

# mysql数据库Insert语句后面加ON DUPLICATE KEY UPDATE，保证唯一性

如果在INSERT语句末尾指定了ON DUPLICATE KEY UPDATE，并且插入行后会导致在一个UNIQUE索引或PRIMARY KEY中出现重复值，则在出现重复值的行执行UPDATE；如果不会导致唯一值列重复的问题，则插入新行。

例如：

建表语句：

```
CREATE TABLE `consume_locus` (  
    `rds` varchar(255) NOT NULL,  
    `topic` varchar(255) default NULL,  
    `createTime` datetime default NULL,  
    `dtsTime` datetime default NULL,  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

sql语句：

```
insert into t_tableName(topic,createTime,dtsTime,rds ) values(  
null,'2017-02-08 14:02:50','','') on duplicate key update topic =  
values(topic),createTime = values(createTime),dtsTime =  
values(dtsTime),rds = values(rds) ;
```

这可以保证主键只有一条记录。

附上一段java代码，算是工具类。根据传入BEAN生成SQL语句。

```
private Map<String, String> fieldMap;
```

```

private String tablename;

public Map<String, String> getFieldMap() {

public void setFieldMap(Map<String, String> fieldMap) {

    this.fieldMap = fieldMap;

public String getTablename() {

public void setTablename(String tablename) {

    this.tablename = tablename;


    public static void main(String args[]) throws
InterruptedException{

        String createTimeStr=DateUtil.dateToStr(new Date(),
DateUtil.YYYYMMDDHHMMSS2);

        Map<String, String> fieldMap=new HashMap<String, String>
();

        fieldMap.put("rds", ClusterClientCore.rdsCode);

        fieldMap.put("topic", ClusterClientCore.topic);

        fieldMap.put("createTime", createTimeStr);

        fieldMap.put("dtsTime", ClusterClientCore.dtsTimestamp);

        DbBean bean =new DbBean();

        bean.setFieldMap(fieldMap);

        bean.setTablename("t_tableName");

        String sql=getInsertSqlStr(bean);

        System.out.println(sql);

public static String getInsertSqlStr(DbBean bean){

    StringBuffer columnName=new StringBuffer();

    StringBuffer columValue=new StringBuffer();

    StringBuffer duplicateSb=new StringBuffer();

    Map<String, String> fieldMap=bean.getFieldMap();

```

```

        Iterator it=fieldMap.entrySet().iterator();

        Map.Entry<String, Object> entry=(Map.Entry<String,
Object>)it.next();

        String key=entry.getKey();

        duplicateSb.append(key).append(" =
values(").append(key).append("),");

        columnName.append(key).append(",");

        if(entry.getValue()!=null){

            String value=(String)entry.getValue();

            columValue.append("'").append(value).append("'").append(",");

            columValue.append(entry.getValue()).append(",");

            String
columnNameStr=columnName.deleteCharAt(columnName.length()-1).toString();

            String
columValueStr=columValue.deleteCharAt(columValue.length()-1).toString();

            String sqlStr="insert into "+ bean.getTablename()+"
("+columnNameStr+" ) values( "+columValueStr+" ) ";

            String duplicateStr=" on duplicate key update
"+duplicateSb.deleteCharAt(duplicateSb.length()-1).toString()+" ";

            return sqlStr+duplicateStr;

```