

# 数据建模产品培训手册

2025年8月

扬数起元 [www.yonsum.com](http://www.yonsum.com)



官方公众号



# 目录

1. 培训概述	4
1.1 培训目标	4
1.2 适用人员	4
1.3 时间安排	4
1.4 预期成果	5
2. 数据建模理论基础	6
2.1 什么是数据建模	6
2.2 为什么数据建模如此重要	6
2.3 数据建模与其他"建模"的区别	7
3. 三层建模方法论	9
3.1 三层模型概览	9
3.2 概念模型：业务视角的"草图"	9
3.3 逻辑模型：系统视角的"细化图"	10
3.4 物理模型：落实到数据库的"施工蓝图"	11
3.5 三层模型的递进关系	12
4. 外卖平台业务场景实践	13
4.1 业务场景分析	13
4.2 概念模型设计	13
4.3 逻辑模型设计	15
4.4 物理模型设计	18
4.5 模型验证与优化	22
5. PDMAas建模工具应用	24
5.1 PDMAas建模工具概述	24
5.2 环境准备与登录	24
5.3 模型资源分类	30
5.4 概念建模操作	31
5.5 逻辑建模操作	36
5.6 物理建模操作	42
5.7 关系图设计	44
5.8 数据导入导出	47
5.9 团队协作功能	53
6. 实战练习与评估	58
6.1 基础练习	58
6.2 进阶练习	59

# 1. 培训概述

## 1.1 培训目标

本培训旨在帮助学员掌握数据建模的核心理论与实践技能，具体目标包括：

- 理解数据建模的基本概念和重要价值
- 掌握三层递进式建模方法论（概念-逻辑-物理）
- 熟练运用PDMAas建模工具进行实际项目开发
- 能够独立完成业务场景到数据库设计的完整流程
- 具备解决常见数据建模问题的能力

## 1.2 适用人员

本培训适合以下人员参加：

- **数据架构师**：负责企业数据架构设计的技术人员
- **数据分析师**：需要理解数据结构进行分析工作的人员
- **软件开发工程师**：参与系统开发的后端工程师
- **产品经理**：需要理解数据结构以便更好地进行产品设计
- **项目经理**：负责数据相关项目管理的人员

## 1.3 时间安排

## 培训总时长：2天（16小时）

- 第一天上午：理论基础 + 方法论（4小时）
- 第一天下午：业务场景实践（4小时）
- 第二天上午：工具应用培训（4小时）
- 第二天下午：实战练习 + 答疑（4小时）

## 1.4 预期成果

培训结束后，学员应能够：

1. 独立分析业务需求并设计数据模型
2. 熟练使用PDMAas工具完成建模任务
3. 输出符合规范的数据库设计文档
4. 解决项目中遇到的常见建模问题

## 2. 数据建模理论基础

### 2.1 什么是数据建模

**定义：**数据建模是基于对业务的理解，给数据做结构化设计的过程。它把业务里的对象（如客户、产品、订单）、行为（如下单、付款）和规则（如唯一、非空、格式要求），用结构化的数据蓝图描述出来，让数据易理解、可用、可分析。

简单来说，数据建模就是给业务设计数据结构。就像建筑师在建造房屋之前会设计蓝图一样，数据建模是业务利益相关方在进行数据开发前必备的“数据蓝图”。

#### 2.1.1 数据模型的核心要素

一个完整的数据模型包含三个核心要素：

- **实体 (Entity)：**业务中的核心对象，如客户、订单、商品
- **关系 (Relationship)：**实体之间的关联，如客户下订单、订单包含商品
- **约束 (Constraint)：**数据必须遵守的规则，如非空、唯一、格式限制

### 2.2 为什么数据建模如此重要

在企业数字化转型过程中，很多管理者都想实现以下目标：用好数据、管好数据、省钱、让业务跑得快、还想搞点高大上的AI。要实现这些目标，数据建模是必须先打好的地基。

#### 2.2.1 没有数据建模会遇到的问题

问题类型	具体表现	影响
------	------	----

搞不清家底	不知道自己有多少数据，也看不懂它们有什么用	决策缺乏数据支撑
管不住质量	数据脏乱差、安全性差、标准不统一	数据可信度低，分析结果不准确
沟通成本高	IT和业务开会就是鸡同鸭讲，开发又慢又贵还容易出错	项目延期，成本超支
AI成了摆设	想玩人工智能，喂进去的都是"垃圾数据"	AI应用效果差，投资回报率低

## 2.2.2 数据建模的价值

权威研究显示：数据建模能让开发效率提升15-70%，维护成本降低1-10%，有效减少因数据质量问题造成的损失（Gartner研究显示企业平均每年因此损失1290万美元）。

具体价值体现在：

1. 提升开发效率：明确的数据结构减少开发过程中的反复沟通
2. 降低维护成本：标准化的数据模型便于系统维护和扩展
3. 保证数据质量：从源头控制数据的准确性和一致性
4. 促进业务理解：可视化的模型帮助团队更好地理解业务

## 2.3 数据建模与其他"建模"的区别

在数据领域，"建模"是一个高频术语。数据建模、业务建模、数学建模、算法建模以及当前备受关注的大模型，虽然都含有"建模"二字，但其定位、目标与作用存在显著差异。

建模类型	聚焦点	产出物	解决问题
业务建模	业务流程规则	流程图/权责矩阵	业务怎么做
数据建模	数据结构设计	ER图/数据字典	数据怎么存
数学建模	业务关系量化	数学模型	问题怎么算
算法建模	可执行步骤	算法流程	功能怎么实现
大模型	语言规律学习	预训练模型	复杂交互怎么处理

**重要提示：**数据建模是后续所有建模工作的基础。只有数据建模做好了，后面的数学建模、算法建模和大模型才能真正用得上劲，帮你省钱、跑快业务、做好AI。

### 3. 三层建模方法论

数据建模不是一步到位的工作，而是一个从抽象到具体、从业务到技术的递进过程。就像盖房子要先画草图、再出施工图、最后确定砖瓦尺寸一样，数据建模分为三个层次，层层深入地把业务需求转化为可落地的数据库结构。

#### 3.1 三层模型概览

模型层次	关注点	参与角色	主要产出
概念模型	业务对象与关系	业务专家、产品经理	实体关系图
逻辑模型	数据结构与约束	数据架构师、分析师	详细数据字典
物理模型	技术实现与优化	数据库工程师、开发人员	DDL建表语句

#### 3.2 概念模型：业务视角的"草图"

##### 3.2.1 什么是概念模型

概念模型是数据建模的第一步，也是最贴近业务的一层。它不涉及任何技术细节，只聚焦于"业务中存在哪些核心对象，以及这些对象之间有什么关联"。

**概念模型的核心价值：**它是业务和技术的"翻译器"。很多企业的数据混乱，根源就是概念模型缺失——业务说的"客户"和技术理解的"用户"不是一回事，后续的数据结构自然会出现偏差。

##### 3.2.2 概念模型的关键要素

###### 1. 实体识别：找出业务中的核心对象

2. **关系定义**: 明确实体之间的关联

3. **业务规则**: 确定基本的业务约束

### 3.2.3 概念模型设计步骤

1. **业务调研**: 深入了解业务流程和需求

2. **实体识别**: 从业务描述中提取核心对象

3. **关系梳理**: 分析实体间的业务关联

4. **模型验证**: 与业务专家确认模型的正确性

## 3.3 逻辑模型: 系统视角的"细化图"

### 3.3.1 什么是逻辑模型

在概念模型明确业务框架后，逻辑模型进一步细化为含字段、主键、外键及依赖关系的结构化设计，以系统语言衔接业务与技术，且不绑定具体数据库类型。

### 3.3.2 逻辑模型的重要性

逻辑模型通过统一字段定义、固化业务规则、建立数据关联，消除业务与技术的认知偏差，为数据存储和使用提供标准化基础，是解决业务需求到技术实现翻译断层的核心环节。

### 3.3.3 逻辑模型设计要点

1. **字段定义**: 明确每个字段的名称、含义、数据类型

2. **主键设计**: 为每个实体定义唯一标识

3. **外键关系**: 建立实体间的引用关系

4. **约束规则**: 定义数据完整性约束

## 3.4 物理模型：落实到数据库的“施工蓝图”

### 3.4.1 什么是物理模型

物理模型将逻辑模型转化为数据库可直接执行的技术方案，通过明确存储格式、性能优化策略和技术约束，最终生成可落地的数据库构建方案。

**物理模型的关键作用：**物理模型是技术落地的“最后一公里”，直接决定系统的运行效率与稳定性。合理设计可让海量数据查询从秒级降至毫秒级；若设计失当（如缺索引、大表未分区），会导致系统卡顿、资源耗尽。

### 3.4.2 物理模型的核心要素

表结构技术化落地：

- **物理类型映射：**按数据库特性明确类型（如MySQL中的VARCHAR vs Oracle中的VARCHAR2）
- **约束与默认值：**固化技术规则（如PRIMARY KEY、CURRENT\_TIMESTAMP等）

性能优化设计：

- **索引策略：**按查询类型适配（哈希索引用于精准查询，B树索引用于范围查询）
- **分区规则：**按数据特征拆分大表（时间维度分区、业务维度分区）
- **存储引擎选型：**匹配业务特性（InnoDB支持事务，MyISAM优化读性能）

### 3.4.3 物理模型的典型产出

1. **DDL建表语句：**可直接执行的CREATE TABLE语句
2. **数据库设计文档：**明确字段-类型-用途的索引策略

## 3.5 三层模型的递进关系

**核心理念：**从概念模型到逻辑模型再到物理模型，是"业务抽象→逻辑细化→技术落地"的递进过程。

- 概念模型：明确"业务核心是什么"
- 逻辑模型：定义"数据结构长啥样"
- 物理模型：解决"技术上怎么实现"

**三层模型相互支撑：**脱离概念模型，数据会偏离业务本质；缺少逻辑模型，技术实现会失去标准；忽视物理模型，设计无法高效落地。

**灵活应用：**在实际操作过程中，三层模型并非强制流程，可根据实际情况做出相应取舍，进行适当简化（如简单业务可跳过概念模型直接进入逻辑设计）。

## 4. 外卖平台业务场景实践

本章将以外卖平台为例，展示从业务分析到数据建模的完整过程，帮助学员理解如何将三层建模方法论应用到实际项目中。

### 4.1 业务场景分析

#### 4.1.1 业务背景

外卖平台是一个连接用户、商家和配送员的三方平台，核心业务流程是：用户通过平台下单，商家接单并准备商品，配送员负责将商品送达用户。

#### 4.1.2 核心业务流程

1. 用户注册与登录：用户在平台注册账号并完成身份验证
2. 商家入驻：商家申请入驻平台，上传资质，配置商品信息
3. 配送员注册：配送员注册并通过平台审核，获得配送资格
4. 用户浏览下单：用户浏览商家和商品，添加到购物车并下单
5. 订单处理：商家接单，准备商品，等待配送员取货
6. 配送服务：配送员接单，取货，配送到用户指定地址
7. 订单完成：用户确认收货，订单完成，进行评价

### 4.2 概念模型设计

#### 4.2.1 实体识别

从外卖平台的业务流程中，我们可以识别出以下核心实体：

实体名称	业务含义	关键特征

客户 (CUSTOMER)	平台的终端用户，下单购买商品	需要注册登录，有配送地址
商家 (MERCHANT)	提供商品和服务的店铺	需要资质认证，有营业状态
配送员 (RIDER)	负责商品配送的工作人员	需要身份认证，有工作状态
商品 (PRODUCT)	商家提供的可购买物品	有价格、库存、分类等属性
订单 (ORDER)	客户的购买请求	包含商品信息和配送信息

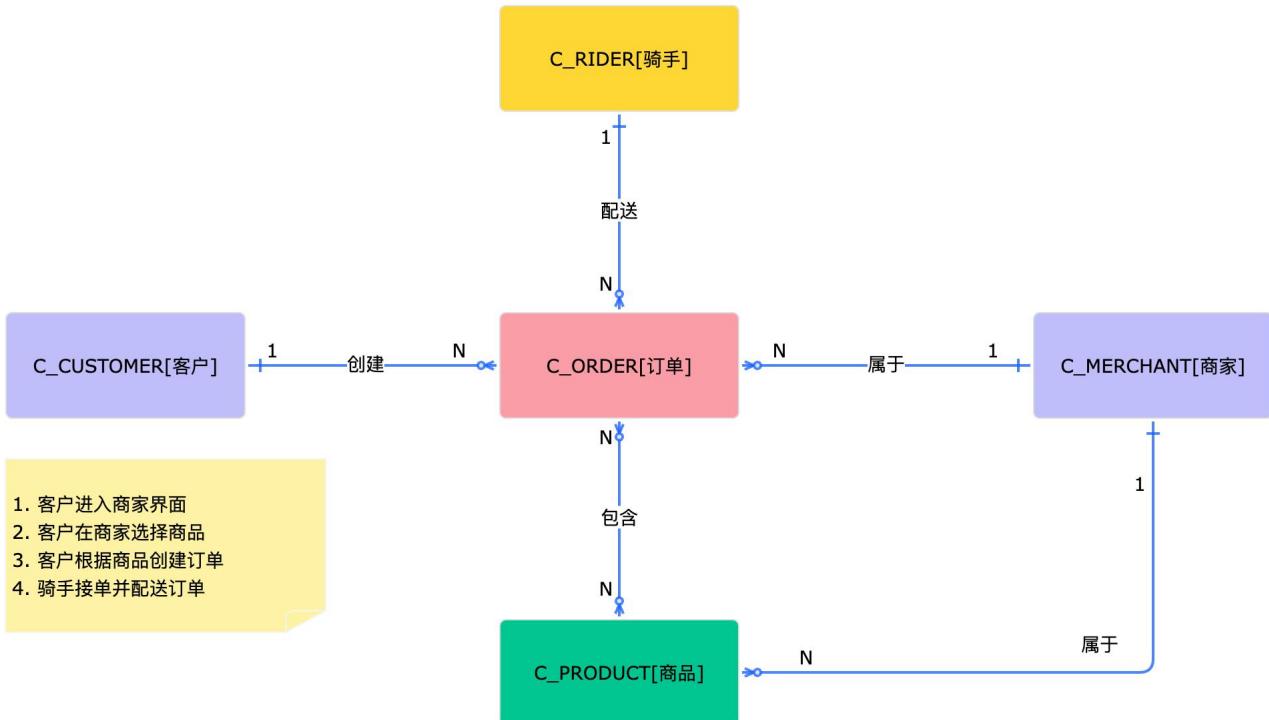
## 4.2.2 关系梳理

识别实体间的业务关系:

- 客户 → 订单: 客户创建订单 (一对多关系)
- 订单 → 商家: 订单关联商家 (多对一关系)
- 订单 → 商品: 订单包含商品 (多对多关系, 需要中间表)
- 订单 → 配送员: 订单分配给配送员 (多对一关系)
- 商家 → 商品: 商家提供商品 (一对多关系)

## 4.2.3 绘制模型ER关系图

根据上一步梳理的关系, 完成概念模型ER图绘制:



此图基于 PDmaas 绘制  
© Powered by yonsum.com

## 4.3 逻辑模型设计

### 4.3.1 逻辑实体字段及完整性约束设计

基于概念模型，我们进一步细化为逻辑模型，定义具体的字段和约束。

代码 L\_CUSTOMER 显示名称 客户

备注说明

列操作

	代码	名称	主键	不为空	基本数据类型	长度	小数点	备注	代码
1	CUSTOMER_ID	客户 ID	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	变长字符串	32		唯一标识客户	
2	FULL_NAME	客户姓名	<input type="checkbox"/>	<input checked="" type="checkbox"/>	变长字符串	50			
3	PHONE_NUMBER	联系电话	<input type="checkbox"/>	<input checked="" type="checkbox"/>	变长字符串	20		需符合手机号格式，唯一	
4	REGISTER_TIME	注册时间	<input type="checkbox"/>	<input checked="" type="checkbox"/>	日期时间				
5	DELIVERY_ADDRESS	常用配送地址	<input type="checkbox"/>	<input type="checkbox"/>	变长字符串	200			
6	ACCOUNT_STATUS	账户状态	<input type="checkbox"/>	<input checked="" type="checkbox"/>	变长字符串	20		枚举值：正常、冻结、注销，默认为正常	

代码 L\_ORDER

显示名称 订单

2

备注说明

代码		名称	主键	不为空	基本数据类型	长度	小数点	备注	代码
1	ORDER_ID	订单 ID	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	变长字符串	32		唯一标识订单	
2	CUSTOMER_ID	客户 ID	<input type="checkbox"/>	<input checked="" type="checkbox"/>	变长字符串	32		外键, 关联客户表	
3	MERCHANT_ID	商家 ID	<input type="checkbox"/>	<input checked="" type="checkbox"/>	变长字符串	32		外键, 关联商家表	
4	RIDER_ID	骑手 ID	<input type="checkbox"/>	<input type="checkbox"/>	变长字符串	32		外键, 关联骑手表, 配送前可空	
5	ORDER_STATUS	订单状态	<input type="checkbox"/>	<input checked="" type="checkbox"/>	变长字符串	20		枚举值: 待支付、已支付、已接单、配送中、已送达	
6	CREATE_TIME	创建时间	<input type="checkbox"/>	<input checked="" type="checkbox"/>	日期时间				
7	PAYMENT_TIME	支付时间	<input type="checkbox"/>	<input type="checkbox"/>	日期时间			已支付时非空	
8	TOTAL_AMOUNT	订单总金额	<input type="checkbox"/>	<input checked="" type="checkbox"/>	标准小数	10	2	大于 0	
9	DELIVERY_ADDRESS	本次配送地址	<input type="checkbox"/>	<input type="checkbox"/>	变长字符串	200			
10	CONTACT_PHONE	收件联系电话	<input type="checkbox"/>	<input type="checkbox"/>	变长字符串	20		需符合手机号格式	

