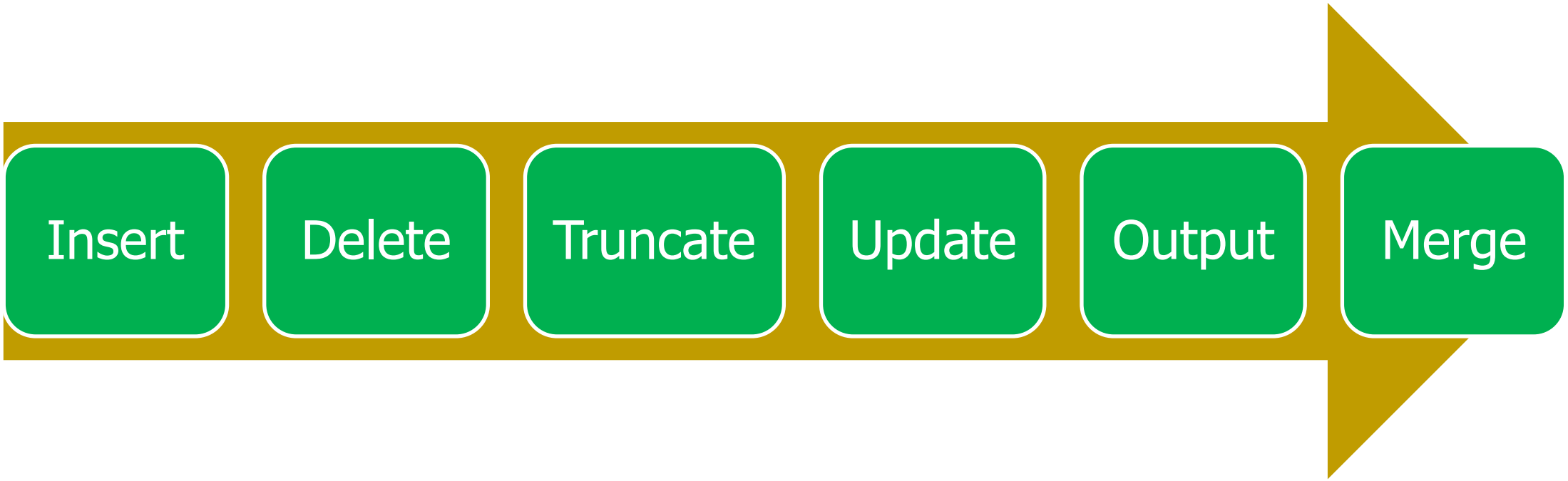
SQL数据更新



如何得到一条语句所影响的行数？

用found_rows()函数

判断Select得到的行数

用row_count()函数判断

Update或Delete影响的行数

```
select * from SC
where grade >= 60
select found_rows()
```

```
delete from SC
where grade > 100
select row_count()
```

插入操作的命令格式

insert into 表名 [（列名[，列名]...）]

values （值[，值]...）

插入一条指定好值的行

insert into 表名 [（列名[，列名]...）]

（子查询）

插入子查询结果中的若干行

插入显式行

```
insert into teacher
    values      ( t123, “王明”, 35, D08, 498 ),
                ( t124, “李明”, 38, D01, 698 );
```

```
insert into teacher (tno, tname, dno)
    values      row( t123, “王明”, D08 ),
                row( t125, “李明”, D08 );
```

思考：salary取何值？如何防止插入带有空值的行？

插入子查询

将平均成绩大于90的学生加入到EXCELLENT中

```
insert into EXCELLENT ( sno, grade)
    select  sno , avg(grade)
    from    SC
    group by (sno)
    having  avg(grade) > 90
```

```
select  sno , avg(grade)
into  EXCELLENT ( sno, grade)
from    SC
group by (sno)
having  avg(grade) > 90
```

replace into：替代主码相同的现有行

```
create table  test_replace(
        id   int auto_increment primary key,
        name   char(10),
        cur_time   datetime)
insert into  test_replace(name, cur_time)
                values ( 'A', now() ), ( 'B', now() )
```

replace into：替代主码相同的现有行

```
replace into  test_replace
values( 1,' AA', now() )
```

id	name	cur_time
1	A	2022-03-28 20:01:59
2	B	2022-03-28 20:01:59

```
replace into  test_replace
values( 3, 'C', now() )
```

id	name	cur_time
1	AA	2022-03-28 20:02:35
2	B	2022-03-28 20:01:59

id	name	cur_time
1	AA	2022-03-28 20:02:35
2	B	2022-03-28 20:01:59
3	C	2022-03-28 20:04:34

