

ALIBABA CLOUD

阿里云

工业互联网平台
最佳实践

文档版本：20210309

阿里云

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或惩罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。未经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
 危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 危险 重置操作将丢失用户配置数据。
 警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告 重启操作将导致业务中断，恢复业务时间约十分钟。
 注意	用于警示信息、补充说明等，是用户必须了解的内容。	 注意 权重设置为0，该服务器不会再接受新请求。
 说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击设置>网络>设置网络类型。
粗体	表示按键、菜单、页面名称等UI元素。	在结果确认页面，单击确定。
Courier字体	命令或代码。	执行 cd /d C:/window 命令，进入 Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid</code> <code>Instance_ID</code>
[] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{} 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

目录

1.工业应用与ERP集成 -----	05
2.工业数据开发实践 -----	14

1. 工业应用与ERP集成

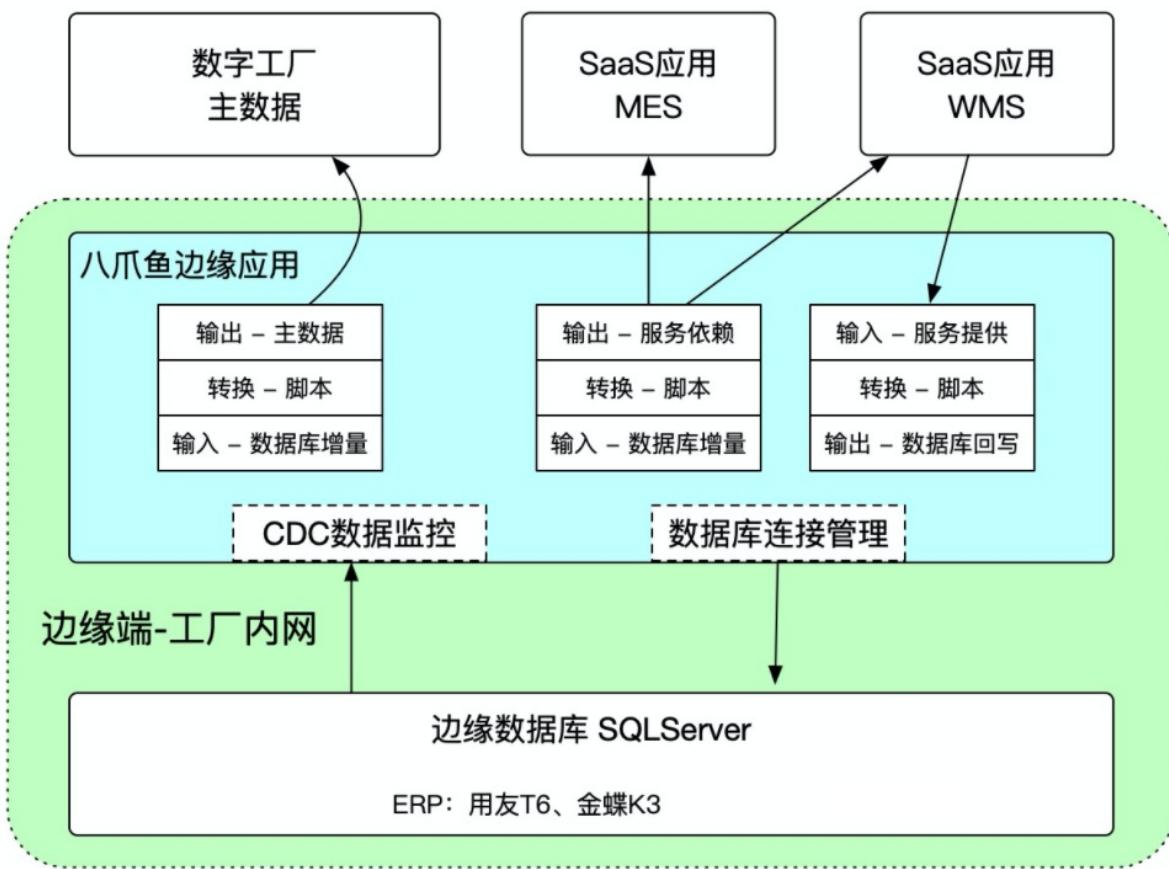
通过阿里云工业互联网平台集成的工业应用在项目交付过程中，经常会遇到需要与企业已有的（ERP）系统进行数据集成，本篇最佳实践介绍了怎么用使用平台的边缘数据集成功能通过最少改造工作，来快速集成。

场景一：部署在企业内部的ERP数据集成

场景描述

安装在企业内网的传统ERP系统（例如用友T6/U8，金蝶K3等），在不改造现有的ERP系统的前提下，将基础数据和业务数据同步到数据工厂以及数字工厂上运行的工业应用。

架构示意图



配置数据集成任务

第一步：数据源定义

参考文档定义[数据库类型数据源](#)。数据源启用增量监控，并且按照指引文档配置数据库。

第二步：同步基础数据到数字工厂主数据

一般从ERP中同步到数字工厂的主数据包括：人员、供应商、客户、物料、物料类型、计量单位。以人员主数据为例，[边缘数据集成](#)中新建一个边缘数据集群任务，任务的输入节点选择[数据库增量](#)类型，任务的转换节点把数据转换成人员主数据要求的格式，任务的输出节点选择[主数据](#)类型。



(1) 输入节点：数据库增量

数据库增量 X

* 数据源：
金蝶k3 ▼

* 增量表名：
dbo.t_Base_Emp

第一次启动是否全量同步
 是 否

确认 取消

(2) 配置转换节点

```
/**  
 * 客户实现: transform()转换函数  
 * 这是脚本转换器的第一个转换函数,负责将 数据来源(如数据库变换的数据) 转换成 目标数据格式,如主数据,服务模型数  
据,物联网数据等  
 * 示例代码是数据源的数据转换成iot工业数字工厂的人员主数据格式  
 * @param eventType为事件类型,对于数据库事件有 insert,update,delete  
 * @param msgPayload 为需要转换的目标数据,注意其格式是一个Map,对应Java的格式为Map<String, Object>  
 * @context 系统上下文参数  
 * @return 返回转换后的数据格式,必须也为一个Map,即数据输入为Map,转换格式后还是为Map,如果返回值是null,  
这条数据将被忽略。  
 */  
function transform(eventType, msgPayload, context) {  
    let masterData = new iotPluginApi.newMap();  
    masterData["name"] = msgPayload["FName"];  
    masterData["id"] = msgPayload["FIItemID"];  
    masterData["phone"] = msgPayload["FMobilePhone"];  
    masterData["postcode"] = msgPayload["cPersonEmail"];  
    return masterData;  
}
```