代码 L\_MERCHANT

显示名称 商家

2

备注说明

代码		名称	主键	不为空	基本数据类型	长度	小数点	备注	代码
1	MERCHANT_ID	商家 ID	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	变长字符串	32		唯一标识商家	
2	MERCHANT_NAME	商家名称	<input type="checkbox"/>	<input checked="" type="checkbox"/>	变长字符串	100			
3	CONTACT_PERSON	联系人姓名	<input type="checkbox"/>	<input checked="" type="checkbox"/>	变长字符串	50			
4	CONTACT_PHONE	联系电话	<input type="checkbox"/>	<input checked="" type="checkbox"/>	变长字符串	20		需符合手机号格式	
5	BUSINESS_LICENSE	营业执照编号	<input type="checkbox"/>	<input checked="" type="checkbox"/>	变长字符串	50		唯一	
6	OPERATE_STATUS	经营状态	<input type="checkbox"/>	<input checked="" type="checkbox"/>	变长字符串	20		枚举值: 营业中、休息中、暂停营业, 默认为营业中	
7	ADDRESS	店铺地址	<input type="checkbox"/>	<input checked="" type="checkbox"/>	变长字符串	200			

代码 L\_PRODUCT

显示名称 商品

2

备注说明

代码		名称	主键	不为空	基本数据类型	长度	小数点	备注	代码
1	PRODUCT_ID	商品 ID	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	变长字符串	32		唯一标识商品	
2	MERCHANT_ID	商家 ID	<input type="checkbox"/>	<input checked="" type="checkbox"/>	变长字符串	32		外键, 关联商家表	
3	CATEGORY	商品分类	<input type="checkbox"/>	<input checked="" type="checkbox"/>	变长字符串	50			
4	PRODUCT_NAME	商品名称	<input type="checkbox"/>	<input checked="" type="checkbox"/>	变长字符串	100			
5	PRICE	商品单价	<input type="checkbox"/>	<input checked="" type="checkbox"/>	标准小数	10	2		
6	PRODUCT_SPEC	商品规格	<input type="checkbox"/>	<input type="checkbox"/>	变长字符串	100			
7	STOCK_QUANTITY	库存数量	<input type="checkbox"/>	<input checked="" type="checkbox"/>	标准整数	10			
8	PRODUCT_STATUS	商品状态	<input type="checkbox"/>	<input checked="" type="checkbox"/>	变长字符串	20		枚举值: 在售、下架、售罄, 默认为在售	
9	INTRO	商品备注	<input type="checkbox"/>	<input type="checkbox"/>	变长字符串	900			

代码 L\_ORDER\_PRODUCT

显示名称 订单商品关联

2

备注说明

代码		名称	主键	不为空	基本数据类型	长度	小数点	备注	代码
1	ORDER_PRODUCT	订单商品关联ID	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	国际变长字符串	▼			
2	ORDER_ID	订单 ID	<input type="checkbox"/>	<input checked="" type="checkbox"/>	变长字符串	▼	32	外键, 关联订单表, 复合主键之一	
3	PRODUCT_ID	商品 ID	<input type="checkbox"/>	<input checked="" type="checkbox"/>	变长字符串	▼	32	外键, 关联商品表, 复合主键之一	
4	QUANTITY	购买数量	<input type="checkbox"/>	<input checked="" type="checkbox"/>	标准整数	▼	10		
5	UNIT_PRICE	购买单价	<input type="checkbox"/>	<input checked="" type="checkbox"/>	标准小数	▼	10	2	
6	SUBTOTAL_AMOUNT	商品小计金额	<input type="checkbox"/>	<input checked="" type="checkbox"/>	标准小数	▼	10	2 等于 QUANTITY×UNIT_PRICE, 大于 0	
7	REMARK	商品备注	<input type="checkbox"/>	<input type="checkbox"/>	变长字符串	▼	200		

代码 L\_RIDER

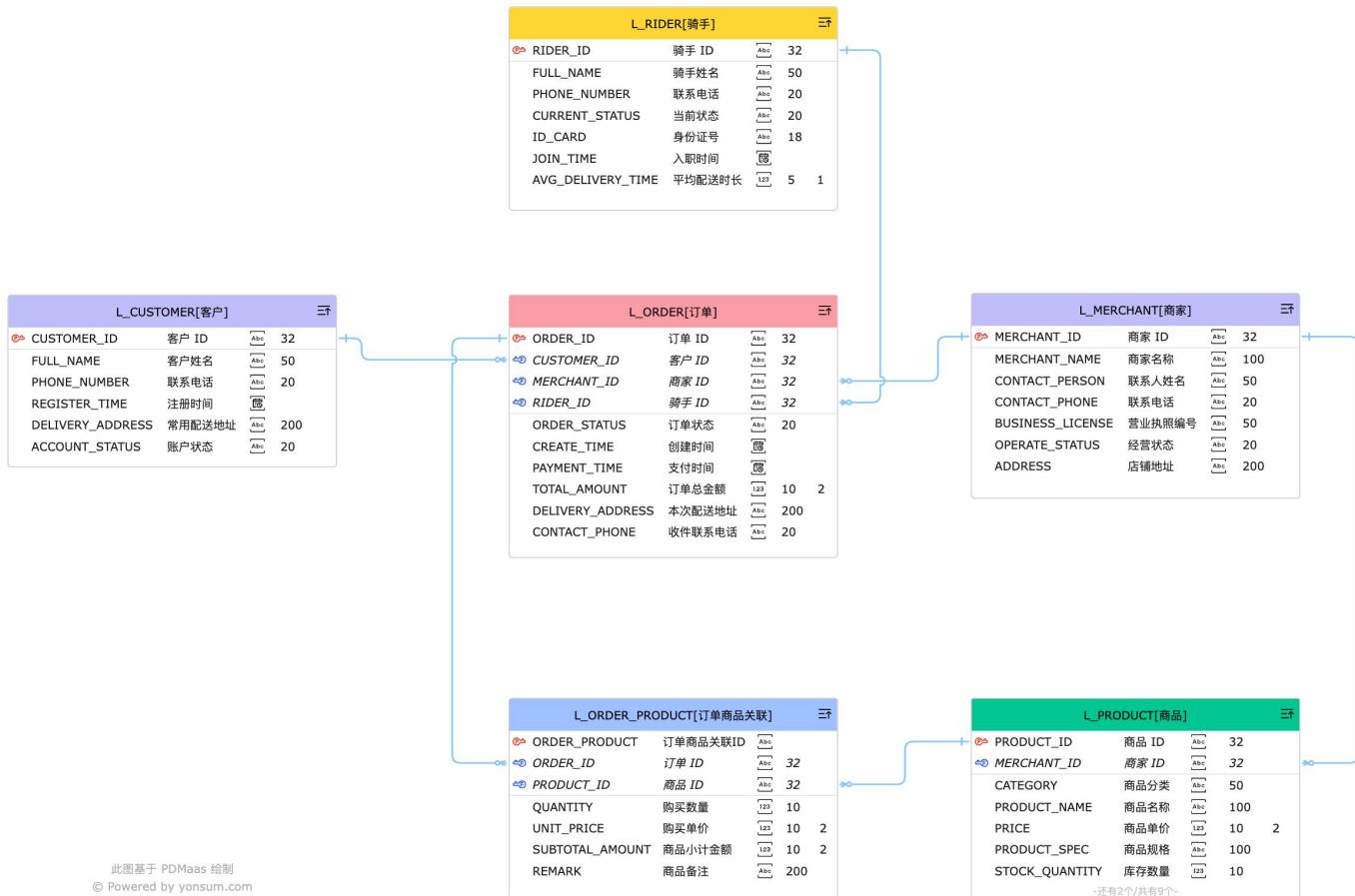
显示名称 骑手

2

备注说明

代码		名称	主键	不为空	基本数据类型	长度	小数点	备注	代码
1	RIDER_ID	骑手 ID	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	变长字符串	▼	32	唯一标识骑手	
2	FULL_NAME	骑手姓名	<input type="checkbox"/>	<input checked="" type="checkbox"/>	变长字符串	▼	50		
3	PHONE_NUMBER	联系电话	<input type="checkbox"/>	<input checked="" type="checkbox"/>	变长字符串	▼	20	需符合手机号格式, 唯一	
4	CURRENT_STATUS	当前状态	<input type="checkbox"/>	<input checked="" type="checkbox"/>	变长字符串	▼	20	枚举值: 在线、离线、忙碌, 默认为在线	
5	ID_CARD	身份证号	<input type="checkbox"/>	<input checked="" type="checkbox"/>	变长字符串	▼	18	需符合身份证格式, 唯一	
6	JOIN_TIME	入职时间	<input type="checkbox"/>	<input checked="" type="checkbox"/>	日期时间	▼			
7	AVG_DELIVERY_TIME	平均配送时长	<input type="checkbox"/>	<input type="checkbox"/>	标准小数	▼	5	1 单位: 分钟	

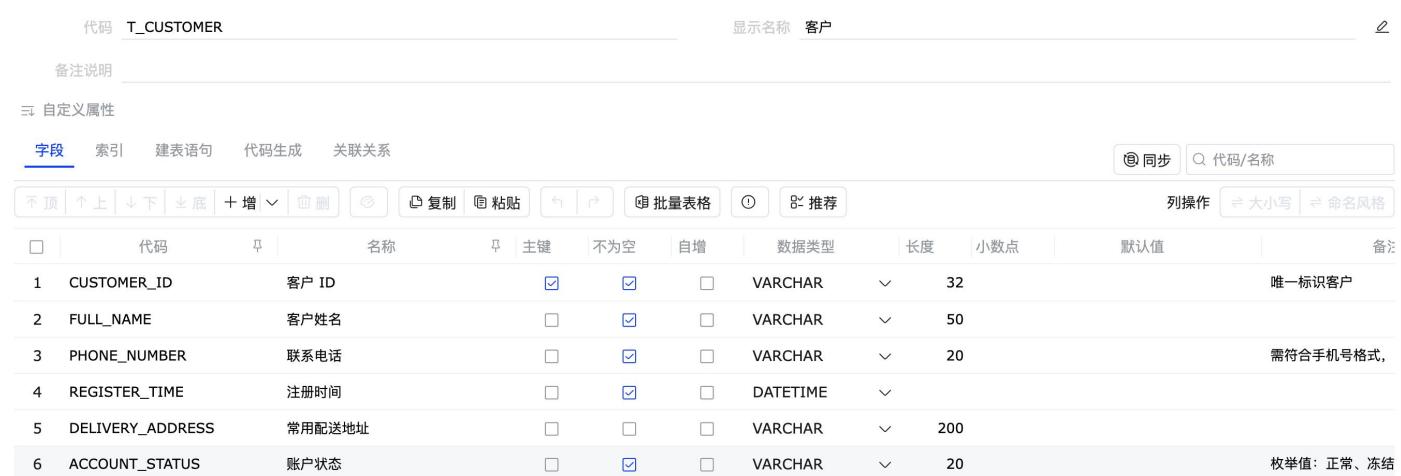
### 4.3.2 逻辑模型ER关系图及关联性设计



## 4.4 物理模型设计

### 4.4.1 逻辑模型转为物理模型

将逻辑模型转化为MySQL数据库的物理模型，转化后，产品自动做数据类型映射转换，成果如下图：



The screenshot shows the physical schema for the **T\_CUSTOMER** table:

字段	名称	主键	不为空	自增	数据类型	长度	小数点	默认值	备注
1 CUSTOMER_ID	客户 ID	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	VARCHAR	32			唯一标识客户
2 FULL_NAME	客户姓名	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	VARCHAR	50			
3 PHONE_NUMBER	联系电话	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	VARCHAR	20			需符合手机号格式,
4 REGISTER_TIME	注册时间	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	DATETIME				
5 DELIVERY_ADDRESS	常用配送地址	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	VARCHAR	200			
6 ACCOUNT_STATUS	账户状态	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	VARCHAR	20			枚举值：正常、冻结

代码 T\_ORDER

显示名称 订单

2

备注说明

自定义属性

存储引擎 InnoDB

▼

字段		索引	建表语句	代码生成	关联关系	操作		列操作	大小写	命名风格	备注
1	ORDER_ID	订单 ID		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	VARCHAR	32			唯一标识订单
2	CUSTOMER_ID	客户 ID		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	VARCHAR	32			外键, 关联客户表
3	MERCHANT_ID	商家 ID		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	VARCHAR	32			外键, 关联商家表
4	RIDER_ID	骑手 ID		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	VARCHAR	32			外键, 关联骑手表,
5	ORDER_STATUS	订单状态		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	VARCHAR	20			枚举值: 待支付, 已
6	CREATE_TIME	创建时间		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	DATETIME				CURRENT_TIMESTAMP
7	PAYMENT_TIME	支付时间		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	DATETIME				已支付时非空
8	TOTAL_AMOUNT	订单总金额		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	DECIMAL	10 2			大于 0
9	DELIVERY_ADDRESS	本次配送地址		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	VARCHAR	200			
10	CONTACT_PHONE	收件联系电话		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	VARCHAR	20			需符合手机号格式

代码 T\_MERCHANT

显示名称 商家

2

备注说明

自定义属性

字段		索引	建表语句	代码生成	关联关系	操作		列操作	大小写	命名风格	备注
1	MERCHANT_ID	商家 ID		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	VARCHAR	32			唯一标识商家
2	MERCHANT_NAME	商家名称		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	VARCHAR	100			
3	CONTACT_PERSON	联系人姓名		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	VARCHAR	50			
4	CONTACT_PHONE	联系电话		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	VARCHAR	20			需符合手机号格式
5	BUSINESS_LICENSE	营业执照编号		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	VARCHAR	50			唯一
6	OPERATE_STATUS	经营状态		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	VARCHAR	20			枚举值: 营业中、休
7	ADDRESS	店铺地址		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	VARCHAR	200			

代码 **T\_PRODUCT** 显示名称 **商品**

备注说明

自定义属性

字段 索引 建表语句 代码生成 关联关系

同步 代码/名称

列操作 大小写 命名风格

	代码	名称	主键	不为空	自增	数据类型	长度	小数点	默认值	备注
1	PRODUCT_ID	商品 ID	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	VARCHAR	32			唯一标识商品
2	MERCHANT_ID	商家 ID	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	VARCHAR	32			外键, 关联商家表
3	CATEGORY	商品分类	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	VARCHAR	50			
4	PRODUCT_NAME	商品名称	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	VARCHAR	100			
5	PRICE	商品单价	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	DECIMAL	10	2		
6	PRODUCT_SPEC	商品规格	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	VARCHAR	100			
7	STOCK_QUANTITY	库存数量	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	INT	10			
8	PRODUCT_STATUS	商品状态	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	VARCHAR	20			枚举值: 在售、下架
9	INTRO	商品备注	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	VARCHAR	900			

代码 **T\_ORDER\_PRODUCT** 显示名称 **订单商品关联**

备注说明

自定义属性

字段 索引 建表语句 代码生成 关联关系

同步 代码/名称

列操作 大小写 命名风格

	代码	名称	主键	不为空	自增	数据类型	长度	小数点	默认值	备注
1	ORDER_PRODUCT	订单商品关联ID	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NVARCHAR				
2	ORDER_ID	订单 ID	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	VARCHAR	32			外键, 关联订单表,
3	PRODUCT_ID	商品 ID	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	VARCHAR	32			外键, 关联商品表,
4	QUANTITY	购买数量	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	INT	10			
5	UNIT_PRICE	购买单价	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	DECIMAL	10	2		
6	SUBTOTAL_AMOUNT	商品小计金额	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	DECIMAL	10	2		等于 QUANTITY×UNIT_PRICE
7	REMARK	商品备注	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	VARCHAR	200			

