

菲斯科白泽产品

详细设计V1.0



2021-10-13

菲斯科上海（软件）有限公司

上海市延安西路728号华敏翰尊国际4楼J座

V1.0

[1 文档说明 2](#_Toc85533051)

[1.1 文档目的 2](#_Toc85533052)

[1.2 背景 2](#_Toc85533053)

[1.3 读者对象 2](#_Toc85533054)

[1.4 术语和定义 2](#_Toc85533055)

[1.5 人员安排 3](#_Toc85533056)

[2 概要设计 0](#_Toc85533057)

[2.1 DMP系统架构 0](#_Toc85533058)

[2.2 DMP微服务架构 0](#_Toc85533059)

[2.3 DMP后台结构 1](#_Toc85533060)

[2.4 DMP系统逻辑架构 2](#_Toc85533061)

[2.5 环境说明 3](#_Toc85533062)

[2.5.1 系统语言 3](#_Toc85533063)

[2.5.2 系统平台 3](#_Toc85533064)

[2.5.3 服务器配置 3](#_Toc85533065)

[3 详细设计 3](#_Toc85533066)

[3.1 数据结构 3](#_Toc85533067)

[3.1.1 数据库设计 3](#_Toc85533068)

[3.2 模块设计 4](#_Toc85533069)

[3.2.1 数据接入 4](#_Toc85533070)

[3.2.2 数据建模 4](#_Toc85533071)

[3.2.3 数据可视化 4](#_Toc85533072)

[3.2.4 数据治理 5](#_Toc85533073)

[3.2.5 数据服务 5](#_Toc85533074)

[3.2.6 系统管理 5](#_Toc85533075)

[3.2.7 系统用户 5](#_Toc85533076)

[3.2.8 消息队列 5](#_Toc85533077)

[3.2.9 数据工厂 5](#_Toc85533078)

[3.2.10 数据治理 9](#_Toc85533079)

[3.2.11 系统管理 9](#_Toc85533080)

# 文档说明

## 文档目的

本文档编写的目的是说明Fisk白泽产品开发模块的设计，为开发人员编码提供依据。

## 背景

软件系统名称：DMP数据治理平台。

## 读者对象

预期读者包括Fisk产品部开发人员。

## 术语和定义

|  |  |  |
| --- | --- | --- |
| 术语 | 描述 |  |
| Fisk | 菲斯科（上海）软件有限公司 |  |
| DMP | 菲斯科白泽产品 |  |
|  |  |  |

## 人员安排

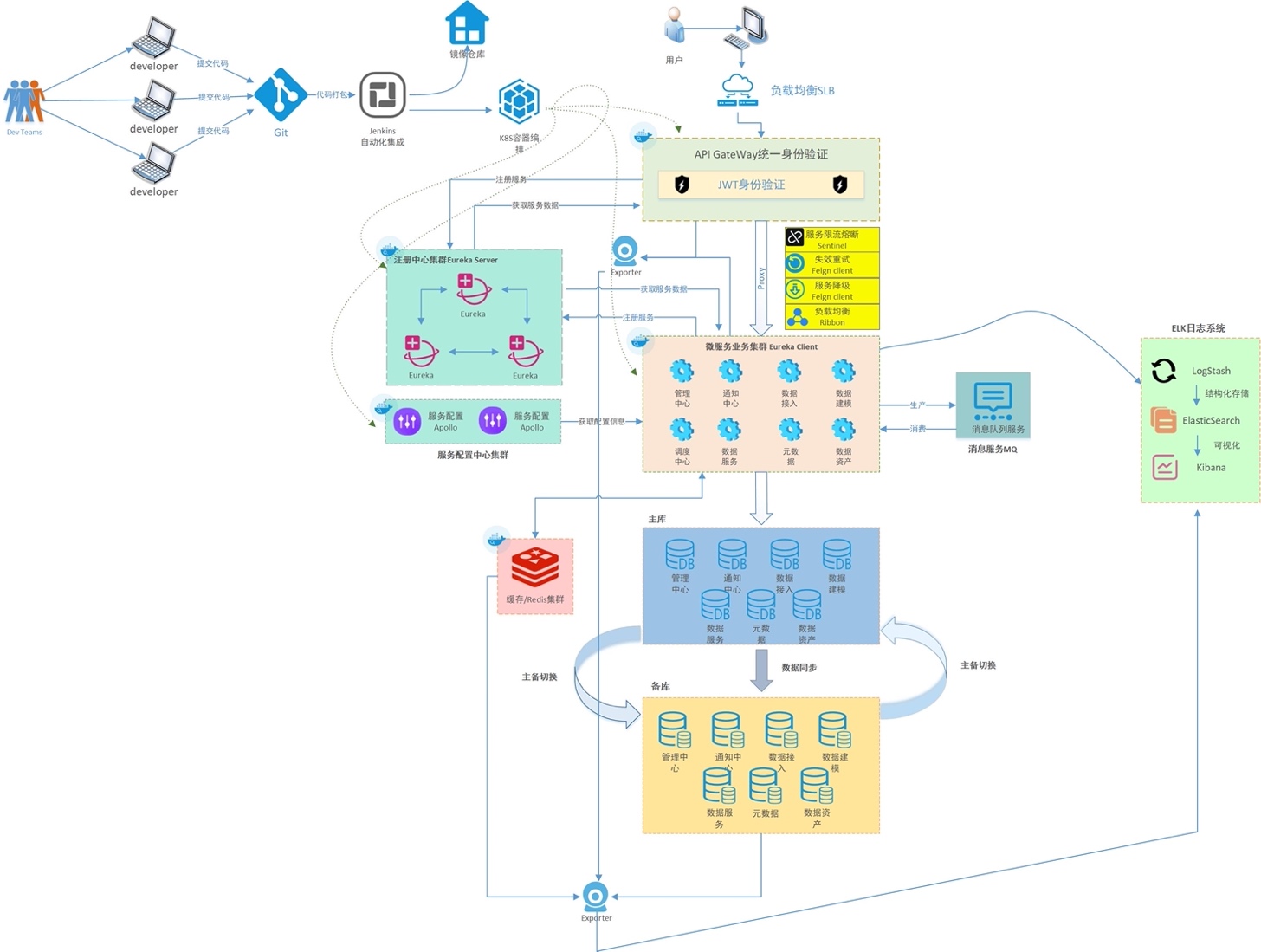
|  |  |  |
| --- | --- | --- |
| **模块** | **前端负责人** | **后端负责人** |
| 数据接入 | 胡卓程 | 赵飞鸿 |
| 数据建模 | 严小韦 | 杨建文 |
| 数据可视化 | 胡丁丁 | 王岩 |
| 数据服务 | 罗丰 | 王岩 |
| 数据工厂 | 罗丰 | 赵飞鸿 |
| 任务调度 | 无 | 崔富凯 |
| 数据治理 | 许俊杰 | 杨建文 |