(3) 输出节点：主数据

主数据



* 主数据：

人员

选择

确认

取消

完成以上三个节点的配置后，保存当前编辑中的边缘数据集成并完成下载配置就可以生效了，系统将监控边缘端的数据库中t_Base_Emp表中数据变化，然后同步到数字工厂中的人员主数据中，并推送给实现**主数据集成**的工业应用。

第三步：同步ERP的数据到数字工厂中MES应用

同步ERP的生产工单到MES为例，**边缘数据集成**中新建一个边缘数据集群任务，任务的输入节点选择**数据库增量**类型，任务的转换节点把数据转换成服务模型要求的格式，任务的输出节点选择**服务依赖**类型。

生产工单	数据库增量	配置	脚本转换	配置	服务依赖	配置	✓ 配置完成	复制	删除	调试	全量操作	查看集成日志
------	-------	----	------	----	------	----	--------	----	----	----	------	--------

(1) 输入节点：数据库增量

数据库增量



* 数据源：

金蝶k3



* 增量表名：

dbo.ICMO

第一次启动是否全量同步

 是 否

确认

取消

(2) 配置转换节点

```
/*
 * 客户实现: transform()转换函数
 * 这是脚本转换器的第一个转换函数,负责将 数据来源(如数据库变换的数据) 转换成 目标数据格式,如主数据,服务模型数
据,物联网数据等
 * 如下的代码是一个示例:仅供参考,需要根据您的实际业务场景转换逻辑进行编写
 * 示例代码是数据源的数据转换成iot工业数字工厂的人员主数据格式
 * @param eventType为事件类型,对于数据库事件有insert,update,delete
 * @param msgPayload 为需要转换的目标数据,注意其格式是一个Map,对应Java的格式为Map<String, Object>
 * @context 系统上下文参数
 * @return 返回转换后的数据格式,必须也为一个Map,即数据输入为Map,转换格式后还是为Map
 */
function transform(eventType, msgPayload, context) {
    let masterData = new iotPluginApi.newMap();
    let fstatus = msgPayload["FStatus"];
    let checkDate= msgPayload["FCheckDate"];//制定时间
    if(checkDate > 1590940800000) {
        if(fstatus == 1){
            let planCommitDate = msgPayload["FPlanCommitDate"];
            masterData["workorderCode"] = msgPayload["FBillNo"];
            masterData["planStartTime"] = planCommitDate;//计划开工日期
            masterData["planEndTime"] = msgPayload["FPlanFinishDate"];//计划完工日期
            masterData["departmentCode"] =msgPayload["FWorkShop"];//生产部门
            masterData["quantity"] = msgPayload["FAuxQty"];//生产数量
            masterData["eventType"] =eventType;
            let fItemID = msgPayload["FItemID"];//存货编码
            if(eventType === 'delete'){
                masterData["materialCode"] = fItemID;
            } else {
                var invCode = fItemID;
                var sql = "SELECT * FROM t_ICItem where FItemID=" + invCode;
                //查询数据库关联数据
                var result = iotPluginApi.iotSqlSelect(sql);
                console.log("获取到存货数据: " + result);
                if (result.code == 200 && result.data.length > 0) {//获取存货编码
                    let materialCode = (result.data)[0].FNumber;
                    masterData["materialCode"] = materialCode;
                }
            }
            return masterData;
        }
        //过期的数据丢掉不上报, 直接return null
        return null;
    }
}
```

(3) 输出节点：服务依赖

服务依赖

选择已有服务接口 自定义脚本

数据源：

默认



* 服务模型：

ManufacturingExecutionService

选择

* 接口：

getWorkorder

选择

完成以上三个节点的配置后，保存当前编辑中的边缘数据集成并完成下载配置就可以生效了。

(4) 定义服务集成

对于服务依赖的节点，需要定义[服务集成](#)确认服务的提供方和服务依赖方。新建一个集成配置，服务依赖方选择[工业八爪鱼](#)，服务选择[生产执行服务](#)服务提供方选择[数字工厂](#)上开通的MES应用。

添加集成配置 X

集成列表

工业八爪鱼-cb59c2

生产执行服务

调用

精工云Mes生产管控...

生产执行服务

删除

+添加

保存

取消

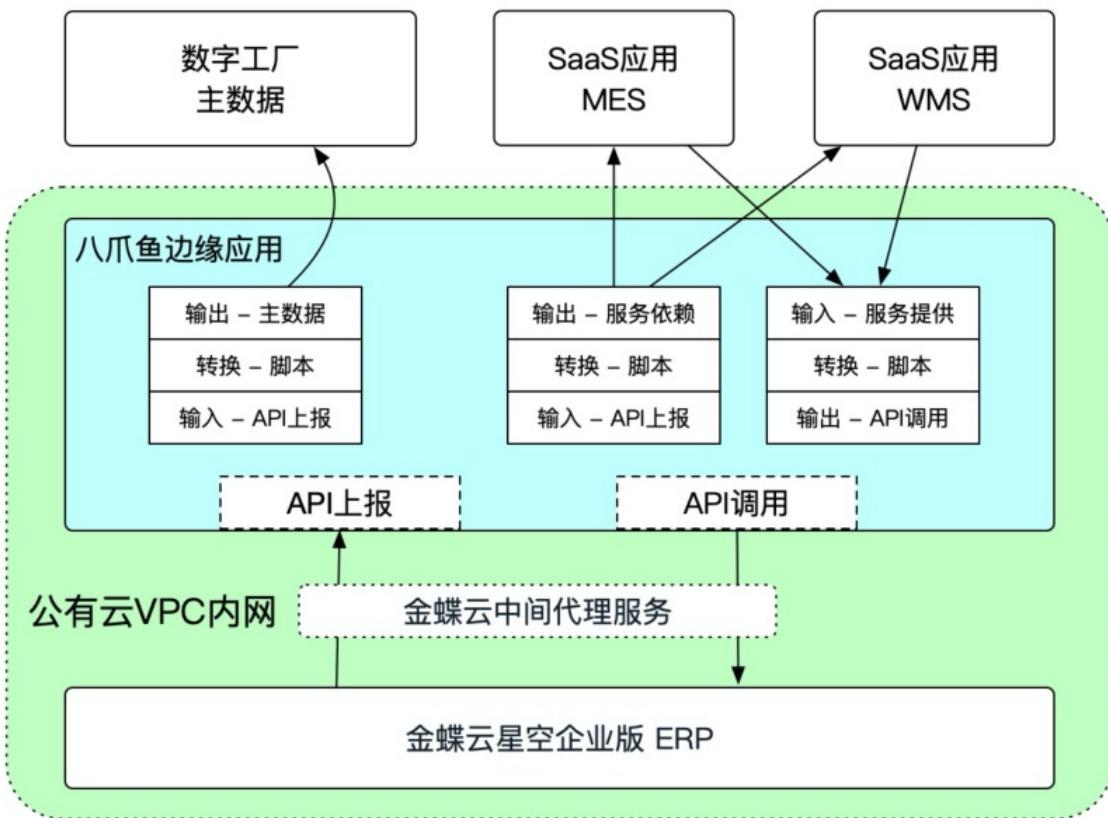
系统将监控边缘端的数据库中ICMO表中数据变化，转换后调用MES提供的getWorkorder接口。

