package com.sandu.api.product.model.po;

import com.sandu.api.product.model.Product;

import com.sandu.api.product.model.bo.ProductPropBO;

import com.sandu.api.resmodel.model.bo.ModelAreaTextureRelBO;

import io.swagger.annotations.ApiModelProperty;

import java.io.Serializable;

import java.util.List;

import javax.validation.constraints.Min;

import javax.validation.constraints.NotNull;

import javax.validation.constraints.Size;

public class ProductUpdatePO implements Serializable {

@ApiModelProperty(

value = "产品id",

required = true

)

@NotNull(

message = "id不能为空"

)

private Long id;

@ApiModelProperty(

value = "产品编码",

required = true

)

@NotNull(

message = "产品编码不能为空"

)

private String code;

@NotNull(

message = "产品名称不能为空"

)

@ApiModelProperty(

value = "产品名称",

required = true

)

private String name;

@ApiModelProperty(

value = "品牌ID",

required = true

)

@NotNull(

message = "品牌ID不能为空"

)

private Integer brandId;

@ApiModelProperty(

value = "所属分类编码",

required = true

)

@NotNull(

message = "所属分类编码不能为空"

)

private String categoryCode;

@ApiModelProperty("产品款式")

private Integer modelingId;

@ApiModelProperty("产品系列")

private Integer seriesId;

@ApiModelProperty("售价单位")

private Integer saleUnionId;

@ApiModelProperty("编辑类型:产品库编辑:libraryEdit;线上产品编辑:onlineEdit;渠道产品编辑:channelEdit")

private String editType;

@ApiModelProperty(

value = "产品型号",

required = true

)

@NotNull(

message = "产品型号不能为空"

)

private String modelNumber;

@ApiModelProperty("产品宽")

private String width;

@ApiModelProperty("产品长")

private String length;

@ApiModelProperty("产品高")

private String height;

@ApiModelProperty("产品的属性")

private List<ProductPropBO> props;

@ApiModelProperty("产品缩略图ID")

private String defaultPicId;

@ApiModelProperty("产品所有图片ID,以逗号分隔")

private String picIds;

@ApiModelProperty("建议售价")

private Double advicePrice;

@ApiModelProperty("售价")

private Double price;

@ApiModelProperty("产品风格")

private List<Integer> baseStyleId;

@ApiModelProperty("产品描述")

private String desc;

private Product product;

@ApiModelProperty("模型产品的贴图详情")

private List<ModelAreaTextureRelBO> modelTextureInfos;

@ApiModelProperty("贴图产品的贴图ID")

private Integer textureId;

@ApiModelProperty("模型产品的模型ID")

private Integer modelId;

@ApiModelProperty("用户ID")

@Min(

value = 1L,

message = "请输入正确的用户ID"

)

private Integer userId;

@ApiModelProperty("属性分类编码集合")

private List<String> categoryAttrs;

private List<Integer> categoryIds;

@ApiModelProperty("产品规格")

@Size(

max = 50,

message = "产品规格休息不能超过50个字符"

)

private String productSpec;

public ProductUpdatePO() {

}

public Long getId() {

return this.id;

}

public String getCode() {

return this.code;

}

public String getName() {

return this.name;

}

public Integer getBrandId() {

return this.brandId;

}

public String getCategoryCode() {

return this.categoryCode;

}

public Integer getModelingId() {

return this.modelingId;

}

public Integer getSeriesId() {

return this.seriesId;

}

public Integer getSaleUnionId() {

return this.saleUnionId;

}

public String getEditType() {

return this.editType;

}

public String getModelNumber() {

return this.modelNumber;

}

public String getWidth() {

return this.width;

}

public String getLength() {

return this.length;

}

public String getHeight() {

return this.height;

}

public List<ProductPropBO> getProps() {

return this.props;

}

public String getDefaultPicId() {

return this.defaultPicId;

}

public String getPicIds() {

return this.picIds;

}

public Double getAdvicePrice() {

return this.advicePrice;

}

public Double getPrice() {

return this.price;

}

public List<Integer> getBaseStyleId() {

return this.baseStyleId;

}

public String getDesc() {

return this.desc;

}

public Product getProduct() {

return this.product;

}

public List<ModelAreaTextureRelBO> getModelTextureInfos() {

return this.modelTextureInfos;

}

public Integer getTextureId() {

return this.textureId;

}

public Integer getModelId() {

return this.modelId;

}

public Integer getUserId() {

return this.userId;

}

public List<String> getCategoryAttrs() {

return this.categoryAttrs;

}

public List<Integer> getCategoryIds() {

return this.categoryIds;

}

public String getProductSpec() {

return this.productSpec;

}

public void setId(Long id) {

this.id = id;

}

public void setCode(String code) {

this.code = code;

}

public void setName(String name) {

this.name = name;

}

public void setBrandId(Integer brandId) {

this.brandId = brandId;

}

public void setCategoryCode(String categoryCode) {

this.categoryCode = categoryCode;

}

public void setModelingId(Integer modelingId) {

this.modelingId = modelingId;

}

public void setSeriesId(Integer seriesId) {

this.seriesId = seriesId;

}

public void setSaleUnionId(Integer saleUnionId) {

this.saleUnionId = saleUnionId;

}

public void setEditType(String editType) {

this.editType = editType;

}

public void setModelNumber(String modelNumber) {

this.modelNumber = modelNumber;

}

public void setWidth(String width) {

this.width = width;

}

public void setLength(String length) {

this.length = length;

}

public void setHeight(String height) {

this.height = height;

}

public void setProps(List<ProductPropBO> props) {

this.props = props;

}

public void setDefaultPicId(String defaultPicId) {

this.defaultPicId = defaultPicId;

}

public void setPicIds(String picIds) {

this.picIds = picIds;

}

public void setAdvicePrice(Double advicePrice) {

this.advicePrice = advicePrice;

}

public void setPrice(Double price) {

this.price = price;

}

public void setBaseStyleId(List<Integer> baseStyleId) {

this.baseStyleId = baseStyleId;

}

public void setDesc(String desc) {

this.desc = desc;

}

public void setProduct(Product product) {

this.product = product;

}

public void setModelTextureInfos(List<ModelAreaTextureRelBO> modelTextureInfos) {

this.modelTextureInfos = modelTextureInfos;

}

public void setTextureId(Integer textureId) {

this.textureId = textureId;

}

public void setModelId(Integer modelId) {

this.modelId = modelId;

}

public void setUserId(Integer userId) {

this.userId = userId;

}

public void setCategoryAttrs(List<String> categoryAttrs) {

this.categoryAttrs = categoryAttrs;

}

public void setCategoryIds(List<Integer> categoryIds) {

this.categoryIds = categoryIds;

}

public void setProductSpec(String productSpec) {

this.productSpec = productSpec;

}

public boolean equals(Object o) {

if (o == this) {

return true;

} else if (!(o instanceof ProductUpdatePO)) {

return false;

} else {

ProductUpdatePO other = (ProductUpdatePO)o;

if (!other.canEqual(this)) {

return false;

} else {

label347: {

Object this$id = this.getId();

Object other$id = other.getId();

if (this$id == null) {

if (other$id == null) {

break label347;

}

} else if (this$id.equals(other$id)) {

break label347;

}

return false;

}

Object this$code = this.getCode();

Object other$code = other.getCode();

if (this$code == null) {

if (other$code != null) {

return false;

}

} else if (!this$code.equals(other$code)) {

return false;

}

Object this$name = this.getName();

Object other$name = other.getName();

if (this$name == null) {

if (other$name != null) {

return false;

}

} else if (!this$name.equals(other$name)) {

return false;

}

label326: {

Object this$brandId = this.getBrandId();

Object other$brandId = other.getBrandId();

if (this$brandId == null) {

if (other$brandId == null) {

break label326;

}

} else if (this$brandId.equals(other$brandId)) {

break label326;

}

return false;

}

label319: {

Object this$categoryCode = this.getCategoryCode();

Object other$categoryCode = other.getCategoryCode();

if (this$categoryCode == null) {

if (other$categoryCode == null) {

break label319;

}

} else if (this$categoryCode.equals(other$categoryCode)) {

break label319;

}

return false;

}

label312: {

Object this$modelingId = this.getModelingId();

Object other$modelingId = other.getModelingId();

if (this$modelingId == null) {

if (other$modelingId == null) {

break label312;

}

} else if (this$modelingId.equals(other$modelingId)) {

break label312;

}

return false;

}

Object this$seriesId = this.getSeriesId();

Object other$seriesId = other.getSeriesId();

if (this$seriesId == null) {

if (other$seriesId != null) {

return false;

}

} else if (!this$seriesId.equals(other$seriesId)) {

return false;

}

label298: {

Object this$saleUnionId = this.getSaleUnionId();

Object other$saleUnionId = other.getSaleUnionId();

if (this$saleUnionId == null) {

if (other$saleUnionId == null) {

break label298;

}

} else if (this$saleUnionId.equals(other$saleUnionId)) {

break label298;

}

return false;

}

Object this$editType = this.getEditType();

Object other$editType = other.getEditType();

if (this$editType == null) {

if (other$editType != null) {

return false;

}

} else if (!this$editType.equals(other$editType)) {

return false;

}

label284: {

Object this$modelNumber = this.getModelNumber();

Object other$modelNumber = other.getModelNumber();

if (this$modelNumber == null) {

if (other$modelNumber == null) {

break label284;

}

} else if (this$modelNumber.equals(other$modelNumber)) {

break label284;

}

return false;

}

Object this$width = this.getWidth();

Object other$width = other.getWidth();

if (this$width == null) {

if (other$width != null) {

return false;

}

} else if (!this$width.equals(other$width)) {

return false;

}

Object this$length = this.getLength();

Object other$length = other.getLength();

if (this$length == null) {

if (other$length != null) {

return false;

}

} else if (!this$length.equals(other$length)) {

return false;

}

label263: {

Object this$height = this.getHeight();

Object other$height = other.getHeight();

if (this$height == null) {

if (other$height == null) {

break label263;

}

} else if (this$height.equals(other$height)) {

break label263;

}

return false;

}

label256: {

Object this$props = this.getProps();

Object other$props = other.getProps();

if (this$props == null) {

if (other$props == null) {

break label256;

}

} else if (this$props.equals(other$props)) {

break label256;

}

return false;

}

Object this$defaultPicId = this.getDefaultPicId();

Object other$defaultPicId = other.getDefaultPicId();

if (this$defaultPicId == null) {

if (other$defaultPicId != null) {

return false;

}

} else if (!this$defaultPicId.equals(other$defaultPicId)) {

return false;

}

Object this$picIds = this.getPicIds();

Object other$picIds = other.getPicIds();

if (this$picIds == null) {

if (other$picIds != null) {

return false;

}

} else if (!this$picIds.equals(other$picIds)) {

return false;

}

label235: {

Object this$advicePrice = this.getAdvicePrice();

Object other$advicePrice = other.getAdvicePrice();

if (this$advicePrice == null) {

if (other$advicePrice == null) {

break label235;

}

} else if (this$advicePrice.equals(other$advicePrice)) {

break label235;

}

return false;

}

Object this$price = this.getPrice();

Object other$price = other.getPrice();

if (this$price == null) {

if (other$price != null) {

return false;

}

} else if (!this$price.equals(other$price)) {

return false;

}

Object this$baseStyleId = this.getBaseStyleId();

Object other$baseStyleId = other.getBaseStyleId();

if (this$baseStyleId == null) {

if (other$baseStyleId != null) {

return false;

}

} else if (!this$baseStyleId.equals(other$baseStyleId)) {

return false;

}

label214: {

Object this$desc = this.getDesc();

Object other$desc = other.getDesc();

if (this$desc == null) {

if (other$desc == null) {

break label214;

}

} else if (this$desc.equals(other$desc)) {

break label214;

}

return false;

}

label207: {

Object this$product = this.getProduct();

Object other$product = other.getProduct();

if (this$product == null) {

if (other$product == null) {

break label207;

}

} else if (this$product.equals(other$product)) {

break label207;

}

return false;

}

label200: {

Object this$modelTextureInfos = this.getModelTextureInfos();

Object other$modelTextureInfos = other.getModelTextureInfos();

if (this$modelTextureInfos == null) {

if (other$modelTextureInfos == null) {

break label200;

}

} else if (this$modelTextureInfos.equals(other$modelTextureInfos)) {

break label200;

}

return false;

}

Object this$textureId = this.getTextureId();

Object other$textureId = other.getTextureId();

if (this$textureId == null) {

if (other$textureId != null) {

return false;

}

} else if (!this$textureId.equals(other$textureId)) {

return false;

}

label186: {

Object this$modelId = this.getModelId();

Object other$modelId = other.getModelId();

if (this$modelId == null) {

if (other$modelId == null) {

break label186;

}

} else if (this$modelId.equals(other$modelId)) {

break label186;

}

return false;

}

Object this$userId = this.getUserId();

Object other$userId = other.getUserId();

if (this$userId == null) {

if (other$userId != null) {

return false;

}

} else if (!this$userId.equals(other$userId)) {

return false;

}

label172: {

Object this$categoryAttrs = this.getCategoryAttrs();

Object other$categoryAttrs = other.getCategoryAttrs();

if (this$categoryAttrs == null) {

if (other$categoryAttrs == null) {

break label172;

}

} else if (this$categoryAttrs.equals(other$categoryAttrs)) {

break label172;

}

return false;

}

Object this$categoryIds = this.getCategoryIds();

Object other$categoryIds = other.getCategoryIds();

if (this$categoryIds == null) {

if (other$categoryIds != null) {

return false;

}

} else if (!this$categoryIds.equals(other$categoryIds)) {

return false;

}

Object this$productSpec = this.getProductSpec();

Object other$productSpec = other.getProductSpec();

if (this$productSpec == null) {

if (other$productSpec != null) {

return false;

}

} else if (!this$productSpec.equals(other$productSpec)) {

return false;

}

return true;

}

}

}

protected boolean canEqual(Object other) {

return other instanceof ProductUpdatePO;

}

public int hashCode() {

int PRIME = true;

int result = 1;

Object $id = this.getId();

int result = result \* 59 + ($id == null ? 43 : $id.hashCode());

Object $code = this.getCode();

result = result \* 59 + ($code == null ? 43 : $code.hashCode());

Object $name = this.getName();

result = result \* 59 + ($name == null ? 43 : $name.hashCode());

Object $brandId = this.getBrandId();

result = result \* 59 + ($brandId == null ? 43 : $brandId.hashCode());

Object $categoryCode = this.getCategoryCode();

result = result \* 59 + ($categoryCode == null ? 43 : $categoryCode.hashCode());

Object $modelingId = this.getModelingId();

result = result \* 59 + ($modelingId == null ? 43 : $modelingId.hashCode());

Object $seriesId = this.getSeriesId();

result = result \* 59 + ($seriesId == null ? 43 : $seriesId.hashCode());

Object $saleUnionId = this.getSaleUnionId();

result = result \* 59 + ($saleUnionId == null ? 43 : $saleUnionId.hashCode());

Object $editType = this.getEditType();

result = result \* 59 + ($editType == null ? 43 : $editType.hashCode());

Object $modelNumber = this.getModelNumber();

result = result \* 59 + ($modelNumber == null ? 43 : $modelNumber.hashCode());

Object $width = this.getWidth();

result = result \* 59 + ($width == null ? 43 : $width.hashCode());

Object $length = this.getLength();

result = result \* 59 + ($length == null ? 43 : $length.hashCode());

Object $height = this.getHeight();

result = result \* 59 + ($height == null ? 43 : $height.hashCode());

Object $props = this.getProps();

result = result \* 59 + ($props == null ? 43 : $props.hashCode());

Object $defaultPicId = this.getDefaultPicId();

result = result \* 59 + ($defaultPicId == null ? 43 : $defaultPicId.hashCode());

Object $picIds = this.getPicIds();

result = result \* 59 + ($picIds == null ? 43 : $picIds.hashCode());

Object $advicePrice = this.getAdvicePrice();

result = result \* 59 + ($advicePrice == null ? 43 : $advicePrice.hashCode());

Object $price = this.getPrice();

result = result \* 59 + ($price == null ? 43 : $price.hashCode());