删除操作的命令格式

delete from 表名

[where 条件表达式]

- 从表中删除符合条件的元组
- 如果没有where语句，则删除所有元组

清除所有选课记录

delete from SC

小贴士：MySQL运行在safe-updates模式时，where条件只允许使用主码

set SQL_SAFE_UPDATES = 0

删除操作示例

删除王明老师所有的任课记录

```
delete  from    TC
where   tno  in
        (select   tno
         from     teacher
         where  tname = “王明”)
```

思考：在这里用in和＝有区别吗？

演示in和=的不同的实例

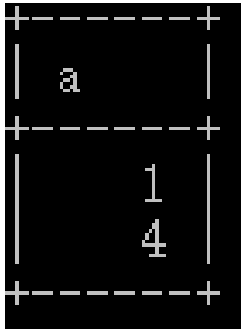
```
create table  delA (a int);

create table  delB (b int);

insert into  delA  values (1),(3),(4);

insert into  delB  values (3),(4),(5);
```

```
delete from  delA  where  a =
   (select  b  from  delB  where b<4);
```



```
delete from  delA  where  a =
   (select  b  from  delB  where b>3);
```

Error Code: 1242.

Subquery returns more than 1 row

如何删除重复行中的1个？

```
create table dupRows ( id int auto_increment primary key, val int);
insert into dupRows(val) values (1), (2), (2), (3), (3), (3);
```

id	val
1	1
2	2
3	2
4	3
5	3
6	3

```
delete from dupRows
where val= 3
order by id desc
limit 1
```

id	val
1	1
2	2
3	2
4	3
5	3

删除操作示例

删除低于平均工资的老师记录

```
delete  from    teacher
where     salary  <
                (select  avg(salary)
                 from    teacher)
```

思考：是先找到所有符合条件的行一并删除，还是找到一个删除一个？

MySQL不允许从子查询中出现的表中删除数据

错误的删除操作写法

```
create table  testDel (val int);

insert into  testDel  values (2), (4), (5), (7), (8)
```

```
delete   from    testDel
where     val  <
      ( select   avg(val)
        from     testDel )
```

Error Code: 1093.

You can't specify target table 'testDel'

