

SQL Server中的Merge关键字 - 宁静致远 - 博客频道 - CSDN

Merge简介

Merge关键字是一个神奇的DML关键字。它在SQL Server 2008被引入，它可将Insert,Update,Delete简单的并为一句。MSDN对于Merge的解释非常的短小精悍：“根据与源表联接的结果，对目标表执行插入、更新或删除操作。例如，根据在另一个表中找到的差异在一个表中插入、更新或删除行，可以对两个表进行同步。”通过这个描述，我们可以看出Merge是关于对于两个表之间的数据进行操作的。

Merge的使用场景:

- 数据同步
- 数据转换
- 基于源表对目标表做Insert,Update,Delete操作

Merge关键字的一些限制

- 使用Merge关键字只能更新一个表
- 源表中不能有重复的记录

Merge使用示例

1.Merge示例

首先建立源表和目标表，并插入相关的数据,如图1所示。

```
--创建源表
CREATE TABLE T_A(id INT,Name VARCHAR(50),book VARCHAR(50),Remark VARCHAR(200));
--创建目标表
CREATE TABLE T_B(id INT,Name VARCHAR(50),book VARCHAR(50),Remark VARCHAR(200));

--源表数据
INSERT INTO T_A(id,name,book)
SELECT 1,'吴承恩','西游记' UNION ALL
SELECT 2,'罗贯中','三国演义' UNION ALL
SELECT 3,'曹雪芹','红楼梦' UNION ALL
SELECT 4,'施耐庵','水浒传';
--目标表数据
INSERT INTO T_B(id,name,book)
SELECT 1,'吴老爷','西游记' UNION ALL
SELECT 2,'罗先生','三国演义' UNION ALL
SELECT 5,'孙大圣','三打白骨精' UNION ALL
SELECT 6,'二月河','李自成' UNION ALL
SELECT 7,'源表不存在, 会被删除','' UNION ALL
SELECT 8,'源表不存在, 会被删除','';
```

下面写一个简单的Merge示例

[sql] view plain copy

```
9. MERGE INTO T_B AS B
10. USING T_A AS A ON B.id=A.ID
11. WHEN MATCHED --当Bid=AID时，更新目标表
12. THENUPDATESET B.name=A.name,B.Remark='更新记录'
13. WHENNOT MATCHED --若目标表中不存在，则新源表记录插入目标表
14. THENINSERT(id,name,book,remark) VALUES(A.id,A.name,A.book,'从源表添加')
15. WHENNOT MATCHED BY SOURCE --源表中不存在，目标表存在
16. THENDELETE;
```

执行前记录:

结果		消息	
id	Name	book	Remark
1	1	吴承恩	西游记
2	2	罗贯中	三国演义
3	3	曹雪芹	红楼梦
4	4	施耐庵	水浒传

id	Name	book	Remark
1	1	吴老爷	西游记
2	2	罗先生	三国演义
3	5	孙大圣	三打白骨精
4	6	二月河	李自成
5	7	源表不存在，会被删除	
6	8	源表不存在，会被删除	

执行后B表记录:

id	Name	book	Remark
1	1	吴承恩	西游记
2	2	罗贯中	三国演义
3	3	曹雪芹	红楼梦
4	4	施耐庵	水浒传

2.Output子句

Merge语句还有一个强大的功能是通过OUTPUT子句，可以将刚刚做过变动的数据进行输出。我们在上面的Merge语句后加入OUTPUT子句,如图所示。

[sql] view plain copy

12. --Merge Output
13. MERGE INTO T_B AS B
14. USING T_A AS A ON B.id=A.ID
15. WHEN MATCHED --当B.id=A.ID时，更新目标表
16. THEN UPDATE SET B.name=A.name,B.Remark='更新记录'
17. WHEN NOT MATCHED --若目标表中不存在，则新源表记录插入目标表
18. THEN INSERT(id,name,book,remark) VALUES(A.id,A.name,A.book,'从源表添加')
19. WHEN NOT MATCHED BY SOURCE --源表中不存在，目标表存在
20. THEN DELETE
21. OUTPUT \$ACTION AS [ACTION],Inserted.id AS 插入的ID,Inserted.name AS 插入的Name,
22. Deleted.id AS 删除的ID,Deleted.name AS 删除的Name;

输出结果:

	ACTION	插入的ID	插入的Name	删除的ID	删除的Name
1	INSERT	3	曹雪芹	NULL	NULL
2	INSERT	4	施耐庵	NULL	NULL
3	UPDATE	1	吴承恩	1	吴老爷
4	UPDATE	2	罗贯中	2	罗先生
5	DELETE	NULL	NULL	5	孙大圣
6	DELETE	NULL	NULL	6	二月河
7	DELETE	NULL	NULL	7	源表不存在，会被删除
8	DELETE	NULL	NULL	8	源表不存在，会被删除