Object $baseStyleId = this.getBaseStyleId();

result = result \* 59 + ($baseStyleId == null ? 43 : $baseStyleId.hashCode());

Object $desc = this.getDesc();

result = result \* 59 + ($desc == null ? 43 : $desc.hashCode());

Object $product = this.getProduct();

result = result \* 59 + ($product == null ? 43 : $product.hashCode());

Object $modelTextureInfos = this.getModelTextureInfos();

result = result \* 59 + ($modelTextureInfos == null ? 43 : $modelTextureInfos.hashCode());

Object $textureId = this.getTextureId();

result = result \* 59 + ($textureId == null ? 43 : $textureId.hashCode());

Object $modelId = this.getModelId();

result = result \* 59 + ($modelId == null ? 43 : $modelId.hashCode());

Object $userId = this.getUserId();

result = result \* 59 + ($userId == null ? 43 : $userId.hashCode());

Object $categoryAttrs = this.getCategoryAttrs();

result = result \* 59 + ($categoryAttrs == null ? 43 : $categoryAttrs.hashCode());

Object $categoryIds = this.getCategoryIds();

result = result \* 59 + ($categoryIds == null ? 43 : $categoryIds.hashCode());

Object $productSpec = this.getProductSpec();

result = result \* 59 + ($productSpec == null ? 43 : $productSpec.hashCode());

return result;

}

public String toString() {

return "ProductUpdatePO(id=" + this.getId() + ", code=" + this.getCode() + ", name=" + this.getName() + ", brandId=" + this.getBrandId() + ", categoryCode=" + this.getCategoryCode() + ", modelingId=" + this.getModelingId() + ", seriesId=" + this.getSeriesId() + ", saleUnionId=" + this.getSaleUnionId() + ", editType=" + this.getEditType() + ", modelNumber=" + this.getModelNumber() + ", width=" + this.getWidth() + ", length=" + this.getLength() + ", height=" + this.getHeight() + ", props=" + this.getProps() + ", defaultPicId=" + this.getDefaultPicId() + ", picIds=" + this.getPicIds() + ", advicePrice=" + this.getAdvicePrice() + ", price=" + this.getPrice() + ", baseStyleId=" + this.getBaseStyleId() + ", desc=" + this.getDesc() + ", product=" + this.getProduct() + ", modelTextureInfos=" + this.getModelTextureInfos() + ", textureId=" + this.getTextureId() + ", modelId=" + this.getModelId() + ", userId=" + this.getUserId() + ", categoryAttrs=" + this.getCategoryAttrs() + ", categoryIds=" + this.getCategoryIds() + ", productSpec=" + this.getProductSpec() + ")";

}

}

//

// Source code recreated from a .class file by IntelliJ IDEA

// (powered by Fernflower decompiler)

//

package com.sandu.api.product.model.po;

import java.io.Serializable;

public class AlloProductPO implements Serializable {

private Integer categoryId;

private Integer brandId;

private String productName;

private String productType;

private Integer productSerierId;

private Integer productStyleId;

private Integer productTextureId;

private String allotPlatformType;

public AlloProductPO() {

}

public Integer getCategoryId() {

return this.categoryId;

}

public Integer getBrandId() {

return this.brandId;

}

public String getProductName() {

return this.productName;

}

public String getProductType() {

return this.productType;

}

public Integer getProductSerierId() {

return this.productSerierId;

}

public Integer getProductStyleId() {

return this.productStyleId;

}

public Integer getProductTextureId() {

return this.productTextureId;

}

public String getAllotPlatformType() {

return this.allotPlatformType;

}

public void setCategoryId(Integer categoryId) {

this.categoryId = categoryId;

}

public void setBrandId(Integer brandId) {

this.brandId = brandId;

}

public void setProductName(String productName) {

this.productName = productName;

}

public void setProductType(String productType) {

this.productType = productType;

}

public void setProductSerierId(Integer productSerierId) {

this.productSerierId = productSerierId;

}

public void setProductStyleId(Integer productStyleId) {

this.productStyleId = productStyleId;

}

public void setProductTextureId(Integer productTextureId) {

this.productTextureId = productTextureId;

}

public void setAllotPlatformType(String allotPlatformType) {

this.allotPlatformType = allotPlatformType;

}

public boolean equals(Object o) {

if (o == this) {

return true;

} else if (!(o instanceof AlloProductPO)) {

return false;

} else {

AlloProductPO other = (AlloProductPO)o;

if (!other.canEqual(this)) {

return false;

} else {

label107: {

Object this$categoryId = this.getCategoryId();

Object other$categoryId = other.getCategoryId();

if (this$categoryId == null) {

if (other$categoryId == null) {

break label107;

}

} else if (this$categoryId.equals(other$categoryId)) {

break label107;

}

return false;

}

Object this$brandId = this.getBrandId();

Object other$brandId = other.getBrandId();

if (this$brandId == null) {

if (other$brandId != null) {

return false;

}

} else if (!this$brandId.equals(other$brandId)) {

return false;

}

Object this$productName = this.getProductName();

Object other$productName = other.getProductName();

if (this$productName == null) {

if (other$productName != null) {

return false;

}

} else if (!this$productName.equals(other$productName)) {

return false;

}

label86: {

Object this$productType = this.getProductType();

Object other$productType = other.getProductType();

if (this$productType == null) {

if (other$productType == null) {

break label86;

}

} else if (this$productType.equals(other$productType)) {

break label86;

}

return false;

}

label79: {

Object this$productSerierId = this.getProductSerierId();

Object other$productSerierId = other.getProductSerierId();

if (this$productSerierId == null) {

if (other$productSerierId == null) {

break label79;

}

} else if (this$productSerierId.equals(other$productSerierId)) {

break label79;

}

return false;

}

label72: {

Object this$productStyleId = this.getProductStyleId();

Object other$productStyleId = other.getProductStyleId();

if (this$productStyleId == null) {

if (other$productStyleId == null) {

break label72;

}

} else if (this$productStyleId.equals(other$productStyleId)) {

break label72;

}

return false;

}

Object this$productTextureId = this.getProductTextureId();

Object other$productTextureId = other.getProductTextureId();

if (this$productTextureId == null) {

if (other$productTextureId != null) {

return false;

}

} else if (!this$productTextureId.equals(other$productTextureId)) {

return false;

}

Object this$allotPlatformType = this.getAllotPlatformType();

Object other$allotPlatformType = other.getAllotPlatformType();

if (this$allotPlatformType == null) {

if (other$allotPlatformType != null) {

return false;

}

} else if (!this$allotPlatformType.equals(other$allotPlatformType)) {

return false;

}

return true;

}

}

}

protected boolean canEqual(Object other) {

return other instanceof AlloProductPO;

}

public int hashCode() {

int PRIME = true;

int result = 1;

Object $categoryId = this.getCategoryId();

int result = result \* 59 + ($categoryId == null ? 43 : $categoryId.hashCode());

Object $brandId = this.getBrandId();

result = result \* 59 + ($brandId == null ? 43 : $brandId.hashCode());

Object $productName = this.getProductName();

result = result \* 59 + ($productName == null ? 43 : $productName.hashCode());

Object $productType = this.getProductType();

result = result \* 59 + ($productType == null ? 43 : $productType.hashCode());

Object $productSerierId = this.getProductSerierId();

result = result \* 59 + ($productSerierId == null ? 43 : $productSerierId.hashCode());

Object $productStyleId = this.getProductStyleId();

result = result \* 59 + ($productStyleId == null ? 43 : $productStyleId.hashCode());

Object $productTextureId = this.getProductTextureId();

result = result \* 59 + ($productTextureId == null ? 43 : $productTextureId.hashCode());

Object $allotPlatformType = this.getAllotPlatformType();

result = result \* 59 + ($allotPlatformType == null ? 43 : $allotPlatformType.hashCode());

return result;

}

public String toString() {

return "AlloProductPO(categoryId=" + this.getCategoryId() + ", brandId=" + this.getBrandId() + ", productName=" + this.getProductName() + ", productType=" + this.getProductType() + ", productSerierId=" + this.getProductSerierId() + ", productStyleId=" + this.getProductStyleId() + ", productTextureId=" + this.getProductTextureId() + ", allotPlatformType=" + this.getAllotPlatformType() + ")";

}

}

<?xml version="1.0" encoding="UTF-8" ?>

<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd" >

<mapper namespace="com.sandu.service.product.dao.ProductDao">

<!--<cache type="org.mybatis.caches.ehcache.LoggingEhcache"/>-->

<!-- \*\*插入定义\*\* -->

<insert id="saveProduct" parameterType="com.sandu.api.product.model.Product" useGeneratedKeys="true"