for update in FROM clause

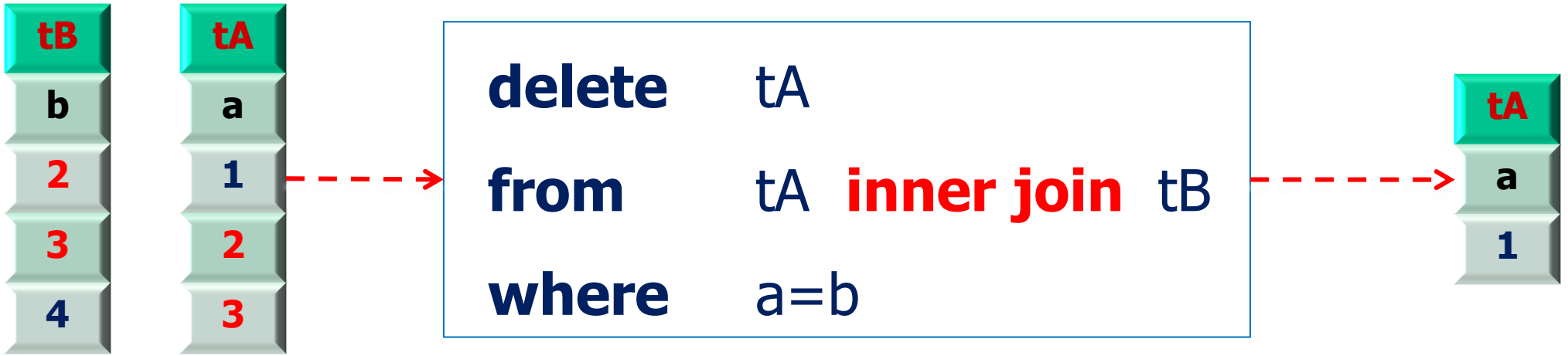
正确的删除操作写法

val
2
4
5
7
8

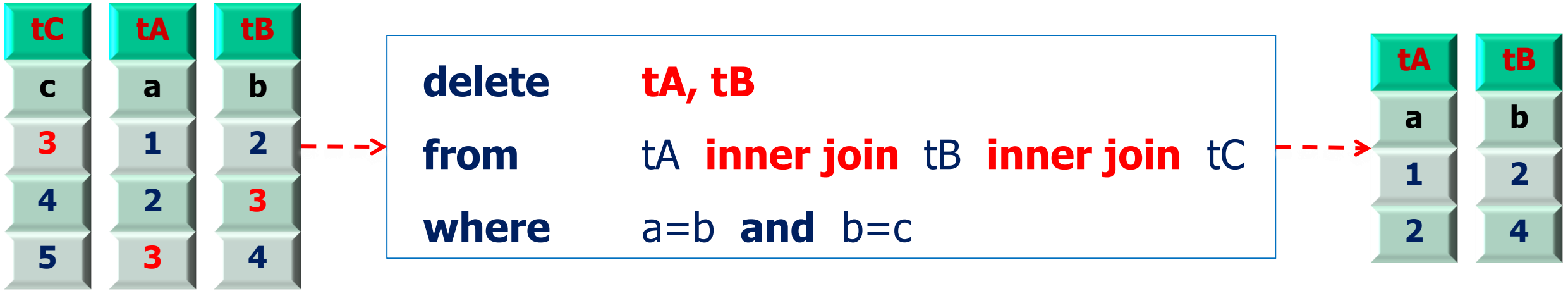
```
with  tmp as
( select   avg(val) as A
  from     testDel )
delete     testDel
from       testDel  inner join  tmp
where      val > A
```

val
2
4
5

多表删除操作



多表删除操作


tC
c
3
4
5
tA
a
1
2
3
tB
b
2
3
4
delete tA, tB
from tA inner join tB inner join tC
where a=b and b=c
tA
a
1
2
tB
b
2
4

清空表

truncate table

删除表中的所有行，而不记录单个行删除操作

- **truncate table**在功能上与不带where子句的delete语句相同
- **truncate table**比delete速度快，使用的系统和事务日志资源少

auto_increment计数器重置为种子值

更新操作的命令格式

```
update  表名
set     列名 = 表达式 | 子查询
        列名 = [, 表达式 | 子查询]...
[where  条件表达式]
```

指定对哪些列进行更新，以及更新后的值是什么

老师工资上调5%

```
update      teacher
set         salary = salary * 1.05
```

将D01系系主任的工资改为该系的平均工资

想当然的

错误写法

```
update  teacher
set     salary =
        (select  avg(salary)
         from    teacher
         where   dno = D01)
where   tno =
        (select  dean
         from    department
        where    dno = D01)
```

将D01系系主任的工资改为该系的平均工资

正确的写法

```
with  tmp  as
        (select  avg(salary)  avgsal  from  teacher)
update  teacher
set     salary = tmp.avgsal
where  tno =
        (select  dean
         from     department
         where  dno = D01)
```

当C1课程成绩小于该课程平均成绩时，将其提高5%

```
with  tmp as  ( select  avg(grade)  ag
                from   SC
                where cno = C1 )
update     SC
set        grade = grade * 1.05
where      cno = C1
       and  grade < tmp.ag
```

对于任意1门课程，当某同学该课程的成绩小于该课程的平均成绩时，将其提高5%

```
with  tmp as  ( select  cno, avg(grade)  ag
                from   SC
                group by cno )
update     SC
set        grade = grade * 1.05
where      cno = tmp.cno
       and  grade < tmp.ag
```

