SQL编码规范

版本变更历史：

|  |  |  |  |
| --- | --- | --- | --- |
| 时间 | 版本 | 作者/修改人 | 变更原因 |
| 2017-05-31 | V1.0 | 喻健民 | 初版 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[1. 简介 4](#_Toc484181260)

[1.1目标 4](#_Toc484181261)

[1.2范围 4](#_Toc484181262)

[1.3使用约定 4](#_Toc484181263)

[2. 通用规则 4](#_Toc484181264)

[3. 配置约定 5](#_Toc484181265)

[4. 格式约定 6](#_Toc484181266)

[5. 命名约定 8](#_Toc484181267)

[5.1 一般规则 8](#_Toc484181268)

[5.2 常用规则 8](#_Toc484181269)

[6. 注释 8](#_Toc484181270)

[6.1 对象注释 8](#_Toc484181271)

[6.2 代码注释 9](#_Toc484181272)

[7. 模板 10](#_Toc484181273)

# 简介

开发小组在进行SQL编程时常常忽略编码规范，但是这些规范是开发小组顺利展开工作非常重要的工具。虽然编码风格不影响程序的执行，但是不同的编码风格会严重影响不同开发人员之间进行技术沟通的效率，同时也给维护带来不便。

## 1.1目标

SQL编码规范希望能达到如下目标：

* 规范不良的编码习惯
* 提高代码的可读性、可维护性以及可移植性

## 1.2范围

规范只针对Hive中SQL编程语言。

## 1.3使用约定

正常情况下优先考虑，如果有违反建议的编码，应该同有经验的同事商量并取得违反建议的理由共识后才能实施，推荐对违反建议的代码进行注解。

# 2. 通用规则

每一次不执行建议的编码规则，必须有合理的理由，理由不包括个人的编码风格因素。

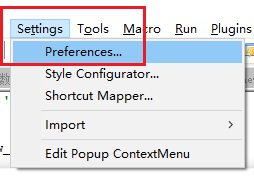
# 3. 配置约定

为方便开发小组成员编写出高质量的SQL代码，需要开发小组各成员按照统一的标准配置适合快速、高效开发的使用环境。这里，将会列出一些较为适用的参数配置约定，需要开发小组成员统一执行该约定，见下表：

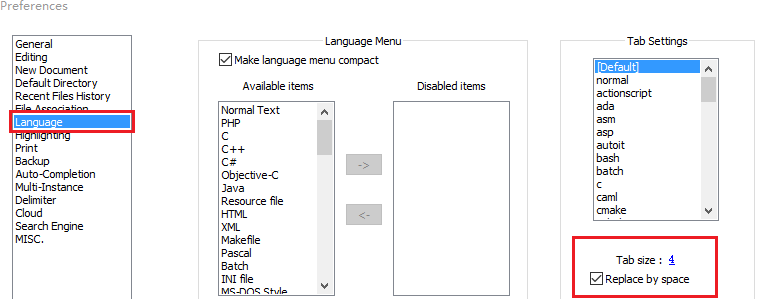
|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 项目 | 推荐方案 | 备注 |
| 1 | SQL代码编辑工具 | Notepad++ |  |
| 2 | 缩进符 | 使用4个空格键 | 需要配置 |
|  |  |  |  |