keyProperty="id">

insert into base\_product

<trim prefix="(" suffix=")" suffixOverrides=",">

<if test="id != null">

id,

</if>

<if test="productCode != null">

product\_code,

</if>

<if test="productName != null">

product\_name,

</if>

<if test="brandId != null">

brand\_id,

</if>

<if test="proStyleId != null">

pro\_style\_id,

</if>

<if test="productSpec != null">

product\_spec,

</if>

<if test="colorId != null">

color\_id,

</if>

<if test="productLength != null">

product\_length,

</if>

<if test="productWidth != null">

product\_width,

</if>

<if test="productHeight != null">

product\_height,

</if>

<if test="salePrice != null">

sale\_price,

</if>

<if test="advicePrice != null">

advice\_price,

</if>

<if test="picId != null">

pic\_id,

</if>

<if test="modelId != null">

model\_id,

</if>

<if test="productDesc != null">

product\_desc,

</if>

<if test="sysCode != null">

sys\_code,

</if>

<if test="creator != null">

creator,

</if>

<if test="gmtCreate != null">

gmt\_create,

</if>

<if test="modifier != null">

modifier,

</if>

<if test="gmtModified != null">

gmt\_modified,

</if>

<if test="isDeleted != null">

is\_deleted,

</if>

<if test="picIds != null">

pic\_ids,

</if>

<if test="materialPicIds != null">

material\_pic\_ids,

</if>

<if test="houseTypeValues != null">

house\_type\_values,

</if>

<if test="productTypeValue != null">

product\_type\_value,

</if>

<if test="u3dmodelId != null">

u3dmodel\_Id,

</if>

<if test="mergeProductIds != null">

merge\_product\_ids,

</if>

<if test="putawayModified != null">

putaway\_modified,

</if>

<if test="dateAtt2 != null">

date\_att2,

</if>

<if test="productSmallTypeValue != null">

product\_small\_type\_value,

</if>

<if test="putawayState != null">

putaway\_state,

</if>

<if test="numAtt3 != null">

num\_att3,

</if>

<if test="numAtt4 != null">

num\_att4,

</if>

<if test="remark != null">

remark,

</if>

<if test="ipadU3dmodelId != null">

ipad\_u3dmodel\_id,

</if>

<if test="iosU3dmodelId != null">

ios\_u3dmodel\_id,

</if>

<if test="androidU3dmodelId != null">

android\_u3dmodel\_id,

</if>

<if test="macbookU3dmodelId != null">

macbook\_u3dmodel\_id,

</if>

<if test="windowsU3dmodelId != null">

windows\_u3dmodel\_id,

</if>

<if test="webU3dmodelId != null">

web\_u3dmodel\_id,

</if>

<if test="decorationId != null">

decoration\_id,

</if>

<if test="designerId != null">

designer\_id,

</if>

<if test="modelingId != null">

modeling\_id,

</if>

<if test="renderingId != null">

rendering\_id,

</if>

<if test="productTypeMark != null">

product\_type\_mark,

</if>

<if test="productSmallTypeMark != null">

product\_small\_type\_mark,

</if>

<if test="originalProductCode != null">

original\_product\_code,

</if>

<if test="spacecomonId != null">

spaceComon\_id,

</if>

<if test="productShortCode != null">

product\_short\_code,

</if>

<if test="bmIds != null">

bm\_ids,

</if>

<if test="codeNumber != null">

code\_number,

</if>

<if test="parentId != null">

parent\_id,

</if>

<if test="designtempletId != null">

designTemplet\_id,

</if>

<if test="materialFileId != null">

material\_file\_id,

</if>

<if test="material3dPicIds != null">

material3d\_pic\_ids,

</if>

<if test="syncStatus != null">

sync\_status,

</if>

<if test="codeIsEmploy != null">

code\_is\_employ,

</if>

<if test="codeBatch != null">

code\_batch,

</if>

<if test="productModelNumber != null">

product\_model\_number,

</if>

<if test="fullPaveLength != null">

full\_pave\_length,

</if>

<if test="testModified != null">

test\_modified,

</if>

<if test="salePriceValue != null">

sale\_price\_value,

</if>

<if test="minHeight != null">

min\_height,

</if>

<if test="productMaterialDesc != null">

product\_material\_desc,

</if>

<if test="colorsLongCode != null">

colors\_long\_code,

</if>

<if test="configId != null">

config\_id,

</if>

<if test="fbxFileId != null">

fbx\_file\_id,

</if>

<if test="fbxTexture != null">

fbx\_texture,

</if>

<if test="fbxState != null">

fbx\_state,

</if>

<if test="productStyleIdInfo != null">

product\_style\_id\_info,

</if>

<if test="styleId != null">

style\_id,

</if>

<if test="regionMark != null">

region\_mark,

</if>

<if test="measureCode != null">

measure\_code,

</if>

<if test="wallType != null">

wall\_type,

</if>

<if test="productSmallpoxIdentify != null">

product\_smallpox\_identify,

</if>

<if test="productGroundIdentify != null">

product\_ground\_identify,

</if>

<if test="seriesId != null">

series\_id,

</if>

<if test="spellingFlowerFileId != null">

spelling\_flower\_file\_id,

</if>

<if test="appPicTextureIds != null">

app\_pic\_texture\_ids,

</if>

<if test="productSmallpoxIdentifyStr != null">

product\_smallpox\_identify\_str,

</if>

<if test="splitTexturesInfo != null">

split\_textures\_info,

</if>

<if test="companyId != null">

company\_id,

</if>

<if test="originId != null">

origin\_id,

</if>

<if test="goodsSpuId != null">

goods\_spu\_id,

</if>

<if test="secrecyFlag != null">

secrecy\_Flag,

</if>

<if test="productBatchId != null">

product\_batch\_id,

</if>

<if test="productBatchType != null">

product\_batch\_type,

</if>

<if test="isCreatedTexture != null">

is\_created\_texture,

</if>

<if test="isComplexParquet != null">

is\_complex\_parquet,

</if>

</trim>

<trim prefix="values (" suffix=")" suffixOverrides=",">

<if test="id != null">

#{id,jdbcType=BIGINT},

</if>

<if test="productCode != null">

#{productCode,jdbcType=VARCHAR},

</if>

<if test="productName != null">

#{productName,jdbcType=VARCHAR},

</if>

<if test="brandId != null">

#{brandId,jdbcType=INTEGER},

</if>

<if test="proStyleId != null">

#{proStyleId,jdbcType=INTEGER},

</if>

<if test="productSpec != null">

#{productSpec,jdbcType=VARCHAR},

</if>

<if test="colorId != null">

#{colorId,jdbcType=INTEGER},

</if>

<if test="productLength != null">

#{productLength,jdbcType=VARCHAR},

</if>

<if test="productWidth != null">

#{productWidth,jdbcType=VARCHAR},

</if>

<if test="productHeight != null">

#{productHeight,jdbcType=VARCHAR},

</if>

<if test="salePrice != null">

#{salePrice,jdbcType=DECIMAL},

</if>

<if test="advicePrice != null">

#{advicePrice,jdbcType=DECIMAL},

</if>

<if test="picId != null">

#{picId,jdbcType=INTEGER},

</if>

<if test="modelId != null">

#{modelId,jdbcType=INTEGER},

</if>

<if test="productDesc != null">

#{productDesc,jdbcType=VARCHAR},

</if>

<if test="sysCode != null">

#{sysCode,jdbcType=VARCHAR},

</if>

<if test="creator != null">

#{creator,jdbcType=VARCHAR},

</if>

<if test="gmtCreate != null">

#{gmtCreate,jdbcType=TIMESTAMP},

</if>

<if test="modifier != null">

#{modifier,jdbcType=VARCHAR},

</if>

<if test="gmtModified != null">

#{gmtModified,jdbcType=TIMESTAMP},

</if>

<if test="isDeleted != null">

#{isDeleted,jdbcType=INTEGER},

</if>

<if test="picIds != null">

#{picIds,jdbcType=VARCHAR},

</if>

<if test="materialPicIds != null">

#{materialPicIds,jdbcType=VARCHAR},

</if>

<if test="houseTypeValues != null">

#{houseTypeValues,jdbcType=VARCHAR},

</if>

<if test="productTypeValue != null">

#{productTypeValue,jdbcType=VARCHAR},

</if>

<if test="u3dmodelId != null">

#{u3dmodelId,jdbcType=VARCHAR},

</if>

<if test="mergeProductIds != null">

#{mergeProductIds,jdbcType=VARCHAR},

</if>

<if test="putawayModified != null">

#{putawayModified,jdbcType=TIMESTAMP},

</if>

<if test="dateAtt2 != null">

#{dateAtt2,jdbcType=TIMESTAMP},

</if>

<if test="productSmallTypeValue != null">

#{productSmallTypeValue,jdbcType=INTEGER},

</if>

<if test="putawayState != null">

#{putawayState,jdbcType=INTEGER},

</if>

<if test="numAtt3 != null">

#{numAtt3,jdbcType=DECIMAL},

</if>

<if test="numAtt4 != null">

#{numAtt4,jdbcType=DECIMAL},

</if>

<if test="remark != null">

#{remark,jdbcType=VARCHAR},

</if>

<if test="ipadU3dmodelId != null">

#{ipadU3dmodelId,jdbcType=INTEGER},

</if>

<if test="iosU3dmodelId != null">

#{iosU3dmodelId,jdbcType=INTEGER},

</if>

<if test="androidU3dmodelId != null">

#{androidU3dmodelId,jdbcType=INTEGER},

</if>

<if test="macbookU3dmodelId != null">

#{macbookU3dmodelId,jdbcType=INTEGER},

</if>

<if test="windowsU3dmodelId != null">

#{windowsU3dmodelId,jdbcType=INTEGER},

</if>

<if test="webU3dmodelId != null">

#{webU3dmodelId,jdbcType=INTEGER},

</if>

<if test="decorationId != null">

#{decorationId,jdbcType=VARCHAR},

</if>

<if test="designerId != null">

#{designerId,jdbcType=INTEGER},

</if>

<if test="modelingId != null">

#{modelingId,jdbcType=INTEGER},

</if>

<if test="renderingId != null">

#{renderingId,jdbcType=INTEGER},

</if>

<if test="productTypeMark != null">

#{productTypeMark,jdbcType=VARCHAR},

</if>

<if test="productSmallTypeMark != null">

#{productSmallTypeMark,jdbcType=VARCHAR},

</if>

<if test="originalProductCode != null">

#{originalProductCode,jdbcType=VARCHAR},

</if>

<if test="spacecomonId != null">

#{spacecomonId,jdbcType=INTEGER},

</if>

<if test="productShortCode != null">

#{productShortCode,jdbcType=VARCHAR},

</if>

<if test="bmIds != null">

#{bmIds,jdbcType=VARCHAR},

</if>

<if test="codeNumber != null">

#{codeNumber,jdbcType=INTEGER},

</if>

<if test="parentId != null">

#{parentId,jdbcType=INTEGER},

</if>

<if test="designtempletId != null">

#{designtempletId,jdbcType=INTEGER},

</if>

<if test="materialFileId != null">

#{materialFileId,jdbcType=INTEGER},

</if>

<if test="material3dPicIds != null">

#{material3dPicIds,jdbcType=VARCHAR},

</if>

<if test="syncStatus != null">

#{syncStatus,jdbcType=VARCHAR},

</if>

<if test="codeIsEmploy != null">

#{codeIsEmploy,jdbcType=INTEGER},

</if>

<if test="codeBatch != null">

#{codeBatch,jdbcType=VARCHAR},

</if>

<if test="productModelNumber != null">

#{productModelNumber,jdbcType=VARCHAR},

</if>

<if test="fullPaveLength != null">

#{fullPaveLength,jdbcType=VARCHAR},

</if>

<if test="testModified != null">

#{testModified,jdbcType=TIMESTAMP},

</if>

<if test="salePriceValue != null">

#{salePriceValue,jdbcType=INTEGER},

</if>

<if test="minHeight != null">

#{minHeight,jdbcType=VARCHAR},

</if>

<if test="productMaterialDesc != null">

#{productMaterialDesc,jdbcType=VARCHAR},

</if>

<if test="colorsLongCode != null">

#{colorsLongCode,jdbcType=VARCHAR},

</if>

<if test="configId != null">

#{configId,jdbcType=INTEGER},

</if>

<if test="fbxFileId != null">

#{fbxFileId,jdbcType=INTEGER},

</if>

<if test="fbxTexture != null">

#{fbxTexture,jdbcType=VARCHAR},

</if>

<if test="fbxState != null">

#{fbxState,jdbcType=INTEGER},

</if>

<if test="productStyleIdInfo != null">

#{productStyleIdInfo,jdbcType=VARCHAR},

</if>

<if test="styleId != null">

#{styleId,jdbcType=BIGINT},

</if>

<if test="regionMark != null">

#{regionMark,jdbcType=VARCHAR},

</if>

<if test="measureCode != null">

#{measureCode,jdbcType=VARCHAR},

</if>

<if test="wallType != null">

#{wallType,jdbcType=VARCHAR},

</if>

<if test="productSmallpoxIdentify != null">

#{productSmallpoxIdentify,jdbcType=INTEGER},

</if>

<if test="productGroundIdentify != null">

#{productGroundIdentify,jdbcType=INTEGER},

</if>

<if test="seriesId != null">

#{seriesId,jdbcType=INTEGER},

</if>

<if test="spellingFlowerFileId != null">

#{spellingFlowerFileId,jdbcType=INTEGER},

</if>

<if test="appPicTextureIds != null">

#{appPicTextureIds,jdbcType=VARCHAR},

</if>

<if test="productSmallpoxIdentifyStr != null">

#{productSmallpoxIdentifyStr,jdbcType=VARCHAR},

</if>

<if test="splitTexturesInfo != null">

#{splitTexturesInfo,jdbcType=LONGVARCHAR},

</if>

<if test="companyId != null">

#{companyId},

</if>

<if test="originId != null">

#{originId},

</if>

<if test="goodsSpuId != null">

#{goodsSpuId},

</if>

<if test="secrecyFlag != null">

#{secrecyFlag},

</if>

<if test="productBatchId != null">

#{productBatchId},

</if>

<if test="productBatchType != null">

#{productBatchType},

</if>

<if test="isCreatedTexture != null">

#{isCreatedTexture},

</if>

<if test="isComplexParquet != null">

#{isComplexParquet},

</if>

</trim>

</insert>

<!-- \*\*更新定义\*\* -->

<update id="updateProduct" parameterType="com.sandu.api.product.model.Product">

update base\_product

<set>

<if test="productCode != null">

product\_code = #{productCode,jdbcType=VARCHAR},

</if>

<if test="productName != null">

product\_name = #{productName,jdbcType=VARCHAR},

</if>

<if test="brandId != null">

brand\_id = #{brandId,jdbcType=INTEGER},

</if>

<if test="proStyleId != null">

pro\_style\_id = #{proStyleId,jdbcType=INTEGER},

</if>

<if test="productSpec != null">

product\_spec = #{productSpec,jdbcType=VARCHAR},

</if>

<if test="colorId != null">

color\_id = #{colorId,jdbcType=INTEGER},

</if>

<if test="productLength != null">

product\_length = #{productLength,jdbcType=VARCHAR},

</if>

<if test="productWidth != null">

product\_width = #{productWidth,jdbcType=VARCHAR},

</if>

<if test="productHeight != null">

product\_height = #{productHeight,jdbcType=VARCHAR},

</if>

<if test="salePrice != null">

sale\_price = #{salePrice,jdbcType=DECIMAL},

</if>

<if test="advicePrice != null">

advice\_price = #{advicePrice,jdbcType=DECIMAL},

</if>

<if test="picId != null">

pic\_id = #{picId,jdbcType=INTEGER},

</if>

<if test="modelId != null">

model\_id = #{modelId,jdbcType=INTEGER},

</if>

<if test="productDesc != null">

product\_desc = #{productDesc,jdbcType=VARCHAR},

</if>

<if test="sysCode != null">

sys\_code = #{sysCode,jdbcType=VARCHAR},

</if>

<if test="creator != null">

creator = #{creator,jdbcType=VARCHAR},

</if>

<if test="gmtCreate != null">

gmt\_create = #{gmtCreate,jdbcType=TIMESTAMP},

</if>

<if test="modifier != null">

modifier = #{modifier,jdbcType=VARCHAR},

</if>

<if test="gmtModified != null">

gmt\_modified = #{gmtModified,jdbcType=TIMESTAMP},

</if>

<if test="isDeleted != null">

is\_deleted = #{isDeleted,jdbcType=INTEGER},

</if>

<if test="picIds != null">

pic\_ids = #{picIds,jdbcType=VARCHAR},

</if>

<if test="materialPicIds != null">

material\_pic\_ids = #{materialPicIds,jdbcType=VARCHAR},

</if>

<if test="houseTypeValues != null">

house\_type\_values = #{houseTypeValues,jdbcType=VARCHAR},

</if>

<if test="productTypeValue != null">

product\_type\_value = #{productTypeValue,jdbcType=VARCHAR},

</if>

<if test="u3dmodelId != null">

u3dmodel\_Id = #{u3dmodelId,jdbcType=VARCHAR},

</if>

<if test="mergeProductIds != null">

merge\_product\_ids = #{mergeProductIds,jdbcType=VARCHAR},

</if>

<if test="putawayModified != null">

putaway\_modified = #{putawayModified,jdbcType=TIMESTAMP},

</if>

<if test="dateAtt2 != null">

date\_att2 = #{dateAtt2,jdbcType=TIMESTAMP},

</if>

<if test="productSmallTypeValue != null">

product\_small\_type\_value = #{productSmallTypeValue,jdbcType=INTEGER},

</if>

<if test="putawayState != null">

putaway\_state = #{putawayState,jdbcType=INTEGER},

</if>

<if test="numAtt3 != null">

num\_att3 = #{numAtt3,jdbcType=DECIMAL},

</if>

<if test="numAtt4 != null">

num\_att4 = #{numAtt4,jdbcType=DECIMAL},

</if>

<if test="remark != null">

remark = #{remark,jdbcType=VARCHAR},

</if>

<if test="ipadU3dmodelId != null">

ipad\_u3dmodel\_id = #{ipadU3dmodelId,jdbcType=INTEGER},

</if>

<if test="iosU3dmodelId != null">

ios\_u3dmodel\_id = #{iosU3dmodelId,jdbcType=INTEGER},

</if>

<if test="androidU3dmodelId != null">

android\_u3dmodel\_id = #{androidU3dmodelId,jdbcType=INTEGER},

</if>

<if test="macbookU3dmodelId != null">

macbook\_u3dmodel\_id = #{macbookU3dmodelId,jdbcType=INTEGER},

</if>

<if test="windowsU3dmodelId != null">

windows\_u3dmodel\_id = #{windowsU3dmodelId,jdbcType=INTEGER},

</if>

<if test="webU3dmodelId != null">

web\_u3dmodel\_id = #{webU3dmodelId,jdbcType=INTEGER},

</if>

<if test="decorationId != null">

decoration\_id = #{decorationId,jdbcType=VARCHAR},

</if>

<if test="designerId != null">

designer\_id = #{designerId,jdbcType=INTEGER},

</if>

<if test="modelingId != null">

modeling\_id = #{modelingId,jdbcType=INTEGER},

</if>

<if test="renderingId != null">

rendering\_id = #{renderingId,jdbcType=INTEGER},

</if>

<if test="productTypeMark != null">

product\_type\_mark = #{productTypeMark,jdbcType=VARCHAR},

</if>

<if test="productSmallTypeMark != null">

product\_small\_type\_mark = #{productSmallTypeMark,jdbcType=VARCHAR},

</if>

<if test="originalProductCode != null">

original\_product\_code = #{originalProductCode,jdbcType=VARCHAR},

</if>

<if test="spacecomonId != null">

spaceComon\_id = #{spacecomonId,jdbcType=INTEGER},

</if>

<if test="productShortCode != null">

product\_short\_code = #{productShortCode,jdbcType=VARCHAR},

</if>

<if test="bmIds != null">

bm\_ids = #{bmIds,jdbcType=VARCHAR},

</if>

<if test="codeNumber != null">

code\_number = #{codeNumber,jdbcType=INTEGER},

</if>

<if test="parentId != null">

parent\_id = #{parentId,jdbcType=INTEGER},

</if>

<if test="designtempletId != null">

designTemplet\_id = #{designtempletId,jdbcType=INTEGER},

</if>

<if test="materialFileId != null">

material\_file\_id = #{materialFileId,jdbcType=INTEGER},

</if>

<if test="material3dPicIds != null">

material3d\_pic\_ids = #{material3dPicIds,jdbcType=VARCHAR},

</if>

<if test="syncStatus != null">

sync\_status = #{syncStatus,jdbcType=VARCHAR},

</if>

<if test="codeIsEmploy != null">

code\_is\_employ = #{codeIsEmploy,jdbcType=INTEGER},

</if>

<if test="codeBatch != null">

code\_batch = #{codeBatch,jdbcType=VARCHAR},

</if>

<if test="productModelNumber != null">

product\_model\_number = #{productModelNumber,jdbcType=VARCHAR},

</if>

<if test="fullPaveLength != null">

full\_pave\_length = #{fullPaveLength,jdbcType=VARCHAR},

</if>

<if test="testModified != null">

test\_modified = #{testModified,jdbcType=TIMESTAMP},

</if>

<if test="salePriceValue != null">

sale\_price\_value = #{salePriceValue,jdbcType=INTEGER},

</if>

<if test="minHeight != null">

min\_height = #{minHeight,jdbcType=VARCHAR},

</if>

<if test="productMaterialDesc != null">

product\_material\_desc = #{productMaterialDesc,jdbcType=VARCHAR},

</if>

<if test="colorsLongCode != null">

colors\_long\_code = #{colorsLongCode,jdbcType=VARCHAR},

</if>

<if test="configId != null">

config\_id = #{configId,jdbcType=INTEGER},

</if>

<if test="fbxFileId != null">

fbx\_file\_id = #{fbxFileId,jdbcType=INTEGER},

</if>

<if test="fbxTexture != null">

fbx\_texture = #{fbxTexture,jdbcType=VARCHAR},

</if>

<if test="fbxState != null">

fbx\_state = #{fbxState,jdbcType=INTEGER},

</if>

<if test="productStyleIdInfo != null">

product\_style\_id\_info = #{productStyleIdInfo,jdbcType=VARCHAR},

</if>

<if test="styleId != null">

style\_id = #{styleId,jdbcType=BIGINT},

</if>

<if test="regionMark != null">

region\_mark = #{regionMark,jdbcType=VARCHAR},

</if>

<if test="measureCode != null">

measure\_code = #{measureCode,jdbcType=VARCHAR},

</if>

<if test="wallType != null">

wall\_type = #{wallType,jdbcType=VARCHAR},

</if>

<if test="productSmallpoxIdentify != null">

product\_smallpox\_identify = #{productSmallpoxIdentify,jdbcType=INTEGER},

</if>

<if test="productGroundIdentify != null">

product\_ground\_identify = #{productGroundIdentify,jdbcType=INTEGER},

</if>

<if test="seriesId != null">

series\_id = #{seriesId,jdbcType=INTEGER},

</if>

<if test="spellingFlowerFileId != null">

spelling\_flower\_file\_id = #{spellingFlowerFileId,jdbcType=INTEGER},

</if>

<if test="appPicTextureIds != null">

app\_pic\_texture\_ids = #{appPicTextureIds,jdbcType=VARCHAR},

</if>

<if test="productSmallpoxIdentifyStr != null">

product\_smallpox\_identify\_str = #{productSmallpoxIdentifyStr,jdbcType=VARCHAR},

