**上海联通大数据平台建设**

**Oracle数据仓库复杂函数**

**到**

**Hive自定义函数转换**

**上海联通信息化事业部**

目录

[1 文档概述 9](#_Toc421653557)

[2 函数总览 9](#_Toc421653558)

[3 Oracle数据仓库DW下函数 12](#_Toc421653559)

[3.1 (已完成)F\_INTEGRAL\_NUMBER 12](#_Toc421653560)

[3.1.1 Oracle数据库中对应OBJECT\_ID 12](#_Toc421653561)

[3.1.2 作用描述 12](#_Toc421653562)

[3.1.3 函数体内容 13](#_Toc421653563)

[3.1.4 Hive中处理思路及方法 14](#_Toc421653564)

[3.2 （已完成）FUNC\_INCOME\_LEVEL\_CODE 17](#_Toc421653565)

[3.2.1 Oracle数据库中对应OBJECT\_ID 17](#_Toc421653566)

[3.2.2 作用描述 17](#_Toc421653567)

[3.2.3 函数体内容 17](#_Toc421653568)

[3.2.4 Hive中处理思路及方法 19](#_Toc421653569)

[3.3 （已完成）FUNC\_PRODUCT\_TYPE\_ID 22](#_Toc421653570)

[3.3.1 Oracle数据库中对应OBJECT\_ID 22](#_Toc421653571)

[3.3.2 作用描述 22](#_Toc421653572)

[3.3.3 函数体内容 22](#_Toc421653573)

[3.3.4 Hive中处理思路及方法 24](#_Toc421653574)

[3.4 （已完成）FUNC\_PRODUCT\_TYPE\_WS 28](#_Toc421653575)

[3.4.1 Oracle数据库中对应OBJECT\_ID 28](#_Toc421653576)

[3.4.2 作用描述 28](#_Toc421653577)

[3.4.3 函数体内容 29](#_Toc421653578)

[3.4.4 Hive中处理思路及方法 33](#_Toc421653579)

[3.5 （已完成）FUNC\_INNET\_DURATION 42](#_Toc421653580)

[3.5.1 Oracle数据库中对应OBJECT\_ID 42](#_Toc421653581)

[3.5.2 作用描述 42](#_Toc421653582)

[3.5.3 函数体内容 42](#_Toc421653583)

[3.5.4 Hive中处理思路及方法 44](#_Toc421653584)

[3.6 （已完成）FUNC\_OWE\_FEE\_LEVEL 46](#_Toc421653585)

[3.6.1 Oracle数据库中对应OBJECT\_ID 46](#_Toc421653586)

[3.6.2 作用描述 46](#_Toc421653587)

[3.6.3 函数体内容 47](#_Toc421653588)

[3.6.4 Hive中处理思路及方法 47](#_Toc421653589)

[3.7 （已完成）FUNC\_INCOME\_LEVEL 49](#_Toc421653590)

[3.7.1 Oracle数据库中对应OBJECT\_ID 49](#_Toc421653591)

[3.7.2 作用描述 49](#_Toc421653592)

[3.7.3 函数体内容 49](#_Toc421653593)

[3.7.4 Hive中处理思路及方法 50](#_Toc421653594)

[3.8 （已完成）FUNC\_CHARGE\_LEVEL 54](#_Toc421653595)

[3.8.1 Oracle数据库中对应OBJECT\_ID 54](#_Toc421653596)

[3.8.2 作用描述 54](#_Toc421653597)

[3.8.3 函数体内容 54](#_Toc421653598)

[3.8.4 Hive中处理思路及方法 55](#_Toc421653599)

[3.9 （已完成）FUNC\_3G\_CHARGE\_LEVEL 57](#_Toc421653600)

[3.9.1 Oracle数据库中对应OBJECT\_ID 57](#_Toc421653601)

[3.9.2 作用描述 58](#_Toc421653602)

[3.9.3 函数体内容 58](#_Toc421653603)

[3.9.4 Hive中处理思路及方法（需补充） 59](#_Toc421653604)

[3.10 （已完成）FUNC\_GW\_UP\_INCOME\_LEVEL 61](#_Toc421653605)

[3.10.1 Oracle数据库中对应OBJECT\_ID 61](#_Toc421653606)

[3.10.2 作用描述 61](#_Toc421653607)

[3.10.3 函数体内容 61](#_Toc421653608)

[3.10.4 Hive中处理思路及方法 63](#_Toc421653609)

[3.11 （已完成）FUNC\_FLUX\_LEVEL 66](#_Toc421653610)

[3.11.1 Oracle数据库中对应OBJECT\_ID 66](#_Toc421653611)

[3.11.2 作用描述 66](#_Toc421653612)

[3.11.3 函数体内容 66](#_Toc421653613)

[3.11.4 Hive中处理思路及方法 67](#_Toc421653614)

[3.12 FUNC\_GW\_COMM\_PRODUCT\_TYPE 69](#_Toc421653615)

[3.12.1 Oracle数据库中对应OBJECT\_ID 69](#_Toc421653616)

[3.12.2 作用描述（需补充） 70](#_Toc421653617)

[3.12.3 函数体内容 70](#_Toc421653618)

[3.12.4 Hive中处理思路及方法（需补充） 72](#_Toc421653619)

[3.13 （已完成）FUNC\_PRODUCT\_TYPE\_YW 72](#_Toc421653620)

[3.13.1 Oracle数据库中对应OBJECT\_ID 72](#_Toc421653621)

[3.13.2 作用描述 72](#_Toc421653622)

[3.13.3 函数体内容 72](#_Toc421653623)

[3.13.4 Hive中处理思路及方法 74](#_Toc421653624)

[3.14 （张海东）F\_GROUP\_ONNET\_USERS\_LVL 76](#_Toc421653625)

[3.14.1 Oracle数据库中对应OBJECT\_ID 76](#_Toc421653626)

[3.14.2 作用描述（需补充） 76](#_Toc421653627)

[3.14.3 函数体内容 76](#_Toc421653628)

[3.14.4 Hive中处理思路及方法（需补充） 77](#_Toc421653629)

[3.15 （张海东）F\_GW\_DURATION\_LEVEL 78](#_Toc421653630)

[3.15.1 Oracle数据库中对应OBJECT\_ID 78](#_Toc421653631)

[3.15.2 作用描述（需补充） 78](#_Toc421653632)

[3.15.3 函数体内容 78](#_Toc421653633)

[3.15.4 Hive中处理思路及方法（需补充） 79](#_Toc421653634)

[3.16 （张海东）F\_GROUP\_CUST\_INCOME\_LVL 79](#_Toc421653635)

[3.16.1 Oracle数据库中对应OBJECT\_ID 79](#_Toc421653636)

[3.16.2 作用描述（需补充） 79](#_Toc421653637)

[3.16.3 函数体内容 79](#_Toc421653638)

[3.16.4 Hive中处理思路及方法（需补充） 80](#_Toc421653639)

[3.17 （张海东）F\_BRAND 80](#_Toc421653640)

[3.17.1 Oracle数据库中对应OBJECT\_ID 80](#_Toc421653641)

[3.17.2 作用描述（需补充） 80](#_Toc421653642)

[3.17.3 函数体内容 80](#_Toc421653643)

[3.17.4 Hive中处理思路及方法（需补充） 81](#_Toc421653644)

[3.18 （张海东）F\_GW\_GROUP\_PRODUCT\_TYPE 81](#_Toc421653645)

[3.18.1 Oracle数据库中对应OBJECT\_ID 81](#_Toc421653646)

[3.18.2 作用描述（需补充） 82](#_Toc421653647)

[3.18.3 函数体内容 82](#_Toc421653648)

[3.18.4 Hive中处理思路及方法（需补充） 83](#_Toc421653649)

[3.19 （张海东）F\_INCOME\_LEVEL 83](#_Toc421653650)

[3.19.1 Oracle数据库中对应OBJECT\_ID 83](#_Toc421653651)

[3.19.2 作用描述（需补充） 83](#_Toc421653652)

[3.19.3 函数体内容 83](#_Toc421653653)

[3.19.4 Hive中处理思路及方法（需补充） 84](#_Toc421653654)

[3.20 （张海东）F\_INNET\_MONTH\_LEVEL\_GW 84](#_Toc421653655)

[3.20.1 Oracle数据库中对应OBJECT\_ID 84](#_Toc421653656)

[3.20.2 作用描述（需补充） 85](#_Toc421653657)

[3.20.3 函数体内容 85](#_Toc421653658)

[3.20.4 Hive中处理思路及方法（需补充） 86](#_Toc421653659)

[3.21 （张海东）F\_GW\_JF\_LEVEL 86](#_Toc421653660)

[3.21.1 Oracle数据库中对应OBJECT\_ID 86](#_Toc421653661)

[3.21.2 作用描述（需补充） 86](#_Toc421653662)

[3.21.3 函数体内容 86](#_Toc421653663)

[3.21.4 Hive中处理思路及方法（需补充） 88](#_Toc421653664)

[3.22 （江峰）F\_GW\_INCOME\_LEVEL 88](#_Toc421653665)

[3.22.1 Oracle数据库中对应OBJECT\_ID 88](#_Toc421653666)

[3.22.2 作用描述（需补充） 88](#_Toc421653667)

[3.22.3 函数体内容 88](#_Toc421653668)

[3.22.4 Hive中处理思路及方法（需补充） 89](#_Toc421653669)

[3.23 （江峰）F\_GW\_INNET\_TIME\_LEVEL 89](#_Toc421653670)

[3.23.1 Oracle数据库中对应OBJECT\_ID 89](#_Toc421653671)

[3.23.2 作用描述（需补充） 89](#_Toc421653672)

[3.23.3 函数体内容 89](#_Toc421653673)

[3.23.4 Hive中处理思路及方法（需补充） 91](#_Toc421653674)

[3.24 （江峰）FUN\_ONETIME\_FEE 91](#_Toc421653675)

[3.24.1 Oracle数据库中对应OBJECT\_ID 91](#_Toc421653676)

[3.24.2 作用描述（需补充） 91](#_Toc421653677)

[3.24.3 函数体内容 91](#_Toc421653678)

[3.24.4 Hive中处理思路及方法（需补充） 92](#_Toc421653679)

[3.25 （江峰）FUN\_MONTH\_FEE\_CHG\_TOBEFORE 92](#_Toc421653680)

[3.25.1 Oracle数据库中对应OBJECT\_ID 92](#_Toc421653681)

[3.25.2 作用描述（需补充） 92](#_Toc421653682)

[3.25.3 函数体内容 92](#_Toc421653683)

[3.25.4 Hive中处理思路及方法（需补充） 94](#_Toc421653684)

[3.26 （江峰）F\_3G\_EYES\_CALL\_COUNTS\_LVL 94](#_Toc421653685)

[3.26.1 Oracle数据库中对应OBJECT\_ID 94](#_Toc421653686)

[3.26.2 作用描述（需补充） 94](#_Toc421653687)

[3.26.3 函数体内容 94](#_Toc421653688)

[3.26.4 Hive中处理思路及方法（需补充） 96](#_Toc421653689)

[3.27 （江峰）FUN\_MONTH\_FEE\_CHG\_BEFORE 96](#_Toc421653690)

[3.27.1 Oracle数据库中对应OBJECT\_ID 96](#_Toc421653691)

[3.27.2 作用描述（需补充） 96](#_Toc421653692)

[3.27.3 函数体内容 96](#_Toc421653693)

[3.27.4 Hive中处理思路及方法（需补充） 98](#_Toc421653694)

[3.28 （江峰）F\_3G\_CALL\_PRE\_LVL 98](#_Toc421653695)

[3.28.1 Oracle数据库中对应OBJECT\_ID 98](#_Toc421653696)

[3.28.2 作用描述（需补充） 98](#_Toc421653697)

[3.28.3 函数体内容 98](#_Toc421653698)

[3.28.4 Hive中处理思路及方法（需补充） 99](#_Toc421653699)

[3.29 （江峰）F\_3G\_CALL\_LONG\_LVL 100](#_Toc421653700)

[3.29.1 Oracle数据库中对应OBJECT\_ID 100](#_Toc421653701)

[3.29.2 作用描述（需补充） 100](#_Toc421653702)

[3.29.3 函数体内容 100](#_Toc421653703)

[3.29.4 Hive中处理思路及方法（需补充） 101](#_Toc421653704)

[3.30 （游标类）GETOLDATTRBYSUBIDATTRCODE 101](#_Toc421653705)

[3.30.1 Oracle数据库中对应OBJECT\_ID 101](#_Toc421653706)

[3.30.2 作用描述（需补充） 102](#_Toc421653707)

[3.30.3 函数体内容 102](#_Toc421653708)

[3.30.4 Hive中处理思路及方法（需补充） 102](#_Toc421653709)

[3.31 （游标类）GETMONTHFEECHGTOBE 103](#_Toc421653710)

[3.31.1 Oracle数据库中对应OBJECT\_ID 103](#_Toc421653711)

[3.31.2 作用描述（需补充） 103](#_Toc421653712)

[3.31.3 函数体内容 103](#_Toc421653713)

[3.31.4 Hive中处理思路及方法（需补充） 105](#_Toc421653714)

[3.32 （游标类）GETMONTHFEECHGBEFORE 105](#_Toc421653715)

[3.32.1 Oracle数据库中对应OBJECT\_ID 105](#_Toc421653716)

[3.32.2 作用描述（需补充） 105](#_Toc421653717)

[3.32.3 函数体内容 105](#_Toc421653718)

[3.32.4 Hive中处理思路及方法（需补充） 107](#_Toc421653719)

[3.33 （游标类）GETOLDSERVICEBYSUBIDSERCODE 107](#_Toc421653720)

[3.33.1 Oracle数据库中对应OBJECT\_ID 107](#_Toc421653721)

[3.33.2 作用描述（需补充） 107](#_Toc421653722)

[3.33.3 函数体内容 107](#_Toc421653723)

[3.33.4 Hive中处理思路及方法（需补充） 108](#_Toc421653724)

[3.34 （游标类）GETSERVICEBYPIIDSERCODE 108](#_Toc421653725)

[3.34.1 Oracle数据库中对应OBJECT\_ID 108](#_Toc421653726)

[3.34.2 作用描述（需补充） 109](#_Toc421653727)

[3.34.3 函数体内容 109](#_Toc421653728)

[3.34.4 Hive中处理思路及方法（需补充） 109](#_Toc421653729)

[3.35 （游标类）GETONETIMEFEEBYSUBORDER 110](#_Toc421653730)

[3.35.1 Oracle数据库中对应OBJECT\_ID 110](#_Toc421653731)

[3.35.2 作用描述（需补充） 110](#_Toc421653732)

[3.35.3 函数体内容 110](#_Toc421653733)

[3.35.4 Hive中处理思路及方法（需补充） 112](#_Toc421653734)

[3.36 （游标类）GETOLDSERVICEBYSUBIDSERKIND 112](#_Toc421653735)

[3.36.1 Oracle数据库中对应OBJECT\_ID 112](#_Toc421653736)

[3.36.2 作用描述（需补充） 112](#_Toc421653737)

[3.36.3 函数体内容 112](#_Toc421653738)

[3.36.4 Hive中处理思路及方法（需补充） 113](#_Toc421653739)

[3.37 （游标类）GETALLCYCLEFEERATE 114](#_Toc421653740)

[3.37.1 Oracle数据库中对应OBJECT\_ID 114](#_Toc421653741)

[3.37.2 作用描述（需补充） 114](#_Toc421653742)

[3.37.3 函数体内容 114](#_Toc421653743)

[3.37.4 Hive中处理思路及方法（需补充） 115](#_Toc421653744)

[3.38 （游标类）GETALLCYCLEFEERATEBYPI 116](#_Toc421653745)

[3.38.1 Oracle数据库中对应OBJECT\_ID 116](#_Toc421653746)

[3.38.2 作用描述（需补充） 116](#_Toc421653747)

[3.38.3 函数体内容 116](#_Toc421653748)

[3.38.4 Hive中处理思路及方法（需补充） 118](#_Toc421653749)

[3.39 （游标类）GETCEASETYPEBYSUBORDERID 118](#_Toc421653750)

[3.39.1 Oracle数据库中对应OBJECT\_ID 118](#_Toc421653751)

[3.39.2 作用描述（需补充） 118](#_Toc421653752)

[3.39.3 函数体内容 118](#_Toc421653753)

[3.39.4 Hive中处理思路及方法（需补充） 119](#_Toc421653754)

[3.40 （游标类）GETATTRBYSUBORDERIDATTRCODE 119](#_Toc421653755)

[3.40.1 Oracle数据库中对应OBJECT\_ID 119](#_Toc421653756)

[3.40.2 作用描述（需补充） 120](#_Toc421653757)

[3.40.3 函数体内容 120](#_Toc421653758)

[3.40.4 Hive中处理思路及方法（需补充） 121](#_Toc421653759)

[3.41 （游标类）GETATTRBYPIIDATTRCODE 121](#_Toc421653760)

[3.41.1 Oracle数据库中对应OBJECT\_ID 121](#_Toc421653761)

[3.41.2 作用描述（需补充） 121](#_Toc421653762)

[3.41.3 函数体内容 121](#_Toc421653763)

[3.41.4 Hive中处理思路及方法（需补充） 122](#_Toc421653764)

[3.42 （游标类）GETSERVICEBYPIIDSERKIND 122](#_Toc421653765)

[3.42.1 Oracle数据库中对应OBJECT\_ID 122](#_Toc421653766)

[3.42.2 作用描述（需补充） 122](#_Toc421653767)

[3.42.3 函数体内容 122](#_Toc421653768)

[3.42.4 Hive中处理思路及方法（需补充） 123](#_Toc421653769)

[3.43 （游标类）GETALLCYCLEFEERATEBYPI\_ALL 124](#_Toc421653770)

[3.43.1 Oracle数据库中对应OBJECT\_ID 124](#_Toc421653771)

[3.43.2 作用描述（需补充） 124](#_Toc421653772)

[3.43.3 函数体内容 124](#_Toc421653773)

[3.43.4 Hive中处理思路及方法（需补充） 127](#_Toc421653774)

[3.44 （游标类）GETSERVATTRINFO 127](#_Toc421653775)

[3.44.1 Oracle数据库中对应OBJECT\_ID 127](#_Toc421653776)

[3.44.2 作用描述（需补充） 127](#_Toc421653777)

[3.44.3 函数体内容 127](#_Toc421653778)

[3.44.4 Hive中处理思路及方法（需补充） 128](#_Toc421653779)

[3.45 （游标类）GETSERVICEBYSUBORDERIDSERKIND 128](#_Toc421653780)

[3.45.1 Oracle数据库中对应OBJECT\_ID 128](#_Toc421653781)

[3.45.2 作用描述（需补充） 129](#_Toc421653782)

[3.45.3 函数体内容 129](#_Toc421653783)

[3.45.4 Hive中处理思路及方法（需补充） 130](#_Toc421653784)

[3.46 （游标类）GETSERVICEBYSUBORDERIDSERCODE 130](#_Toc421653785)

[3.46.1 Oracle数据库中对应OBJECT\_ID 130](#_Toc421653786)

[3.46.2 作用描述（需补充） 130](#_Toc421653787)

[3.46.3 函数体内容 130](#_Toc421653788)

[3.46.4 Hive中处理思路及方法（需补充） 131](#_Toc421653789)

[3.47 （游标类）GETSYSFIELDDESCINFO 131](#_Toc421653790)

[3.47.1 Oracle数据库中对应OBJECT\_ID 131](#_Toc421653791)

[3.47.2 作用描述（需补充） 131](#_Toc421653792)

[3.47.3 函数体内容 131](#_Toc421653793)

[3.47.4 Hive中处理思路及方法（需补充） 132](#_Toc421653794)

[3.48 （游标类）GET\_PROJECTCORRES\_BY\_ORDER\_ID 132](#_Toc421653795)

[3.48.1 Oracle数据库中对应OBJECT\_ID 132](#_Toc421653796)

[3.48.2 作用描述（需补充） 133](#_Toc421653797)

[3.48.3 函数体内容 133](#_Toc421653798)

[3.48.4 Hive中处理思路及方法（需补充） 134](#_Toc421653799)

[3.49 （游标类）GET\_CORRES\_RTN\_BY\_ORDER\_ID 134](#_Toc421653800)

[3.49.1 Oracle数据库中对应OBJECT\_ID 134](#_Toc421653801)

[3.49.2 作用描述（需补充） 134](#_Toc421653802)

[3.49.3 函数体内容 135](#_Toc421653803)

[3.49.4 Hive中处理思路及方法（需补充） 136](#_Toc421653804)

[3.50 （游标类）GETSERVICEBYSUBORDERIDSKVATE 136](#_Toc421653805)

[3.50.1 Oracle数据库中对应OBJECT\_ID 136](#_Toc421653806)

[3.50.2 作用描述（需补充） 136](#_Toc421653807)

[3.50.3 函数体内容 137](#_Toc421653808)

[3.50.4 Hive中处理思路及方法（需补充） 138](#_Toc421653809)

[3.51 （游标类）GETSERVATTRINFOVATE 138](#_Toc421653810)

[3.51.1 Oracle数据库中对应OBJECT\_ID 138](#_Toc421653811)

[3.51.2 作用描述（需补充） 138](#_Toc421653812)

[3.51.3 函数体内容 138](#_Toc421653813)

[3.51.4 Hive中处理思路及方法（需补充） 139](#_Toc421653814)

[3.52 （游标类）GETSERVICEBYPIIDSERKINDVATE 140](#_Toc421653815)

[3.52.1 Oracle数据库中对应OBJECT\_ID 140](#_Toc421653816)

[3.52.2 作用描述（需补充） 140](#_Toc421653817)

[3.52.3 函数体内容 140](#_Toc421653818)

[3.52.4 Hive中处理思路及方法（需补充） 141](#_Toc421653819)

[3.53 （游标类）FUN\_ONETIME\_FEE\_CHG 141](#_Toc421653820)

[3.53.1 Oracle数据库中对应OBJECT\_ID 141](#_Toc421653821)

[3.53.2 作用描述（需补充） 141](#_Toc421653822)

[3.53.3 函数体内容 142](#_Toc421653823)

[3.53.4 Hive中处理思路及方法（需补充） 143](#_Toc421653824)

[3.54 （游标类）FUN\_CYCLE\_FEE\_RATE 143](#_Toc421653825)

[3.54.1 Oracle数据库中对应OBJECT\_ID 143](#_Toc421653826)

[3.54.2 作用描述（需补充） 144](#_Toc421653827)

[3.54.3 函数体内容 144](#_Toc421653828)

[3.54.4 Hive中处理思路及方法（需补充） 146](#_Toc421653829)

[3.55 F\_PARTITION\_EXIST 146](#_Toc421653830)

[3.55.1 Oracle数据库中对应OBJECT\_ID 146](#_Toc421653831)

[3.55.2 作用描述（需补充） 146](#_Toc421653832)

[3.55.3 函数体内容 146](#_Toc421653833)

[3.55.4 Hive中处理思路及方法（需补充） 147](#_Toc421653834)

[3.56 F\_OBJECT\_EXIST 147](#_Toc421653835)

[3.56.1 Oracle数据库中对应OBJECT\_ID 147](#_Toc421653836)

[3.56.2 作用描述（需补充） 147](#_Toc421653837)

[3.56.3 函数体内容 147](#_Toc421653838)

[3.56.4 Hive中处理思路及方法（需补充） 148](#_Toc421653839)

[3.57 F\_WAIT 149](#_Toc421653840)

[3.57.1 Oracle数据库中对应OBJECT\_ID 149](#_Toc421653841)

[3.57.2 作用描述（需补充） 149](#_Toc421653842)

[3.57.3 函数体内容 149](#_Toc421653843)

[3.57.4 Hive中处理思路及方法（需补充） 150](#_Toc421653844)

[3.58 F\_DAY\_PARTITION\_2 150](#_Toc421653845)

[3.58.1 Oracle数据库中对应OBJECT\_ID 150](#_Toc421653846)

[3.58.2 作用描述（需补充） 150](#_Toc421653847)

[3.58.3 函数体内容 150](#_Toc421653848)

[3.58.4 Hive中处理思路及方法（需补充） 151](#_Toc421653849)

[3.59 XPATH\_GET 151](#_Toc421653850)

[3.59.1 Oracle数据库中对应OBJECT\_ID 151](#_Toc421653851)

[3.59.2 作用描述（需补充） 151](#_Toc421653852)

[3.59.3 函数体内容 151](#_Toc421653853)

[3.59.4 Hive中处理思路及方法（需补充） 152](#_Toc421653854)

[3.60 XPATH\_POS 152](#_Toc421653855)

[3.60.1 Oracle数据库中对应OBJECT\_ID 152](#_Toc421653856)

[3.60.2 作用描述（需补充） 152](#_Toc421653857)

[3.60.3 函数体内容 152](#_Toc421653858)

[3.60.4 Hive中处理思路及方法（需补充） 153](#_Toc421653859)

# 文档概述

本文档旨在将Oracle数据仓库复杂函数进行梳理，了解函数具体实现，为后期在Hive中实现此部分函数功能奠定基础。

# 函数总览

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **函数作用** | **原Oracle所属用户** | **原ORACLE函数名** | **Hive函数名** | **输入参数** | **返回** |
| 1 | 积分分档函数 | dw | F\_INTEGRAL\_NUMBER | f\_integral\_number | 积分 | 分档 |
| 2 |  | dw | FUNC\_INCOME\_LEVEL\_CODE | f\_income\_level\_code | key\_source\_type\_cd product\_id user\_type\_id | 业务类型 |
| 3 | 产品分类 | dw | FUNC\_PRODUCT\_TYPE\_ID | f\_product\_type\_id | key\_source\_type product\_id user\_type | 产品类型编码 |
| 4 | 产品分类 | dw | FUNC\_PRODUCT\_TYPE\_WS | f\_product\_type\_ws | key\_source\_type product\_id user\_type | 产品类型编码 |
| 5 | 在网时长分档 | dw | FUNC\_INNET\_DURATION | f\_innet\_duration | 在网时长 | 在网时长分档 |
| 6 | 欠费分档 | dw | FUNC\_OWE\_FEE\_LEVEL | f\_owe\_fee\_level | 欠费 | 欠费分档 |
| 7 |  | dw | FUNC\_INCOME\_LEVEL | f\_income\_level\_func | key\_source\_type\_cd product\_id user\_type\_id | 业务类型 |
| 8 | 收入分档函数 | dw | FUNC\_CHARGE\_LEVEL | f\_charge\_level | 收入 | 收入分档 |
| 9 | 3G用户出账收入分档函数 | dw | FUNC\_3G\_CHARGE\_LEVEL | f\_3g\_charge\_level | 收入 | 收入分档 |
| 10 | 固网月报上报收入分档 | dw | FUNC\_GW\_UP\_INCOME\_LEVEL | f\_gw\_up\_income\_level | 收入 | 收入分档 |
| 11 | 流量分档（指标上报） | dw | FUNC\_FLUX\_LEVEL | f\_flux\_level | 流量 | 流量分档 |
| 12 | 固网佣金产品分类函数 | dw | FUNC\_GW\_COMM\_PRODUCT\_TYPE |  |  |  |
| 13 | 移网产品分类函数 | dw | FUNC\_PRODUCT\_TYPE\_YW | f\_product\_type\_yw | is\_3g\_user card\_type pay\_mode activity\_type dinner\_no trade\_type | 移网产品分类 |
| 14 | 集团网上用户数分档 | dw | F\_GROUP\_ONNET\_USERS\_LVL | f\_group\_onnet\_users\_lvl | group\_user | 用户数分档 |
| 15 | 固网通话时长分档 | dw | F\_GW\_DURATION\_LEVEL | f\_gw\_duration\_level | 通话时长 | 通话时长分档 |
| 16 | 集团业务收入分档 | dw | F\_GROUP\_CUST\_INCOME\_LVL | f\_group\_cust\_income\_lvl | 收入 | 收入分档 |
| 17 | 业务品牌 | dw | F\_BRAND | f\_brand | 商标(brand)，支付类型(pay\_mode) | 业务品牌分档 |
| 18 | 在产品组类型 | dw | F\_GW\_GROUP\_PRODUCT\_TYPE | f\_gw\_group\_product\_type | V\_PROD\_TYPE(单产品ID) V\_USER\_TYPE(用户类型) | 返回在产品组类型 |
| 19 | 出账收入分档 | dw | F\_INCOME\_LEVEL | f\_income\_level | 收入 | 出账收入分档 |
| 20 | 入网时间分段类型 | dw | F\_INNET\_MONTH\_LEVEL\_GW | f\_innet\_month\_level\_gw | 入网时间 | 入网时间类型 |
| 21 | 固网积分分档 | dw | F\_GW\_JF\_LEVEL | f\_gw\_jf\_level | 积分 | 分档 |
| 22 | 固网收入分档 | dw | F\_GW\_INCOME\_LEVEL | f\_gw\_income\_level | 收入 | 分档 |
| 23 | 返回在网时长分档 | dw | F\_GW\_INNET\_TIME\_LEVEL | f\_gw\_innet\_time\_level | 时长 | 分档 |
| 24 |  | dw | FUN\_ONETIME\_FEE |  |  |  |
| 25 |  | dw | FUN\_MONTH\_FEE\_CHG\_TOBEFORE |  |  |  |
| 26 | 可视电话通话次数分档 | dw | F\_3G\_EYES\_CALL\_COUNTS\_LVL | f\_3g\_eyes\_call\_counts\_lvl | 通话次数 | 分档 |
| 27 |  | dw | FUN\_MONTH\_FEE\_CHG\_BEFORE |  |  |  |
| 28 | 通话时长分档 | dw | F\_3G\_CALL\_PRE\_LVL | f\_3g\_call\_pre\_lvl | 通话时长 | 分档 |
| 29 | 通话时长分档 | dw | F\_3G\_CALL\_LONG\_LVL | f\_3g\_call\_long\_lvl | 时长 | 分档 |
| 30 |  | dw | GETOLDATTRBYSUBIDATTRCODE |  |  |  |
| 31 |  | dw | GETMONTHFEECHGTOBE |  |  |  |
| 32 |  | dw | GETMONTHFEECHGBEFORE |  |  |  |
| 33 |  | dw | GETOLDSERVICEBYSUBIDSERCODE |  |  |  |
| 34 |  | dw | GETSERVICEBYPIIDSERCODE |  |  |  |
| 35 |  | dw | GETONETIMEFEEBYSUBORDER |  |  |  |
| 36 |  | dw | GETOLDSERVICEBYSUBIDSERKIND |  |  |  |
| 37 |  | dw | GETALLCYCLEFEERATE |  |  |  |
| 38 |  | dw | GETALLCYCLEFEERATEBYPI |  |  |  |
| 39 |  | dw | GETCEASETYPEBYSUBORDERID |  |  |  |
| 40 |  | dw | GETATTRBYSUBORDERIDATTRCODE |  |  |  |
| 41 |  | dw | GETATTRBYPIIDATTRCODE |  |  |  |
| 42 |  | dw | GETSERVICEBYPIIDSERKIND |  |  |  |
| 43 |  | dw | GETALLCYCLEFEERATEBYPI\_ALL |  |  |  |
| 44 |  | dw | GETSERVATTRINFO |  |  |  |
| 45 |  | dw | GETSERVICEBYSUBORDERIDSERKIND |  |  |  |
| 46 |  | dw | GETSERVICEBYSUBORDERIDSERCODE |  |  |  |
| 47 |  | dw | GETSYSFIELDDESCINFO |  |  |  |
| 48 |  | dw | GET\_PROJECTCORRES\_BY\_ORDER\_ID |  |  |  |
| 49 |  | dw | GET\_CORRES\_RTN\_BY\_ORDER\_ID |  |  |  |
| 50 |  | dw | GETSERVICEBYSUBORDERIDSKVATE |  |  |  |
| 51 |  | dw | GETSERVATTRINFOVATE |  |  |  |
| 52 |  | dw | GETSERVICEBYPIIDSERKINDVATE |  |  |  |
| 53 |  | dw | FUN\_ONETIME\_FEE\_CHG |  |  |  |
| 54 |  | dw | FUN\_CYCLE\_FEE\_RATE |  |  |  |
| 55 |  | dw | F\_PARTITION\_EXIST |  |  |  |
| 56 |  | dw | F\_OBJECT\_EXIST |  |  |  |
| 57 |  | dw | F\_WAIT |  |  |  |
| 58 |  | dw | F\_DAY\_PARTITION\_2 |  |  |  |
| 59 |  | dw | XPATH\_GET |  |  |  |
| 60 |  | dw | XPATH\_POS |  |  |  |

