

🏠 / 开发指南 / 后端手册

👤 芋道源码 📅 2023-11-17

## 代码生成【主子表】

### 友情提示：

本文接 [《代码生成【单表】》](#)，请务必先阅读。因为重复的内容，本文会不再赘述！

主子表，指的是一个主表，被多个子表所关联，关联的关系是一对一或一对多。

例如说：主表是【学生】，子表可以是：

- 子表是【成绩】，两者是“一对多”的关系，一个学生可以有多个成绩。
- 子表是【班级】，两者是“一对一”的关系，一个学生只能有一个班级。

下面，我们将演示“主子表”的使用，基于代码生成器，在 `yudao-module-system` 模块中，开发一个【学生】的功能。

### 友情提示：

目前只有 `yudao-ui-admin-vue3` 支持主子表，`yudao-ui-admin-vue2`、`yudao-ui-admin-vben` 正在适配中！

## 0. 主子表模式

针对不同的交互模式，项目提供了三种主子表模式：标准、ERP、内嵌。

### 0.1 标准模式

对应 [基础设施 -> 代码生成案例 -> 主子表（标准）] 菜单。

在新增和修改时，主表和子表在一个弹窗表单中，一起提交。如下图所示：

新增

\* 名字

请输入名字

① 主表

\* 性别

☐男☐女

\* 出生日期

选择出生日期

\* 简介

正文

B U I ... A A

默认字号 默认字体

默认行高

: : ☑ ≡ ≡ ☺ 🔗 🖼️ ▶ 📄 </> ⋮

↶ ↷ [ ]

请输入内容...

学生课程

学生班级

② 子表，通过 tab 切换表单

| 序号 | 名字               | 分数               | 操作 |
|----|------------------|------------------|----|
| 1  | <div>请输入名字</div> | <div>请输入分数</div> | —  |
| 2  | <div>请输入名字</div> | <div>请输入分数</div> | —  |
| 3  | <div>请输入名字</div> | <div>请输入分数</div> | —  |

