```
//每个表的建表脚本
    private static final String EVENT_SQL_CREATE= "create table " +
EVENT_TABLE_NAME +
    " ( eid integer primary key autoincrement,"
    + " title text not null, "
    + " place text, "
    + " content text, "
    + " creattime text, "
    + " remindtime text, "
    + " starttime text, "
    + " endtime text);";

    private static final String MODE_SQL_CREATE = "create table " +
MODE_TABLE_NAME +
    " ( mid integer primary key autoincrement,"
    + " name text, "
    + " volume integer, "
    + " vibrate integer); ";

    private static final String MODIFY_SQL_CREATE = "create table " +
MODIFY_TABLE_NAME +
    " ( eid integer , "
    + " mid integer , "
    + " primary key (eid), "
    + " foreign key (eid) references event , "
    + " foreign key (mid) references mode ); ";


    public iScheduleDB(Context context, String name, CursorFactory factory,
int version) {
        super(context, name, factory, version);
    }
    public iScheduleDB(Context context){
        super(context, DB_NAME, null, DB_VRESION);
    }

    public void onCreate(SQLiteDatabase db) {
        //允许建表脚本，创建3个表
        db.execSQL(EVENT_SQL_CREATE);
        db.execSQL(MODE_SQL_CREATE);
        db.execSQL(MODIFY_SQL_CREATE);
    }
```

```
// 插入事件的脚本
    public long insert(Event entity) {
        SQLiteDatabase db = getWritableDatabase();
        ContentValues values = new ContentValues();
        values.put("title", entity.getTitle());
        values.put("place", entity.getPlace());
        values.put("content", entity.getContent());
        values.put("creattime",
dateFormat.format(entity.getCreatTime()));
        values.put("remindtime",
dateFormat.format(entity.getRemindTime()));
        values.put("starttime",
dateFormat.format(entity.getStartTime()));
        values.put("endtime", dateFormat.format(entity.getEndTime()));
        // 必须保证 values 至少一个字段不为null ， 否则出错
        long rid = db.insert(EVENT_TABLE_NAME, null, values);
        entity.setEventId(rid);
        db.close();
        return rid;
    }

// 插入情景模式的脚本
    public long insert(Mode entity) {
        SQLiteDatabase db = getWritableDatabase();
        Cursor c = db.rawQuery("select * from " + MODE_TABLE_NAME + " ;",
null);
        ContentValues values = new ContentValues();
        values.put("name", entity.getName());
        values.put("volume", entity.getVolume());
        values.put("vibrate", entity.getVibrate());
        // 必须保证 values 至少一个字段不为null ， 否则出错
        long rid = db.insert(MODE_TABLE_NAME, null, values);
        entity.setModeId(rid);
        db.close();
        return rid;
    }

    // 插入事件和情景模式的脚本
    public long insert(Event event, Mode mode) {
        SQLiteDatabase db = getWritableDatabase();
        ContentValues values = new ContentValues();
        values.put("eid", event.getEventId());
        values.put("mid", mode.getModeId());
        // 必须保证 values 至少一个字段不为null ， 否则出错
```

```
        long rid = db.insert(MODIFY_TABLE_NAME, null, values);
        db.close();
        return rid;
    }

    //删除数据操作的脚本
    public int deleteEventById(Integer id){
        SQLiteDatabase db = getWritableDatabase();
        String whereClause = "eid = ?";
        String[] whereArgs = { id.toString() };
        int row = db.delete(EVENT_TABLE_NAME, whereClause, whereArgs);
        db.close();
        return row;
    }

    //根据事件名称删除事件操作的脚本
    public int deleteEventByTitle(String title){
        SQLiteDatabase db = getWritableDatabase();
        String whereClause = "title = ?";
        String[] whereArgs = { title };
        int row = db.delete(EVENT_TABLE_NAME, whereClause, whereArgs);
        db.close();
        return row;
    }

    //根据情景模式的代号来删除情景模式操作的脚本
    public int deleteModeById(Integer id){
        SQLiteDatabase db = getWritableDatabase();
        String whereClause = "mid = ?";
        String[] whereArgs = { id.toString() };
        int row = db.delete(MODE_TABLE_NAME, whereClause, whereArgs);
        db.close();
        return row;
    }

    //根据事件名称删除情景模式操作的脚本
    public int deleteModify(Event event){
        SQLiteDatabase db = getWritableDatabase();
        String whereClause = "eid = ?";
        String[]      whereArgs      =      {      Integer.toString((int)
event.getEventId()) };
        int row = db.delete(MODIFY_TABLE_NAME, whereClause, whereArgs);