</if>

<if test="splitTexturesInfo != null">

split\_textures\_info = #{splitTexturesInfo,jdbcType=LONGVARCHAR},

</if>

<if test="secrecyFlag != null">

secrecy\_flag = #{secrecyFlag},

</if>

<if test="productBatchId != null">

product\_batch\_id = #{productBatchId},

</if>

<if test="productBatchType != null">

product\_batch\_type= #{productBatchType},

</if>

<if test="isCreatedTexture != null">

is\_created\_texture = #{isCreatedTexture},

</if>

<if test="isComplexParquet != null">

is\_complex\_parquet = #{isComplexParquet},

</if>

</set>

where id = #{id,jdbcType=BIGINT}

</update>

<!-- \*\*删除定义\*\* -->

<update id="deleteProductById">

UPDATE base\_product SET is\_deleted = 1

where id = #{id,jdbcType=INTEGER}

</update>

<update id="deleteProductByCode" parameterType="java.lang.String">

UPDATE base\_product SET is\_deleted = 1

where product\_code = #{productCode,jdbcType=VARCHAR}

</update>

<!-- \*\*常量定义\*\* -->

<sql id="Base\_Column\_List">

id, product\_code, product\_name, brand\_id, pro\_style\_id, product\_spec, color\_id, product\_length,

product\_width, product\_height, sale\_price, advice\_price, pic\_id, model\_id, product\_desc, sys\_code,

creator, gmt\_create, modifier, gmt\_modified, is\_deleted, pic\_ids, material\_pic\_ids,

house\_type\_values, product\_type\_value, u3dmodel\_Id, merge\_product\_ids, putaway\_modified,

date\_att2, product\_small\_type\_value, putaway\_state, num\_att3, num\_att4, remark, ipad\_u3dmodel\_id,

ios\_u3dmodel\_id, android\_u3dmodel\_id, macbook\_u3dmodel\_id, windows\_u3dmodel\_id, web\_u3dmodel\_id,

decoration\_id, designer\_id, modeling\_id, rendering\_id, product\_type\_mark, product\_small\_type\_mark,

original\_product\_code, spaceComon\_id, product\_short\_code, bm\_ids, code\_number, parent\_id,

designTemplet\_id, material\_file\_id, material3d\_pic\_ids, sync\_status, code\_is\_employ,

code\_batch, product\_model\_number, full\_pave\_length, test\_modified, sale\_price\_value,

min\_height, product\_material\_desc, colors\_long\_code, config\_id, fbx\_file\_id, fbx\_texture,

fbx\_state, product\_style\_id\_info, style\_id, region\_mark, measure\_code, wall\_type,

product\_smallpox\_identify, product\_ground\_identify, series\_id, spelling\_flower\_file\_id,

app\_pic\_texture\_ids, product\_smallpox\_identify\_str, secrecy\_flag , product\_batch\_id, product\_batch\_type,is\_created\_texture,

is\_complex\_parquet

</sql>

<!-- \*\*结果定义\*\* -->

<resultMap id="AllResultMap" type="com.sandu.api.product.model.Product">

<id column="id" property="id" jdbcType="BIGINT"/>

<result column="product\_code" property="productCode" jdbcType="VARCHAR"/>

<result column="product\_name" property="productName" jdbcType="VARCHAR"/>

<result column="brand\_id" property="brandId" jdbcType="INTEGER"/>

<result column="pro\_style\_id" property="proStyleId" jdbcType="INTEGER"/>

<result column="product\_spec" property="productSpec" jdbcType="VARCHAR"/>

<result column="color\_id" property="colorId" jdbcType="INTEGER"/>

<result column="product\_length" property="productLength" jdbcType="VARCHAR"/>

<result column="product\_width" property="productWidth" jdbcType="VARCHAR"/>

<result column="product\_height" property="productHeight" jdbcType="VARCHAR"/>

<result column="sale\_price" property="salePrice" jdbcType="DECIMAL"/>

<result column="advice\_price" property="advicePrice" jdbcType="DECIMAL"/>

<result column="pic\_id" property="picId" jdbcType="INTEGER"/>

<result column="model\_id" property="modelId" jdbcType="INTEGER"/>

<result column="product\_desc" property="productDesc" jdbcType="VARCHAR"/>

<result column="sys\_code" property="sysCode" jdbcType="VARCHAR"/>

<result column="creator" property="creator" jdbcType="VARCHAR"/>

<result column="gmt\_create" property="gmtCreate" jdbcType="TIMESTAMP"/>

<result column="modifier" property="modifier" jdbcType="VARCHAR"/>

<result column="gmt\_modified" property="gmtModified" jdbcType="TIMESTAMP"/>

<result column="is\_deleted" property="isDeleted" jdbcType="INTEGER"/>

<result column="pic\_ids" property="picIds" jdbcType="VARCHAR"/>

<result column="material\_pic\_ids" property="materialPicIds" jdbcType="VARCHAR"/>

<result column="house\_type\_values" property="houseTypeValues" jdbcType="VARCHAR"/>

<result column="product\_type\_value" property="productTypeValue" jdbcType="VARCHAR"/>

<result column="u3dmodel\_Id" property="u3dmodelId" jdbcType="VARCHAR"/>

<result column="merge\_product\_ids" property="mergeProductIds" jdbcType="VARCHAR"/>

<result column="putaway\_modified" property="putawayModified" jdbcType="TIMESTAMP"/>

<result column="date\_att2" property="dateAtt2" jdbcType="TIMESTAMP"/>

<result column="product\_small\_type\_value" property="productSmallTypeValue" jdbcType="INTEGER"/>

<result column="putaway\_state" property="putawayState" jdbcType="INTEGER"/>

<result column="num\_att3" property="numAtt3" jdbcType="DECIMAL"/>

<result column="num\_att4" property="numAtt4" jdbcType="DECIMAL"/>

<result column="remark" property="remark" jdbcType="VARCHAR"/>

<result column="ipad\_u3dmodel\_id" property="ipadU3dmodelId" jdbcType="INTEGER"/>

<result column="ios\_u3dmodel\_id" property="iosU3dmodelId" jdbcType="INTEGER"/>

<result column="android\_u3dmodel\_id" property="androidU3dmodelId" jdbcType="INTEGER"/>

<result column="macbook\_u3dmodel\_id" property="macbookU3dmodelId" jdbcType="INTEGER"/>

<result column="windows\_u3dmodel\_id" property="windowsU3dmodelId" jdbcType="INTEGER"/>

<result column="web\_u3dmodel\_id" property="webU3dmodelId" jdbcType="INTEGER"/>

<result column="decoration\_id" property="decorationId" jdbcType="VARCHAR"/>

<result column="designer\_id" property="designerId" jdbcType="INTEGER"/>

<result column="modeling\_id" property="modelingId" jdbcType="INTEGER"/>

<result column="rendering\_id" property="renderingId" jdbcType="INTEGER"/>

<result column="product\_type\_mark" property="productTypeMark" jdbcType="VARCHAR"/>

<result column="product\_small\_type\_mark" property="productSmallTypeMark" jdbcType="VARCHAR"/>

<result column="original\_product\_code" property="originalProductCode" jdbcType="VARCHAR"/>

<result column="spaceComon\_id" property="spacecomonId" jdbcType="INTEGER"/>

<result column="product\_short\_code" property="productShortCode" jdbcType="VARCHAR"/>

<result column="bm\_ids" property="bmIds" jdbcType="VARCHAR"/>

<result column="code\_number" property="codeNumber" jdbcType="INTEGER"/>

<result column="parent\_id" property="parentId" jdbcType="INTEGER"/>

<result column="designTemplet\_id" property="designtempletId" jdbcType="INTEGER"/>

<result column="material\_file\_id" property="materialFileId" jdbcType="INTEGER"/>

<result column="material3d\_pic\_ids" property="material3dPicIds" jdbcType="VARCHAR"/>

<result column="sync\_status" property="syncStatus" jdbcType="VARCHAR"/>

<result column="code\_is\_employ" property="codeIsEmploy" jdbcType="INTEGER"/>

<result column="code\_batch" property="codeBatch" jdbcType="VARCHAR"/>

<result column="product\_model\_number" property="productModelNumber" jdbcType="VARCHAR"/>

<result column="full\_pave\_length" property="fullPaveLength" jdbcType="VARCHAR"/>

<result column="test\_modified" property="testModified" jdbcType="TIMESTAMP"/>

<result column="sale\_price\_value" property="salePriceValue" jdbcType="INTEGER"/>

<result column="min\_height" property="minHeight" jdbcType="VARCHAR"/>

<result column="product\_material\_desc" property="productMaterialDesc" jdbcType="VARCHAR"/>

<result column="colors\_long\_code" property="colorsLongCode" jdbcType="VARCHAR"/>

<result column="config\_id" property="configId" jdbcType="INTEGER"/>

<result column="fbx\_file\_id" property="fbxFileId" jdbcType="INTEGER"/>

<result column="fbx\_texture" property="fbxTexture" jdbcType="VARCHAR"/>

<result column="fbx\_state" property="fbxState" jdbcType="INTEGER"/>

<result column="product\_style\_id\_info" property="productStyleIdInfo" jdbcType="VARCHAR"/>

<result column="style\_id" property="styleId" jdbcType="BIGINT"/>

<result column="region\_mark" property="regionMark" jdbcType="VARCHAR"/>

<result column="measure\_code" property="measureCode" jdbcType="VARCHAR"/>

<result column="wall\_type" property="wallType" jdbcType="VARCHAR"/>

<result column="product\_smallpox\_identify" property="productSmallpoxIdentify" jdbcType="INTEGER"/>

<result column="product\_ground\_identify" property="productGroundIdentify" jdbcType="INTEGER"/>

<result column="series\_id" property="seriesId" jdbcType="INTEGER"/>

<result column="spelling\_flower\_file\_id" property="spellingFlowerFileId" jdbcType="INTEGER"/>

<result column="app\_pic\_texture\_ids" property="appPicTextureIds" jdbcType="VARCHAR"/>

<result column="product\_smallpox\_identify\_str" property="productSmallpoxIdentifyStr" jdbcType="VARCHAR"/>

<result column="is\_created\_texture" property="isCreatedTexture" jdbcType="INTEGER"/>

<result column="is\_complex\_parquet" property="isComplexParquet" jdbcType="INTEGER"/>

</resultMap>

<!-- \*\*查询定义\*\* -->

<!-- 根据code查询 -->

<select id="getProductByCode" resultMap="AllResultMap">

select

<include refid="Base\_Column\_List"/>

from base\_product where product\_code = #{code,jdbcType=VARCHAR}

</select>

<!-- 按主键查询-->

<select id="getProductInfoById" resultType="com.sandu.api.product.model.Product">

select

<include refid="Base\_Column\_List"/>

from base\_product

where id = #{id,jdbcType=INTEGER} and is\_deleted = 0

</select>

<!-- 按分类主键批量查询产品 -->

<select id="getProductIdsByCategoryIds" resultType="java.lang.Integer">

select

bp.id, bp.product\_name, bp.product\_code, bp.brand\_id, bp.pic\_id

from base\_product bp

left join product\_category\_rel pcr

on bp.id = pcr.product\_id

WHERE pcr.category\_id in

<foreach collection="categoryIds" item="item" open="(" separator="," close=")">

#{item}

</foreach>

</select>

<!-- 根据品牌ID查询产品 -->

<select id="getProductByBrandId" resultMap="AllResultMap">

select

bp.id, bp.product\_name, bp.product\_code, bp.brand\_id, bp.pic\_id

from base\_product WHERE brand\_id = #{brandId}

</select>

<!-- 根据类目ID查询产品 -->

<select id="getProductByCategoryId" resultMap="AllResultMap">

select

bp.id, bp.product\_name, bp.product\_code, bp.brand\_id, bp.pic\_id

from base\_product bp

left join product\_category\_rel pcr

on bp.id = pcr.product\_id

where pcr.category\_id = #{categoryId}

</select>

<!-- 根据产品组ID查询产品 -->

<select id="getProductByGroupProductId" resultMap="AllResultMap">

select

bp.id, bp.product\_name, bp.product\_code, bp.brand\_id, bp.pic\_id

FROM

base\_product bp

LEFT JOIN group\_product\_details gpd ON gpd.product\_id = bp.id

WHERE

gpd.group\_id = #{groupProductId}

</select>

<!--查询产品列表(商家后台) -->

<select id="queryProducts" parameterType="com.sandu.api.product.model.po.ProductQueryPO"

resultType="com.sandu.api.product.model.bo.ProductListBO">

select

bp.id,bp.product\_code,bp.product\_name,bp.brand\_id,bp.pic\_id,bp.windows\_u3dmodel\_id as model\_id,

bp.gmt\_create,bp.material\_pic\_ids as textureId, bp.product\_small\_type\_mark as smallType,bp.product\_type\_mark as

type, bp.product\_batch\_type,

bp.secrecy\_flag as secrecy,

bp.product\_model\_number as modelNumber,

bp.is\_created\_texture as isCreatedTexture

from base\_product bp FORCE INDEX (idx\_pic)

where product\_batch\_type in (1,2)

and bp.is\_deleted = 0

<if test="brandIds!=null and brandIds.size > 0">

and bp.brand\_id in

<foreach collection="brandIds" open="(" separator="," close=")" item="bid">

#{bid}

</foreach>

</if>

<if test="companyId != null">

and company\_id = #{companyId}

</if>

<if test="platformId!= null">

<choose>

<when test="platformId != 0">

AND EXISTS

(SELECT

1

FROM

platform2b\_product\_rel p2b

WHERE bp.id = p2b.product\_id

AND p2b.allot\_state = 1

AND p2b.putaway\_state = 1

AND p2b.is\_deleted = 0

AND p2b.platform\_id = #{platformId}

UNION

ALL

SELECT

1

FROM

platform2c\_product\_rel p2c

WHERE bp.id = p2c.product\_id

AND p2c.allot\_state = 1

AND p2c.putaway\_state = 1

AND p2c.is\_deleted = 0

AND p2c.platform\_id = #{platformId})

</when>

<otherwise>

AND NOT EXISTS

(

SELECT

1

FROM

platform2b\_product\_rel p2b

WHERE bp.id = p2b.product\_id

AND p2b.allot\_state = 1

AND p2b.putaway\_state = 1

AND p2b.is\_deleted = 0

UNION

ALL

SELECT

1

FROM

platform2c\_product\_rel p2c

WHERE bp.id = p2c.product\_id

AND p2c.allot\_state = 1

AND p2c.putaway\_state = 1

AND p2c.is\_deleted = 0

)

</otherwise>

</choose>

</if>

<if test="modelNumber != null and modelNumber != ''">

and bp.product\_model\_number like concat(#{modelNumber},'%')

</if>

<if test="productType != null and productType != ''">

and product\_small\_type\_mark = #{productType}

</if>

<if test="productCode != null and productCode!=''">

and bp.product\_code like concat(#{productCode},'%')

</if>

<if test="productName != null and productName!=''">

and bp.product\_name like concat(#{productName},'%')

</if>

<if test="allotState != null and allotState == 'nonAllot'">

and (( p2b.id is null and p2c.id is null) or (p2c.allot\_State = 0 and p2b.allot\_State = 0))

</if>

<if test="allotState != null and allotState == 'channel'">

and p2b.allot\_State = 1 and p2b.platform\_id = #{platform2bId}

</if>

<if test="allotState != null and allotState == 'online'">

and p2c.allot\_State = 1 and p2c.platform\_id = #{platform2cId}

</if>

<if test="secrecy != null and secrecy != -1">

<if test="secrecy == 1">

and bp.secrecy\_flag = #{secrecy}

</if>

<if test="secrecy == 0">

and (bp.secrecy\_flag = #{secrecy} or bp.secrecy\_flag is null)

</if>

</if>

and bp.id &lt;= (

select distinct bp.id

from base\_product bp FORCE INDEX (idx\_pic)

where product\_batch\_type in (1,2)

and bp.is\_deleted = 0

<if test="brandIds!=null and brandIds.size > 0">

and bp.brand\_id in

<foreach collection="brandIds" open="(" separator="," close=")" item="bid">

#{bid}

</foreach>

</if>

<if test="companyId != null">

and company\_id = #{companyId}

</if>

<if test="platformId!= null">

<choose>

<when test="platformId != 0">

AND EXISTS

(SELECT

1

FROM

platform2b\_product\_rel p2b

WHERE bp.id = p2b.product\_id

AND p2b.allot\_state = 1

AND p2b.putaway\_state = 1

AND p2b.is\_deleted = 0

AND p2b.platform\_id = #{platformId}

UNION

ALL

SELECT

1

FROM

platform2c\_product\_rel p2c

WHERE bp.id = p2c.product\_id

AND p2c.allot\_state = 1

AND p2c.putaway\_state = 1

AND p2c.is\_deleted = 0

AND p2c.platform\_id = #{platformId})