+ 添加学生课程

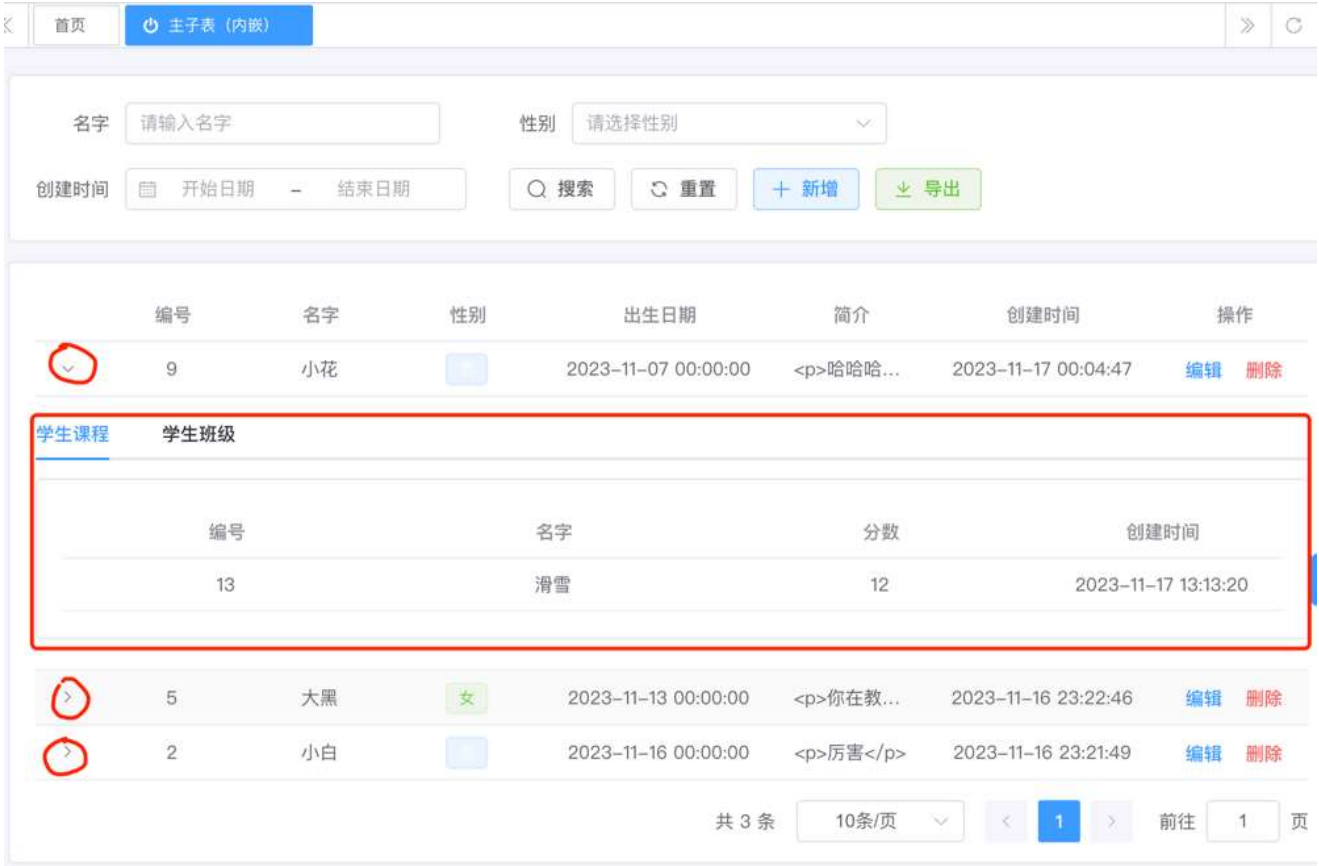
确定

取消

## 0.2 内嵌模式

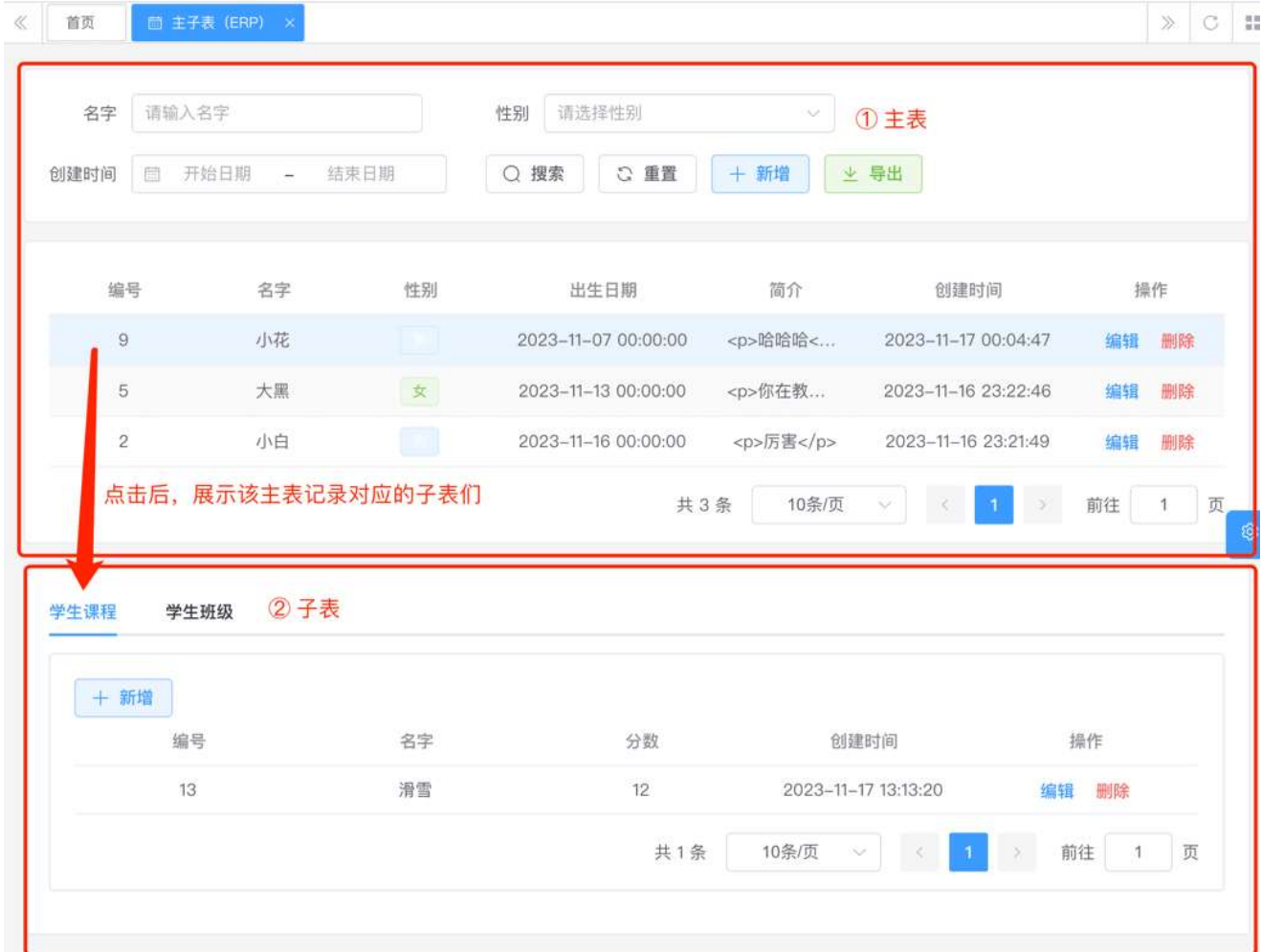
对应[基础设施->代码生成案例->主子表(内嵌)]菜单。

在「标准模式」的基础之上，列表 **内嵌** 子表的列表。如下图所示：



### 0.3 ERP 模式

对应 [基础设施 -> 代码生成案例 -> 主子表 (ERP) ] 菜单。  
主表和子表，独立列表，也独立表单。如下图所示：



# 1. 数据库表结构设计

① 设计 **主表** 的数据库表名为 `system_student` 学生表，其建表语句如下：

```
CREATE TABLE `system_student` (  
  `id` bigint NOT NULL AUTO_INCREMENT COMMENT '编号',  
  `name` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL  
  `birthday` datetime NOT NULL COMMENT '出生日期',  
  `description` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NC  
  `creator` varchar(64) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci DEFAULT  
  `create_time` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',  
  `updater` varchar(64) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci DEFAULT  
  `update_time` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TI  
  `deleted` bit(1) NOT NULL DEFAULT b'0' COMMENT '是否删除',  
  `tenant_id` bigint NOT NULL DEFAULT '0' COMMENT '租户编号',  
  PRIMARY KEY (`id`) USING BTREE  
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicoc
```

② 设计 **子表** 的数据库表名为 `system_student_course` 学生课程表，其建表语句如下：

```
CREATE TABLE `system_student_course` (  
  `id` bigint NOT NULL AUTO_INCREMENT COMMENT '编号',  
  `student_id` bigint NOT NULL COMMENT '学生编号',  
  `name` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL  
  `score` tinyint NOT NULL COMMENT '分数',  