# Oracle数据仓库DW下函数

## (已完成)F\_INTEGRAL\_NUMBER

### Oracle数据库中对应OBJECT\_ID

350402

### 作用描述

积分分档函数

### 函数体内容

FUNCTION F\_INTEGRAL\_NUMBER(INTEGRAL\_NUMBER IN NUMBER)

RETURN VARCHAR2 IS

/\*-------------------------------------------------------------------------------

函 数 名 ： F\_INTEGRAL\_NUMBER(积分数分档)

备 注 ：

编 写 人 ：HULEI

编写时间 : 20100917

-------------------------------------------------------------------------------\*/

RESULT VARCHAR2(2);

BEGIN

RESULT := CASE WHEN INTEGRAL\_NUMBER>=0 AND INTEGRAL\_NUMBER<1000 THEN '01'

WHEN INTEGRAL\_NUMBER>=1000 AND INTEGRAL\_NUMBER<2000 THEN '02'

WHEN INTEGRAL\_NUMBER>=2000 AND INTEGRAL\_NUMBER<3000 THEN '03'

WHEN INTEGRAL\_NUMBER>=3000 AND INTEGRAL\_NUMBER<4000 THEN '04'

WHEN INTEGRAL\_NUMBER>=4000 AND INTEGRAL\_NUMBER<5000 THEN '05'

WHEN INTEGRAL\_NUMBER>=5000 AND INTEGRAL\_NUMBER<6000 THEN '06'

WHEN INTEGRAL\_NUMBER>=6000 AND INTEGRAL\_NUMBER<7000 THEN '07'

WHEN INTEGRAL\_NUMBER>=7000 AND INTEGRAL\_NUMBER<8000 THEN '08'

WHEN INTEGRAL\_NUMBER>=8000 AND INTEGRAL\_NUMBER<9000 THEN '09'

WHEN INTEGRAL\_NUMBER>=9000 AND INTEGRAL\_NUMBER<10000 THEN '10'

WHEN INTEGRAL\_NUMBER>=10000 AND INTEGRAL\_NUMBER<11000 THEN '11'

WHEN INTEGRAL\_NUMBER>=11000 AND INTEGRAL\_NUMBER<12000 THEN '12'

WHEN INTEGRAL\_NUMBER>=12000 AND INTEGRAL\_NUMBER<13000 THEN '13'

WHEN INTEGRAL\_NUMBER>=13000 AND INTEGRAL\_NUMBER<14000 THEN '14'

WHEN INTEGRAL\_NUMBER>=14000 AND INTEGRAL\_NUMBER<15000 THEN '15'

WHEN INTEGRAL\_NUMBER>=15000 AND INTEGRAL\_NUMBER<16000 THEN '16'

WHEN INTEGRAL\_NUMBER>=16000 AND INTEGRAL\_NUMBER<17000 THEN '17'

WHEN INTEGRAL\_NUMBER>=17000 AND INTEGRAL\_NUMBER<18000 THEN '18'

WHEN INTEGRAL\_NUMBER>=18000 AND INTEGRAL\_NUMBER<19000 THEN '19'

WHEN INTEGRAL\_NUMBER>=19000 AND INTEGRAL\_NUMBER<20000 THEN '20'

WHEN INTEGRAL\_NUMBER>=20000 AND INTEGRAL\_NUMBER<30000 THEN '21'

WHEN INTEGRAL\_NUMBER>=30000 AND INTEGRAL\_NUMBER<40000 THEN '22'

WHEN INTEGRAL\_NUMBER>=40000 AND INTEGRAL\_NUMBER<50000 THEN '23'

WHEN INTEGRAL\_NUMBER>=50000 AND INTEGRAL\_NUMBER<60000 THEN '24'

WHEN INTEGRAL\_NUMBER>=60000 AND INTEGRAL\_NUMBER<70000 THEN '25'

WHEN INTEGRAL\_NUMBER>=70000 AND INTEGRAL\_NUMBER<80000 THEN '26'

WHEN INTEGRAL\_NUMBER>=80000 AND INTEGRAL\_NUMBER<90000 THEN '27'

WHEN INTEGRAL\_NUMBER>=90000 AND INTEGRAL\_NUMBER<100000 THEN '28'

WHEN INTEGRAL\_NUMBER>=100000 THEN '29'

ELSE '01'

END;

RETURN(RESULT);

END F\_INTEGRAL\_NUMBER;

### Hive中处理思路及方法

#### 思路

通过IF判断，对传入参数（积分）进行积分分档

#### 类名

org.apache.hadoop.hive.ql.udf. FIntegralNumber.java

#### HIVE函数名

f\_integral\_number

#### Hive自定义函数代码

**package** org.apache.hadoop.hive.ql.udf;

**import** org.apache.hadoop.hive.ql.exec.UDF;

/\*

\* 函数功能：积分数分档

\* Oracle中创建者：HULEI

\* Hive自定义函数创建：平晓刚

\* 创建时间：2015-06-08

\* 对应原Oracle函数：DW.F\_INTEGRAL\_NUMBER

\* 传入参数：积分

\* 返回参数：分档

\* \*/

**public** **class** FIntegralNumber **extends** UDF {

**public** String evaluate(**double** INTEGRAL\_NUMBER) {

String return\_valu = **null**;

**if** (INTEGRAL\_NUMBER >= 0 && INTEGRAL\_NUMBER < 1000) {

return\_valu = "01";

}**else** **if**(INTEGRAL\_NUMBER >= 1000 && INTEGRAL\_NUMBER < 2000){

return\_valu = "02";

}**else** **if**(INTEGRAL\_NUMBER >= 2000 && INTEGRAL\_NUMBER < 3000){

return\_valu = "03";

}**else** **if**(INTEGRAL\_NUMBER >= 3000 && INTEGRAL\_NUMBER < 4000){

return\_valu = "04";

}**else** **if**(INTEGRAL\_NUMBER >= 4000 && INTEGRAL\_NUMBER < 5000){

return\_valu = "05";

}**else** **if**(INTEGRAL\_NUMBER >= 5000 && INTEGRAL\_NUMBER < 6000){

return\_valu = "06";

}**else** **if**(INTEGRAL\_NUMBER >= 6000 && INTEGRAL\_NUMBER < 7000){

return\_valu = "07";

}**else** **if**(INTEGRAL\_NUMBER >= 7000 && INTEGRAL\_NUMBER < 8000){

return\_valu = "08";

}**else** **if**(INTEGRAL\_NUMBER >= 8000 && INTEGRAL\_NUMBER < 9000){

return\_valu = "09";

}**else** **if**(INTEGRAL\_NUMBER >= 9000 && INTEGRAL\_NUMBER < 10000){

return\_valu = "10";

}**else** **if**(INTEGRAL\_NUMBER >= 10000 && INTEGRAL\_NUMBER < 11000){

return\_valu = "11";

}**else** **if**(INTEGRAL\_NUMBER >= 11000 && INTEGRAL\_NUMBER < 12000){

return\_valu = "12";

}**else** **if**(INTEGRAL\_NUMBER >= 12000 && INTEGRAL\_NUMBER < 13000){

return\_valu = "13";

}**else** **if**(INTEGRAL\_NUMBER >= 13000 && INTEGRAL\_NUMBER < 14000){

return\_valu = "14";

}**else** **if**(INTEGRAL\_NUMBER >= 14000 && INTEGRAL\_NUMBER < 15000){

return\_valu = "15";

}**else** **if**(INTEGRAL\_NUMBER >= 15000 && INTEGRAL\_NUMBER < 16000){

return\_valu = "16";

}**else** **if**(INTEGRAL\_NUMBER >= 16000 && INTEGRAL\_NUMBER < 17000){

return\_valu = "17";

}**else** **if**(INTEGRAL\_NUMBER >= 17000 && INTEGRAL\_NUMBER < 18000){

return\_valu = "18";

}**else** **if**(INTEGRAL\_NUMBER >= 18000 && INTEGRAL\_NUMBER < 19000){

return\_valu = "19";

}**else** **if**(INTEGRAL\_NUMBER >= 19000 && INTEGRAL\_NUMBER < 20000){

return\_valu = "20";

}**else** **if**(INTEGRAL\_NUMBER >= 20000 && INTEGRAL\_NUMBER < 30000){

return\_valu = "21";

}**else** **if**(INTEGRAL\_NUMBER >= 30000 && INTEGRAL\_NUMBER < 40000){

return\_valu = "22";

}**else** **if**(INTEGRAL\_NUMBER >= 40000 && INTEGRAL\_NUMBER < 50000){

return\_valu = "23";

}**else** **if**(INTEGRAL\_NUMBER >= 50000 && INTEGRAL\_NUMBER < 60000){

return\_valu = "24";

}**else** **if**(INTEGRAL\_NUMBER >= 60000 && INTEGRAL\_NUMBER < 70000){

return\_valu = "25";

}**else** **if**(INTEGRAL\_NUMBER >= 70000 && INTEGRAL\_NUMBER < 80000){

return\_valu = "26";

}**else** **if**(INTEGRAL\_NUMBER >= 80000 && INTEGRAL\_NUMBER < 90000){

return\_valu = "27";

}**else** **if**(INTEGRAL\_NUMBER >= 90000 && INTEGRAL\_NUMBER < 100000){

return\_valu = "28";

}**else** **if**(INTEGRAL\_NUMBER >= 100000){

return\_valu = "29";

}**else**{

return\_valu = "01";

}

**return** return\_valu;

}

}

## （已完成）FUNC\_INCOME\_LEVEL\_CODE

### Oracle数据库中对应OBJECT\_ID

350377

### 作用描述

### 函数体内容

FUNCTION func\_income\_level\_code(key\_source\_type\_cd varchar2,

product\_id varchar2,

user\_type\_id varchar2)

/\*-----------------------------------------------------------------------

修改内容：收入分档

修 改 人：东方国信 LUZHE

修改时间：20061130

------------------------------------------------------------------------\*/

RETURN VARCHAR2 IS

BEGIN

IF key\_source\_type\_cd = '52007' THEN

RETURN('C08');

ELSIF key\_source\_type\_cd = '52003' THEN

RETURN('C11');

ELSIF key\_source\_type\_cd = '52005' THEN

RETURN('C12');

ELSIF key\_source\_type\_cd = '52006' THEN

RETURN('C17');

ELSIF key\_source\_type\_cd = '52004' and product\_id not in ('142035', '142848') THEN

RETURN('C03');

ELSIF key\_source\_type\_cd = '52004' and product\_id = '142848' THEN

RETURN('C06');

ELSIF key\_source\_type\_cd = '52004' and product\_id = '142035' THEN

RETURN('C07');

ELSIF key\_source\_type\_cd = '52001' and product\_id not in ('142877') and

user\_type\_id not in ('143807', '900130', '900150') THEN

RETURN('C02');

ELSIF key\_source\_type\_cd = '52001' and product\_id = '142877' THEN

RETURN('C04');

ELSIF key\_source\_type\_cd = '52001' and user\_type\_id in ('143807', '900150') THEN

RETURN('C01');

ELSIF key\_source\_type\_cd = '52001' and user\_type\_id = '900130' THEN

RETURN('C05');

ELSIF key\_source\_type\_cd = '52002' and

product\_id in ('143668', '143669', '955243', '955244') THEN

RETURN('C15');

ELSIF key\_source\_type\_cd = '52002' and product\_id in ('143678', '143679') THEN

RETURN('C19');

ELSIF key\_source\_type\_cd = '52002' and product\_id in ('143674', '143675') THEN

RETURN('C16');

ELSIF key\_source\_type\_cd = '52002' and product\_id in ('143676', '143677') THEN

RETURN('C20');

ELSIF key\_source\_type\_cd = '52002' and product\_id in ('143672', '143673') THEN

RETURN('C21');

ELSIF key\_source\_type\_cd = '52002' and product\_id in ('143670', '143671') THEN

RETURN('C18');

ELSIF key\_source\_type\_cd = '52002' and product\_id in ('142149', '142126') THEN

RETURN('C22');

ELSIF key\_source\_type\_cd = '52002' and

product\_id in ('143607', '901359', '901337', '143604') THEN

RETURN('C00');

ELSIF key\_source\_type\_cd = '52002' and product\_id in ('901104') THEN

RETURN('C24');

ELSIF key\_source\_type\_cd = '52002' and

product\_id in

('956101', '956102', '956103', '956104', '956105', '956106', '956107', '956108') THEN

RETURN('C25');

ELSIF key\_source\_type\_cd = '52014' THEN

RETURN('C23');

else

return('C99');

END IF;

END;

### Hive中处理思路及方法

#### 思路

通过IF判断，进行相关代码转换

#### 类名

org.apache.hadoop.hive.ql.udf. FuncIncomeLevelCode.java

#### HIVE函数名

f\_income\_level\_code

#### Hive自定义函数代码

**package** org.apache.hadoop.hive.ql.udf;

**import** java.util.ArrayList;

**import** java.util.List;

**import** org.apache.hadoop.hive.ql.exec.UDF;

/\*

\* 函数功能：

\* Oracle中创建者：

\* Hive自定义函数创建：平晓刚

\* 创建时间：2015-06-08

\* 对应原Oracle函数：DW.FUNC\_INCOME\_LEVEL\_CODE

\* 传入参数1：

\* 传入参数2：

\* 传入参数3：

\* 返回参数：

\* \*/

**public** **class** FuncIncomeLevelCode **extends** UDF {

**public** String evaluate(String key\_source\_type\_cd, String product\_id,

String user\_type\_id) {

String output = **null**;

List key\_source\_52004 = **new** ArrayList();

key\_source\_52004.add("142035");

key\_source\_52004.add("142848");

List key\_source\_52001\_user\_type = **new** ArrayList();

key\_source\_52001\_user\_type.add("143807");

key\_source\_52001\_user\_type.add("900130");

key\_source\_52001\_user\_type.add("900150");

**if** (key\_source\_type\_cd.equals("52007") ){

output="C08";

}**else** **if**(key\_source\_type\_cd.equals("52003")){

output="C11";

}**else** **if**(key\_source\_type\_cd.equals("52005")){

output="C12";

}**else** **if**(key\_source\_type\_cd.equals("52006")){

output="C17";

}**else** **if**(key\_source\_type\_cd.equals("52004") && !key\_source\_52004.contains(product\_id)){

output="C03";

}**else** **if**(key\_source\_type\_cd.equals("52004") && product\_id.equals("142848")){

output="C06";

}**else** **if**(key\_source\_type\_cd.equals("52004") && product\_id.equals("142035")){

output="C07";

}**else** **if**(key\_source\_type\_cd.equals("52001") && product\_id!=("142877") && !key\_source\_52001\_user\_type.contains(user\_type\_id)){

output="C02";

}**else** **if**(key\_source\_type\_cd.equals("52001") && product\_id.equals("142877")){

output="C04";

}**else** **if**(key\_source\_type\_cd.equals("52001") && user\_type\_id.equals("900130")){

output="C05";

}**else** **if**(key\_source\_type\_cd.equals("52001") && (user\_type\_id.equals("143807") | user\_type\_id.equals("900150"))){

output="C01";

}**else** **if**(key\_source\_type\_cd.equals("52002") && (product\_id.equals("143668") | product\_id.equals("143669")|product\_id.equals("955243") | product\_id.equals("955244"))){

output="C15";

}**else** **if**(key\_source\_type\_cd.equals("52002") && (product\_id.equals("143678") | product\_id.equals("143679"))){

output="C19";

}**else** **if**(key\_source\_type\_cd.equals("52002") && (product\_id.equals("143674") | product\_id.equals("143675"))){

output="C16";

}**else** **if**(key\_source\_type\_cd.equals("52002") && (product\_id.equals("143676") | product\_id.equals("143677"))){

output="C20";

}**else** **if**(key\_source\_type\_cd.equals("52002") && (product\_id.equals("143672") | product\_id.equals("143673"))){

output="C21";

}**else** **if**(key\_source\_type\_cd.equals("52002") && (product\_id.equals("143670") | product\_id.equals("143671"))){

output="C18";

}**else** **if**(key\_source\_type\_cd.equals("52002") && (product\_id.equals("142149") | product\_id.equals("142126"))){

output="C22";

}**else** **if**(key\_source\_type\_cd.equals("52002") && (product\_id.equals("143607") | product\_id.equals("901359")|product\_id.equals("901337") | product\_id.equals("143604"))){

output="C00";

}**else** **if**(key\_source\_type\_cd.equals("52002") && product\_id.equals("901104")){

output="C24";

}**else** **if**(key\_source\_type\_cd.equals("52002") && (product\_id.equals("956101") | product\_id.equals("956102")|product\_id.equals("956103") | product\_id.equals("956104") | product\_id.equals("956105") | product\_id.equals("956106")|product\_id.equals("956107") | product\_id.equals("956108"))){

output="C25";

}**else** **if**(key\_source\_type\_cd.equals("52014") ){

output="C23";

}**else**{

output="C99";

}

**return** output;

}

}

## （已完成）FUNC\_PRODUCT\_TYPE\_ID

### Oracle数据库中对应OBJECT\_ID

350380

### 作用描述

产品分类函数

### 函数体内容

FUNCTION FUNC\_PRODUCT\_TYPE\_ID(KEY\_SOURCE\_TYPE VARCHAR2,

PRODUCT\_ID NUMBER,

USER\_TYPE NUMBER)

/\*-----------------------------------------------------------------------

修改内容：产品分类函数

修 改 人：东方国信

修改时间：20110424

------------------------------------------------------------------------\*/

RETURN VARCHAR2 IS

BEGIN

IF KEY\_SOURCE\_TYPE = '52007' THEN

RETURN('020201'); --800,4006

ELSIF KEY\_SOURCE\_TYPE = '52001' AND

PRODUCT\_ID IN (142717, 141792, 900097, 3074, 141727, 141729) AND

USER\_TYPE NOT IN (143807, 900130, 900150) THEN

RETURN('020202'); --'E1'

ELSIF KEY\_SOURCE\_TYPE = '52004' AND PRODUCT\_ID NOT IN (142035, 142848) THEN

RETURN('020203'); -- 家庭固话 原'宽带电话(NGN)'

ELSIF KEY\_SOURCE\_TYPE = '52001' AND PRODUCT\_ID IN (8, 9, 141974, 142763) AND

USER\_TYPE NOT IN (143807, 900130, 900150) THEN

RETURN('020204'); --家庭模拟线

ELSIF KEY\_SOURCE\_TYPE = '52001' AND

PRODUCT\_ID IN (955239, 955240, 955238, 955237) AND

USER\_TYPE NOT IN (143807, 900130, 900150) THEN

RETURN('020205'); --企业模拟线

ELSIF KEY\_SOURCE\_TYPE = '52001' AND PRODUCT\_ID = 142877 THEN

RETURN('020206'); --'网通新传真'

ELSIF KEY\_SOURCE\_TYPE = '52001' AND USER\_TYPE IN (143807, 900150) THEN

RETURN('020207'); --'公话'

ELSIF KEY\_SOURCE\_TYPE = '52001' AND USER\_TYPE = 900130 THEN

RETURN('020208'); --'话批'

ELSIF KEY\_SOURCE\_TYPE = '52004' AND PRODUCT\_ID = 142848 THEN

RETURN('020209'); --'空中会议室'

ELSIF KEY\_SOURCE\_TYPE = '52004' AND PRODUCT\_ID = 142035 THEN

RETURN('020210'); --'一号通后付'

ELSIF KEY\_SOURCE\_TYPE = '52001' AND

PRODUCT\_ID NOT IN

(142717, 141792, 900097, 3074, 141727, 141729, 8, 9, 141974, 142763,

955239, 955240, 955238, 955237, 142877) AND

USER\_TYPE NOT IN (143807, 900130, 900150) THEN

RETURN('020211'); --'其他语音业务'

ELSIF KEY\_SOURCE\_TYPE = '52002' AND

PRODUCT\_ID IN (143668, 143669, 955243, 955244) THEN

RETURN('020101'); --DIA

ELSIF KEY\_SOURCE\_TYPE = '52002' AND PRODUCT\_ID IN (143674, 143675) THEN

RETURN('020102'); --DPLC

ELSIF KEY\_SOURCE\_TYPE = '52002' AND PRODUCT\_ID IN (143678, 143679) THEN

RETURN('020103'); --IDC

ELSIF KEY\_SOURCE\_TYPE = '52006' THEN

RETURN('020104'); --CPN

ELSIF KEY\_SOURCE\_TYPE = '52002' AND PRODUCT\_ID IN (143676, 143677) THEN

RETURN('020105'); --IPLC

ELSIF KEY\_SOURCE\_TYPE = '52002' AND PRODUCT\_ID IN (143672, 143673) THEN

RETURN('020106'); --MPLS VPN

ELSIF KEY\_SOURCE\_TYPE = '52002' AND PRODUCT\_ID IN (143670, 143671) THEN

RETURN('020107'); --FR VPN

ELSIF KEY\_SOURCE\_TYPE = '52002' AND PRODUCT\_ID IN (142149, 142126) THEN

RETURN('020108'); --WLAN

ELSIF KEY\_SOURCE\_TYPE = '52002' AND

PRODUCT\_ID IN (143607, 901359, 901337, 143604) THEN

RETURN('020109'); --网元出租

ELSIF KEY\_SOURCE\_TYPE = '52002' AND PRODUCT\_ID IN (901104) THEN

RETURN('020110'); --互联网

ELSIF KEY\_SOURCE\_TYPE = '52002' AND

PRODUCT\_ID IN

(956101, 956102, 956103, 956104, 956105, 956106, 956107, 956108) THEN

RETURN('020111'); --165ZX

ELSIF KEY\_SOURCE\_TYPE = '-1' AND PRODUCT\_ID = 99999901 THEN

RETURN('020302'); --196200196201卡

ELSIF KEY\_SOURCE\_TYPE = '-1' AND PRODUCT\_ID = 99999902 THEN

RETURN('020112'); --跨域

ELSIF KEY\_SOURCE\_TYPE = '-1' AND PRODUCT\_ID = 99999903 THEN

RETURN('020303'); --信息传媒

ELSIF KEY\_SOURCE\_TYPE = '-1' AND PRODUCT\_ID = 99999904 THEN

RETURN('020113'); --跨区集团DA

ELSIF KEY\_SOURCE\_TYPE = '-1' AND PRODUCT\_ID = 99999905 THEN

RETURN('020114'); --纯本地集团DA

ELSIF KEY\_SOURCE\_TYPE = '-1' AND PRODUCT\_ID = 99999906 THEN

RETURN('020115'); --纯跨区DA

ELSE

RETURN('020301'); --其他

END IF;

END;

### Hive中处理思路及方法

#### 思路

通过IF判断，进行相关代码转换

#### 类名

org.apache.hadoop.hive.ql.udf. FuncProductTypeId.java

#### HIVE函数名

f\_product\_type\_id

#### Hive自定义函数代码

**package** org.apache.hadoop.hive.ql.udf;

**import** java.util.ArrayList;

**import** java.util.List;

**import** org.apache.hadoop.hive.ql.exec.UDF;

/\*

\* 函数功能：产品分类函数

\* Oracle中创建者：

\* Hive自定义函数创建：平晓刚

\* 创建时间：2015-06-08

\* 对应原Oracle函数：DW.2.34 FUNC\_PRODUCT\_TYPE\_ID

\* 传入参数1：

\* 传入参数2：

\* 传入参数3：

\* 返回参数：

\* \*/

**public** **class** FuncProductTypeId **extends** UDF {

**public** String evaluate(String key\_source\_type, String product\_id,

String user\_type) {

String output = **null**;

List key\_source\_52001\_user\_type = **new** ArrayList();

key\_source\_52001\_user\_type.add("143807");

key\_source\_52001\_user\_type.add("900130");

key\_source\_52001\_user\_type.add("900150");

List key\_source\_52004\_product = **new** ArrayList();

key\_source\_52004\_product.add("142035");

key\_source\_52004\_product.add("142848");

List key\_source\_52001\_product = **new** ArrayList();

key\_source\_52001\_product.add("142717");

key\_source\_52001\_product.add("141792");

key\_source\_52001\_product.add("900097");

key\_source\_52001\_product.add("3074");

key\_source\_52001\_product.add("141727");

key\_source\_52001\_product.add("141729");

key\_source\_52001\_product.add("8");

key\_source\_52001\_product.add("9");

key\_source\_52001\_product.add("141974");

key\_source\_52001\_product.add("142763");

key\_source\_52001\_product.add("955239");

key\_source\_52001\_product.add("955240");

key\_source\_52001\_product.add("955238");

key\_source\_52001\_product.add("955237");

key\_source\_52001\_product.add("142877");

**if** (key\_source\_type.equals("52007")) {

output = "020201";// 800,4006

} **else** **if** (key\_source\_type.equals("52001")

&& (product\_id.equals("142717") | product\_id.equals("141792")

| product\_id.equals("900097")

| product\_id.equals("3074")

| product\_id.equals("141727") | product\_id

.equals("141729"))

&& !key\_source\_52001\_user\_type.contains(user\_type)) {

output = "020202";// 'e1'

} **else** **if** (key\_source\_type.equals("52004")

&& !key\_source\_52004\_product.contains(product\_id)) {

output = "020203";// 家庭固话 原'宽带电话(NGN)'

} **else** **if** (key\_source\_type.equals("52001")

&& (product\_id.equals("8") | product\_id.equals("9")

| product\_id.equals("141974") | product\_id

.equals("142763"))

&& !key\_source\_52001\_user\_type.contains(user\_type)) {

output = "020204";// 家庭模拟线

} **else** **if** (key\_source\_type.equals("52001")

&& (product\_id.equals("955239") | product\_id.equals("955240")

| product\_id.equals("955238") | product\_id

.equals("955237"))