# 概要设计

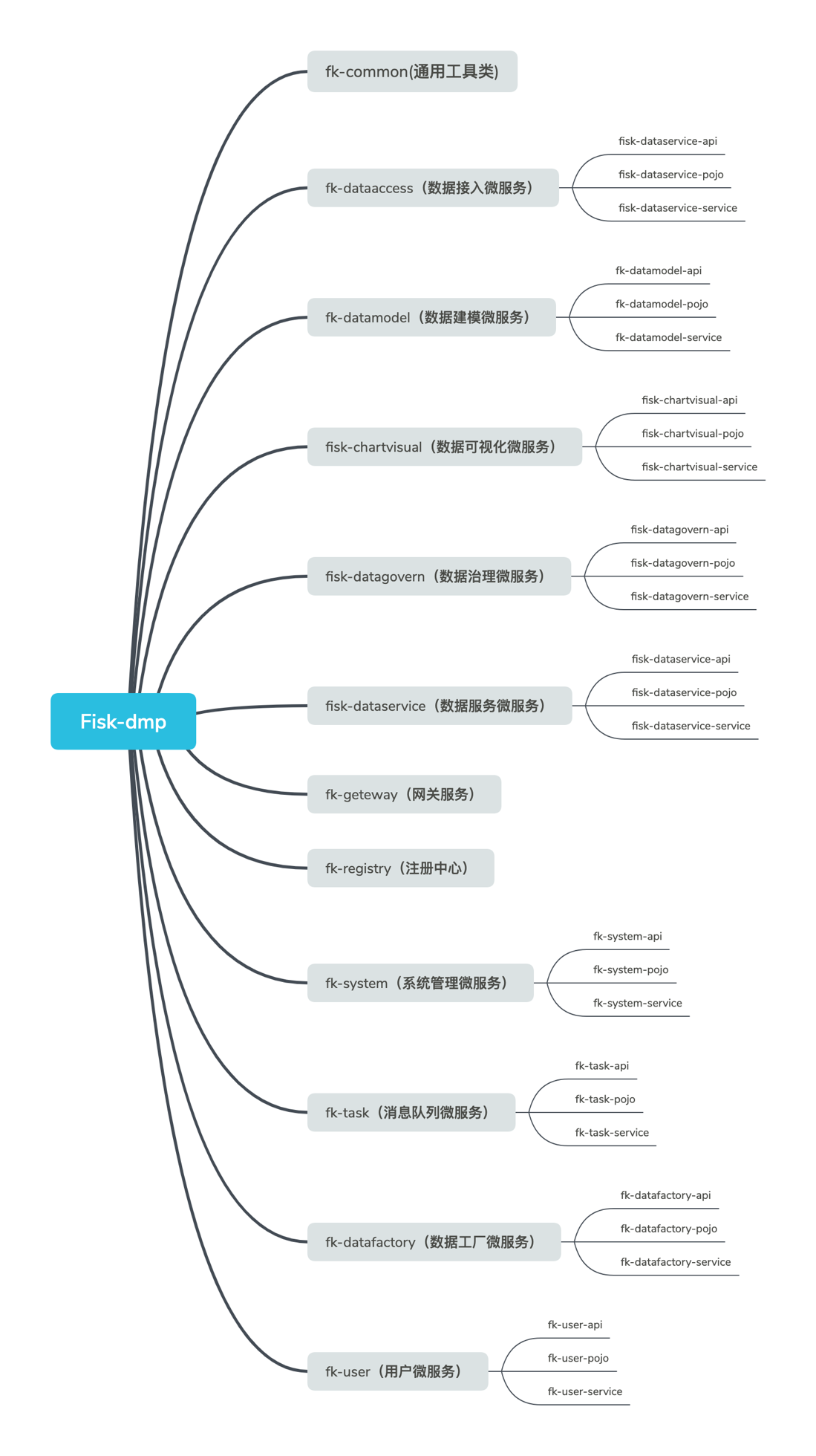
## DMP系统架构

DMP考虑到后期对系统的升级和维护，功能的水平扩展以及开发技术的更新等因素，故后台使用JAVA微服务根据系统业务按照功能拆分为更加细粒度的服务，所拆分的每一个服务都是一个独立的应用，这些应用对外提供公共的API，可以独立承担对外服务的职责。

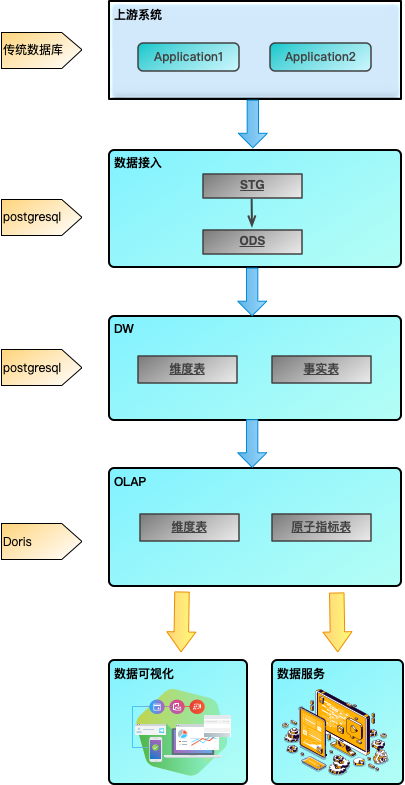
## DMP微服务架构



## DMP后台结构



## DMP系统逻辑架构



## DMP 开源组件

### 后端

1. Apollo 配置中心
2. ELK 日志分析
3. NIFI 数据调度ETL工具
4. RabbitMQ 消息队列
5. Redis 分布式缓存
6. Nginx 负载均衡
7. Atlas 元数据管理

### 前端

1. Echarts：JavaScript开源可视化图标库；
2. Element UI：基于VUE 2.0的前端页面组件库
3. Antv：数据可视化引擎

## 环境说明

### 系统语言

### 系统平台

### 服务器配置

# 详细设计

## 数据结构

### 数据库设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **数据库** | **数据库类型** | **所属微服务** | **描述** | **备注** |
| dmp\_chartvisualt\_db | mysql | 数据可视化 | 数据可视化数据库 |  |
| dmp\_datagovern\_db | mysql | 数据治理 | 数据治理数据库 |  |
| dmp\_datainput\_db | mysql | 数据接入 | 数据接入数据库 |  |
| dmp\_datamodel\_db | mysql | 数据建模 | 数据建模数据库 |  |
| dmp\_dataservice\_db | mysql | 数据服务 | 数据服务数据库 |  |
| dmp\_system\_db | mysql | 系统管理，用户 | 系统管理数据库 |  |
| dmp\_task\_db | mysql | 任务调度 | 任务调度数据库 |  |
| dmp\_datafactory\_db | mysql | 数据工厂 | 数据工厂数据库 |  |
| dmp\_ods | postgresql | 无 | 数据同步的ods表 |  |
| dmp\_dw | postgresql | 无 | 数仓 |  |
| dmp\_dw | doris | 无 | 分析模型 | 分析模型 |