</when>

<otherwise>

AND NOT EXISTS

(

SELECT

1

FROM

platform2b\_product\_rel p2b

WHERE bp.id = p2b.product\_id

AND p2b.allot\_state = 1

AND p2b.putaway\_state = 1

AND p2b.is\_deleted = 0

UNION

ALL

SELECT

1

FROM

platform2c\_product\_rel p2c

WHERE bp.id = p2c.product\_id

AND p2c.allot\_state = 1

AND p2c.putaway\_state = 1

AND p2c.is\_deleted = 0

)

</otherwise>

</choose>

</if>

<if test="modelNumber != null and modelNumber != ''">

and bp.product\_model\_number like concat(#{modelNumber},'%')

</if>

<if test="productType != null and productType != ''">

and product\_small\_type\_mark = #{productType}

</if>

<if test="productCode != null and productCode!=''">

and bp.product\_code like concat(#{productCode},'%')

</if>

<if test="productName != null and productName!=''">

and bp.product\_name like concat(#{productName},'%')

</if>

<if test="allotState != null and allotState == 'nonAllot'">

and (( p2b.id is null and p2c.id is null) or (p2c.allot\_State = 0 and p2b.allot\_State = 0))

</if>

<if test="allotState != null and allotState == 'channel'">

and p2b.allot\_State = 1 and p2b.platform\_id = #{platform2bId}

</if>

<if test="allotState != null and allotState == 'online'">

and p2c.allot\_State = 1 and p2c.platform\_id = #{platform2cId}

</if>

<if test="secrecy != null and secrecy != -1">

<if test="secrecy == 1">

and bp.secrecy\_flag = #{secrecy}

</if>

<if test="secrecy == 0">

and (bp.secrecy\_flag = #{secrecy} or bp.secrecy\_flag is null)

</if>

</if>

order by bp.id desc

limit #{offsetNum},1

)

group by bp.id

<if test="orderField!=null and orderField!=''">

order by bp.${orderField}

<if test="orderMethod!=null and orderMethod!=''">

${orderMethod}

</if>

</if>

limit #{limit}

</select>

<select id="queryProductsCount" parameterType="com.sandu.api.product.model.po.ProductQueryPO"

resultType="java.lang.Integer">

select

DISTINCT bp.id

from base\_product bp FORCE INDEX (idx\_pic)

where product\_batch\_type in (1,2)

and bp.is\_deleted = 0

<if test="brandIds!=null and brandIds.size > 0">

and bp.brand\_id in

<foreach collection="brandIds" open="(" separator="," close=")" item="bid">

#{bid}

</foreach>

</if>

<if test="productType != null and productType != ''">

and product\_small\_type\_mark = #{productType}

</if>

<if test="modelNumber != null and modelNumber != ''">

and bp.product\_model\_number like concat(#{modelNumber},'%')

</if>

<if test="companyId != null">

and company\_id = #{companyId}

</if>

<if test="platformId != null">

<choose>

<when test="platformId != 0">

AND EXISTS

(SELECT

1

FROM

platform2b\_product\_rel p2b

WHERE bp.id = p2b.product\_id

AND p2b.allot\_state = 1

AND p2b.putaway\_state = 1

AND p2b.is\_deleted = 0

AND p2b.platform\_id = #{platformId}

UNION

ALL

SELECT

1

FROM

platform2c\_product\_rel p2c

WHERE bp.id = p2c.product\_id

AND p2c.allot\_state = 1

AND p2c.putaway\_state = 1

AND p2c.is\_deleted = 0

AND p2c.platform\_id = #{platformId})

</when>

<otherwise>

AND NOT EXISTS

(

SELECT

1

FROM

platform2b\_product\_rel p2b

WHERE bp.id = p2b.product\_id

AND p2b.allot\_state = 1

AND p2b.putaway\_state = 1

AND p2b.is\_deleted = 0

UNION

ALL

SELECT

1

FROM

platform2c\_product\_rel p2c

WHERE bp.id = p2c.product\_id

AND p2c.allot\_state = 1

AND p2c.putaway\_state = 1

AND p2c.is\_deleted = 0

)

</otherwise>

</choose>

</if>

<if test="productCode != null and productCode!=''">

and bp.product\_code like concat(#{productCode},'%')

</if>

<if test="productName != null and productName!=''">

and bp.product\_name like concat(#{productName},'%')

</if>

<if test="allotState != null and allotState == 'nonAllot'">

and (( p2b.id is null and p2c.id is null) or (p2c.allot\_State = 0 and p2b.allot\_State = 0))

</if>

<if test="allotState != null and allotState == 'channel'">

and p2b.allot\_State = 1 and p2b.platform\_id = #{platform2bId}

</if>

<if test="allotState != null and allotState == 'online'">

and p2c.allot\_State = 1 and p2c.platform\_id = #{platform2cId}

</if>

<if test="secrecy != null and secrecy != -1">

<if test="secrecy == 1">

and bp.secrecy\_flag = #{secrecy}

</if>

<if test="secrecy == 0">

and (bp.secrecy\_flag = #{secrecy} or bp.secrecy\_flag is null)

</if>

</if>

</select>

<!-- 获取产品已上架列表 -->

<select id="queryUpPlatForm" resultType="java.util.Map">

SELECT

t.productId,

GROUP\_CONCAT(t.platformId) AS platformId

FROM

(SELECT

product\_id AS productId,

GROUP\_CONCAT(p2b.platform\_id) AS platformId

FROM

platform2b\_product\_rel p2b

WHERE p2b.allot\_state = 1

AND p2b.putaway\_state = 1

AND p2b.is\_deleted = 0

<if test="productIds!=null and productIds.size > 0">

and p2b.product\_id in

<foreach collection="productIds" item="item" open="(" close=")" separator=",">

#{item}

</foreach>

</if>

GROUP BY product\_id

UNION

SELECT

product\_id AS productId,

GROUP\_CONCAT(p2c.platform\_id) AS platformId

FROM

platform2c\_product\_rel p2c

WHERE p2c.allot\_state = 1

AND p2c.putaway\_state = 1

AND p2c.`is\_deleted` = 0

<if test="productIds!=null and productIds.size > 0">

and p2c.product\_id in

<foreach collection="productIds" item="item" open="(" close=")" separator=",">

#{item}

</foreach>

</if>

GROUP BY product\_id) t

GROUP BY t.productId

ORDER BY t.productId

</select>

<!--查询产品列表(渠道)-->

<select id="query2bProducts" parameterType="com.sandu.api.product.model.po.ProductQueryPO"

resultType="com.sandu.api.product.model.bo.ProductListBO">

SELECT

bp.id,

bp.product\_code,

bp.product\_name,

bp.brand\_id,

bp.pic\_id,

bp.windows\_u3dmodel\_id AS model\_id,

bp.gmt\_create,

bp.material\_pic\_ids AS textureId,

bp.product\_small\_type\_mark AS smallType,

bp.product\_type\_mark AS type,

bp.secrecy\_flag AS secrecy,

bp.product\_model\_number AS modelNumber,

max(p2b.putaway\_state) AS putAwayStatus2b,

max(p2b.pic\_id) AS pic2b\_id,

max(p2b.gmt\_create) AS gmtCreate2b

FROM base\_product bp LEFT JOIN platform2b\_product\_rel p2b ON bp.id = p2b.product\_id

where bp.is\_deleted = 0 and bp.brand\_id != -1

and p2b.allot\_State = 1

<if test="brandIds!=null and brandIds.size > 0">

and bp.brand\_id in

<foreach collection="brandIds" open="(" separator="," close=")" item="bid">

#{bid}

</foreach>

</if>

<if test="productType != null and productType != ''">

and (product\_type\_mark = #{productType} or product\_small\_type\_mark = #{productType})

</if>

<if test="modelNumber != null and modelNumber != ''">

and bp.product\_model\_number like concat('%',#{modelNumber},'%')

</if>

<if test="companyId != null">

and company\_id = #{companyId}

</if>

<if test="productCode != null and productCode !='' ">

and bp.product\_code like concat('%',#{productCode},'%')

</if>

<if test="productName != null and productName !=''">

and bp.product\_name like concat('%',#{productName},'%')

</if>

<if test="putawayState!=null and putawayState >= 0">

and p2b.putaway\_state = #{putawayState} and p2b.platform\_id = #{platform2bId}

</if>

<if test="secrecy != null and secrecy != -1">

<if test="secrecy == 1">

and bp.secrecy\_flag = #{secrecy}

</if>

<if test="secrecy == 0">

and (bp.secrecy\_flag = #{secrecy} or bp.secrecy\_flag is null)

</if>

</if>

GROUP BY bp.id

<if test="orderField!=null and orderField!=''">

order by bp.${orderField}

<if test="orderMethod!=null and orderMethod!=''">

${orderMethod}

</if>

</if>

limit #{page},#{limit}

</select>

<select id="query2bProductsCount" parameterType="com.sandu.api.product.model.po.ProductQueryPO"

resultType="java.lang.Integer">

select count(1) from (

SELECT

distinct bp.id

FROM

base\_product bp

LEFT JOIN platform2b\_product\_rel p2b FORCE INDEX (idx\_product\_id) ON bp.id = p2b.product\_id

where bp.is\_deleted = 0 and bp.brand\_id != -1

and p2b.allot\_State = 1

<if test="brandIds!=null and brandIds.size > 0">

and bp.brand\_id in

<foreach collection="brandIds" open="(" separator="," close=")" item="bid">

#{bid}

</foreach>

</if>

<if test="modelNumber != null and modelNumber != ''">

and bp.product\_model\_number like concat('%',#{modelNumber},'%')

</if>

<if test="productType != null and productType != ''">

and (product\_type\_mark = #{productType} or product\_small\_type\_mark = #{productType})

</if>

<if test="companyId != null">

and company\_id = #{companyId}

</if>

<if test="productCode != null and productCode !='' ">

and bp.product\_code like concat('%',#{productCode},'%')

</if>

<if test="productName != null and productName !=''">

and bp.product\_name like concat('%',#{productName},'%')

</if>

<if test="putawayState!=null and putawayState >= 0">

and p2b.putaway\_state = #{putawayState} and p2b.platform\_id = #{platform2bId}

</if>

<if test="secrecy != null and secrecy != -1">

<if test="secrecy == 1">

and bp.secrecy\_flag = #{secrecy}

</if>

<if test="secrecy == 0">

and (bp.secrecy\_flag = #{secrecy} or bp.secrecy\_flag is null)

</if>

</if>

) as t

</select>

<!--查询产品列表(线上)-->

<select id="query2cProducts" parameterType="com.sandu.api.product.model.po.ProductQueryPO"

resultType="com.sandu.api.product.model.bo.ProductListBO">

select

bp.id,bp.product\_code,bp.product\_name,bp.brand\_id,bp.pic\_id,bp.windows\_u3dmodel\_id as model\_id,

bp.gmt\_create,bp.material\_pic\_ids as textureId, bp.product\_small\_type\_mark as smallType,bp.product\_type\_mark as

type,

p2c.allot\_State as status2c,

p2c.gmt\_create as gmtCreate2c,

p2c.pic\_id as pic2c\_id,

bp.secrecy\_flag as secrecy,

bp.product\_model\_number as modelNumber,

GROUP\_CONCAT(p2c.putaway\_state) as putStatus,

GROUP\_CONCAT(p2c.platform\_id) as platformIds

from base\_product bp

left join platform2c\_product\_rel p2c on bp.id = p2c.product\_id

where bp.is\_deleted = 0 and bp.brand\_id != -1

and p2c.allot\_State = 1

<if test="brandIds!=null and brandIds.size > 0">

and bp.brand\_id in

<foreach collection="brandIds" open="(" separator="," close=")" item="bid">

#{bid}

</foreach>

</if>

<if test="modelNumber != null and modelNumber != ''">

and bp.product\_model\_number like concat('%',#{modelNumber},'%')

</if>

<if test="productType != null and productType != ''">

and (product\_type\_mark = #{productType} or product\_small\_type\_mark = #{productType})

</if>

<if test="companyId != null">

and company\_id = #{companyId}

</if>

<if test="productCode != null and productCode !=''">

and bp.product\_code like concat('%',#{productCode},'%')

</if>

<if test="productName != null and productName !=''">

and bp.product\_name like concat('%',#{productName},'%')

</if>

<if test="platformId != null and platformId>0">

and p2c.putaway\_state = 1 and p2c.platform\_id = #{platformId}

</if>

<if test="platformId != null and platformId == -1">

and (p2c.putaway\_state = 0 or p2c.putaway\_state is null)

</if>

<if test="secrecy != null and secrecy != -1">

<if test="secrecy == 1">

and bp.secrecy\_flag = #{secrecy}

</if>

<if test="secrecy == 0">

and (bp.secrecy\_flag = #{secrecy} or bp.secrecy\_flag is null)

</if>

</if>

GROUP BY bp.id

<if test="orderField!=null and orderField!=''">

order by bp.${orderField}

<if test="orderMethod!=null and orderMethod!=''">

${orderMethod}

</if>

</if>

limit #{page},#{limit}

</select>

<select id="query2cProductsCount" parameterType="com.sandu.api.product.model.po.ProductQueryPO"

resultType="java.lang.Integer">

select count(1) from (

select

DISTINCT bp.id

from base\_product bp

left join platform2c\_product\_rel p2c FORCE INDEX (idx\_product\_id) on bp.id = p2c.product\_id

where bp.is\_deleted = 0 and bp.brand\_id != -1

and p2c.allot\_State = 1

<if test="brandIds!=null and brandIds.size > 0">

and bp.brand\_id in

<foreach collection="brandIds" open="(" separator="," close=")" item="bid">

#{bid}

</foreach>

</if>

<if test="modelNumber != null and modelNumber != ''">

and bp.product\_model\_number like concat('%',#{modelNumber},'%')

</if>

<if test="companyId != null">

and company\_id = #{companyId}

</if>

<if test="productType != null and productType != ''">

and (product\_type\_mark = #{productType} or product\_small\_type\_mark = #{productType})

</if>

<if test="productCode != null and productCode !=''">

and bp.product\_code like concat('%',#{productCode},'%')

</if>

<if test="productName != null and productName !=''">

and bp.product\_name like concat('%',#{productName},'%')

</if>

<if test="platformId != null and platformId>0">

and p2c.putaway\_state = 1 and p2c.platform\_id = #{platformId}

</if>

<if test="platformId != null and platformId == -1">

and (p2c.putaway\_state = 0 or p2c.putaway\_state is null)

</if>

<if test="secrecy != null and secrecy != -1">

<if test="secrecy == 1">

and bp.secrecy\_flag = #{secrecy}

</if>

<if test="secrecy == 0">

and (bp.secrecy\_flag = #{secrecy} or bp.secrecy\_flag is null)

</if>

</if>

) as t

</select>

<!-- 设计方案产品列表(根据产品ID集合查询) -->

<select id="querySolutionProducts" parameterType="com.sandu.api.product.model.po.ProductQueryPO"

resultType="com.sandu.api.product.model.bo.ProductListBO">

select

bp.id,bp.product\_code,bp.product\_name,bp.brand\_id,bp.pic\_id,bp.windows\_u3dmodel\_id as model\_id,

bp.gmt\_create,bp.material\_pic\_ids as textureId, bp.product\_small\_type\_mark as smallType,bp.product\_type\_mark as

type,

p2b.putaway\_state as putAwayStatus2b,

p2b.gmt\_create as gmtCreate2b,

p2c.gmt\_create as gmtCreate2c,

bp.secrecy\_flag as secrecy,

bp.product\_model\_number as modelNumber

from base\_product bp

left join platform2b\_product\_rel p2b on bp.id = p2b.product\_id

left join platform2c\_product\_rel p2c on bp.id = p2c.product\_id

where bp.is\_deleted = 0 and bp.product\_code not like 'baimo\_%'