&& !key\_source\_52001\_user\_type.contains(user\_type)) {

output = "020205";// 企业模拟线

} **else** **if** (key\_source\_type.equals("52001")

&& product\_id.equals("142877")) {

output = "020206";// 网通新传真

} **else** **if** (key\_source\_type.equals("52001")

&& (user\_type.equals("143807") | user\_type.equals("900150"))) {

output = "020207";// 公话

} **else** **if** (key\_source\_type.equals("52001")

&& user\_type.equals("900130")) {

output = "020208";// 话批

} **else** **if** (key\_source\_type.equals("52004")

&& product\_id.equals("142848")) {

output = "020209";// 空中会议室

} **else** **if** (key\_source\_type.equals("52004")

&& product\_id.equals("142035")) {

output = "020210";// 一号通后付

} **else** **if** (key\_source\_type.equals("52001")

&& !key\_source\_52001\_product.contains(product\_id)

&& !key\_source\_52001\_user\_type.contains(user\_type)) {

output = "020211";// 其他语音业务

} **else** **if** (key\_source\_type.equals("52002")

&& (product\_id.equals("143668") | product\_id.equals("143669")

| product\_id.equals("955243") | product\_id

.equals("955244"))) {

output = "020101";// DIA

} **else** **if** (key\_source\_type.equals("52002")

&& (product\_id.equals("143674") | product\_id.equals("143675"))) {

output = "020102";// DPLC

} **else** **if** (key\_source\_type.equals("52002")

&& (product\_id.equals("143678") | product\_id.equals("143679"))) {

output = "020103";// IDC

} **else** **if** (key\_source\_type.equals("52006")) {

output = "020104";// CPN

} **else** **if** (key\_source\_type.equals("52002")

&& (product\_id.equals("143676") | product\_id.equals("143677"))) {

output = "020105";// IPLC

} **else** **if** (key\_source\_type.equals("52002")

&& (product\_id.equals("143672") | product\_id.equals("143673"))) {

output = "020106";// MPLS VPN

} **else** **if** (key\_source\_type.equals("52002")

&& (product\_id.equals("143670") | product\_id.equals("143671"))) {

output = "020107";// FR VPN

} **else** **if** (key\_source\_type.equals("52002")

&& (product\_id.equals("142149") | product\_id.equals("142126"))) {

output = "020108";// WLAN

} **else** **if** (key\_source\_type.equals("52002")

&& (product\_id.equals("143607") | product\_id.equals("901359")

| product\_id.equals("901337") | product\_id

.equals("143604"))) {

output = "020109";// 网元出租

} **else** **if** (key\_source\_type.equals("52002")

&& product\_id.equals("901104")) {

output = "020110";// 互联网

} **else** **if** (key\_source\_type.equals("52002")

&& (product\_id.equals("956101") | product\_id.equals("956102")

| product\_id.equals("956103")

| product\_id.equals("956104")

| product\_id.equals("956105")

| product\_id.equals("956106")

| product\_id.equals("956107") | product\_id

.equals("956108"))) {

output = "020111";// 165ZX

} **else** **if** (key\_source\_type.equals("-1")

&& product\_id.equals("99999901")) {

output = "020302";// 196200196201卡

} **else** **if** (key\_source\_type.equals("-1")

&& product\_id.equals("99999902")) {

output = "020112";// 跨域

} **else** **if** (key\_source\_type.equals("-1")

&& product\_id.equals("99999903")) {

output = "020303";// 信息传媒

} **else** **if** (key\_source\_type.equals("-1")

&& product\_id.equals("99999904")) {

output = "020113";// 跨区集团DA

} **else** **if** (key\_source\_type.equals("-1")

&& product\_id.equals("99999905")) {

output = "020114";// 纯本地集团DA

} **else** **if** (key\_source\_type.equals("-1")

&& product\_id.equals("99999906")) {

output = "020115";// 纯跨区DA

} **else** {

output = "020301";// 其他

}

**return** output;

}

}

## （已完成）FUNC\_PRODUCT\_TYPE\_WS

### Oracle数据库中对应OBJECT\_ID

350381

### 作用描述

### 函数体内容

FUNCTION FUNC\_PRODUCT\_TYPE\_WS(KEY\_SOURCE\_TYPE varchar2,

PRODUCT\_ID VARCHAR2,

USER\_TYPE varchar2)

/\*-----------------------------------------------------------------------

修改内容：产品分类函数

修 改 人：王松

修改时间：20120621

------------------------------------------------------------------------\*/

RETURN VARCHAR2 IS

BEGIN

IF PRODUCT\_ID = '901272' AND KEY\_SOURCE\_TYPE = '52014' AND

USER\_TYPE IN ('142565', '901191') THEN

RETURN('020101'); --备件与现场服务

ELSIF PRODUCT\_ID = '901212' AND KEY\_SOURCE\_TYPE = '52014' AND

USER\_TYPE = '901191' THEN

RETURN('020102'); --第三方软硬件销售

ELSIF PRODUCT\_ID = '901208' AND KEY\_SOURCE\_TYPE = '52014' AND

USER\_TYPE = '901195' THEN

RETURN('020103'); --外包服务

ELSIF PRODUCT\_ID = '901207' AND KEY\_SOURCE\_TYPE = '52014' AND

USER\_TYPE IN ('901186', '901187', '901188') THEN

RETURN('020104'); --系统集成

ELSIF PRODUCT\_ID IS NULL AND KEY\_SOURCE\_TYPE = 'IP' AND

(USER\_TYPE = '0' OR USER\_TYPE IS NULL) THEN

RETURN('020201'); --IP

ELSIF PRODUCT\_ID IN ('3022', '955232') AND KEY\_SOURCE\_TYPE = '52001' AND

USER\_TYPE IN ('142837', '143807') THEN

RETURN('020202'); --公话

ELSIF PRODUCT\_ID IN ('141974', '142027', '142763') AND

KEY\_SOURCE\_TYPE = '52004' AND

USER\_TYPE IN ('142157', '142158', '142160', '142284') THEN

RETURN('020203'); --家庭固话

ELSIF PRODUCT\_ID IN ('8', '9') AND KEY\_SOURCE\_TYPE = '52001' AND

USER\_TYPE IN ('141794',

'141795',

'141797',

'142647',

'142648',

'900412',

'900582') THEN

RETURN('020204'); --家庭模拟线

ELSIF PRODUCT\_ID IN ('901143', '955239') AND

KEY\_SOURCE\_TYPE IN ('52001', '52004') AND

USER\_TYPE IN ('142158', '142159', '142283', '900412') THEN

RETURN('020205'); --企业固话

ELSIF PRODUCT\_ID IN ('143263', '955237', '955238') AND

KEY\_SOURCE\_TYPE = '52001' AND

USER\_TYPE IN

('141794', '141795', '141797', '142647', '142648', '900412') THEN

RETURN('020206'); --企业模拟线

ELSIF PRODUCT\_ID IN ('900097', '955234', '955235', '955236') AND

KEY\_SOURCE\_TYPE = '52001' AND USER\_TYPE IN ('901037', '901038') THEN

RETURN('020207'); --数字中继线

ELSIF PRODUCT\_ID = '955391' AND KEY\_SOURCE\_TYPE = '52019' AND

USER\_TYPE = '142562' THEN

RETURN('020208'); --校园电话卡

ELSIF PRODUCT\_ID IS NULL AND KEY\_SOURCE\_TYPE = '165' AND

(USER\_TYPE = '0' OR USER\_TYPE IS NULL) THEN

RETURN('020301'); --165

ELSIF PRODUCT\_ID IN ('956101', '956102') AND KEY\_SOURCE\_TYPE = '52002' AND

USER\_TYPE = '142565' THEN

RETURN('020302'); --165ZX

ELSIF PRODUCT\_ID IS NULL AND KEY\_SOURCE\_TYPE = 'BST' AND

(USER\_TYPE = '0' OR USER\_TYPE IS NULL) THEN

RETURN('020303'); --BST

ELSIF (PRODUCT\_ID IN ('3078', '50963', '954973', '955392') OR

PRODUCT\_ID IS NULL) AND KEY\_SOURCE\_TYPE IN ('52006', '52019') AND

(USER\_TYPE IS NULL OR

USER\_TYPE IN

('141794', '141795', '142005', '142565', '142932', '900582')) THEN

RETURN('020304'); --CPN

ELSIF PRODUCT\_ID IN ('143668', '143669', '955243', '955244','DIA') AND KEY\_SOURCE\_TYPE = '52002' AND

(USER\_TYPE IS NULL OR

USER\_TYPE IN ('0', '142565', '143667', '900712')) THEN

RETURN('020305'); --DIA

ELSIF PRODUCT\_ID = '901104' AND KEY\_SOURCE\_TYPE = '52002' AND

USER\_TYPE = '142565' THEN

RETURN('020306'); --ISP大带宽

ELSIF PRODUCT\_ID = '900502' AND KEY\_SOURCE\_TYPE = '52013' AND

USER\_TYPE = '900582' THEN

RETURN('020307'); --亲情1+(固话+CPN)

ELSIF PRODUCT\_ID IN ('143678', '143679', '956107', '956108','IDC') AND KEY\_SOURCE\_TYPE = '52002' AND

( USER\_TYPE IN ('142565', '142566') or USER\_TYPE IS NULL) THEN

RETURN('020401'); --IDC

ELSIF PRODUCT\_ID = '955393' AND KEY\_SOURCE\_TYPE = '52019' AND

USER\_TYPE ='142511' THEN

RETURN('020402'); --WLAN卡

ELSIF PRODUCT\_ID IN ('142155', '900636', '901612') AND

KEY\_SOURCE\_TYPE IN ('52002', '52017') AND

USER\_TYPE IN ('141794', '142565') THEN

RETURN('020403'); --互联网媒体广告

ELSIF PRODUCT\_ID IN ('900038', '901227') AND

KEY\_SOURCE\_TYPE IN ('52002', '52017') AND

USER\_TYPE IN ('141794', '142565') THEN

RETURN('020404'); --其他

ELSIF PRODUCT\_ID = '955193' AND KEY\_SOURCE\_TYPE = '52017' AND

USER\_TYPE = '141794' THEN

RETURN('020405'); --网络信息安全

ELSIF PRODUCT\_ID = '142877' AND KEY\_SOURCE\_TYPE = '52001' AND

USER\_TYPE IN ('141794', '141795') THEN

RETURN('020406'); --网通新传真

ELSIF PRODUCT\_ID = '955248' AND KEY\_SOURCE\_TYPE = '52004' AND

USER\_TYPE = '141794' THEN

RETURN('020407'); --新会场

ELSIF PRODUCT\_ID = '956348' AND KEY\_SOURCE\_TYPE = '52002' AND

USER\_TYPE = '142565' THEN

RETURN('020408'); --云计算

ELSIF PRODUCT\_ID IN ('956103', '956104') AND KEY\_SOURCE\_TYPE = '52002' AND

USER\_TYPE = '142565' THEN

RETURN('020501'); --165ZX

ELSIF PRODUCT\_ID IN ('143064',

'143202',

'143202',

'143674',

'143675',

'901047',

'901049','数字电路','TeleHouse','以太网专线') AND

KEY\_SOURCE\_TYPE IN ('52002', '52002', '52010', '52002') AND

(USER\_TYPE IN ('141794', '142562', '142564', '142565', '143667') OR

USER\_TYPE IS NULL) THEN

RETURN('020502'); --DPLC

ELSIF PRODUCT\_ID IN ('143676', '143677','ATM','DDN','ATM端口') AND

KEY\_SOURCE\_TYPE = '52002' AND

(USER\_TYPE = '142565' OR USER\_TYPE IS NULL) THEN

RETURN('020503'); --IPLC

ELSIF PRODUCT\_ID IN ('901337', '901359','国际基础设施') AND

KEY\_SOURCE\_TYPE = '52002' AND

(USER\_TYPE = '142565' OR USER\_TYPE IS NULL) THEN

RETURN('020504'); --网元出租

ELSIF PRODUCT\_ID IN ('142608', '142609', '143829', '956185', '956409') AND

KEY\_SOURCE\_TYPE = '52007' AND USER\_TYPE IN ('141794', '901037') THEN

RETURN('020601'); --400/800

ELSIF PRODUCT\_ID IN ('955684', '956087') AND KEY\_SOURCE\_TYPE = '52018' AND

USER\_TYPE IN ('141794', '141795') THEN

RETURN('020602'); --超级总机

ELSIF PRODUCT\_ID IN ('142774', '956412') AND

KEY\_SOURCE\_TYPE IN ('52009', '52018') AND

(USER\_TYPE = '141794' OR USER\_TYPE IS NULL) THEN

RETURN('020603'); --其他

ELSIF PRODUCT\_ID = '955669' AND KEY\_SOURCE\_TYPE = '52004' AND

USER\_TYPE = '141794' THEN

RETURN('020604'); --企业固话

ELSIF PRODUCT\_ID = '954757' AND KEY\_SOURCE\_TYPE = '52018' AND

USER\_TYPE = '143667' THEN

RETURN('020605'); --融合类

ELSIF PRODUCT\_ID = '954954' AND KEY\_SOURCE\_TYPE = '52018' AND

USER\_TYPE = '141794' THEN

RETURN('020606'); --声讯类

ELSIF PRODUCT\_ID = '900536' AND KEY\_SOURCE\_TYPE = '52018' AND

USER\_TYPE = '141794' THEN

RETURN('020607'); --通信助理

ELSIF PRODUCT\_ID IN ('901613', '956190') AND KEY\_SOURCE\_TYPE = '52018' AND

USER\_TYPE = '141794' THEN

RETURN('020608'); --外包呼叫中心

ELSIF PRODUCT\_ID = '900296' AND KEY\_SOURCE\_TYPE = '52012' AND

(USER\_TYPE = '141794' OR USER\_TYPE IS NULL) THEN

RETURN('020609'); --行业信息

ELSIF PRODUCT\_ID IN ('142035', '955152') AND KEY\_SOURCE\_TYPE = '52004' AND

USER\_TYPE IN ('141794', '141795') THEN

RETURN('020610'); --一号通

ELSIF PRODUCT\_ID = '955327' AND KEY\_SOURCE\_TYPE = '52001' AND

USER\_TYPE = '141794' THEN

RETURN('020611'); --悦铃

ELSIF PRODUCT\_ID IN ('50981', '50985', '50986', '955241') AND

KEY\_SOURCE\_TYPE = '52003' AND

USER\_TYPE IN ('0', '141794', '141795') THEN

RETURN('020701'); --17969

ELSIF PRODUCT\_ID IS NULL AND KEY\_SOURCE\_TYPE = '193' AND

(USER\_TYPE = '0' OR USER\_TYPE IS NULL) THEN

RETURN('020702'); --193

ELSIF PRODUCT\_ID IN ('141782', '2754', '3027', '901165') AND

KEY\_SOURCE\_TYPE = '52005' AND

USER\_TYPE IN ('141794', '141795', '141796', '142511') THEN

RETURN('020703'); --196

ELSIF PRODUCT\_ID IN ('956105', '956106') AND KEY\_SOURCE\_TYPE = '52002' AND

USER\_TYPE = '142565' THEN

RETURN('020801'); --165ZX

ELSIF PRODUCT\_ID IN ('143670', '143671','FR','FR端口','FR端口') AND

KEY\_SOURCE\_TYPE = '52002' AND

(USER\_TYPE = '142565' OR USER\_TYPE IS NULL) THEN

RETURN('020802'); --FR VPN

ELSIF PRODUCT\_ID IS NULL AND KEY\_SOURCE\_TYPE = 'ZX' AND

(USER\_TYPE = '0' OR USER\_TYPE IS NULL) THEN

RETURN('020803'); --ZX

ELSIF PRODUCT\_ID IN ('143672', '143673','MPLS-VPN') AND

KEY\_SOURCE\_TYPE = '52002' AND

(USER\_TYPE IN ('142565', '143667') OR USER\_TYPE IS NULL) THEN

RETURN('020901'); --MPLS VPN

ELSIF PRODUCT\_ID = '900236' AND KEY\_SOURCE\_TYPE = '52011' AND

USER\_TYPE = '141794' THEN

RETURN('021001'); --橙卡

ELSIF PRODUCT\_ID IS NULL AND KEY\_SOURCE\_TYPE IS NULL AND

USER\_TYPE IS NULL THEN

RETURN('021101'); --信息传媒+卡

ELSE

RETURN('021102'); --其他

END IF;

END;

### Hive中处理思路及方法

#### 思路

通过if进行产品分类

#### 类名

org.apache.hadoop.hive.ql.udf.FuncProductTypeWs.java

#### Hive函数名

f\_product\_type\_ws

#### Hive自定义函数代码

**package** org.apache.hadoop.hive.ql.udf;

**import** org.apache.hadoop.hive.ql.exec.UDF;

/\*

\* 函数功能：产品分类函数

\* Oracle中创建者：王松

\* Hive自定义函数创建：平晓刚

\* 创建时间：2015-06-08

\* 对应原Oracle函数：DW.2.34 FUNC\_PRODUCT\_TYPE\_ID

\* 传入参数1：

\* 传入参数2：

\* 传入参数3：

\* 返回参数：

\* \*/

**public** **class** FuncProductTypeWs **extends** UDF {

**public** String evaluate(String key\_source\_type, String product\_id,

String user\_type) {

String output = **null**;

**if** (product\_id.equals("901272") && key\_source\_type.equals("52014")

&& (user\_type.equals("142565") | user\_type.equals("901191"))) {

output = "020101";// 备件与现场服务

} **else** **if** (product\_id.equals("901212")

&& key\_source\_type.equals("52014")

&& user\_type.equals("901191")) {

output = "020102";// 第三方软硬件销售

} **else** **if** (product\_id.equals("901208")

&& key\_source\_type.equals("52014")

&& user\_type.equals("901195")) {

output = "020103";// 外包服务

} **else** **if** (product\_id.equals("901207")

&& key\_source\_type.equals("52014")

&& (user\_type.equals("901186") | user\_type.equals("901187") | user\_type

.equals("901188"))) {

output = "020104";// 系统集成

} **else** **if** (product\_id.equals("") && key\_source\_type.equals("IP")

&& (user\_type.equals("0") | user\_type.equals(""))) {

output = "020201";// IP

} **else** **if** ((product\_id.equals("3022") | product\_id.equals("955232"))

&& key\_source\_type.equals("52001")

&& (user\_type.equals("142837") | user\_type.equals("143807"))) {

output = "020202";// 公话

} **else** **if** ((product\_id.equals("141974") | product\_id.equals("142027") | product\_id

.equals("142763"))

&& key\_source\_type.equals("52004")

&& (user\_type.equals("142157") | user\_type.equals("142158")

| user\_type.equals("142160") | user\_type

.equals("142284"))) {

output = "020203";// 家庭固话

} **else** **if** ((product\_id.equals("8") | product\_id.equals("9"))

&& key\_source\_type.equals("52001")

&& (user\_type.equals("141794") | user\_type.equals("141795")

| user\_type.equals("141797")

| user\_type.equals("142647")

| user\_type.equals("142648")

| user\_type.equals("900412") | user\_type

.equals("900582"))) {

output = "020204";// 家庭模拟线

} **else** **if** ((product\_id.equals("901143") | product\_id.equals("955239"))

&& (key\_source\_type.equals("52001") | key\_source\_type

.equals("52004"))

&& (user\_type.equals("142158") | user\_type.equals("142159")

| user\_type.equals("142283") | user\_type

.equals("900412"))) {

output = "020205";// 企业固话

} **else** **if** ((product\_id.equals("143263") | product\_id.equals("955237") | product\_id

.equals("955238"))

&& (key\_source\_type.equals("52001"))

&& (user\_type.equals("141794") | user\_type.equals("141795")

| user\_type.equals("141797")

| user\_type.equals("142647")

| user\_type.equals("142648") | user\_type

.equals("900412"))) {

output = "020206";// 企业模拟线

} **else** **if** ((product\_id.equals("900097") | product\_id.equals("955234")

| product\_id.equals("955235") | product\_id.equals("955236"))

&& (key\_source\_type.equals("52001"))

&& (user\_type.equals("901037") | user\_type.equals("901038"))) {

output = "020207";// 数字中继线

} **else** **if** ((product\_id.equals("955391"))

&& (key\_source\_type.equals("52019"))

&& (user\_type.equals("142562"))) {

output = "020208";// 校园电话卡

} **else** **if** ((product\_id.equals("")) && (key\_source\_type.equals("165"))

&& (user\_type.equals("0") | user\_type.equals(""))) {

output = "020301";// 165

} **else** **if** ((product\_id.equals("956101") | product\_id.equals("956102"))

&& (key\_source\_type.equals("52002"))

&& (user\_type.equals("142565"))) {

output = "020302";// 165ZX

} **else** **if** ((product\_id.equals("")) && (key\_source\_type.equals("BST"))

&& (user\_type.equals("0") | user\_type.equals(""))) {

output = "020303";// BST

} **else** **if** ((product\_id.equals("3078") | product\_id.equals("50963")

| product\_id.equals("954973") | product\_id.equals("955392") | product\_id

.equals(""))

&& (key\_source\_type.equals("52006") | key\_source\_type

.equals("52019"))

&& (user\_type.equals("") | user\_type.equals("141794")

| user\_type.equals("141795")

| user\_type.equals("142005")

| user\_type.equals("142565")

| user\_type.equals("142932") | user\_type

.equals("900582"))) {

output = "020304";// CPN

} **else** **if** ((product\_id.equals("143668") | product\_id.equals("143669")

| product\_id.equals("955243") | product\_id.equals("955244") | product\_id

.equals("DIA"))

&& (key\_source\_type.equals("52002"))

&& (user\_type.equals("0") | user\_type.equals("142565")

| user\_type.equals("143667")

| user\_type.equals("900712") | user\_type.equals(""))) {

output = "020305";// DIA

} **else** **if** ((product\_id.equals("901104"))

&& (key\_source\_type.equals("52002"))

&& (user\_type.equals("142565"))) {

output = "020306";// ISP大带宽

} **else** **if** ((product\_id.equals("900502"))

&& (key\_source\_type.equals("52013"))

&& (user\_type.equals("900582"))) {

output = "020307";// 亲情1+(固话+CPN)

} **else** **if** ((product\_id.equals("143678") | product\_id.equals("143679")

| product\_id.equals("956107") | product\_id.equals("956108") | product\_id

.equals("IDC"))

&& (key\_source\_type.equals("52002"))

&& (user\_type.equals("142565") | user\_type.equals("142566") | user\_type

.equals(""))) {

output = "020401";// IDC

} **else** **if** ((product\_id.equals("955393"))

&& (key\_source\_type.equals("52019"))

&& (user\_type.equals("142511"))) {

output = "020402";// WLAN卡

} **else** **if** ((product\_id.equals("142155") | product\_id.equals("900636") | product\_id

.equals("901612"))

&& (key\_source\_type.equals("52002") | key\_source\_type

.equals("52017"))

&& (user\_type.equals("141794") | user\_type.equals("142565"))) {

output = "020403";// 互联网媒体广告

} **else** **if** ((product\_id.equals("900038") | product\_id.equals("901227"))

&& (key\_source\_type.equals("52002") | key\_source\_type

.equals("52017"))

&& (user\_type.equals("141794") | user\_type.equals("142565"))) {

output = "020404";// 其他

} **else** **if** ((product\_id.equals("955193"))

&& (key\_source\_type.equals("52017"))

&& (user\_type.equals("141794"))) {

output = "020405";// 网络信息安全

} **else** **if** ((product\_id.equals("142877"))

&& (key\_source\_type.equals("52001"))

&& (user\_type.equals("141794") | user\_type.equals("141795"))) {

output = "020406";// 网通新传真

} **else** **if** ((product\_id.equals("955248"))

&& (key\_source\_type.equals("52004"))

&& (user\_type.equals("141794"))) {

output = "020407";// 新会场

} **else** **if** ((product\_id.equals("956348"))

&& (key\_source\_type.equals("52002"))

&& (user\_type.equals("142565"))) {

output = "020408";// 云计算

} **else** **if** ((product\_id.equals("956103") | product\_id.equals("956104"))

&& (key\_source\_type.equals("52002"))

&& (user\_type.equals("142565"))) {

output = "020501";// 165ZX

} **else** **if** ((product\_id.equals("143064") | product\_id.equals("143202")

| product\_id.equals("143202") | product\_id.equals("143674")

| product\_id.equals("143675") | product\_id.equals("901047")

| product\_id.equals("901049") | product\_id.equals("数字电路")

| product\_id.equals("TeleHouse") | product\_id.equals("以太网专线"))

&& (key\_source\_type.equals("52002")

| key\_source\_type.equals("52002")

| key\_source\_type.equals("52010") | key\_source\_type

.equals("52002"))

&& (user\_type.equals("141794") | user\_type.equals("142562")

| user\_type.equals("142564")

| user\_type.equals("142565")

| user\_type.equals("143667") | user\_type.equals(""))) {

output = "020502";// DPLC

} **else** **if** ((product\_id.equals("143676") | product\_id.equals("143677")

| product\_id.equals("ATM") | product\_id.equals("DDN") | product\_id

.equals("ATM端口"))

&& (key\_source\_type.equals("52002"))

&& (user\_type.equals("142565") | user\_type.equals(""))) {

output = "020503";// IPLC

} **else** **if** ((product\_id.equals("901337") | product\_id.equals("901359") | product\_id

.equals("国际基础设施"))

&& (key\_source\_type.equals("52002"))

&& (user\_type.equals("142565") | user\_type.equals(""))) {

output = "020504";// 网元出租

} **else** **if** ((product\_id.equals("142608") | product\_id.equals("142609")

| product\_id.equals("143829") | product\_id.equals("956185") | product\_id

.equals("956409"))

&& (key\_source\_type.equals("52007"))

&& (user\_type.equals("141794") | user\_type.equals("901037"))) {

output = "020601";// 400/800

} **else** **if** ((product\_id.equals("955684") | product\_id.equals("956087"))

&& (key\_source\_type.equals("52018"))

&& (user\_type.equals("141794") | user\_type.equals("141795"))) {

output = "020602";// 超级总机

} **else** **if** ((product\_id.equals("142774") | product\_id.equals("956412"))

&& (key\_source\_type.equals("52009") | key\_source\_type

.equals("52018"))

&& (user\_type.equals("141794") | user\_type.equals(""))) {

output = "020603";// 其他

} **else** **if** ((product\_id.equals("955669"))

&& (key\_source\_type.equals("52004"))

&& (user\_type.equals("141794"))) {

output = "020604";// 企业固话

} **else** **if** ((product\_id.equals("954757"))

&& (key\_source\_type.equals("52018"))

&& (user\_type.equals("143667"))) {

output = "020605";// 融合类

} **else** **if** ((product\_id.equals("954954"))

&& (key\_source\_type.equals("52018"))

&& (user\_type.equals("141794"))) {

output = "020606";// 声讯类

} **else** **if** ((product\_id.equals("900536"))

&& (key\_source\_type.equals("52018"))

&& (user\_type.equals("141794"))) {

output = "020607";// 通信助理

} **else** **if** ((product\_id.equals("901613") | product\_id.equals("956190"))

&& (key\_source\_type.equals("52018"))

&& (user\_type.equals("141794"))) {

output = "020608";// 外包呼叫中心

} **else** **if** ((product\_id.equals("900296"))

&& (key\_source\_type.equals("52012"))

&& (user\_type.equals("141794") | user\_type.equals(""))) {

output = "020609";// 行业信息

} **else** **if** ((product\_id.equals("142035") | product\_id.equals("955152"))

&& (key\_source\_type.equals("52004"))

&& (user\_type.equals("141794") | user\_type.equals("141795"))) {

output = "020610";// 一号通

} **else** **if** (product\_id.equals("955327")

&& key\_source\_type.equals("52001")

&& user\_type.equals("141794")) {

output = "020611";// 悦铃

} **else** **if** ((product\_id.equals("50981") | product\_id.equals("50985")

| product\_id.equals("50986") | product\_id.equals("955241"))

&& key\_source\_type.equals("52003")

&& (user\_type.equals("0") | user\_type.equals("141794") | user\_type

.equals("141795"))) {

output = "020701";// 17969

} **else** **if** ((product\_id.equals("")) && key\_source\_type.equals("193")

&& (user\_type.equals("0") | user\_type.equals(""))) {

output = "020702";// 193

} **else** **if** ((product\_id.equals("141782") | product\_id.equals("2754")

| product\_id.equals("3027") | product\_id.equals("901165"))

&& key\_source\_type.equals("52005")

&& (user\_type.equals("141794") | user\_type.equals("141795")

| user\_type.equals("141796") | user\_type

.equals("142511"))) {

output = "020703";// 196

} **else** **if** ((product\_id.equals("956105") | product\_id.equals("956106"))

&& key\_source\_type.equals("52002")

&& (user\_type.equals("142565"))) {

output = "020801";// 165ZX

} **else** **if** ((product\_id.equals("143670") | product\_id.equals("143671")

| product\_id.equals("FR") | product\_id.equals("FR端口") | product\_id

.equals("FR端口"))