## 模块设计

根据DMP后台微服务的拆分，主要分为以下模块：数据接入，数据建模，数据可视化，数据治理，数据服务，系统管理，系统用户，消息队列，任务调度。

### 数据接入

#### 表设计

#### 数据接入-数据结构（dmp\_datainput\_db）

* 数据源信息（tb\_app\_datasource）

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **备注** |
| id | int | 自增主键 |
| app\_id | varchar(50) | tb\_app\_registration（id） |
| drive\_type | varchar(50) | tb\_app\_drivetype（type） |
| host | varchar(255) | 主机名 |
| port | varchar(20) | 端口 |
| db\_name | varchar(50) | 数据库名 |
| atlas\_db\_id | varchar(255) | atlas\_db\_id |
| connect\_str | varchar(500) | 数据源连接字符串 |
| connect\_account | varchar(50) | 连接账号 |
| connect\_pwd | varchar(255) | 连接密码 |
| realtime\_account | varchar(50) | 验证方式（实时） 登录账号 |
| realtime\_pwd | varchar(50) | 验证方式（实时） 登录密码 |
| create\_time | datetime | 创建时间 |
| create\_user | varchar(20) | 创建人 |
| update\_time | datetime | 更新时间 |
| update\_user | varchar(20) | 更新人 |
| del\_flag | tinyint | 逻辑删除（1未删除，0删除） |

* 数据库类型表（tb\_app\_drivetype）

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **备注** |
| id | int | 自增主键 |
| name | varchar(50) | 数据源名称 |
| connect\_str | vatchar(255) | 数据源连接字符串模板 |

* 应用注册表（tb\_app\_registration）、

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **备注** |
| id | int | 自增主键 |
| atlas\_instance\_id | varchar(255) | atlas实例id |
| component\_id | vatchar(255) | 组件id |
| target\_db\_pool\_component\_id | vatchar(255) | targetDbPoolComponentId |
| source\_db\_pool\_component\_id | vatchar(255) | sourceDbPoolComponentId |
| app\_name | varchar(200) | 应用名称 |
| app\_abbreviation | varchar(200) | 应用简称 |
| app\_des | varchar(255) | 应用描述 |
| app\_type | tinyint | 应用类型 (0:实时应用 1:非实时应用) |
| app\_principal | varchar(50) | 应用负责人 |
| app\_principal\_email | varchar(100) | 应用负责人邮箱 |
| sync\_mode | varchar(255) | 同步方式 |
| expression | varchar(255) | 表达式 |
| msg | varchar(500) | 日志 |
| create\_time | datetime | 创建时间 |
| create\_user | varchar(20) | 创建人 |
| update\_time | datetime | 更新时间 |
| update\_user | varchar(20) | 更新人 |
| del\_flag | tinyint | 逻辑删除（1未删除，0删除） |

* 数据同步增量表（tb\_etl\_Incremental）

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **备注** |
| id | int | 自增主键 |
| object\_name | varchar(255) | 数据同步流程的表名 |
| enable\_flag | tinyint | 同步流程是否启动,1启动 2停止 |
| incremental\_objectivescore\_batchno | varchar(50) | 预留字段；数据同步批次号 |
| incremental\_objectivescore\_start | datetime | 最近一次同步的数据的时间范围的开始时间 |
| incremental\_objectivescore\_end | datetime | 最近一次同步的数据的时间范围的结束时间 |
| createtime | datetime | 创建时间 |
| del\_flag | tinyint | 逻辑删除（1未删除，0删除） |

* 数据同步日志表（tb\_etl\_log）

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **备注** |
| id | int | 自增主键 |
| tablename | varchar(255) | 定义更新的表名 |
| startdate | datetime | 表数据开始同步时间 |
| enddate | datetime | 表数据同步结束时间 |
| datarows | int | 本次同步的数据行 |
| status | tinyint | 状态；0代表正在同步，1代表同步成功，2代表同步失败 |
| code | varchar(50) | 标识code |
| errordesc | varchar(500) | 记录同步失败的原因 |
| createtime | datetime | 创建时间 |

* 表连接信息表（tb\_table\_access）

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **备注** |
| id | int | 自增主键 |
| app\_id | int | app\_registration（id） |
| atlas\_table\_id | varchar(255) | atlas\_table\_id(添加数据时后台生成) |
| component\_id | varchar(255) | 组件id |
| scheduler\_component\_id | varchar(255) | 调度组件id |
| doris\_select\_sql\_str | text | nifi sql |
| table\_name | varchar(255) | 物理表名 |
| table\_des | varchar(255) | 物理表描述 |
| sync\_src | varchar(500) | 如果是实时物理表，需要提供数据同步地址 |
| is\_realtime | tinyint | 0是实时物理表，1是非实时物理表 |
| create\_time | datetime | 创建时间 |
| create\_user | varchar(20) | 创建人 |
| update\_time | datetime | 更新时间 |
| update\_user | varchar(20) | 更新人 |
| del\_flag | tinyint | 逻辑删除（1未删除，0删除） |

* 表数据同步-业务时间覆盖（tb\_table\_business）

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **备注** |
| id | int | 自增主键 |
| access\_id | int | tb\_table\_access(id) |
| business\_time\_field | varchar(20) | 业务时间字段 |
| business\_flag | int | 1: 取上一个月数据,覆盖上一个月数据  2: 取当月数据,覆盖当月数据  3: 当月  4: 取上一年数据,覆盖上一年  5: 取当年数据,覆盖当年 |
| business\_day | int | business\_flag3选项,当月具体多少号 |

* 表字段（tb\_table\_fields）

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **备注** |
| id | int | 自增主键 |
| table\_access\_id | int | table\_access（id） |
| atlas\_field\_id | varchar(255) | atlas\_field\_id(添加数据时后台生成) |
| field\_name | varchar(255) | 字段名称 |
| field\_des | varchar(255) | 字段描述 |
| field\_type | varchar(50) | 字段类型 |
| field\_length | int | 字段长度 |
| is\_primarykey | int | 1是主键，0非主键 |
| is\_businesstime | int | 1是业务时间，0非业务时间 |
| is\_timestamp | int | 1是时间戳，0非时间戳 |
| is\_realtime | int | 1：是实时物理表的字段，  0：非实时物理表的字段 |
| create\_time | datetime | 创建时间 |
| create\_user | varchar(20) | 创建人 |
| update\_time | datetime | 更新时间 |
| update\_user | varchar(20) | 更新人 |
| del\_flag | tinyint | 逻辑删除（1未删除，0删除） |

* 表数据同步逻辑（tb\_table\_syncmode）

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **备注** |
| id | int | 自增主键 |
| sync\_mode | int | 同步方式；1：全量、2：时间戳增量、3：业务时间覆盖、4：自定义覆盖；  全量：每次都全量覆盖整张表；  时间戳增量：根据表字段中配置的业务主键和时间戳字段进行增量覆盖；  业务时间覆盖：根据配置的业务时间字段进行业务时间覆盖： |
| sync\_field | varchar(20) | 时间戳字段(增量字段) |
| custom\_delete\_condition | text | 自定义删除条件：定义每次同步的时候删除我们已有的数据条件： |
| custom\_insert\_condition | text | 自定义插入条件：定义删除之后获取插入条件的数据进行插入 |
| timer\_driver | varchar(255) | timer driver |
| corn\_expression | varchar(255) | corn表达式 |