3.其它条件组合使用

3.1 TOP约束

我们还可以使用TOP关键字限制目标表被操作的行。在上例的语句基础上加上了TOP关键字，我们看到只有指定行被执行。

```

42 --Merge Output
43 MERGE TOP (3) T_B AS B
44 USING T_A AS A ON B.id=A.ID
45 WHEN MATCHED --当B.id=A.ID时, 更新目标表
46 THEN UPDATE SET B.name=A.name,B.Remark='更新记录'
47 WHEN NOT MATCHED --若目标表中不存在, 则新源表记录插入目标表
48 THEN INSERT(id,name,book,remark) VALUES(A.id,A.name,A.book,'从源表添加')
49 WHEN NOT MATCHED BY SOURCE --源表中不存在, 目标表存在
50 THEN UPDATE SET b.remark='源表无此记录'
51 OUTPUT $ACTION AS [ACTION],Inserted.id AS 插入的ID,Inserted.name 插入的Name,
52 Deleted.id AS 删除的ID,Deleted.name AS 删除的Name;
53
54

```

	ACTION	插入的ID	插入的Name	删除的ID	删除的Name
1	INSERT	3	曹雪芹	NULL	NULL
2	UPDATE	1	吴承恩	1	吴老爷
3	UPDATE	2	罗贯中	2	罗先生

3.2AND附加限制条件

注意:

- 1.不能有多个 MATCHED 和 Not MATCHED, 即使是加and 条件的也不行, 只给有一次。
- 2.NOT MATCHED BY SOURCE 可以加条件出现多次, 但不带and 条件限制的NOT MATCHED BY SOURCE 必须出现在带and 条件的后面。

SQL语句

```

29 --Merge
30 MERGE INTO T_B AS B
31 USING T_A AS A ON B.id=A.ID
32 WHEN MATCHED AND A.id<4
33 --条件满足时, 更新目标表
34 THEN UPDATE SET B.name=A.name,B.Remark='更新记录'
35 WHEN NOT MATCHED
36 --若目标表中不存在, 则新源表记录插入目标表
37 THEN INSERT(id,name,book,remark) VALUES(A.id,A.name,A.book,'从源表添加')
38 --源表中不存在, 且目标表id=7,8的记录删除
39 WHEN NOT MATCHED BY SOURCE AND B.id IN(7,8)
40 THEN DELETE
41 --源表中不存在, 目标表存在的标识记录
42 WHEN NOT MATCHED BY SOURCE
43 THEN UPDATE SET b.remark='源表无此记录'
44 --输出执行结果
45 OUTPUT $ACTION AS [ACTION],Inserted.id AS 插入的ID,Inserted.name 插入的Name,
46 Deleted.id AS 删除的ID,Deleted.name AS 删除的Name;
47

```

输出结果如下图:

	ACTION	插入的ID	插入的Name	删除的ID	删除的Name
1	INSERT	3	曹雪芹	NULL	NULL
2	INSERT	4	施耐庵	NULL	NULL
3	UPDATE	1	吴承恩	1	吴老爷
4	UPDATE	2	罗贯中	2	罗先生
5	UPDATE	5	孙大圣	5	孙大圣
6	UPDATE	6	二月河	6	二月河
7	DELETE	NULL	NULL	7	源表不存在, 会被删除
8	DELETE	NULL	NULL	8	源表不存在, 会被删除

执行前源表与目标表记录:

	id	Name	book	Remark
1	1	吴承恩	西游记	NULL
2	2	罗贯中	三国演义	NULL
3	3	曹雪芹	红楼梦	NULL
4	4	施耐庵	水浒传	NULL

	id	Name	book	Remark
1	1	吴老爷	西游记	NULL
2	2	罗先生	三国演义	NULL
3	5	孙大圣	三打白骨精	NULL
4	6	二月河	李自成	NULL
5	7	源表不存在, 会被删除		NULL
6	8	源表不存在, 会被删除		NULL

执行后目标表结果:

	id	Name	book	Remark
1	1	吴承恩	西游记	更新记录
2	2	罗贯中	三国演义	更新记录
3	5	孙大圣	三打白骨精	源表无此记录
4	6	二月河	李自成	源表无此记录
5	3	曹雪芹	红楼梦	从源表添加
6	4	施耐庵	水浒传	从源表添加