Notepad++ Tab键配置如下：

菜单【设置】-> 【首选项】

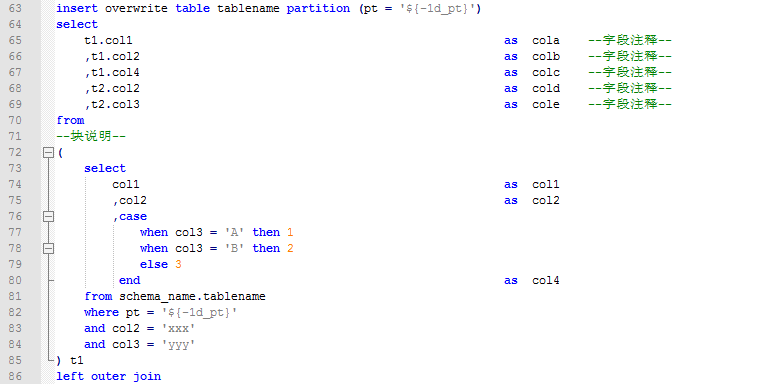


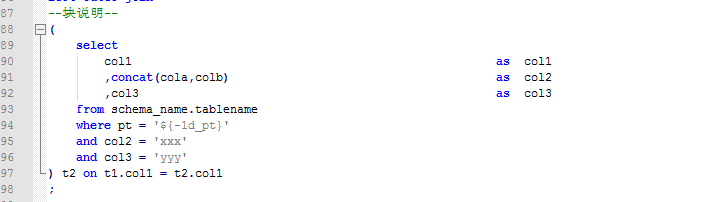
选择【语言】 -> 【勾选用空格代替Tab】



# 4. 格式约定

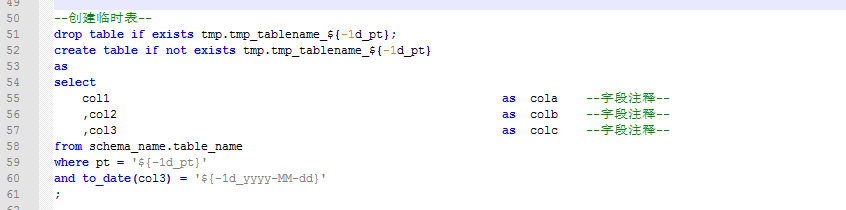
下面是代码实例，用以说明约定格式：





* 所有的SQL关键词、字段名、库名、对象名都小写，用一个空格分隔
* 如果查询的是分区表，where后的过滤条件应该先限定分区，包括一级分区或者二级分区
* Insert/select/from/left outer join/inner join/semi join/full outer join/ where / group by/order by/having/limit 要在一行的开头，同一个语句块中关键词左对齐
* 紧挨Insert关键词的Select语句的字段，需要取别名，且跟目标表字段名一致，“as”要对齐
* Select的字段缩进4个空格，左对齐，逗号放在字段的前面
* Case语句中，Case和End单独成行左对齐，When、Else换行缩进4个空格左对齐
* “<”,”>” ,”=”, ”!=”等运算符前后都需要一个空格隔开
* 子查询中的括号对齐，子查询的右边括号与别名之间用一个空格分开，后面紧挨连接条件
* 有多表、子查询关联查询时，字段必须指定表、子查询别名
* 不予许出现select \*

临时表的使用：



* 创建临时表之前先判断临时表是否存在，存在则删除，在程序结尾删除临时表
* 临时表的命名tmp.tmp\_tablename\_xxx\_${-1d\_pt}，如果一段脚本中有多个临时表，则需要xxx加以区分，否则不需要
* 临时表创建中select之前的as另起一行

# 5. 命名约定

## 5.1 一般规则

* 使用小写的us-english作为标识符命名依据
* 多个标识符通过下划线连接

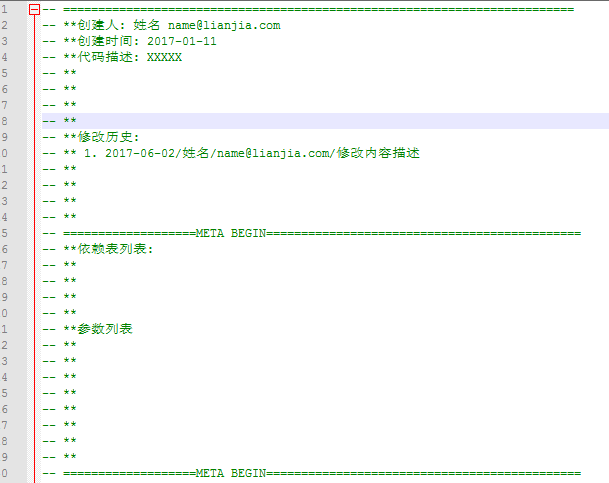
## 5.2 常用规则

* 表的别名尽量简短，建议单字符加阿拉伯数字组成，比如：t1, t2 ……, r1, r2 ……
* 分区参数，尽量用简单的，比如昨的分区，建议用’${-1d\_pt}’
* t第一层，tt第二层，ttt第三层……

# 6. 注释

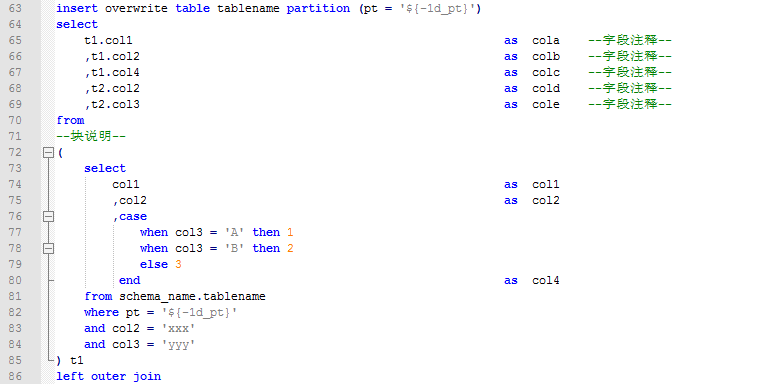
## 6.1 对象注释

对象注释包括创建人、联系方式、创建时间、代码描述、修改历史、依赖表、目标表。



## 6.2 代码注释

代码注释包括语句块的注释、字段的注释、复杂逻辑的注释，如图：



* 每个语句块的开始需要注释本语句块的内容
* 目标表的字段需要注释字段含义且要对齐
* 复杂逻辑需要简要说明处理逻辑

# 7. 模板