        db.close();
        return row;
```

```
    }

    //根据事件id来更新数据操作的脚本
    public int updateEventById(Event entity) {
        SQLiteDatabase db = getWritableDatabase();
        String whereClause = "eid = ?";
        String[]      whereArgs      =      {      Integer.toString((int)
entity.getEventId()) };
        Log.d("test", Integer.toString((int) entity.getEventId()));
        ContentValues values = new ContentValues();
        values.put("title", entity.getTitle());
        values.put("place", entity.getPlace());
        values.put("content", entity.getContent());
        values.put("creattime",
dateFormat.format(entity.getCreatTime()));
        values.put("remindtime",
dateFormat.format(entity.getRemindTime()));
        values.put("starttime",
dateFormat.format(entity.getStartTime()));
        values.put("endtime", dateFormat.format(entity.getEndTime()));
        int  rows  =  db.update(EVENT_TABLE_NAME,  values,  whereClause,
whereArgs);
        db.close();
        return rows;
    }

    //根据事件名称来更新数据操作的脚本
    public int updateEventByTitle(Event entity) {
        SQLiteDatabase db = getWritableDatabase();
        String whereClause = "title = ?";
        String[] whereArgs = { entity.getTitle() };
        ContentValues values = new ContentValues();
        values.put("title", entity.getTitle());
        values.put("place", entity.getPlace());
        values.put("content", entity.getContent());
        values.put("creattime",
dateFormat.format(entity.getCreatTime()));
        values.put("remindtime",
dateFormat.format(entity.getRemindTime()));
        values.put("starttime",
dateFormat.format(entity.getStartTime()));
        values.put("endtime", dateFormat.format(entity.getEndTime()));
        int  rows  =  db.update(EVENT_TABLE_NAME,  values,  whereClause,
whereArgs);
```

```
        db.close();
        return rows;
    }

    //根据情景模式的id来更新数据操作的脚本
    public int updateModeById(Mode entity){
        SQLiteDatabase db = getWritableDatabase();
        String whereClause = "mid = ?";
        String[]      whereArgs      =      {      Integer.toString((int)
entity.getModeId()) };
        ContentValues values = new ContentValues();
        values.put("name", entity.getName());
        values.put("volume", entity.getVolume());
        values.put("vibrate", entity.getVibrate());
        int   row   =   db.update(MODE_TABLE_NAME,   values,   whereClause,
whereArgs);
        db.close();
        return row;
    }

    //更新情景设置操作的脚本
    public int updateModify(Event event, Mode mode){
        SQLiteDatabase db = getWritableDatabase();
        String whereClause = "eid = ?";
        String[]      whereArgs      =      {      Integer.toString((int)
event.getEventId()) };
        ContentValues values = new ContentValues();
        values.put("eid", event.getEventId());
        values.put("mid", mode.getModeId());
        int   row   =   db.update(MODIFY_TABLE_NAME,   values,   whereClause,
whereArgs);
        db.close();
        return row;
    }

    //根据事件id来查询事件操作的脚本
    public Event getEventById(Integer id) throws ParseException{
        Event event = null;
        SQLiteDatabase db = getReadableDatabase();
        String selection = "eid = ?";
        String[] selectionArgs = { id.toString() };
        Cursor    c    =    db.query(EVENT_TABLE_NAME,    null,    selection,
selectionArgs, null, null, null);
        if (c.moveToNext()){
```

```
            event                                =                                new
Event(c.getString(1),c.getString(2),c.getString(3),
                    new  Date(dateFormat.parse(c.getString(4)).getTime()),
new Date(dateFormat.parse(c.getString(5)).getTime()),
                    new  Date(dateFormat.parse(c.getString(6)).getTime()),
new Date(dateFormat.parse(c.getString(7)).getTime()));
            event.setEventId(c.getLong(0));
        }
        c.close();
        db.close();
        return event;
    }

    //根据情景模式id来获取情景操作的脚本
    public Mode getModeById(Integer id){
        Mode mode = null;
        SQLiteDatabase db = getReadableDatabase();
        String selection = "mid = ?";
        String[] selectionArgs = {id.toString()};
       Cursor c = db.query(MODE_TABLE_NAME, null, selection, selectionArgs,
null, null, null);
        if (c.moveToNext()){