代码 **T RIDER** 显示名称 **骑手**

备注说明

自定义属性

字段 索引 建表语句 代码生成 关联关系

同步 代码/名称

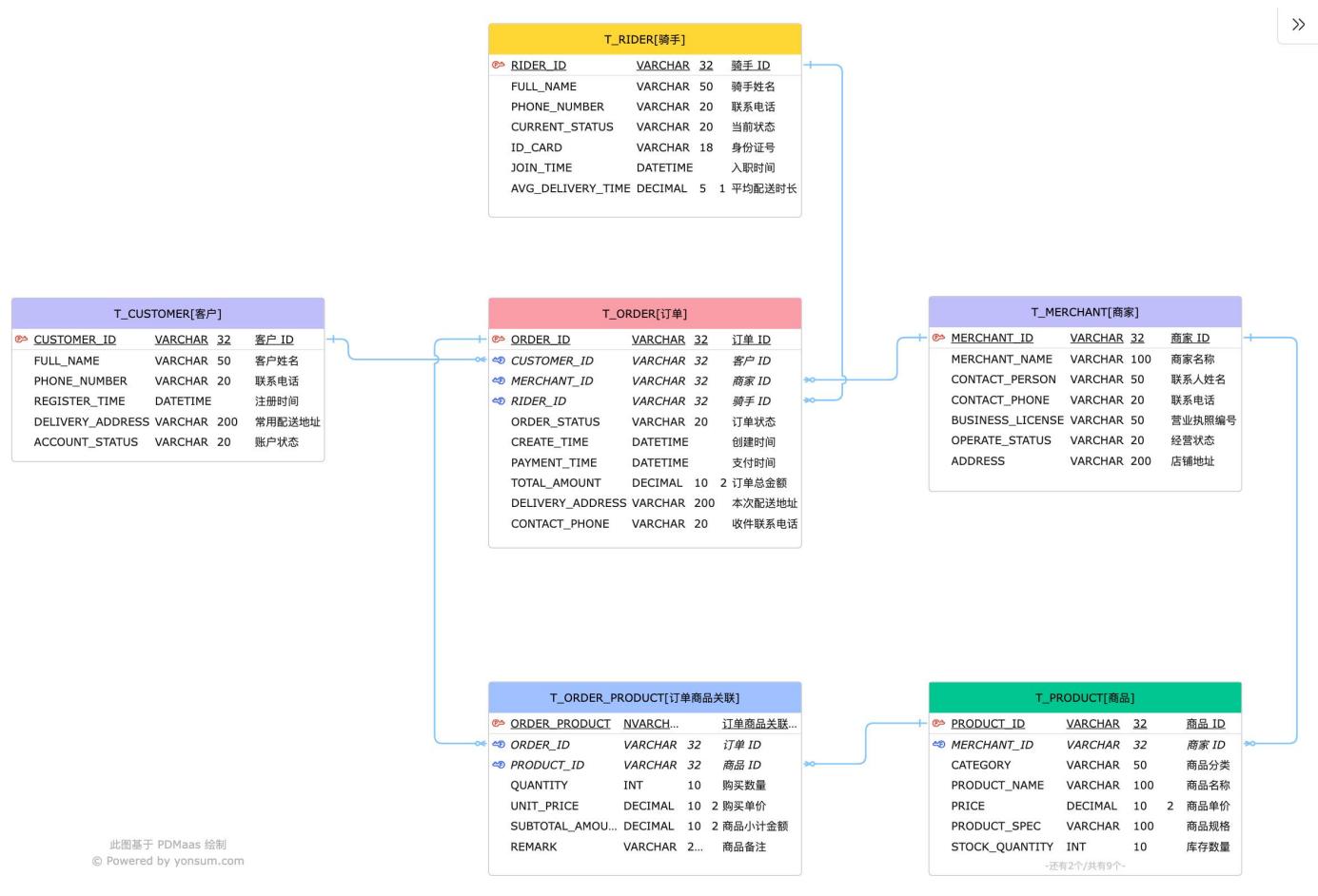
列操作 大小写 命名风格

	代码	名称	主键	不为空	自增	数据类型	长度	小数点	默认值	备注
1	RIDER_ID	骑手 ID	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	VARCHAR	32			唯一标识骑手
2	FULL_NAME	骑手姓名	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	VARCHAR	50			
3	PHONE_NUMBER	联系电话	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	VARCHAR	20			需符合手机号格式,
4	CURRENT_STATUS	当前状态	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	VARCHAR	20			枚举值: 在线、离线
5	ID_CARD	身份证号	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	VARCHAR	18			需符合身份证格式,
6	JOIN_TIME	入职时间	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	DATETIME				
7	AVG_DELIVERY_TIME	平均配送时长	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	DECIMAL	5	1		单位: 分钟

## 4.4.2 物理模型ER图

将逻辑模型转化为MySQL数据库的物理模型，转化后，自动生成物理模型ER图如下

图：



#### 4.4.3 索引策略设计

- **主键索引**：所有表的主键自动创建聚集索引
- **外键索引**：为提升关联查询性能，在外键字段上创建索引
- **业务查询索引**：在经常用于查询条件的字段上创建索引
  - 客户手机号索引：支持按手机号快速查找用户
  - 订单状态索引：支持按状态筛选订单

#### 4.4.4 生成DDL建表语句

通过工具软件自动生成数据库建表DDL（以MySQL为例），如下图：

导出DDL

全选 反选 全部展开/收起

搜索名称或代码

当前数据库: MySQL ▾

```

1  DROP TABLE IF EXISTS T_CUSTOMER;
2  CREATE TABLE T_CUSTOMER(
3      `CUSTOMER_ID` VARCHAR(32) NOT NULL COMMENT '客户 ID;唯一标识客户',
4      `FULL_NAME` VARCHAR(50) NOT NULL COMMENT '客户姓名',
5      `PHONE_NUMBER` VARCHAR(20) NOT NULL COMMENT '联系电话;需符合手机号格式, 唯一',
6      `REGISTER_TIME` DATETIME NOT NULL COMMENT '注册时间',
7      `DELIVERY_ADDRESS` VARCHAR(200) COMMENT '常用配送地址',
8      `ACCOUNT_STATUS` VARCHAR(20) NOT NULL COMMENT '账户状态;枚举值: 正常、冻结、注销, 默认为正常',
9      PRIMARY KEY (`CUSTOMER_ID`)
10 ) COMMENT '客户';
11
12
13
14 DROP TABLE IF EXISTS T_ORDER;
15 CREATE TABLE T_ORDER(
16     `ORDER_ID` VARCHAR(32) NOT NULL COMMENT '订单 ID;唯一标识订单',
17     `CUSTOMER_ID` VARCHAR(32) NOT NULL COMMENT '客户 ID;外键, 关联客户表',
18     `MERCHANT_ID` VARCHAR(32) NOT NULL COMMENT '商家 ID;外键, 关联商家表',
19     `RIDER_ID` VARCHAR(32) COMMENT '骑手 ID;外键, 关联骑手表, 配送前可空',
20     `ORDER_STATUS` VARCHAR(20) NOT NULL COMMENT '订单状态;枚举值: 待支付、已支付、已接单、配送中、已完成、已取消, 默认',
21     `CREATE_TIME` DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',
22     `PAYMENT_TIME` DATETIME COMMENT '支付时间;已支付时非空',
23     `TOTAL_AMOUNT` DECIMAL(10,2) NOT NULL COMMENT '订单总金额;大于 0',
24     `DELIVERY_ADDRESS` VARCHAR(200) COMMENT '本次配送地址',
25     `CONTACT_PHONE` VARCHAR(20) COMMENT '收件联系电话;需符合手机号格式',
26     PRIMARY KEY (`ORDER_ID`)
27 ) COMMENT '订单';
28
29
30 DROP TABLE IF EXISTS T_MERCHANT;
31 CREATE TABLE T_MERCHANT(
32     `MERCHANT_ID` VARCHAR(32) NOT NULL COMMENT '商家 ID;唯一标识商家',
33     `MERCHANT_NAME` VARCHAR(100) NOT NULL COMMENT '商家名称',
34     `CONTACT_PERSON` VARCHAR(50) NOT NULL COMMENT '联系人姓名',
35     `CONTACT_PHONE` VARCHAR(20) NOT NULL COMMENT '联系电话;需符合手机号格式',
36     `BUSINESS_LICENSE` VARCHAR(50) NOT NULL COMMENT '营业执照编号;唯一',
37     `OPERATE_STATUS` VARCHAR(20) NOT NULL COMMENT '经营状态;枚举值: 营业中、休息中、暂停营业, 默认为营业中',
38     `ADDRESS` VARCHAR(200) NOT NULL COMMENT '店铺地址',
39     PRIMARY KEY (`MERCHANT_ID`)
40 ) COMMENT '商家';
41
42
43 DROP TABLE IF EXISTS T_RIDER;
44 CREATE TABLE T_RIDER(
45     `RIDER_ID` VARCHAR(32) NOT NULL COMMENT '骑手 ID;唯一标识骑手',
46     `FULL_NAME` VARCHAR(50) NOT NULL COMMENT '骑手姓名',
47     `PHONE_NUMBER` VARCHAR(20) NOT NULL COMMENT '联系电话;需符合手机号格式, 唯一',
48     `CURRENT_STATUS` VARCHAR(20) NOT NULL COMMENT '当前状态;枚举值: 在线、离线、忙碌, 默认为在线',

```

建表  建索引  建外键

重新生成 下载脚本 复制脚本 执行脚本

## 4.5 模型验证与优化

### 4.5.1 业务场景验证

通过典型业务场景验证模型设计的合理性:

1. 用户下单场景: 验证订单表能否支持完整的下单流程
2. 商家接单场景: 验证状态流转和数据更新的逻辑
3. 配送跟踪场景: 验证订单状态管理和配送员分配

## 4. 数据分析场景：验证是否支持常见的业务分析需求

### 4.5.2 性能优化考虑

- **分表策略**：订单表按月分区，避免单表数据量过大
- **读写分离**：订单查询使用读库，减少对主库的压力
- **缓存策略**：热点商家和商品信息进行缓存

**模型演进**：数据模型不是一成不变的，需要根据业务发展和系统性能要求持续优化。建议建立定期的模型评审机制，及时发现和解决问题。

## 5. PDMaas建模工具应用

PDMaas（元数建模）是一款专业的数据建模工具，支持完整的三层建模方法论，提供可视化的建模界面和丰富的协作功能。本章将详细介绍如何使用PDMaas完成数据建模任务。

### 5.1 PDMaas建模工具概述

#### 5.1.1 工具特点

- **全面支持三层建模**：完整支持概念模型、逻辑模型、物理模型
- **可视化建模**：拖拽式ER图绘制，直观展示数据关系
- **多数据库支持**：支持MySQL、Oracle、PostgreSQL、Hive等主流数据库
- **团队协作**：支持多人实时协作，版本控制和冲突解决
- **标准管理**：内置数据标准管理，支持字典标准和基础标准
- **自动生成**：自动生成DDL语句、数据字典、设计文档

#### 5.1.2 系统架构

PDMaas采用BS架构，支持浏览器访问，主要包含以下模块：

- **管理系统**：团队管理、项目管理、用户权限管理
- **建模系统**：模型设计、关系图绘制、数据标准贯标
- **标准系统**：数据标准制定和管理
- **协作系统**：版本控制、多人协作、操作历史