#### 应用注册

##### 页面设计

图1

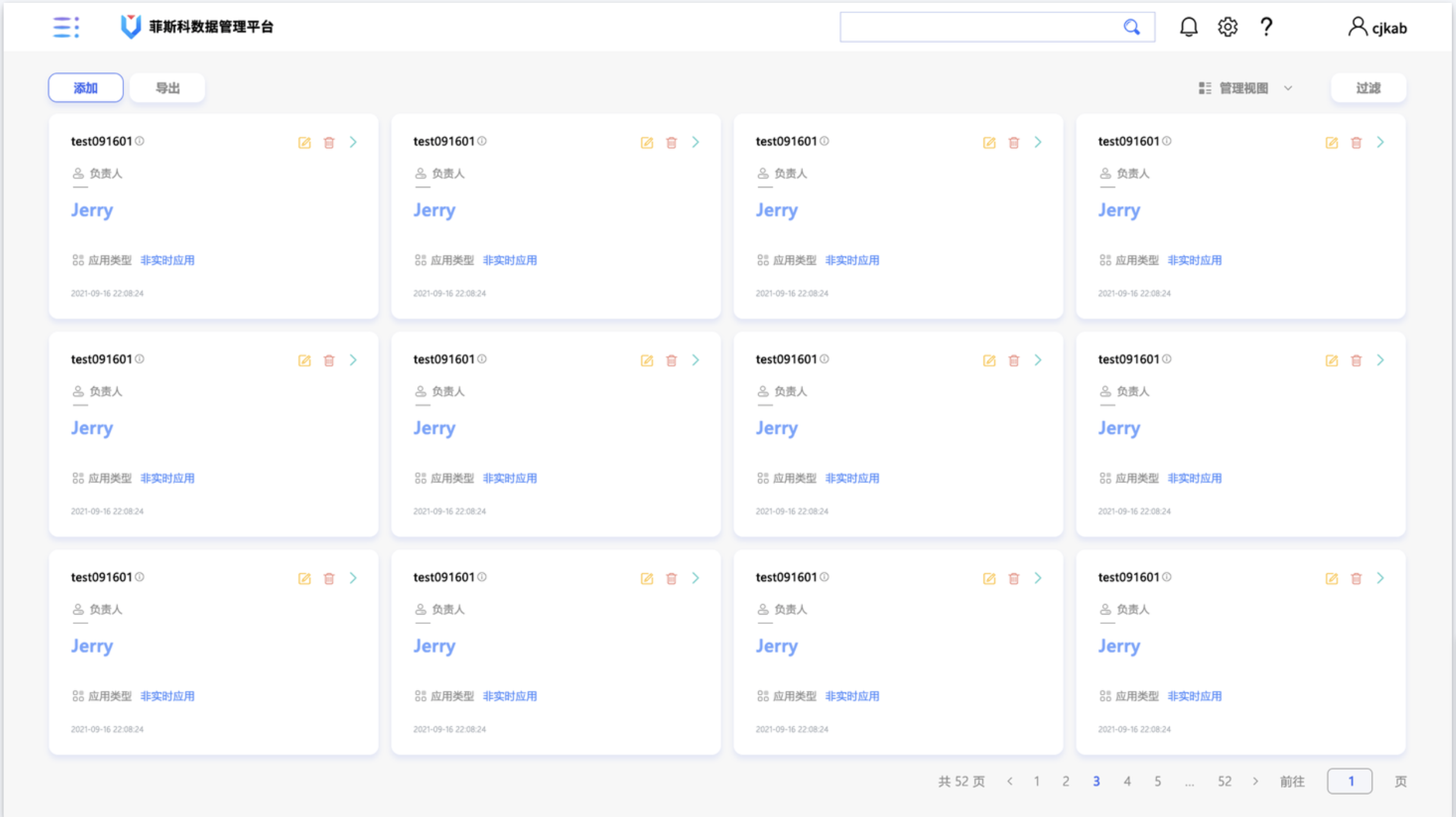


图2

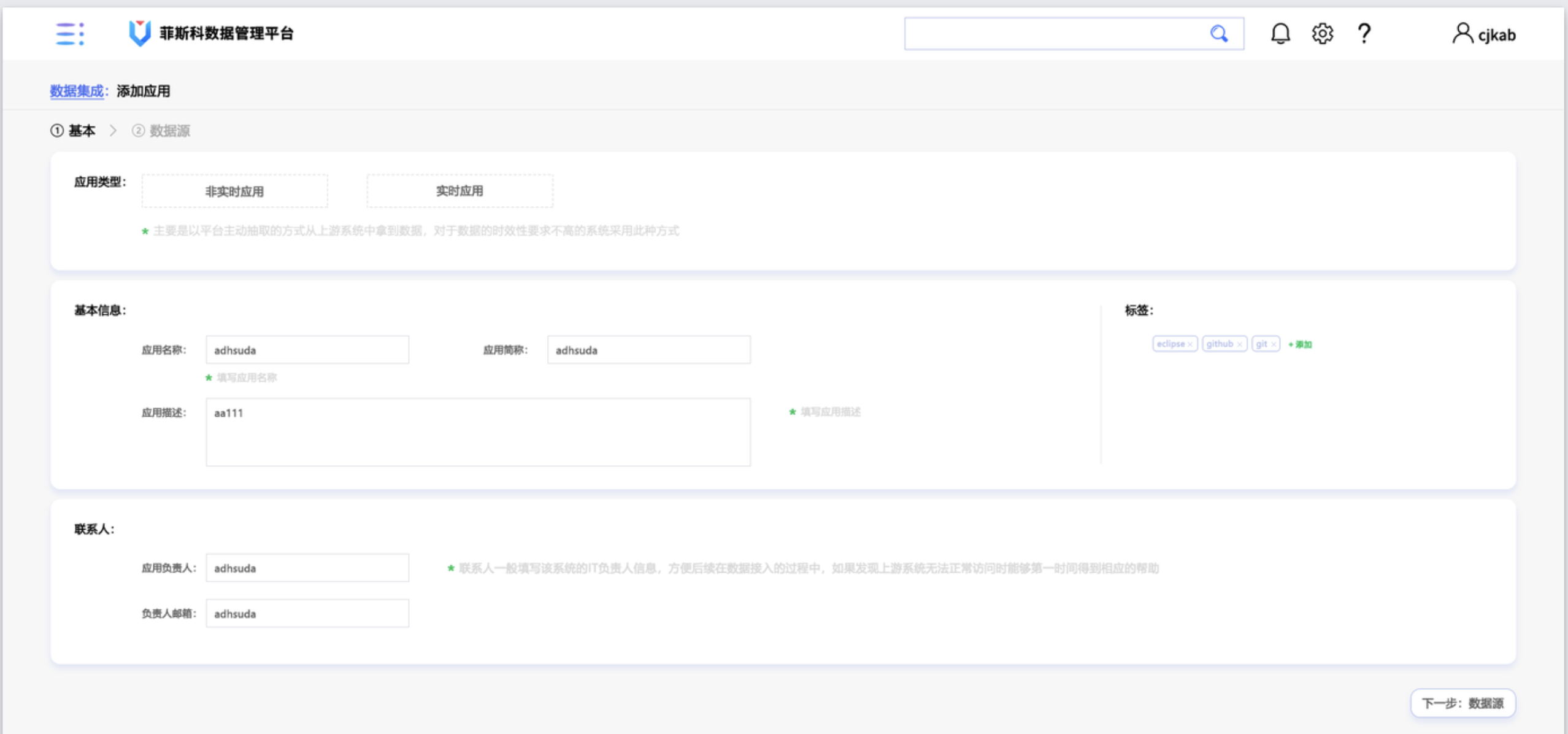
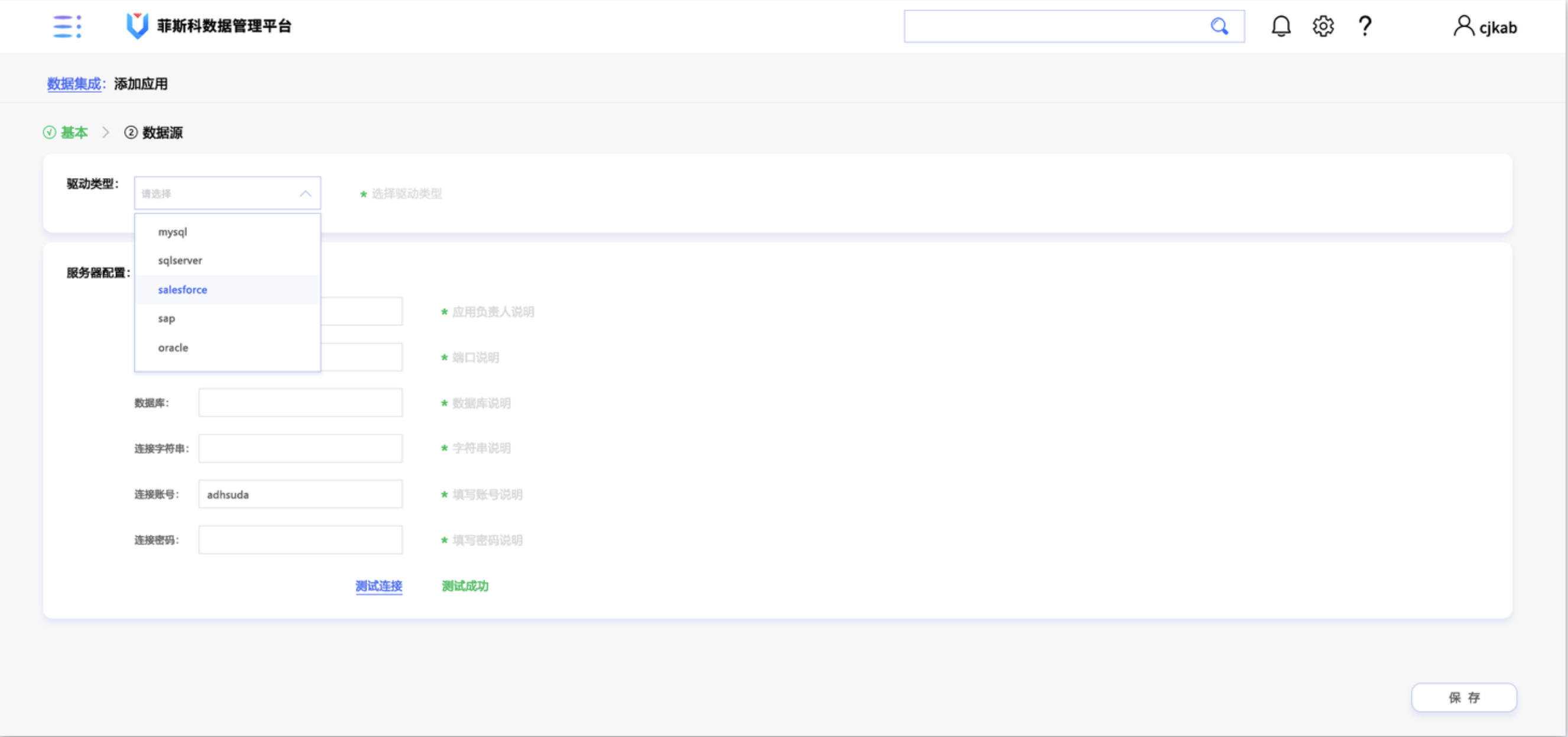


图3



数据接入模块现阶段支持接入的数据源类型有以下几种：

1. MySQL
2. Sqlserver
3. Oracle

##### 接口设计

|  |  |
| --- | --- |
| 接口路径 | 备注 |
|  |  |
|  |  |

#### 物理表

##### 页面设计

### 数据建模

#### 表设计

#### 数据建模-数据结构（dmp\_datamodel\_db）

* 业务域（tb\_area\_business）

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **备注** |
| id | int | 自增主键 |
| business\_name | varchar(50) | 业务域名称 |
| business\_des | varchar(255) | 业务域描述 |
| business\_admin | varchar(50) | 业务需求管理员 |
| business\_email | varchar(100) | 应用负责人邮箱 |
| is\_publish | int | 1:未发布、2：发布成功、3：发布失败 |
| publish\_time | datetime | 发布时间 |
| create\_time | datetime | 创建时间 |
| create\_user | varchar(20) | 创建人 |
| update\_time | datetime | 更新时间 |