&& key\_source\_type.equals("52002")

&& (user\_type.equals("142565") | user\_type.equals(""))) {

output = "020802";// FR VPN

} **else** **if** ((product\_id.equals("")) && key\_source\_type.equals("ZX")

&& (user\_type.equals("0") | user\_type.equals(""))) {

output = "020803";// ZX

} **else** **if** ((product\_id.equals("143672") | product\_id.equals("143673") | product\_id

.equals("MPLS-VPN"))

&& key\_source\_type.equals("52002")

&& (user\_type.equals("142565") | user\_type.equals("143667") | user\_type

.equals(""))) {

output = "020901";// MPLS VPN

} **else** **if** ((product\_id.equals("900236"))

&& key\_source\_type.equals("52011")

&& (user\_type.equals("141794"))) {

output = "021001";// 橙卡

} **else** **if** ((product\_id.equals("")) && key\_source\_type.equals("")

&& (user\_type.equals(""))) {

output = "021101";// 信息传媒+卡

} **else** {

output = "021102";// 其他

}

**return** output;

}

}

## （已完成）FUNC\_INNET\_DURATION

### Oracle数据库中对应OBJECT\_ID

350378

### 作用描述

在网时长分档函数

### 函数体内容

FUNCTION FUNC\_INNET\_DURATION(INNET\_DURATION NUMBER)

/\*-----------------------------------------------------------------------

修改内容：上网时长分档

修 改 人：东方国信 孟令利

修改时间：20111022

------------------------------------------------------------------------\*/

RETURN VARCHAR2 IS

BEGIN

IF INNET\_DURATION IS NULL THEN

RETURN('00');

ELSIF INNET\_DURATION IN (0, 1) THEN

RETURN('01');

ELSIF INNET\_DURATION = 2 THEN

RETURN('02');

ELSIF INNET\_DURATION = 3 THEN

RETURN('03');

ELSIF INNET\_DURATION = 4 THEN

RETURN('04');

ELSIF INNET\_DURATION = 5 THEN

RETURN('05');

ELSIF INNET\_DURATION = 6 THEN

RETURN('06');

ELSIF INNET\_DURATION = 7 THEN

RETURN('07');

ELSIF INNET\_DURATION = 8 THEN

RETURN('08');

ELSIF INNET\_DURATION = 9 THEN

RETURN('09');

ELSIF INNET\_DURATION = 10 THEN

RETURN('10');

ELSIF INNET\_DURATION = 11 THEN

RETURN('11');

ELSIF INNET\_DURATION = 12 THEN

RETURN('12');

ELSIF INNET\_DURATION = 13 THEN

RETURN('13');

ELSIF INNET\_DURATION = 14 THEN

RETURN('14');

ELSIF INNET\_DURATION = 15 THEN

RETURN('15');

ELSIF INNET\_DURATION = 16 THEN

RETURN('16');

ELSIF INNET\_DURATION = 17 THEN

RETURN('17');

ELSIF INNET\_DURATION = 18 THEN

RETURN('18');

ELSIF INNET\_DURATION = 19 THEN

RETURN('19');

ELSIF INNET\_DURATION = 20 THEN

RETURN('20');

ELSIF INNET\_DURATION = 21 THEN

RETURN('21');

ELSIF INNET\_DURATION = 22 THEN

RETURN('22');

ELSIF INNET\_DURATION = 23 THEN

RETURN('23');

ELSIF INNET\_DURATION = 24 THEN

RETURN('24');

ELSIF (INNET\_DURATION > 24 AND INNET\_DURATION <= 36) THEN

RETURN('25');

ELSIF (INNET\_DURATION > 36 AND INNET\_DURATION <= 48) THEN

RETURN('26');

ELSIF (INNET\_DURATION > 48 AND INNET\_DURATION <= 60) THEN

RETURN('27');

ELSIF (INNET\_DURATION > 60 AND INNET\_DURATION <= 72) THEN

RETURN('28');

ELSIF (INNET\_DURATION > 72 AND INNET\_DURATION <= 84) THEN

RETURN('29');

ELSIF (INNET\_DURATION > 84 AND INNET\_DURATION <= 96) THEN

RETURN('30');

ELSE

RETURN('31');

END IF;

END;

### Hive中处理思路及方法

#### 思路

通过IF判断，进行在网时长分档

#### 类名

org.apache.hadoop.hive.ql.udf.FuncInnetDuration.java

#### Hive函数名

f\_innet\_duration

#### Hive自定义函数代码

**package** org.apache.hadoop.hive.ql.udf;

**import** org.apache.hadoop.hive.ql.exec.UDF;

/\*

\* 函数功能：在网时长分档函数

\* Oracle中创建者：孟令利

\* Hive自定义函数创建：平晓刚

\* 创建时间：2015-06-08

\* 对应原Oracle函数：DW.FUNC\_INNET\_DURATION

\* 传入参数1：在网时长

\* 返回参数：在网时长分档

\* \*/

**public** **class** FuncInnetDuration **extends** UDF {

**public** String evaluate(**int** innet\_duration) {

String output = **null**;

**if** (String.*valueOf*(innet\_duration).equals("")) {

output = "00";

} **else** **if** (innet\_duration == 0 | innet\_duration == 1) {

output = "01";

} **else** **if** (innet\_duration == 2) {

output = "02";

} **else** **if** (innet\_duration == 3) {

output = "03";

} **else** **if** (innet\_duration == 4) {

output = "04";

} **else** **if** (innet\_duration == 5) {

output = "05";

} **else** **if** (innet\_duration == 6) {

output = "06";

} **else** **if** (innet\_duration == 7) {

output = "07";

} **else** **if** (innet\_duration == 8) {

output = "08";

} **else** **if** (innet\_duration == 9) {

output = "09";

} **else** **if** (innet\_duration == 10) {

output = "10";

} **else** **if** (innet\_duration == 11) {

output = "11";

} **else** **if** (innet\_duration == 12) {

output = "12";

} **else** **if** (innet\_duration == 13) {

output = "13";

} **else** **if** (innet\_duration == 14) {

output = "14";

} **else** **if** (innet\_duration == 15) {

output = "15";

} **else** **if** (innet\_duration == 16) {

output = "16";

} **else** **if** (innet\_duration == 17) {

output = "17";

} **else** **if** (innet\_duration == 18) {

output = "18";

} **else** **if** (innet\_duration == 19) {

output = "19";

} **else** **if** (innet\_duration == 20) {

output = "20";

} **else** **if** (innet\_duration == 21) {

output = "21";

} **else** **if** (innet\_duration == 22) {

output = "22";

} **else** **if** (innet\_duration == 23) {

output = "23";

} **else** **if** (innet\_duration == 24) {

output = "24";

} **else** **if** (innet\_duration > 24 && innet\_duration <= 36) {

output = "25";

} **else** **if** (innet\_duration > 36 && innet\_duration <= 48) {

output = "26";

} **else** **if** (innet\_duration > 48 && innet\_duration <= 60) {

output = "27";

} **else** **if** (innet\_duration > 60 && innet\_duration <= 72) {

output = "28";

} **else** **if** (innet\_duration > 72 && innet\_duration <= 84) {

output = "29";

} **else** **if** (innet\_duration > 84 && innet\_duration <= 96) {

output = "30";

} **else** {

output = "31";

}

**return** output;

}

}

## （已完成）FUNC\_OWE\_FEE\_LEVEL

### Oracle数据库中对应OBJECT\_ID

350379

### 作用描述

欠费分档函数

### 函数体内容

FUNCTION FUNC\_OWE\_FEE\_LEVEL(

INCOME\_FEE NUMBER)

RETURN VARCHAR2

IS

/\*------------------------------------------------------------------------------------

过 程 名 : 出帐费用分档

生成时间 ： 2009.12.26

编 写 人 ： 平晓刚

注意事项 ：

------------------------------------------------------------------------------------\*/

BEGIN

IF INCOME\_FEE <= 50 THEN RETURN '1';

ELSIF INCOME\_FEE > 50 AND INCOME\_FEE <= 100 THEN RETURN '2';

ELSIF INCOME\_FEE > 100 AND INCOME\_FEE <= 150 THEN RETURN '3';

ELSIF INCOME\_FEE > 150 AND INCOME\_FEE <= 300 THEN RETURN '4';

ELSIF INCOME\_FEE > 300 AND INCOME\_FEE <= 500 THEN RETURN '5';

ELSIF INCOME\_FEE > 500 AND INCOME\_FEE <= 1000 THEN RETURN '6';

ELSIF INCOME\_FEE > 1000 THEN RETURN '7';

END IF;

END;

### Hive中处理思路及方法

#### 思路

通过If判断进行欠费分档

#### 类名

org.apache.hadoop.hive.ql.udf.FuncOweFeeLevel.java

#### Hive函数名

f\_owe\_fee\_level

#### Hive自定义函数代码

**package** org.apache.hadoop.hive.ql.udf;

**import** org.apache.hadoop.hive.ql.exec.UDF;

/\*

\* 函数功能：欠费分档函数

\* Oracle中创建者：平晓刚

\* Hive自定义函数创建：平晓刚

\* 创建时间：2015-06-08

\* 对应原Oracle函数：DW.FUNC\_OWE\_FEE\_LEVEL

\* 传入参数1：欠费

\* 返回参数：欠费分档

\* \*/

**public** **class** FuncOweFeeLevel **extends** UDF {

**public** String evaluate(**double** income\_fee) {

String output = **null**;

**if** (income\_fee <= 50) {

output = "1";

} **else** **if** (income\_fee > 50 && income\_fee <= 100) {

output = "2";

} **else** **if** (income\_fee > 100 && income\_fee <= 150) {

output = "3";

} **else** **if** (income\_fee > 150 && income\_fee <= 300) {

output = "4";

} **else** **if** (income\_fee > 300 && income\_fee <= 500) {

output = "5";

} **else** **if** (income\_fee > 500 && income\_fee <= 1000) {

output = "6";

} **else** **if** (income\_fee > 1000) {

output = "7";

}

**return** output;

}

}

## （已完成）FUNC\_INCOME\_LEVEL

### Oracle数据库中对应OBJECT\_ID

350376

### 作用描述

### 函数体内容

FUNCTION FUNC\_INCOME\_LEVEL(key\_source\_type\_cd NUMBER,

product\_id number,

user\_type\_id number)

/\*-----------------------------------------------------------------------

修改内容：收入分档

修 改 人：东方国信 LUZHE

修改时间：20061130

------------------------------------------------------------------------\*/

RETURN VARCHAR2 IS

BEGIN

IF key\_source\_type\_cd = 52007 THEN

RETURN('800,4006');

ELSIF key\_source\_type\_cd = 52003 THEN

RETURN('17969主叫直拨');

ELSIF key\_source\_type\_cd = 52005 THEN

RETURN('196');

ELSIF key\_source\_type\_cd = 52006 THEN

RETURN('CPN业务');

ELSIF key\_source\_type\_cd = 52004 and product\_id not in (142035, 142848) THEN

RETURN('宽带电话(NGN)');

ELSIF key\_source\_type\_cd = 52004 and product\_id = 142848 THEN

RETURN('空中会议室');

ELSIF key\_source\_type\_cd = 52004 and product\_id = 142035 THEN

RETURN('一号通后付');

ELSIF key\_source\_type\_cd = 52001 and product\_id not in (142877) and

user\_type\_id not in (143807, 900130, 900150) THEN

RETURN('语音业务');

ELSIF key\_source\_type\_cd = 52001 and product\_id = 142877 THEN

RETURN('网通新传真');

ELSIF key\_source\_type\_cd = 52001 and user\_type\_id in (143807, 900150) THEN

RETURN('公话');

ELSIF key\_source\_type\_cd = 52001 and user\_type\_id = 900130 THEN

RETURN('话批');

ELSIF key\_source\_type\_cd = 52002 and

product\_id in (143668, 143669, 955243, 955244) THEN

RETURN('DIA');

ELSIF key\_source\_type\_cd = 52002 and product\_id in (143678, 143679) THEN

RETURN('IDC');

ELSIF key\_source\_type\_cd = 52002 and product\_id in (143674, 143675) THEN

RETURN('DPLC');

ELSIF key\_source\_type\_cd = 52002 and product\_id in (143676, 143677) THEN

RETURN('IPLC');

ELSIF key\_source\_type\_cd = 52002 and product\_id in (143672, 143673) THEN

RETURN('MPLS VPN');

ELSIF key\_source\_type\_cd = 52002 and product\_id in (143670, 143671) THEN

RETURN('FR VPN');

ELSIF key\_source\_type\_cd = 52002 and product\_id in (142149, 142126) THEN

RETURN('WLAN');

ELSIF key\_source\_type\_cd = 52002 and

product\_id in (143607, 901359, 901337, 143604) THEN

RETURN('网元出租');

ELSIF key\_source\_type\_cd = 52002 and product\_id in (901104) THEN

RETURN('互联网');

ELSIF key\_source\_type\_cd = 52002 and

product\_id in

(956101, 956102, 956103, 956104, 956105, 956106, 956107, 956108) THEN

RETURN('165ZX');

ELSIF key\_source\_type\_cd = 52014 THEN

RETURN('ICT');

else

return('语音业务1');

END IF;

END;

### Hive中处理思路及方法

#### 思路

通过IF判断，进行函数改写

#### 类名

org.apache.hadoop.hive.ql.udf.FuncIncomeLevel.java

#### Hive函数名

f\_income\_level\_func

#### Hive自定义函数代码

**package** org.apache.hadoop.hive.ql.udf;

**import** java.util.ArrayList;

**import** java.util.List;

**import** org.apache.hadoop.hive.ql.exec.UDF;

/\*

\* 函数功能：

\* Oracle中创建者：

\* Hive自定义函数创建：平晓刚

\* 创建时间：2015-06-08

\* 对应原Oracle函数：DW.FUNC\_OWE\_FEE\_LEVEL

\* 传入参数1：

\* 返回参数：

\* \*/

**public** **class** FuncIncomeLevel **extends** UDF {

**public** String evaluate(String key\_source\_type\_cd, String product\_id,

String user\_type\_id) {

String output = **null**;

List key\_source\_52004\_product = **new** ArrayList();

key\_source\_52004\_product.add("142035");

key\_source\_52004\_product.add("142848");

List key\_source\_52001\_user\_type = **new** ArrayList();

key\_source\_52001\_user\_type.add("143807");

key\_source\_52001\_user\_type.add("900130");

key\_source\_52001\_user\_type.add("900150");

**if** (key\_source\_type\_cd.equals("52007")) {

output = "800,4006";

} **else** **if** (key\_source\_type\_cd.equals("52003")) {

output = "17969主叫直拨";

} **else** **if** (key\_source\_type\_cd.equals("52005")) {

output = "196";

} **else** **if** (key\_source\_type\_cd.equals("52006")) {

output = "CPN业务";

} **else** **if** (key\_source\_type\_cd.equals("52004")

&& !key\_source\_52004\_product.contains(product\_id)) {

output = "宽带电话(NGN)";

}**else** **if** (key\_source\_type\_cd.equals("52004")

&& product\_id.equals("142848")) {

output = "空中会议室";

}**else** **if** (key\_source\_type\_cd.equals("52004")

&& product\_id.equals("142035")) {

output = "一号通后付";

}**else** **if** (key\_source\_type\_cd.equals("52001")

&& !product\_id.equals("142877") && !key\_source\_52001\_user\_type.contains(user\_type\_id)) {

output = "语音业务";

}**else** **if** (key\_source\_type\_cd.equals("52001")

&& product\_id.equals("142877") ) {

output = "网通新传真";

}**else** **if** (key\_source\_type\_cd.equals("52001")

&& (user\_type\_id.equals("143807") | user\_type\_id.equals("900150")) ) {

output = "公话";

}**else** **if** (key\_source\_type\_cd.equals("52001")

&& (user\_type\_id.equals("900130")) ) {

output = "话批";

}**else** **if** (key\_source\_type\_cd.equals("52002")

&& (product\_id.equals("143668")

|product\_id.equals("143669")

|product\_id.equals("955243")

|product\_id.equals("955244")) ) {

output = "DIA";

}**else** **if** (key\_source\_type\_cd.equals("52002")

&& (product\_id.equals("143678")

|product\_id.equals("143679")) ) {

output = "IDC";

}**else** **if** (key\_source\_type\_cd.equals("52002")

&& (product\_id.equals("143674")

|product\_id.equals("143675")) ) {

output = "DPLC";

}**else** **if** (key\_source\_type\_cd.equals("52002")

&& (product\_id.equals("143676")

|product\_id.equals("143677")) ) {

output = "IPLC";

}**else** **if** (key\_source\_type\_cd.equals("52002")

&& (product\_id.equals("143672")

|product\_id.equals("143673")) ) {

output = "MPLS VPN";

}**else** **if** (key\_source\_type\_cd.equals("52002")

&& (product\_id.equals("143670")

|product\_id.equals("143671")) ) {

output = "FR VPN";

}**else** **if** (key\_source\_type\_cd.equals("52002")

&& (product\_id.equals("142149")

|product\_id.equals("142126")) ) {

output = "WLAN";

}**else** **if** (key\_source\_type\_cd.equals("52002")

&& (product\_id.equals("143607")

|product\_id.equals("901359")

|product\_id.equals("901337")

|product\_id.equals("143604")) ) {

output = "网元出租";

}**else** **if** (key\_source\_type\_cd.equals("52002")

&& (product\_id.equals("901104")) ) {

output = "互联网";

}**else** **if** (key\_source\_type\_cd.equals("52002")

&& (product\_id.equals("956101")

|product\_id.equals("956102")

|product\_id.equals("956103")

|product\_id.equals("956104")

|product\_id.equals("956105")

|product\_id.equals("956106")

|product\_id.equals("956107")

|product\_id.equals("956108")) ) {

output = "165ZX";

}**else** **if** (key\_source\_type\_cd.equals("52014")) {

output = "ICT";

}**else**{

output = "语音业务1";

}

**return** output;

}

}

## （已完成）FUNC\_CHARGE\_LEVEL

### Oracle数据库中对应OBJECT\_ID

350372

### 作用描述

收入分档函数

### 函数体内容

FUNCTION FUNC\_CHARGE\_LEVEL(ALL\_FEE number)

/\*-----------------------------------------------------------------------

修改内容：为适应上报V2.0细分1000-2000的分档为1000-1500、1500-2000

修 改 人：东方国信 赵鹏

修改时间：20061130

------------------------------------------------------------------------\*/

RETURN VARCHAR2 IS

BEGIN

IF ALL\_FEE <= 0 THEN RETURN ('00');

ELSIF ALL\_FEE > 0 AND ALL\_FEE <= 10 THEN RETURN ('01');

ELSIF ALL\_FEE > 10 AND ALL\_FEE <= 20 THEN RETURN ('02');

ELSIF ALL\_FEE > 20 AND ALL\_FEE <= 30 THEN RETURN ('03');

ELSIF ALL\_FEE > 30 AND ALL\_FEE <= 40 THEN RETURN ('04');

ELSIF ALL\_FEE > 40 AND ALL\_FEE <= 50 THEN RETURN ('05');

ELSIF ALL\_FEE > 50 AND ALL\_FEE <= 60 THEN RETURN ('06');

ELSIF ALL\_FEE > 60 AND ALL\_FEE <= 70 THEN RETURN ('07');

ELSIF ALL\_FEE > 70 AND ALL\_FEE <= 80 THEN RETURN ('08');

ELSIF ALL\_FEE > 80 AND ALL\_FEE <= 90 THEN RETURN ('09');

ELSIF ALL\_FEE > 90 AND ALL\_FEE <= 100 THEN RETURN ('10');

ELSIF ALL\_FEE > 100 AND ALL\_FEE <= 110 THEN RETURN ('11');

ELSIF ALL\_FEE > 110 AND ALL\_FEE <= 120 THEN RETURN ('12');

ELSIF ALL\_FEE > 120 AND ALL\_FEE <= 130 THEN RETURN ('13');

ELSIF ALL\_FEE > 130 AND ALL\_FEE <= 140 THEN RETURN ('14');

ELSIF ALL\_FEE > 140 AND ALL\_FEE <= 150 THEN RETURN ('15');

ELSIF ALL\_FEE > 150 AND ALL\_FEE <= 160 THEN RETURN ('16');

ELSIF ALL\_FEE > 160 AND ALL\_FEE <= 170 THEN RETURN ('17');

ELSIF ALL\_FEE > 170 AND ALL\_FEE <= 180 THEN RETURN ('18');

ELSIF ALL\_FEE > 180 AND ALL\_FEE <= 190 THEN RETURN ('19');

ELSIF ALL\_FEE > 190 AND ALL\_FEE <= 200 THEN RETURN ('20');

ELSIF ALL\_FEE > 200 AND ALL\_FEE <= 250 THEN RETURN ('25');

ELSIF ALL\_FEE > 250 AND ALL\_FEE <= 300 THEN RETURN ('30');

ELSIF ALL\_FEE > 300 AND ALL\_FEE <= 350 THEN RETURN ('35');

ELSIF ALL\_FEE > 350 AND ALL\_FEE <= 400 THEN RETURN ('40');

ELSIF ALL\_FEE > 400 AND ALL\_FEE <= 450 THEN RETURN ('45');

ELSIF ALL\_FEE > 450 AND ALL\_FEE <= 500 THEN RETURN ('50');

ELSIF ALL\_FEE > 500 AND ALL\_FEE <= 800 THEN RETURN ('80');

ELSIF ALL\_FEE > 800 AND ALL\_FEE <= 1000 THEN RETURN ('A0');

-- ELSIF ALL\_FEE > 1000 AND ALL\_FEE <= 2000 THEN RETURN ('A1');

ELSIF ALL\_FEE > 1000 AND ALL\_FEE <= 1500 THEN RETURN ('A2');

ELSIF ALL\_FEE > 1500 AND ALL\_FEE <= 2000 THEN RETURN ('A3');

ELSIF ALL\_FEE > 2000 THEN RETURN ('AF');

ELSE RETURN('FF');

END IF;

END;

### Hive中处理思路及方法

#### 思路

通过If判断进行收入分档

#### 类名

org.apache.hadoop.hive.ql.udf. FuncChargeLevel.java

#### Hive函数名

f\_charge\_level

#### Hive自定义函数代码

**package** org.apache.hadoop.hive.ql.udf;

**import** org.apache.hadoop.hive.ql.exec.UDF;

/\*

\* 函数功能：收入分档函数

\* Oracle中创建者：赵鹏

\* Hive自定义函数创建：平晓刚

\* 创建时间：2015-06-08

\* 对应原Oracle函数：DW.FUNC\_CHARGE\_LEVEL

\* 传入参数1：收入

\* 返回参数：收入分档

\* \*/

**public** **class** FuncChargeLevel **extends** UDF {

**public** String evaluate(**double** charge\_fee) {

String output = **null**;

**if** (charge\_fee <= 0) {

output = "00";

} **else** **if** (charge\_fee > 0 && charge\_fee <= 10) {

output = "01";

} **else** **if** (charge\_fee > 10 && charge\_fee <= 20) {

output = "02";

} **else** **if** (charge\_fee > 20 && charge\_fee <= 30) {

output = "03";

} **else** **if** (charge\_fee > 30 && charge\_fee <= 40) {

output = "04";

} **else** **if** (charge\_fee > 40 && charge\_fee <= 50) {

output = "05";

} **else** **if** (charge\_fee > 50 && charge\_fee <= 60) {

output = "06";

} **else** **if** (charge\_fee > 60 && charge\_fee <= 70) {

output = "07";

} **else** **if** (charge\_fee > 70 && charge\_fee <= 80) {

output = "08";

} **else** **if** (charge\_fee > 80 && charge\_fee <= 90) {

output = "09";

} **else** **if** (charge\_fee > 90 && charge\_fee <= 100) {

output = "10";

} **else** **if** (charge\_fee > 100 && charge\_fee <= 110) {

output = "11";

} **else** **if** (charge\_fee > 110 && charge\_fee <= 120) {

output = "12";

} **else** **if** (charge\_fee > 120 && charge\_fee <= 130) {

output = "13";

} **else** **if** (charge\_fee > 130 && charge\_fee <= 140) {

output = "14";

} **else** **if** (charge\_fee > 140 && charge\_fee <= 150) {

output = "15";

} **else** **if** (charge\_fee > 150 && charge\_fee <= 160) {

output = "16";

} **else** **if** (charge\_fee > 160 && charge\_fee <= 170) {

output = "17";

} **else** **if** (charge\_fee > 170 && charge\_fee <= 180) {

output = "18";

} **else** **if** (charge\_fee > 180 && charge\_fee <= 190) {

output = "19";

} **else** **if** (charge\_fee > 190 && charge\_fee <= 200) {

output = "20";

} **else** **if** (charge\_fee > 200 && charge\_fee <= 250) {

output = "25";

} **else** **if** (charge\_fee > 250 && charge\_fee <= 300) {

output = "30";

} **else** **if** (charge\_fee > 300 && charge\_fee <= 350) {

output = "35";

} **else** **if** (charge\_fee > 350 && charge\_fee <= 400) {

output = "40";

} **else** **if** (charge\_fee > 400 && charge\_fee <= 450) {

output = "45";

} **else** **if** (charge\_fee > 450 && charge\_fee <= 500) {

output = "50";

} **else** **if** (charge\_fee > 500 && charge\_fee <= 800) {

output = "80";

} **else** **if** (charge\_fee > 800 && charge\_fee <= 1000) {

output = "A0";

} **else** **if** (charge\_fee > 1000 && charge\_fee <= 1500) {

output = "A2";

} **else** **if** (charge\_fee > 1500 && charge\_fee <= 2000) {

output = "A3";

} **else** **if** (charge\_fee > 2000) {

output = "AF";

} **else** {

output = "FF";

}

**return** output;

}

}

## （已完成）FUNC\_3G\_CHARGE\_LEVEL

### Oracle数据库中对应OBJECT\_ID

350371

### 作用描述

3G用户出账收入分档函数

### 函数体内容

FUNCTION FUNC\_3G\_CHARGE\_LEVEL(TOTAL\_CHARGE IN VARCHAR2)

RETURN VARCHAR2 IS

/\*-------------------------------------------------------------------------------

函 数 名 : FUNC\_3G\_CHARGE\_LEVEL(3G9.11 出帐金额分档)

备 注 ：

编 写 人 CHENJUN

编写时间 : 20100916

-------------------------------------------------------------------------------\*/

RESULT VARCHAR2(2);

BEGIN

RESULT := CASE

WHEN TOTAL\_CHARGE <= 0 THEN '01'

WHEN TOTAL\_CHARGE>0 AND TOTAL\_CHARGE<=10 THEN '02'

WHEN TOTAL\_CHARGE>10 AND TOTAL\_CHARGE<=20 THEN '03'

WHEN TOTAL\_CHARGE>20 AND TOTAL\_CHARGE<=30 THEN '04'

WHEN TOTAL\_CHARGE>30 AND TOTAL\_CHARGE<=40 THEN '05'

WHEN TOTAL\_CHARGE>40 AND TOTAL\_CHARGE<=50 THEN '06'

WHEN TOTAL\_CHARGE<50 AND TOTAL\_CHARGE<=60 THEN '07'

WHEN TOTAL\_CHARGE>60 AND TOTAL\_CHARGE<=70 THEN '08'

WHEN TOTAL\_CHARGE>70 AND TOTAL\_CHARGE<=80 THEN '09'

WHEN TOTAL\_CHARGE>80 AND TOTAL\_CHARGE<=90 THEN '10'

WHEN TOTAL\_CHARGE>90 AND TOTAL\_CHARGE<=100 THEN '11'

WHEN TOTAL\_CHARGE>100 AND TOTAL\_CHARGE<=110 THEN '12'

WHEN TOTAL\_CHARGE>110 AND TOTAL\_CHARGE<=120 THEN '13'

WHEN TOTAL\_CHARGE>120 AND TOTAL\_CHARGE<=130 THEN '14'

WHEN TOTAL\_CHARGE>130 AND TOTAL\_CHARGE<=140 THEN '15'

WHEN TOTAL\_CHARGE>140 AND TOTAL\_CHARGE<=150 THEN '16'

WHEN TOTAL\_CHARGE<150 AND TOTAL\_CHARGE<=200 THEN '17'

WHEN TOTAL\_CHARGE>200 AND TOTAL\_CHARGE<=250 THEN '18'

WHEN TOTAL\_CHARGE>250 AND TOTAL\_CHARGE<=300 THEN '19'

WHEN TOTAL\_CHARGE>300 AND TOTAL\_CHARGE<=350 THEN '20'

WHEN TOTAL\_CHARGE>350 AND TOTAL\_CHARGE<=400 THEN '21'

WHEN TOTAL\_CHARGE>400 AND TOTAL\_CHARGE<=450 THEN '22'

WHEN TOTAL\_CHARGE>450 AND TOTAL\_CHARGE<=500 THEN '23'

WHEN TOTAL\_CHARGE>500 AND TOTAL\_CHARGE<=800 THEN '24'

WHEN TOTAL\_CHARGE<800 AND TOTAL\_CHARGE<=1000 THEN '25'