<if test="productIds!=null and productIds.size > 0">

and bp.id in

<foreach collection="productIds" item="item" open="(" close=")" separator=",">

#{item}

</foreach>

</if>

<if test="productType != null and productType != ''">

and (bp.product\_type\_mark = #{productType} or bp.product\_small\_type\_mark = #{productType})

</if>

<if test="categoryCode != null and categoryCode != ''">

and (product\_type\_mark = #{categoryCode} or product\_small\_type\_mark = #{categoryCode})

</if>

<if test="productCode != null and productCode!=''">

and bp.product\_code like concat('%',#{productCode},'%')

</if>

<if test="productName != null and productName!=''">

and bp.product\_name like concat('%',#{productName},'%')

</if>

<if test="allotState != null and allotState == 'nonAllot'">

and (( p2b.id is null and p2c.id is null) or (p2c.allot\_State = 0 and p2b.allot\_State = 0))

</if>

<if test="allotState != null and allotState == 'channel'">

and p2b.allot\_State = 1 and p2b.platform\_id = #{platform2bId}

</if>

<if test="allotState != null and allotState == 'online'">

and p2c.allot\_State = 1 and p2c.platform\_id = #{platform2cId}

</if>

<if test="secrecy != null and secrecy == 0">

and ( bp.secrecy\_flag = #{secrecy} or bp.secrecy\_flag is null)

</if>

<if test="secrecy != null and secrecy == 1">

and bp.secrecy\_flag = #{secrecy}

</if>

group by bp.id

<if test="orderField!=null and orderField!=''">

order by bp.${orderField}

<if test="orderMethod!=null and orderMethod!=''">

${orderMethod}

</if>

</if>

</select>

<update id="deleteBussinessyProduct">

update base\_product SET

is\_deleted = 1,

gmt\_modified = now()

where id in

<foreach collection="list" open="(" separator="," close=")" item="id">

#{id}

</foreach>

</update>

<select id="getListByIds" resultMap="AllResultMap">

select id,pro\_style\_id,sale\_price,advice\_price,pic\_id,pic\_ids,product\_desc,product\_style\_id\_info from

base\_product

where id in

<foreach collection="list" item="id" open="(" close=")" separator=",">

#{id}

</foreach>

</select>

<select id="getListByModelIds" resultMap="AllResultMap">

select id,product\_name

from base\_product

where model\_id in

<foreach collection="list" open="(" separator="," close=")" item="modelId">

#{modelId}

</foreach>

</select>

<select id="getProductByModelIds" resultType="com.sandu.api.product.model.Product">

select id,product\_code,windows\_u3dmodel\_id as modelId

from base\_product

where 1 =1

and base\_product.windows\_u3dmodel\_id in

<foreach collection="modelIds" item="id" separator="," close=")" open="(">

#{id}

</foreach>

</select>

<select id="getProductByIds" resultMap="AllResultMap">

select id,product\_code,product\_name,windows\_u3dmodel\_id as model\_id , pic\_id

from base\_product

where id in

<foreach collection="list" separator="," close=")" open="(" item="id">

#{id}

</foreach>

</select>

<select id="mapProductId2PutStatus" resultType="com.sandu.api.platform.model.PlatformProductRel">

select product\_id,group\_concat(putaway\_state) as att1 from platform2c\_product\_rel

where product\_id in

<foreach collection="list" item="id" open="(" close=")" separator=",">

#{id}

</foreach>

GROUP BY product\_id

</select>

<select id="queryProductsInEditor" parameterType="com.sandu.api.product.input.EditorProductQuery"

resultType="com.sandu.api.product.model.bo.EditorProductListBO">

SELECT DISTINCT bp.id,bp.product\_name,bp.product\_code,rm.id as modelId,

bp.product\_model\_number as modelNumber,bp.pic\_id,

rm.model\_code,rm.category\_names

from base\_product bp

INNER JOIN res\_model rm on bp.windows\_u3dmodel\_id = rm.id

LEFT join product\_category\_rel pcr on bp.id = pcr.product\_id

where 1 =1 and bp.is\_deleted = 0

and bp.windows\_u3dmodel\_id is not null and windows\_u3dmodel\_id != 0

<if test="companyId != null">

and bp.company\_id = #{companyId}

</if>

<if test="brandIds!=null and brandIds.size > 0">

and bp.brand\_id in

<foreach collection="brandIds" open="(" separator="," close=")" item="bid">

#{bid}

</foreach>

</if>

<if test="categoryIds != null and categoryIds.size > 0">

and pcr.category\_id in

<foreach collection="categoryIds" item="item" open="(" close=")" separator=",">

#{item}

</foreach>

</if>

<if test="productCode != null and productCode!=''">

and bp.product\_code like concat('%',#{productCode},'%')

</if>

<if test="productName != null and productName!=''">

and bp.product\_name like concat('%',#{productName},'%')

</if>

<if test="modelCode != null and modelCode!=''">

and bp.model\_code like concat('%',#{modelCode},'%')

</if>

<if test="modelNumber != null and modelNumber!=''">

and bp.product\_model\_number like concat('%',#{modelNumber},'%')

</if>

</select>

<select id="getNotBeSInitProductIds" resultType="java.lang.Integer">

select id from base\_product

where is\_deleted = 0 and brand\_id != -1

and num\_att3 is null and putaway\_state = 3

</select>

<update id="updateProductInitStatus" parameterType="java.lang.Integer">

update base\_product set num\_att3 = 1 where id in

<foreach collection="list" item="id" separator="," close=")" open="(">

#{id}

</foreach>

</update>

<update id="synProductCompanyInfoWithBrandId">

UPDATE

base\_product bp,

base\_brand bb

SET

bp.company\_id = IFNULL(bb.company\_id, 0),

bp.remark = '根据品牌同步产品公司ID'

WHERE bp.brand\_id = bb.id

and bp.id in

<foreach collection="needSynProductIds" item="id" separator="," open="(" close=")">

#{id}

</foreach>

</update>

<update id="synProductPutawayState">

update base\_product set putaway\_state = 3,remark='商家后台同步产品状态(1->3)' where

id in

<foreach collection="needSynProductIds" item="id" separator="," open="(" close=")">

#{id}

</foreach>

</update>

<update id="updateProductTypeInfo" parameterType="com.sandu.api.product.model.Product">

<foreach collection="list" item="item" separator=";" close=";">

update base\_product set product\_type\_mark = #{item.productTypeMark},product\_small\_type\_mark =

#{item.productSmallTypeMark}

where id = #{item.id}

</foreach>

</update>

<select id="getAutoSynProductIds" resultType="java.lang.Integer">

select id from base\_product

where is\_deleted = 0 and brand\_id != -1

and num\_att3 is null and putaway\_state = 3 and pic\_id != 0 and pic\_id is not null

limit 3000

</select>

<select id="getSoftHardProduct" resultType="java.lang.String">

select GROUP\_CONCAT(id) from base\_product where

product\_batch\_id in

<foreach collection="productIds" item="item" open="(" close=")" separator=",">

#{item}

</foreach>

AND id !=product\_batch\_id

</select>

<select id="getBePutAwayProduct" resultType="java.lang.Integer">

select \* from base\_product where

id in

<foreach collection="list" item="item" open="(" close=")" separator=",">

#{item}

</foreach>

<!-- and putaway\_state = 3 and (windows\_u3dmodel\_id != -1 or material\_pic\_ids != '-1') -->

</select>

<select id="getProductDeliveredInfo" resultType="java.lang.String">

SELECT group\_concat(bp.company\_id) from base\_product bp, base\_company bc

where bp.company\_id = bc.id and bp.origin\_id = #{productId} GROUP BY bp.company\_id

</select>

<select id="getMaxId" resultType="java.lang.String">

SELECT max(id)

FROM base\_product

</select>

<select id="getNeedBeSyncProductInfo" resultType="com.sandu.api.product.model.Product">

SELECT bp.id, bp.product\_name, sc.type as product\_type\_mark, sc.valuekey as product\_small\_type\_mark

FROM sys\_dictionary s ,sys\_dictionary sc , base\_product bp where s.valuekey = sc.type and s.value = bp.product\_type\_value

and s.type = 'productType' and sc.value = bp.product\_small\_type\_value and product\_type\_mark is null limit 1500

</select>

<select id="getProductNumberWordRecommends" resultType="java.lang.String">

select distinct product\_model\_number from base\_product where is\_deleted = 0

and company\_id = #{companyId}

<if test="wordStart !=null and wordStart != ''">

and product\_model\_number like Concat(#{wordStart},'%')

</if>

<if test="brandId != null ">

and brand\_id = #{brandId}

</if>

AND product\_model\_number IS NOT NULL AND product\_model\_number!=''

order by product\_model\_number

</select>

<!-- 获取商家后台创建并且有效的非白膜产品 -->

<select id="selectMerchantProductList" resultType="com.sandu.api.product.model.Product">

SELECT \* FROM base\_product a

WHERE a.`product\_code` LIKE 'P%'

AND a.`is\_deleted` = 0

AND a.`product\_code` NOT LIKE 'baimo%'

</select>

<select id="getNeedSynProductIds" resultType="java.lang.Integer">

select id from base\_product where putaway\_state = 1 and is\_deleted = 0

and (windows\_u3dmodel\_id != -1 or material\_pic\_ids != '-1') limit 500

</select>

<select id="getNeedUpdateProductIds" resultType="java.lang.Integer">

SELECT id from base\_product where company\_id =0 and brand\_id > 0 limit 500

</select>

<select id="listSecondProductId2ModelIdWithMainId" resultType="com.sandu.api.product.model.Product">

select id , windows\_u3dmodel\_id from base\_product where is\_deleted = 0 and product\_batch\_id = #{id}

</select>

<select id="getHardProductIds" resultType="java.lang.String">

select group\_concat(id) from base\_product where product\_batch\_id in

<foreach collection="ids" item="id" separator="," close=")" open="(">

#{id}

</foreach>

and product\_batch\_type != 1

</select>

<delete id="fixedProductToSku" statementType="CALLABLE">

{call resolve\_sandu\_product()}

</delete>

<delete id="hardDeleteProductByIds">

delete from base\_product where id in

<foreach collection="productIds" item="id" separator="," open="(" close=")">

#{id}

</foreach>

</delete>

</mapper>

package com.sandu.api.product.model.bo;

import lombok.Data;

import java.io.Serializable;

import java.util.Date;

import java.util.List;

import java.util.Objects;

/\*\*

\* @author Sandu

\*/

@Data

public class ProductListBO implements Serializable {

private Integer id;

/\*\*

\* 产品编码

\*/

private String productCode;

/\*\*

\* 产品名称

\*/

private String productName;

/\*\*

\* 产品缩略图ID

\*/

private Integer picId;

/\*\*

\* 产品缩略图路径

\*/

private String picPath;

/\*\*

\* 产品品牌名称

\*/

private String brandName;

/\*\*

\* 产品品牌主键

\*/

private Integer brandId;

/\*\*

\* 产品分类ID(最高级)

\*/

private Integer categoryId;

/\*\*

\* 产品分类信息(用于列表/详情页展示)

\*/

private String categoryNames;

/\*\*

\* 产品模型id

\*/

private Integer modelId;

/\*\*

\* 产品模型编码

\*/

private String modelCode;

/\*\*

\* 产品贴图编码

\*/

private String textureId;

/\*\*

\* 产品渠道分配状态

\*/

private Integer status2b;

/\*\*

\* 产品渠道分配时间

\*/

private Date gmtCreate2b;

/\*\*

\* 产品线上分配状态

\*/

private Integer status2c;

/\*\*

\* 产品线上分配时间

\*/

private Date gmtCreate2c;

/\*\*

\* 产品创建时间

\*/

private Date gmtCreate;

/\*\*

\* 产品公开状态：0不公开，1公开

\*/

private Integer secrecy;

/\*\*

\* 渠道产品上架状态

\*/

private Integer putAwayStatus2b;

/\*\*

\* 渠道产品上架状态

\*/

private Integer putAwayStatus2c;

/\*\*

\* 线上产品以上架平台名称

\*/

private List<String> putAwayPlatformNames;

/\*\*

\* 线上产品以上架平台名称

\*/

private List<Integer> putAwayPlatformIds;

/\*\*

\* 产品型号

\*/

private String modelNumber;

/\*\*

\* 产品所属公司

\*\*/

private String companyName;

/\*\*

\* 此产品分配的所有平台ID id,id,id字符串

\*/

private String platformIds;

/\*\*

\* 此产品分配的所有平台ID id,id,id字符串

\*/

private String platform2cIds;

/\*\*

\* 此产品在各平台的上架状态 1,0,1字符串

\*/

private String putStatus;

/\*\*

\* 渠道默认缩略图

\*/

private Integer pic2bId;

/\*\*

\* 线上默认缩略图

\*/

private Integer pic2cId;

/\*\*

\* 产品小类

\*/

private String smallType;

private String Type;

/\*\*

\* 产品类型(定制/软装)

\*/

private Integer productBatchType;

private Integer isCreatedTexture;

@Override

public String toString() {

return "ProductListBO{" +

"id=" + id +

", productCode='" + productCode + '\'' +

", productName='" + productName + '\'' +

", picId=" + picId +

", picPath='" + picPath + '\'' +

", brandName='" + brandName + '\'' +

", brandId=" + brandId +

", categoryId=" + categoryId +

", categoryNames='" + categoryNames + '\'' +

", modelId=" + modelId +

", modelCode='" + modelCode + '\'' +

", textureId='" + textureId + '\'' +

", status2b=" + status2b +

", gmtCreate2b=" + gmtCreate2b +

", status2c=" + status2c +

", gmtCreate2c=" + gmtCreate2c +

", gmtCreate=" + gmtCreate +

", secrecy=" + secrecy +

", putAwayStatus2b=" + putAwayStatus2b +

", putAwayStatus2c=" + putAwayStatus2c +

", putAwayPlatformNames=" + putAwayPlatformNames +

", putAwayPlatformIds=" + putAwayPlatformIds +

", modelNumber='" + modelNumber + '\'' +

", companyName='" + companyName + '\'' +

", platformIds='" + platformIds + '\'' +

", platform2cIds='" + platform2cIds + '\'' +

", putStatus='" + putStatus + '\'' +

", pic2bId=" + pic2bId +

", pic2cId=" + pic2cId +

", smallType='" + smallType + '\'' +

", Type='" + Type + '\'' +

", productBatchType=" + productBatchType +

", isCreatedTexture=" + isCreatedTexture +

'}';

}

@Override

public boolean equals(Object o) {

if (this == o) return true;

if (!(o instanceof ProductListBO)) return false;

ProductListBO that = (ProductListBO) o;

return Objects.equals(getId(), that.getId()) &&

Objects.equals(getProductCode(), that.getProductCode()) &&

Objects.equals(getProductName(), that.getProductName()) &&

Objects.equals(getPicId(), that.getPicId()) &&

Objects.equals(getPicPath(), that.getPicPath()) &&

Objects.equals(getBrandName(), that.getBrandName()) &&

Objects.equals(getBrandId(), that.getBrandId()) &&

Objects.equals(getCategoryId(), that.getCategoryId()) &&

Objects.equals(getCategoryNames(), that.getCategoryNames()) &&

Objects.equals(getModelId(), that.getModelId()) &&

Objects.equals(getModelCode(), that.getModelCode()) &&

Objects.equals(getTextureId(), that.getTextureId()) &&

Objects.equals(getStatus2b(), that.getStatus2b()) &&

Objects.equals(getGmtCreate2b(), that.getGmtCreate2b()) &&

Objects.equals(getStatus2c(), that.getStatus2c()) &&

Objects.equals(getGmtCreate2c(), that.getGmtCreate2c()) &&

Objects.equals(getGmtCreate(), that.getGmtCreate()) &&

Objects.equals(getSecrecy(), that.getSecrecy()) &&

Objects.equals(getPutAwayStatus2b(), that.getPutAwayStatus2b()) &&

Objects.equals(getPutAwayStatus2c(), that.getPutAwayStatus2c()) &&

Objects.equals(getPutAwayPlatformNames(), that.getPutAwayPlatformNames()) &&

Objects.equals(getPutAwayPlatformIds(), that.getPutAwayPlatformIds()) &&

Objects.equals(getModelNumber(), that.getModelNumber()) &&

Objects.equals(getCompanyName(), that.getCompanyName()) &&

Objects.equals(getPlatformIds(), that.getPlatformIds()) &&

Objects.equals(getPlatform2cIds(), that.getPlatform2cIds()) &&

Objects.equals(getPutStatus(), that.getPutStatus()) &&

Objects.equals(getPic2bId(), that.getPic2bId()) &&

Objects.equals(getPic2cId(), that.getPic2cId()) &&

Objects.equals(getSmallType(), that.getSmallType()) &&

Objects.equals(getType(), that.getType()) &&

Objects.equals(getProductBatchType(), that.getProductBatchType()) &&

Objects.equals(getIsCreatedTexture(), that.getIsCreatedTexture());

