## 银河大数据平台操作文档

### 银河信用风险系统资源列表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 资源url | | | 账号 | 密码 |
| 虚拟桌面1 | https://hz-connect02.chinastock.com.cn/portal/webclient/index.html#/desktop | | | vdi.CRS02 | ToX;fdiFp( |
| 虚拟桌面2 | https://hz-connect01.chinastock.com.cn/portal/webclient/index.html#/desktop | | |
| etl管理平台（虚拟桌面内） | http://10.1.72.3:8887/dev/ | | | vdi.CRS01 | Sy)H]304dn |
| Hue（虚拟桌面内） | 10.1.72.11:8888 | | | sys\_crs | ~~Y3ebRPgBPpvE~~  4WegMuafbrbB |
| Wind门户 | http://wds.wind.com.cn/rdf/?#/main | | | 15210381731 | 112358 |
| ssh | 10.1.72.3 | | | sys\_dw\_dev | s2T9UsScYW0D |
| 大数据平台表结构 | https://shimo.im/sheets/nySFjJL4WQ4VD7Ul/E48Rq | | | | |
| 业务系统处理表清单 | https://shimo.im/sheets/5A2AR8fLEBkeykR6/ZNElc | | | | |
| 沟通 | 周诗咏 | | | | |
| 数据推送 | 发邮件到：[tao.wang25@pactera.com，](mailto:tao.wang25@pactera.com，)  抄送：zhoushiyong\_it@chinastock.com.cn  数据推送样例：<信用风险数仓表推送列表.xlsx>(只写需要推送的任务) | | | | |
|  | | | | | |
| 脚本名称 | | 脚本描述 | 涉及表 | | |
| prd\_adm.mid\_corp\_fin\_d.sql | | 财务基础指标表 | adm.mid\_corp\_fin\_d | | |
| prd\_adm.mid\_ashare\_mkt\_der\_idx\_d.sql | | a股衍生指标表 | adm.mid\_ashare\_mkt\_der\_idx\_d | | |
| prd\_adm.mid\_sw\_industry\_inf\_d.sql | | 申万二级中值表 | adm.mid\_industry\_sw\_inf\_d | | |
| prd\_mart\_rpt.crs\_stock\_fin\_idx\_d.sql | | 信用风险指标表 | adm.crs\_sec\_fin\_idx\_d | | |
| sync\_sample\_data.py | | 定时检测样本 |  | | |
| adm.mid\_corp\_main\_holders\_std\_qrt\_inf\_d.sql | | 10大股东标准化表 | fdm.inf\_ashare\_inter\_hld\_quar | | |
| mart\_app.crs\_sec\_industry\_plate.sql | | 行业板块信息表 | mart\_app.crs\_sec\_industry\_plate | | |
| adm.mid\_main\_cir\_holders\_inf\_d.sql | | 10大流通股东 | fdm.inf\_ashare\_out\_stkhld\_quar | | |
| MART\_RPT.CRS\_COMP\_ATTENT\_INF\_D.sql | | 银河质押标的表 | mart\_app.crs\_pledge\_underlying\_inf\_d | | |
| MART\_RPT.CRS\_RELATION\_SHIP\_D.sql | | 公司关联表 | fdm.inf\_ashare\_co\_rela\_ha | | |

说明：任务可以通过对应的表名查询，表结构可以通过hue界面查看，**上述脚本列表内容以etl平台为准**，因为不能从etl粘贴出来，后期有些测试直和修改时直接在管理平台的脚本改的；标红颜色的是随着字段需求可能变化的表，其他表比较稳定，不需要修改。

etl管理平台中[mid\_corp\_fin\_d],[mid\_ashare\_mkt\_der\_idx\_d],[mid\_industry\_sw\_inf\_d],[crs\_sec\_fin\_idx\_d],带dev的是我自己补数的任务，不带dev的是需要修改对应脚本后提交上线申请的，两套任务的脚本略微不一致。

注意：业务库新增字段时数据平台中是decimal(xx,x)的禁止使用float或者double等数据类型

测试库（xxx\_dev\_xxx）中的表自己可以修改，删除，创建等等,只能操作和自己有关的表。

Hive/impala常用命令：

Insert overwrite xxxx; //覆盖插入

Insert into xxxx;//追加插入

show partitions tb\_name;

eg:show partitions dw\_dev\_adm.mid\_corp\_fin\_d;

