MaxCompute数据开发规范

更新时间：2020-10-14 14:35:15

[**本页目录**](javascript:void(0))

* [项目空间管理规范](https://help.aliyun.com/document_detail/117439.html?spm=a2c4g.11186623.6.922.61751eefJTKKxM#title-qn7-9h2-jte)
* [表和视图相关规范](https://help.aliyun.com/document_detail/117439.html?spm=a2c4g.11186623.6.922.61751eefJTKKxM#title-s32-fub-hhw)
* [DataWorks工作流节点设计规范](https://help.aliyun.com/document_detail/117439.html?spm=a2c4g.11186623.6.922.61751eefJTKKxM#title-ja5-84x-wqn)
* [编码规范](https://help.aliyun.com/document_detail/117439.html?spm=a2c4g.11186623.6.922.61751eefJTKKxM#title-pxq-bub-8a7)

本文为您介绍MaxCompute数据开发规范，包括项目空间、表、视图、工作流节点和编码规范。

在进行数据开发前，请做好数据仓库研发流程的阶段规划，了解各种角色及其职责，具体内容请参见[数据仓库研发规范概述](https://help.aliyun.com/document_detail/115496.html#concept-186668)。

**项目空间管理规范**

* 关于项目划分和命名规范的详解，请参见[MaxCompute项目分配](https://help.aliyun.com/document_detail/117433.html#section-ocv-imw-bwu)、[项目命名规范](https://help.aliyun.com/document_detail/117433.html#section-v60-ej0-v2e)。
* 关于项目安全管理规范的详解，请参见[安全模型](https://help.aliyun.com/document_detail/104045.html#concept-bcx-q25-ngb)。
* DataWorks项目空间目录构建建议：
  + 数据开发时建议建立两层目录：
    - 第一层目录表示数据域（对于中间层项目）或业务线（对于应用层项目），例如日志、会员等。
    - 第二层目录表示层次，如DWD、DWS、DIM以及数据同步任务。
  + 为避免临时查询文件列表过多，建议以开发人员姓名作为文件夹的名称进行管理。

**表和视图相关规范**

* 表设计规范
  + 表（Table）和字段命名规范请参见[ODS层设计规范](https://help.aliyun.com/document_detail/117434.html#concept-226666)、[CDM公共维度层设计规范](https://help.aliyun.com/document_detail/117435.html#concept-226667)、[CDM明细层设计规范](https://help.aliyun.com/document_detail/117437.html#concept-226671)、[CDM汇总层设计规范](https://help.aliyun.com/document_detail/117438.html#concept-226672)。
  + 表（Table）整体设计规范请参见[表设计规范](https://help.aliyun.com/document_detail/117390.html#concept-226512)。

**说明** 建议通过DataWorks数据管理中的管理配置模块进行表的分类管理，同时通过DataWorks数据开发中的表管理模块将表与对应分类关联。

* 视图设计规范
  + 视图的命名规范与表保持一致。
  + 建议创建独立的刷新任务以产生视图，创建视图的脚本如下。

**create** **or** **replace** **view** \*\*\*;

**DataWorks工作流节点设计规范**

* 工作流节点类型和命名
  + 工作流节点的输出表命名规范

projectname.tablename

* + 工作流节点命名规范

| **节点类型** | **命名规范** | **备注** |
| --- | --- | --- |
| 虚拟节点 | vt\_{虚拟节点含义} | 任务根节点。 |
| 同步节点导入任务 | imp{表名}[{源库标示}] | 如果多个源库存在表名重复的情况，可以增加源库标识的后缀。 |
| 同步节点导出任务 | exp{表名}[{目标库标示}] | 如果存在多个目标库，可以增加目标库标识的后缀。 |
| 数据处理节点 | {输出表名} | * + - 多个目标表输出任务时，选定一个主要表名作为节点名。     - 多个任务插入同一张表的不同分区时，可以建一个虚拟目标表任务。 |
| Shell节点 | sh\_{脚本命名} | 不涉及 |
| MapReduce节点 | mr\_{脚本命名} | 不涉及 |

* 资源文件命名规范

资源名称需有后缀表示资源类型，例如**.JAVA**、**.PY**、**.SH**等。

* 任务设计规范

SQL任务：

* + 每个MaxCompute SQL任务至少有一个输出表。
  + 脚本需支持重跑，例如使用INSERT OVERWRITE等语句，以便在系统错误时，重跑任务不会出现重复数据等脏数据。
  + 代码如果有时间或日期参数，需采用类似{bdp.system.cyctime}的调度参数，以方便调试。
  + 自定义参数，采用**${变量名}**在发布任务时进行配置。变量名即为调度参数。

