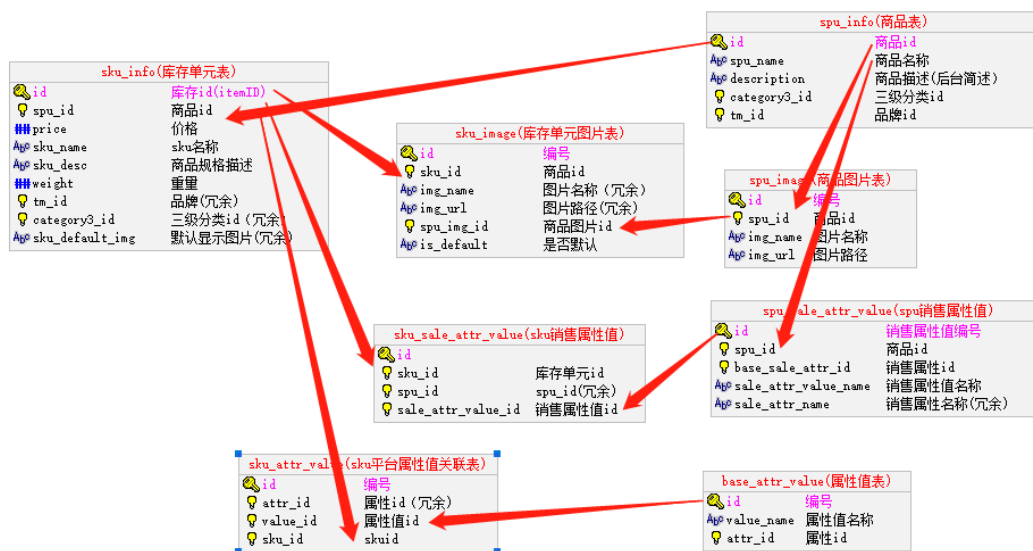


尚品汇商城

一、业务介绍

1.1 数据库表结构

根据以上的需求，以此将 SKU 关联的数据库表结构设计为如下：



1.2 数据准备

1.1 平台属性添加

三 首页 / 基本信息管理 / 平台属性列表
按线上传

一级分类 手机
 二级分类 手机通讯
 三级分类 手机

+ 添加平台属性

序号	属性id	属性名称	操作
1	1	价格	修改
2	2	屏幕尺寸	修改
3	3	运行内存	修改
4	4	机身存储	修改
5	5	CPU型号	修改

1.2 商品 spu 管理

三 首页 / 商品信息管理 / 商品属性SPU管理
头像

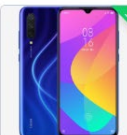
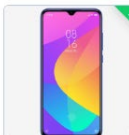



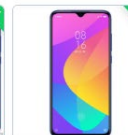
一级分类 手机
 二级分类 手机通讯
 三级分类 手机

spu名称 小米CC9手机

品牌 小米

spu描述 小米手机真是好，深得民心，不错，很好，good! nice, perfect!

图片上传

添加销售属性信息

+ 添加销售属性

序号	销售属性id	销售属性名称	销售属性值	操作
1	1	选择颜色	<div style="display: flex; align-items: center;"> <div style="border: 1px solid #007bff; padding: 2px 5px; margin-right: 5px;">仙女渐变</div> <div style="border: 1px solid #007bff; padding: 2px 5px; margin-right: 5px;">深蓝星球</div> <div style="border: 1px solid #007bff; padding: 2px 5px; margin-left: 5px;">+ 添加</div> </div>	🗑️ 删除
2	2	选择版本	<div style="display: flex; align-items: center;"> <div style="border: 1px solid #007bff; padding: 2px 5px; margin-right: 5px;">6G+128G</div> <div style="border: 1px solid #007bff; padding: 2px 5px; margin-right: 5px;">6G+64G</div> <div style="border: 1px solid #007bff; padding: 2px 5px; margin-right: 5px;">8G+256G</div> <div style="border: 1px solid #007bff; padding: 2px 5px; margin-left: 5px;">+ 添加</div> </div>	🗑️ 删除

保存
返回

二、保存 skuInfo 功能

2.1 图片加载功能

功能分析：图片列表是根据 spuId 得来，涉及到的数据库表 spu_image

2.1.1 添加接口实现类

接口

```
/**
 * 根据 spuId 查询 spuImageList
 * @param spuId
 * @return
 */
List<SpuImage> getSpuImageList(Long spuId);
```