删除单个分区：alter table dw\_dev\_adm.mid\_corp\_fin\_d drop partition (biz\_date = 20200930);

删除多个分区：alter table dw\_dev\_adm.mid\_corp\_fin\_d drop

partition (biz\_date = 20200930),

partition (biz\_date=20200929);

增加列：ALTER TABLE table\_name ADD|REPLACE COLUMNS (col\_name data\_type [COMMENT col\_comment], ...);

Eg：

|  |
| --- |
| ALTER TABLE dw\_dev\_adm.mid\_corp\_fin\_d ADD COLUMNS  (  NET\_ASSET\_PAYOFF\_RATE DECIMAL(38,8) COMMENT '净资产收益率',  ASSET\_LIAB\_RATE DECIMAL(38,8) COMMENT '资产负债率',  TOTL\_ASSET\_REWARD\_RATE DECIMAL(38,8) COMMENT '总资产报酬率',  SYNTHETIC\_PAYOFF\_GROSS\_PARENT DECIMAL(38,8) COMMENT '归属于母公司所有者的净利润',  BIZ\_INCOME DECIMAL(38,8) COMMENT '营业收入',  SALEGROSS\_PROFIT\_MARGIN DECIMAL(38,8) COMMENT '销售毛利率'  ); |

历史补数步骤：

1. 分别修改DW\_DEV\_ADM\_MID\_INDUSTRY\_SW\_INF\_D，DW\_DEV\_ADM\_MID\_ASHARE\_MKT\_DER\_IDX\_D，DW\_DEV\_MART\_CRS\_SEC\_FIN\_IDX\_D任务脚本中的固定值(如果没有则无需修改)到需要补数的日期，比如【20210115】；历史最早支持可追溯的数据到20190408
2. 删除表【dw\_dev\_adm.mid\_corp\_fin\_d】的所有非报告期的分区；
3. 复制【DW\_DEV\_ADM\_MID\_CORP\_FIN\_D】任务中的脚本在hue中执行，执行的时候需要输入变量，输入对应的日期，这里考虑到已经发布的财报日期没有必要按照天跑重复的数据，所以执行完脚本后数据会落在固定的4个财报日期对应的分区里，接下来的两个任务【DW\_DEV\_ADM\_MID\_INDUSTRY\_SW\_INF\_D】，【DW\_DEV\_MART\_CRS\_SEC\_FIN\_IDX\_D】也是从对应的财报日期的分区取得数据，变量输入的时候依次输入最近的日期和接下来的财报日期即可，比如：今天是20210118，txdate=20210115,exe\_date=20210115,执行；接下来改exe\_date为20201231,执行；20200930,执行；20200630执行；20200331执行。。。执行到20190630即可；这里也可以用【show partitions xxx;】来查看表的最新分区确定最新的执行日期。
4. 登录上述资源列表【SSH】资源服务器移步如下路径：/etc/bigdata\_jobs/dev/dpl/
5. 补dw\_dev\_adm.mid\_ashare\_mkt\_der\_idx\_d 表数据：python3 run\_ndays\_job.py DW\_DEV\_ADM\_MID\_ASHARE\_MKT\_DER\_IDX\_D 20190408 20210115 1
6. 补dw\_dev\_adm.mid\_industry\_sw\_inf\_d数据：python3 run\_ndays\_job.py DW\_DEV\_ADM\_MID\_INDUSTRY\_SW\_INF\_D 20190408 20210115 1
7. 补dw\_dev\_mart.crs\_sec\_fin\_idx\_d 数据：python3 run\_ndays\_job.py DW\_DEV\_MART\_CRS\_SEC\_FIN\_IDX\_D 20190408 20210115 1
8. Etl管理平台编辑DTS\_DW\_DEV\_MART\_CRS\_PLEDGE\_RISK\_STOCK\_FIN\_IDX任务，修改推送为全量推送，刷新元数据，如元数据没有变化(stock\_fin\_idx表没有新增或者删除字段),无需刷新；
9. 推送数据：python3 etljob.py DTS\_DW\_DEV\_MART\_CRS\_PLEDGE\_RISK\_STOCK\_FIN\_IDX 20210115
10. 同时在etl管理平台的测试环境找到对应的有修改的不带dev的任务脚本，修改完后测试，单次执行，查看日志没问题后提交上线申请即可。