## 5.2 环境准备与登录

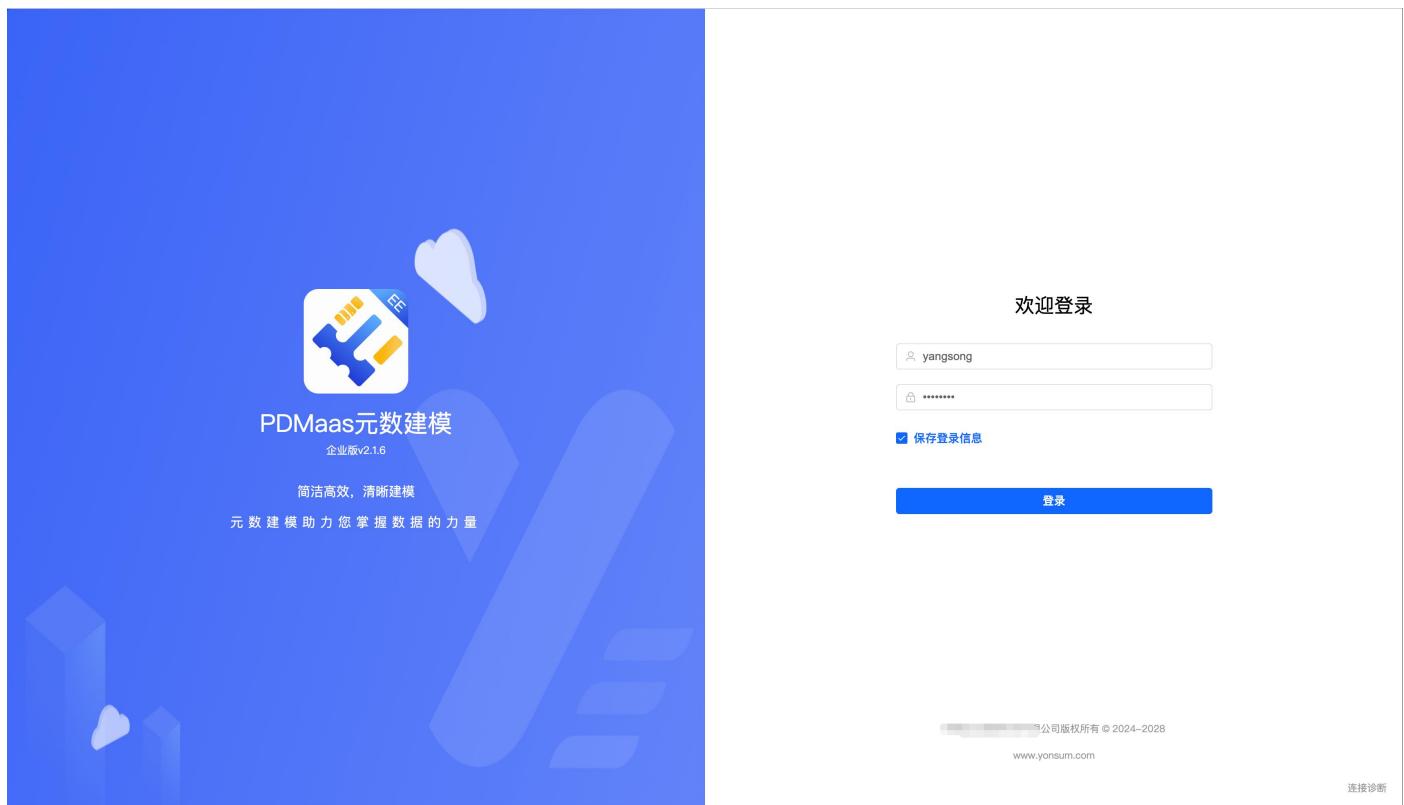
### 5.2.1 系统登录

默认账户信息：

- 超级管理员：yangsong / Y202403\$

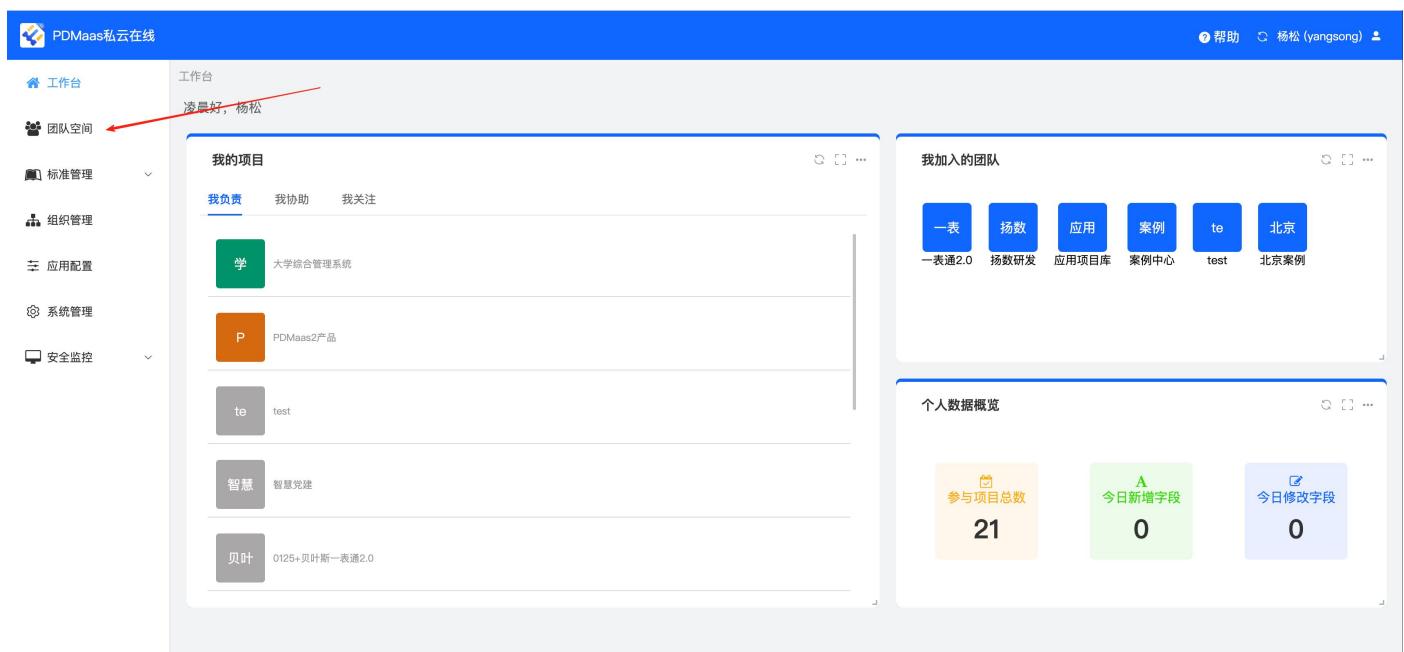
■ 团队管理员：liuqiang / Y202403\$

■ 标准维护员：shibin / Y202403\$



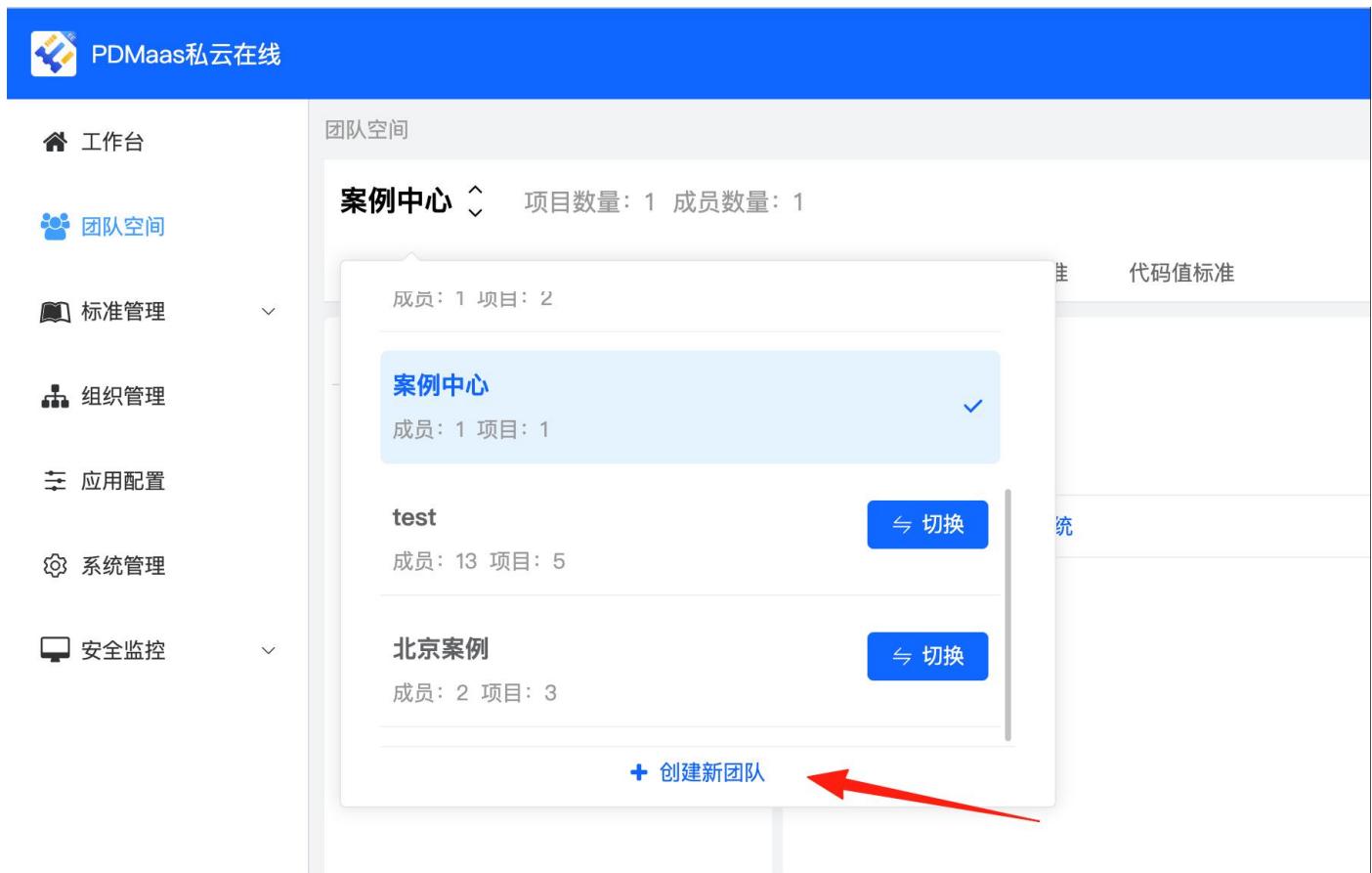
## 5.2.2 创建团队和项目

进入系统后首先进入团队空间，如下图：



首次使用需要完成以下配置：

## 1. 创建团队：在团队空间中创建工作团队



The screenshot shows the PD Maas private cloud interface. On the left, there is a sidebar with various management options: Workbench, Team Space (selected), Standard Management, Organization Management, Application Configuration, System Management, and Security Monitoring. The main area is titled 'Case Center' and displays four team spaces: 'Case Center' (selected, with 1 member and 1 project), 'test' (with 13 members and 5 projects), and 'Beijing Case' (with 2 members and 3 projects). At the bottom of the list, there is a button labeled '+ Create New Team' with a red arrow pointing to it.

## 2. 添加成员：邀请团队成员并分配角色

待选用户

已选用户 (2个)

搜索框: 登录名/用户名

选择机构: 选择机构

已选用户列表 (2个):

- 杨松 部门: <空> 用户ID: Y1001 | 登录名: yangsong
- 体验-领导 部门: <空> 用户ID: X7PHYVNV7D00 | 登录名: pdmaasLeader

加载中...

操作按钮: 选入 (指向杨松) 放回

待选用户列表 (8个):

- 重置 (用户ID: X7GS0MD7D00 | 登录名: dongliusuo | 部门: 扬数起元) 选入
- 体验-领导 (用户ID: X7PHYVNV7D00 | 登录名: pdmaasLeader | 部门: 外部体验) 已选
- 体验-开发 (用户ID: X7PJNY687D00 | 登录名: pdmaasDeveloper | 部门: 外部体验) 选入 (指向)
- 体验-开发2 (用户ID: X7PJSDK47D00 | 登录名: pdmaasDeveloper2 | 部门: 外部体验) 选入
- 体验-观察员 (用户ID: X7PJU51D7D00 | 登录名: pdmaasViewer | 部门: 外部体验) 选入
- 杨松 (用户ID: Y1001 | 登录名: yangsong | 部门: 扬数起元) 已选
- 刘强 (用户ID: Y1002 | 登录名: liuqiang | 部门: 扬数起元) 选入
- 时斌 (用户ID: Y1005 | 登录名: shibin | 部门: 扬数起元) 选入

取消 确定

### 3. 配置环境: 设置数据库连接信息 (可选)

北京案例 项目数量: 3 成员数量: 2

修改

项目管理 协作成员 数据库环境 业务域类型 数据项标准 代码值标准

成员名单

全部成员 2

全部成员

搜索框: 搜索昵称 / 用户名称

添加成员

杨松 登录名: yangsong 角色: 暂无角色

体验-领导 登录名: pdmaasLeader 角色: 团队负责人

操作按钮: 移除 (指向杨松) 移除 (指向体验-领导)

角色下成员

需求分析师

模型开发设计员

模型查看员

模型评审观察员

模型版本管理员

数据库管理员

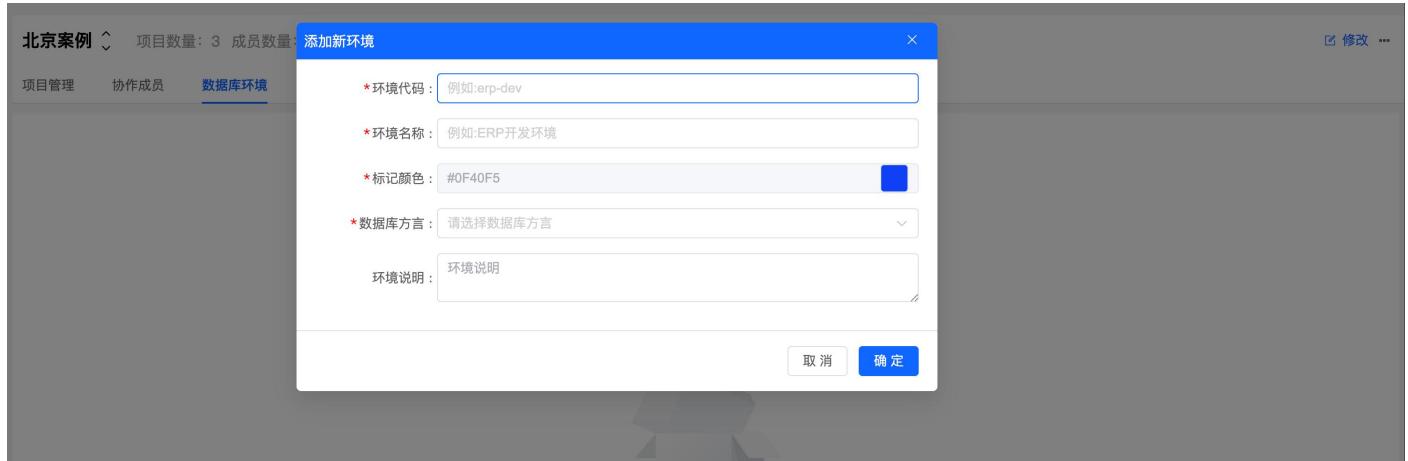
项目管理员

团队负责人

北京案例 ▾ 项目数量: 3 成员数量: 2

[修改 ...](#)项目管理 协作成员 **数据库环境** 业务域类型 数据项标准 代码值标准

当前环境为空, 点击下方按钮进行创建环境

[创建环境](#)

4. 创建并打开项目：新建数据建模项目，选择数据库类型

团队空间

北京案例 项目数量: 3 成员数量: 3

项目管理 协作成员 数据库环境

团队分类目录 + ⌂ ⌂ ⌂

所有项目 我负责 我协助 我关注

新建项目

创建新项目

项目名称: 外卖平台案例

团队目录: 团队目录

头像文字: 头像文字 (两个字以内)

头像颜色: 头像颜色

分支模型: 单分支模型 (只创建master分支)

数据库类型: PostgreSQL

项目说明: 项目说明

取消 确定

负责人: 杨松 更新时间: 2025-07-23 07:31:04 操作

杨松 2025-07-03 22:58:19 操作

杨松 2025-07-03 09:53:22 操作

北京案例 项目数量: 3 成员数量: 2

项目管理 协作成员 数据库环境 业务域类型 数据项标准 代码值标准

团队分类目录 + ⌂ ⌂ ⌂

所有项目 我负责 我协助 我关注

新建项目

所有项目

项目名称: 外卖 并指向 外卖平台案例

负责人: 杨松 更新时间: 2025-08-14 13:18:59 操作

智慧 智慧党建

杨松 2025-07-23 07:31:04 操作

集团 集团资产运营平台

杨松 2025-07-03 22:58:19 操作

产品 产品销售维度建模

杨松 2025-07-03 09:53:22 操作

欢迎您

PD Maas 元数建模 企业版 v2.1.6

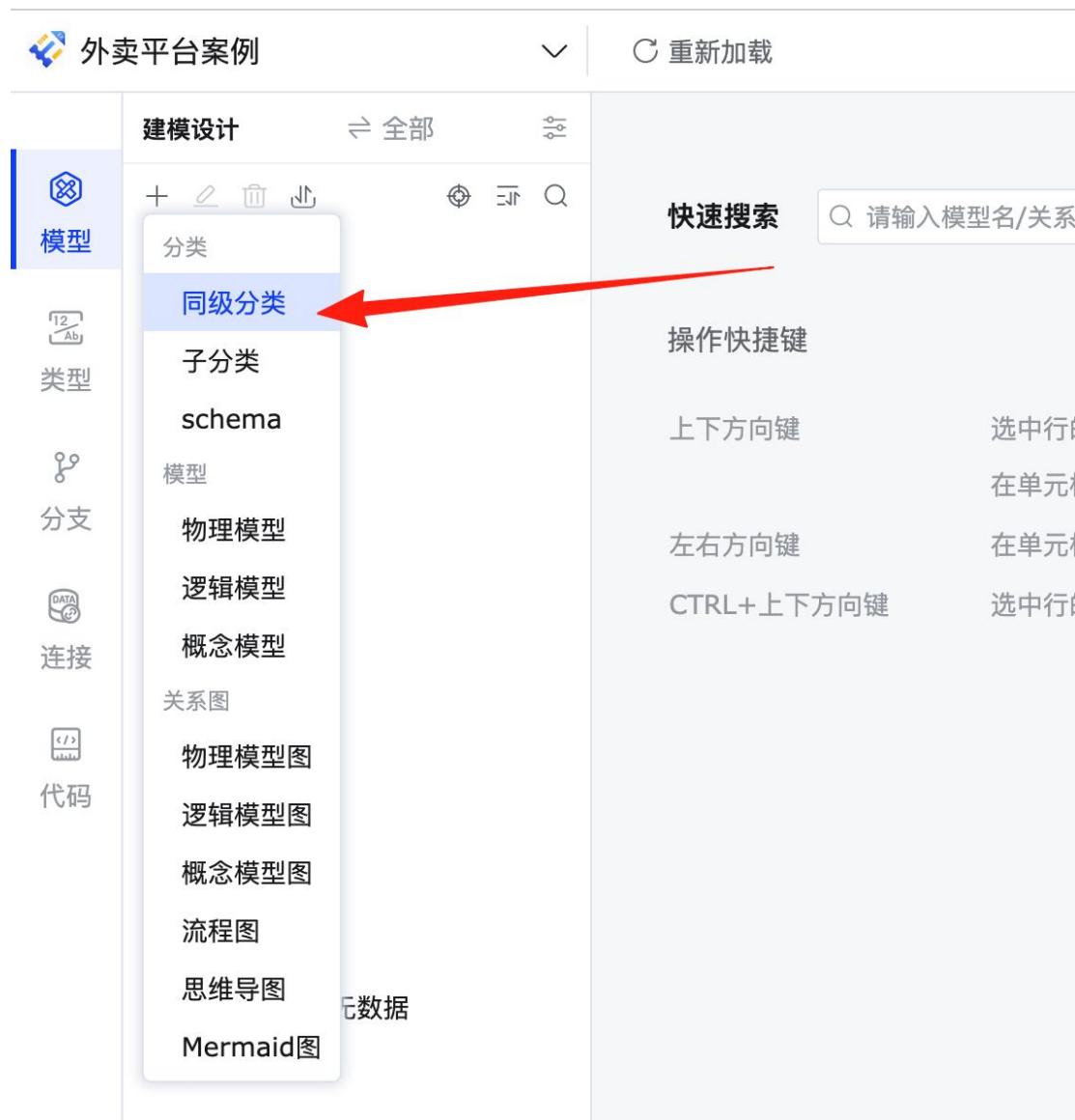
数据发送中..... 00:00:00 86.36%

14. 插件[wordRoot]加载完成  
15. 项目-插件加载完成  
16. 读取最新操作位置开始  
17. 读取最新操作位置结束  
18. 处理排序  
19. 数据发送中.....

有限公司版权所有 © 2024-2028

### 5.3 模型资源分类

进行建模前，需要对建模项目的分类目录进行规划，规划完成后，在建模平台上完成相关资源分类，新建分类操作如下：



外卖平台案例

建模设计 全部

模型

类型

分支

连接

代码

分类

同级分类

子分类

schema

模型

物理模型

逻辑模型

概念模型

关系图

物理模型图

逻辑模型图

概念模型图

流程图

思维导图

Mermaid图

重新加载

快速搜索

请输入模型名/关系

操作快捷键

上下方向键 选中行

在单元格

左右方向键 选中行

在单元格

CTRL+上下方向键 选中行

在单元格

新建分类

\* 节点类型  分类  数据库schema

参考节点  ▼

\* 分类代码  4/20  我不想填, 自动生成 转大写 | 转小写

分类名称  4/20  代码作为名称

\* 对象类型分类  自动分类  手动选择  不分类

取消 保存并继续新建 确认

## 5.4 概念建模操作

### 5.4.1 创建概念模型

1. 创建概念模型画布：右击“关系图”节点，选择“新建-概念模型图”

外卖平台案例

重新加载 最近保存: 13:27:14

模型

类型

分支

连接

代码

建模设计 全部

快速搜索

操作快捷键

上下方向键

左右方向键

CTRL+上下方向键

选中行的情况下, 切换至下一单元格  
在单元格编辑情况下, 则是  
在单元格编辑情况下, 如果  
选中行的情况下, 从当前行

关系图 (0)

