**商品表（官方）**

**官方：公司发布商品的商品表**

**个人：会员售卖商品的商品表**

**满足条件：**

1.要支持公司自主发布商品、修改商品、删除商品

2.商品有多种品类，具体有模型类、陵园扩容类、现金充值类等等（后期可能会扩展品类）

**表存储结构及内容：**

序列编号（uuid） STRING

商品名称 STRING

商品展示图片 STRING

商品售价 DOUBLE

类别（大门、墓碑、人物...） INTEGER

简介 TEXT

扩展属性（用于支持商品品类数据特别属性）TEXT

**商品扩展属性例子：**

模型类：

模型名称、文件地址、应用类别

扩容类：

扩容单位、扩容量

充值类：

充值金额

考虑到商品品类的不同存储结构，决定采用JSON格式进行存储，存储例子如下：

模型类：

{“type\_name”:”model”,”model\_name”:”模型名称”,”file\_address”:”文件地址”,”type”:”应用类别”}

扩容类：

{“type\_name”:”capacity”,”unity”:”单位”,”value”:”扩容量”}

充值类：

{“type\_name”:”recharge”,”amount”:”金额”}

**动植物表**

序列号

名称

图片

价格

模型编号

类型（1、动物：2、植物）

分组（将一类动物规程一类）

阶段（1、2、3、4）

简介

**商品表（个人）**

**官方：公司发布商品的商品表**

**个人：会员售卖动植物的商品表**

**满足条件：**

1.要支持会员对个人背包可售商品（动物、植物）的售卖发布

**表存储结构及内容：**

序列编号（uuid） STRING

会员编号 INTEGER

动植物商品编号 STRING

商品售价 DOUBLE

库存 INTEGER

**购物车**

购物车功能全由unity3D维护管理，java服务仅提供下单功能接口（接收提交过来的表单数据（各种商品数据），进行金额计算后，调用支付，成功后进行数据写入数据库等操作，失败后返回错误）

**背包表**

**满足条件：**

1. 商城购物后，当背包中不存在会员购买商品时，可以增加购买成功的商品数据
2. 背包商品使用后，要可以对背包商品（消耗品）剩余数量进行扣减，当剩余数量为0时，彻底删除数据
3. 商城购物后，当背包中存在相同商品，则修改（对商品拥有数量进行叠加）

**表存储结构及内容：**

序列编号（uuid） STRING

会员编号 INTEGER

商品编号 STRING

剩余数量 INTEGER

商品来源（1：动植物表、2：官网商城、3：个人商城、4：公园（动植物园）） INTEGER

会员订单表

订单号

会员编号

下单时间

支付总金额

订单商品（JSON[{商品名称、商品购买数量、商品单价}]）

**养成系统**

**满足条件：**

1. 要可以记录动植物成长阶段（成长有效天数和当前生命值）
2. 要支持动植物死亡、收取（收回到背包）
3. 要支持收取后进行售卖

**表存储结构及内容：**

**动植物园**

序列编号（uuid） STRING

所属陵园编号 STRING

商品编号 STRING

总养成天数（有效天数） BIGINT

生命值 INTEGER

根据生命值来判断获取活跃状态（行走、趴伏..）

**使用存储过程每天检查成长系统、变更商品编号（小狗编号、大狗编号）**

**成长算法：**

**当动植物生命值大于0时，每天扣减1点，当扣减成功后，总养成天数增加1**

**{"integral\_deduction":"是否积分抵扣", "deduction":"积分抵扣量",**

**[{"goods\_id":"商品编号", "source":"商品来源", "count":"购买数量"},**

**{"goods\_id":"商品编号", "source":"商品来源", "count":"购买数量"},**

**{"goods\_id":"商品编号", "source":"商品来源", "count":"购买数量"}]}**

**商品来源：1：官网、2：会员商城、3：动植物园**