            mode = new Mode(c.getString(1), c.getInt(2), c.getInt(3));
            mode.setModeId(c.getLong(0));
        }
        c.close();
        db.close();
        return mode;
    }

    //根据事件来查看日程操作的脚本
    public List<Event> getEventByDate(Date date) throws ParseException {
        List<Event> list = new ArrayList<Event>();
        SQLiteDatabase db = getReadableDatabase();
        SimpleDateFormat format_begin = new SimpleDateFormat("yyyy-MM-dd
00:00:00");
        SimpleDateFormat  format_end  =  new  SimpleDateFormat("yyyy-MM-dd
23:59:59");
        String dateBegin = format_begin.format(date);
        String dateEnd = format_end.format(date);
        String SEARCH_EVENT = "select * from " + EVENT_TABLE_NAME
                + " where not ( datetime(starttime) > \"" + dateEnd + "\" or
"
                + "datetime(endtime) < \"" + dateBegin + "\" );";
```

```
        // Log.d("test", SEARCH_EVENT);

        Cursor c = db.rawQuery(SEARCH_EVENT, null);
        while (c.moveToNext()){
            Event event = new
Event(c.getString(1),c.getString(2),c.getString(3),
                    new Date(dateFormat.parse(c.getString(4)).getTime()),
new Date(dateFormat.parse(c.getString(5)).getTime()),
                    new Date(dateFormat.parse(c.getString(6)).getTime()),
new Date(dateFormat.parse(c.getString(7)).getTime()));
            event.setEventId(c.getLong(0));
            list.add(event);
        }
        c.close();
        db.close();

        return list;
    }

    //根据事件名称来查询日程操作的脚本
    public List<Event> getEventByTitle(String title) throws ParseException
{
        List<Event> list = new ArrayList<Event>();
        SQLiteDatabase db = getReadableDatabase();
        String selection = "title = ?";
        String[] selectionArgs = { title };
        Cursor c = db.query(EVENT_TABLE_NAME, null, selection,
selectionArgs, null, null, null);

        while (c.moveToNext()){
            Event event = new
Event(c.getString(1),c.getString(2),c.getString(3),
                    new Date(dateFormat.parse(c.getString(4)).getTime()),
new Date(dateFormat.parse(c.getString(5)).getTime()),
                    new Date(dateFormat.parse(c.getString(6)).getTime()),
new Date(dateFormat.parse(c.getString(7)).getTime()));
            event.setEventId(c.getLong(0));
            list.add(event);
        }
        c.close();
        db.close();

        return list;
    }
```

```
    //查看所有日程操作的脚本
    public List<Event> getALLEvent() throws ParseException {
        List<Event> list = new ArrayList<Event>();
        SQLiteDatabase db = getReadableDatabase();
        Cursor c = db.query(EVENT_TABLE_NAME, null, null, null, null, null,
"starttime");

        while (c.moveToNext()){
            Event                event                =                new
Event(c.getString(1),c.getString(2),c.getString(3),
                    new  Date(dateFormat.parse(c.getString(4)).getTime()),
new Date(dateFormat.parse(c.getString(5)).getTime()),
                    new  Date(dateFormat.parse(c.getString(6)).getTime()),
new Date(dateFormat.parse(c.getString(7)).getTime()));
            event.setEventId(c.getLong(0));
            list.add(event);
        }
        c.close();
        db.close();

        return list;
    }

    //查看所有情景模式操作的脚本
    public List<Mode> getAllModes() {
        List<Mode> list = new ArrayList<Mode>();
        SQLiteDatabase db = getReadableDatabase();

        Cursor c = db.query(MODE_TABLE_NAME, null, null, null, null,null,
null);
        while (c.moveToNext()){
            Mode mode = new Mode(c.getString(1), c.getInt(2), c.getInt(3));
            mode.setModeId(c.getLong(0));
            list.add(mode);
        }
        c.close();
        db.close();

        return list;
    }

    //根据事件id来查看情景模式操作的脚本
    public Mode getModeByEventId(Integer id){
```

```
        Mode mode = null;
        SQLiteDatabase db = getReadableDatabase();
        String selection = "eid = ?";
        String[] selectionArgs = {id.toString()};
        Cursor c = db.query(MODIFY_TABLE_NAME, null, selection,
selectionArgs, null, null, null);
        if (c.moveToNext()){
            long mid = c.getLong(1);
            mode = getModeById((int) mid);
        }
        c.close();
        db.close();
        return mode;
    }
}
```