[MySQL](http://apps.hi.baidu.com/share/detail/:;) 当记录不存在时插入，当记录存在时更新  
网上基本有三种解决方法。

第一种：

示例一：插入多条记录

假设有一个主键为 client\_id 的 clients 表，可以使用下面的语句：

INSERTINTOclients(client\_id,client\_name,client\_type)SELECTsupplier\_id,supplier\_name,'advertising'FROMsuppliersWHEREnotexists(select\*fromclients  
whereclients.client\_id=suppliers.supplier\_id);

示例一：插入单条记录

INSERTINTOclients(client\_id,client\_name,client\_type)SELECT10345,'IBM','advertising'FROMdualWHEREnotexists(select\*fromclientswhereclients.client\_id=10345);

使用 dual 做表名可以让你在 select 语句后面直接跟上要插入字段的值，即使这些值还不存在当前表中。

第二种：

INSERT 中**ON DUPLICATE KEY UPDAT**E的使用

如果您指定了ON DUPLICATE KEY UPDATE，并且插入行后会导致在一个UNIQUE[索引](http://apps.hi.baidu.com/share/detail/:;)或PRIMARY KEY中出现重复值，则执行旧行UPDATE。

例如，如果列a被定义为UNIQUE，并且包含值1，则以下两个语句具有相同的效果：

mysql>INSERT INTO table (a,b,c) VALUES (1,2,3) ON DUPLICATE KEY UPDATE c=c+1;

mysql>UPDATE table SET c=c+1 WHERE a=1;

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。

注释：如果列b也是唯一列，则INSERT与此UPDATE语句相当：

mysql>UPDATE table SET c=c+1 WHERE a=1 OR b=2 LIMIT 1;

如果a=1 OR b=2与多个行向匹配，则只有一个行被更新。通常，您应该尽量避免对带有多个唯一关键字的表使用ON DUPLICATE KEY子句。

您可以在UPDATE子句中使用VALUES(col\_name)[函数](http://apps.hi.baidu.com/share/detail/:;)从INSERT...UPDATE语句的INSERT部分引用列值。换句话说，如果没有发生重复关键字冲突，则UPDATE子句中的VALUES(col\_name)可以引用被插入的col\_name的值。本函数特别适用于多行插入。VALUES()函数只在INSERT...UPDATE语句中有意义，其它时候会返回NULL。

示例：

mysql>INSERT INTO table (a,b,c) VALUES (1,2,3),(4,5,6) ON DUPLICATE KEY UPDATE c=VALUES(a)+VALUES(b);

本语句与以下两个语句作用相同：

mysql>INSERT INTO table (a,b,c) VALUES (1,2,3) ON DUPLICATE KEY UPDATE c=3;//上c=a+b=1+2

mysql>INSERT INTO table (a,b,c) VALUES (4,5,6) ON DUPLICATE KEY UPDATE c=9;//上c=a+b=4+5

当您使用ON DUPLICATE KEY UPDATE时，DELAYED选项被忽略。

举个例子，字段a被定义为UNIQUE，并且原数据库表table中已存在记录(2,2,9)和(3,2,1)，如果插入记录的a值与原有记录重复，则更新原有记录，否则插入新行：

INSERT INTO TABLE (a,b,c) VALUES (1,2,3),(2,5,7),(3,3,6),(4,8,2)

ON DUPLICATE KEY UPDATE b=VALUES(b);

以上SQL语句的执行，发现(2,5,7)中的a与原有记录(2,2,9)发生唯一值冲突，则执行ON DUPLICATE KEY UPDATE，将原有记录(2,2,9)更新成(2,5,9)，将(3,2,1)更新成(3,3,1)，插入新记录(1,2,3)和(4,8,2).

第三种：

**REPLACE**语句

　　我们在使用[数据库](http://hi.baidu.com/magine2003/blog/item/:;)时可能会经常遇到这种情况。如果一个表在一个字段上建立了唯一索引，当我们再向这个表中使用已经存在的键值插入一条记录，那将会抛出一个主键冲突的错误。当然，我们可能想用新记录的值来覆盖原来的记录值。如果使用传统的做法，必须先使用DELETE语句删除原先的记录，然后再使用INSERT插入新的记录。而在MySQL中为我们提供了一种新的解决方案，这就是REPLACE语句。使用REPLACE插入一条记录时，如果不重复，REPLACE就和INSERT的功能一样，如果有重复记录，REPLACE就使用新记录的值来替换原来的记录值。  
　　使用REPLACE的最大好处就是可以将DELETE和INSERT合二为一，形成一个原子操作。这样就可以不必考虑在同时使用DELETE和INSERT时添加事务等复杂操作了。  
　　在使用REPLACE时，表中必须有唯一索引，而且这个索引所在的字段不能允许空值，否则REPLACE就和INSERT完全一样的。  
　　在执行REPLACE后，系统返回了所影响的行数，如果返回1，说明在表中并没有重复的记录，如果返回2，说明有一条重复记录，系统自动先调用了DELETE删除这条记录，然后再记录用INSERT来插入这条记录。如果返回的值大于2，那说明有多个唯一索引，有多条记录被删除和插入。  
　　REPLACE的语法和INSERT非常的相似，如下面的REPLACE语句是插入或更新一条记录。  
　　REPLACE INTO users (id,name,age) VALUES(123, '赵本山', 50);  
　  
　　插入多条记录：  
　　REPLACE INTO users(id, name, age)VALUES(123, '赵本山', 50), (134,'Mary',15);  
  
　　REPLACE也可以使用SET语句  
　　REPLACE INTO users SET id = 123, name = '赵本山', age = 50;  
  
　　上面曾提到REPLACE可能影响3条以上的记录，这是因为在表中有超过一个的唯一索引。在这种情况下，REPLACE将考虑每一个唯一索引，并对每一个索引对应的重复记录都删除，然后插入这条新记录。假设有一个table1表，有3个字段a, b, c。它们都有一个唯一索引。  
　　CREATE TABLE table1(a INT NOT NULL UNIQUE,b INT NOT NULL UNIQUE,c INT NOT NULL UNIQUE);  
　　假设table1中已经有了3条记录  
　　a b c  
　　1 1 1  
　　2 2 2  
　　3 3 3  
  
　　下面我们使用REPLACE语句向table1中插入一条记录。  
　　REPLACE INTO table1(a, b, c) VALUES(1,2,3);  
　　返回的结果如下  
　　Query OK, 4 rows affected (0.00 sec)  
　　在table1中的记录如下  
　　a b c  
　　1 2 3  
  
　　我们可以看到，REPLACE将原先的3条记录都删除了，然后将（1, 2, 3）插入。

总结：虽然没有具体测试，感觉第一种最费资源了（只是感觉），不过你要是没有主键的话也只能用他了。

第二种和第三种的区别是：

1)insert是先尝试插入，若主键存在则更新。REPLACE是先尝试插入，若主键存在则删除原纪录再插入。

2)如果有多个唯一关键字发生冲突(不同关键字的冲突发生在不同记录),比如现在有2个字段2条记录冲突了(没条记录冲突一个字段)，则insert是选择排序后在前面的一条进行更新，REPLACE是删除那两条记录，然后插入新记录。