场景二：部署公有云上的SaaS化的ERP数据集成

场景描述

ERP系统为公有云的SaaS应用服务端（例如金蝶云），企业购买开通以后没办法直接访问到数据库，这种往往通过开放OpenAPI的方式进行数据集成，一般采用定时访问API（不需要修改ERP代码）和实现一个HTTP接口由ERP的代理应用来调用上报数据。

架构示意图



配置数据集成任务

第一步：数据源定义

参考文档定义消息队列类型数据源。

第二步：同步基础数据到数字工厂主数据

一般从ERP中同步到数字工厂的主数据包括：人员、供应商、客户、物料、物料类型、计量单位。以人员主数据为例，[边缘数据集成](#)中新建一个边缘数据集群任务，任务的输入节点选择API上报类型，任务的转换节点把数据转换成人员主数据要求的格式，任务的输出节点选择主数据类型，ERP的代理应用调用数字工厂发布的API上报人员主数据。



(1) 输入节点：API上报，选择API类型的数据源，定义一个API的路径，比如取名/reportData

API上报



* 数据源：

APIDataSource



URL: http://192.168.1.102:8089

* path:

/reportData

确认

取消

ERP的代理应用就可以调用http://192.168.1.102:8089/reportData，上报人员主数据；

(2) 配置转换节点，把接口上传的参数结构体数据结构转换成数字工厂定义的人员主数据结构；

(3) 输出节点：主数据

主数据



* 主数据：

人员

选择

确认

取消

完成以上三个节点的配置后，保存当前编辑中的边缘数据集成并完成下载配置就可以生效了，系统将把ERP代理应用上报的数据，然后同步到数字工厂中的人员主数据中，并推送给实现主数据集成的工业应用。

第三步：同步ERP的数据到数字工厂中MES应用

同步ERP的生产工单到MES为例，边缘数据集成中新建一个边缘数据集群任务，任务的输入节点选择API查询类型，任务的转换节点把数据转换成服务模型要求的格式，任务的输出节点选择服务依赖类型。

同步工单

API查询

配置

脚本转换

配置

服务依赖

配置

✓ 配置完成

复制 删除 调试 全量操作 查看集成日志

(1) 输入节点：API查询，设置定时查询的周期，比如每10分钟查询一次（0 */1 * * * ?），选择API类型的数据源以及API的接口访问路径

API查询

* 定时设置：[?](#)

0 */10 * * * ?

* 数据源：

金蝶Cloud

URL: 192.45.6.34

* path:

/reportEmployee

(2) 配置转换节点

(3) 输出节点：服务依赖

服务依赖

选择已有服务接口 自定义脚本

数据源：

默认

* 服务模型：

ManufacturingExecutionService

[选择](#)

* 接口：

getWorkorder

[选择](#)

完成以上三个节点的配置后，保存当前编辑中的边缘数据集成并完成下载配置就可以生效了。

(4) 定义服务集成

对于服务依赖的节点，需要定义**服务集成**确认服务的提供方和服务依赖方。新建一个集成配置，服务依赖方选择**工业八爪鱼**，服务选择**生产执行服务**服务提供方选择数字工厂上开通的MES应用。



系统将每10分钟调用ERP开放的接口把查询的数据结果转换后调用MES提供的getWorkorder接口。

2. 工业数据开发实践

工业互联网平台企业级平台（数字工厂）提供标准官方应用给制造企业用户进行业务数据的分析和可视化展示，如果标准官方应用不能满足客户要求，企业业务用户可针对自己的业务需求进行个性化的数据分析和可视化展示。为满足个性化的数据分析，企业级平台提供[数据开发](#)工具，对收集到的工业数据多维度分析，最后通过[Quick BI](#)和[DataV](#)完成数据可视化开发，数据可视化结果最终集成到[经营驾驶舱](#)中进行配置使用。本开发实践以统计工厂每月每款产品的销售金额为例，介绍如何采集边缘信息系统数据、完成数据开发和结果可视化展示的完整过程。

销售金额计算逻辑

销售订单ER图



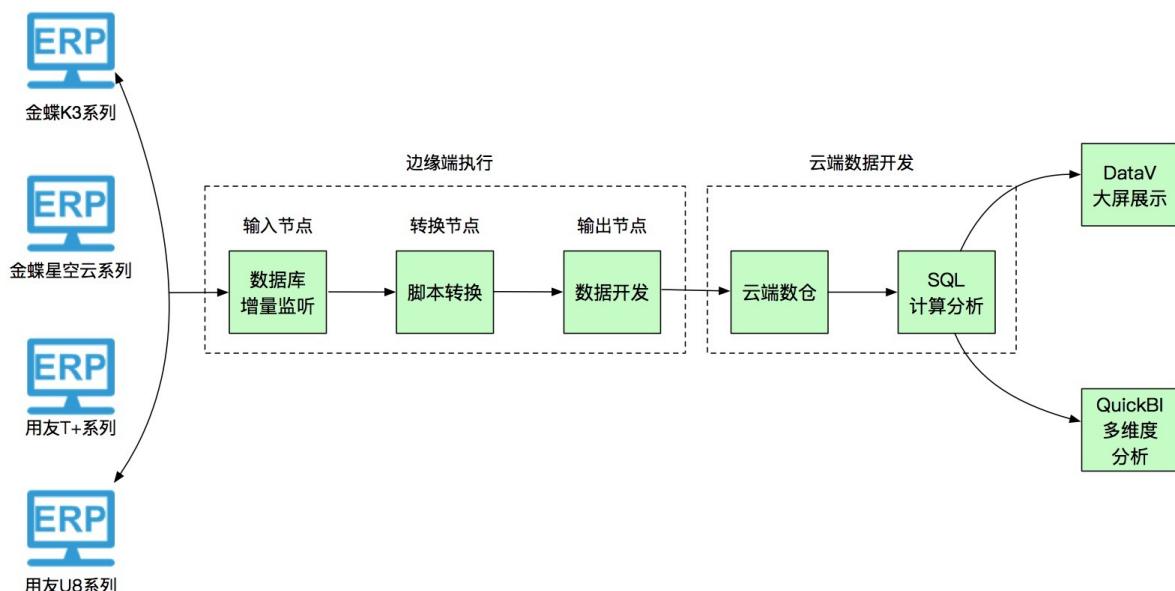
销售订单主要来自这三张表，分别是销售订单表头、销售订单明细、销售订单体财务明细，需要通过边缘数据集成从ERP里面获取出这三张表的数据放到数据开发提供的数据表中。