### 5开发流程思路

1. 需求分析
2. 根据必要的字段和常用的字段以及复用性等方面确定最终表结构
3. 确定表依赖的所有表的结构，数据维度，一条记录的意义等细节
4. 确定需求表的主键，数据维度，计算，更新策略等
5. 将自己的想法用程序或者脚本落地实现
6. 执行脚本查看结果是否复合预期结果
7. 优化，部署上线

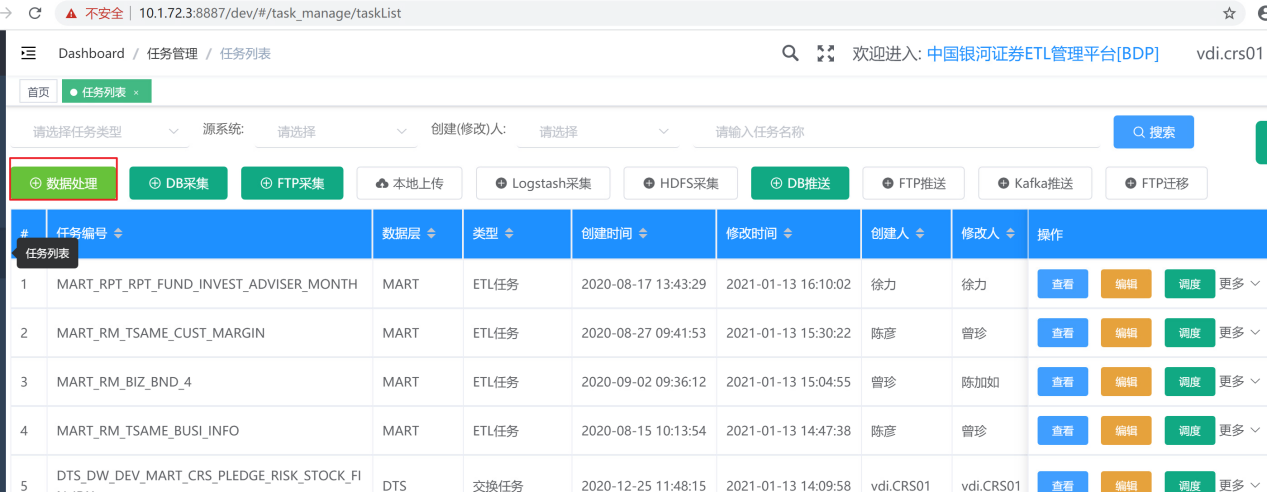