实现类

```
@Override
public List<SpuImage> getSpuImageList(Long spuId) {
    QueryWrapper<SpuImage> queryWrapper = new
    QueryWrapper<>();
    queryWrapper.eq("spu_id", spuId);
    return spuImageMapper.selectList(queryWrapper);
}
```

2.1.2 添加控制器

```
@Api(tags = "商品 SKU 接口")
@RestController
@RequestMapping("admin/product")
public class SkuManageController {

    @Autowired
    private ManageService manageService;

    /**
     * 根据 spuId 查询 spuImageList
     * @param spuId
     * @return
     */
    @GetMapping("spuImageList/{spuId}")
    public Result<List<SpuImage>>
    getSpuImageList(@PathVariable("spuId") Long spuId) {
        List<SpuImage> spuImageList =
        manageService.getSpuImageList(spuId);
        return Result.ok(spuImageList);
    }
}
```

2.2 销售属性

2.2.1 编写接口以及实现类

接口:

```
/**
 * 根据 spuId 查询销售属性集合
 * @param spuId
 * @return
 */
List<SpuSaleAttr> getSpuSaleAttrList(Long spuId);
```

实现类:

```
@Override
public List<SpuSaleAttr> getSpuSaleAttrList(Long spuId) {
    return spuSaleAttrMapper.selectSpuSaleAttrList(spuId);
}
```

```
@Mapper
public interface SpuSaleAttrMapper extends BaseMapper<SpuSaleAttr> {
    // 根据 spuId 查询销售属性集合
    List<SpuSaleAttr> selectSpuSaleAttrList(Long spuId);
}

SpuSaleAttrMapper.xml

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper SYSTEM "http://mybatis.org/dtd/mybatis-3-mapper.dtd" >
<mapper
    namespace="com.atguigu.gmall.product.mapper.SpuSaleAttrMapper">

    <resultMap id="spuSaleAttrMap"
        type="com.atguigu.gmall.model.product.SpuSaleAttr"
        autoMapping="true">

        <id property="id" column="id"></id>
        <!-- 一对多 -->
        <collection property="spuSaleAttrValueList"
            ofType="com.atguigu.gmall.model.product.SpuSaleAttrValue"
            autoMapping="true">
            <id property="id" column="sale_attr_value_id"></id>
        </collection>
    </resultMap>

    <sql id="spuSaleAttr">
        sa.id ,sa.spu_id, sa.sale_attr_name,sa.base_sale_attr_id,
        sv.id sale_attr_value_id,
        sv.sale_attr_value_name
    </sql>
    <select id="selectSpuSaleAttrList" resultMap="spuSaleAttrMap">
        select
            <include refid="spuSaleAttr"></include>
        from spu_sale_attr sa inner join spu_sale_attr_value sv
        on
            sa.spu_id=sv.spu_id and
        sa.base_sale_attr_id=sv.base_sale_attr_id
        where sa.spu_id=#{spu_id}
    </select>
</mapper>
```

2.2.2 编写控制器

```
* 根据 spuId 查询销售属性集合
* @param spuId
* @return
*/
@GetMapping("spuSaleAttrList/{spuId}")
public Result<List<SpuSaleAttr>>
getSpuSaleAttrList(@PathVariable("spuId") Long spuId) {
    List<SpuSaleAttr> spuSaleAttrList =
manageService.getSpuSaleAttrList(spuId);
    return Result.ok(spuSaleAttrList);
}
```

2.3 点击保存按钮

2.3.1 创建 mapper

创建对应数据库表实体类，以及接口 mapper

```
@Mapper
public interface SkuInfoMapper extends BaseMapper<SkuInfo> {
}
```

```
@Mapper
public interface SkuImageMapper extends BaseMapper<SkuImage> {
}
```

```
@Mapper
public interface SkuAttrValueMapper extends BaseMapper<SkuAttrValue>
{
}
```

```
@Mapper
public interface SkuSaleAttrValueMapper extends
BaseMapper<SkuSaleAttrValue> {
}
```

