

- 产品编码双轨制实施方案 (SPU/SKU Coding System)
 - 1. 核心理念
 - 1.1 编码体系概览
 - 1.2 战略价值评估 (Strategic Value)
 - 2. 详细编码规则
 - 2.1 SPU 编码 (特征码)
 - 2.2 SKU 编码 (双码并行)
 - A. SKU 短码 (Logistics Code) - 核心
 - B. SKU 特征码 (Descriptive Code) - 辅助
 - 2.3 特殊场景处理 (通用件/多款通用)
 - 3. 数据库模型设计变更 (Schema)
 - 3.1 Category (分类表)
 - 3.2 Product (SPU表)
 - 3.3 AttributeDefinition (属性定义表)
 - 3.4 ProductFitment (适配车型表)
 - 3.5 ProductVariant (SKU表)
 - 4. 业务流程逻辑
 - 4.1 产品录入 (UI交互)
 - 4.2 标签打印逻辑
 - 5. 开发任务清单

产品编码双轨制实施方案 (SPU/SKU Coding System)

1. 核心理念

为解决汽配行业 SKU 数量庞大、属性复杂与仓库物流高效作业之间的矛盾，本系统采用双轨制编码体系。

1.1 编码体系概览

层级	编码名称	类型	核心用途	示例	是否做条码
SPU	SPU特征码 (Feature Code)	逻辑码	搜索、分类、聚合变体	HL-CHE-SIL-07-13	✗ 否
SKU	SKU短码 (Logistics Code)	主键码	WMS扫码、标签打印、唯一标识	127240081WB	✓ 是
SKU	SKU特征码 (Descriptive Code)	辅助码	标签文字说明、人工核对、防错	...-2P-CH-AM-WB	✗ 否

1.2 战略价值评估 (Strategic Value)

本方案经过 CTO、供应链专家、汽配业务专家及跨境电商专家联合评审，旨在构建企业数字化转型的底层基建：

- 兼容历史 (Legacy Compatibility):** 通过保留 **SKU短码**，完美兼容旧系统库存和实物标签，零切换成本。
- 数据驱动 (Data Driven):** 通过 **特征码** 强制结构化业务数据（车型、年份、属性），为未来 AI 选品、自动客服奠定基础。
- 多平台适配 (Multi-Platform):** 特征码包含标准 ACES 数据 (Make/Model/Year)，可自动映射 Amazon/eBay 刊登数据。
- 扩展性 (Scalability):** 特征码逻辑独立于物理库存，支持未来业务规则升级（如新增版本号）而不破坏底层数据。

2. 详细编码规则

2.1 SPU 编码 (特征码)

定义：SPU (Standard Product Unit) 是虚拟的产品集合（如“07-13款 索罗德 大灯”），不直接关联库存。

结构公式：

[类目缩写]-[品牌缩写]-[车型缩写]-[年份段]

生成逻辑:

1. 类目缩写 (Category Abbr): 来自 `Category.abbreviation`。
 - 例: Headlight -> HL
2. 品牌缩写 (Brand Abbr): 来自 `ProductVehicleBrand.abbreviation`。
 - 例: Chevrolet -> CHE
3. 车型缩写 (Model Abbr): 来自产品录入时的 `vehicle_model_abbr` 字段。
 - 例: Silverado -> SIL
4. 年份段 (Year Range): 来自 `model_year_start` 和 `model_year_end`。
 - 例: 2007-2013 -> 07-13

最终示例: HL-CHE-SIL-07-13

2.2 SKU 编码 (双码并行)

SKU (Stock Keeping Unit) 是实际库存单元。

A. SKU 短码 (Logistics Code) - 核心

定义: 仓库作业的唯一凭证, 要求短小、易扫码。 **结构公式** (针对新品):

[类目数字码 3位][流水号 5位][核心后缀 2位]

- 旧品兼容: 直接存储历史编码 (如 127240081WB)。
- 新品生成:
 - 类目数字码: 188 (对应大灯)
 - 流水号: 00521 (系统自动递增)
 - 后缀: WB (With Bulb, 选填)
 - 结果: 18800521WB 约束: 长度建议控制在 9-12 位, 全大写字母+数字。

B. SKU 特征码 (Descriptive Code) - 辅助

定义: SPU 特征码的延伸, 用于描述该变体的具体属性。 **结构公式**:

[SPU特征码]-[属性1]-[属性2]-...

属性代码字典 (示例):

属性维度	英文名称	选项	代码 (Code)
规格	Set Size	Pair (一对)	2P
		Left (左边)	L
外壳颜色	Housing Color	Chrome (电镀)	CH
		Black (黑底)	BK
转向灯色	Signal Color	Amber (黄光)	AM
		Clear (白光)	CL
组件	Component	With Bulb (带泡)	WB

生成示例: HL-CHE-SIL-07-13-2P-CH-AM-WB

属性排序规则 (重要): 为了保证编码唯一性，属性必须按照固定权重排序，不可随意排列：

- 1. 规格 (Set/Pair) -> 2P
- 2. 位置 (Position) -> L/R
- 3. 外观 (Color) -> CH/BK/SM
- 4. 组件 (Feature) -> WB/LED

示例: 先颜色后位置 ...-SM-L ❌ -> 自动纠正为 ...-L-SM ✅

2.3 特殊场景处理 (通用件/多款通用)