| update\_user | varchar(20) | 更新人 |
| del\_flag | tinyint | 逻辑删除（1未删除，0删除） |

* 业务域数据源（tb\_area\_datasource）

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **备注** |
| id | int | 自增主键 |
| datasource\_name |  | 数据源名称 |
| datasource\_des |  | 数据源描述 |
| database\_name |  | 数据库名称 |
| datasource\_address |  | 数据源连接地址 |
| datasource\_account |  | 数据源连接账号 |
| datasource\_pwd |  | 数据源连接密码 |
| create\_time | datetime | 创建时间 |
| create\_user | varchar(20) | 创建人 |
| update\_time | datetime | 更新时间 |
| update\_user | varchar(20) | 更新人 |
| del\_flag | tinyint | 逻辑删除（1未删除，0删除） |

* 业务限定（tb\_business\_limited）
* 业务限定字段（tb\_business\_limited\_attribute）
* 业务过程（tb\_business\_process）
* 维度表（tb\_dimension）
* 维度表字段（tb\_dimension\_attribute）
* 维度表目录（tb\_dimension\_folder）
* 事实表（tb\_fact）
* 事实表字段（tb\_fact\_attribute）
* 事实表同步方式（tb\_fact\_sync\_mode）
* 原子指标（tb\_indicators）