新建

新建-概念模型图

新建-逻辑模型图

新建-物理模型图

新建-流程图

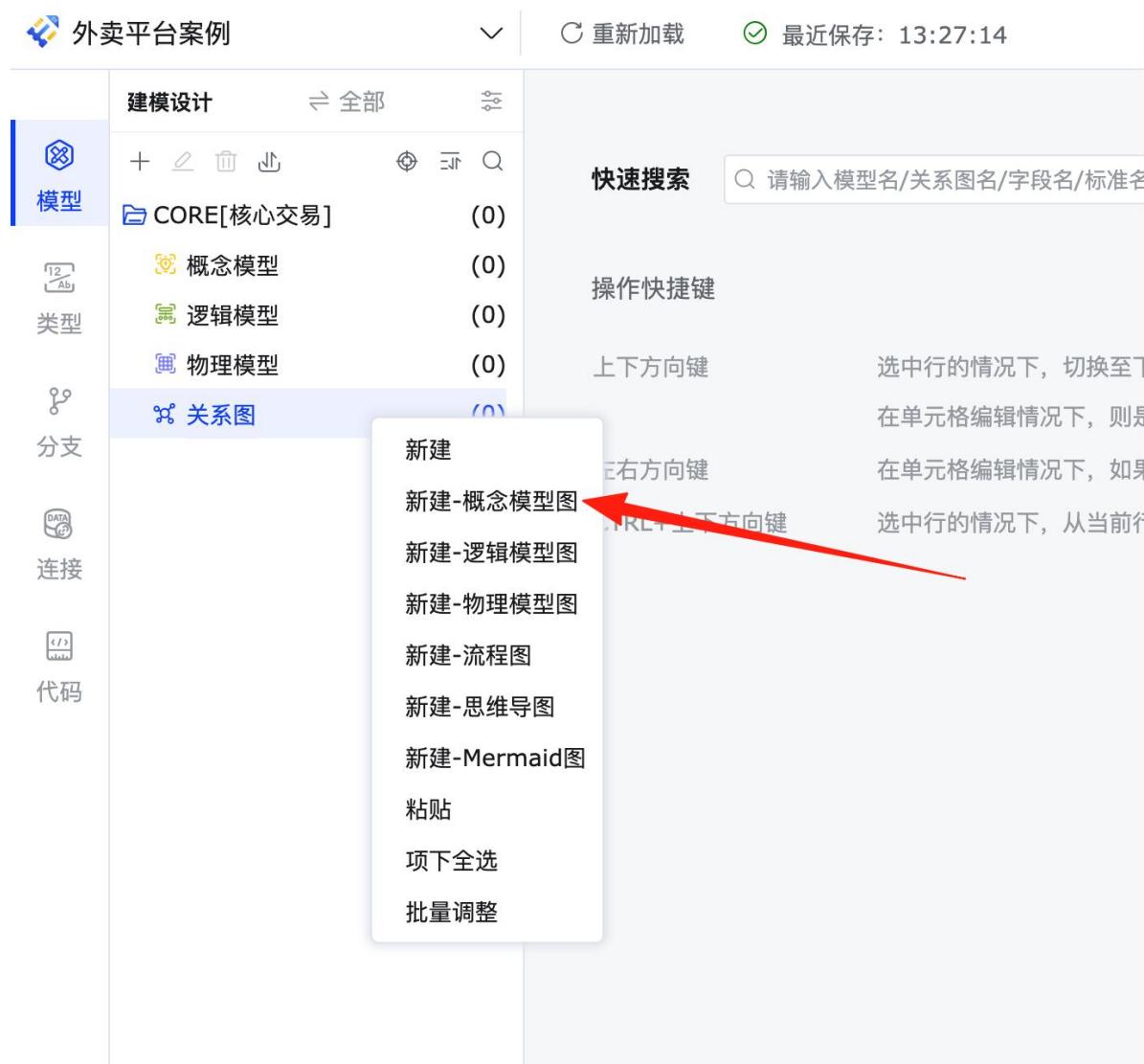
新建-思维导图

新建-Mermaid图

粘贴

项下全选

批量调整



新建概念模型图

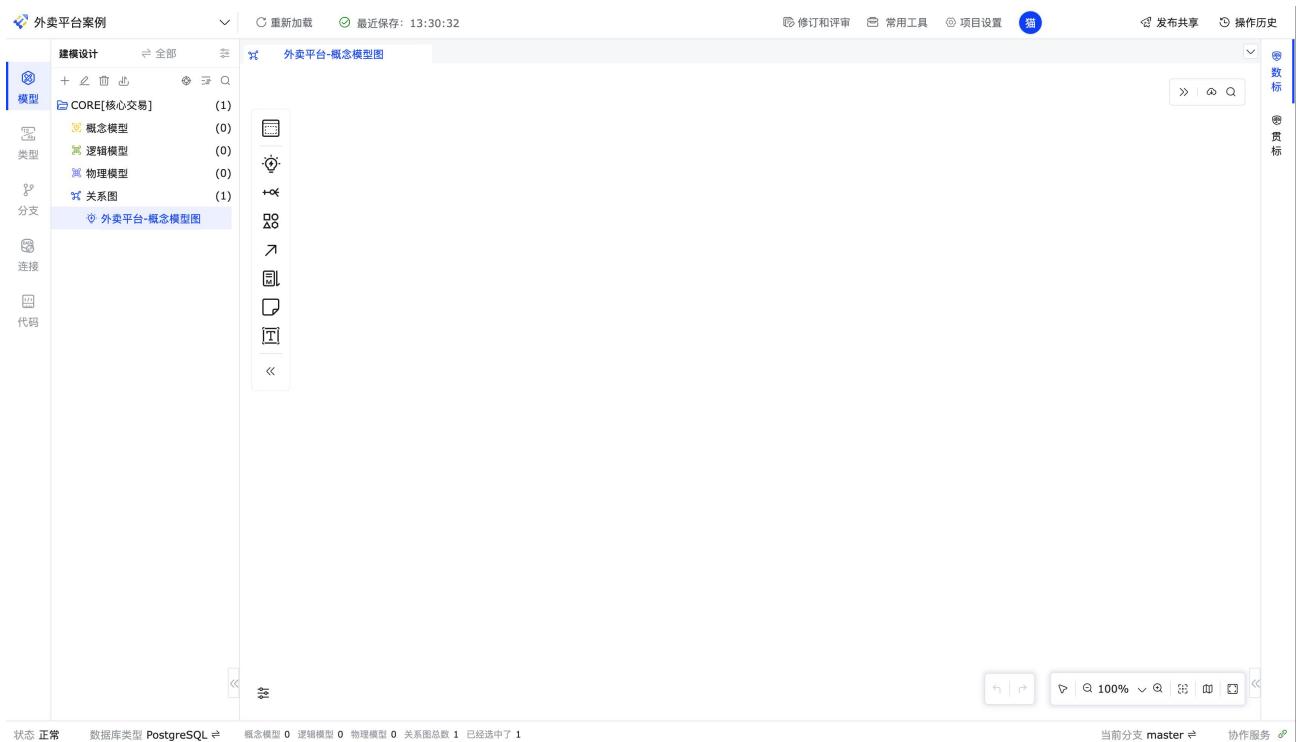
归属分类 核心交易

\* 绘图类型 概念模型图

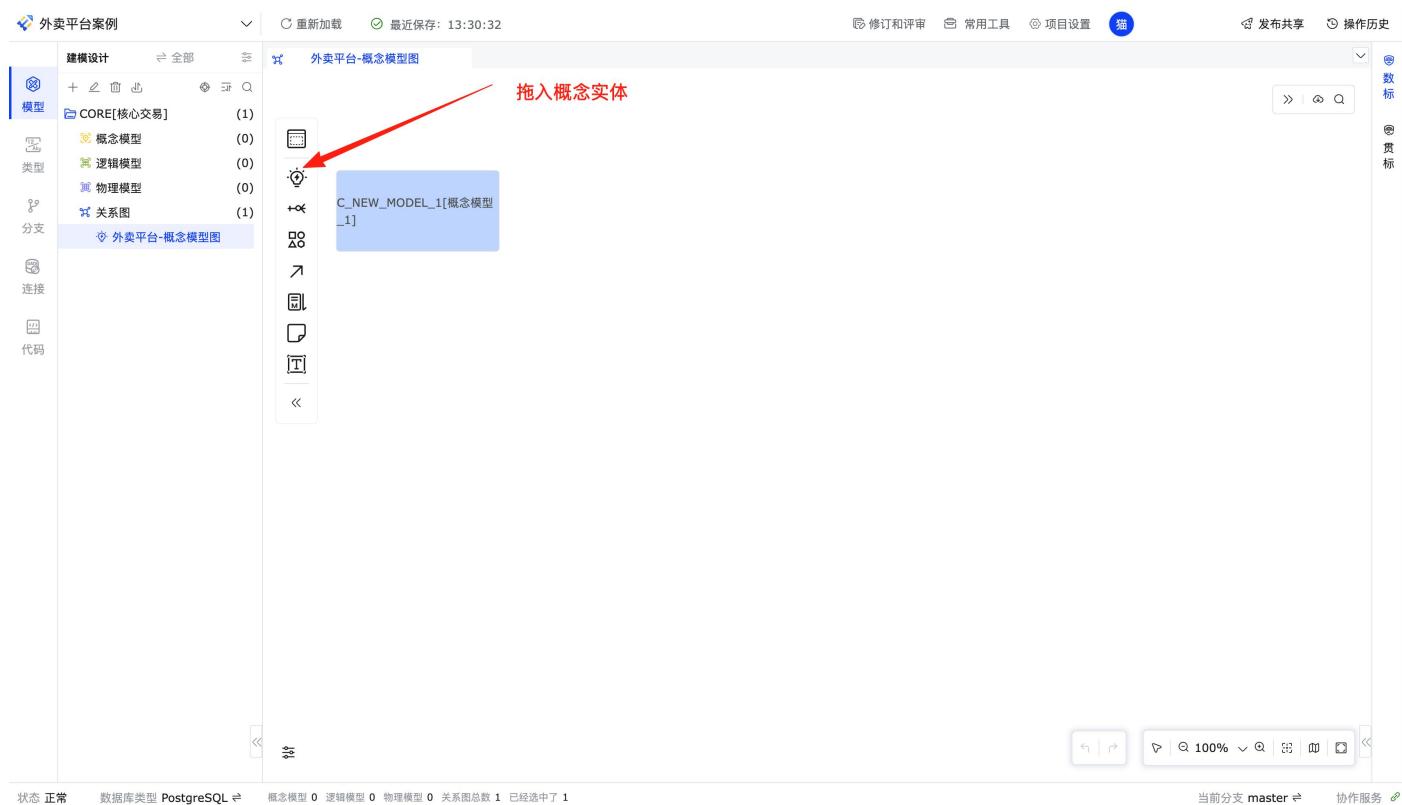
\* 代码 外卖平台-概念模型图 10/64 转大写 | 转小写

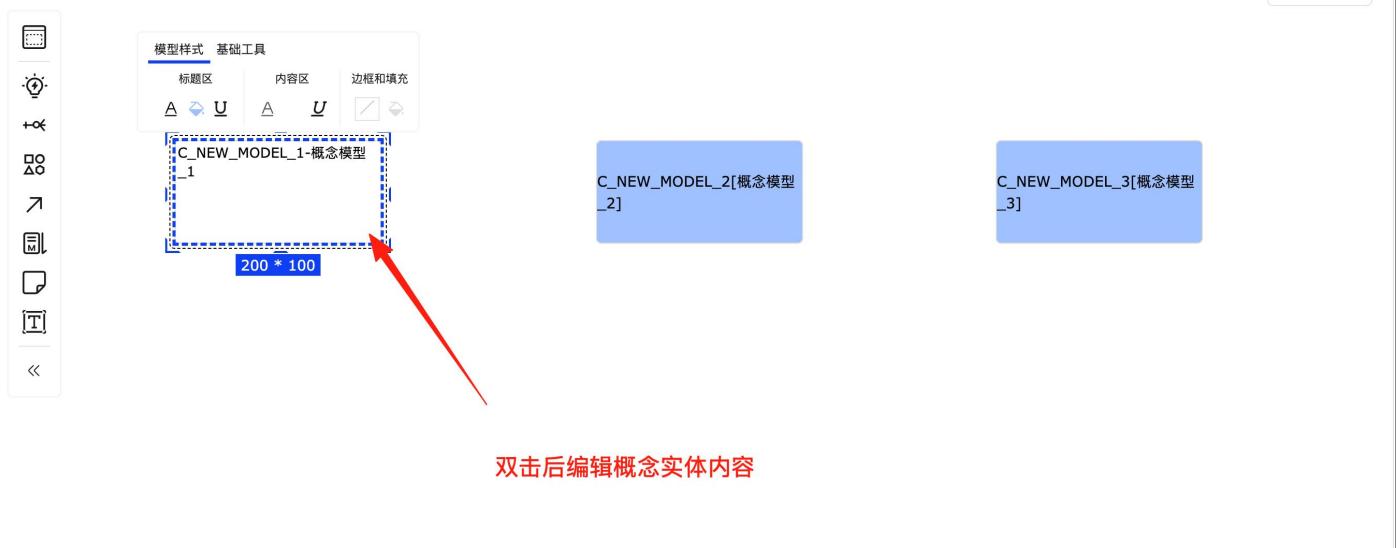
显示名称 外卖平台-概念模型图 10/64  代码作为名称

取消 保存并继续新建 确认

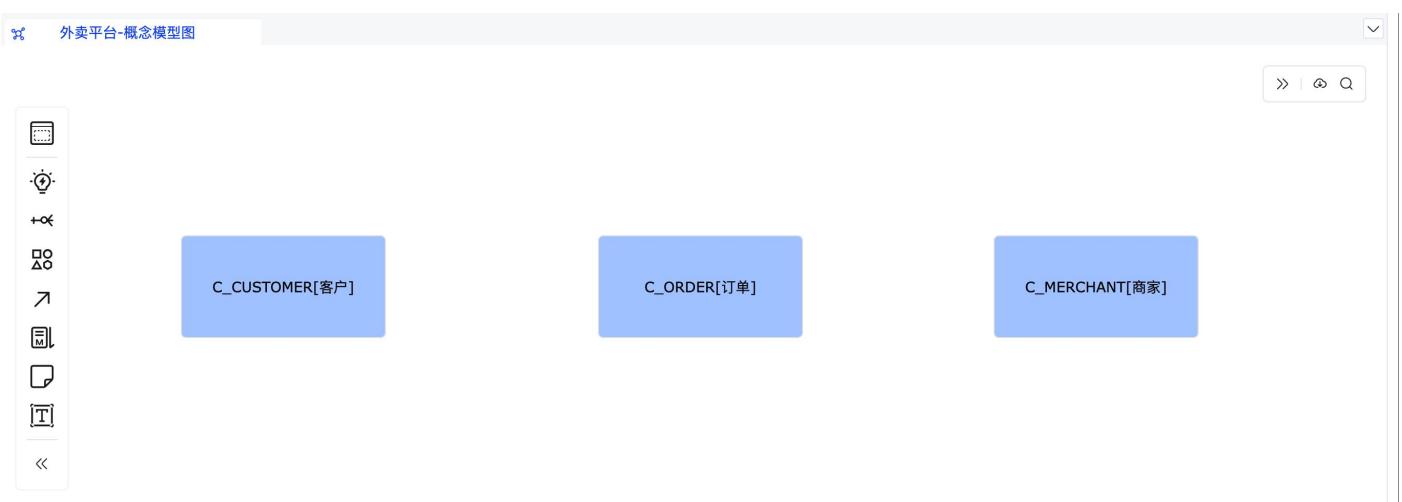


## 2. 新建概念实体：识别业务对象，创建对应概念实体

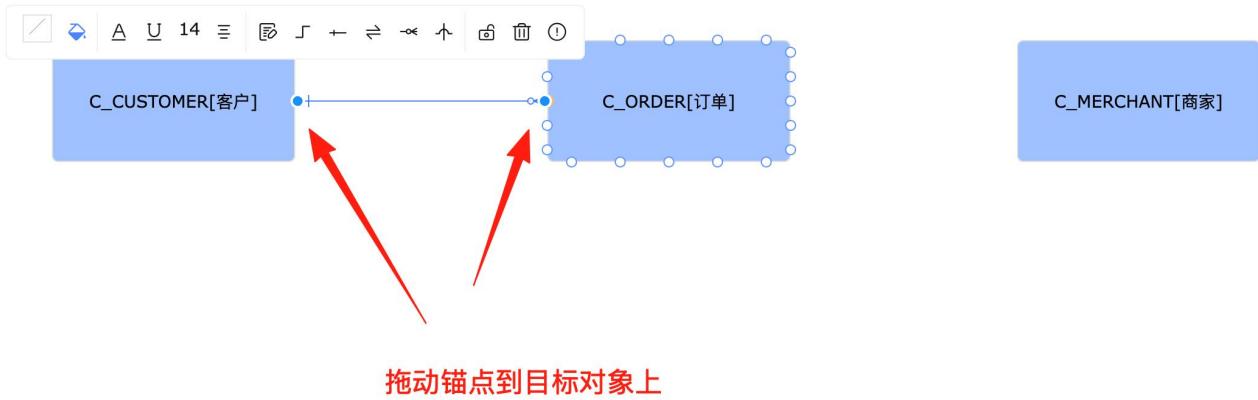
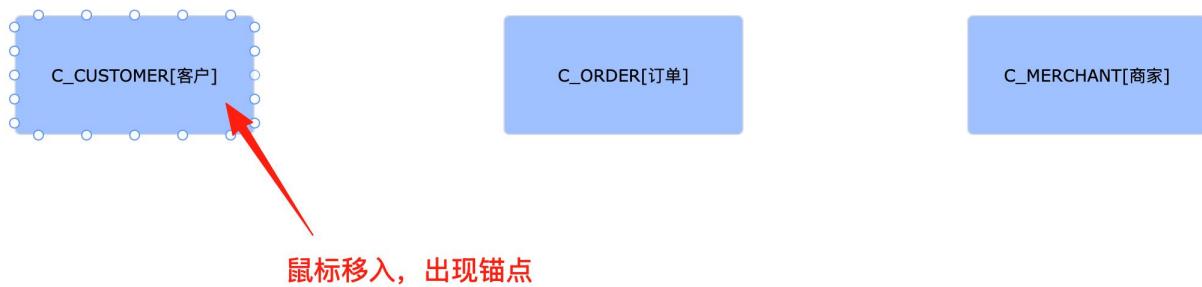