2.3.2 编写接口与实现

接口

```
/**
 * 保存数据
 * @param skuInfo
```

```
*/
void saveSkuInfo(SkuInfo skuInfo);

实现类

@Override
@Transactional(rollbackFor = Exception.class)
public void saveSkuInfo(SkuInfo skuInfo) {
    /*
        skuInfo 库存单元表 --- spuInfo!
        skuImage 库存单元图片表 --- spuImage!
        skuSaleAttrValue sku 销售属性值表{sku 与销售属性值的中间表} --
    - skuInfo , spuSaleAttrValue
        skuAttrValue sku 与平台属性值的中间表 --- skuInfo ,
    baseAttrValue
    */
    skuInfoMapper.insert(skuInfo);
    List<SkuImage> skuImageList = skuInfo.getSkuImageList();
    if (skuImageList != null && skuImageList.size() > 0) {
        // 循环遍历
        for (SkuImage skuImage : skuImageList) {
            skuImage.setSkuId(skuInfo.getId());
            skuImageMapper.insert(skuImage);
        }
    }

    List<SkuSaleAttrValue> skuSaleAttrValueList =
    skuInfo.getSkuSaleAttrValueList();
    // 调用判断集合方法
    if (!CollectionUtils.isEmpty(skuSaleAttrValueList)) {
        for (SkuSaleAttrValue skuSaleAttrValue :
    skuSaleAttrValueList) {
            skuSaleAttrValue.setSkuId(skuInfo.getId());
            skuSaleAttrValue.setSpuId(skuInfo.getSpuId());
            skuSaleAttrValueMapper.insert(skuSaleAttrValue);
        }
    }

    List<SkuAttrValue> skuAttrValueList =
    skuInfo.getSkuAttrValueList();
    if (!CollectionUtils.isEmpty(skuAttrValueList)) {
        for (SkuAttrValue skuAttrValue : skuAttrValueList) {
            skuAttrValue.setSkuId(skuInfo.getId());
            skuAttrValueMapper.insert(skuAttrValue);
        }
    }
}
```

2.3.3 编写控制器

SkuManageController

```
/**
 * 保存 sku
 * @param skuInfo
 * @return
 */
@PostMapping("saveSkuInfo")
public Result saveSkuInfo(@RequestBody SkuInfo skuInfo) {
    // 调用服务层
    manageService.saveSkuInfo(skuInfo);
    return Result.ok();
}
```

2.4 查询 sku 列表以及上下架处理

2.4.1 编写接口与实现类

在 ManageService 中添加

```
/**
 * SKU 分页列表
 * @param pageParam
 * @return
 */
IPage<SkuInfo> getPage(Page<SkuInfo> pageParam);

/**
 * 商品上架
 * @param skuId
 */
void onSale(Long skuId);

/**
 * 商品下架
 * @param skuId
 */
void cancelSale(Long skuId);
```


接口实现类

```
@Override
public IPage<SkuInfo> getPage(Page<SkuInfo> pageParam) {
    QueryWrapper<SkuInfo> queryWrapper = new QueryWrapper<>();
    queryWrapper.orderByDesc("id");

    IPage<SkuInfo> page = skuInfoMapper.getPage(pageParam,
        queryWrapper);
    return page;
}

@Override
@Transactional
public void onSale(Long skuId) {
    // 更改销售状态
    SkuInfo skuInfoUp = new SkuInfo();
    skuInfoUp.setId(skuId);
    skuInfoUp.setIsSale(1);
    skuInfoMapper.updateById(skuInfoUp);
}

@Override
@Transactional
public void cancelSale(Long skuId) {
    // 更改销售状态
    SkuInfo skuInfoUp = new SkuInfo();
    skuInfoUp.setId(skuId);
    skuInfoUp.setIsSale(0);
    skuInfoMapper.updateById(skuInfoUp);
}
```

2.4.2 编写控制器

BaseManageController 控制器

```
/**
 * SKU 分页列表
 * @param page
 * @param limit
 * @return
 */
@GetMapping("/list/{page}/{limit}")
public Result index(
    @PathVariable Long page,
    @PathVariable Long limit) {
```

```
Page<SkuInfo> pageParam = new Page<>(page, limit);
IPage<SkuInfo> pageModel = manageService.getPage(pageParam);
return Result.ok(pageModel);
}

/**
 * 商品上架
 * @param skuId
 * @return
 */
@GetMapping("onSale/{skuId}")
public Result onSale(@PathVariable("skuId") Long skuId) {
    manageService.onSale(skuId);
    return Result.ok();
}

/**
 * 商品下架
 * @param skuId
 * @return
 */
@GetMapping("cancelSale/{skuId}")
public Result cancelSale(@PathVariable("skuId") Long skuId) {
    manageService.cancelSale(skuId);
    return Result.ok();
}
```