### 数据可视化

#### 表设计

#### 数据可视化结构（dmp\_chartvisual\_db）

* 可视化图表（tb\_char）

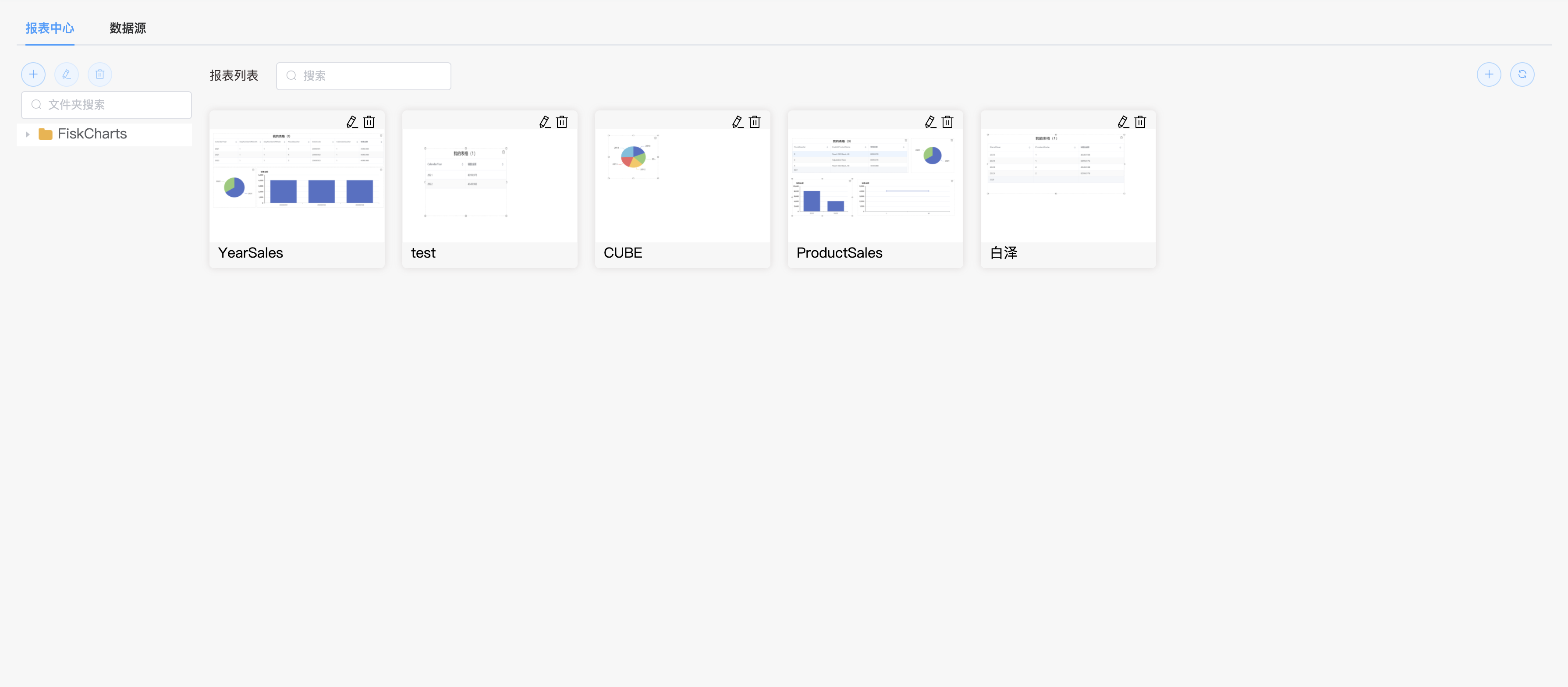
|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **备注** |
| id | int | 主键 |
| fid | int | 文件夹id |
| name | varchar(50) | 报表名称 |
| content | varchar(10000) | 图表格式（json字符串） |
| detail | varchar(200) | 报表描述 |
| image | longblob | 报表缩略图 |
| create\_user | varchar(20) | 创建人 |
| create\_time | datetime | 创建时间 |
| update\_user | varchar(20) | 更新人 |
| update\_time | datetime | 更新时间 |
| del\_flag | tinyint | 逻辑删除标识（1未删除，0删除） |

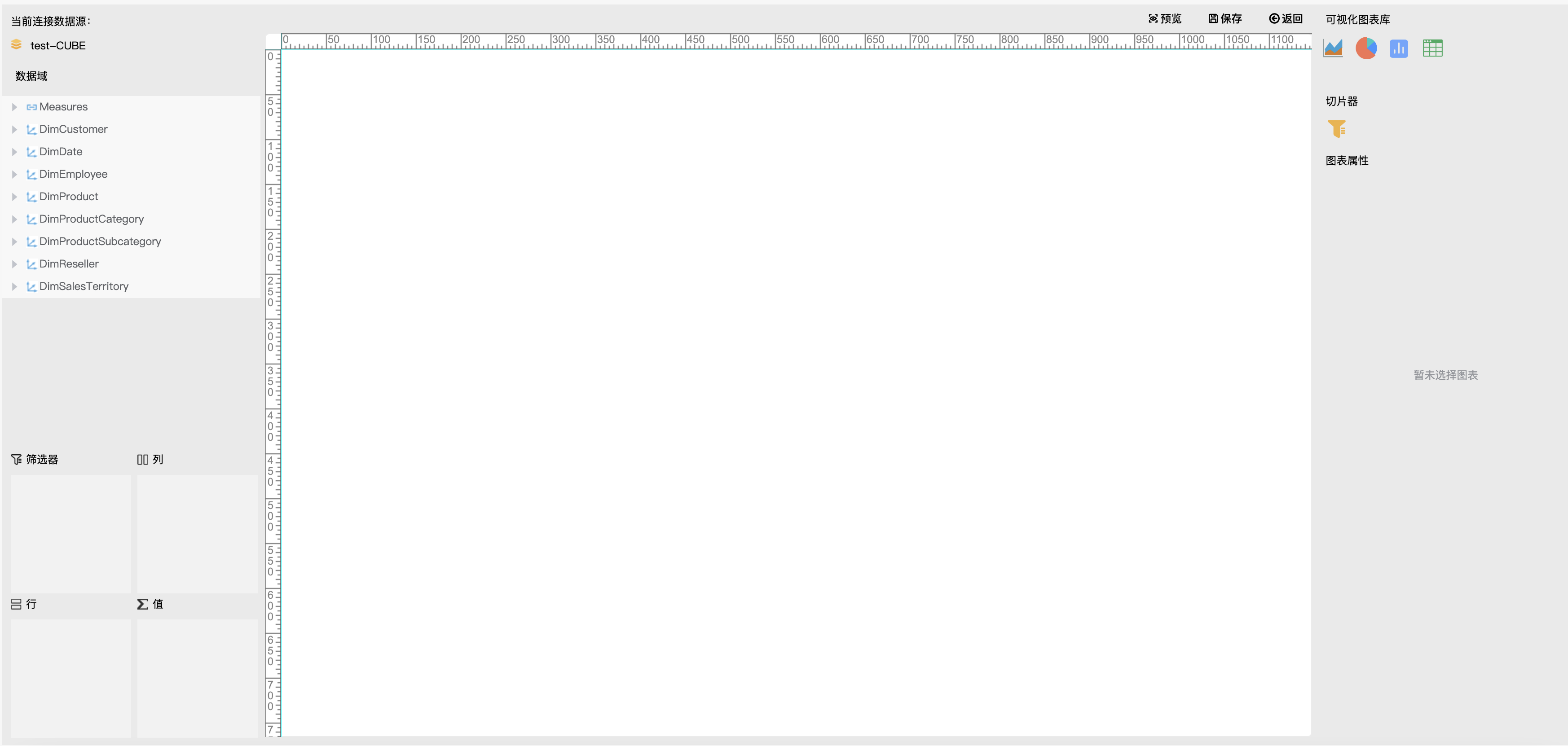
* 数据连接表（tb\_datasource\_con）

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **备注** |
| id | int | 主键 |
| name | varchar(50) | 连接名称 |
| con\_str | varchar(500) | 连接字符串 |
| con\_ip | varchar(50) | ip |
| con\_port | int | 端口 |
| con\_cube | varchar(50) | 模型 |
| con\_dbname | varchar(50) | 数据库名称 |
| con\_type | tinyint | 连接类型 |
| con\_account | varchar(50) | 连接账号 |
| con\_password | varchar(50) | 连接密码 |
| create\_time | datetime | 创建时间 |
| update\_user | varchar(20) | 更新人 |
| update\_time | datetime | 更新时间 |
| del\_flag | tinyint | 逻辑删除标识（1未删除，0删除） |

