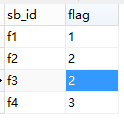
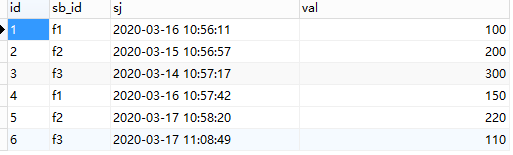
两张表：

t\_sb（设备表）



其中sb\_id是主键。

表2，t\_sbcj（设备采集信息记录表）



其中id为主键。这两张表通过sb\_id字段关联。

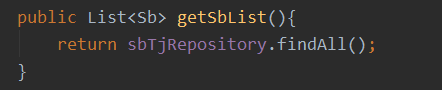
## 需求1：多对多，以t\_sb为主表建立实体关系获取t\_sb关联的t\_sbcj信息

建立实体关系：

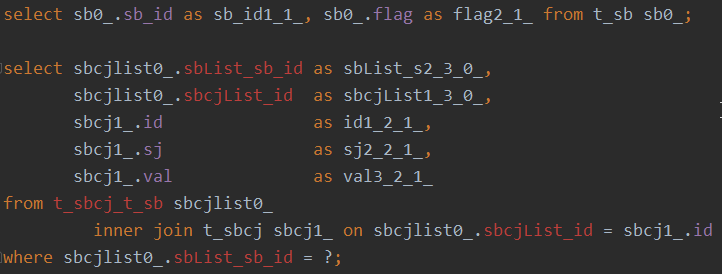


在SbService（相关的repository结构为SbTjRepository extends JpaRepository<Sb,String>

）中使用下面的方法进行查询：

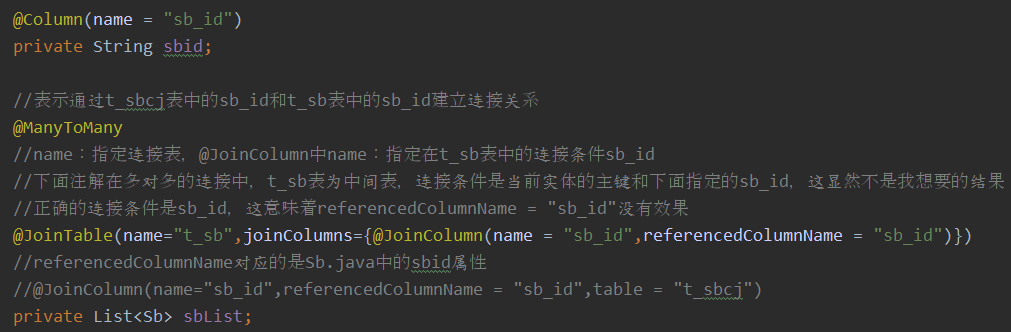


实际生产的sql如下所示：

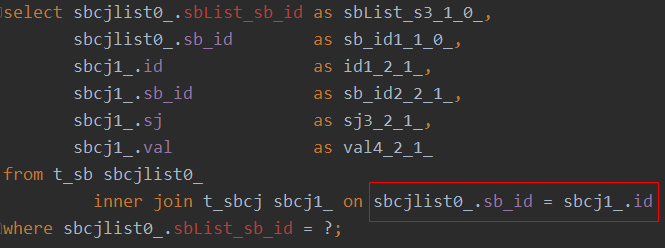


从图中明显可以得知是错误的sql。在多对多关联中，@JoinColumn注解定义的实体关系，在查询时最终生成的sql是通过中间关系表来进行连接查询的，上面图中的t\_sbcj\_t\_sb就是中间关系表，这个表根本不存在。

将上面的Sbcj类该一下如下：



还是那个查询，最终生成的sql如下：



可以看出生成的sql是通过sb\_id字段和主键id作连接条件的，很明显也是错的（连接条件是sb\_id）。如果在t\_sbcj表中sb\_id作为主键，那么可以按着此种方式建立实体关系后进行多对多关联查询。但是也有局限性，就算t\_sbcj表中sb\_id作为主键，查询改为了以Sbcj为主获取相关的Sb信息，那么上面@JoinTable中的name指定的关联表就不对了，也要改。所以，在多对多关联查询中，关联的表之间没有关联关系表时，最好不要通过建立实体关系的方式来进行查询。