  `creator` varchar(64) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci DEFAULT  
  `create_time` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',  
  `updater` varchar(64) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci DEFAULT  
  `update_time` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TI  
  `deleted` bit(1) NOT NULL DEFAULT b'0' COMMENT '是否删除',  
  `tenant_id` bigint NOT NULL DEFAULT '0' COMMENT '租户编号',  
  PRIMARY KEY (`id`) USING BTREE  
) ENGINE=InnoDB AUTO_INCREMENT=14 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicoc
```

它和主表的关系是一对多，一个学生可以有多个课程，通过 `student_id` 字段进行关联。

③ 设计 **子表** 的数据表名为 `system_student_grade` 学生班级表，其建表语句如下：

```
CREATE TABLE `system_student_grade` (  
  `id` bigint NOT NULL AUTO_INCREMENT COMMENT '编号',  
  `student_id` bigint NOT NULL COMMENT '学生编号',  
  `name` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL  
  `teacher` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '班主任',  
  `creator` varchar(64) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci DEFAULT  
  `create_time` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',  
  `updater` varchar(64) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci DEFAULT  
  `update_time` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TI  
  `deleted` bit(1) NOT NULL DEFAULT b'0' COMMENT '是否删除',  
  `tenant_id` bigint NOT NULL DEFAULT '0' COMMENT '租户编号',  
  PRIMARY KEY (`id`) USING BTREE  
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_uni
```