### 开发步鄹

具体见下图及描述

1. 根据需求确定表结构发诗咏老师评审；
2. 评审完成建表后在测试库新建对应的表；

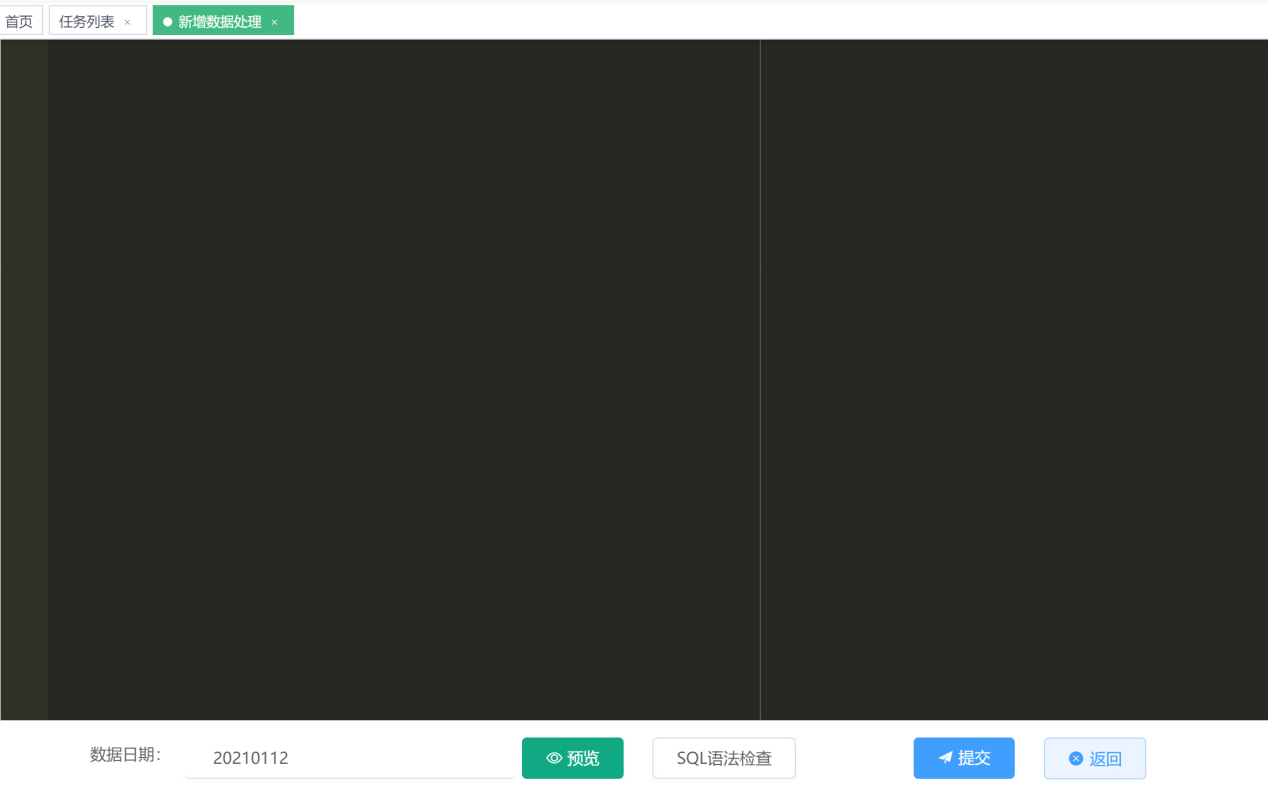
建表可以在hue中执行对应的建表语句；

1. Etl测试环境新建任务：任务列表->数据处理，见下图中的数据处理；