计算逻辑描述

维度字段：单据日期（月维度）、物料Id

度量字段：金额本位币

整体开发流程



边缘端执行

边缘数据集成 实时获取ERP数据库的数据，通过JS脚本提取数据转换成想要的数据模型，脚本转换后的数据输出到数据开发的**自定义表**数据开发里面。

云端数据开发

ERP数据保存到云端企业级平台的数据仓库中，通过编写SQL脚本做多表的JOIN和聚合计算，计算结果通过**Quick BI**和**DataV**完成数据可视化开发。

新建表

通过**自定义表**新建表：

列名	类型	描述	标识	操作
fid	string	单据内码	<input checked="" type="checkbox"/>	删除
f_entry_id	string	明细内码	<input checked="" type="checkbox"/>	删除
f_seq	string	序号	<input type="checkbox"/>	删除
f_material_id	string	物料ID	<input type="checkbox"/>	删除
f_unit_id	string	单位ID	<input type="checkbox"/>	删除
f_qty	double	数量	<input type="checkbox"/>	删除

自定义表必须指定主键，数据的增、删、改、查都是根据主键进行。只有用户自定义创建的表才可以做为边缘数据集成的点**输出节点**。

销售订单表头

表名：ods_iot_hologres_t_sal_order

参数中文	参数英文	数据类型	描述
单据内码	fid	string	PK
单据类型	f_bill_type_id	string	
单据编号	f_bill_no	string	
单据日期	f_date	bigint	
客户ID	f_cust_id	string	
销售组织ID	f_sale_org_id	string	
销售部门ID	f_sal_dept_id	string	
销售员ID	f_saler_id	string	

参数中文	参数英文	数据类型	描述
单据状态	f_document_status	string	A:新建 Z:暂存 B:审核中 C:已审核 D:重新审核
备注	f_note	string	
审核日期	f_approve_date	bigint	

销售订单明细

表名: ods_iot_hologres_t_sal_orderentry

参数中文	参数英文	数据类型	描述
单据内码	fid	string	
明细内码	f_entry_id	string	PK
序号	f_seq	string	
物料ID	f_material_id	bigint	
单位ID	f_unit_id	string	
数量	f_qty	float	
明细备注	f_note	string	
库存组织ID	f_stock_org_id	string	
供应组织ID	f_supply_org_id	string	

销售订单体财务明细

表名: ods_iot_hologres_t_sal_orderentry_f

参数中文	参数英文	数据类型	描述
单据内码	fid	string	
分录内码	f_entry_id	string	PK
序号	f_receipt_org_id	string	
收款组织ID	f_settle_org_id	string	
结算组织ID	f_amount_lc	string	
金额本位币	f_all_amount_lc	float	金额统计字段
价税合计本位币	f_all_amount_lc	float	
折扣率	f_discount	float	在税后单价上的折扣

参数中文	参数英文	数据类型	描述
折扣额	f_tax_amount	float	税后扣除折扣后的金额; 折扣额 = 含税单价 * 数量 * 折扣率
税额	f_amount_lc	float	
税额本位币	f_tax_amount_lc	float	
单价	f_price	float	
含税单价	f_tax_price	float	增加税点后的单价；含税 单价 = 单价 * (1 + 税率)
含税净价	f_tax_net_price	float	税后单价减去折扣后的净 价 含税净价 = 含税单价 * (1 - 折扣率)
金额	f_amount	float	
价税合计	f_all_amount	float	

边缘数据集成

在[边缘数据集成](#)中配置把边缘端ERP的数据实时同步到自定义表中。

模板选择

选择系统提供的对接ERP销售业务数据的模板，自动生成数据集成任务：

新建边缘数据集成 X

* 集成名称：

* 模板选择：

* 边缘集群：

参数

参数名称	配置状态	操作
__tenantid	✓ 配置完成	配置值 删除

[添加](#)

修改转换节点脚本

在转换节点脚本中定义了自定义表中的字段与对接边缘ERP的数据映射关系，如果需要调整可以通过[任务管理](#)修改对应的脚本：

The screenshot shows the 'Script Transformation Node' configuration page. The 'Script Transformation' section is highlighted with a red box. Below it is a code editor showing a Java script for mapping fields from a database table to a custom order table.

```

13 // 同步到主表
14 if (context.table === "dbo.T_SAL_ORDER") {
15     //被监控的表是主表，拿代码
16     masterData["fid"] = msgPayload["ID"];
17     masterData["f_bill_type_id"] = msgPayload["FBILLTYPEID"];
18     masterData["f_bill_no"] = msgPayload["FBILLNO"];
19     masterData["f_date"] = types.getLong(msgPayload["FDATE"] * 1000);
20     masterData["f_cust_id"] = msgPayload["FCUSTID"];
21     masterData["f_sale_org_id"] = msgPayload["FSALEORGID"];
22     masterData["f_sal_dept_id"] = msgPayload["FSALDEPTID"];
23     masterData["f_saler_id"] = msgPayload["FSALERID"];
24     masterData["f_document_status"] = msgPayload["FDOCUMENTSTATUS"];

```

Below the code editor are tabs for 'Mock Data', 'Run Result', and 'Log'. A 'Script Debug' button is also visible. At the bottom are 'Confirm' and 'Cancel' buttons.

修改输出节点

输出节点的类型为数据开发，并指向自定义的表名：

The screenshot shows the 'Output Node Configuration' page. The 'Data Development' output node type is highlighted with a red box.

