Atitit 数据同步merge 最佳实践 流程 如果存在就更新,否则新建.docx

[1.1. 我们经常会遇到这样的情况：1、首先判断数据是否存在；2、如果不存在，则插入；3、如果存在，则更新。 1](#_Toc25160)

[1.2. INSERT INTO … ON DUPLICATE KEY UPDATE 推荐 1](#_Toc7126)

[1.3. INSTEAD OF INSERT触发器（不推荐） 2](#_Toc19119)

[1.4. merge sql语句 2](#_Toc14598)

[1.5. mysql有一个类似merge的sql语句，那就是replace into（不推荐）  2](#_Toc5387)

[1.6. 执行完毕会提示影响了俩行，这个正常的。实际相当于修改了。 3](#_Toc1232)

[1.7. GPL语言代码实现（不推荐） 3](#_Toc32567)

[1.8. 3 总结从上面的测试结果看出，相同之处： 两者的区别只有一个 3](#_Toc29522)

[1.9. 提升性能 一次插入多条记录： 4](#_Toc27584)

## 我们经常会遇到这样的情况：1、首先判断数据是否存在；2、如果不存在，则插入；3、如果存在，则更新。

有一下几种实现模式

## INSERT INTO … ON DUPLICATE KEY UPDATE 推荐，保留原值

MySQL 自4.1版以后开始支持INSERT … ON DUPLICATE KEY UPDATE语法，使得原本需要执行3条SQL语句（SELECT,INSERT,UPDATE），缩减为1条语句即可完成

因此，如果replace填充的字段不全，则会导致未被更新的字段都会修改为默认值，并且如果有自增id的话，自增id会变化为最新的 值（这样如果是以自增id为标志的话可能导致记录丢失）；而insert\*update只是更新部分字段，对于未被更新的字段不会变化（不会强制修改为默 认值）。

如果行作为新记录被插入，则受影响行的值为1；如果原有的记录被更新，则受影响行的值为2。

MySQL 在数据冲突时实际上是删掉了旧记录，再写入新记录，这是使用 REPLACE INTO 时最大的一个误区，拿之前的例子来说，执行完 REPLACE INTO auto (k, v) VALUES (1, ‘1-1’) 之后，由于新写入记录时并未给 extra 字段指定值，原记录 extra 字段的值就「丢失」了，而通常这并非是业务上所预期的，更常见的需求实际上是，当存在 k=1 的记录时，就把 v 字段的值更新为 ‘1-1’，其他未指定的字段则保持原状，而满足这一需求的 MySQL 方言是 INSERT INTO auto (k, v) VALUES (1, ‘1-1’) ON DUPLICATE KEY UPDATE v=VALUES(v); 鉴于此，很多使用 REPLACE INTO 的场景，实际上需要的是 INSERT INTO … ON DUPLICATE KEY UPDATE，在正确理解 REPLACE INTO 行为和副作用的前提下，谨慎使用 REPLACE INTO。

、

INSERT INTO l\_department( title,dptno) VALUES ('服务管理 科','00021747')

ON DUPLICATE KEY UPDATE title='服务管理 科'

## INSTEAD OF INSERT触发器（不推荐）

Mysql不支持

## merge sql语句

mysql不支持（推荐）

## mysql有一个类似merge的sql语句，那就是replace into（不推荐） 不保留原值

replace into的用法，真的很好用，是insert into的增强版。在向表中插入数据时，我们经常会遇到这样的情况：1、首先判断数据是否存在；2、如果不存在，则插入；3、如果存在，则更新。

replace into 跟 insert 功能类似，不同点在于：replace into 首先尝试插入数据到表中， 1. 如果发现表中已经有此行数据（根据主键或者唯一索引判断）则先删除此行数据，然后插入新的数据。 2. 否则，直接插入新数据。

要注意的是：插入数据的表必须有主键或者是唯一索引！否则的话，replace into 会直接插入数据，这将导致表中出现重复的数据。

MySQL replace into 有三种形式：

1. replace into tbl\_name(col\_name, ...) values(...)