#### 制作数据可视化报表

##### 页面设计





#### 添加自定义视图组件

##### 添加地图视图组件

地图视图效果：



地图数据结构：

var seriesData = [{

name: '北京',

value: 100

}, {

name: '天津',

value: 0

}, {

name: '上海',

value: 60

}, {

name: '重庆',

value: 0

}

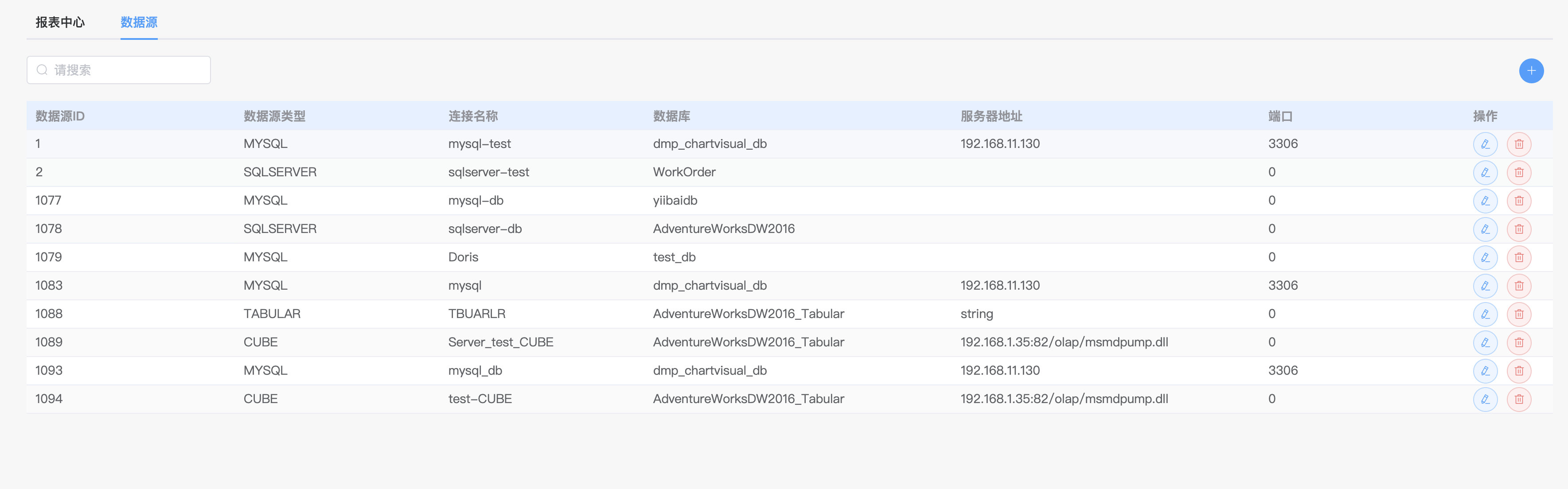
………

];

中国地图json文件：https://www.jq22.com/demo/echarts-china-map-master201811192226/js/map/china.js

#### 添加可视化数据集

##### 页面设计



### 数据治理

### 数据服务

#### 表设计

#### 数据服务结构（dmp\_dataservice\_db）

* API配置（api\_configure）
* API配置字段（tb\_configure\_field）
* API配置用户权限（configure\_user）

### 系统管理

### 系统用户

### 消息队列

### 数据工厂

#### 表设计

#### 数据工厂结构（dmp\_datafactory\_db）

* NIFI组件类型表（tb\_nifi\_components）

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **备注** |
| id | int | 主键 |
| name | varchar(50) | 组件名称 |

* 自定义NIFI数据管道表（tb\_nifi\_custom\_workflow）

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **备注** |
| id | int | 自增主键 |
| workflowid | varchar(50) | 主键GUID |
| workflowname | varchar(200) | 管道名称 |
| status | tinyint | 状态：已发布（1）、未发布（0） |
| create\_user | varchar(100) | 创建人 |
| create\_time | datetime | 创建时间 |
| update\_user | varchar(100) | 更新人 |
| update\_time | datetime | 更新时间 |
| del\_flag | tinyint | 逻辑删除（1未删除，0删除） |

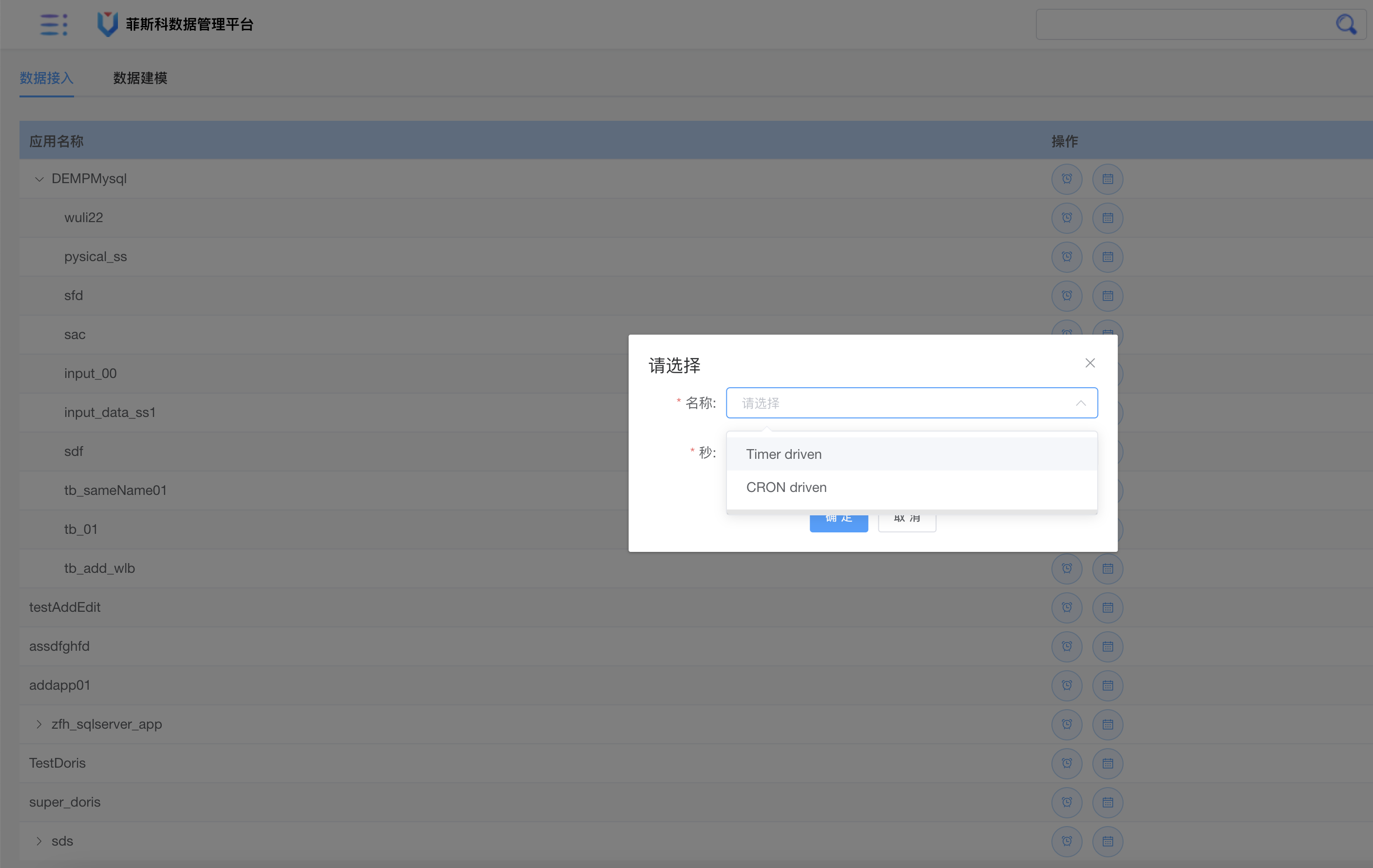
* 自定义NIFI数据管道详细表（tb\_nifi\_custom\_workflow\_detail）