}

@Override

public int hashCode() {

return Objects.hash(getId(), getProductCode(), getProductName(), getPicId(), getPicPath(), getBrandName(), getBrandId(), getCategoryId(), getCategoryNames(), getModelId(), getModelCode(), getTextureId(), getStatus2b(), getGmtCreate2b(), getStatus2c(), getGmtCreate2c(), getGmtCreate(), getSecrecy(), getPutAwayStatus2b(), getPutAwayStatus2c(), getPutAwayPlatformNames(), getPutAwayPlatformIds(), getModelNumber(), getCompanyName(), getPlatformIds(), getPlatform2cIds(), getPutStatus(), getPic2bId(), getPic2cId(), getSmallType(), getType(), getProductBatchType(), getIsCreatedTexture());

}

public ProductListBO(Integer id, String productCode, String productName, Integer picId, String picPath, String brandName, Integer brandId, Integer categoryId, String categoryNames, Integer modelId, String modelCode, String textureId, Integer status2b, Date gmtCreate2b, Integer status2c, Date gmtCreate2c, Date gmtCreate, Integer secrecy, Integer putAwayStatus2b, Integer putAwayStatus2c, List<String> putAwayPlatformNames, List<Integer> putAwayPlatformIds, String modelNumber, String companyName, String platformIds, String platform2cIds, String putStatus, Integer pic2bId, Integer pic2cId, String smallType, String type, Integer productBatchType, Integer isCreatedTexture) {

this.id = id;

this.productCode = productCode;

this.productName = productName;

this.picId = picId;

this.picPath = picPath;

this.brandName = brandName;

this.brandId = brandId;

this.categoryId = categoryId;

this.categoryNames = categoryNames;

this.modelId = modelId;

this.modelCode = modelCode;

this.textureId = textureId;

this.status2b = status2b;

this.gmtCreate2b = gmtCreate2b;

this.status2c = status2c;

this.gmtCreate2c = gmtCreate2c;

this.gmtCreate = gmtCreate;

this.secrecy = secrecy;

this.putAwayStatus2b = putAwayStatus2b;

this.putAwayStatus2c = putAwayStatus2c;

this.putAwayPlatformNames = putAwayPlatformNames;

this.putAwayPlatformIds = putAwayPlatformIds;

this.modelNumber = modelNumber;

this.companyName = companyName;

this.platformIds = platformIds;

this.platform2cIds = platform2cIds;

this.putStatus = putStatus;

this.pic2bId = pic2bId;

this.pic2cId = pic2cId;

this.smallType = smallType;

Type = type;

this.productBatchType = productBatchType;

this.isCreatedTexture = isCreatedTexture;

}

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

public String getProductCode() {

return productCode;

}

public void setProductCode(String productCode) {

this.productCode = productCode;

}

public String getProductName() {

return productName;

}

public void setProductName(String productName) {

this.productName = productName;

}

public Integer getPicId() {

return picId;

}

public void setPicId(Integer picId) {

this.picId = picId;

}

public String getPicPath() {

return picPath;

}

public void setPicPath(String picPath) {

this.picPath = picPath;

}

public String getBrandName() {

return brandName;

}

public void setBrandName(String brandName) {

this.brandName = brandName;

}

public Integer getBrandId() {

return brandId;

}

public void setBrandId(Integer brandId) {

this.brandId = brandId;

}

public Integer getCategoryId() {

return categoryId;

}

public void setCategoryId(Integer categoryId) {

this.categoryId = categoryId;

}

public String getCategoryNames() {

return categoryNames;

}

public void setCategoryNames(String categoryNames) {

this.categoryNames = categoryNames;

}

public Integer getModelId() {

return modelId;

}

public void setModelId(Integer modelId) {

this.modelId = modelId;

}

public String getModelCode() {

return modelCode;

}

public void setModelCode(String modelCode) {

this.modelCode = modelCode;

}

public String getTextureId() {

return textureId;

}

public void setTextureId(String textureId) {

this.textureId = textureId;

}

public Integer getStatus2b() {

return status2b;

}

public void setStatus2b(Integer status2b) {

this.status2b = status2b;

}

public Date getGmtCreate2b() {

return gmtCreate2b;

}

public void setGmtCreate2b(Date gmtCreate2b) {

this.gmtCreate2b = gmtCreate2b;

}

public Integer getStatus2c() {

return status2c;

}

public void setStatus2c(Integer status2c) {

this.status2c = status2c;

}

public Date getGmtCreate2c() {

return gmtCreate2c;

}

public void setGmtCreate2c(Date gmtCreate2c) {

this.gmtCreate2c = gmtCreate2c;

}

public Date getGmtCreate() {

return gmtCreate;

}

public void setGmtCreate(Date gmtCreate) {

this.gmtCreate = gmtCreate;

}

public Integer getSecrecy() {

return secrecy;

}

public void setSecrecy(Integer secrecy) {

this.secrecy = secrecy;

}

public Integer getPutAwayStatus2b() {

return putAwayStatus2b;

}

public void setPutAwayStatus2b(Integer putAwayStatus2b) {

this.putAwayStatus2b = putAwayStatus2b;

}

public Integer getPutAwayStatus2c() {

return putAwayStatus2c;

}

public void setPutAwayStatus2c(Integer putAwayStatus2c) {

this.putAwayStatus2c = putAwayStatus2c;

}

public List<String> getPutAwayPlatformNames() {

return putAwayPlatformNames;

}

public void setPutAwayPlatformNames(List<String> putAwayPlatformNames) {

this.putAwayPlatformNames = putAwayPlatformNames;

}

public List<Integer> getPutAwayPlatformIds() {

return putAwayPlatformIds;

}

public void setPutAwayPlatformIds(List<Integer> putAwayPlatformIds) {

this.putAwayPlatformIds = putAwayPlatformIds;

}

public String getModelNumber() {

return modelNumber;

}

public void setModelNumber(String modelNumber) {

this.modelNumber = modelNumber;

}

public String getCompanyName() {

return companyName;

}

public void setCompanyName(String companyName) {

this.companyName = companyName;

}

public String getPlatformIds() {

return platformIds;

}

public void setPlatformIds(String platformIds) {

this.platformIds = platformIds;

}

public String getPlatform2cIds() {

return platform2cIds;

}

public void setPlatform2cIds(String platform2cIds) {

this.platform2cIds = platform2cIds;

}

public String getPutStatus() {

return putStatus;

}

public void setPutStatus(String putStatus) {

this.putStatus = putStatus;

}

public Integer getPic2bId() {

return pic2bId;

}

public void setPic2bId(Integer pic2bId) {

this.pic2bId = pic2bId;

}

public Integer getPic2cId() {

return pic2cId;

}

public void setPic2cId(Integer pic2cId) {

this.pic2cId = pic2cId;

}

public String getSmallType() {

return smallType;

}

public void setSmallType(String smallType) {

this.smallType = smallType;

}

public String getType() {

return Type;

}

public void setType(String type) {

Type = type;

}

public Integer getProductBatchType() {

return productBatchType;

}

public void setProductBatchType(Integer productBatchType) {

this.productBatchType = productBatchType;

}

public Integer getIsCreatedTexture() {

return isCreatedTexture;

}

public void setIsCreatedTexture(Integer isCreatedTexture) {

this.isCreatedTexture = isCreatedTexture;

}

}

//

// Source code recreated from a .class file by IntelliJ IDEA

// (powered by Fernflower decompiler)

//

package com.sandu.api.goods.input;

import io.swagger.annotations.ApiModelProperty;

import java.io.Serializable;

import java.math.BigDecimal;

import javax.validation.constraints.Min;

import org.hibernate.validator.constraints.NotBlank;

public class GoodsSKUAdd implements Serializable {

@ApiModelProperty(

value = "产品ID",

required = true

)

@Min(

value = 1L,

message = "ID必须大于0"

)

private Integer productId;

@ApiModelProperty("库存")

@Min(

value = 0L,

message = "库存必须大于等于0"

)

private Integer repertory;

@ApiModelProperty("优惠价")

@Min(

value = 0L,

message = "优惠价必须大于等于0"

)

private BigDecimal cost;

@ApiModelProperty("规格图片")

private Integer spePictureId;

@ApiModelProperty("属性值ID")

@NotBlank(

message = "属性ID为空！！！"

)

private String attrValueIds;

public GoodsSKUAdd() {

}

public Integer getProductId() {

return this.productId;

}

public Integer getRepertory() {

return this.repertory;

}

public BigDecimal getCost() {

return this.cost;

}

public Integer getSpePictureId() {

return this.spePictureId;

}

public String getAttrValueIds() {

return this.attrValueIds;

}

public void setProductId(Integer productId) {

this.productId = productId;

}

public void setRepertory(Integer repertory) {

this.repertory = repertory;

}

public void setCost(BigDecimal cost) {

this.cost = cost;

}

public void setSpePictureId(Integer spePictureId) {

this.spePictureId = spePictureId;

}

public void setAttrValueIds(String attrValueIds) {

this.attrValueIds = attrValueIds;

}

public boolean equals(Object o) {

if (o == this) {

return true;

} else if (!(o instanceof GoodsSKUAdd)) {

return false;

} else {

GoodsSKUAdd other = (GoodsSKUAdd)o;

if (!other.canEqual(this)) {

return false;

} else {

label71: {

Object this$productId = this.getProductId();

Object other$productId = other.getProductId();

if (this$productId == null) {

if (other$productId == null) {

break label71;

}

} else if (this$productId.equals(other$productId)) {

break label71;

}

return false;

}

Object this$repertory = this.getRepertory();

Object other$repertory = other.getRepertory();

if (this$repertory == null) {

if (other$repertory != null) {

return false;

}

} else if (!this$repertory.equals(other$repertory)) {

return false;

}

label57: {

Object this$cost = this.getCost();

Object other$cost = other.getCost();

if (this$cost == null) {

if (other$cost == null) {

break label57;

}

} else if (this$cost.equals(other$cost)) {

break label57;

}

return false;

}

Object this$spePictureId = this.getSpePictureId();

Object other$spePictureId = other.getSpePictureId();

if (this$spePictureId == null) {

if (other$spePictureId != null) {

return false;

}

} else if (!this$spePictureId.equals(other$spePictureId)) {

return false;

}

Object this$attrValueIds = this.getAttrValueIds();

Object other$attrValueIds = other.getAttrValueIds();

if (this$attrValueIds == null) {

if (other$attrValueIds == null) {

return true;

}

} else if (this$attrValueIds.equals(other$attrValueIds)) {

return true;

}

return false;

}

}

}

protected boolean canEqual(Object other) {

return other instanceof GoodsSKUAdd;

}

public int hashCode() {

int PRIME = true;

int result = 1;

Object $productId = this.getProductId();

int result = result \* 59 + ($productId == null ? 43 : $productId.hashCode());

Object $repertory = this.getRepertory();

result = result \* 59 + ($repertory == null ? 43 : $repertory.hashCode());

Object $cost = this.getCost();

result = result \* 59 + ($cost == null ? 43 : $cost.hashCode());

Object $spePictureId = this.getSpePictureId();

result = result \* 59 + ($spePictureId == null ? 43 : $spePictureId.hashCode());

Object $attrValueIds = this.getAttrValueIds();

result = result \* 59 + ($attrValueIds == null ? 43 : $attrValueIds.hashCode());

return result;

}

public String toString() {

return "GoodsSKUAdd(productId=" + this.getProductId() + ", repertory=" + this.getRepertory() + ", cost=" + this.getCost() + ", spePictureId=" + this.getSpePictureId() + ", attrValueIds=" + this.getAttrValueIds() + ")";

}

}

//

// Source code recreated from a .class file by IntelliJ IDEA

// (powered by Fernflower decompiler)

//

package com.sandu.api.goods.input;

import io.swagger.annotations.ApiModelProperty;

import java.io.Serializable;

import java.util.List;

import javax.validation.constraints.Min;

import javax.validation.constraints.NotNull;

import org.hibernate.validator.constraints.NotEmpty;

public class GoodsSKUListAdd implements Serializable {

@ApiModelProperty("sku集合")

@NotEmpty(

message = "传入的SKU为空"

)

private List<GoodsSKUAdd> goodsSKUAddList;

@ApiModelProperty("用户ID")

@NotNull(

message = "用户ID为空"

)

@Min(

value = 1L,

message = "用户ID错误"

)

private Integer userId;

@ApiModelProperty("SpuId")

@NotNull(

message = "SpuId为空"

)

@Min(

value = 1L,

message = "SpuId错误"

)

private Integer spuId;

public GoodsSKUListAdd() {

}

public List<GoodsSKUAdd> getGoodsSKUAddList() {

return this.goodsSKUAddList;

}

public Integer getUserId() {

return this.userId;

}

public Integer getSpuId() {

return this.spuId;

}

public void setGoodsSKUAddList(List<GoodsSKUAdd> goodsSKUAddList) {

this.goodsSKUAddList = goodsSKUAddList;

}

public void setUserId(Integer userId) {

this.userId = userId;

}

public void setSpuId(Integer spuId) {

this.spuId = spuId;

}

public boolean equals(Object o) {

if (o == this) {

return true;

} else if (!(o instanceof GoodsSKUListAdd)) {

return false;

} else {

GoodsSKUListAdd other = (GoodsSKUListAdd)o;

if (!other.canEqual(this)) {

return false;

} else {

label47: {

Object this$goodsSKUAddList = this.getGoodsSKUAddList();

Object other$goodsSKUAddList = other.getGoodsSKUAddList();

if (this$goodsSKUAddList == null) {

if (other$goodsSKUAddList == null) {

break label47;

}

} else if (this$goodsSKUAddList.equals(other$goodsSKUAddList)) {

break label47;

}

return false;

}

Object this$userId = this.getUserId();

Object other$userId = other.getUserId();

if (this$userId == null) {

if (other$userId != null) {

return false;

}

} else if (!this$userId.equals(other$userId)) {

return false;

}

Object this$spuId = this.getSpuId();

Object other$spuId = other.getSpuId();

if (this$spuId == null) {

if (other$spuId != null) {

return false;

}

} else if (!this$spuId.equals(other$spuId)) {

return false;

}

return true;

}

}

}

protected boolean canEqual(Object other) {

return other instanceof GoodsSKUListAdd;

}

public int hashCode() {

int PRIME = true;

int result = 1;

Object $goodsSKUAddList = this.getGoodsSKUAddList();

int result = result \* 59 + ($goodsSKUAddList == null ? 43 : $goodsSKUAddList.hashCode());

Object $userId = this.getUserId();

result = result \* 59 + ($userId == null ? 43 : $userId.hashCode());

Object $spuId = this.getSpuId();

result = result \* 59 + ($spuId == null ? 43 : $spuId.hashCode());

return result;

}

public String toString() {

return "GoodsSKUListAdd(goodsSKUAddList=" + this.getGoodsSKUAddList() + ", userId=" + this.getUserId() + ", spuId=" + this.getSpuId() + ")";

}

}

//

// Source code recreated from a .class file by IntelliJ IDEA

// (powered by Fernflower decompiler)

//

package com.sandu.api.goods.input;

import io.swagger.annotations.ApiModelProperty;

import java.io.Serializable;

import java.math.BigDecimal;

import java.util.Date;

import javax.validation.constraints.Future;

import javax.validation.constraints.Min;

import javax.validation.constraints.NotNull;

import javax.validation.constraints.Size;

import org.hibernate.validator.constraints.NotBlank;

public class GoodsSPUAdd implements Serializable {

@ApiModelProperty(

value = "商品ID",

required = true

)