工资超过2000的缴纳10%所得税，其余的缴纳5%所得税

①

```
update  teacher
set     salary = salary * 0.9
where salary> 2000
```

②

```
update  teacher
set     salary = salary * 0.95
where  salary <= 2000
```

执行顺序是①, ②, 还是②, ①？

墨菲定律

$$1-(1-p)^n$$

使用case when表达不同条件更新分支

```
update  teacher

set     salary =

        case salary

                when salary > 2000 then salary * 0.9

                when salary <= 2000 then salary * 0.95
```

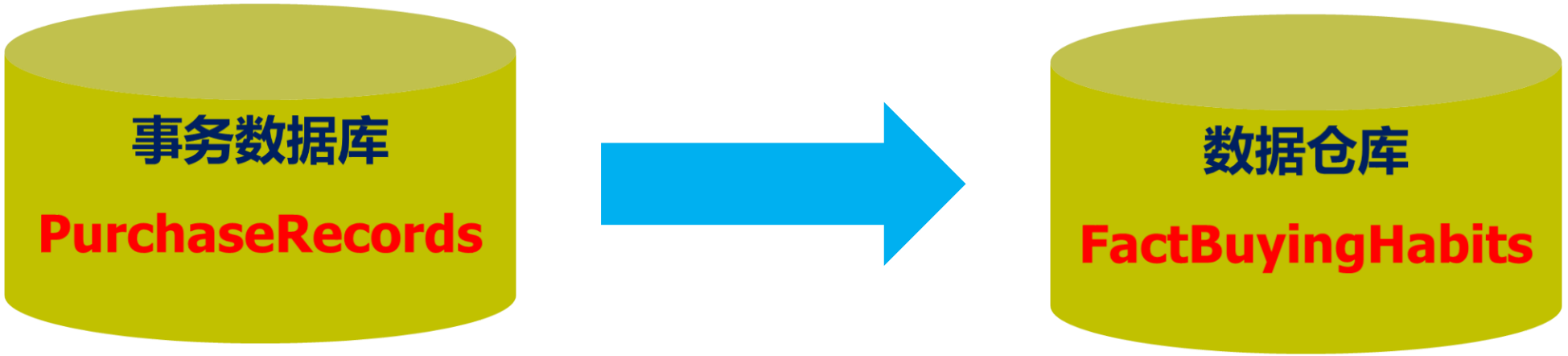
UPSERT：表同步

工作表→仓库表

- 在把一组记录加载到表中时，一个经典的挑战是如何识别和处理目标表中已有的记录
- 常用的方法是如果某记录不存在，就将它插入；如果存在，就用源表中的数据 更新该记录

需要定义复杂的存储过程来完成一系列INSERT或UPDATE命令，这个技术通常被称为UPSERT

表同步：跟踪购买习惯



- 定期将PurchaseRecords表中的信息合并到FactBuyingHabits表中
- 对于不存在的产品-客户对，插入新行
- 对于已存在的产品-客户对，更新最近的购买日期

merge

- **using**子句
- 指定数据源

- **merge**子句
- 指定目标表

- **on**子句指定目标与源的匹配连接条件

- **when**子句基于**on**子句的结果指定所要采取的操作
  - **when matched**
  - **when not matched by target**
  - **when not matched by source**

merge实现业务表到仓库表同步的示例


```
merge  dbo.FactBuyingHabits  as target
using  (select      CustomerID, ProductID, PurchaseDate
        from        dbo.Purchases)  as source
    on (Target.ProductID = Source.ProductID
        and Target.CustomerID = Source.CustomerID)
when matched then
    update set Target.LastPurchaseDate = Source.PurchaseDate
when not matched by target then
    insert (CustomerID, ProductID, LastPurchaseDate)
    values (Source.CustomerID, Source.ProductID, Source.PurchaseDate)
```