* 名称	输入节点	转换节点	输出节点	配置状态	操作
销售订单表头_EM	数据库增量	脚本转换	数据开发	未配置	复制 删除 调试 全量操作 查看集成日志
销售订单明细_EM	数据库增量	脚本转换	数据开发	未配置	复制 删除 调试 全量操作 查看集成日志
销售订单财务明细	数据库增量	脚本转换	数据开发	未配置	复制 删除 调试 全量操作 查看集成日志

At the bottom are 'Add' and 'Import Template' buttons.

数据开发

X

* 表名
ods_iot_hologres_t_sal_order_emr

描述：
销售订单表头

确认 取消

启动同步

保存并下载配置，会把该任务列表同步至边缘集群的应用中，在边缘端启动该任务，用于实时增量监听ERP数据库对应的表数据。当捕捉到该表数据变化时，会自动上报变化的数据到企业级平台数据开发的自定义表里面：

任务列表

导入模板

名称	输入节点	转换节点	输出节点	配置状态	操作
销售订单表头_EM	数据库增量	配置 脚本转换	数据开发	✓ 配置完成	复制 删除 调试 全量操作 查看集成日志
销售订单明细_EM	数据库增量	配置 脚本转换	数据开发	✓ 配置完成	复制 删除 调试 全量操作 查看集成日志
销售订单财务明细	数据库增量	配置 脚本转换	数据开发	✓ 配置完成	复制 删除 调试 全量操作 查看集成日志

添加

保存 下载配置 取消

数据查看

边缘ERP中新建一个销售订单后，将自动同步到指定的自定义表中，可使用数据查看中选择表后单击数据预览，查看销售订单数据是否同步到数据开发的表里面：

数据查看

集群状态： 空闲

基本信息		删除	编辑																				
表名：	ods_iot_hologres_t_sal_orderentry_emr	描述：	销售订单明细																				
信息列	数据预览																						
<p>① 数据预览提供数据表中最新的10条记录，详细的数据查询请通过作业完成</p> <table border="1"> <thead> <tr> <th>fid</th> <th>f_unit_id</th> <th>f_qty</th> <th>f_note</th> <th>f_stock_org_id</th> </tr> </thead> <tbody> <tr> <td>100086</td> <td>100297</td> <td>120.0</td> <td></td> <td>100038</td> </tr> <tr> <td>100085</td> <td>100297</td> <td>120.0</td> <td></td> <td>100041</td> </tr> <tr> <td>100084</td> <td>100297</td> <td>11.0</td> <td></td> <td>100038</td> </tr> </tbody> </table>				fid	f_unit_id	f_qty	f_note	f_stock_org_id	100086	100297	120.0		100038	100085	100297	120.0		100041	100084	100297	11.0		100038
fid	f_unit_id	f_qty	f_note	f_stock_org_id																			
100086	100297	120.0		100038																			
100085	100297	120.0		100041																			
100084	100297	11.0		100038																			

新建表

- > default
- 用户自定义
 - ods_1111
 - [ods_iot_hologres_t_sal_o...](#)
 - ods_iot_hologres_t_sal_o...
 - ods_iot_hologres_t_sal_o...
 - person_digfty_table
 - test
 - test_001
 - zilu_car

数据统计分析

从ERP同步的数据表通过[项目管理](#)中调度任务实现统计合并成一张销售统计表。

创建输出统计表

在自定义表创建销售统计表: dwd_iot_emr_sal_order_entry

The screenshot shows the 'Data View' interface with the 'New Table' dialog open. The table name 'dwd_iot_emr_sal_order_entry' is entered. The description is 'Define sales order detail table'. The table structure is defined with the following columns:

列名	类型	描述	标识	操作
fid	string	Single document code	<input checked="" type="checkbox"/>	Delete
f_entry_id	string	Detailed document code	<input checked="" type="checkbox"/>	Delete
f_date	bigint	Document date	<input type="checkbox"/>	Delete
f_document_status	string	Document status	<input type="checkbox"/>	Delete
f_cust_id	string	Customer ID	<input type="checkbox"/>	Delete

参数中文	参数英文	数据类型	描述
单据内码	fid	string	PK
明细内码	f_entry_id	string	PK
单据日期	f_date	bigint	
单据状态	f_document_status	string	
客户ID	f_cust_id	string	
物料ID	f_material_id	string	
销售数量	f_qty	double	
销售金额	f_amount	double	
销售金额本位币	f_amount_lc	double	

创建项目

项目管理中新建一个销售统计项目：

The screenshot shows the 'Project Management' interface. On the left sidebar, the 'Project Management' item is highlighted with a red box. A modal window titled '新建项目' (Create New Project) is open in the center. Inside the modal, the '项目名称' (Project Name) field contains '销售数据统计分析'. The '项目描述' (Project Description) field contains '统计工厂每月每种产品的销售金额和数量'. At the bottom right of the modal are '确定' (Confirm) and '取消' (Cancel) buttons. In the background, there is a list of existing projects with IDs FP-57EC6499BB0A5DF9, FP-473C3835CA9319A2, and FP-B00A9928B3842985.

新建作业

新建一个SQL分析类型的作业：

The screenshot shows the 'Project Management' interface. The 'Project Management' item in the sidebar is highlighted with a red box. A modal window titled '新建作业' (Create New Task) is open. Inside, the '作业名称' (Task Name) field contains '统计每月产品销售金额', '作业类型' (Task Type) is set to 'SQL分析', and the '作业描述' (Task Description) field contains '统计每月产品销售金额'. At the bottom right of the modal are '确认' (Confirm) and '取消' (Cancel) buttons. In the background, the main interface shows a project named '销售数据统计分析' with ID FP-BDA2E59E27D3778A.