**编码规范**

* 编写原则
  + 代码行清晰、整齐，具有一定的可观赏性。
  + 代码编写要充分考虑执行速度最优原则。
  + 代码行整体层次分明、结构化强。
  + 代码中应有必要的注释以增强代码的可读性。
  + 规范要求非强制性地约束代码开发人员的代码编写行为。在实际应用中，只要不违反常规要求，允许存在可理解的偏差。
* 基本要求
  + 代码中应用到的所有SQL关键字、保留字都需使用全大写或小写，例如select/SELECT、from/FROM、where/WHERE、and/AND、or/OR、union/UNION、insert/INSERT、delete/DELETE、group/GROUP、having/HAVING、count/COUNT等。不能使用大小写混合的方式，例如Select或seLECT等方式。
  + 代码中应用到的除关键字、保留字之外的代码，都要求使用小写。
  + 四个空格为一个缩进量，所有的缩进均为一个缩进量的整数倍。
  + 禁止使用SELECT \*操作，所有操作必须明确指定列名。
  + 通常要求对应的括号在同一列上。

**说明** 通过DataWorks或MaxCompute Studio编码时，可以用格式化工具对SQL代码进行格式化。

* 数据类型
  + MaxCompute Project的表字段类型应尽量与业务系统一致。
  + 不推荐大量使用STRING类型，以免数据加工环节的数据质量问题无法及时暴露。
  + 在对精度要求极其严格的场景下请谨慎使用DECIMAL类型。
  + 关于货币类型
    - 中国货币单位统一为人民币元，国际货币单位统一为美元。
    - 除非模型有特殊说明，否则中间层金额相关的数据不执行任何四舍五入操作，以避免后续的汇总计算中出现不同口径的汇总结果不一致的情况。
* DataWorks编码规范

通过DataWorks进行数据开发时，在DataWorks的数据开发工作台上进行代码编辑的规范。

* + 代码头部

代码头部添加主题、功能描述、作者、日期等信息。并提供修改日志及标题栏的功能，以便后续修改人员添加修改记录。每一行不能超过80个字符。

DataWorks中针对不同的任务类型提供了不同的代码头部模板，也可以在[配置中心](https://ide2-cn-shanghai.data.aliyun.com/page/config/template?spm=a1z3jh.11719701.0.0.59526242vXfiQO)自定义头部模板。例如，SQL类型任务头部模板默认为如下。

--odps sql

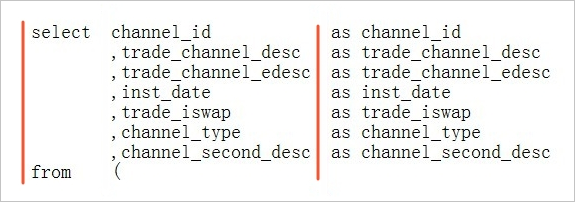
--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*--

--author:${author}

--create time:${createTime}

--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*--

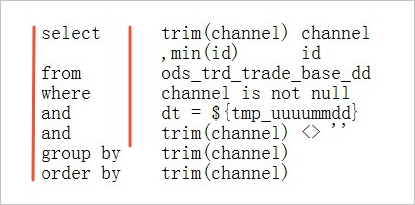
* + 字段排列要求
    - SELECT语句选择的字段按每行一个字段方式编排。
    - SELECT单字后面一个缩进量后应直接跟首个选择的字段，即字段离首起二个缩进量。
    - 其它字段前导二个缩进量再跟一个逗号（，）后放置字段名。
    - 两个字段之间的逗号（，）分割符紧跟在第二个字段的前面。
    - AS语句应与相应的字段在同一行，多个字段的AS建议尽量对齐在同一列上。

[](https://static-aliyun-doc.oss-cn-hangzhou.aliyuncs.com/assets/img/zh-CN/4407562061/p46109.jpg)

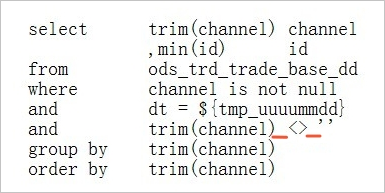
* + SELECT子句排列要求

SELECT语句中所用到的FROM、WHERE、GROUP BY、HAVING、ORDER BY、JOIN、UNION等子句，需遵循如下要求：

* + - 换行编写。
    - 与相应的SELECT语句左对齐编排。
    - 子句后续的代码离子句首字母二个缩进量起编写。
    - WHERE子句下的逻辑判断符AND、OR等与WHERE左对齐编排。
    - 超过两个缩进量长度的子句加一空格后编写后续代码。例如ORDER BY、GROUP BY等。

[](https://static-aliyun-doc.oss-cn-hangzhou.aliyuncs.com/assets/img/zh-CN/4407562061/p46114.jpg)

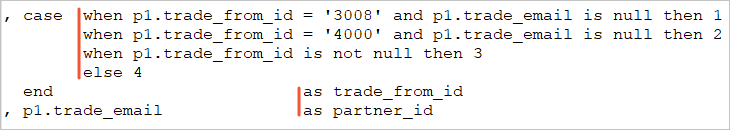
* + 运算符前后间隔要求

算术运算符、逻辑运算符的前后要保留一个空格。[](https://static-aliyun-doc.oss-cn-hangzhou.aliyuncs.com/assets/img/zh-CN/4407562061/p46110.jpg)