WHEN TOTAL\_CHARGE>1000 AND TOTAL\_CHARGE<=1500 THEN '26'

WHEN TOTAL\_CHARGE>1500 AND TOTAL\_CHARGE<=2000 THEN '27'

WHEN TOTAL\_CHARGE>2000 THEN '28'

END;

RETURN(RESULT);

END FUNC\_3G\_CHARGE\_LEVEL;

### Hive中处理思路及方法

#### 思路

通过if判断进行收入分档

#### 类名

org.apache.hadoop.hive.ql.udf.Func3GChargeLevel.java

#### Hive函数名

f\_3g\_charge\_level

#### Hive自定义函数代码

**package** org.apache.hadoop.hive.ql.udf;

**import** org.apache.hadoop.hive.ql.exec.UDF;

/\*

\* 函数功能：3G收入分档函数

\* Oracle中创建者：陈军

\* Hive自定义函数创建：平晓刚

\* 创建时间：2015-06-08

\* 对应原Oracle函数：DW.FUNC\_3G\_CHARGE\_LEVEL

\* 传入参数1：3G收入

\* 返回参数：收入分档

\* \*/

**public** **class** Func3GChargeLevel **extends** UDF {

**public** String evaluate(**double** charge\_fee) {

String output = **null**;

**if** (charge\_fee <= 0) {

output = "01";

} **else** **if** (charge\_fee > 0 && charge\_fee <= 10) {

output = "02";

} **else** **if** (charge\_fee > 10 && charge\_fee <= 20) {

output = "03";

} **else** **if** (charge\_fee > 20 && charge\_fee <= 30) {

output = "04";

} **else** **if** (charge\_fee > 30 && charge\_fee <= 40) {

output = "05";

} **else** **if** (charge\_fee > 40 && charge\_fee <= 50) {

output = "06";

} **else** **if** (charge\_fee > 50 && charge\_fee <= 60) {

output = "07";

} **else** **if** (charge\_fee > 60 && charge\_fee <= 70) {

output = "08";

} **else** **if** (charge\_fee > 70 && charge\_fee <= 80) {

output = "09";

} **else** **if** (charge\_fee > 80 && charge\_fee <= 90) {

output = "10";

} **else** **if** (charge\_fee > 90 && charge\_fee <= 100) {

output = "11";

} **else** **if** (charge\_fee > 100 && charge\_fee <= 110) {

output = "12";

} **else** **if** (charge\_fee > 110 && charge\_fee <= 120) {

output = "13";

} **else** **if** (charge\_fee > 120 && charge\_fee <= 130) {

output = "14";

} **else** **if** (charge\_fee > 130 && charge\_fee <= 140) {

output = "15";

} **else** **if** (charge\_fee > 140 && charge\_fee <= 150) {

output = "16";

} **else** **if** (charge\_fee > 150 && charge\_fee <= 200) {

output = "17";

} **else** **if** (charge\_fee > 200 && charge\_fee <= 250) {

output = "18";

} **else** **if** (charge\_fee > 250 && charge\_fee <= 300) {

output = "19";

} **else** **if** (charge\_fee > 300 && charge\_fee <= 350) {

output = "20";

} **else** **if** (charge\_fee > 350 && charge\_fee <= 400) {

output = "21";

} **else** **if** (charge\_fee > 400 && charge\_fee <= 450) {

output = "22";

} **else** **if** (charge\_fee > 450 && charge\_fee <= 500) {

output = "23";

} **else** **if** (charge\_fee > 500 && charge\_fee <= 800) {

output = "24";

} **else** **if** (charge\_fee > 800 && charge\_fee <= 1000) {

output = "25";

} **else** **if** (charge\_fee > 1000 && charge\_fee <= 1500) {

output = "26";

} **else** **if** (charge\_fee > 1500 && charge\_fee <= 2000) {

output = "27";

} **else** **if** (charge\_fee > 2000) {

output = "28";

}

**return** output;

}

}

## （已完成）FUNC\_GW\_UP\_INCOME\_LEVEL

### Oracle数据库中对应OBJECT\_ID

350375

### 作用描述

固网月报上报收入分档

### 函数体内容

FUNCTION FUNC\_GW\_UP\_INCOME\_LEVEL(CHARGE NUMBER)

/\*-----------------------------------------------------------------------

修改内容：固网月报上报收入分档

修 改 人：东方国信 孟令利

修改时间：20111022

------------------------------------------------------------------------\*/

RETURN VARCHAR2 IS

BEGIN

IF CHARGE <= 0 THEN

RETURN('01');

ELSIF CHARGE > 0 AND CHARGE <= 5 THEN

RETURN('02');

ELSIF CHARGE > 5 AND CHARGE <= 10 THEN

RETURN('03');

ELSIF CHARGE > 10 AND CHARGE <= 15 THEN

RETURN('04');

ELSIF CHARGE > 15 AND CHARGE <= 20 THEN

RETURN('05');

ELSIF CHARGE > 20 AND CHARGE <= 25 THEN

RETURN('06');

ELSIF CHARGE > 25 AND CHARGE <= 30 THEN

RETURN('07');

ELSIF CHARGE > 30 AND CHARGE <= 35 THEN

RETURN('08');

ELSIF CHARGE > 35 AND CHARGE <= 40 THEN

RETURN('09');

ELSIF CHARGE > 40 AND CHARGE <= 45 THEN

RETURN('10');

ELSIF CHARGE > 45 AND CHARGE <= 50 THEN

RETURN('11');

ELSIF CHARGE > 50 AND CHARGE <= 55 THEN

RETURN('12');

ELSIF CHARGE > 55 AND CHARGE <= 60 THEN

RETURN('13');

ELSIF CHARGE > 60 AND CHARGE <= 65 THEN

RETURN('14');

ELSIF CHARGE > 65 AND CHARGE <= 70 THEN

RETURN('15');

ELSIF CHARGE > 70 AND CHARGE <= 75 THEN

RETURN('16');

ELSIF CHARGE > 75 AND CHARGE <= 80 THEN

RETURN('17');

ELSIF CHARGE > 80 AND CHARGE <= 85 THEN

RETURN('18');

ELSIF CHARGE > 85 AND CHARGE <= 90 THEN

RETURN('19');

ELSIF CHARGE > 90AND CHARGE <= 95 THEN

RETURN('20');

ELSIF CHARGE > 95AND CHARGE <= 100 THEN

RETURN('21');

ELSIF CHARGE > 100 AND CHARGE <= 105 THEN

RETURN('22');

ELSIF CHARGE > 105 AND CHARGE <= 110 THEN

RETURN('23');

ELSIF CHARGE > 110 AND CHARGE <= 115 THEN

RETURN('24');

ELSIF CHARGE > 115 AND CHARGE <= 120 THEN

RETURN('25');

ELSIF CHARGE > 120 AND CHARGE <= 125 THEN

RETURN('26');

ELSIF CHARGE > 125 AND CHARGE <= 130 THEN

RETURN('27');

ELSIF CHARGE > 130 AND CHARGE <= 135 THEN

RETURN('28');

ELSIF CHARGE > 135 AND CHARGE <= 140 THEN

RETURN('29');

ELSIF CHARGE > 140 AND CHARGE <= 145 THEN

RETURN('30');

ELSIF CHARGE > 145 AND CHARGE <= 150 THEN

RETURN('31');

ELSIF CHARGE > 150 AND CHARGE <= 155 THEN

RETURN('32');

ELSIF CHARGE > 155 AND CHARGE <= 160 THEN

RETURN('33');

ELSIF CHARGE > 160 AND CHARGE <= 165 THEN

RETURN('34');

ELSIF CHARGE > 165 AND CHARGE <= 170 THEN

RETURN('35');

ELSIF CHARGE > 170 AND CHARGE <= 175 THEN

RETURN('36');

ELSIF CHARGE > 175 AND CHARGE <= 180 THEN

RETURN('37');

ELSIF CHARGE > 180 THEN

RETURN('38');

ELSE

RETURN('00');

END IF;

END;

### Hive中处理思路及方法

#### 思路

通过If判断进行收入分档

#### 类名

org.apache.hadoop.hive.ql.udf.FuncGWupIncomeLevel.java

#### Hive函数名

f\_gw\_up\_income\_level

#### Hive自定义函数代码

**package** org.apache.hadoop.hive.ql.udf;

**import** org.apache.hadoop.hive.ql.exec.UDF;

/\*

\* 函数功能：固网月报上报收入分档

\* Oracle中创建者：孟令利

\* Hive自定义函数创建：平晓刚

\* 创建时间：2015-06-08

\* 对应原Oracle函数：DW.FUNC\_GW\_UP\_INCOME\_LEVEL

\* 传入参数1：固网收入

\* 返回参数：收入分档

\* \*/

**public** **class** FuncGWupIncomeLevel **extends** UDF {

**public** String evaluate(**double** charge\_fee) {

String output = **null**;

**if** (charge\_fee <= 0) {

output = "01";

} **else** **if** (charge\_fee > 0 && charge\_fee <= 5) {

output = "02";

} **else** **if** (charge\_fee > 5 && charge\_fee <= 10) {

output = "03";

} **else** **if** (charge\_fee > 10 && charge\_fee <= 15) {

output = "04";

} **else** **if** (charge\_fee > 15 && charge\_fee <= 20) {

output = "05";

} **else** **if** (charge\_fee > 20 && charge\_fee <= 25) {

output = "06";

} **else** **if** (charge\_fee > 25 && charge\_fee <= 30) {

output = "07";

} **else** **if** (charge\_fee > 30 && charge\_fee <= 35) {

output = "08";

} **else** **if** (charge\_fee > 35 && charge\_fee <= 40) {

output = "09";

} **else** **if** (charge\_fee > 40 && charge\_fee <= 45) {

output = "10";

} **else** **if** (charge\_fee > 45 && charge\_fee <= 50) {

output = "11";

} **else** **if** (charge\_fee > 50 && charge\_fee <= 55) {

output = "12";

} **else** **if** (charge\_fee > 55 && charge\_fee <= 60) {

output = "13";

} **else** **if** (charge\_fee > 60 && charge\_fee <= 65) {

output = "14";

} **else** **if** (charge\_fee > 65 && charge\_fee <= 70) {

output = "15";

} **else** **if** (charge\_fee > 70 && charge\_fee <= 75) {

output = "16";

} **else** **if** (charge\_fee > 75 && charge\_fee <= 80) {

output = "17";

} **else** **if** (charge\_fee > 80 && charge\_fee <= 85) {

output = "18";

} **else** **if** (charge\_fee > 85 && charge\_fee <= 90) {

output = "19";

} **else** **if** (charge\_fee > 90 && charge\_fee <= 95) {

output = "20";

} **else** **if** (charge\_fee > 95 && charge\_fee <= 100) {

output = "21";

} **else** **if** (charge\_fee > 100 && charge\_fee <= 105) {

output = "22";

} **else** **if** (charge\_fee > 105 && charge\_fee <= 110) {

output = "23";

} **else** **if** (charge\_fee > 110 && charge\_fee <= 115) {

output = "24";

} **else** **if** (charge\_fee > 115 && charge\_fee <= 120) {

output = "25";

} **else** **if** (charge\_fee > 120 && charge\_fee <= 125) {

output = "26";

} **else** **if** (charge\_fee > 125 && charge\_fee <= 130) {

output = "27";

} **else** **if** (charge\_fee > 130 && charge\_fee <= 135) {

output = "28";

} **else** **if** (charge\_fee > 135 && charge\_fee <= 140) {

output = "29";

} **else** **if** (charge\_fee > 140 && charge\_fee <= 145) {

output = "30";

} **else** **if** (charge\_fee > 145 && charge\_fee <= 150) {

output = "31";

} **else** **if** (charge\_fee > 150 && charge\_fee <= 155) {

output = "32";

} **else** **if** (charge\_fee > 155 && charge\_fee <= 160) {

output = "33";

} **else** **if** (charge\_fee > 160 && charge\_fee <= 165) {

output = "34";

} **else** **if** (charge\_fee > 165 && charge\_fee <= 170) {

output = "35";

} **else** **if** (charge\_fee > 170 && charge\_fee <= 175) {

output = "36";

} **else** **if** (charge\_fee > 175 && charge\_fee <= 180) {

output = "37";

} **else** **if** (charge\_fee > 180) {

output = "38";

} **else** {

output = "00";

}

**return** output;

}

}

## （已完成）FUNC\_FLUX\_LEVEL

### Oracle数据库中对应OBJECT\_ID

350373

### 作用描述

流量分档（指标上报）

### 函数体内容

FUNCTION FUNC\_FLUX\_LEVEL(

FLUX NUMBER)

/\*-----------------------------------------------------------------------

修改内容：为适应上报V2.0细分1000-2000的分档为1000-1500、1500-2000

修 改 人：东方国信 赵鹏

修改时间：20061130

------------------------------------------------------------------------\*/

RETURN VARCHAR2

IS

BEGIN

IF FLUX <= 0 THEN RETURN ('00');

ELSIF FLUX > 0 AND FLUX <= 5 THEN RETURN ('01');

ELSIF FLUX > 5 AND FLUX <= 10 THEN RETURN ('02');

ELSIF FLUX > 10 AND FLUX <= 15 THEN RETURN ('03');

ELSIF FLUX > 15 AND FLUX <= 20 THEN RETURN ('04');

ELSIF FLUX > 20 AND FLUX <= 25 THEN RETURN ('05');

ELSIF FLUX > 25 AND FLUX <= 30 THEN RETURN ('06');

ELSIF FLUX > 30 AND FLUX <= 40 THEN RETURN ('07');

ELSIF FLUX > 40 AND FLUX <= 50 THEN RETURN ('08');

ELSIF FLUX > 50 AND FLUX <= 60 THEN RETURN ('09');

ELSIF FLUX > 60 AND FLUX <= 70 THEN RETURN ('10');

ELSIF FLUX > 70 AND FLUX <= 80 THEN RETURN ('11');

ELSIF FLUX > 80 AND FLUX <= 90 THEN RETURN ('12');

ELSIF FLUX > 90 AND FLUX <= 100 THEN RETURN ('13');

ELSIF FLUX > 100 AND FLUX <= 120 THEN RETURN ('14');

ELSIF FLUX > 120 AND FLUX <= 140 THEN RETURN ('15');

ELSIF FLUX > 140 AND FLUX <= 160 THEN RETURN ('16');

ELSIF FLUX > 160 AND FLUX <= 200 THEN RETURN ('17');

ELSIF FLUX > 200 AND FLUX <= 300 THEN RETURN ('18');

ELSIF FLUX > 300 AND FLUX <= 500 THEN RETURN ('19');

ELSIF FLUX > 500 AND FLUX <= 1000 THEN RETURN ('20');

ELSE RETURN('FF');

END IF;

END;

### Hive中处理思路及方法

#### 思路

通过If判断进行流量分档

#### 类名

org.apache.hadoop.hive.ql.udf.FuncFluxLevel.java

#### Hive函数名

f\_flux\_level

#### Hive自定义函数代码

**package** org.apache.hadoop.hive.ql.udf;

**import** org.apache.hadoop.hive.ql.exec.UDF;

/\*

\* 函数功能：流量分档（指标上报）

\* Oracle中创建者：赵鹏

\* Hive自定义函数创建：平晓刚

\* 创建时间：2015-06-08

\* 对应原Oracle函数：DW.FUNC\_GW\_UP\_INCOME\_LEVEL

\* 传入参数1：流量

\* 返回参数：流量分档

\* \*/

**public** **class** FuncFluxLevel **extends** UDF {

**public** String evaluate(**double** flux) {

String output = **null**;

**if** (flux <= 0) {

output = "00";

} **else** **if** (flux > 0 && flux <= 5) {

output = "01";

} **else** **if** (flux > 5 && flux <= 10) {

output = "02";

} **else** **if** (flux > 10 && flux <= 15) {

output = "03";

} **else** **if** (flux > 15 && flux <= 20) {

output = "04";

} **else** **if** (flux > 20 && flux <= 25) {

output = "05";

} **else** **if** (flux > 25 && flux <= 30) {

output = "06";

} **else** **if** (flux > 30 && flux <= 40) {

output = "07";

} **else** **if** (flux > 40 && flux <= 50) {

output = "08";

} **else** **if** (flux > 50 && flux <= 60) {

output = "09";

} **else** **if** (flux > 60 && flux <= 70) {

output = "10";

} **else** **if** (flux > 70 && flux <= 80) {

output = "11";

} **else** **if** (flux > 80 && flux <= 90) {

output = "12";

} **else** **if** (flux > 90 && flux <= 100) {

output = "13";

} **else** **if** (flux > 100 && flux <= 120) {

output = "14";

} **else** **if** (flux > 120 && flux <= 140) {

output = "15";

} **else** **if** (flux > 140 && flux <= 160) {

output = "16";

} **else** **if** (flux > 160 && flux <= 200) {

output = "17";

} **else** **if** (flux > 200 && flux <= 300) {

output = "18";

} **else** **if** (flux > 300 && flux <= 500) {

output = "19";

} **else** **if** (flux > 500 && flux <= 1000) {

output = "20";

} **else** {

output = "FF";

}

**return** output;

}

}

## FUNC\_GW\_COMM\_PRODUCT\_TYPE

### Oracle数据库中对应OBJECT\_ID

350374

### 作用描述

固网佣金产品分类函数

### 函数体内容

FUNCTION FUNC\_GW\_COMM\_PRODUCT\_TYPE(KEY\_SOURCE\_TYPE\_CD NUMBER,

PRODUCT\_KIND\_ID NUMBER,

SVCID NUMBER,

MODID NUMBER)

/\*-----------------------------------------------------------------------

内容：固网佣金产品分档

创建人：张惠

创建时间：20100226

------------------------------------------------------------------------\*/

RETURN VARCHAR2 IS

BEGIN

IF SVCID IN (6352) THEN

RETURN('17960帐号后付');

ELSIF SVCID IN ('6351', '6354') THEN

RETURN('17969IP主叫直拨');

ELSIF SVCID IN ('6242', '6243') THEN

RETURN('800');

ELSIF SVCID IN (6241,'7148') THEN

RETURN('4006');

ELSIF SVCID IN ('6161') THEN

RETURN('11698');

ELSIF SVCID IN ('6281','6282', '6283','6284') THEN

RETURN('196主叫直拨');

ELSIF SVCID IN ('7430', '7431', '7447', '7448','7455','7456','7453','6713','7458') THEN

RETURN('IDC业务');

ELSIF (SVCID IN ('6602','6612','6066','6611') or (SVCID in ('1050') and KEY\_SOURCE\_TYPE\_CD=52014 )) THEN

RETURN('ICT业务');

ELSIF SVCID IN ('6457') THEN

RETURN('超级总机');

ELSIF SVCID IN ('7105', '7114') THEN

RETURN('公用电话');

ELSIF SVCID IN ('6321', '6322','6324') THEN

RETURN('公众宽带接入');

ELSIF SVCID IN ('6704') THEN

RETURN('固话业务');

ELSIF SVCID IN ('6460') THEN

RETURN('呼叫中心套餐（联通）');

ELSIF SVCID IN ('7138') THEN

RETURN('话批话吧');

ELSIF SVCID IN ('6716') THEN

RETURN('会议电话');

ELSIF SVCID IN ('7423') THEN

RETURN('基础数据业务');

ELSIF SVCID IN ('6701', '6708', '7101', '7102') THEN

RETURN('家庭固话业务');

ELSIF SVCID IN ('6710') THEN

RETURN('空中会议室');

ELSIF SVCID IN ('6722') THEN

RETURN('临时展会');

ELSIF SVCID IN ('7438', '7439') THEN

RETURN('普通专线业务');

ELSIF (SVCID IN ('7424', '7425') or (SVCID in ('1050') and KEY\_SOURCE\_TYPE\_CD=52002 ) )THEN

RETURN('其他数据业务');

ELSIF SVCID IN ('6714', '7127', '7132', '7140', '7143', '7144','6285') THEN

RETURN('企业固话业务');

ELSIF SVCID IN ('7420', '7421', '7441', '7442','6461') THEN

RETURN('企业宽带业务');

ELSIF SVCID IN ('6451') THEN

RETURN('商务热线');

ELSIF SVCID IN ('6723') THEN

RETURN('上网卡');

ELSIF SVCID IN ('6303') THEN

RETURN('网通E盾'); ------20130402 张玉龙添加

ELSIF SVCID IN ('7133', '7141', '7142') THEN

RETURN('数字中继');

ELSIF SVCID IN ('6717') THEN

RETURN('随行通');

ELSIF SVCID IN ('6712','7131') THEN

RETURN('网通新传真');

ELSIF SVCID IN

('6504','7426', '7427', '7428', '7429', '7443', '7444', '7445', '7446','6353','7452') THEN

RETURN('网元出租业务');

ELSIF SVCID IN ('6452','6459') THEN

RETURN('语音天地呼叫中心');

ELSIF SVCID IN ('6454') THEN

RETURN('视频导航');

ELSIF SVCID IN ('6705') THEN

RETURN('软PHONE(一号通)(后付)');

ELSIF SVCID IN ('7457') THEN

RETURN('中小企业应用(预付)');

ELSE

RETURN(SVCID);

END IF;

END FUNC\_GW\_COMM\_PRODUCT\_TYPE;

### Hive中处理思路及方法

#### 思路

#### 类名

#### Hive函数名

#### Hive自定义函数代码

## （已完成）FUNC\_PRODUCT\_TYPE\_YW

### Oracle数据库中对应OBJECT\_ID

350382

### 作用描述

移网产品分类函数

### 函数体内容

FUNCTION FUNC\_PRODUCT\_TYPE\_YW(IS\_3G\_USER VARCHAR2,

CARD\_TYPE VARCHAR2,

PAY\_MODE VARCHAR2,

ACTIVITY\_TYPE VARCHAR2,

DINNER\_NO VARCHAR2,

TRADE\_TYPE VARCHAR2)

/\*-----------------------------------------------------------------------

创建内容：移网品分类函数

创建人：刘时燕（网格化系统使用）

创建时间：20120401

------------------------------------------------------------------------\*/

RETURN VARCHAR2 IS

BEGIN

IF IS\_3G\_USER = '0' AND PAY\_MODE = '01' THEN

RETURN('010101'); --2G后付费

ELSIF IS\_3G\_USER = '0' AND PAY\_MODE = '02' THEN

RETURN('010102'); --2G准预付费

ELSIF IS\_3G\_USER = '0' AND PAY\_MODE IN ('03', '04', '05') THEN

RETURN('010103'); -----2G预付费

ELSIF IS\_3G\_USER = '1' AND CARD\_TYPE = '0' AND PAY\_MODE = '01' THEN --3G手机后付费

IF ACTIVITY\_TYPE = '01' AND SUBSTR(DINNER\_NO, 1, 2) NOT IN ('03') THEN --(去掉IPHONE合约)

RETURN('01020101'); --预存话费优惠购机

ELSIF ACTIVITY\_TYPE = '02' AND SUBSTR(DINNER\_NO, 1, 2)NOT IN ('03') THEN

RETURN('01020102'); --购机送话费

ELSIF ACTIVITY\_TYPE = '03' AND SUBSTR(DINNER\_NO, 1, 2) NOT IN ('03') THEN

RETURN('01020103'); --存费送费

ELSIF SUBSTR(DINNER\_NO, 1, 2) = '03' THEN

RETURN('01020104'); --IPHONE合约

ELSIF TRADE\_TYPE = '行业' THEN

RETURN('01020106'); --行业应用

ELSE

RETURN('01020105'); --后付费其他

END IF;

ELSIF IS\_3G\_USER = '1' AND CARD\_TYPE = '0' AND PAY\_MODE = '02' THEN

RETURN('010202'); --3G手机准预付费

ELSIF IS\_3G\_USER = '1' AND CARD\_TYPE = '0' AND PAY\_MODE IN ('03', '04', '05') THEN

RETURN('010203'); --3G手机预付费

ELSIF IS\_3G\_USER = '1' AND CARD\_TYPE = '1' AND PAY\_MODE = '01' THEN

RETURN('010301'); --3G上网卡后付费

ELSIF IS\_3G\_USER = '1' AND CARD\_TYPE = '1' AND PAY\_MODE = '02' THEN

RETURN('010302'); --3G上网卡准预付费

ELSIF IS\_3G\_USER = '1' AND CARD\_TYPE = '1' AND PAY\_MODE IN ('03', '04', '05') THEN

RETURN('010303'); --3G上网卡预付费

END IF;

END;

### Hive中处理思路及方法

#### 思路

通过If判断进行移网业务类型分类

#### 类名

org.apache.hadoop.hive.ql.udf.FuncProductTypeYW.java

#### Hive函数名

f\_product\_type\_yw

#### Hive自定义函数代码

**package** org.apache.hadoop.hive.ql.udf;

**import** org.apache.hadoop.hive.ql.exec.UDF;

/\*

\* 函数功能：移网产品分类函数

\* Oracle中创建者：刘时燕

\* Hive自定义函数创建：平晓刚

\* 创建时间：2015-06-08

\* 对应原Oracle函数：DW.FUNC\_PRODUCT\_TYPE\_YW

\* 传入参数1：

\* 传入参数2：

\* 传入参数3：

\* 传入参数4：

\* 传入参数5：

\* 传入参数6：

\* 返回参数：移网产品分类

\* \*/

**public** **class** FuncProductTypeYW **extends** UDF {

**public** String evaluate(String is\_3g\_user, String card\_type,

String pay\_mode, String activity\_type, String dinner\_no,

String trade\_type) {

String output = **null**;

**if** (is\_3g\_user.equals("0") && pay\_mode.equals("01")) {

output = "010101";// 2G后付费

} **else** **if** (is\_3g\_user.equals("0") && pay\_mode.equals("02")) {

output = "010102";// 2G准预付费

} **else** **if** (is\_3g\_user.equals("0")

&& (pay\_mode.equals("03") | pay\_mode.equals("04") | pay\_mode

.equals("05"))) {

output = "010103";// 2G预付费

} **else** **if** (is\_3g\_user.equals("1") && card\_type.equals("0")

&& pay\_mode.equals("01")) {

// 3G手机后付费

**if** (activity\_type.equals("01")

&& !dinner\_no.substring(0, 2).equals("03")) {// 去掉IPHONE合约

output = "01020101";// 预存话费优惠购机

} **else** **if** (activity\_type.equals("02")

&& !dinner\_no.substring(0, 2).equals("03")) {

output = "01020102";// 购机送话费

} **else** **if** (activity\_type.equals("03")

&& !dinner\_no.substring(0, 2).equals("03")) {

output = "01020103";// 存费送费

} **else** **if** (dinner\_no.substring(0, 2).equals("03")) {

output = "01020104";// IPHONE合约

} **else** **if** (trade\_type.equals("行业")) {

output = "01020106";// 行业应用

} **else** {

output = "01020105";// 后付费其他

}

} **else** **if** (is\_3g\_user.equals("1") && card\_type.equals("0")

&& pay\_mode.equals("02")) {

output = "010202";// 3G手机准预付费

} **else** **if** (is\_3g\_user.equals("1")

&& card\_type.equals("0")

&& (pay\_mode.equals("03") | pay\_mode.equals("04") | pay\_mode

.equals("05"))) {

output = "010203";// 3G手机预付费

} **else** **if** (is\_3g\_user.equals("1") && card\_type.equals("1")

&& pay\_mode.equals("01")) {

output = "010301";// 3G上网卡后付费

} **else** **if** (is\_3g\_user.equals("1") && card\_type.equals("1")

&& pay\_mode.equals("02")) {

output = "010302";// 3G上网卡准预付费

} **else** **if** (is\_3g\_user.equals("1")

&& card\_type.equals("1")

&& (pay\_mode.equals("03") | pay\_mode.equals("04") | pay\_mode

.equals("05"))) {

output = "010303";// 3G上网卡预付费

}

**return** output;

}

}

## （已完成）F\_GROUP\_ONNET\_USERS\_LVL

### Oracle数据库中对应OBJECT\_ID

350394

### 作用描述

集团网上用户分档

### 函数体内容

FUNCTION F\_GROUP\_ONNET\_USERS\_LVL(GROUP\_USER IN VARCHAR2)

RETURN VARCHAR2 IS

/\*-------------------------------------------------------------------------------

函数名 : F\_GROUP\_ONNET\_USERS\_LVL(集团网上用户数分档)

备注：

编写人：jiaoxiulian

编写时间 : 20070703

修改人:平晓刚

修改时间:20090112

修改原因:上海本地化

-------------------------------------------------------------------------------\*/

BEGIN

IF GROUP\_USER = 0 THEN

RETURN('00');

ELSIF GROUP\_USER > 0 AND GROUP\_USER <= 10 THEN

RETURN('01');

ELSIF GROUP\_USER > 10 AND GROUP\_USER <= 20 THEN

RETURN('02');

ELSIF GROUP\_USER > 20 AND GROUP\_USER <= 50 THEN

RETURN('03');

ELSIF GROUP\_USER > 50 AND GROUP\_USER <= 100 THEN

RETURN('04');

ELSIF GROUP\_USER > 100 AND GROUP\_USER <= 200 THEN

RETURN('05');