编辑如下的SQL脚本：

The screenshot shows the 'Project Management' interface. The 'Project Management' item in the sidebar is highlighted with a red box. A modal window titled '销售数据统计分析' is open, showing an existing task. The task details are: 项目ID: FP-BDA2E59E27D3778A, 项目名称: 销售数据统计分析, 项目描述: 统计工厂每月每种产品的销售金额和数量. Below this, a table lists tasks with one entry: 作业ID FJ-3C3372945AB3151D, 作业名称 '统计每月产品销售金额', 作业类型 'SQL分析', 描述 '统计每月产品销售金额'. At the bottom right of the modal are '编辑' (Edit) and '删除' (Delete) buttons. The '编辑' button is highlighted with a red box.

```
use custom;
INSERT INTO dwd_iot_emr_sal_order_entry
SELECT sal_order_view.fid AS fid
    ,orderentry_view.f_entry_id AS f_entry_id
    -- 单据日期
    ,sal_order_view.f_date AS f_date
    -- 单据状态
    ,sal_order_view.f_document_status AS f_document_status
    -- 客户Id
    ,sal_order_view.f_cust_id AS f_cust_id
    -- 物料Id
    ,orderentry_view.f_material_id AS f_material_id
    -- 销售数量
    ,orderentry_view.f_qty AS f_qty
    -- 销售金额
    ,orderentry_view.f_amount AS f_amount
    -- 销售金额本位币
    ,orderentry_view.f_amount_lc AS f_amount_lc
FROM ods_iot_hologres_t_sal_order_emr AS sal_order_view
JOIN (
    SELECT orderentry_view.fid AS fid
        ,orderentry_view.f_entry_id AS f_entry_id
        ,orderentry_view.f_material_id AS f_material_id
        ,orderentry_view.f_qty AS f_qty
        ,orderentry_f_view.f_amount AS f_amount
        ,orderentry_f_view.f_amount_lc AS f_amount_lc
    FROM ods_iot_hologres_t_sal_orderentry_emr AS orderentry_view
    JOIN ods_iot_hologres_t_sal_orderentry_f_emr AS orderentry_f_view
    ON orderentry_view.fid = orderentry_f_view.fid
    AND orderentry_view.f_entry_id = orderentry_f_view.f_entry_id
) orderentry_view
ON sal_order_view.fid = orderentry_view.fid
;
```

注：必须使用custom数据库，所有自定义表的数据都是存在该数据库中。

调试

可以单次运行任务，然后查看运行日志和输出的数据统计表结果：

项目管理 > 项目开发 > 编辑作业

编辑作业

作业名称: 统计每月产品销售金额 作业类型: SQL分析 编辑

脚本编写

```

1 use custom;
2
3 INSERT INTO dwd_iot_emr_sal_order_entry
4 SELECT sal_order_view.fid AS fid
5     ,orderentry_view.f_entry_id AS f_entry_id
6     -- 单据日期
7     ,sal_order_view.f_date AS f_date
8     -- 单据状态
9     ,sal_order_view.f_document_status AS f_document_status
10    -- 客户Id
11    ,sal_order_view.f_cust_id AS f_cust_id
12    -- 物料Id
13    ,orderentry_view.f_material_id AS f_material_id
14    -- 销售数量
15    ,orderentry_view.f_qty AS f_qty
16    -- 销售金额
17    ,orderentry_view.f_amount AS f_amount
18    -- 销售金额本位币
19    ,orderentry_view.f_amount_lc AS f_amount_lc
20    ods_iot_hologres_t_sal_order_emr AS sal_order_view

```

保存 运行 停止 全屏模式

查看运行记录

数据查看

集群状态: 空闲

请输入表名搜索

新建表

表名: dwd_iot_emr_sal_order_entry 描述: 定义销售订单明细表

基本信息

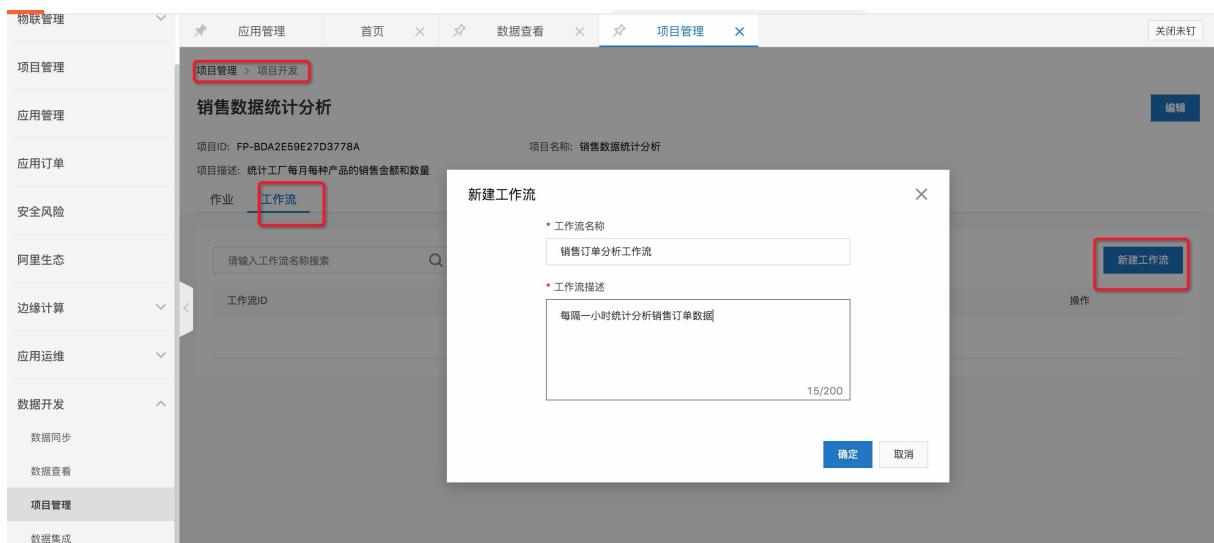
信息列 数据预览

① 数据预览提供数据表中最新的10条记录，详细的数据查询请通过作业完成

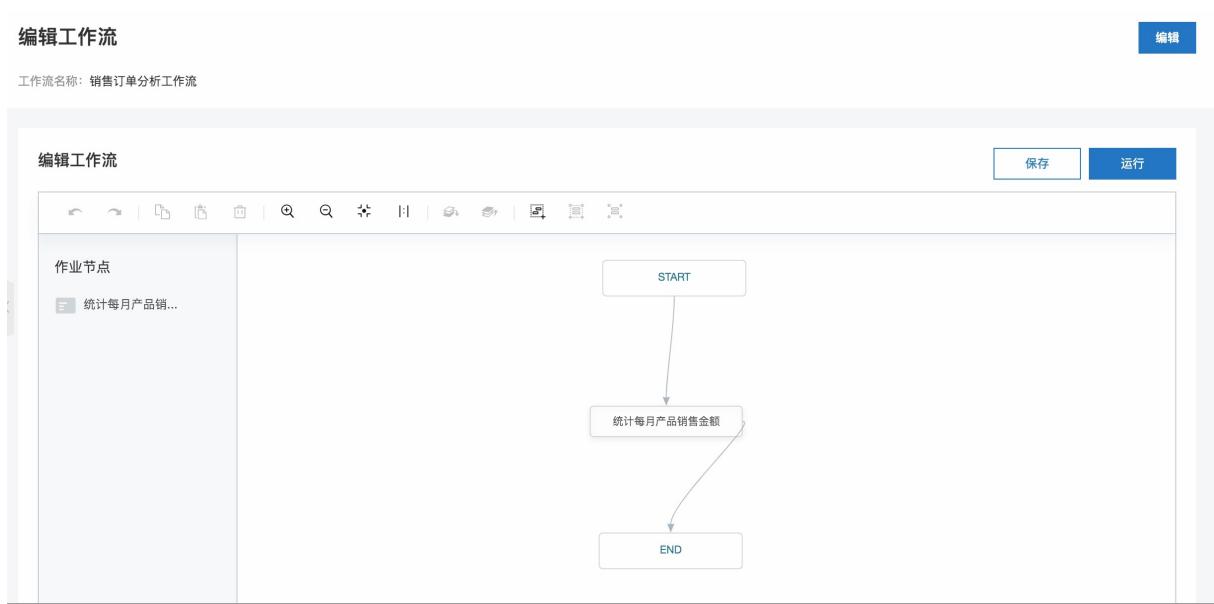
fid	f_entry_id	f_date	f_document_status	f_cust_id
100085	100097	1614182400000000	D	101335
100086	100098	1614528000000000	C	101313
100084	100096	1614182400000000	C	101313

