## 369.android.database.sqlite.SQLiteDatabaseLockedException

【出错堆栈】

http://bugly.qq.com/share/DPv6Hf

【出错代码】

**new** Thread() {  
 @Override  
 **public void** run() {  
 SQLiteDatabase db = **dbHelper**.getWritableDatabase();  
 db.beginTransaction();  
 Cursor c = db.rawQuery(**"select \* from table1"**, **null**);  
 **if** (c != **null** && c.getCount() != 0)  
 **while** (c.moveToNext()) {  
 String name = c.getString(0);  
 String pwd = c.getString(1);  
 System.***out***.println(**"name:"** + name + **" pwd:"** + pwd);  
 }  
 db.close();  
 System.***out***.println(**"db close"**);  
 **dbHelper**.getWritableDatabase();  
  
 }  
}.start();

【出错原因】

通过错误分析，我们可以看到在db.getwriteabeDatabase中定位到SqliteDataBase.open()方法中

**private void** open() {  
 **try** {  
 **try** {  
 openInner();  
 } **catch** (SQLiteDatabaseCorruptException ex) {  
 onCorruption();  
 openInner();  
 }  
 } **catch** (SQLiteException ex) {  
 Log.*e*(***TAG***, **"Failed to open database '"** + getLabel() + **"'."**, ex);  
 close();  
 **throw** ex;  
 }  
}

最终可以看到是执行了SqliteConnection.open()方法来开启一个sqliteConnection。

当调用db.close()方法时，如果所有的db对象都被关闭了会调用

SqliteDatabase.onAllReferencesReleased（）方法，而这个方法最后是关闭了SqliteConnection。

在调用了db.beginTransaction()方法后，也获取了一个sqliteconnection,此时不调用db.endTransaction()方法直接执行db.close()方法，在下一次调用db.getWriteableDatabase的时候就会抛出这个异常，与崩溃堆栈相吻合。

【解决方案】

由出错原因可知在调用db.beginTransaction()方法后必须调用db.endTransaction。

所以我们的解决方案如下：

Try{

db.beginTransaction();

//执行db操作

}catch(Exception e){

}finally{

db.endTransaction();

}

确保endTransaction方法的调用