它和主表的关系是一对一，一个学生只能有一个班级，通过 `student_id` 字段进行关联。

## 2. 代码生成

### 2.1 导入表

点击 [基础设施 -> 代码生成] 菜单，点击 [基于 DB 导入] 按钮，选择 `system_student` 、  
`system_student_course` 、 `system_student_grade` 表，后点击 [确认] 按钮。

### 2.2 编辑配置（主表）

点击 `system_student` 所在行的 [编辑] 按钮，修改生成配置。后操作如下：

基本信息

字段信息

生成信息

\* 生成模板

主表（标准模式）

\* 生成场景

管理后台 关键!!!

\* 模块名 ?

system

\* 类名称 ?

Student

\* 前端类型

Vue3 Element Plus 标准

上级菜单 ?

系统管理

\* 业务名 ?

student

\* 类描述 ?

学生

保存

返回

- 将【生成模版】设置为【主表（标准模式）】。🔥最最关键的步骤！
- 将【上级菜单】设置为【系统管理】。
- 将【前端类型】设置为“前端项目”对应的“前端类型”。例如说，我们这里演示的是 `yudao-ui-admin-vue3` 前端项目，则选择了【Vue3 Element Plus 标准模版】。

## 2.3 编辑配置（子表）

① 点击 `system_student_course` 所在行的 [编辑] 按钮，修改生成配置。后操作如下：

基本信息

字段信息

生成信息

\* 生成模板子表

\* 生成场景管理后台

\* 模块名 ?system

\* 类名称 ?StudentCourse

\* 前端类型Vue3 Element Plus 标准

上级菜单 ?系统管理

\* 业务名 ?student

\* 类描述 ?学生课程

主表信息

\* 关联的主表 ?system\_student: 学生表

\* 子表关联的字段 ?student\_id: 学生编号

\* 关联关系 ?☒ 一对多 ☐ 一对一

保存

返回

- 将【生成模版】设置为【子表】。🔥最最关键的步骤！
- 业务名：一般建议和【主表】保持一致，所以这里改成了 `student` 。
- 主表信息：将【关联的主表】设置为 `system_student` 表，将【子表关联的字段】设置为 `student_id` 字段，将【关联关系】设置为“一对多”。

② 点击 `system_student_grade` 所在行的 [编辑] 按钮，修改生成配置。后操作如下：

基本信息

字段信息

生成信息

\* 生成模板子表

\* 生成场景管理后台

\* 模块名 ?system

\* 类名称 ?StudentGrade

\* 前端类型Vue3 Element Plus 标准

上级菜单 ?系统管理

\* 业务名 ?student

\* 类描述 ?学生班级

主表信息

\* 关联的主表 ?system\_student: 学生表

\* 子表关联的字段 ?student\_id: 学生编号

\* 关联关系 ?☐ 一对多 ☒ 一对一

保存

返回

- （同上）将【生成模版】设置为【子表】。🔥最最关键的步骤！

- （同上）业务名：一般建议和【主表】保持一致，所以这里改成了 `student` 。
- （基本同上，关联关系不同）主表信息：将【关联的主表】设置为 `system_student` 表，将【子表关联的字段】设置为 `student_id` 字段，将【关联关系】设置为“一对一”。

## 2.4 预览代码

点击 `system_student` 所在行的 [预览] 按钮，在线预览生成的代码，检查是否符合预期。

## 2.5 生成代码

点击 `system_student` 所在行的 [生成] 按钮，生成代码。

# 3. 代码运行

和 《代码生成【单表】》 的「3. 代码运行」一致，就不重复赘述。

---

← 代码生成【单表】（新增功能）

代码生成（树表） →



Theme by [Vdoing](#) | Copyright © 2019-2024 芋道源码 | MIT License