ELSIF GROUP\_USER > 200 AND GROUP\_USER <= 300 THEN

RETURN('06');

ELSIF GROUP\_USER > 300 AND GROUP\_USER <= 500 THEN

RETURN('07');

ELSIF GROUP\_USER > 500 AND GROUP\_USER <= 1000 THEN

RETURN('08');

ELSIF GROUP\_USER > 1000 AND GROUP\_USER <= 2000 THEN

RETURN('09');

ELSIF GROUP\_USER > 2000 AND GROUP\_USER <= 2500 THEN

RETURN('10');

ELSIF GROUP\_USER > 2500 AND GROUP\_USER <= 3000 THEN

RETURN('11');

ELSIF GROUP\_USER > 3000 AND GROUP\_USER <= 3500 THEN

RETURN('12');

ELSIF GROUP\_USER > 3500 AND GROUP\_USER <= 4000 THEN

RETURN('13');

ELSIF GROUP\_USER > 4000 AND GROUP\_USER <= 4500 THEN

RETURN('14');

ELSIF GROUP\_USER > 4500 AND GROUP\_USER <= 5000 THEN

RETURN('15');

ELSIF GROUP\_USER > 5000 THEN

RETURN('16');

ELSE

RETURN('00');

END IF;

END F\_GROUP\_ONNET\_USERS\_LVL;

### Hive中处理思路及方法

#### 思路

通过if-else代替

#### 类名

org.apache.hadoop.hive.ql.udf.FGroupOnnetUsersLvl.java

#### Hive函数名

f\_group\_onnet\_users\_lvl

#### Hive自定义函数代码

|  |
| --- |
| **package** org.apache.hadoop.hive.ql.udf;  **import** org.apache.hadoop.hive.ql.exec.udf;  /\*  \* 函数功能：集团网上用户数分档  \* Oracle中创建者：jiaoxiulian  \* Hive自定义函数创建：张海东  \* 创建时间：2015-06-09  \* 对应原Oracle函数：DW.F\_GROUP\_ONNET\_USERS\_LVL  \* 传入参数：用户  \* 返回参数：分档  \* \*/  **public** **class** FGroupOnnetUsersLvl **extends** UDF{    **public** String evaluate(**int** group\_user) {    **if**(group\_user == 0)  **return** "00";  **else** **if**(group\_user > 0 && group\_user <= 10)  **return** "01";  **else** **if**(group\_user > 10 && group\_user <= 20)  **return** "02";  **else** **if**(group\_user > 20 && group\_user <= 50)  **return** "03";  **else** **if**(group\_user > 50 && group\_user <= 100)  **return** "04";  **else** **if**(group\_user > 100 && group\_user <= 200)  **return** "05";  **else** **if**(group\_user > 200 && group\_user <= 300)  **return** "06";  **else** **if**(group\_user > 300 && group\_user <= 500)  **return** "07";  **else** **if**(group\_user > 500 && group\_user <= 1000)  **return** "08";  **else** **if**(group\_user > 1000 && group\_user <= 2000)  **return** "09";  **else** **if**(group\_user > 2000 && group\_user <= 2500)  **return** "10";  **else** **if**(group\_user > 2500 && group\_user <= 3000)  **return** "11";  **else** **if**(group\_user > 3000 && group\_user <= 3500)  **return** "12";  **else** **if**(group\_user > 3500 && group\_user <= 4000)  **return** "13";  **else** **if**(group\_user > 4000 && group\_user <= 4500)  **return** "14";  **else** **if**(group\_user > 4500 && group\_user <= 5000)  **return** "15";  **else** **if**(group\_user > 5000)  **return** "16";  **return** "00";  }  } |

## （已完成）F\_GW\_DURATION\_LEVEL

### Oracle数据库中对应OBJECT\_ID

350395

### 作用描述

固网通话时长分档

### 函数体内容

FUNCTION F\_GW\_DURATION\_LEVEL(V\_DURATION IN NUMBER)

RETURN VARCHAR2 IS

/\*------------------------------------------------------------------

函数名 : F\_GW\_DURATION\_LEVEL

使用参数：V\_DURATION 通话时长

生成时间：2009.8.27

编写人：徐杰

函数说明：返回固网通话时长分档

-------------------------------------------------------------------\*/

V\_RESULT VARCHAR2(4);

BEGIN

V\_RESULT :=

CASE WHEN V\_DURATION <= 0 THEN '-1'

WHEN V\_DURATION > 0 AND V\_DURATION <= 5 THEN '01'

WHEN V\_DURATION > 5 AND V\_DURATION <= 10 THEN '02'

WHEN V\_DURATION > 10 AND V\_DURATION <= 15 THEN '03'

WHEN V\_DURATION > 15 AND V\_DURATION <= 20 THEN '04'

WHEN V\_DURATION > 20 AND V\_DURATION <= 25 THEN '05'

WHEN V\_DURATION > 25 AND V\_DURATION <= 30 THEN '06'

WHEN V\_DURATION > 30 AND V\_DURATION <= 35 THEN '07'

WHEN V\_DURATION > 35 AND V\_DURATION <= 40 THEN '08'

WHEN V\_DURATION > 40 AND V\_DURATION <= 45 THEN '09'

WHEN V\_DURATION > 45 AND V\_DURATION <= 50 THEN '10'

WHEN V\_DURATION > 50 AND V\_DURATION <= 55 THEN '11'

WHEN V\_DURATION > 55 AND V\_DURATION <= 60 THEN '12'

WHEN V\_DURATION > 60 AND V\_DURATION <= 65 THEN '13'

WHEN V\_DURATION > 65 AND V\_DURATION <= 70 THEN '14'

WHEN V\_DURATION > 70 AND V\_DURATION <= 75 THEN '15'

WHEN V\_DURATION > 75 AND V\_DURATION <= 80 THEN '16'

WHEN V\_DURATION > 80 AND V\_DURATION <= 85 THEN '17'

WHEN V\_DURATION > 85 AND V\_DURATION <= 90 THEN '18'

ELSE '19' END;

RETURN V\_RESULT;

END F\_GW\_DURATION\_LEVEL;

### Hive中处理思路及方法

#### 思路

通过if-else代替

#### 类名

org.apache.hadoop.hive.ql.udf.FGWDurationLevel.java

#### Hive函数名

f\_gw\_duration\_level

#### Hive自定义函数代码

|  |
| --- |
| **package** org.apache.hadoop.hive.ql.udf;  **import** org.apache.hadoop.hive.ql.exec.udf;  /\*  \* 函数功能：返回固网通话时长分档  \* Oracle中创建者：徐杰  \* Hive自定义函数创建：张海东  \* 创建时间：2015-06-09  \* 对应原Oracle函数：DW.F\_GW\_DURATION\_LEVEL  \* 传入参数：通话时长  \* 返回参数：分档  \* \*/  **public** **class** FGWDurationLevel **extends** UDF{  **public** String evaluate(**int** v\_duration) {    **if**(v\_duration <= 0)  **return** "-1";  **else** **if**(v\_duration > 0 && v\_duration <= 5)  **return** "01";  **else** **if**(v\_duration > 5 && v\_duration <= 10)  **return** "02";  **else** **if**(v\_duration > 10 && v\_duration <= 15)  **return** "03";  **else** **if**(v\_duration > 15 && v\_duration <= 20)  **return** "04";  **else** **if**(v\_duration > 20 && v\_duration <= 25)  **return** "05";  **else** **if**(v\_duration > 25 && v\_duration <= 30)  **return** "06";  **else** **if**(v\_duration > 30 && v\_duration <= 35)  **return** "07";  **else** **if**(v\_duration > 35 && v\_duration <= 40)  **return** "08";  **else** **if**(v\_duration > 40 && v\_duration <= 45)  **return** "09";  **else** **if**(v\_duration > 45 && v\_duration <= 50)  **return** "10";  **else** **if**(v\_duration > 50 && v\_duration <= 55)  **return** "11";  **else** **if**(v\_duration > 55 && v\_duration <= 60)  **return** "12";  **else** **if**(v\_duration > 60 && v\_duration <= 65)  **return** "13";  **else** **if**(v\_duration > 65 && v\_duration <= 70)  **return** "14";  **else** **if**(v\_duration > 70 && v\_duration <= 75)  **return** "15";  **else** **if**(v\_duration > 75 && v\_duration <= 80)  **return** "16";  **else** **if**(v\_duration > 80 && v\_duration <= 85)  **return** "17";  **else** **if**(v\_duration > 85 && v\_duration <= 90)  **return** "18";  **return** "19";  }  } |

## （已完成）F\_GROUP\_CUST\_INCOME\_LVL

### Oracle数据库中对应OBJECT\_ID

350393

### 作用描述

集团业务收入分档

### 函数体内容

FUNCTION F\_GROUP\_CUST\_INCOME\_LVL(TOTAL\_FEE IN VARCHAR2) RETURN VARCHAR2

IS

/\*-------------------------------------------------------------------------------

函数名 : F\_GROUP\_CUST\_INCOME\_LVL(集团业务收入分档)

备注：

编写人：jiaoxiulian

编写时间 : 20070703

修改人:平晓刚

修改时间:20090112

修改原因:上海本地化

-------------------------------------------------------------------------------\*/

BEGIN

IF TOTAL\_FEE=0 THEN RETURN ('00');

ELSIF TOTAL\_FEE>0 AND TOTAL\_FEE<=500 THEN RETURN ('01');

ELSIF TOTAL\_FEE>500 AND TOTAL\_FEE<=1000 THEN RETURN ('02');

ELSIF TOTAL\_FEE>1000 AND TOTAL\_FEE<=2000 THEN RETURN ('03');

ELSIF TOTAL\_FEE>2000 AND TOTAL\_FEE<=3000 THEN RETURN ('04');

ELSIF TOTAL\_FEE>3000 AND TOTAL\_FEE<=5000 THEN RETURN ('05');

ELSIF TOTAL\_FEE>5000 AND TOTAL\_FEE<=10000 THEN RETURN ('06');

ELSIF TOTAL\_FEE>10000 AND TOTAL\_FEE<=20000 THEN RETURN ('07');

ELSIF TOTAL\_FEE>20000 AND TOTAL\_FEE<=30000 THEN RETURN ('08');

ELSIF TOTAL\_FEE>30000 AND TOTAL\_FEE<=40000 THEN RETURN ('09');

ELSIF TOTAL\_FEE>40000 AND TOTAL\_FEE<=50000 THEN RETURN ('10');

ELSIF TOTAL\_FEE>50000 AND TOTAL\_FEE<=60000 THEN RETURN ('11');

ELSIF TOTAL\_FEE>60000 AND TOTAL\_FEE<=80000 THEN RETURN ('12');

ELSIF TOTAL\_FEE>80000 THEN RETURN ('13');

ELSE RETURN ('00');

END IF;

END F\_GROUP\_CUST\_INCOME\_LVL;

### Hive中处理思路及方法

#### 思路

通过if-else代替

#### 类名

org.apache.hadoop.hive.ql.udf.FGroupCustIncomeLvl.java

#### Hive函数名

f\_group\_cust\_income\_lvl

#### Hive自定义函数代码

|  |
| --- |
| **package** org.apache.hadoop.hive.ql.udf;  **import** org.apache.hadoop.hive.ql.exec.UDF;  /\*  \* 函数功能：集团业务收入分档  \* Oracle中创建者：jiaoxiulian  \* Hive自定义函数创建：张海东  \* 创建时间：2015-06-09  \* 对应原Oracle函数：DW. F\_GROUP\_CUST\_INCOME\_LVL  \* 传入参数：业务收入  \* 返回参数：分档  \* \*/  **public** **class** FGroupCustIncomeLvl **extends** UDF{    **public** String evaluate(**int** total\_fee) {    **if**(total\_fee == 0)  **return** "00";  **else** **if**(total\_fee > 0 && total\_fee <= 500)  **return** "01";  **else** **if**(total\_fee > 500 && total\_fee <= 1000)  **return** "02";  **else** **if**(total\_fee > 1000 && total\_fee <= 2000)  **return** "03";  **else** **if**(total\_fee > 2000 && total\_fee <= 3000)  **return** "04";  **else** **if**(total\_fee > 3000 && total\_fee <= 5000)  **return** "05";  **else** **if**(total\_fee > 5000 && total\_fee <= 10000)  **return** "06";  **else** **if**(total\_fee > 10000 && total\_fee <= 20000)  **return** "07";  **else** **if**(total\_fee > 20000 && total\_fee <= 30000)  **return** "08";  **else** **if**(total\_fee > 30000 && total\_fee <= 40000)  **return** "09";  **else** **if**(total\_fee > 40000 && total\_fee <= 50000)  **return** "10";  **else** **if**(total\_fee > 50000 && total\_fee <= 60000)  **return** "11";  **else** **if**(total\_fee > 60000 && total\_fee <= 80000)  **return** "12";  **else** **if**(total\_fee > 80000)  **return** "13";  **return** "00";  }  } |

## （已完成）F\_BRAND

### Oracle数据库中对应OBJECT\_ID

350391

### 作用描述

业务品牌分档

### 函数体内容

FUNCTION F\_BRAND(BRAND IN VARCHAR2, PAY\_MODE IN VARCHAR2)

RETURN VARCHAR2 IS

/\*-------------------------------------------------------------------------------

函数名：F\_CHANGE\_BRAND(业务品牌)

备注：

编写人：JIAOXIULIAN

编写时间 : 20070619

修改人：平晓刚

修改时间：2008-09-22

修改原因：根据上海本地化修改

-------------------------------------------------------------------------------\*/

RESULT VARCHAR2(2);

BEGIN

RESULT := CASE WHEN PAY\_MODE = '01' AND BRAND = '10' THEN '01' --世界风

WHEN PAY\_MODE = '01' AND BRAND = '20' THEN '01' --世界风

WHEN PAY\_MODE = '01' AND BRAND = '40' THEN '02' --新势力

WHEN PAY\_MODE = '01' AND BRAND = '90' THEN '01' --世界风

WHEN PAY\_MODE = '01' AND BRAND = '30' THEN '01' --世界风

WHEN PAY\_MODE = '01' AND BRAND = '50' THEN '01' --世界风

when pay\_mode='01' and (brand is null) then '01'--世界风

WHEN PAY\_MODE = '02' AND BRAND = '10' THEN '03' --如意通

WHEN PAY\_MODE = '02' AND BRAND = '20' THEN '01' --世界风

WHEN PAY\_MODE = '02' AND BRAND = '30' THEN '01' --世界风

WHEN PAY\_MODE = '02' AND BRAND = '50' THEN '01' --世界风

WHEN PAY\_MODE = '02' AND BRAND = '40' THEN '02' --新势力

WHEN PAY\_MODE = '02' AND BRAND = '90' THEN '01' --世界风

when pay\_mode='02' and (brand is null) then '01'--世界风

WHEN PAY\_MODE = '03' THEN '03' --如意通

WHEN PAY\_MODE = '05' THEN '03' --如意通--OCS用户

WHEN PAY\_MODE = '04' AND BRAND = '10' THEN '03' --如意通

WHEN PAY\_MODE = '04' AND BRAND = '20' THEN '03' --如意通

WHEN PAY\_MODE = '04' AND BRAND = '30' THEN '03' --如意通

WHEN PAY\_MODE = '04' AND BRAND = '50' THEN '03' --如意通

WHEN PAY\_MODE = '04' AND BRAND = '40' THEN '02' --新势力

WHEN PAY\_MODE = '04' AND BRAND = '90' THEN '03' --如意通

when pay\_mode='04' and (brand is null) then '03'--如意通

else '01'

END;

RETURN(RESULT);

END F\_BRAND;

### Hive中处理思路及方法

#### 思路

通过if-else代替

#### 类名

org.apache.hadoop.hive.ql.udf.FBrand.java

#### Hive函数名

f\_brand

#### Hive自定义函数代码

|  |
| --- |
| **package** org.apache.hadoop.hive.ql.udf;  **import** org.apache.hadoop.hive.ql.exec.UDF;  /\*  \* 函数功能：业务品牌分档  \* Oracle中创建者：jiaoxiulian  \* Hive自定义函数创建：张海东  \* 创建时间：2015-06-09  \* 对应原Oracle函数：DW.F\_CHANGE\_BRAND  \* 传入参数：商标（brand），支付类型（pay mode）  \* 返回参数：分档  \* \*/  **public** **class** FBrand **extends** UDF{    **public** String evaluate(String pay\_mode,String brand) {  **if**(pay\_mode == "01" && brand == "10")  **return** "01";//世界风  **else** **if**(pay\_mode == "01" && brand == "20")  **return** "01";//世界风  **else** **if**(pay\_mode == "01" && brand == "40")  **return** "02";//新势力  **else** **if**(pay\_mode == "01" && brand == "90")  **return** "01";//世界风  **else** **if**(pay\_mode == "01" && brand == "30")  **return** "01";//世界风  **else** **if**(pay\_mode == "01" && brand == "50")  **return** "01";//世界风  **else** **if**(pay\_mode == "01" && brand == **null**)  **return** "01";//世界风  **else** **if**(pay\_mode == "02" && brand == "10")  **return** "03";//如意通  **else** **if**(pay\_mode == "02" && brand == "20")  **return** "01";//新势力  **else** **if**(pay\_mode == "02" && brand == "30")  **return** "01";//世界风  **else** **if**(pay\_mode == "02" && brand == "50")  **return** "01";//世界风  **else** **if**(pay\_mode == "02" && brand == "40")  **return** "02";//新势力  **else** **if**(pay\_mode == "02" && brand == "90")  **return** "01";//世界风  **else** **if**(pay\_mode == "02" && brand == **null**)  **return** "01";//世界风  **else** **if**(pay\_mode == "03")  **return** "03";//如意通  **else** **if**(pay\_mode == "05")  **return** "03";//如意通--OCS用户  **else** **if**(pay\_mode == "04" && brand == "10")  **return** "03";//如意通  **else** **if**(pay\_mode == "04" && brand == "20")  **return** "03";//如意通  **else** **if**(pay\_mode == "04" && brand == "30")  **return** "03";//如意通  **else** **if**(pay\_mode == "04" && brand == "50")  **return** "03";//如意通  **else** **if**(pay\_mode == "04" && brand == "40")  **return** "02";//新势力  **else** **if**(pay\_mode == "04" && brand == "90")  **return** "03";//如意通  **else** **if**(pay\_mode == "04" && brand == **null**)  **return** "03";//如意通  **return** "01";  }  } |

## （已完成）F\_GW\_GROUP\_PRODUCT\_TYPE

### Oracle数据库中对应OBJECT\_ID

350396

### 作用描述

返回在产品组类型

### 函数体内容

FUNCTION F\_GW\_GROUP\_PRODUCT\_TYPE(

V\_PROD\_TYPE IN NUMBER,

V\_USER\_TYPE IN NUMBER,

V\_CUST\_TYPE IN NUMBER,

V\_GROUP\_PRODUCT\_TYPE IN VARCHAR2

)RETURN VARCHAR2 IS

/\*------------------------------------------------------------------

函数名 : F\_GW\_GROUP\_PRODUCT\_TYPE

使用参数：V\_PROD\_TYPE(单产品ID) V\_USER\_TYPE(用户类型)

生成时间：2009.8.27

编写人：徐杰

函数说明：返回在产品组类型

-------------------------------------------------------------------\*/

V\_RESULT VARCHAR2(20);

BEGIN

IF V\_PROD\_TYPE = 141729 AND V\_USER\_TYPE = 142551

THEN V\_RESULT := '99000001';

ELSIF V\_PROD\_TYPE = 141729 AND V\_USER\_TYPE <> 142551

THEN V\_RESULT := '990000011';

ELSIF V\_PROD\_TYPE IN (8, 955238) AND V\_USER\_TYPE = 900150

THEN V\_RESULT := 'A1100';

ELSIF V\_PROD\_TYPE IN (8, 955238) AND V\_USER\_TYPE = 142647

THEN V\_RESULT := '990000004';

ELSIF V\_PROD\_TYPE IN (8, 955238) AND V\_USER\_TYPE = 900412

THEN V\_RESULT := '990000020';

ELSIF V\_PROD\_TYPE IN (8, 955238) AND V\_USER\_TYPE = 142648

THEN V\_RESULT := '990000009';

ELSIF (V\_PROD\_TYPE = 8 AND V\_USER\_TYPE IN (141795, 141797)) OR

(V\_PROD\_TYPE IN (8, 955238) AND V\_USER\_TYPE = 0 AND V\_CUST\_TYPE IN (0, 4))

THEN V\_RESULT := 'B1300';

ELSIF (V\_PROD\_TYPE = 9 AND V\_USER\_TYPE = 141795) OR

(V\_PROD\_TYPE IN (9, 955237) AND V\_USER\_TYPE = 0 AND V\_CUST\_TYPE IN (0, 4))

THEN V\_RESULT := 'B1400';

ELSIF V\_PROD\_TYPE = 8 AND V\_USER\_TYPE = 900582

THEN V\_RESULT := 'B1800';

ELSIF V\_PROD\_TYPE NOT IN (8, 9, 955238, 955237, 141729)

THEN V\_RESULT := V\_GROUP\_PRODUCT\_TYPE;

ELSE V\_RESULT := '990000010';

END IF;

RETURN V\_RESULT;

END F\_GW\_GROUP\_PRODUCT\_TYPE;

### Hive中处理思路及方法

#### 思路

通过if-else代替

#### 类名

org.apache.hadoop.hive.ql.udf.FGWGroupProductType.java

#### Hive函数名

f\_gw\_group\_product\_type

#### Hive自定义函数代码

|  |
| --- |
| package org.apache.hadoop.hive.ql.udf;  import java.util.ArrayList;  import java.util.Collections;  import java.util.List;  import org.apache.hadoop.hive.ql.exec.UDF;  /\*  \* 函数功能：返回在产品组类型  \* Oracle中创建者：徐杰  \* Hive自定义函数创建：张海东  \* 创建时间：2015-06-09  \* 对应原Oracle函数：DW. F\_GW\_GROUP\_PRODUCT\_TYPE  \* 传入参数：v\_prod\_type(单产品ID) v\_user\_type(用户类型) V\_CUST\_TYPE V\_GROUP\_PRODUCT\_TYPE  \* 返回参数：分档  \* \*/  public class FGWGroupProductType extends UDF{  public String evaluate(int v\_prod\_type,int v\_user\_type,int v\_cust\_type,String v\_group\_product\_type) {  List<Integer> list = new ArrayList<Integer>(2);  list.add(8); list.add(955238);  List<Integer> listt = new ArrayList<Integer>(5);  list.add(8); list.add(9);  list.add(955238); list.add(955237); list.add(141729);  List<Integer> listtt = new ArrayList<Integer>();  list.add(0); list.add(4);  List<Integer> li = new ArrayList<Integer>();  list.add(9); list.add(955237);  List<Integer> lis = new ArrayList<Integer>();  list.add(141795); list.add(141797);  if(v\_prod\_type == 141729 && v\_user\_type == 142551)  return "99000001";  else if(v\_prod\_type == 141729 && v\_user\_type != 142551)  return "990000011";  else if(list.contains(v\_prod\_type) && v\_user\_type == 900150)  return "A1100";  else if(list.contains(v\_prod\_type) && v\_user\_type == 142647)  return "990000004";  else if(list.contains(v\_prod\_type) && v\_user\_type == 900412)  return "990000020";  else if(list.contains(v\_prod\_type) && v\_user\_type == 142648)  return "990000009";  else if(v\_prod\_type == 8 && lis.contains(v\_user\_type) || list.contains(v\_prod\_type) && v\_user\_type == 0 && listtt.contains(v\_cust\_type))  return "B1300";  else if (v\_prod\_type == 9 && v\_user\_type == 141795 || (li.contains(v\_prod\_type) && v\_user\_type == 0 && listtt.contains(v\_cust\_type)))  return "B1400";  else if(v\_prod\_type == 8 && v\_user\_type == 900582)  return "B1800";  else if(!listt.contains(v\_prod\_type))  return v\_group\_product\_type;  else  return "990000010";  }  } |

## （已完成）F\_INCOME\_LEVEL

### Oracle数据库中对应OBJECT\_ID

350400

### 作用描述

出账收入分档

### 函数体内容

FUNCTION F\_INCOME\_LEVEL(TOTAL IN NUMBER) RETURN VARCHAR2

IS

/\*-------------------------------------------------------------------------------

函数名 F\_INCOME\_LEVEL(出账收入分档)

备注：

编写人： hulei

编写时间 : 20100921

修改人:

修改时间:

修改原因:

-------------------------------------------------------------------------------\*/

RESULT VARCHAR2(2);

BEGIN

RESULT := CASE WHEN TOTAL<=0 THEN '01'

WHEN TOTAL>0 AND TOTAL<=10 THEN '02'

WHEN TOTAL>10 AND TOTAL<=20 THEN '03'

WHEN TOTAL>20 AND TOTAL<=30 THEN '04'

WHEN TOTAL>30 AND TOTAL<=40 THEN '05'

WHEN TOTAL>40 AND TOTAL<=50 THEN '06'

WHEN TOTAL>50 AND TOTAL<=60THEN '07'

WHEN TOTAL>60 AND TOTAL<=70 THEN '08'

WHEN TOTAL>70 AND TOTAL<=80 THEN '09'

WHEN TOTAL>80 AND TOTAL<=90 THEN '10'

WHEN TOTAL>90 AND TOTAL<=100 THEN '11'

WHEN TOTAL>100 AND TOTAL<=110 THEN '12'

WHEN TOTAL>110 AND TOTAL<=120 THEN '13'

WHEN TOTAL>120 AND TOTAL<=130 THEN '14'

WHEN TOTAL>130 AND TOTAL<=140 THEN '15'

WHEN TOTAL>140 AND TOTAL<=150 THEN '16'

WHEN TOTAL>150 AND TOTAL<=200 THEN '17'

WHEN TOTAL>200 AND TOTAL<=250 THEN '18'

WHEN TOTAL>250 AND TOTAL<=300 THEN '19'

WHEN TOTAL>300 AND TOTAL<=350 THEN '20'

WHEN TOTAL>350 AND TOTAL<=400 THEN '21'

WHEN TOTAL>400 AND TOTAL<=450 THEN '22'

WHEN TOTAL>450 AND TOTAL<=500 THEN '23'

WHEN TOTAL>500 AND TOTAL<=800 THEN '24'

WHEN TOTAL>800 AND TOTAL<=1000THEN '25'

WHEN TOTAL>1000 AND TOTAL<=1500 THEN '26'

WHEN TOTAL>1500 AND TOTAL<=2000 THEN '27'

WHEN TOTAL>2000 THEN '28'

ELSE '01'

END;

RETURN(RESULT);

END F\_INCOME\_LEVEL;

### Hive中处理思路及方法

#### 思路

通过if-else代替

#### 类名

org.apache.hadoop.hive.ql.udf.FIncomeLevel.java

#### Hive函数名

f\_income\_level

#### Hive自定义函数代码

|  |
| --- |
| **package** org.apache.hadoop.hive.ql.udf;  **import** org.apache.hadoop.hive.ql.exec.UDF;  /\*  \* 函数功能：出账收入分档  \* Oracle中创建者：hulei  \* Hive自定义函数创建：张海东  \* 创建时间：2015-06-09  \* 对应原Oracle函数：DW.F\_INCOME\_LEVEL  \* 传入参数：收入  \* 返回参数：分档  \* \*/  **public** **class** FIncomeLevel **extends** UDF{  **public** String evaluate(**int** total) {    **if**(total <= 0)  **return** "01";  **else** **if**(total > 0 && total <= 10)  **return** "02";  **else** **if**(total > 10 && total <= 20)  **return** "03";  **else** **if**(total > 20 && total <= 30)  **return** "04";  **else** **if**(total > 30 && total <= 40)  **return** "05";  **else** **if**(total > 40 && total <= 50)  **return** "06";  **else** **if**(total > 50 && total <= 60)  **return** "07";  **else** **if**(total > 60 && total <= 70)  **return** "08";  **else** **if**(total > 70 && total <= 80)  **return** "09";  **else** **if**(total > 80 && total <= 90)  **return** "10";  **else** **if**(total > 90 && total <= 100)  **return** "11";  **else** **if**(total > 100 && total <= 110)  **return** "12";  **else** **if**(total > 110 && total <= 120)  **return** "13";  **else** **if**(total > 120 && total <= 130)  **return** "14";  **else** **if**(total > 130 && total <= 140)  **return** "15";  **else** **if**(total > 140 && total <= 150)  **return** "16";  **else** **if**(total > 150 && total <= 200)  **return** "17";  **else** **if**(total > 200 && total <= 250)  **return** "18";  **else** **if**(total > 250 && total <= 300)  **return** "19";  **else** **if**(total > 300 && total <= 350)  **return** "20";  **else** **if**(total > 350 && total <= 400)  **return** "21";  **else** **if**(total > 400 && total <= 450)  **return** "22";  **else** **if**(total > 450 && total <= 500)  **return** "23";  **else** **if**(total > 500 && total <= 800)  **return** "24";  **else** **if**(total > 800 && total <= 1000)  **return** "25";  **else** **if**(total > 1000 && total <= 1500)  **return** "26";  **else** **if**(total > 1500 && total <= 2000)  **return** "27";  **else** **if**(total > 2000)  **return** "28";  **return** "01";  }  } |