2. replace into tbl\_name(col\_name, ...) select ...

3. replace into tbl\_name set col\_name=value, ...

第一种形式类似于insert into的用法，

第二种replace select的用法也类似于insert select，这种用法并不一定要求列名匹配，事实上，MYSQL甚至不关心select返回的列名，它需要的是列的位置。例如，replace into tb1( name, title, mood) select rname, rtitle, rmood from tb2;?这个例子使用replace into从?tb2中将所有数据导入tb1中。

第三种replace set用法类似于update set用法，使用一个例如“SET col\_name = col\_name + 1”的赋值，则对位于右侧的列名称的引用会被作为DEFAULT(col\_name)处理。因此，该赋值相当于SET col\_name = DEFAULT(col\_name) + 1。

前两种形式用的多些。其中 “into” 关键字可以省略，不过最好加上 “into”，这样意思更加直观。另外，对于那些没有给予值的列，MySQL 将自动为这些列赋上默认值。

## 执行完毕会提示影响了俩行，这个正常的。实际相当于修改了。

[SQL]replace into l\_department(dptpid,title,dptno)values('00021744','服务管理 科','00021747')

受影响的行: 2

时间: 0.023s

## GPL语言代码实现（不推荐）

从开发效率考虑，sql实现更快更好。 可读性 高。。

编程语言实现此功能，较为繁琐

## ****3 总结从上面的测试结果看出，相同之处：**** 两者的区别只有一个

（1），没有key的时候，replace与insert .. on deplicate udpate相同。  
（2），有key的时候，都保留主键值，并且auto\_increment自动+1  
不同之处：有key的时候，replace是delete老记录，而录入新的记录，所以原有的所有记录会被清除，这个时候，如果replace语句的字段不全的话，有些原有的比如例子中c字段的值会被自动填充为默认值。  
      而insert .. deplicate update则只执行update标记之后的sql，从表象上来看相当于一个简单的update语句。  
      **但是实际上，根据我推测，如果是简单的update语句，auto\_increment不会+1，应该也是先delete，再insert的操作，只是在insert的过程中保留除update后面字段以外的所有字段的值。**  
  
 所以两者的区别只有一个，**insert .. on deplicate udpate保留了所有字段的旧值，再覆盖然后一起insert进去，而replace没有保留旧值，直接删除再insert新值。**  
 从底层执行效率上来讲，replace要比insert .. on deplicate update效率要高，但是在写replace的时候，字段要写全，防止老的字段数据被删除

## 提升性能 一次插入多条记录：

**[sql]** [view plain](http://blog.csdn.net/lalaguozhe/article/details/9150049" \o "view plain) [copy](http://blog.csdn.net/lalaguozhe/article/details/9150049" \o "copy) [print](http://blog.csdn.net/lalaguozhe/article/details/9150049" \o "print)[?](http://blog.csdn.net/lalaguozhe/article/details/9150049" \o "?)

1. **insert** **into**
2. ​   Shops(shopid, viewtotal)
3. **values**(1,123456),
4. (2, 234567),
5. (3, 345678)
6. **on** duplicate **key** **update**
7. shopid = **values**(shopid),
8. viewtotal = **values**(viewtotal)

[IMG_256](http://blog.csdn.net/lalaguozhe/article/details/javascript:;)

这样我在wormhole中增加"INSERT ON DUPLICATE KEY UPDATE"操作，通过sql模板拼接出update多条记录的语句（比如一次更新一个batch size 1000条），提交语句，批量更新。

## 参考资料

“INSERT ON DUPLICATE KEY UPDATE”更新部分字段 - yukangkk的技术博客 - 博客频道 - CSDN.NET.html

INSERT ... ON DUPLICATE KEY UPDATE - 高爽\_Coder - 博客频道 - CSDN.NET.html