启动定时运行

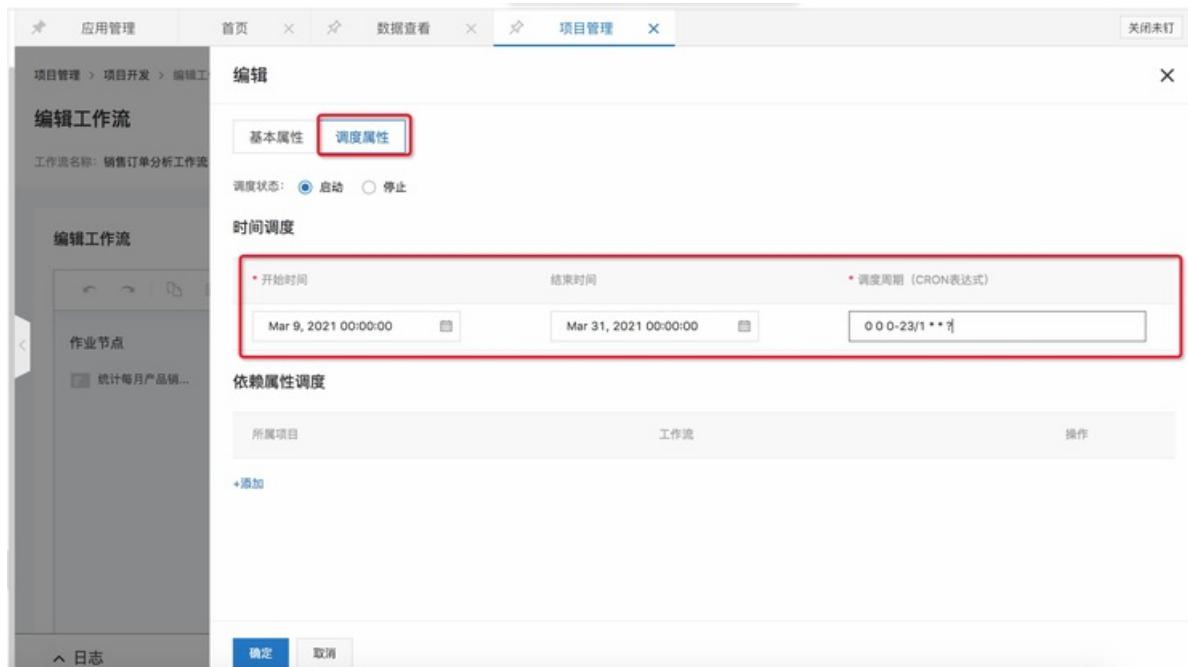
在数据开发的项目管理中创建定时调度工作流：



在工作流中设定执行销售数据统计分析的作业：



工作流的调度周期设置为每隔一个小时运行一次：



配置好工作流以后，启动该流程，并查看运行日志，数字工厂将每小时把边缘同步的ERP订单数据做一次数据统计并写入到自定义表中：

作业实例ID	作业名称	执行集群	作业类型	作业节点提交主机	开始时间	作业完成时间	执行时长	执行状态	操作
FNI-3B0ED4CD9DA6BEA1		C-DB4B5BC5C6EA12AB			2021-03-09 11:18:38	2021-03-09 11:18:38	0s	OK	详情
FNI-9CDB7A276DFF7F4A	统计每月产品销售金额	C-DB4B5BC5C6EA12AB	SPARK_SQL	emr-header-1.cluster-194732	2021-03-09 11:18:38	2021-03-09 11:19:51	73s	OK	详情
FNI-5BFBDAA45B3040B2		C-DB4B5BC5C6EA12AB			2021-03-09 11:19:51	2021-03-09 11:19:51	0s	OK	详情