## （已完成）F\_INNET\_MONTH\_LEVEL\_GW

### Oracle数据库中对应OBJECT\_ID

350401

### 作用描述

入网时间分段类型

### 函数体内容

FUNCTION F\_INNET\_MONTH\_LEVEL\_GW(INNET\_MONTH\_LEVEL IN NUMBER) RETURN VARCHAR2

IS

/\*-------------------------------------------------------------------------------

函数名 F\_INNET\_MONTH\_LEVEL\_GW(入网时间分段类型)

备注：

编写人：jiaoxiulian

编写时间 : 20070718

修改人:常勇

修改时间:2010-10-25

修改原因:上报总部

-------------------------------------------------------------------------------\*/

RESULT VARCHAR2(2);

BEGIN

RESULT := CASE WHEN INNET\_MONTH\_LEVEL>=0 and INNET\_MONTH\_LEVEL<=1 THEN '01'

WHEN INNET\_MONTH\_LEVEL=2 THEN '02'

WHEN INNET\_MONTH\_LEVEL=3 THEN '03'

WHEN INNET\_MONTH\_LEVEL=4 THEN '04'

WHEN INNET\_MONTH\_LEVEL=5 THEN '05'

WHEN INNET\_MONTH\_LEVEL=6 THEN '06'

WHEN INNET\_MONTH\_LEVEL=7 THEN '07'

WHEN INNET\_MONTH\_LEVEL=8 THEN '08'

WHEN INNET\_MONTH\_LEVEL=9 THEN '09'

WHEN INNET\_MONTH\_LEVEL=10 THEN '10'

WHEN INNET\_MONTH\_LEVEL=11 THEN '11'

WHEN INNET\_MONTH\_LEVEL=12 THEN '12'

WHEN INNET\_MONTH\_LEVEL=13 THEN '13'

WHEN INNET\_MONTH\_LEVEL=14 THEN '14'

WHEN INNET\_MONTH\_LEVEL=15 THEN '15'

WHEN INNET\_MONTH\_LEVEL=16 THEN '16'

WHEN INNET\_MONTH\_LEVEL=17 THEN '17'

WHEN INNET\_MONTH\_LEVEL=18 THEN '18'

WHEN INNET\_MONTH\_LEVEL=19 THEN '19'

WHEN INNET\_MONTH\_LEVEL=20 THEN '20'

WHEN INNET\_MONTH\_LEVEL=21 THEN '21'

WHEN INNET\_MONTH\_LEVEL=22 THEN '22'

WHEN INNET\_MONTH\_LEVEL=23 THEN '23'

WHEN INNET\_MONTH\_LEVEL=24 THEN '24'

WHEN INNET\_MONTH\_LEVEL>=25 AND INNET\_MONTH\_LEVEL<=36 THEN '25'

WHEN INNET\_MONTH\_LEVEL>=37 AND INNET\_MONTH\_LEVEL<=48 THEN '26'

WHEN INNET\_MONTH\_LEVEL>=49 AND INNET\_MONTH\_LEVEL<=60 THEN '27'

WHEN INNET\_MONTH\_LEVEL>=61 AND INNET\_MONTH\_LEVEL<=72 THEN '28'

WHEN INNET\_MONTH\_LEVEL>=73 AND INNET\_MONTH\_LEVEL<=84 THEN '29'

WHEN INNET\_MONTH\_LEVEL>=85 AND INNET\_MONTH\_LEVEL<=96 THEN '30'

WHEN INNET\_MONTH\_LEVEL>=97 THEN '31'

ELSE '31'

END;

RETURN(RESULT);

END F\_INNET\_MONTH\_LEVEL\_GW;

### Hive中处理思路及方法

#### 思路

通过if-else代替

#### 类名

org.apache.hadoop.hive.ql.udf.FInnetMonthLevelGW.java

#### Hive函数名

f\_innet\_month\_level\_gw

#### Hive自定义函数代码

|  |
| --- |
| **package** org.apache.hadoop.hive.ql.udf;  **import** org.apache.hadoop.hive.ql.exec.UDF;  /\*  \* 函数功能：入网时间分段类型  \* Oracle中创建者：jiaoxiulian  \* Hive自定义函数创建：张海东  \* 创建时间：2015-06-09  \* 对应原Oracle函数：DW.F\_INNET\_MONTH\_LEVEL\_GW  \* 传入参数：INNET\_MONTH\_LEVEL  \* 返回参数：分档  \* \*/  **public** **class** FInnetMonthLevelGW **extends** UDF{  **public** String evaluate(**int** innet\_momth\_level) {    **if**(innet\_momth\_level >= 0 && innet\_momth\_level <= 1)  **return** "01";  **else** **if**(innet\_momth\_level == 2)  **return** "02";  **else** **if**(innet\_momth\_level == 3)  **return** "03";  **else** **if**(innet\_momth\_level == 4)  **return** "04";  **else** **if**(innet\_momth\_level == 5)  **return** "05";  **else** **if**(innet\_momth\_level == 6)  **return** "06";  **else** **if**(innet\_momth\_level == 7)  **return** "07";  **else** **if**(innet\_momth\_level == 8)  **return** "08";  **else** **if**(innet\_momth\_level == 9)  **return** "09";  **else** **if**(innet\_momth\_level == 10)  **return** "10";  **else** **if**(innet\_momth\_level == 11)  **return** "11";  **else** **if**(innet\_momth\_level == 12)  **return** "12";  **else** **if**(innet\_momth\_level == 13)  **return** "13";  **else** **if**(innet\_momth\_level == 14)  **return** "14";  **else** **if**(innet\_momth\_level == 15)  **return** "15";  **else** **if**(innet\_momth\_level == 16)  **return** "16";  **else** **if**(innet\_momth\_level == 17)  **return** "17";  **else** **if**(innet\_momth\_level == 18)  **return** "18";  **else** **if**(innet\_momth\_level == 19)  **return** "19";  **else** **if**(innet\_momth\_level == 20)  **return** "20";  **else** **if**(innet\_momth\_level == 21)  **return** "21";  **else** **if**(innet\_momth\_level == 22)  **return** "22";  **else** **if**(innet\_momth\_level == 23)  **return** "23";  **else** **if**(innet\_momth\_level == 24)  **return** "24";  **else** **if**(innet\_momth\_level > 25 && innet\_momth\_level <= 36)  **return** "25";  **else** **if**(innet\_momth\_level > 37 && innet\_momth\_level <= 48)  **return** "26";  **else** **if**(innet\_momth\_level > 49 && innet\_momth\_level <= 60)  **return** "27";  **else** **if**(innet\_momth\_level > 61 && innet\_momth\_level <= 72)  **return** "28";  **else** **if**(innet\_momth\_level > 73 && innet\_momth\_level <= 84)  **return** "29";  **else** **if**(innet\_momth\_level > 85 && innet\_momth\_level <= 96)  **return** "30";  **else** **if**(innet\_momth\_level > 97)  **return** "31";  **return** "31";  }  } |

## （已完成）F\_GW\_JF\_LEVEL

### Oracle数据库中对应OBJECT\_ID

350399

### 作用描述

固网积分分档

### 函数体内容

FUNCTION F\_GW\_JF\_LEVEL(JF\_SCORE IN NUMBER) RETURN VARCHAR2

IS

/\*-------------------------------------------------------------------------------

函数名 F\_GW\_JF\_LEVEL(固网积分分档)

备注：

编写人： CY

编写时间 : 20101025

修改人:

修改时间:

修改原因:

-------------------------------------------------------------------------------\*/

RESULT VARCHAR2(2);

BEGIN

RESULT := CASE WHEN JF\_SCORE>=0 AND JF\_SCORE<1000 THEN '01'

WHEN JF\_SCORE>=1000 AND JF\_SCORE<2000 THEN '02'

WHEN JF\_SCORE>=2000 AND JF\_SCORE<3000 THEN '03'

WHEN JF\_SCORE>=3000 AND JF\_SCORE<4000 THEN '04'

WHEN JF\_SCORE>=4000 AND JF\_SCORE<5000 THEN '05'

WHEN JF\_SCORE>=5000 AND JF\_SCORE<6000 THEN '06'

WHEN JF\_SCORE>=6000 AND JF\_SCORE<7000 THEN '07'

WHEN JF\_SCORE>=7000 AND JF\_SCORE<8000 THEN '08'

WHEN JF\_SCORE>=8000 AND JF\_SCORE<9000 THEN '09'

WHEN JF\_SCORE>=9000 AND JF\_SCORE<10000 THEN '10'

WHEN JF\_SCORE>=10000 AND JF\_SCORE<11000 THEN '11'

WHEN JF\_SCORE>=11000 AND JF\_SCORE<12000 THEN '12'

WHEN JF\_SCORE>=12000 AND JF\_SCORE<13000 THEN '13'

WHEN JF\_SCORE>=13000 AND JF\_SCORE<14000 THEN '14'

WHEN JF\_SCORE>=14000 AND JF\_SCORE<15000 THEN '15'

WHEN JF\_SCORE>=15000 AND JF\_SCORE<16000 THEN '16'

WHEN JF\_SCORE>=16000 AND JF\_SCORE<17000 THEN '17'

WHEN JF\_SCORE>=17000 AND JF\_SCORE<18000 THEN '18'

WHEN JF\_SCORE>=18000 AND JF\_SCORE<19000 THEN '19'

WHEN JF\_SCORE>=19000 AND JF\_SCORE<20000 THEN '20'

WHEN JF\_SCORE>=20000 AND JF\_SCORE<30000 THEN '21'

WHEN JF\_SCORE>=30000 AND JF\_SCORE<40000 THEN '22'

WHEN JF\_SCORE>=40000 AND JF\_SCORE<50000 THEN '23'

WHEN JF\_SCORE>=50000 AND JF\_SCORE<60000 THEN '24'

WHEN JF\_SCORE>=60000 AND JF\_SCORE<70000 THEN '25'

WHEN JF\_SCORE>=70000 AND JF\_SCORE<80000 THEN '26'

WHEN JF\_SCORE>=80000 AND JF\_SCORE<90000 THEN '27'

WHEN JF\_SCORE>=90000 AND JF\_SCORE<100000 THEN '28'

WHEN JF\_SCORE>=100000 THEN '29'

else '01'

END;

RETURN(RESULT);

END F\_GW\_JF\_LEVEL;

### Hive中处理思路及方法

#### 思路

通过if-else代替

#### 类名

org.apache.hadoop.hive.ql.udf.FGwJfLevel.java

#### Hive函数名

f\_gw\_jf\_level

#### Hive自定义函数代码

|  |
| --- |
| **package** org.apache.hadoop.hive.ql.udf;  **import** org.apache.hadoop.hive.ql.exec.UDF;  /\*  \* 函数功能：固网积分分档  \* Oracle中创建者：CY  \* Hive自定义函数创建：张海东  \* 创建时间：2015-06-09  \* 对应原Oracle函数：DW.F\_GW\_JF\_LEVEL  \* 传入参数：积分（JF\_SCORE）  \* 返回参数：分档  \* \*/  **public** **class** FGwJfLevel **extends** UDF{    **public** String evaluate(**int** jf\_score) {    **if**(jf\_score >= 0 && jf\_score < 1000)  **return** "01";  **else** **if**(jf\_score >= 1000 && jf\_score < 2000)  **return** "02";  **else** **if**(jf\_score >= 2000 && jf\_score < 3000)  **return** "03";  **else** **if**(jf\_score >= 3000 && jf\_score < 4000)  **return** "04";  **else** **if**(jf\_score >= 4000 && jf\_score < 5000)  **return** "05";  **else** **if**(jf\_score >= 5000 && jf\_score < 6000)  **return** "06";  **else** **if**(jf\_score >= 6000 && jf\_score < 7000)  **return** "07";  **else** **if**(jf\_score >= 7000 && jf\_score < 8000)  **return** "08";  **else** **if**(jf\_score >= 8000 && jf\_score < 9000)  **return** "09";  **else** **if**(jf\_score >= 9000 && jf\_score < 10000)  **return** "10";  **else** **if**(jf\_score >= 10000 && jf\_score < 11000)  **return** "11";  **else** **if**(jf\_score >= 11000 && jf\_score < 12000)  **return** "12";  **else** **if**(jf\_score >= 12000 && jf\_score < 13000)  **return** "13";  **else** **if**(jf\_score >= 13000 && jf\_score < 14000)  **return** "14";  **else** **if**(jf\_score >= 14000 && jf\_score < 15000)  **return** "15";  **else** **if**(jf\_score >= 15000 && jf\_score < 16000)  **return** "16";  **else** **if**(jf\_score >= 16000 && jf\_score < 17000)  **return** "17";  **else** **if**(jf\_score >= 17000 && jf\_score < 18000)  **return** "18";  **else** **if**(jf\_score >= 18000 && jf\_score < 19000)  **return** "19";  **else** **if**(jf\_score >= 19000 && jf\_score < 20000)  **return** "20";  **else** **if**(jf\_score >= 20000 && jf\_score < 30000)  **return** "21";  **else** **if**(jf\_score >= 30000 && jf\_score < 40000)  **return** "22";  **else** **if**(jf\_score >= 40000 && jf\_score < 50000)  **return** "23";  **else** **if**(jf\_score >= 50000 && jf\_score < 60000)  **return** "24";  **else** **if**(jf\_score >= 60000 && jf\_score < 70000)  **return** "25";  **else** **if**(jf\_score >= 70000 && jf\_score < 80000)  **return** "26";  **else** **if**(jf\_score >= 80000 && jf\_score < 90000)  **return** "27";  **else** **if**(jf\_score >= 90000 && jf\_score < 100000)  **return** "28";  **else** **if**(jf\_score >= 100000)  **return** "29";  **return** "01";  }  } |

## （已完成）F\_GW\_INCOME\_LEVEL

### Oracle数据库中对应OBJECT\_ID

350397

### 作用描述

返回固网收入分档

### 函数体内容

FUNCTION F\_GW\_INCOME\_LEVEL(V\_FEE IN NUMBER)

RETURN VARCHAR2 IS

/\*------------------------------------------------------------------

函数名 : F\_GW\_INCOME\_LEVEL

使用参数：V\_FEE 收入

生成时间：2009.8.27

编写人：徐杰

函数说明：返回固网收入分档

-------------------------------------------------------------------\*/

V\_RESULT VARCHAR2(4);

BEGIN

V\_RESULT :=

CASE WHEN V\_FEE <= 0 THEN '01'

WHEN V\_FEE > 0 AND V\_FEE <= 5 THEN '02'

WHEN V\_FEE > 5 AND V\_FEE <= 10 THEN '03'

WHEN V\_FEE > 10 AND V\_FEE <= 15 THEN '04'

WHEN V\_FEE > 15 AND V\_FEE <= 20 THEN '05'

WHEN V\_FEE > 20 AND V\_FEE <= 25 THEN '06'

WHEN V\_FEE > 25 AND V\_FEE <= 30 THEN '07'

WHEN V\_FEE > 30 AND V\_FEE <= 35 THEN '08'

WHEN V\_FEE > 35 AND V\_FEE <= 40 THEN '09'

WHEN V\_FEE > 40 AND V\_FEE <= 45 THEN '10'

WHEN V\_FEE > 45 AND V\_FEE <= 50 THEN '11'

WHEN V\_FEE > 50 AND V\_FEE <= 55 THEN '12'

WHEN V\_FEE > 55 AND V\_FEE <= 60 THEN '13'

WHEN V\_FEE > 60 AND V\_FEE <= 65 THEN '14'

WHEN V\_FEE > 65 AND V\_FEE <= 70 THEN '15'

WHEN V\_FEE > 70 AND V\_FEE <= 80 THEN '16'

WHEN V\_FEE > 80 AND V\_FEE <= 90 THEN '17'

WHEN V\_FEE > 90 AND V\_FEE <= 100 THEN '18'

WHEN V\_FEE > 100 AND V\_FEE <= 120 THEN '19'

WHEN V\_FEE > 120 AND V\_FEE <= 140 THEN '20'

WHEN V\_FEE > 140 AND V\_FEE <= 160 THEN '21'

ELSE '22' END;

RETURN V\_RESULT;

END F\_GW\_INCOME\_LEVEL;

### Hive中处理思路及方法

#### 思路

通过if-else代替

#### 类名

org.apache.hadoop.hive.ql.udf.FGwIncomeLevel.java

#### Hive函数名

f\_gw\_income\_level

#### Hive自定义函数代码

|  |
| --- |
| **package** org.apache.hadoop.hive.ql.udf;  **import** org.apache.hadoop.hive.ql.exec.UDF;  /\*  \* 函数功能：固网收入分档  \* Oracle中创建者：徐杰  \* Hive自定义函数创建：张海东  \* 创建时间：2015-06-09  \* 对应原Oracle函数：DW.F\_GW\_INCOME\_LEVEL  \* 传入参数：收入（V\_FEE）  \* 返回参数：分档  \* \*/  **public** **class** FGwIncomeLevel **extends** UDF{  **public** String evaluate(**int** v\_fee) {  **if**(v\_fee <= 0)  **return** "01";  **else** **if**(v\_fee > 0 && v\_fee <= 5)  **return** "02";  **else** **if**(v\_fee > 5 && v\_fee <= 10)  **return** "03";  **else** **if**(v\_fee > 10 && v\_fee <= 15)  **return** "04";  **else** **if**(v\_fee > 15 && v\_fee <= 20)  **return** "05";  **else** **if**(v\_fee > 20 && v\_fee <= 25)  **return** "06";  **else** **if**(v\_fee > 25 && v\_fee <= 30)  **return** "07";  **else** **if**(v\_fee > 30 && v\_fee <= 35)  **return** "08";  **else** **if**(v\_fee > 35 && v\_fee <= 40)  **return** "09";  **else** **if**(v\_fee > 40 && v\_fee <= 45)  **return** "10";  **else** **if**(v\_fee > 45 && v\_fee <= 50)  **return** "11";  **else** **if**(v\_fee > 50 && v\_fee <= 55)  **return** "12";  **else** **if**(v\_fee > 55 && v\_fee <= 60)  **return** "13";  **else** **if**(v\_fee > 60 && v\_fee <= 65)  **return** "14";  **else** **if**(v\_fee > 65 && v\_fee <= 70)  **return** "15";  **else** **if**(v\_fee > 70 && v\_fee <= 80)  **return** "16";  **else** **if**(v\_fee > 80 && v\_fee <= 90)  **return** "17";  **else** **if**(v\_fee > 90 && v\_fee <= 100)  **return** "18";  **else** **if**(v\_fee > 100 && v\_fee <= 120)  **return** "19";  **else** **if**(v\_fee > 120 && v\_fee <= 140)  **return** "20";  **else** **if**(v\_fee > 140 && v\_fee <= 160)  **return** "21";  **return** "22";  }  } |

## （已完成）F\_GW\_INNET\_TIME\_LEVEL

### Oracle数据库中对应OBJECT\_ID

350398

### 作用描述

返回在网时长分档

### 函数体内容

FUNCTION F\_GW\_INNET\_TIME\_LEVEL(V\_TIME IN NUMBER)

RETURN NUMBER IS

/\*------------------------------------------------------------------

函数名 : DW.F\_INNET\_TIME

使用参数：V\_TIME

生成时间：2009.8.27

编写人：徐杰

函数说明：返回在网时长分档

-------------------------------------------------------------------\*/

V\_RESULT NUMBER(2);

BEGIN

IF V\_TIME <= 0 THEN

V\_RESULT := -1;

ELSIF V\_TIME > 0 AND V\_TIME <= 1 THEN

V\_RESULT := 1;

ELSIF V\_TIME > 1 AND V\_TIME <= 2 THEN

V\_RESULT := 2;

ELSIF V\_TIME > 2 AND V\_TIME <= 3 THEN

V\_RESULT := 3;

ELSIF V\_TIME > 3 AND V\_TIME <= 4 THEN

V\_RESULT := 4;

ELSIF V\_TIME > 4 AND V\_TIME <= 5 THEN

V\_RESULT := 5;

ELSIF V\_TIME > 5 AND V\_TIME <= 6 THEN

V\_RESULT := 6;

ELSIF V\_TIME > 6 AND V\_TIME <= 7 THEN

V\_RESULT := 7;

ELSIF V\_TIME > 7 AND V\_TIME <= 8 THEN

V\_RESULT := 8;

ELSIF V\_TIME > 8 AND V\_TIME <= 9 THEN

V\_RESULT := 9;

ELSIF V\_TIME > 9 AND V\_TIME <= 10 THEN

V\_RESULT := 10;

ELSIF V\_TIME > 10 AND V\_TIME <= 11 THEN

V\_RESULT := 11;

ELSIF V\_TIME > 11 AND V\_TIME <= 12 THEN

V\_RESULT := 12;

ELSIF V\_TIME > 12 AND V\_TIME <= 13 THEN

V\_RESULT := 13;

ELSIF V\_TIME > 13 AND V\_TIME <= 14 THEN

V\_RESULT := 14;

ELSIF V\_TIME > 14 AND V\_TIME <= 15 THEN

V\_RESULT := 15;

ELSIF V\_TIME > 15 AND V\_TIME <= 16 THEN

V\_RESULT := 16;

ELSIF V\_TIME > 16 AND V\_TIME <= 17 THEN

V\_RESULT := 17;

ELSIF V\_TIME > 17 AND V\_TIME <= 18 THEN

V\_RESULT := 18;

ELSIF V\_TIME > 18 AND V\_TIME <= 19 THEN

V\_RESULT := 19;

ELSIF V\_TIME > 19 AND V\_TIME <= 20 THEN

V\_RESULT := 20;

ELSIF V\_TIME > 20 AND V\_TIME <= 21 THEN

V\_RESULT := 21;

ELSIF V\_TIME > 21 AND V\_TIME <= 22 THEN

V\_RESULT := 22;

ELSIF V\_TIME > 22 AND V\_TIME <= 23 THEN

V\_RESULT := 23;

ELSIF V\_TIME > 23 AND V\_TIME <= 24 THEN

V\_RESULT := 24;

ELSE

V\_RESULT := 25;

END IF;

RETURN V\_RESULT;

END F\_GW\_INNET\_TIME\_LEVEL;

### Hive中处理思路及方法

#### 思路

通过if-else代替

#### 类名

org.apache.hadoop.hive.ql.udf.FGwInnetTimeLevel.java

#### Hive函数名

f\_gw\_innet\_time\_level

#### Hive自定义函数代码

|  |
| --- |
| **package** org.apache.hadoop.hive.ql.udf;  **import** org.apache.hadoop.hive.ql.exec.UDF;  /\*  \* 函数功能：返回在网时长分档  \* Oracle中创建者：徐杰  \* Hive自定义函数创建：张海东  \* 创建时间：2015-06-09  \* 对应原Oracle函数：DW.F\_INNET\_TIME  \* 传入参数：时长（V\_TIME）  \* 返回参数：分档  \* \*/  **public** **class** FGwInnetTimeLevel **extends** UDF{  **public** **int** evaluate(**int** v\_time) {  **if**(v\_time <= 0)  **return** -1;  **else** **if**(v\_time > 0 && v\_time <= 1)  **return** 1;  **else** **if**(v\_time > 1 && v\_time <= 2)  **return** 2;  **else** **if**(v\_time > 2 && v\_time <= 3)  **return** 3;  **else** **if**(v\_time > 3 && v\_time <= 4)  **return** 4;  **else** **if**(v\_time > 4 && v\_time <= 5)  **return** 5;  **else** **if**(v\_time > 5 && v\_time <= 6)  **return** 6;  **else** **if**(v\_time > 6 && v\_time <= 7)  **return** 7;  **else** **if**(v\_time > 7 && v\_time <= 8)  **return** 8;  **else** **if**(v\_time > 8 && v\_time <= 9)  **return** 9;  **else** **if**(v\_time > 9 && v\_time <= 10)  **return** 10;  **else** **if**(v\_time > 10 && v\_time <= 11)  **return** 11;  **else** **if**(v\_time > 11 && v\_time <= 12)  **return** 12;  **else** **if**(v\_time > 12 && v\_time <= 13)  **return** 13;  **else** **if**(v\_time > 13 && v\_time <= 14)  **return** 14;  **else** **if**(v\_time > 14 && v\_time <= 15)  **return** 15;  **else** **if**(v\_time > 15 && v\_time <= 16)  **return** 16;  **else** **if**(v\_time > 16 && v\_time <= 17)  **return** 17;  **else** **if**(v\_time > 17 && v\_time <= 18)  **return** 18;  **else** **if**(v\_time > 18 && v\_time <= 19)  **return** 19;  **else** **if**(v\_time > 19 && v\_time <= 20)  **return** 20;  **else** **if**(v\_time > 20 && v\_time <= 21)  **return** 21;  **else** **if**(v\_time > 21 && v\_time <= 22)  **return** 22;  **else** **if**(v\_time > 22 && v\_time <= 23)  **return** 23;  **else** **if**(v\_time > 23 && v\_time <= 24)  **return** 24;  **return** 25;  }  } |

## （江峰）FUN\_ONETIME\_FEE

### Oracle数据库中对应OBJECT\_ID

350386

### 作用描述（需补充）

### 函数体内容

FUNCTION FUN\_ONETIME\_FEE(PI\_ID IN NSRC.PRODUCT\_INSTANCE.ID%TYPE)

RETURN NUMBER IS

RESULT NUMBER;

BEGIN

SELECT NVL(SUM(BILL\_AMOUNT), 0)

INTO RESULT

FROM NSRC.BILL /\*SRC.BILL\_ONETIMEFEE\*/

WHERE PRODUCT\_INSTANCE\_ID = PI\_ID

AND ROOT\_TYPE = 'SO'

AND STATE <> 'L';

RETURN RESULT;

EXCEPTION

WHEN OTHERS THEN

RETURN 0;

END FUN\_ONETIME\_FEE;

### Hive中处理思路及方法（需补充）

#### 思路

通过if-else代替

#### 类名

org.apache.hadoop.hive.ql.udf.JF3\_FIntegralNumber.java

#### Hive函数名

f\_onetime\_fee

#### Hive自定义函数代码

## （江峰）FUN\_MONTH\_FEE\_CHG\_TOBEFORE

### Oracle数据库中对应OBJECT\_ID

350385

### 作用描述（需补充）

### 函数体内容

FUNCTION FUN\_MONTH\_FEE\_CHG\_TOBEFORE(I\_SO\_SUB\_ORDER\_ID SRC.SO\_SUB\_ORDER.ID%TYPE)

RETURN NUMBER IS

RESULT NUMBER(12);

I\_OPERATE\_TYPE SRC.BUSINESS\_OPERATE.OPERATE\_TYPE%TYPE;

I\_PRODUCT\_INSTANCE\_ID SRC.SO\_SUB\_ORDER.PRODUCT\_INSTANCE\_ID%TYPE;

I\_PRI\_ID SRC.SO\_SUB\_ORDER.PRE\_ID%TYPE;

BEGIN

SELECT XB.OPERATE\_TYPE, XA.PRODUCT\_INSTANCE\_ID, XA.PRE\_ID

INTO I\_OPERATE\_TYPE, I\_PRODUCT\_INSTANCE\_ID, I\_PRI\_ID

FROM NSRC.SO\_SUB\_ORDER XA, NSRC.BUSINESS\_OPERATE XB

WHERE XA.ID = I\_SO\_SUB\_ORDER\_ID

AND XA.BUSINESS\_ID = XB.BUSINESS\_ID;

--'0','新装','12','变更','2','销户','4','停机','5','复机','7','过户','9','撤单','17','改套餐'

--'0','新装','4','停机','5','复机',12','变更',17','改套餐'

IF I\_OPERATE\_TYPE IN (0, 4, 5, 12, 17) THEN

SELECT SUM(TO\_NUMBER(CASE

WHEN (XB.FACTOR\_VALUE = 'NAN') THEN

'0'

ELSE

NVL(XB.FACTOR\_VALUE, '0')

END))

INTO RESULT

FROM NSRC.SO\_SUB\_ORDER\_PRICE\_PLAN XA, NSRC.SO\_SUB\_ORDER\_PARAM\_VALUE XB

WHERE XA.ID = XB.SUB\_ORDER\_PRICE\_PLAN\_ID

AND EXISTS (SELECT 1

FROM NDIM.DIM\_GW\_PRICE\_PLAN PP

WHERE PP.PRICE\_PLAN\_ID = XA.PRICE\_PLAN\_ID

AND PP.PRICE\_PLAN\_TYPE\_CD = 2)

AND XA.DEL\_FLAG = '1'

AND XB.DEL\_FLAG = '1'

AND XA.SO\_SUB\_ORDER\_ID = I\_SO\_SUB\_ORDER\_ID;

--'2','销户','7','过户'

ELSIF I\_OPERATE\_TYPE IN (2, 7) THEN

RETURN NULL;

--'9' '撤单'

ELSIF I\_OPERATE\_TYPE IN (9) THEN

BEGIN

SELECT XB.OPERATE\_TYPE

INTO I\_OPERATE\_TYPE

FROM NSRC.SO\_SUB\_ORDER XA, NSRC.BUSINESS\_OPERATE XB

WHERE XA.ID = I\_PRI\_ID

AND XA.BUSINESS\_ID = XB.BUSINESS\_ID;