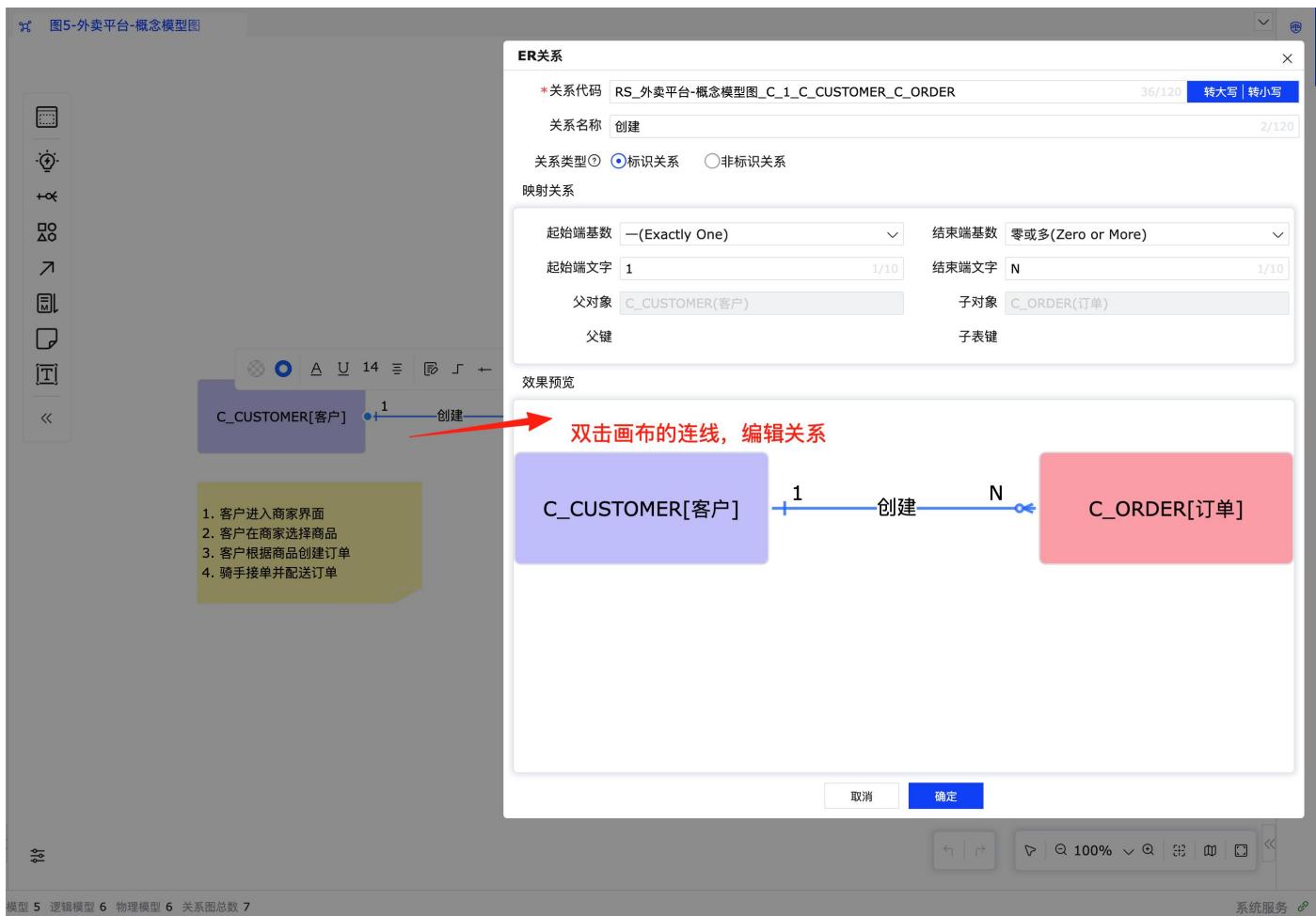


### 3. 完善实体描述：为实体和关系添加业务描述



### 4. 建立实体关联：建立实体间的关联关系





## 5.4.2 概念模型设计要点

- 实体命名：使用业务友好的名称，便于业务人员理解
- 关系表达：清晰表达实体间的业务关联

## 5.5 逻辑建模操作

### 5.5.1 创建逻辑模型

可以从概念模型转换，也可以直接创建：

1. 模型转换：将概念模型转换为逻辑模型字段设计：为每个实体添加具体字段

概念模型	(5)	
逻辑模型	(6)	
物理模型	(6)	
关系图	(7)	
图2-数据建模的地位		
图1-什么是数据模型		
图4-数据建模三阶段		
图5-外卖平台-概念模型图		

图5-外卖平台-概念模型图

图6-外卖平台-逻辑模型图

图7-外卖平台-物理模型图

图3-五种建模的对比

编辑

新建

新建-概念模型图

新建-逻辑模型图

新建-物理模型图

新建-流程图

新建-思维导图

新建-Mermaid图

复制

粘贴

创建副本

分类

转逻辑

转物理

设为首页封面

删除

1.  
2.  
3.  
4.



2. 直接创建：通过右击逻辑模型节点后弹出快捷菜单，然后录入概念模型以及概念模型的属性信息



新建逻辑模型

快速录入 例如: 学生(姓名, 性别, 年龄, 身高, 体重)

归属分类 CORE

\*代码 L\_CUSTOMER 10/64 转大写 | 转小写 显示名称 客户逻辑模型 6/64

备注说明

列操作 大小写 命名风格

代码 名称 主键 不为空 基本数据类型 长度 小数点 备注

暂无数据

取消 保存并继续新建 确认

### 3. 类型定义: 设置字段的数据类型和长度, 主键、外键、非空等

快速录入

客户(客户 ID,客户姓名,联系电话,注册时间,常用配送地址,账户状态)

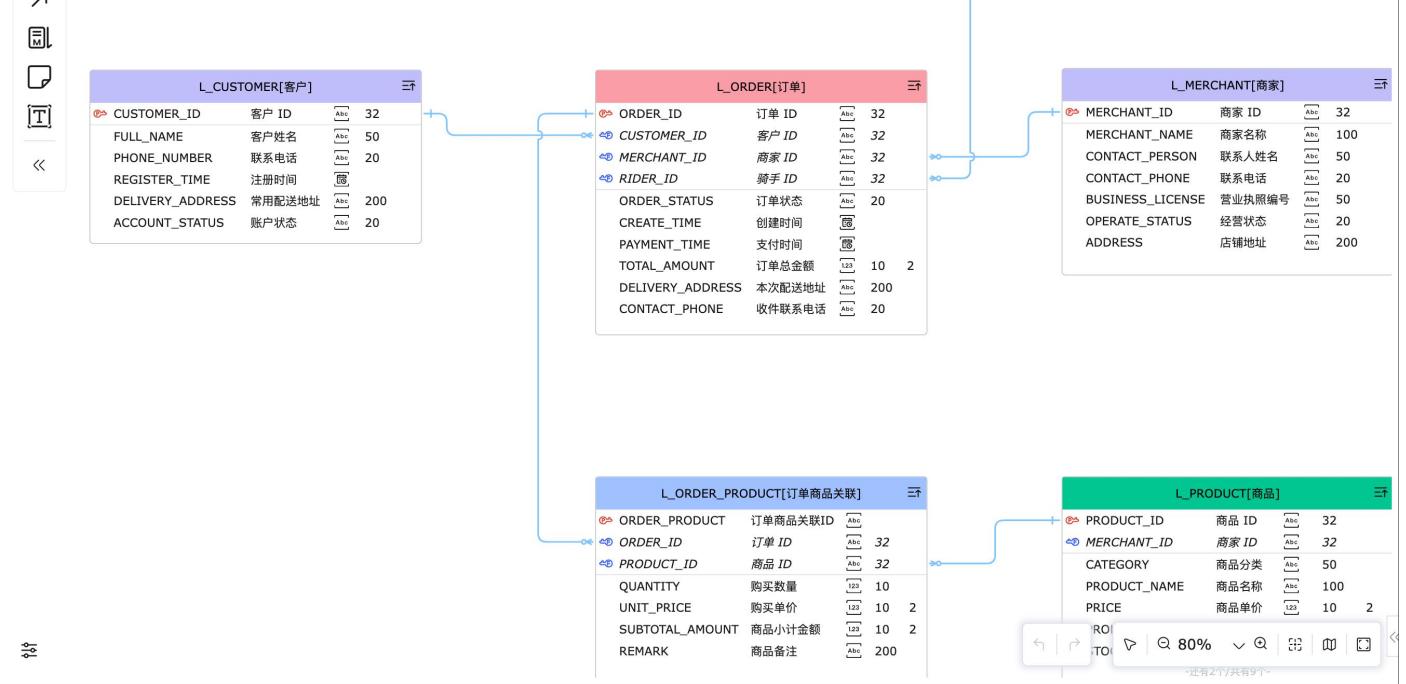
代码 L\_CUSTOMER 显示名称 客户

备注说明

列操作 大小写 命名风格

序号	代码	名称	主键	不为空	基本数据类型	长度	小数点	备注	代码
1	CUSTOMER_ID	客户 ID	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	变长字符串	32		唯一标识客户	
2	FULL_NAME	客户姓名	<input type="checkbox"/>	<input checked="" type="checkbox"/>	变长字符串	50			
3	PHONE_NUMBER	联系电话	<input type="checkbox"/>	<input checked="" type="checkbox"/>	变长字符串	20		需符合手机号格式, 唯一	
4	REGISTER_TIME	注册时间	<input type="checkbox"/>	<input checked="" type="checkbox"/>	日期时间				
5	DELIVERY_ADDRESS	常用配送地址	<input type="checkbox"/>	<input type="checkbox"/>	变长字符串	200			
6	ACCOUNT_STATUS	账户状态	<input type="checkbox"/>	<input checked="" type="checkbox"/>	变长字符串	20		枚举值: 正常、冻结、注销, 默认为正常	

### 4. 关联设置: 建立表间的关联关系 (画布操作方法与概念模型类似)



## 5.5.2 字段操作技巧

### ■ 快速录入功能

PDMaas 提供快速录入功能，可以通过简单语法快速创建字段：

### ■ 标准录入功能

1. 可以从标准库中选择标准字段，确保字段定义的一致性：
2. 打开标准录入开关
3. 从右侧标准库选择字段
4. 拖拽到当前模型中



外卖平台-概念模型图 L\_CUSTOMER[客户逻辑模型]

快速录入

客户逻辑模型()

代码 L\_CUSTOMER 显示名称 客户逻辑模型

备注说明

代码 名称 主键 不为空 基本数据类型 长度 小数点

暂无数据

数据项标准 代码值标准

全局 (167)

- 参与者 (79)
- 机构 (16)
- 员工 (15)
- 基本信息 (9)
- 系统用户号 [Abc] 32 -
- 员工号 [Abc] 32 -
- 员工姓名 [Abc] 60 -
- 员工证件类型 [Abc] 32 -
- 员工证件号码 [Abc] 32 -
- 员工所属部门编码 [Abc] 32 -
- 用工类别 [Abc] 32 -
- 员工状态 [Abc] 32 -
- 来行时间 [Abc] -
- 人力资源 (6)
- 商户 (6)
- 基本信息 (6)
- 商户编号 [Abc] 32 -
- 商户名称 [Abc] 32 -
- 商户类别码 [Abc] 32 -
- 商户类别码名称 [Abc] 90 -
- 商户地区 [Abc] 32 -
- 发卡行手续费率 [123] 10 6
- 客户通用 (2)
- 个人客户 (18)
- 对公客户 (22)
- 产品 (51)

概念模型 3 逻辑模型 1 物理模型 0 关系图总数 1 已经选中了 1

当前分支 master 协作服务

客户逻辑模型()

代码 L\_CUSTOMER

显示名称 客户逻辑模型

备注说明

不顶 个上 下底 + 增 | 复制 粘贴 | ① | 批量表格 | 推荐 列操作 | 大小写 命名

□ 代码 名称 主键 不为空 基本数据类型 长度 小数点

1 column\_0

编号

自动推荐

数标推荐

- 全局 (6)
- 参与人 (3)
- 员工 (2)
- 人力资源 (2)
- 岗位 编号 (POST\_ID)
- 客户经理 编号 (CUST\_MAN\_ID)
- 商户 (1)
- 基本信息 (1)
- 商户 编号 (MCC\_ID)
- 合约 (1)
- 左键关系 (1)

### 5.5.3 批量操作功能

- 批量添加字段：一次性添加多个字段
- 批量删除字段：删除不需要的字段
- 批量格式转换：统一字段命名风格

## 5.6 物理建模操作

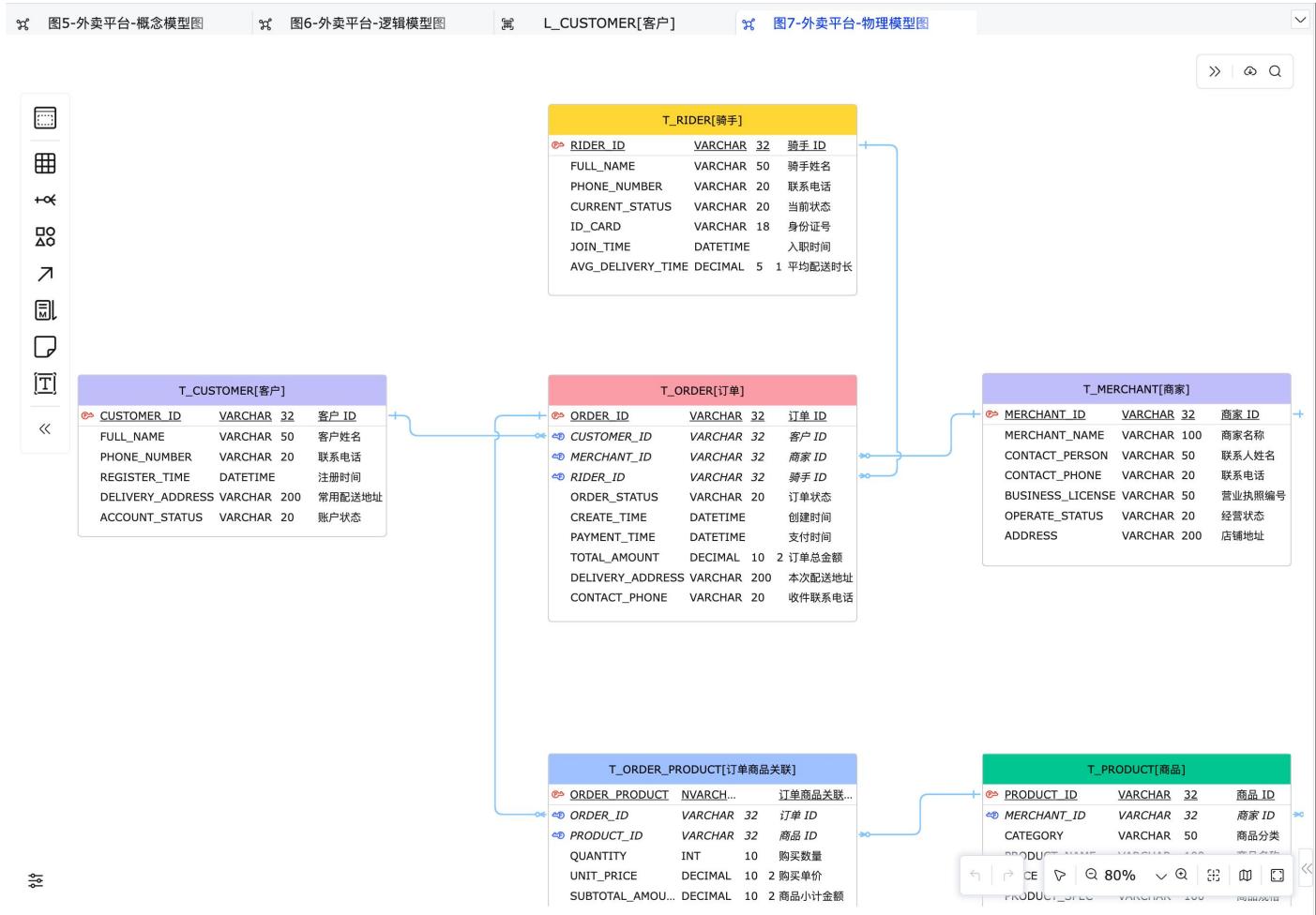
### 5.6.1 创建物理模型

1. 模型转换：从逻辑模型转换为物理模型（方法与概念模型转逻辑模型一

致)

2. 录入模型: 通过手动的形式录入数据表模型 (方法与创建逻辑模型一致)

3. 关联关系: 通过转换的方式生成ER关系图或者手动维护ER关系图



## 5.6.2 索引管理

在物理模型的索引标签页中进行索引管理:

新增索引: 为单个或多个字段创建索引

索引类型: 选择普通索引、唯一索引、全文索引

索引名称: 遵循命名规范, 便于维护

代码 **T\_CUSTOMER** 显示名称 **客户**

备注说明

自定义属性

字段 索引 建表语句 代码生成 关联关系

↑ 顶 ↓ 上 ↓ 下 ↓ 底 + 增 ↴ | 🔁

索引类型	索引代码	字段	索引名称	备注
1 一般索引	idx_customer_1	PHONE_NUMBER		

字段	排序顺序
1 PHONE_NUMBER	✓

+ 添加 - 移除 取消 确认

## 5.6.3 DDL生成与管理

PDMaas 自动生成建表语句：

1. 多数据库支持：可生成不同数据库的DDL

代码 **T\_CUSTOMER** 显示名称 **客户**

备注说明

自定义属性

字段 索引 建表语句 代码生成 关联关系

MySQL | 请选择 | 结构数据 模板调整 执行脚本

```

1 DROP TABLE IF EXISTS T_CUSTOMER;
2 CREATE TABLE T_CUSTOMER(
3     `CUSTOMER_ID` VARCHAR(32) NOT NULL COMMENT '客户 ID;唯一标识客户',
4     `FULL_NAME` VARCHAR(50) NOT NULL COMMENT '客户姓名',
5     `PHONE_NUMBER` VARCHAR(20) NOT NULL COMMENT '联系电话;需符合手机号格式, 唯一',
6     `REGISTER_TIME` DATETIME NOT NULL COMMENT '注册时间',
7     `DELIVERY_ADDRESS` VARCHAR(200) COMMENT '常用配送地址',
8     `ACCOUNT_STATUS` VARCHAR(20) NOT NULL COMMENT '账户状态;枚举值: 正常、冻结、注销, 默认为正常',
9     PRIMARY KEY (`CUSTOMER_ID`)
10 ) COMMENT '客户';
11
12
13