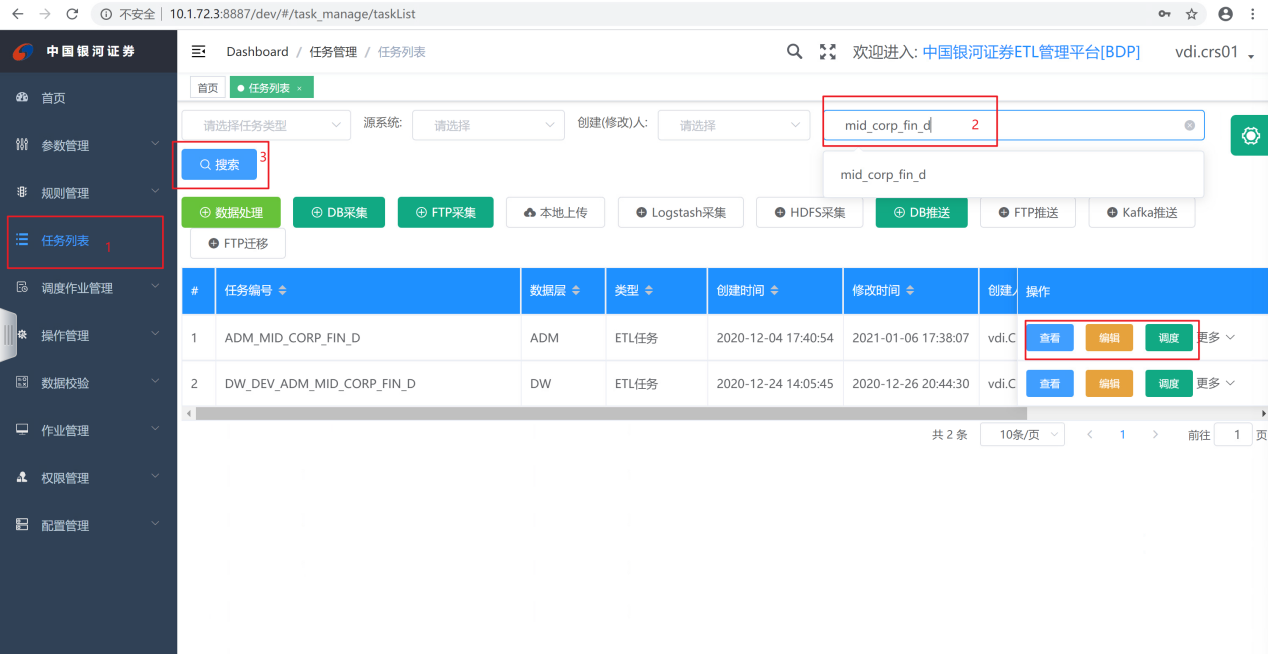


1. 选择对应的表，编写处理逻辑提交；



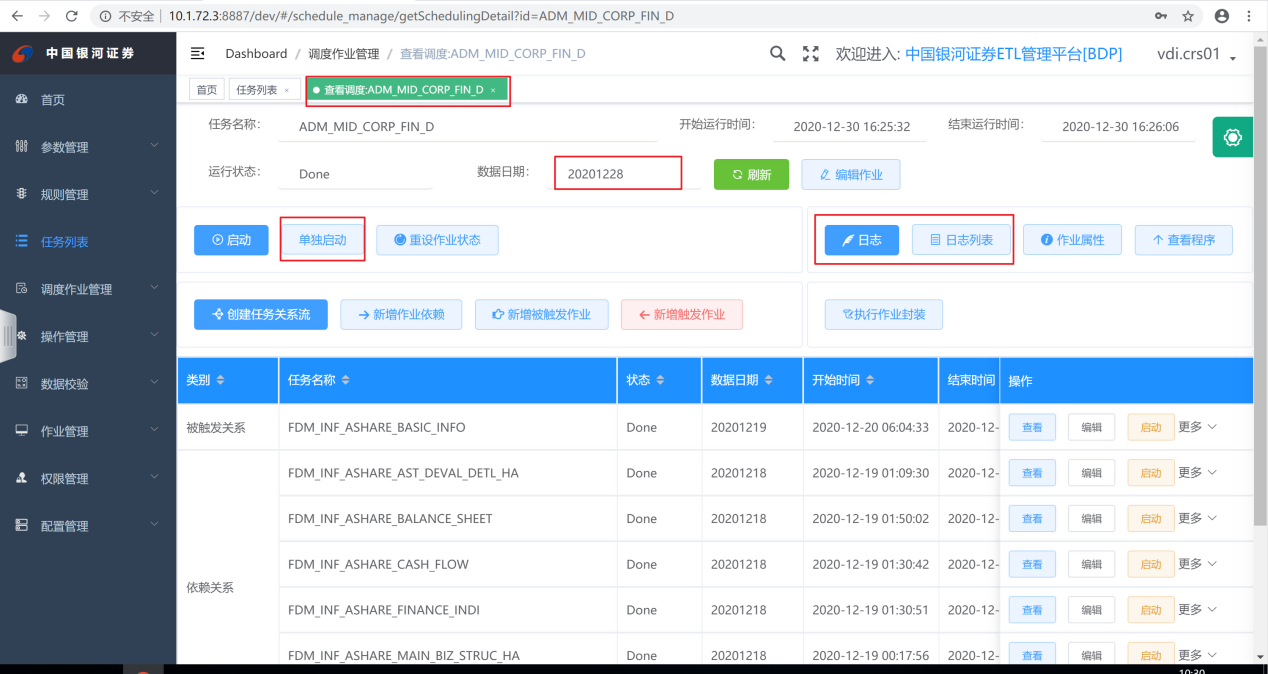


1. 保存任务并增加调度，增加表之前的依赖关系





1. 执行测试；打开对应的任务点击调度到达如下界面，点击单独启动，查看日志是否报错；



1. 申请上线

