATE SET A.WEIGHT=B.WEIGHT
 WHEN NOT MATCHED THEN
  INSERT(FIRSTNAME,GENDER,BIRTHDATE,HEIGHT,WEIGHT)
   VALUES(B.FIRSTNAME,B.GENDER,B.BIRTHDATE,B.HEIGHT,B.WEIGHT)
```

Example #1

- **Task: Update CHILDSTAT using CHANGES.**

FIRSTNAME	GENDER	BIRTHDATE	HEIGHT	WEIGHT
ROSEMARY	F	08-MAY-00	35	123
LAUREN	F	10-JUN-00	54	**85**
ALBERT	M	02-AUG-00	45	150
BUDDY	M	02-OCT-98	45	189
FARQUAR	M	05-NOV-98	76	198
TOMMY	M	11-DEC-98	78	167
SIMON	M	03-JAN-99	87	256
BOB	**M**	**12-JUN-10**	**55**	**125**

Additional Conditions

- **Additional Conditions and the MERGE Statement**
  - The MERGE statement allows for additional merge conditions
  - SQL Server uses the AND keyword
  - Oracle uses the WHERE keyword

Example #2

- **Task: Update CHILDSTAT using CHANGES.**
- **Note: Ensure updated row is female!**

```
/* ORACLE SYNTAX */
MERGE
 INTO CHILDSTAT A
 USING CHANGES B
 ON (A.FIRSTNAME=B.FIRSTNAME)
 WHEN MATCHED THEN
  UPDATE SET A.WEIGHT=B.WEIGHT
   WHERE A.GENDER='F'
 WHEN NOT MATCHED THEN
  INSERT(FIRSTNAME,GENDER,BIRTHDATE,HEIGHT,WEIGHT)
   VALUES(B.FIRSTNAME,B.GENDER,B.BIRTHDATE,B.HEIGHT,B.WEIGHT)
```

Example #2

- **Task: Update CHILDSTAT using CHANGES.**
- **Note: Ensure updated row is female!**

```
/* SQL SERVER SYNTAX */
MERGE
 INTO CHILDSTAT A
 USING CHANGES B
 ON (A.FIRSTNAME=B.FIRSTNAME)
 WHEN MATCHED AND A.GENDER='F' THEN
  UPDATE SET A.WEIGHT=B.WEIGHT
 WHEN NOT MATCHED THEN
  INSERT(FIRSTNAME,GENDER,BIRTHDATE,HEIGHT,WEIGHT)
   VALUES(B.FIRSTNAME,B.GENDER,B.BIRTHDATE,B.HEIGHT,B.WEIGHT)
```

Deleting Rows with DELETE

- **The MERGE Statement and the DELETE**
  - SQL Server allows you to delete rows (WHEN NOT MATCHED BY SOURCE)
  - Oracle's delete feature is limited to WHEN MATCHED

```
/* SQL SERVER SYNTAX */
MERGE
 INTO CHILDSTAT A
 USING CHANGES B
 ON (A.FIRSTNAME=B.FIRSTNAME)
 WHEN MATCHED THEN
  UPDATE SET A.WEIGHT=B.WEIGHT
 WHEN NOT MATCHED BY TARGET THEN
  INSERT(FIRSTNAME,GENDER,BIRTHDATE,HEIGHT,WEIGHT)
   VALUES(B.FIRSTNAME,B.GENDER,B.BIRTHDATE,B.HEIGHT,B.WEIGHT)
 WHEN NOT MATCHED BY SOURCE AND A.FIRSTNAME='SIMON' THEN
  DELETE
```

Summary

- MERGE allows for inserting, updating and deleting
- Single statement instead of multiple statements
- Great for Master/Reference Data Management
- But…only use when inserting/updating/deleting!