```

2. 模板定制：可调整生成模板，满足特定需求

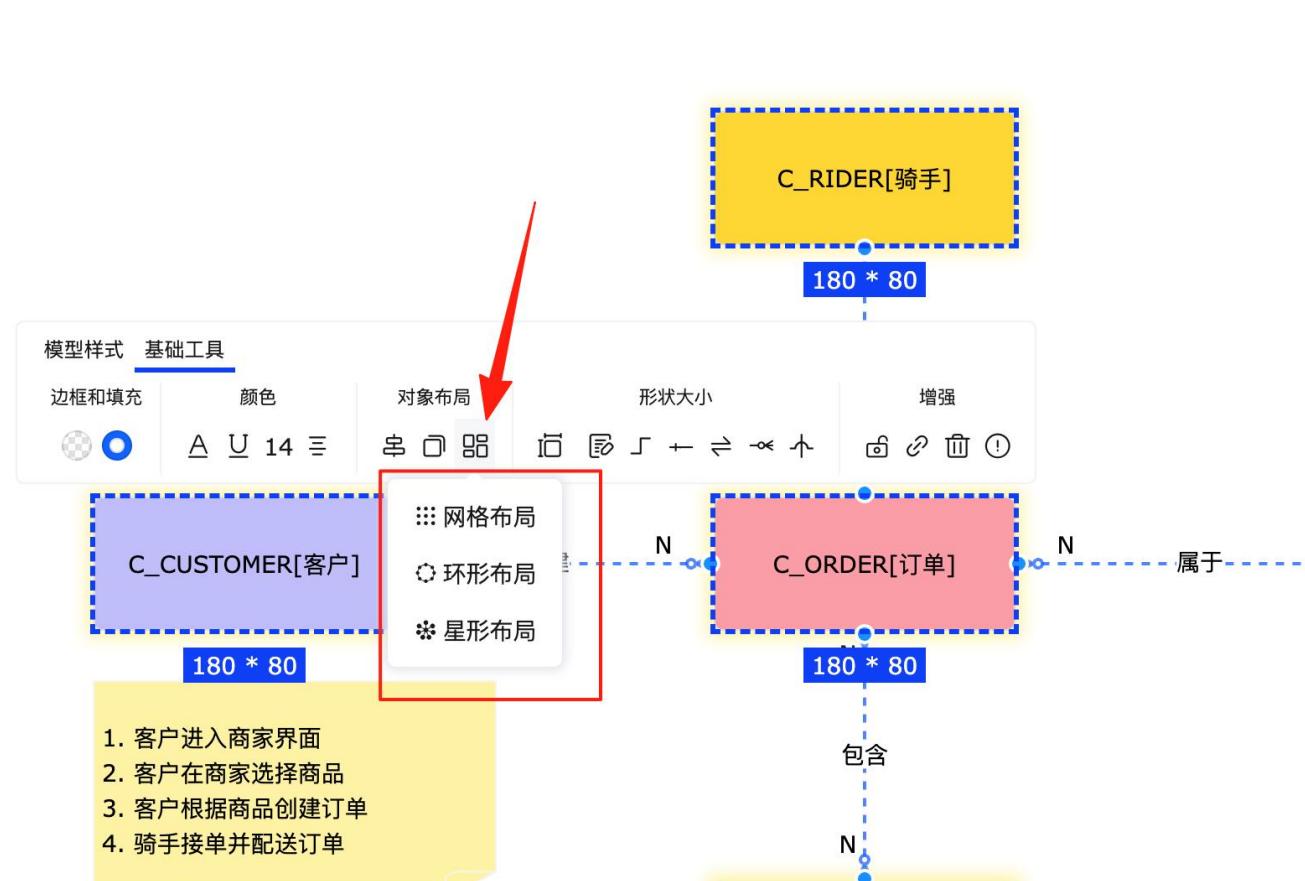
3. 增量更新：支持生成表结构变更的DDL

## 5.7 关系图设计

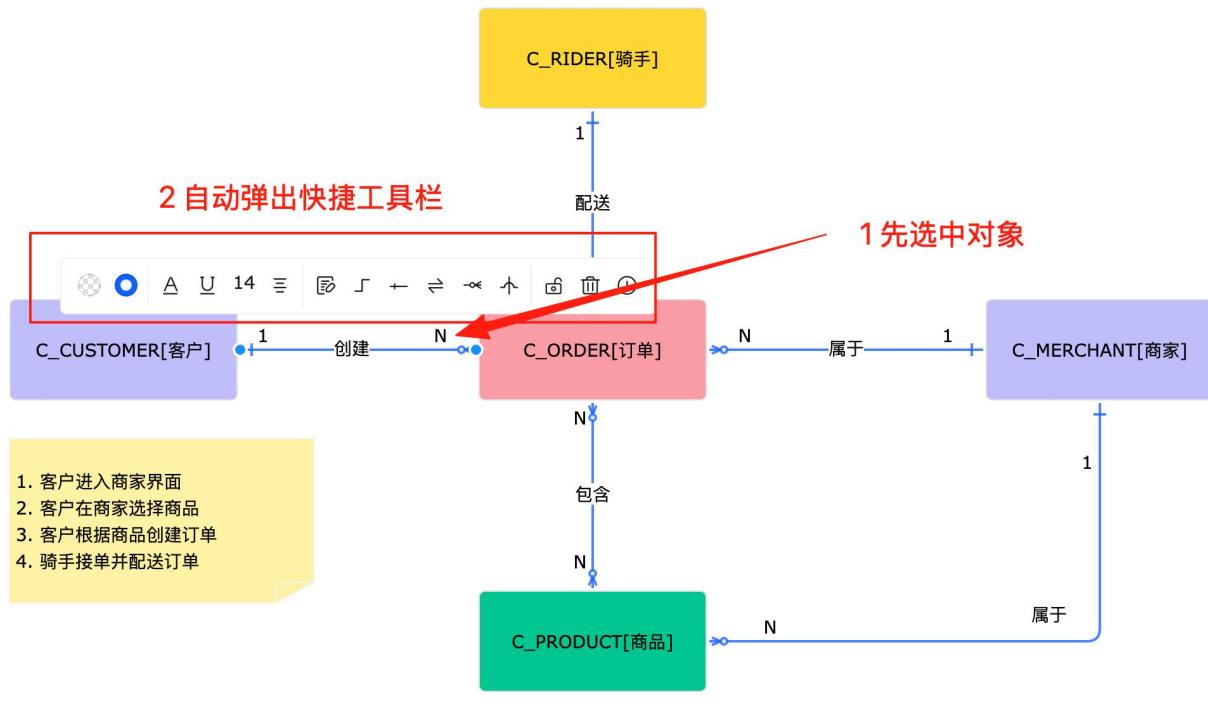
## 5.7.1 ER图绘制

PDMaas建模工具提供专业的ER图绘制功能：

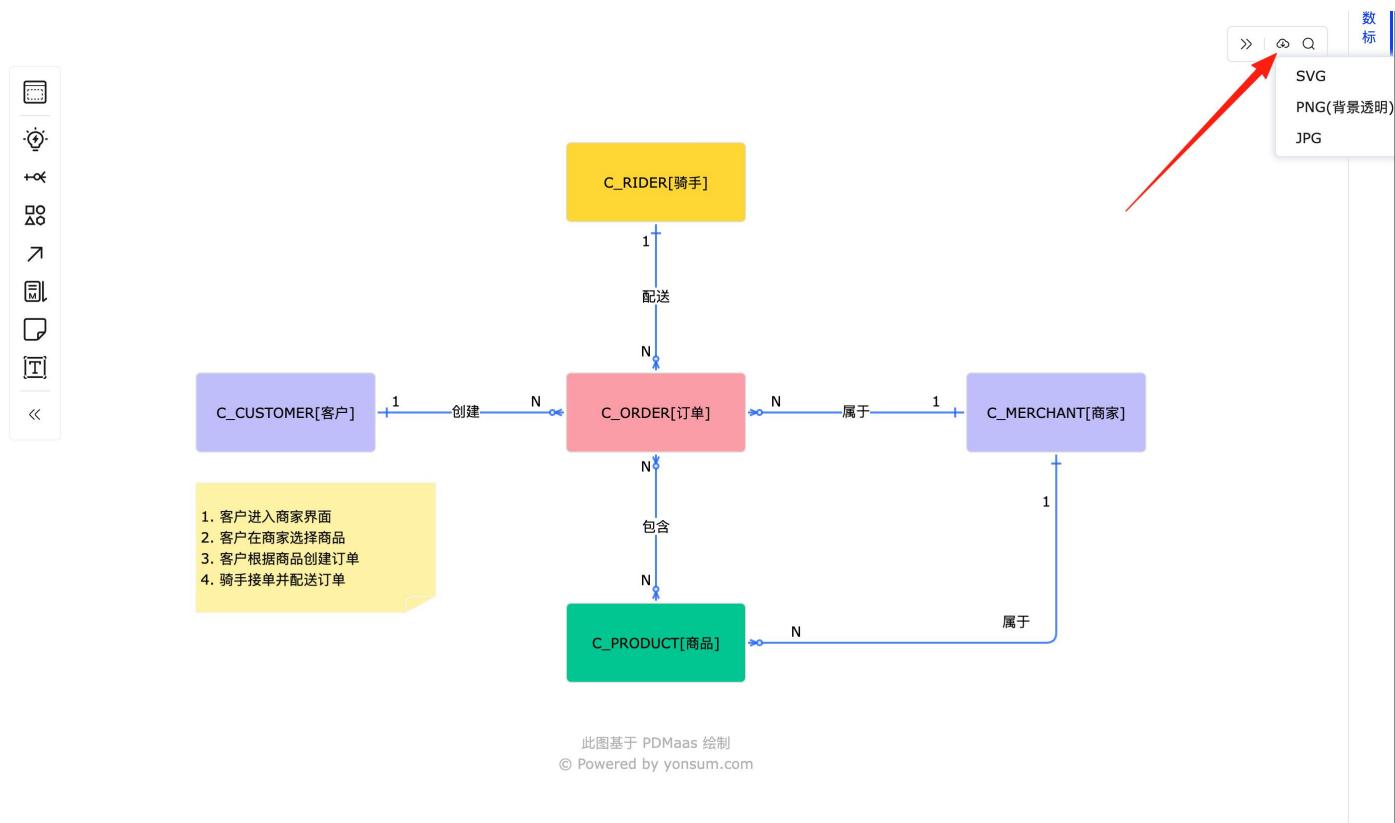
1. 拖拽式设计：通过拖拽创建实体和关系
2. 自动布局：自动优化图形布局



3. 样式设置：自定义实体和连线样式



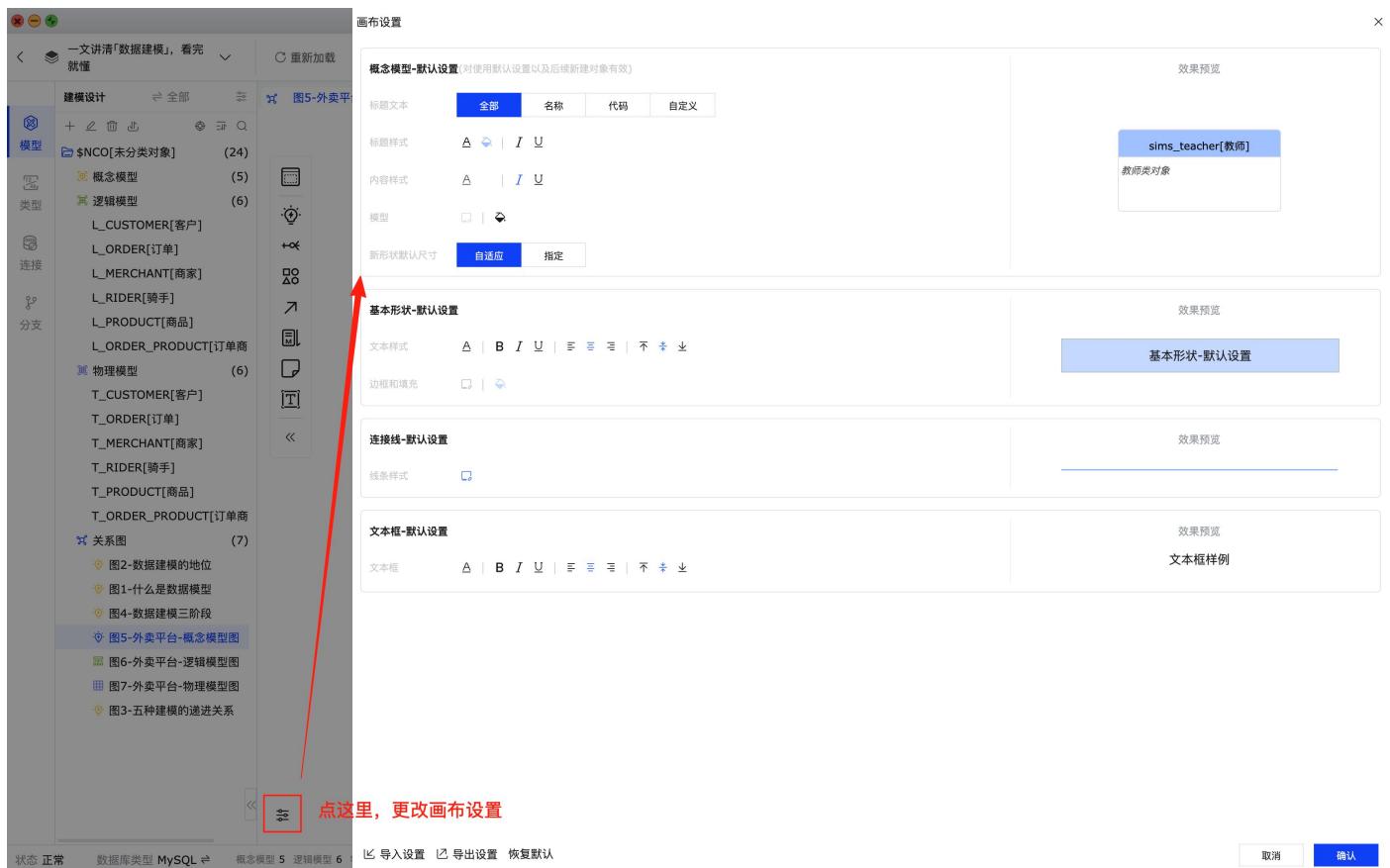
#### 4. 导出功能：导出为图片或PDF格式



#### 5.7.2 画布设置

可以对ER图的显示进行个性化设置：

1. 字段显示：选择显示的字段信息
2. 颜色主题：设置实体和关系的颜色
3. 字体大小：调整文字显示大小



## 5.8 数据导入导出

### 5.8.1 数据导入功能

PDMaas建模工具支持多种数据源的导入：

1. 从数据库逆向解析

模型

类型

连接

分支

建模设计

全部

快速搜索

请输入模型名/关系图名/字段名/标准名

导入

- 从数据库逆向读取解析
- 从DDL建表语句
- 从EXCEL文件
- 从PowerDesigner文件
- 从EZDML文件
- 从PDMaker及老版本文件
- 从项目文件导入
- 导出
- 导出项目
- 导出DDL
- 导出EXCEL文档
- 导出WORD文档
- 导出HTML文档
- 导出Markdown文档

操作快捷键

上下方向键	选中行的情况下，切换至下一选中行
左右方向键	在单元格编辑情况下，则是移至上/下一行单元格
单元格	在单元格编辑情况下，如果是在文本开始或者末尾，
CTRL+上下方向键	选中行的情况下，从当前行开始，增加选择上/下一行

从数据库逆向读取解析

1 选择数据库环境      2 选择表并进行分类      3 执行读取

批量设置分类目录  突出显示项目存在的表  隐藏已经存在表 按住shift可选中多行, 按住ctrl可多选

	分类目录	顺序	schema	表代码	表名称	表注释
1		1	democ	sims_instruct	授课	
2		2	democ	sims_class_fee	班费	
3		3	democ	sims_admin	管理员	
4		4	democ	lbra_catalog	图书分类	
5		5	democ	lbra_book	图书	
6		6	democ	lbra_reader	读者	
7		7	democ	lbra_book_discuss	图书评论	
8		8	democ	dorm_building	宿舍楼	
9		9	democ	dorm_room	宿舍房间	
10		10	democ	dorm_room_member	宿舍入住	
11		11	democ	sims_college	学院	
12		12	democ	sims_lesson	课程	
13		13	democ	sims_class	班级	
14		14	democ	lbra_borrow	借阅	

您已选择了0张表 加载完成 总共19张表

1. 针对项目中已存在的表您可以通过：“常用工具/物理表模型比较/与数据库比较”比较后，将差异同步至模型

极速模式（不记录操作历史，大量的数据库表结构需要初始化进项目时适用）

上一步 下一步

## 2. 从DDL语句导入

🕒 复制成功

模型  
类型  
连接  
分支

建模设计

⇒ 全部

...



+

/

-

山

⊕

≡

?