场景: 一个零件适配多个年份段或多个车型。

- **案例:** 尾灯通用 2003-2013 (跨越了两代车型 03-06 和 07-13)。

原则: 最大覆盖范围原则 (Max Coverage Principle) + 主车型法则 (Primary Application)。 SPU 编码应体现该零件**物理上的最大兼容范围**。

- **处理方式:**
 - SPU 码: TL-CHE-SIL-03-13
 - 含义: 适用于 Silverado 03款至13款。
 - 搜索逻辑: 搜 "05 Silverado" -> 命中; 搜 "10 Silverado" -> 命中。

注意:

- 1. 属性级差异 (颜色/左右/带灯泡) -> 同 SPU, 不同 SKU。

2. 设计级差异 (卤素款/LED款/不同模具) -> 不同 SPU。

3. 数据库模型设计变更 (Schema)

3.1 Category (分类表)

新增字段支持双编码前缀。

```
class Category(db.Model):
    # ...
    code: str          # 纯数字码, 如 "188" (用于 SKU 短码)
    abbreviation: str   # 字符缩写, 如 "HL" (用于 SPU 特征码)
```

3.2 Product (SPU表)

SPU 仅存储特征码, 无短码。采用 JSONB 存储编码元数据, 以兼容不同品类 (汽配/非汽配) 和未来扩展。

```
class Product(db.Model):
    # ...
    # 核心标识: SPU特征码
    spu_code: str = Column(String(100), unique=True, comment="HL-CHE-SIL-07-13")

    # 编码元数据 (JSONB)
    # 汽配存: {"brand": "CHE", "model": "SIL", "year": "07-13"}
    # 通用存: {"series": "Pro", "power": "60W"}
    spu_coding_metadata: dict = Column(JSONB, comment="生成SPU码所用的元数据")
```

3.3 AttributeDefinition (属性定义表)

新增 `code_weight` 字段, 从数据库层面强制规定属性在编码中的顺序。

```
class AttributeDefinition(db.Model):
    # ...
    # 编码生成权重 (数字越小越靠前)
    # 10: 规格 (Set/Pair)
    # 20: 位置 (Position)
```

```
# 30: 颜色 (Color)
# 99: 其他
code_weight: int = Column(Integer, default=99, comment="生成SKU特征码时的排序权重")
include_in_code: bool = Column(Boolean, default=True)
```

3.4 ProductFitment (适配车型表)

用于记录完整适配性 (Full Fitment)，与 SPU 主车型逻辑分离。

```
class ProductFitment(db.Model):
    # ...
    # 记录该产品实际适配的所有车型
    make: str
    model: str
    year_start: int
    year_end: int
    notes: str # e.g. "Does NOT Fit Hybrid"
```

3.5 ProductVariant (SKU表)

同时存储短码和特征码。

```
class ProductVariant(db.Model):
    # ...
    # 1. 物流主键 (短码) - 必须唯一
    sku: str = Column(String(50), unique=True, index=True, comment="127240081WB")

    # 2. 业务描述 (特征码) - 允许重复(极少数情况)或加后缀区分
    feature_code: str = Column(String(200), index=True, comment="...-2P-CH-AM-WB")

    # 3. 生成短码用的后缀
    suffix_code: str = Column(String(10), comment="WB")
```

4. 业务流程逻辑

4.1 产品录入 (UI交互)

1. 第一步：填写 SPU 信息

- 选择分类 **Headlight** -> 系统获取 **HL**。
- 选择品牌 **Chevy** -> 系统获取 **CHE**。
- 输入车型 **Silverado** 对应缩写 **SIL** (可建立常用车型字典)。
- 输入年份 **2007, 2013**。
- 🖱 界面实时预览 SPU 码: **HL-CHE-SIL-07-13**。

2. 第二步：添加 SKU 变体

- 用户勾选属性: **Chrome, Amber, Pair**。
- 系统根据字典匹配代码: **CH, AM, 2P**。
- 🖱 自动生成 SKU 特征码: **HL-CHE-SIL-07-13-2P-CH-AM-WB**。

3. 第三步：确定 SKU 短码

- **模式 A (存量)**: 用户勾选“手动输入”，填入旧编码 **127240081WB**。
- **模式 B (新品)**: 系统自动计算 **188 + Sequence + WB**。

4.2 标签打印逻辑

- **条码内容**: 只包含 SKU 短码 (**127240081WB**)。
- **文字内容**:
 - Line 1: SKU 短码 (大字)
 - Line 2: 中文名称 (如: 07-13 索罗德大灯电镀一对)
 - Line 3: SKU 特征码 (小字, 用于核对: **...-2P-CH-AM-WB**)

5. 开发任务清单

- ☐ **数据库迁移**:
 - ☐ **Category** 表添加 **abbreviation** 字段。
 - ☐ **Product** 表重构 **spu_code** 逻辑, 添加 **model_abbr** 等字段。
 - ☐ **ProductVariant** 表添加 **feature_code** 字段。
- ☐ **字典数据准备**:
 - ☐ 整理常用汽配属性 (Attribute) 及其对应的 2 位代码 (Code)。
 - ☐ 整理常用车型 (Vehicle Model) 及其缩写。
- ☐ **后端服务开发**:

- ☐ 实现 `CodeBuilderService`，包含 `generate_spu_feature_code` 和 `generate_sku_short_code` 方法。
- ☐ 前端页面适配:
 - ☐ 产品发布页面支持属性选择自动生成编码预览。