* + CASE语句的编写

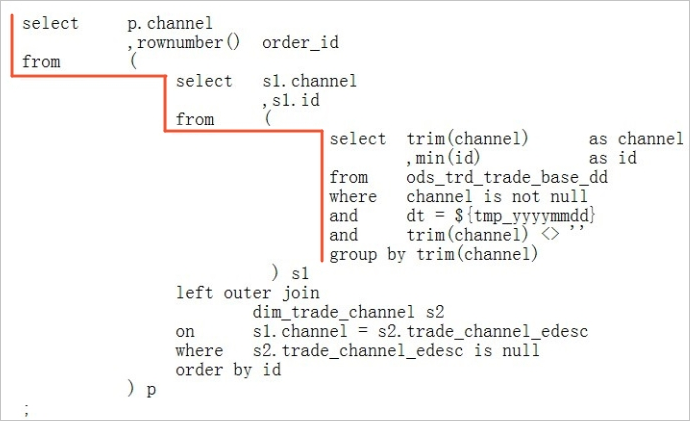
SELECT语句中对字段值进行判断取值的操作将用到CASE语句，正确的编排CASE语句对加强代码行的可读性也是很关键的一部分。对CASE语句编排的约定如下：

* + - WHEN子句在CASE语句的同一行并缩进一个缩进量后开始编写。
    - 每个WHEN子句单独一行编写，如果语句较长可换行编写。
    - CASE语句必须包含ELSE子语，ELSE子句与WHEN子句对齐。

[](https://static-aliyun-doc.oss-cn-hangzhou.aliyuncs.com/assets/img/zh-CN/4407562061/p46111.jpg)

* + 子查询嵌套编写规范

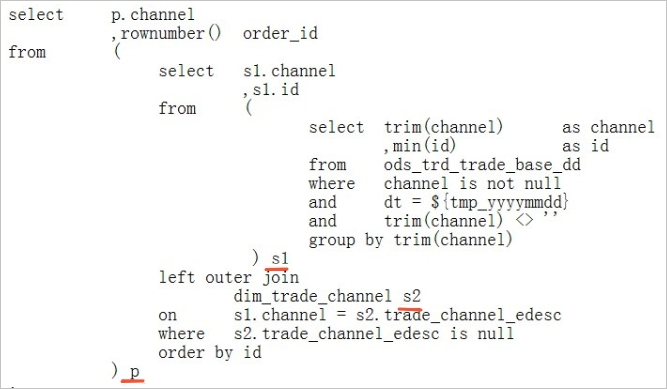
在数据仓库系统ETL开发中经常需要用到子查询嵌套，因此代码的分层编排变得非常重要。

[](https://static-aliyun-doc.oss-cn-hangzhou.aliyuncs.com/assets/img/zh-CN/5407562061/p46112.jpg)

* + 表别名定义约定

建议对所有的表加上别名。一旦在SELECT语句中对表定义了别名，在整个语句中对此表的引用都必须以别名替代。考虑到编写代码的便捷性，约定别名尽量简洁，同时避免使用关键字。 表别名定义约定如下：

* + - 表别名采用简单字符命名。
    - 多层次的嵌套子查询，在别名之前要体现层次关系。SQL语句别名或分层的命名，从第一层次至第四层次，分别用P、S、U、D表示，取意为Part，Segment，Unit，Detail。也可用a、b、c、d来表示第一层次到第四层次。对于同一层次的多个子句，可以在字母后加1、2、3、4区分。
    - 必要时，为表别名添加注释。

[](https://static-aliyun-doc.oss-cn-hangzhou.aliyuncs.com/assets/img/zh-CN/5407562061/p46113.jpg)

* + SQL注释
    - 每条SQL语句均应添加注释说明。
    - 每条SQL语句的注释单独成行并置于语句前面。
    - 字段注释紧跟在字段后面。
    - 应为不易理解的分支条件表达式添加注释。
    - 应说明重要计算的功能。
    - 过长的函数实现，应将其语句按实现的功能分段加以概括性说明。
    - 常量及变量注释时，必须注释被保存值的含义，按需注释合法的取值范围 。
  + MaxCompute项目空间名称的编写

本项目空间（Project）的名称不需在编码中体现。如果引用了其他项目空间的表则需要在表名前加上项目空间名称。示例如下。

--当前Project为prj\_bi。

**INSERT** OVERWRITE **TABLE** test\_2

**SELECT** c1

,c2

,c3

**FROM** prj\_ods.test\_1

**WHERE** pt = 20181212

;

* DataWorks任务发布规范
  + 发布上线的时间根据业务定义。
  + 无QA人员参与的项目，由开发负责自测，开发测试环境通过后再自行发布到生产环境。
  + 有QA人员参与的项目，开发负责提交到调度开发环境并测试通过，而正式上线则由QA负责打包发布到生产环境。