快速搜索

○ 请输入模型名/关系图名/字段名/标准名

导入

从数据库逆向读取解析

从DDL建表语句

从EXCEL文件

从PowerDesinger文件

从EZDML文件

从PDManner及老版本文件

从项目文件导入

导出

物理模型

导出项目

导出DDL

导出EXCEL文档

导出WORD文档

导出HTML文档

导出MarkDown文档

操作快捷键

上下方向键

选中行的情况下，切换至下一选中行

左右方向键

在单元格编辑情况下，则是移至上/下一行单元格

单元格

CTRL+上下方向键

选中行的情况下，从当前行开始，增加选择上/下一行

模型  
类型  
连接  
分支

从DDL建表语句



```

47     'PHONE_NUMBER' VARCHAR(20) NOT NULL COMMENT '联系电话;需符合手机号格式, 唯一',
48     'CURRENT_STATUS' VARCHAR(20) NOT NULL COMMENT '当前状态;枚举值: 在线、离线、忙碌, 默认为在线',
49     'ID_CARD' VARCHAR(18) NOT NULL COMMENT '身份证号;需符合身份证格式, 唯一',
50     'JOIN_TIME' DATETIME NOT NULL COMMENT '入职时间',
51     'AVG_DELIVERY_TIME' DECIMAL(5,1) COMMENT '平均配送时长;单位: 分钟',
52     PRIMARY KEY (`RIDER_ID`)
53 ) COMMENT '骑手';
54
55
56 DROP TABLE IF EXISTS T_PRODUCT;
57 CREATE TABLE T_PRODUCT(
58     'PRODUCT_ID' VARCHAR(32) NOT NULL COMMENT '商品 ID;唯一标识商品',
59     'MERCHANT_ID' VARCHAR(32) NOT NULL COMMENT '商家 ID;外键, 关联商家表',
60     'CATEGORY' VARCHAR(50) NOT NULL COMMENT '商品分类',
61     'PRODUCT_NAME' VARCHAR(100) NOT NULL COMMENT '商品名称',
62     'PRICE' DECIMAL(10,2) NOT NULL COMMENT '商品单价',
63     'PRODUCT_SPEC' VARCHAR(100) COMMENT '商品规格',
64     'STOCK_QUANTITY' INT(10) NOT NULL COMMENT '库存数量',
65     'PRODUCT_STATUS' VARCHAR(20) NOT NULL COMMENT '商品状态;枚举值: 在售、下架、售罄, 默认为在售',
66     'INTRO' VARCHAR(900) COMMENT '商品备注',
67     PRIMARY KEY (`PRODUCT_ID`)
68 ) COMMENT '商品';
69
70
71 DROP TABLE IF EXISTS T_ORDER_PRODUCT;
72 CREATE TABLE T_ORDER_PRODUCT(
73     'ORDER_PRODUCT' NVARCHAR(128) NOT NULL COMMENT '订单商品关联ID',
74     'ORDER_ID' VARCHAR(32) NOT NULL COMMENT '订单 ID;外键, 关联订单表, 复合主键之一',
75     'PRODUCT_ID' VARCHAR(32) NOT NULL COMMENT '商品 ID;外键, 关联商品表, 复合主键之一',
76     'QUANTITY' INT(10) NOT NULL COMMENT '购买数量',
77     'UNIT_PRICE' DECIMAL(10,2) NOT NULL COMMENT '购买单价',
78     'SUBTOTAL_AMOUNT' DECIMAL(10,2) NOT NULL COMMENT '商品小计金额;等于 QUANTITY×UNIT_PRICE, 大于 0',
79     'REMARK' VARCHAR(200) COMMENT '商品备注',
80     PRIMARY KEY (`ORDER_PRODUCT`)
81 ) COMMENT '订单商品关联';
82
83

```

取消

下一步

从DDL建表语句

1 加载DDL语句 2 选择表并进行分类 3 执行读取

批量设置分类目录  突出显示项目存在的表  隐藏已经存在表

按住shift可选中多行, 按住ctrl可多选

	分类目录	顺序	schema	表代码	表名称	表注释
1		1		T_CUSTOMER	客户	
2		2		T_ORDER	订单	
3		3		T_MERCHANT	商家	
4		4		T_RIDER	骑手	
5		5		T_PRODUCT	商品	
6		6		T_ORDER_PRODUCT	订单商品关联	

您已选择了0张表 加载完成 总共6张表

1. 针对项目中已存在的表可以通过：“常用工具/物理表模型比较/与数据库比较”比较后, 将差异同步至模型

上一步 下一步

状态: 正常 数据库:

### 3. 从Excel导入

**模型****类型****连接****分支**

建模设计    全部

+   $\mathcal{L}$    $\mathbb{A}$    $\mathbb{B}$    $\mathbb{C}$    $\mathbb{D}$    $\mathbb{E}$    $\mathbb{F}$    $\mathbb{G}$    $\mathbb{H}$    $\mathbb{I}$    $\mathbb{J}$    $\mathbb{K}$    $\mathbb{L}$    $\mathbb{M}$    $\mathbb{N}$    $\mathbb{O}$    $\mathbb{P}$    $\mathbb{Q}$ **快速搜索**

请输入模型名/关系图名/字段名/标准名

艺龙产品计

艺龙酒店计

艺龙订单

dbrevers

- 从数据库逆向读取解析
- 从DDL建表语句
- 从EXCEL文件**
- 从PowerDesinger文件
- 从EZDML文件
- 从PDMAner及老版本文件
- 从项目文件导入

- 物理
- 导出项目
- 导出DDL
- 导出EXCEL文档
- 导出WORD文档
- 导出HTML文档
- 导出MarkDown文档

**操作快捷键**

上下方向键

选中行的情况下，切换至下一选中行

左右方向键

在单元格编辑情况下，则是移至上/下一行单

单元格

在单元格编辑情况下，如果是在文本开始或

单元格

单元格

选中行的情况下，从当前行开始，增加选择

## 5.8.2 数据导出功能

支持多种格式的数据导出：

- 导出DDL：生成建表语句脚本
- 导出Excel：生成数据字典文档
- 导出Word：生成设计文档
- 导出HTML：生成网页版文档



The screenshot shows the Yonsum modeling tool interface. On the left, there is a sidebar with icons for Model (selected), Type, Connection, and Branch. The main workspace shows a project named 'test'. The top bar includes a back arrow, a folder icon, the project name 'test', a dropdown for '重新加载' (Reload), and a '最近保存' (Last saved) timestamp '02:04:33'. A context menu is open over a project named '艺龙产品线'. The menu is divided into '导入' (Import) and '导出' (Export) sections. The '导出' section is highlighted with a red box and contains the following options: '导出DDL', '导出EXCEL文档', '导出WORD文档', '导出HTML文档', and '导出MarkDown文档'. To the right of the menu, there is a '快速搜索' (Quick search) input field and a list of keyboard shortcuts with their descriptions.

操作快捷键	描述
上下方向键	选中行的情况下，切换至下一选中行
左右方向键	在单元格编辑情况下，则是移至上/下一行单元格
单元格	在单元格编辑情况下，如果是在文本开始或者末尾，则是移至上/下一行单元格
CTRL+上下方向键	选中行的情况下，从当前行开始，增加选择上/下一行

## 5.9 团队协作功能

## 5.9.1 版本控制

PDMaas建模工具提供完善的版本管理功能：

- 分支管理：创建开发分支进行并行开发
- 版本标签：为重要版本打标签
- 变更历史：记录所有操作历史
- 版本比较：比较不同版本间的差异

The screenshot displays the PD Maas version control interface. On the left, a sidebar shows navigation links for '模型', '类型', '连接', and '代码'. The main area is divided into three sections: '分支' (Branches), '版本标签' (Version Tags), and '变更历史' (Change History).

**分支 (Branches):** A table lists four branches: master (默认), develop, pro, and dev\_saas. The 'develop' branch is highlighted with a red arrow pointing to it from the sidebar.

**版本标签 (Version Tags):** A table lists version tags: v2.1.7, v2.1.6, v2.1.5, v2.1.4, v2.1.3, v2.1.2, and v2.1.1.

**变更历史 (Change History):** A detailed timeline shows commits for the 'develop' branch. The commits are color-coded by author (blue for 'pro', orange for 'master', and red for 'dev\_saas') and include timestamps and descriptions of operations like creating branches and tags, and merging branches.

At the bottom, the status is shown as '正常' (Normal), the database type is 'MySQL', and the current branch is 'develop'.

## 5.9.2 多人协作

- 实时同步：多人同时编辑，实时同步变更
- 冲突解决：自动检测冲突并提供解决方案
- 权限控制：基于角色的访问权限控制

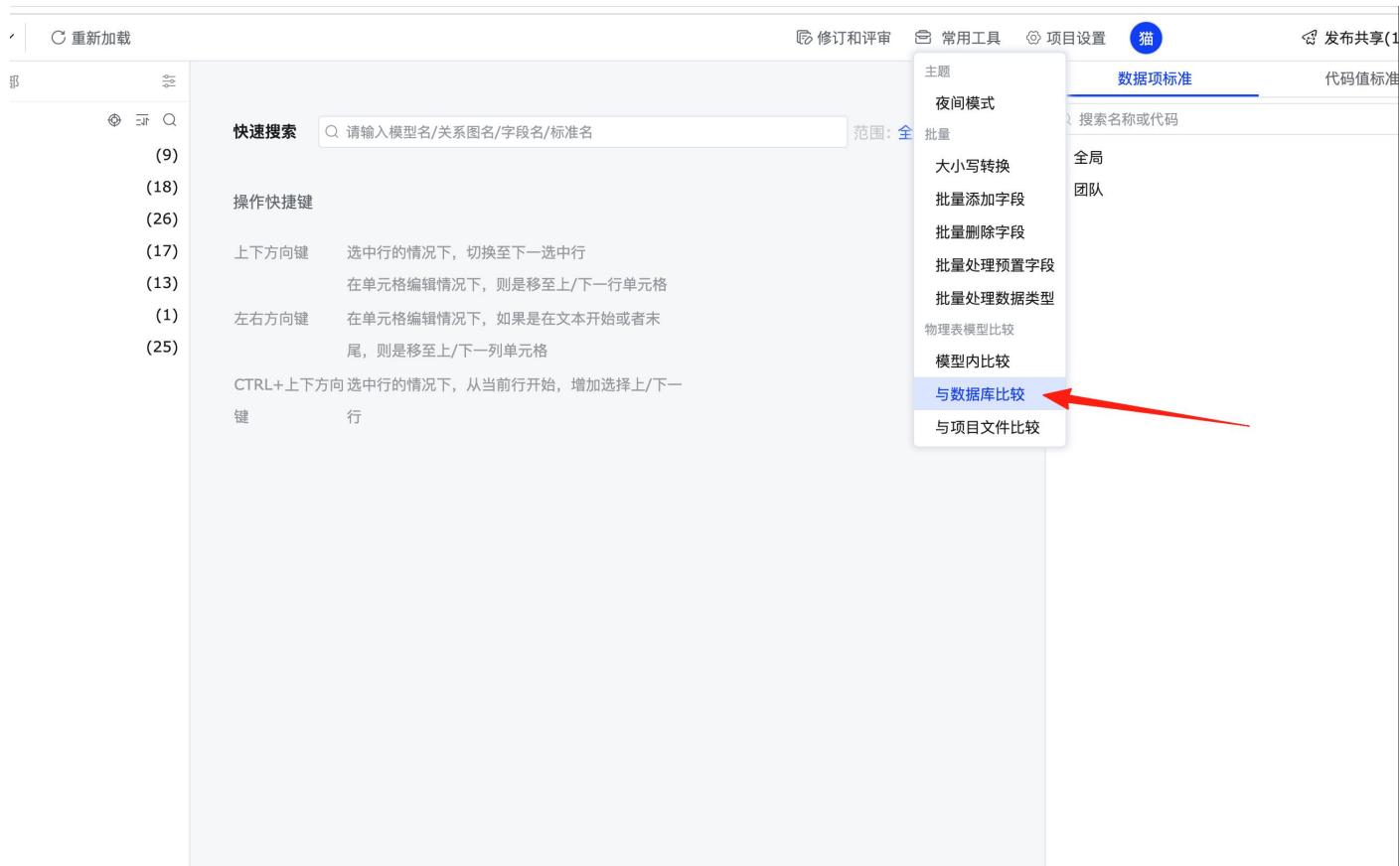
## ■ 操作审计：记录每个用户的操作日志

### 模型效率工具

#### 5.9.3 模型比较工具

PDMaas提供强大的模型比较功能：

- 模型内比较：比较项目内不同模型的差异
- 与数据库比较：比较模型与实际数据库的差异



The screenshot shows the PDMaas application interface. At the top, there is a navigation bar with various icons and text links: '重新加载', '修订和评审', '常用工具', '项目设置', '猫' (a user icon), and '发布共享(1)'. Below the navigation bar, there is a search bar labeled '请输入模型名/关系图名/字段名/标准名' and a dropdown menu labeled '范围: 全'. The main content area displays a table of operations with their descriptions. On the right side, a vertical sidebar contains several options: '主题', '夜间模式', '批量', '大小写转换', '批量添加字段', '批量删除字段', '批量处理预置字段', '批量处理数据类型', '物理表模型比较', '模型内比较', and '与数据库比较'. The '与数据库比较' option is highlighted with a blue background and a red arrow points to it from the bottom right.

当前数据库环境: PDMAas2苏州研发... sz-pdmaas2\_1

标准比较

自定义比较

结果		模型侧				操作		
差异	分类目录	schema	代码	显示名称	说明	字段数	schema	查看
1 一致	CLAB[协作平台]	pdmaas_ee_boot_saas	CLAB_TEAM	团队空间		14	pdmaas_ee_b	查看
2 一致	CLAB[协作平台]	pdmaas_ee_boot_saas	CLAB_USER_CATEGORY	用户分类目录		14	pdmaas_ee_b	查看
3 一致	CLAB[协作平台]	pdmaas_ee_boot_saas	CLAB_SPEC_CHECK	规范检查		11	pdmaas_ee_b	查看
4 一致	CLAB[协作平台]	pdmaas_ee_boot_saas	CLAB_USER_CATEGORY_ITEM	用户目录内容		12	pdmaas_ee_b	查看
5 一致	CLAB[协作平台]	pdmaas_ee_boot_saas	CLAB_TEAM_CATEGORY	团队目录		14	pdmaas_ee_b	查看
6 一致	CLAB[协作平台]	pdmaas_ee_boot_saas	CLAB_TEAM_CATEGORY_ITEM	团队目录内容		14	pdmaas_ee_b	查看
7 一致	CLAB[协作平台]	pdmaas_ee_boot_saas	CLAB_TEAM_USER	团队成员		13	pdmaas_ee_b	查看
8 不一致	CLAB[协作平台]	pdmaas_ee_boot_saas	CLAB_BIZ_DATA_TYPE	业务数据类型		22	pdmaas_ee_b	查看
9 一致	SYST[系统管理]	pdmaas_ee_boot_saas	SYST_DB_DIALECT	数据库方言		31	pdmaas_ee_b	查看
10 一致	CLAB[协作平台]	pdmaas_ee_boot_saas	CLAB_DB_CONN	数据库连接		24	pdmaas_ee_b	查看
11 不一致	CLAB[协作平台]	pdmaas_ee_boot_saas	CLAB_DB_ENV	数据库环境		18	pdmaas_ee_b	查看
12 一致	CLAB[协作平台]	pdmaas_ee_boot_saas	CLAB_DB_ENV_EXCLUDE_TABLE	环境例外表		15	pdmaas_ee_b	查看
13 一致	CLAB[协作平台]	pdmaas_ee_boot_saas	CLAB_DB_ENV_MEMBER_PERMISSION	数据库环境对用户的权限设置		12	pdmaas_ee_b	查看
14 一致	SYST[系统管理]	pdmaas_ee_boot_saas	SYST_DATA_TYPE	基本数据类型		18	pdmaas_ee_b	查看
15 一致	SYST[系统管理]	pdmaas_ee_boot_saas	SYST_DATA_TYPE_DB	数据类型与方言映射		12	pdmaas_ee_b	查看
16 一致	STND[数据标准]	pdmaas_ee_boot_saas	STND_POLICY_CATEGORY	制度目录		13	pdmaas_ee_b	查看
17 一致	STND[数据标准]	pdmaas_ee_boot_saas	STND_POLICY_DOC	制度文档		33	pdmaas_ee_b	查看
18 一致	STND[数据标准]	pdmaas_ee_boot_saas	STND_POLICY_CARRY	制度文档贯彻		13	pdmaas_ee_b	查看
19 不一致	STND[数据标准]	pdmaas_ee_boot_saas	STND SUBJECT	主题		18	pdmaas_ee_b	查看
20 一致	STND[数据标准]	pdmaas_ee_boot_saas	STND_DICT	字典标准		28	pdmaas_ee_b	查看

## 5.9.4 批量处理工具

- 大小写转换：统一字段命名格式
- 批量修改：批量添加以及删除字段

### 最佳实践建议：

- 建立团队的建模规范和标准定期进行模型评审和优化
- 充分利用标准库提高建模效率做好版本管理和备份工作
- 建立模型与数据库的同步机制

## 6. 实战练习与评估

通过实际操作练习，巩固数据建模理论知识和工具使用技能。本章提供了从简单到复杂的练习题目，帮助学员逐步提升建模能力。

### 6.1 基础练习

#### 6.1.1 练习1：图书管理系统概念建模

- **场景描述：**设计一个简单的图书管理系统，支持图书借阅功能。
- **业务需求：**

读者可以注册成为会员，借阅图书，图书馆管理员负责图书入库和借还管理。

每本图书可以被多次借阅，但同一时间只能被一个读者借阅需要记录借阅历史和归还情况

- **任务要求：**

1. 识别系统中的核心实体
2. 分析实体间的关系
3. 绘制概念模型ER图

- **预期产出：**

概念模型图，包含至少4个实体和相应关系

#### 6.1.2 练习2：在线商城订单逻辑建模

- **场景描述：**为在线商城的订单模块设计逻辑模型。已知概念模型：用户、商品、订单、购物车
- **任务要求：**

1. 将概念模型转换为逻辑模型
2. 为每个实体设计详细字段
3. 定义主键、外键关系
4. 设置必要的数据约束

■ 评估标准：

1. 字段设计是否完整、合理
2. 主外键关系是否正确
3. 数据约束是否符合业务规则

## 6.2 进阶练习

### 6.2.1 练习3：医院管理系统完整建模

■ 场景描述：设计医院管理系统的数据模型，涵盖挂号、诊疗、开药等核心业务。

■ 业务流程：

1. 患者挂号预约医生
2. 医生接诊并记录病历
3. 医生开具处方
4. 患者到药房取药
5. 财务科进行费用结算

■ 任务要求：

1. 完成从概念模型到物理模型的完整设计

2. 使用PDMaas工具完成建模
3. 生成MySQL的DDL建表语句
4. 设计合理的索引策略
5. 输出完整的数据字典文档