@Min(

value = 1L,

message = "ID必须大于0"

)

private Integer spuId;

@ApiModelProperty(

value = "商品名称",

required = true

)

@NotBlank(

message = "商品名称不能为空"

)

@Size(

max = 100,

message = "商品名称最大为100个字"

)

private String name;

@ApiModelProperty("商品描述")

private String describe;

@ApiModelProperty("主缩略图ID")

private Integer mainPicId;

@ApiModelProperty("所有图片ID，逗号分隔")

private String picIds;

@ApiModelProperty("限购信息(0不限购，>0为限购数量，<0只限会员购买)")

@NotNull(

message = "限购信息不能为空"

)

private Integer limitation;

@ApiModelProperty("固定运费")

@Min(

value = 0L,

message = "运费必须大于等于0"

)

private BigDecimal fixTransCost;

@ApiModelProperty("是否预售")

@NotNull

private Integer presell;

@ApiModelProperty("预售结束时间")

@Future(

message = "截止时间不能早于当前时间"

)

private Date presellDDL;

@ApiModelProperty("支付尾款开始时间")

@Future(

message = "支付尾款开始时间不能早于当前时间"

)

private Date payTailStarttime;

@ApiModelProperty("支付尾款结束时间")

@Future(

message = "支付尾款结束时间不能早于当前时间"

)

private Date payTailEndtime;

@ApiModelProperty("定金")

@Min(

value = 0L,

message = "定金必须大于等于0"

)

private BigDecimal earnest;

@ApiModelProperty("预售商品发货时间")

@Min(

value = 0L,

message = "预售商品发货时间必须大于等于0"

)

private Integer shipmentsPresell;

@ApiModelProperty("总库存")

@Min(

value = 0L,

message = "总库存必须大于等于0"

)

private Integer totalRepertory;

@ApiModelProperty("是否特卖")

@NotNull

private Integer isSpeSell;

@ApiModelProperty("最早发货时间（天数）")

@Min(

value = 0L,

message = "最早发货时间（天数）必须大于等于0"

)

private Integer shipmentsDay;

@ApiModelProperty("最早发货时间（日期）")

@Future(

message = "最早发货时间（日期）不能早于当前时间"

)

private Date shipmentsDate;

private String creator;

private Date gmtCreate;

private String modifier;

private Date gmtModified;

private Integer isDeleted;

public GoodsSPUAdd() {

}

public Integer getSpuId() {

return this.spuId;

}

public String getName() {

return this.name;

}

public String getDescribe() {

return this.describe;

}

public Integer getMainPicId() {

return this.mainPicId;

}

public String getPicIds() {

return this.picIds;

}

public Integer getLimitation() {

return this.limitation;

}

public BigDecimal getFixTransCost() {

return this.fixTransCost;

}

public Integer getPresell() {

return this.presell;

}

public Date getPresellDDL() {

return this.presellDDL;

}

public Date getPayTailStarttime() {

return this.payTailStarttime;

}

public Date getPayTailEndtime() {

return this.payTailEndtime;

}

public BigDecimal getEarnest() {

return this.earnest;

}

public Integer getShipmentsPresell() {

return this.shipmentsPresell;

}

public Integer getTotalRepertory() {

return this.totalRepertory;

}

public Integer getIsSpeSell() {

return this.isSpeSell;

}

public Integer getShipmentsDay() {

return this.shipmentsDay;

}

public Date getShipmentsDate() {

return this.shipmentsDate;

}

public String getCreator() {

return this.creator;

}

public Date getGmtCreate() {

return this.gmtCreate;

}

public String getModifier() {

return this.modifier;

}

public Date getGmtModified() {

return this.gmtModified;

}

public Integer getIsDeleted() {

return this.isDeleted;

}

public void setSpuId(Integer spuId) {

this.spuId = spuId;

}

public void setName(String name) {

this.name = name;

}

public void setDescribe(String describe) {

this.describe = describe;

}

public void setMainPicId(Integer mainPicId) {

this.mainPicId = mainPicId;

}

public void setPicIds(String picIds) {

this.picIds = picIds;

}

public void setLimitation(Integer limitation) {

this.limitation = limitation;

}

public void setFixTransCost(BigDecimal fixTransCost) {

this.fixTransCost = fixTransCost;

}

public void setPresell(Integer presell) {

this.presell = presell;

}

public void setPresellDDL(Date presellDDL) {

this.presellDDL = presellDDL;

}

public void setPayTailStarttime(Date payTailStarttime) {

this.payTailStarttime = payTailStarttime;

}

public void setPayTailEndtime(Date payTailEndtime) {

this.payTailEndtime = payTailEndtime;

}

public void setEarnest(BigDecimal earnest) {

this.earnest = earnest;

}

public void setShipmentsPresell(Integer shipmentsPresell) {

this.shipmentsPresell = shipmentsPresell;

}

public void setTotalRepertory(Integer totalRepertory) {

this.totalRepertory = totalRepertory;

}

public void setIsSpeSell(Integer isSpeSell) {

this.isSpeSell = isSpeSell;

}

public void setShipmentsDay(Integer shipmentsDay) {

this.shipmentsDay = shipmentsDay;

}

public void setShipmentsDate(Date shipmentsDate) {

this.shipmentsDate = shipmentsDate;

}

public void setCreator(String creator) {

this.creator = creator;

}

public void setGmtCreate(Date gmtCreate) {

this.gmtCreate = gmtCreate;

}

public void setModifier(String modifier) {

this.modifier = modifier;

}

public void setGmtModified(Date gmtModified) {

this.gmtModified = gmtModified;

}

public void setIsDeleted(Integer isDeleted) {

this.isDeleted = isDeleted;

}

public boolean equals(Object o) {

if (o == this) {

return true;

} else if (!(o instanceof GoodsSPUAdd)) {

return false;

} else {

GoodsSPUAdd other = (GoodsSPUAdd)o;

if (!other.canEqual(this)) {

return false;

} else {

Object this$spuId = this.getSpuId();

Object other$spuId = other.getSpuId();

if (this$spuId == null) {

if (other$spuId != null) {

return false;

}

} else if (!this$spuId.equals(other$spuId)) {

return false;

}

Object this$name = this.getName();

Object other$name = other.getName();

if (this$name == null) {

if (other$name != null) {

return false;

}

} else if (!this$name.equals(other$name)) {

return false;

}

Object this$describe = this.getDescribe();

Object other$describe = other.getDescribe();

if (this$describe == null) {

if (other$describe != null) {

return false;

}

} else if (!this$describe.equals(other$describe)) {

return false;

}

label254: {

Object this$mainPicId = this.getMainPicId();

Object other$mainPicId = other.getMainPicId();

if (this$mainPicId == null) {

if (other$mainPicId == null) {

break label254;

}

} else if (this$mainPicId.equals(other$mainPicId)) {

break label254;

}

return false;

}

label247: {

Object this$picIds = this.getPicIds();

Object other$picIds = other.getPicIds();

if (this$picIds == null) {

if (other$picIds == null) {

break label247;

}

} else if (this$picIds.equals(other$picIds)) {

break label247;

}

return false;

}

Object this$limitation = this.getLimitation();

Object other$limitation = other.getLimitation();

if (this$limitation == null) {

if (other$limitation != null) {

return false;

}

} else if (!this$limitation.equals(other$limitation)) {

return false;

}

label233: {

Object this$fixTransCost = this.getFixTransCost();

Object other$fixTransCost = other.getFixTransCost();

if (this$fixTransCost == null) {

if (other$fixTransCost == null) {

break label233;

}

} else if (this$fixTransCost.equals(other$fixTransCost)) {

break label233;

}

return false;

}

label226: {

Object this$presell = this.getPresell();

Object other$presell = other.getPresell();

if (this$presell == null) {

if (other$presell == null) {

break label226;

}

} else if (this$presell.equals(other$presell)) {

break label226;

}

return false;

}

Object this$presellDDL = this.getPresellDDL();

Object other$presellDDL = other.getPresellDDL();

if (this$presellDDL == null) {

if (other$presellDDL != null) {

return false;

}

} else if (!this$presellDDL.equals(other$presellDDL)) {

return false;

}

Object this$payTailStarttime = this.getPayTailStarttime();

Object other$payTailStarttime = other.getPayTailStarttime();

if (this$payTailStarttime == null) {

if (other$payTailStarttime != null) {

return false;

}

} else if (!this$payTailStarttime.equals(other$payTailStarttime)) {

return false;

}

label205: {

Object this$payTailEndtime = this.getPayTailEndtime();

Object other$payTailEndtime = other.getPayTailEndtime();

if (this$payTailEndtime == null) {

if (other$payTailEndtime == null) {

break label205;

}

} else if (this$payTailEndtime.equals(other$payTailEndtime)) {

break label205;

}

return false;

}

label198: {

Object this$earnest = this.getEarnest();

Object other$earnest = other.getEarnest();

if (this$earnest == null) {

if (other$earnest == null) {

break label198;

}

} else if (this$earnest.equals(other$earnest)) {

break label198;

}

return false;

}

Object this$shipmentsPresell = this.getShipmentsPresell();

Object other$shipmentsPresell = other.getShipmentsPresell();

if (this$shipmentsPresell == null) {

if (other$shipmentsPresell != null) {

return false;

}

} else if (!this$shipmentsPresell.equals(other$shipmentsPresell)) {

return false;

}

label184: {

Object this$totalRepertory = this.getTotalRepertory();

Object other$totalRepertory = other.getTotalRepertory();

if (this$totalRepertory == null) {

if (other$totalRepertory == null) {

break label184;

}

} else if (this$totalRepertory.equals(other$totalRepertory)) {

break label184;

}

return false;

}

Object this$isSpeSell = this.getIsSpeSell();

Object other$isSpeSell = other.getIsSpeSell();

if (this$isSpeSell == null) {

if (other$isSpeSell != null) {

return false;

}

} else if (!this$isSpeSell.equals(other$isSpeSell)) {

return false;

}

label170: {

Object this$shipmentsDay = this.getShipmentsDay();

Object other$shipmentsDay = other.getShipmentsDay();

if (this$shipmentsDay == null) {

if (other$shipmentsDay == null) {

break label170;

}

} else if (this$shipmentsDay.equals(other$shipmentsDay)) {

break label170;

}

return false;

}

Object this$shipmentsDate = this.getShipmentsDate();

Object other$shipmentsDate = other.getShipmentsDate();

if (this$shipmentsDate == null) {

if (other$shipmentsDate != null) {

return false;

}

} else if (!this$shipmentsDate.equals(other$shipmentsDate)) {

return false;

}

Object this$creator = this.getCreator();

Object other$creator = other.getCreator();

if (this$creator == null) {

if (other$creator != null) {

return false;

}

} else if (!this$creator.equals(other$creator)) {

return false;

}

Object this$gmtCreate = this.getGmtCreate();

Object other$gmtCreate = other.getGmtCreate();

if (this$gmtCreate == null) {

if (other$gmtCreate != null) {

return false;

}

} else if (!this$gmtCreate.equals(other$gmtCreate)) {

return false;

}

label142: {

Object this$modifier = this.getModifier();

Object other$modifier = other.getModifier();

if (this$modifier == null) {

if (other$modifier == null) {

break label142;

}

} else if (this$modifier.equals(other$modifier)) {

break label142;

}

return false;

}

label135: {

Object this$gmtModified = this.getGmtModified();

Object other$gmtModified = other.getGmtModified();

if (this$gmtModified == null) {

if (other$gmtModified == null) {

break label135;

}

} else if (this$gmtModified.equals(other$gmtModified)) {

break label135;

}

return false;

}

Object this$isDeleted = this.getIsDeleted();

Object other$isDeleted = other.getIsDeleted();

if (this$isDeleted == null) {

if (other$isDeleted != null) {

return false;

}

} else if (!this$isDeleted.equals(other$isDeleted)) {

return false;

}

return true;

}

}

}

protected boolean canEqual(Object other) {

return other instanceof GoodsSPUAdd;

}

public int hashCode() {

int PRIME = true;

int result = 1;

Object $spuId = this.getSpuId();

int result = result \* 59 + ($spuId == null ? 43 : $spuId.hashCode());

Object $name = this.getName();

result = result \* 59 + ($name == null ? 43 : $name.hashCode());

Object $describe = this.getDescribe();

result = result \* 59 + ($describe == null ? 43 : $describe.hashCode());

Object $mainPicId = this.getMainPicId();

result = result \* 59 + ($mainPicId == null ? 43 : $mainPicId.hashCode());

Object $picIds = this.getPicIds();

result = result \* 59 + ($picIds == null ? 43 : $picIds.hashCode());

Object $limitation = this.getLimitation();

result = result \* 59 + ($limitation == null ? 43 : $limitation.hashCode());

Object $fixTransCost = this.getFixTransCost();

result = result \* 59 + ($fixTransCost == null ? 43 : $fixTransCost.hashCode());

Object $presell = this.getPresell();

result = result \* 59 + ($presell == null ? 43 : $presell.hashCode());

Object $presellDDL = this.getPresellDDL();

result = result \* 59 + ($presellDDL == null ? 43 : $presellDDL.hashCode());

Object $payTailStarttime = this.getPayTailStarttime();

result = result \* 59 + ($payTailStarttime == null ? 43 : $payTailStarttime.hashCode());

Object $payTailEndtime = this.getPayTailEndtime();

result = result \* 59 + ($payTailEndtime == null ? 43 : $payTailEndtime.hashCode());

Object $earnest = this.getEarnest();

result = result \* 59 + ($earnest == null ? 43 : $earnest.hashCode());

Object $shipmentsPresell = this.getShipmentsPresell();

result = result \* 59 + ($shipmentsPresell == null ? 43 : $shipmentsPresell.hashCode());

Object $totalRepertory = this.getTotalRepertory();

result = result \* 59 + ($totalRepertory == null ? 43 : $totalRepertory.hashCode());

Object $isSpeSell = this.getIsSpeSell();

result = result \* 59 + ($isSpeSell == null ? 43 : $isSpeSell.hashCode());

Object $shipmentsDay = this.getShipmentsDay();

result = result \* 59 + ($shipmentsDay == null ? 43 : $shipmentsDay.hashCode());

Object $shipmentsDate = this.getShipmentsDate();

result = result \* 59 + ($shipmentsDate == null ? 43 : $shipmentsDate.hashCode());

Object $creator = this.getCreator();

result = result \* 59 + ($creator == null ? 43 : $creator.hashCode());

Object $gmtCreate = this.getGmtCreate();

result = result \* 59 + ($gmtCreate == null ? 43 : $gmtCreate.hashCode());

Object $modifier = this.getModifier();

result = result \* 59 + ($modifier == null ? 43 : $modifier.hashCode());

Object $gmtModified = this.getGmtModified();

result = result \* 59 + ($gmtModified == null ? 43 : $gmtModified.hashCode());

Object $isDeleted = this.getIsDeleted();

result = result \* 59 + ($isDeleted == null ? 43 : $isDeleted.hashCode());

return result;

}

public String toString() {

return "GoodsSPUAdd(spuId=" + this.getSpuId() + ", name=" + this.getName() + ", describe=" + this.getDescribe() + ", mainPicId=" + this.getMainPicId() + ", picIds=" + this.getPicIds() + ", limitation=" + this.getLimitation() + ", fixTransCost=" + this.getFixTransCost() + ", presell=" + this.getPresell() + ", presellDDL=" + this.getPresellDDL() + ", payTailStarttime=" + this.getPayTailStarttime() + ", payTailEndtime=" + this.getPayTailEndtime() + ", earnest=" + this.getEarnest() + ", shipmentsPresell=" + this.getShipmentsPresell() + ", totalRepertory=" + this.getTotalRepertory() + ", isSpeSell=" + this.getIsSpeSell() + ", shipmentsDay=" + this.getShipmentsDay() + ", shipmentsDate=" + this.getShipmentsDate() + ", creator=" + this.getCreator() + ", gmtCreate=" + this.getGmtCreate() + ", modifier=" + this.getModifier() + ", gmtModified=" + this.getGmtModified() + ", isDeleted=" + this.getIsDeleted() + ")";

}

}