通过Output记录更新历史：SQL Server

执行修改操作只返回影响了多少行的信息，无从获知到底影响到了哪些行。如果在修改操作语句中带上**output**，就可以输出具体的影响信息

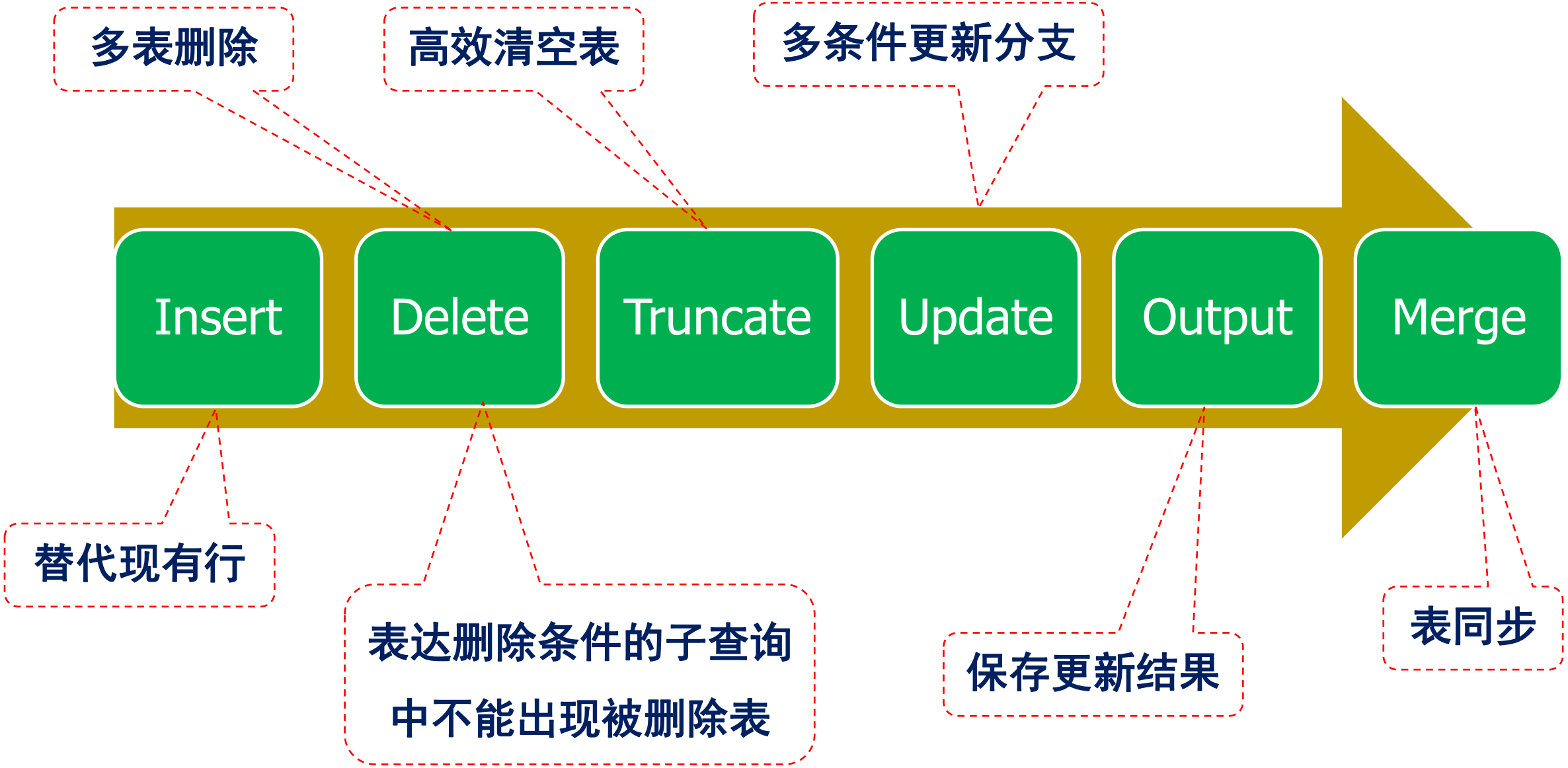
```
output { deleted | inserted }.{ * | column_name }
[into table_name]
```

deleted 和**inserted** 是两个虚表，**deleted** 里面存放修改之前的值，**inserted** 里面存放的是修改之后的值

Output：自定义审计小助手

```
declare  @recordChange  table
    ( beforeGrade   int,
      afterGrade    int  )

update  sc
set     grade = grade * 1.05
where   grade < 60
output  deleted.grade,  inserted.grade  into  @recordChange
```


多表删除
高效清空表
多条件更新分支
Insert
Delete
Truncate
Update
Output
Merge
替代现有行
表达删除条件的子查询
中不能出现被删除表
保存更新结果
表同步