数据可视化开发

销售业务统计结果通过Quick BI进行可视化的BI分析。

获取引擎连接信息

集群状态: 空闲

数据库地址	用户名	数据库名称	端口	操作
[REDACTED]	[REDACTED]	default	10000	复制

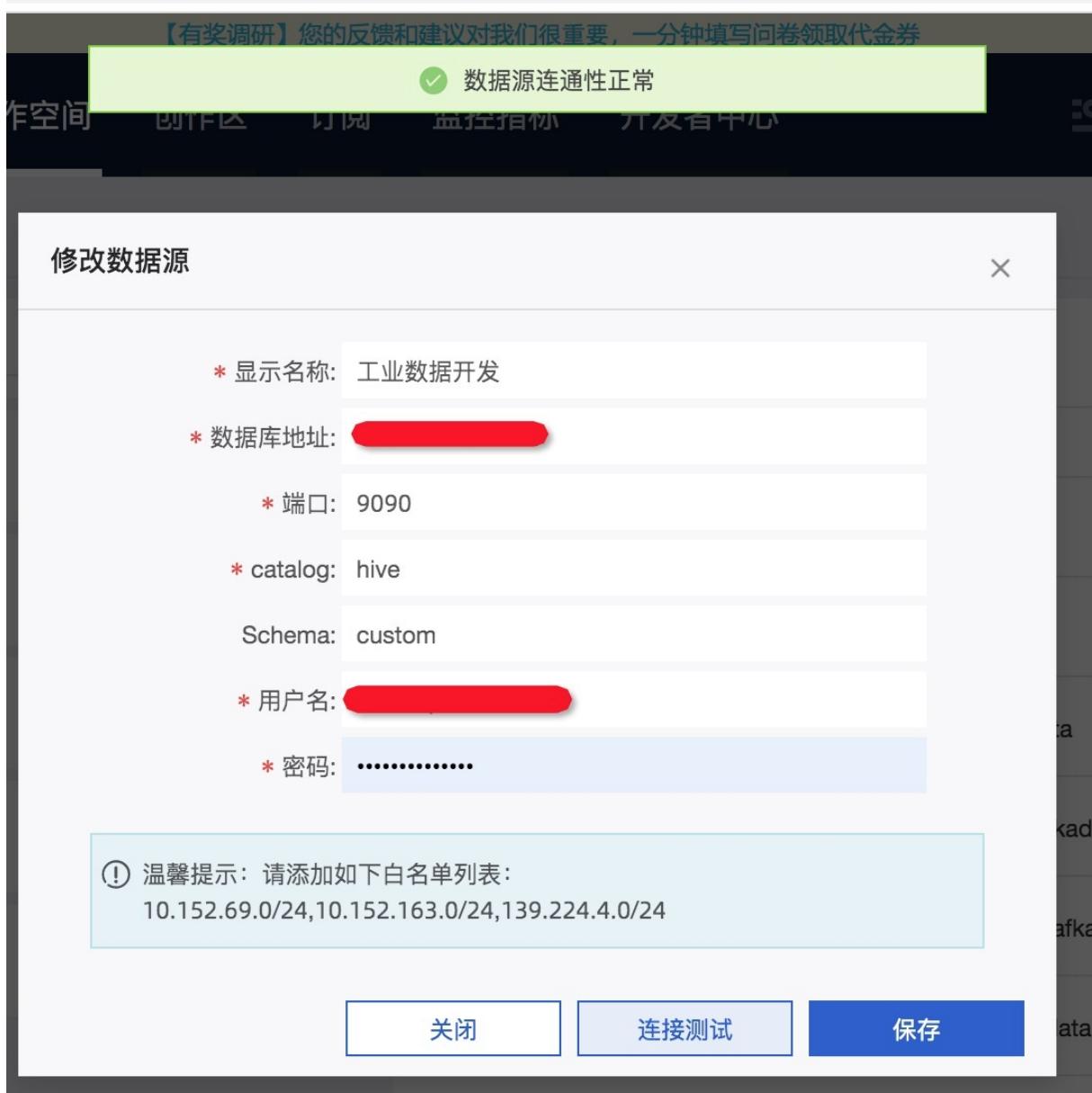
数据分析仪表盘

数据分析仪表盘	描述	操作
销售经营分析.EMR	-	删除
MaxCompute月度分析	测试	删除
Lightning月度分析	Lightning月度分析	删除
ADB月度分析	ADB月度分析	删除
数据开发人均产量产值分析	工厂人员人均产能产值分析	删除

共有5条 < 上一页 1 下一页 > 每页显示: 10

创建数据源

在Quick BI中创建Presto数据源：



创建数据集

使用Presto数据源来创建对应的分析数据集：

The screenshot shows the 'Data Source' management interface. On the left, under 'My Data Sources', there is a list of sources including 'MaxCompute Development Environment', 'MaxCompute Production Environment', 'ERP Indicator Operation Analysis Data Source...', 'test Account 开发', 'test Account 线上', and 'Industrial Data Development'. The last item, 'Industrial Data Development', is highlighted with a red box. On the right, a list of datasets is shown with one entry, 'dwd_iot_emr_sal_order_entry_presto_custom', highlighted with a red box. A red box also highlights the 'Edit' icon for this dataset. The top right corner has a button labeled '+ New Data Source'.

注：创建数据集一定要用xxx_presto_custom结尾的表，否则数据集内容无法查询。

创建好数据集后就可以创建分析仪表板，具体操作请参看[制作仪表板快速入门](#)。

数字工厂集成

最终开发完成的可视化页面可集成到数字工厂分配给对应的账号查看。

获取集成参数

集成仪表板报表到数字工厂需要该报表的PageId，以及QuickBI的AK：

The screenshot shows the QuickBI dashboard editor. On the left is a line chart titled '销售金额本位币' (Sales Amount in Local Currency) showing data for two material IDs: 100326 and 100414. The Y-axis ranges from 18.4万 to 20.4万. The chart area has a watermark 'aliyun Industry BI'. On the right is the 'Chart Design' panel, which includes sections for 'Data' (扇区角度/度量), 'Style' (样式), and 'Advanced' (高级). Under 'Data', the '度量' (Measure) section is set to '销售金额本位币(求和)' (Sum of Sales Amount in Local Currency). The '维度' (Dimension) section lists dimensions like '客户ID' (Customer ID), '单据状态' (Bill Status), 'f_entry_id', '物料ID' (Material ID), and 'fid'. The URL bar at the top contains the workspace ID 'das.base.shuju.aliyun.com/dashboard/pc.htm?workspaceId=e696848a-242b-45a3-a3b1-e5f69284287f&pageId=ca6dd33c-fbaa-4f76-906e-08e67cac7425'.

The screenshot shows the 'Organization Management' section of the Quick BI interface. On the left sidebar, 'Organization Management' is highlighted with a red box. The main area displays two input fields: 'Quick BI AccessKey ID:' and 'Quick BI AccessKey Secret:', both of which are also highlighted with red boxes. A 'Copy' button is located next to each field.

报表嵌入

在QuickBI中选择嵌入报表：

The screenshot shows the 'Report Embedding' section of the Quick BI developer center. The left sidebar has 'Report Embedding' highlighted with a red box. The main area shows a list of dashboards with one named '销售订单分析_EMR' highlighted with a red box. The dashboard list includes: 销售金额趋势分析_钉... (工业BI_DEV), 产品大类库存分析... (工业BI_DEV), 渠道销售结构分析... (工业BI_DEV).

然后从数据开发的数据集成中集成新的仪表盘：

The screenshot shows the 'Data Integration' interface. The left sidebar has 'Data Integration' highlighted with a red box. The main area shows a 'New Dashboard' dialog. It requires filling in 'Dashboard Name' (销售经营分析_EMR), 'Quick BI AccessKey ID' (highlighted with a red box), 'Quick BI AccessKey Secret' (highlighted with a red box), and a 'Description' field. A 'Confirm' button is at the bottom right.

数据分析的仪表盘集成到数字工厂后，页面将加入到经营驾驶舱应用中，可以在**角色管理**给不同的业务角色访问和使用，分配权限的业务用户登录数字工厂以后就能在经营驾驶舱中使用：