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **备注** |
| id | int | 自增主键 |
| workflowid | varchar(50) | tb\_nifi\_custom\_workflow（workflowid） |
| componentsid | int | tb\_nifi\_components（id） |
| tableid | varchar(50) | 关联的表id |
| schedule | tinyint | 调度类型，1 timer driven、2 cron表达式 |
| script | varchar(50) | 调度的脚本，如果是timer driven 则值应是秒数，如果是cron driven则值应是cron表达式 |
| inport | varchar(50) | 串行入口 |
| outport | varchar(50) | 串行出口 |
| create\_user | varchar(100) | 创建人 |
| create\_time | datetime | 创建时间 |
| update\_user | varchar(100) | 更新人 |
| update\_time | datetime | 更新时间 |
| del\_flag | tinyint | 逻辑删除（1未删除，0删除） |

#### 单表调度

##### 页面设计





在DMP中完成数据接入、数据建模等一系列操作之后即可到数据调度页面对之前接入的数据表设置定时同步频率。

##### 调度方式

对于数据表的调度频率有两种方式：

1. 时间驱动（Timer Driven）
2. Cron 表达式（Cron Driven）

时间驱动可以设置每隔多少秒执行一次调度。

Cron表达式可以设置定时调度



#### 数据管道

##### 页面设计

管道首页：

图形用户界面, 应用程序

描述已自动生成

管道配置页面：

图形用户界面, 表格

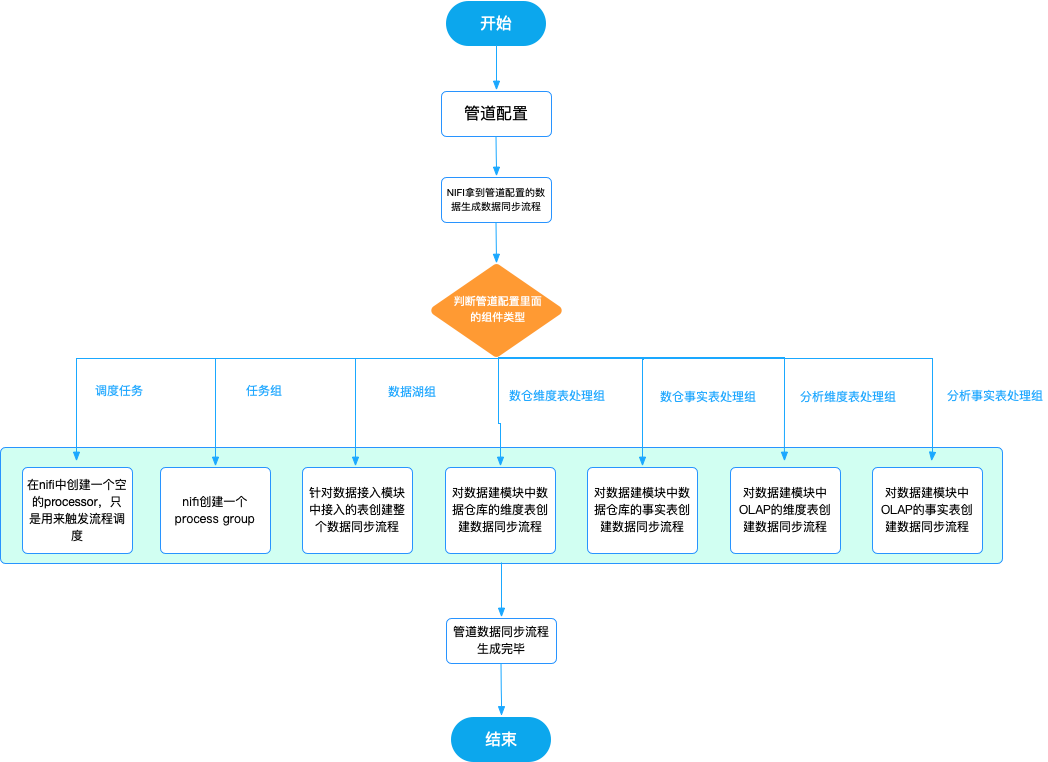
描述已自动生成

在DMP管道设置中可以针对多表串行数据同步进行配置能够按照设定的流程从上游系统直接流入到最终的分析建模中，以便报表访问人员能够在报表中直观的看到数据的变化，确保了数据流的连续性。

##### 管道配置-NIFI组件类型

|  |  |
| --- | --- |
| **组件** | **备注** |
| 调度任务 | 用来设置触发数据同步的组件-processor |
| 活动组（任务组） | processor group |
| 数据湖组 | 对应目前数据接入这块一张表的整个数据同步流程。 |
| 数仓维度表处理组 | 数据仓库维度表的数据同步流程-processor |
| 数仓事实表处理组 | 数据仓库事实表的数据同步流程-processor |
| 分析维度表处理组 | Doris维度表的数据同步流程 |
| 分析事实表处理组 | Doris事实表的数据同步流程 |

##### NIFI生成规则



##### 接口设计

|  |  |
| --- | --- |
| **接口** | **备注** |
| nifiCustomWorkflow/add | 添加管道 |
| nifiCustomWorkflow/delete | 删除管道 |
| nifiCustomWorkflow/edit | 编辑管道 |
| nifiCustomWorkflow/get | 获取管道 |
| nifiCustomWorkflow/getColumn | 获取管道过滤器表字段 |
| nifiCustomWorkflow/pageFilter | 管道筛选器 |
|  |  |

### 数据治理

### 系统管理

##### 表设计

##### 系统管理结构（dmp\_system\_db）

* 授权白名单（tb\_authenticate\_whitelist）
* 客户端信息（tb\_client\_info）
* 系统角色（tb\_role\_info）
* 角色与服务对应关系（tb\_role\_service\_assignment）
* 用户与角色对应关系（tb\_role\_user\_assignment）
* 服务注册（tb\_service\_registry）
* 系统用户（tb\_user\_info）