--'0','新装','12','变更','2','销户','4','停机','5','复机','7','过户','9','撤单','17','改套餐'

IF I\_OPERATE\_TYPE IN (0, 7) THEN

--'0','新装','7','过户'

RETURN NULL;

ELSIF I\_OPERATE\_TYPE IN (4, 5, 12, 17) THEN

--'4','停机','5','复机',12','变更',17','改套餐'

SELECT NVL(SUM(XA.FACTOR\_VALUE), 0)

INTO RESULT

FROM NSRC.SO\_PRD\_INST\_PRICE\_PLAN XA

WHERE XA.STATE = 'U'

AND XA.DEL\_FLAG = '1'

AND XA.SO\_SUB\_ORDER\_ID = I\_PRI\_ID;

ELSIF I\_OPERATE\_TYPE IN (2) THEN

--'2','销户'

SELECT NVL(FUN\_CYCLE\_FEE\_RATE(I\_PRODUCT\_INSTANCE\_ID), 0)

INTO RESULT

FROM DUAL;

ELSE

RETURN NULL;

END IF;

END;

ELSE

RETURN NULL;

END IF;

RETURN RESULT;

EXCEPTION

WHEN OTHERS THEN

RETURN NULL;

END FUN\_MONTH\_FEE\_CHG\_TOBEFORE;

### Hive中处理思路及方法（需补充）

## （已完成）F\_3G\_EYES\_CALL\_COUNTS\_LVL

### Oracle数据库中对应OBJECT\_ID

350390

### 作用描述

可视电话通话次数分档

### 函数体内容

FUNCTION F\_3G\_EYES\_CALL\_COUNTS\_LVL( IN NUMBER)

RETURN VARCHAR2 IS

/\*-------------------------------------------------------------------------------

函数名 : F\_STOP\_MONTH\_LEVEL(8.48 可视电话通话次数分档)

备注：

编写人：聂向春

编写时间 : 20070621

-------------------------------------------------------------------------------\*/

RESULT VARCHAR2(2);

BEGIN

IF CALL\_NUB = 1 THEN

RETURN '01';

ELSIF CALL\_NUB = 2 THEN

RETURN '02';

ELSIF CALL\_NUB = 3 THEN

RETURN '03';

ELSIF CALL\_NUB = 4 THEN

RETURN '04';

ELSIF CALL\_NUB = 5 THEN

RETURN '05';

ELSIF CALL\_NUB = 6 THEN

RETURN '06';

ELSIF CALL\_NUB > 6 AND CALL\_NUB <= 10 THEN

RETURN '07';

ELSIF CALL\_NUB > 10 AND CALL\_NUB <= 20 THEN

RETURN '08';

ELSIF CALL\_NUB > 20 AND CALL\_NUB <= 30 THEN

RETURN '09';

ELSIF CALL\_NUB > 30 AND CALL\_NUB <= 40 THEN

RETURN '10';

ELSIF CALL\_NUB > 40 AND CALL\_NUB <= 50 THEN

RETURN '11';

ELSIF CALL\_NUB > 50 AND CALL\_NUB <= 100 THEN

RETURN '12';

ELSIF CALL\_NUB > 100 AND CALL\_NUB <= 200 THEN

RETURN '13';

ELSIF CALL\_NUB > 200 AND CALL\_NUB <= 300 THEN

RETURN '14';

ELSIF CALL\_NUB > 300 AND CALL\_NUB <= 400 THEN

RETURN '15';

ELSIF CALL\_NUB > 400 AND CALL\_NUB <= 500 THEN

RETURN '16';

ELSIF CALL\_NUB > 500 AND CALL\_NUB <= 1000 THEN

RETURN '17';

ELSIF CALL\_NUB > 1000 THEN

RETURN '18';

ELSE

RETURN '01';

END IF;

RETURN(RESULT);

END F\_3G\_EYES\_CALL\_COUNTS\_LVL;

### Hive中处理思路及方法

#### 思路

通过if-else代替

#### 类名

org.apache.hadoop.hive.ql.udf.F3GEyesCallCountsLvl.java

#### Hive函数名

f\_3g\_eyes\_call\_counts\_lvl

#### Hive自定义函数代码

|  |
| --- |
| **package** org.apache.hadoop.hive.ql.udf;  **import** org.apache.hadoop.hive.ql.exec.UDF;  /\*  \* 函数功能：可视电话通话次数分档  \* Oracle中创建者：聂向春  \* Hive自定义函数创建：张海东  \* 创建时间：2015-06-09  \* 对应原Oracle函数：DW.F\_3G\_EYES\_CALL\_COUNTS\_LVL  \* 传入参数：通话次数（CALL\_NUB）  \* 返回参数：分档  \* \*/  **public** **class** F3GEyesCallCountsLvl **extends** UDF{    **public** String evaluate(**int** call\_nub) {    **if**(call\_nub == 1)  **return** "01";  **else** **if**(call\_nub == 2)  **return** "02";  **else** **if**(call\_nub == 3)  **return** "03";  **else** **if**(call\_nub == 4)  **return** "04";  **else** **if**(call\_nub == 5)  **return** "05";  **else** **if**(call\_nub == 6)  **return** "06";  **else** **if**(call\_nub > 6 && call\_nub <= 10)  **return** "07";  **else** **if**(call\_nub > 10 && call\_nub <= 20)  **return** "08";  **else** **if**(call\_nub > 20 && call\_nub <= 30)  **return** "09";  **else** **if**(call\_nub > 30 && call\_nub <= 40)  **return** "10";  **else** **if**(call\_nub > 40 && call\_nub <= 50)  **return** "11";  **else** **if**(call\_nub > 50 && call\_nub <= 100)  **return** "12";  **else** **if**(call\_nub > 100 && call\_nub <= 200)  **return** "13";  **else** **if**(call\_nub > 200 && call\_nub <= 300)  **return** "14";  **else** **if**(call\_nub > 300 && call\_nub <= 400)  **return** "15";  **else** **if**(call\_nub > 400 && call\_nub <= 500)  **return** "16";  **else** **if**(call\_nub > 500 && call\_nub <= 1000)  **return** "17";  **else** **if**(call\_nub > 1000)  **return** "18";  **return** "01";  }  } |

## （江峰）FUN\_MONTH\_FEE\_CHG\_BEFORE

### Oracle数据库中对应OBJECT\_ID

350384

### 作用描述（需补充）

### 函数体内容

FUNCTION FUN\_MONTH\_FEE\_CHG\_BEFORE(I\_SO\_SUB\_ORDER\_ID SRC.SO\_SUB\_ORDER.ID%TYPE)

RETURN NUMBER IS

RESULT INTEGER;

I\_OPERATE\_TYPE SRC.BUSINESS\_OPERATE.OPERATE\_TYPE%TYPE;

I\_PRODUCT\_INSTANCE\_ID SRC.SO\_SUB\_ORDER.PRODUCT\_INSTANCE\_ID%TYPE;

I\_PRI\_ID SRC.SO\_SUB\_ORDER.PRE\_ID%TYPE;

I\_ORDER\_APPLY\_DATE SRC.SO\_ORDER.APPLY\_DATE%TYPE;

BEGIN

SELECT XB.OPERATE\_TYPE, XA.PRODUCT\_INSTANCE\_ID, XA.PRE\_ID, XC.APPLY\_DATE

INTO I\_OPERATE\_TYPE, I\_PRODUCT\_INSTANCE\_ID, I\_PRI\_ID, I\_ORDER\_APPLY\_DATE

FROM NSRC.SO\_SUB\_ORDER XA, NSRC.BUSINESS\_OPERATE XB, NSRC.SO\_ORDER XC

WHERE XA.ID = I\_SO\_SUB\_ORDER\_ID

AND XA.BUSINESS\_ID = XB.BUSINESS\_ID

AND XA.SO\_ORDER\_ID = XC.ID;

--'0','新装','12','变更','2','销户','4','停机','5','复机','7','过户','9','撤单','17','改套餐'

IF I\_OPERATE\_TYPE IN (0, 7) THEN

--'0','新装','7','过户'

RETURN NULL;

ELSIF I\_OPERATE\_TYPE IN (4, 5, 12, 17) THEN

--'4','停机','5','复机',12','变更',17','改套餐'

SELECT SUM(TO\_NUMBER(XA.FACTOR\_VALUE))

INTO RESULT

FROM NSRC.SO\_PRD\_INST\_PRICE\_PLAN XA

WHERE XA.SO\_SUB\_ORDER\_ID = I\_SO\_SUB\_ORDER\_ID

AND XA.STATE = 'U'

AND XA.BEGIN\_DATE < = I\_ORDER\_APPLY\_DATE

AND (XA.END\_DATE IS NULL OR XA.END\_DATE >= I\_ORDER\_APPLY\_DATE)

AND XA.DEL\_FLAG = '1';

ELSIF I\_OPERATE\_TYPE IN (2) THEN

--'2','销户'

SELECT FUN\_CYCLE\_FEE\_RATE(I\_PRODUCT\_INSTANCE\_ID)

INTO RESULT

FROM DUAL;

ELSIF I\_OPERATE\_TYPE IN (9) THEN

--'9' '撤单'

BEGIN

SELECT XB.OPERATE\_TYPE

INTO I\_OPERATE\_TYPE

FROM NSRC.SO\_SUB\_ORDER XA, NSRC.BUSINESS\_OPERATE XB

WHERE XA.ID = I\_PRI\_ID

AND XA.BUSINESS\_ID = XB.BUSINESS\_ID;

--'0','新装','12','变更','2','销户','4','停机','5','复机','7','过户','9','撤单','17','改套餐'

IF I\_OPERATE\_TYPE IN (0, 4, 5, 12, 17) THEN

--'0','新装','4','停机','5','复机',12','变更',17','改套餐'

SELECT NVL(SUM(TO\_NUMBER(CASE

WHEN (XB.FACTOR\_VALUE = 'NAN') THEN

'0'

ELSE

NVL(XB.FACTOR\_VALUE, '0')

END)),

0)

INTO RESULT

FROM NSRC.SO\_SUB\_ORDER\_PRICE\_PLAN XA, NSRC.SO\_SUB\_ORDER\_PARAM\_VALUE XB

WHERE XA.ID = XB.SUB\_ORDER\_PRICE\_PLAN\_ID

AND EXISTS (SELECT 1

FROM NDIM.DIM\_GW\_PRICE\_PLAN PP

WHERE PP.PRICE\_PLAN\_ID = XA.PRICE\_PLAN\_ID

AND PP.PRICE\_PLAN\_TYPE\_CD = 2)

AND XA.DEL\_FLAG = '1'

AND XB.DEL\_FLAG = '1'

AND XA.SO\_SUB\_ORDER\_ID = I\_PRI\_ID;

ELSIF I\_OPERATE\_TYPE IN (2, 7) THEN

--'2','销户','7','过户'

RETURN NULL;

ELSE

RETURN NULL;

END IF;

END;

ELSE

RETURN NULL;

END IF;

RETURN RESULT;

EXCEPTION

WHEN OTHERS THEN

RETURN NULL;

END FUN\_MONTH\_FEE\_CHG\_BEFORE;

### Hive中处理思路及方法（需补充）

## （已完成）F\_3G\_CALL\_PRE\_LVL

### Oracle数据库中对应OBJECT\_ID

350389

### 作用描述

通话时长分档

### 函数体内容

FUNCTION F\_3G\_CALL\_PRE\_LVL(SEE\_CALL\_TIME IN NUMBER,

CALL\_TIME IN NUMBER)

RETURN VARCHAR2 IS

/\*-------------------------------------------------------------------------------

函数名 : 8.9 通话时长分档

备注：

编写人：niexiangchun

编写时间 : 20070621

-------------------------------------------------------------------------------\*/

BEGIN

IF CALL\_TIME <= 0 THEN RETURN ('01');

ELSIF 0 <(SEE\_CALL\_TIME / CALL\_TIME) AND(SEE\_CALL\_TIME / CALL\_TIME) <= 0.1 THEN

RETURN '01';

ELSIF 0.1 < (SEE\_CALL\_TIME / CALL\_TIME) AND

(SEE\_CALL\_TIME / CALL\_TIME) <= 0.2 THEN

RETURN '02';

ELSIF 0.2 < (SEE\_CALL\_TIME / CALL\_TIME) AND

(SEE\_CALL\_TIME / CALL\_TIME) <= 0.3 THEN

RETURN '03';

ELSIF 0.3 < (SEE\_CALL\_TIME / CALL\_TIME) AND

(SEE\_CALL\_TIME / CALL\_TIME) <= 0.4 THEN

RETURN '04';

ELSIF 0.4 < (SEE\_CALL\_TIME / CALL\_TIME) AND

(SEE\_CALL\_TIME / CALL\_TIME) <= 0.5 THEN

RETURN '05';

ELSIF 0.5 < (SEE\_CALL\_TIME / CALL\_TIME) AND

(SEE\_CALL\_TIME / CALL\_TIME) <= 0.6 THEN

RETURN '06';

ELSIF 0.6 < (SEE\_CALL\_TIME / CALL\_TIME) AND

(SEE\_CALL\_TIME / CALL\_TIME) <= 0.7 THEN

RETURN '07';

ELSIF 0.7 < (SEE\_CALL\_TIME / CALL\_TIME) AND

(SEE\_CALL\_TIME / CALL\_TIME) <= 0.8 THEN

RETURN '08';

ELSIF 0.8 < (SEE\_CALL\_TIME / CALL\_TIME) AND

(SEE\_CALL\_TIME / CALL\_TIME) <= 0.9 THEN

RETURN '09';

ELSIF 0.9 < (SEE\_CALL\_TIME / CALL\_TIME) AND

(SEE\_CALL\_TIME / CALL\_TIME) <= 1 THEN

RETURN '10';

ELSE RETURN '01';

END IF;

END F\_3G\_CALL\_PRE\_LVL;

### Hive中处理思路及方法（需补充）

#### 思路

通过if-else代替

#### 类名

org.apache.hadoop.hive.ql.udf.F3GCallPreLvl.java

#### Hive函数名

f\_3g\_call\_pre\_lvl

#### Hive自定义函数代码

|  |
| --- |
| **package** org.apache.hadoop.hive.ql.udf;  **import** org.apache.hadoop.hive.ql.exec.UDF;  /\*  \* 函数功能：通话时长分档  \* Oracle中创建者：niexiangchun  \* Hive自定义函数创建：张海东  \* 创建时间：2015-06-09  \* 对应原Oracle函数：DW. F\_3G\_CALL\_PRE\_LVL  \* 传入参数：call\_time,see\_call\_time  \* 返回参数：分档  \* \*/  **public** **class** F3GCallPreLvl **extends** UDF{    **public** String evaluate(**int** see\_call\_time,**int** call\_time) {    **int** tmp = see\_call\_time/call\_time;  **if**(tmp <= 0)  **return** "01";  **else** **if**(tmp > 0 && tmp <= 0.1)  **return** "01";  **else** **if**(tmp > 0.1 && tmp <= 0.2)  **return** "02";  **else** **if**(tmp > 0.2 && tmp <= 0.3)  **return** "03";  **else** **if**(tmp > 0.3 && tmp <= 0.4)  **return** "04";  **else** **if**(tmp > 0.4 && tmp <= 0.5)  **return** "05";  **else** **if**(tmp > 0.5 && tmp <= 0.6)  **return** "06";  **else** **if**(tmp > 0.6 && tmp <= 0.7)  **return** "07";  **else** **if**(tmp > 0.7 && tmp <= 0.8)  **return** "08";  **else** **if**(tmp > 0.8 && tmp <= 0.9)  **return** "09";  **else** **if**(tmp > 0.9 && tmp <= 1)  **return** "10";  **return** "01";  }  } |

## （已完成）F\_3G\_CALL\_LONG\_LVL

### Oracle数据库中对应OBJECT\_ID

350388

### 作用描述

通话时长分档

### 函数体内容

FUNCTION F\_3G\_CALL\_LONG\_LVL(CALL\_TIME IN NUMBER)

RETURN VARCHAR2 IS

/\*-------------------------------------------------------------------------------

函数名 : 8.9 通话时长分档

备注： MOFIFY : YANGBING 2010-01-14

编写人：jiaoxiulian

编写时间 : 20070621

-------------------------------------------------------------------------------\*/

RESULT VARCHAR2(2);

BEGIN

IF CALL\_TIME > 0 AND CALL\_TIME <= 1 THEN

RETURN '01';

ELSIF CALL\_TIME > 1 AND CALL\_TIME <= 2 THEN

RETURN '02';

ELSIF CALL\_TIME > 2 AND CALL\_TIME <= 3 THEN

RETURN '03';

ELSIF CALL\_TIME > 3 AND CALL\_TIME <= 4 THEN

RETURN '04';

ELSIF CALL\_TIME > 4 AND CALL\_TIME <= 5 THEN

RETURN '05';

ELSIF CALL\_TIME > 5 AND CALL\_TIME <= 10 THEN

RETURN '06';

ELSIF CALL\_TIME > 10 AND CALL\_TIME <= 20 THEN

RETURN '07';

ELSIF CALL\_TIME > 20 AND CALL\_TIME <= 30 THEN

RETURN '08';

ELSIF CALL\_TIME > 30 AND CALL\_TIME <= 40 THEN

RETURN '09';

ELSIF CALL\_TIME > 40 AND CALL\_TIME <= 50 THEN

RETURN '10';

ELSIF CALL\_TIME > 50 AND CALL\_TIME <= 100 THEN

RETURN '11';

ELSIF CALL\_TIME > 100 AND CALL\_TIME <= 200 THEN

RETURN '12';

ELSIF CALL\_TIME > 200 AND CALL\_TIME <= 250 THEN

RETURN '13';

ELSIF CALL\_TIME > 250 AND CALL\_TIME <= 300 THEN

RETURN '14';

ELSIF CALL\_TIME > 300 AND CALL\_TIME <= 350 THEN

RETURN '15';

ELSIF CALL\_TIME > 350 AND CALL\_TIME <= 400 THEN

RETURN '16';

ELSIF CALL\_TIME > 400 AND CALL\_TIME <= 500 THEN

RETURN '17';

ELSIF CALL\_TIME > 500 AND CALL\_TIME <= 1000 THEN

RETURN '18';

ELSIF CALL\_TIME > 1000 THEN

RETURN '19';

ELSE

RETURN '01';---99

END IF;

RETURN(RESULT);

RETURN(RESULT);

END F\_3G\_CALL\_LONG\_LVL;

### Hive中处理思路及方法

#### 思路

通过if-else代替

#### 类名

org.apache.hadoop.hive.ql.udf.F3GCallLongLvl.java

#### Hive函数名

f\_3g\_call\_long\_lvl

#### Hive自定义函数代码

|  |
| --- |
| **package** org.apache.hadoop.hive.ql.udf;  **import** org.apache.hadoop.hive.ql.exec.UDF;  /\*  \* 函数功能：通话时长分档  \* Oracle中创建者：jiaoxiulian  \* Hive自定义函数创建：张海东  \* 创建时间：2015-06-09  \* 对应原Oracle函数：DW. F\_3G\_CALL\_LONG\_LVL  \* 传入参数：时长（CALL\_TIME）  \* 返回参数：分档  \* \*/  **public** **class** F3GCallLongLvl **extends** UDF{    **public** String evaluate(**int** call\_time) {  **if**(call\_time > 0 && call\_time <= 1)  **return** "01";  **else** **if**(call\_time > 1 && call\_time <= 2)  **return** "02";  **else** **if**(call\_time > 2 && call\_time <= 3)  **return** "03";  **else** **if**(call\_time > 3 && call\_time <= 4)  **return** "04";  **else** **if**(call\_time > 4 && call\_time <= 5)  **return** "05";  **else** **if**(call\_time > 5 && call\_time <= 10)  **return** "06";  **else** **if**(call\_time > 10 && call\_time <= 20)  **return** "07";  **else** **if**(call\_time > 20 && call\_time <= 30)  **return** "08";  **else** **if**(call\_time > 30 && call\_time <= 40)  **return** "09";  **else** **if**(call\_time > 40 && call\_time <= 50)  **return** "10";  **else** **if**(call\_time > 50 && call\_time <= 100)  **return** "11";  **else** **if**(call\_time > 100 && call\_time <= 200)  **return** "12";  **else** **if**(call\_time > 200 && call\_time <= 250)  **return** "13";  **else** **if**(call\_time > 250 && call\_time <= 300)  **return** "14";  **else** **if**(call\_time > 300 && call\_time <= 350)  **return** "15";  **else** **if**(call\_time > 350 && call\_time <= 400)  **return** "16";  **else** **if**(call\_time > 400 && call\_time <= 500)  **return** "17";  **else** **if**(call\_time > 500 && call\_time <= 1000)  **return** "18";  **else** **if**(call\_time > 1000)  **return** "19";  **return** "01";  }  } |

## （游标类）GETOLDATTRBYSUBIDATTRCODE

### Oracle数据库中对应OBJECT\_ID

350412

### 作用描述（需补充）

### 函数体内容

function getOldAttrBySubIdAttrCode(s\_id number,

attrCode VARCHAR2)

return varchar2 is

Result varchar2(4000);

c\_name varchar2(4000);

i number := 0;

cursor c\_t is

select xa.attr\_value

from src.fl\_so\_attr\_instance xa, src.fl\_product\_item xb

where xa.so\_sub\_order\_id = s\_id

and xa.attr\_id = xb.id

and xb.code = attrCode

and xb.state = 'U'

and xb.del\_flag = '1'

and rownum = 1;

begin

open c\_t;

loop

fetch c\_t

into c\_name;

exit when c\_t%notfound;

if i > 0 then

Result := Result || ';' || c\_name;

else

Result := c\_name;

end if;

i := i + 1;

end loop;

close c\_t;

return Result;

end getOldAttrBySubIdAttrCode;

### Hive中处理思路及方法（需补充）

## （游标类）GETMONTHFEECHGTOBE

### Oracle数据库中对应OBJECT\_ID

350411

### 作用描述（需补充）

### 函数体内容

function getMonthFeeChgToBe(i\_so\_sub\_order\_id Nsrc.so\_sub\_order.id%type)

return number is

Result number(12);

i\_operate\_type Nsrc.business\_operate.operate\_type%type;

i\_product\_instance\_id Nsrc.so\_sub\_order.product\_instance\_id%type;

i\_pri\_id Nsrc.so\_sub\_order.pre\_id%type;

begin

select xb.operate\_type, xa.product\_instance\_id, xa.pre\_id

into i\_operate\_type, i\_product\_instance\_id, i\_pri\_id

from Nsrc.so\_sub\_order xa, Nsrc.business\_operate xb

where xa.id = i\_so\_sub\_order\_id

and xa.business\_id = xb.business\_id;

--'0','新装','12','变更','2','销户','4','停机','5','复机','7','过户','9','撤单','17','改套餐'

--'0','新装','4','停机','5','复机',12','变更',17','改套餐' '6'变更

if i\_operate\_type in (0, 4, 5,6, 12, 17,19) then

select sum(to\_number(case

when (xb.factor\_value = 'NaN') then

'0'

else

nvl(xb.factor\_value, '0')

end))

into Result

from Nsrc.so\_sub\_order\_price\_plan xa, Nsrc.so\_sub\_order\_param\_value xb

where xa.id = xb.sub\_order\_price\_plan\_id

and exists (select 1

from src.DIM\_GW\_price\_plan pp

where pp.price\_plan\_id = xa.price\_plan\_id

and pp.price\_plan\_type\_cd = 2)

and xa.del\_flag = '1'

and xb.del\_flag = '1'

and xa.so\_sub\_order\_id = i\_so\_sub\_order\_id;

--'2','销户','7','过户'

elsif i\_operate\_type in (2, 7) then

return null;

--'9' '撤单'

elsif i\_operate\_type in (9) then

begin

select xb.operate\_type

into i\_operate\_type

from Nsrc.so\_sub\_order xa, Nsrc.business\_operate xb

where xa.id = i\_pri\_id

and xa.business\_id = xb.business\_id;

--'0','新装','12','变更','2','销户','4','停机','5','复机','7','过户','9','撤单','17','改套餐' '6' 变更

if i\_operate\_type in (0, 7) then

--'0','新装','7','过户'

return null;

elsif i\_operate\_type in (4, 5,6,12, 17,19) then

--'4','停机','5','复机', '6' 变更 12','变更',17','改套餐'

select nvl(sum(xa.factor\_value), 0)

into Result

from NSRC.SO\_PRD\_INST\_PRICE\_PLAN xa

where xa.state = 'U'

and xa.del\_flag = '1'

and xa.so\_sub\_order\_id = i\_pri\_id;

elsif i\_operate\_type in (2) then

--'2','销户'

select nvl(GetAllCycleFeeRateByPi(i\_product\_instance\_id), 0)

into Result

from dual;

else

return null;

end if;

end;

else

return null;

end if;

return Result;

exception

when others then

return null;

end getMonthFeeChgToBe;

### Hive中处理思路及方法（需补充）

## （游标类）GETMONTHFEECHGBEFORE

### Oracle数据库中对应OBJECT\_ID

350410

### 作用描述（需补充）

### 函数体内容

FUNCTION GETMONTHFEECHGBEFORE(I\_SO\_SUB\_ORDER\_ID NSRC.SO\_SUB\_ORDER.ID%TYPE)

RETURN NUMBER IS

RESULT INTEGER;

I\_OPERATE\_TYPE NSRC.BUSINESS\_OPERATE.OPERATE\_TYPE%TYPE;

I\_PRODUCT\_INSTANCE\_ID NSRC.SO\_SUB\_ORDER.PRODUCT\_INSTANCE\_ID%TYPE;

I\_PRI\_ID NSRC.SO\_SUB\_ORDER.PRE\_ID%TYPE;

I\_ORDER\_APPLY\_DATE NSRC.SO\_ORDER.APPLY\_DATE%TYPE;

BEGIN

SELECT XB.OPERATE\_TYPE, XA.PRODUCT\_INSTANCE\_ID, XA.PRE\_ID, XC.APPLY\_DATE

INTO I\_OPERATE\_TYPE,

I\_PRODUCT\_INSTANCE\_ID,

I\_PRI\_ID,

I\_ORDER\_APPLY\_DATE

FROM NSRC.SO\_SUB\_ORDER XA, NSRC.BUSINESS\_OPERATE XB, NSRC.SO\_ORDER XC

WHERE XA.ID = I\_SO\_SUB\_ORDER\_ID

AND XA.BUSINESS\_ID = XB.BUSINESS\_ID

AND XA.SO\_ORDER\_ID = XC.ID;

--'0','新装','12','变更','2','销户','4','停机','5','复机', '6'变更 '7','过户','9','撤单','17','改套餐'

IF I\_OPERATE\_TYPE IN (0, 7) THEN

--'0','新装','7','过户'

RETURN NULL;

ELSIF I\_OPERATE\_TYPE IN (4, 5,6, 12, 17,19) THEN

--'4','停机','5','复机', '6'变更, 12','变更',17','改套餐'

SELECT SUM(TO\_NUMBER(XA.FACTOR\_VALUE))

INTO RESULT

FROM NSRC.SO\_PRD\_INST\_PRICE\_PLAN XA

WHERE XA.SO\_SUB\_ORDER\_ID = I\_SO\_SUB\_ORDER\_ID

AND XA.STATE = 'U'

AND XA.BEGIN\_DATE < = I\_ORDER\_APPLY\_DATE

AND (XA.END\_DATE IS NULL OR XA.END\_DATE >= I\_ORDER\_APPLY\_DATE)

AND XA.DEL\_FLAG = '1';

ELSIF I\_OPERATE\_TYPE IN (2) THEN

--'2','销户'

SELECT GETALLCYCLEFEERATEBYPI(I\_PRODUCT\_INSTANCE\_ID)

INTO RESULT

FROM DUAL;

ELSIF I\_OPERATE\_TYPE IN (9) THEN

--'9' '撤单'

BEGIN

SELECT XB.OPERATE\_TYPE

INTO I\_OPERATE\_TYPE

FROM NSRC.SO\_SUB\_ORDER XA, NSRC.BUSINESS\_OPERATE XB

WHERE XA.ID = I\_PRI\_ID

AND XA.BUSINESS\_ID = XB.BUSINESS\_ID;

--'0','新装','12','变更','2','销户','4','停机','5','复机','6'变更 '7','过户','9','撤单','17','改套餐'

IF I\_OPERATE\_TYPE IN (0, 4, 5, 6, 12, 17,19) THEN

--'0','新装','4','停机','5','复机', '6'变更 12','变更',17','改套餐'

SELECT NVL(SUM(TO\_NUMBER(CASE

WHEN (XB.FACTOR\_VALUE = 'NaN') THEN

'0'

ELSE

NVL(XB.FACTOR\_VALUE, '0')

END)),

0)

INTO RESULT

FROM NSRC.SO\_SUB\_ORDER\_PRICE\_PLAN XA,

NSRC.SO\_SUB\_ORDER\_PARAM\_VALUE XB

WHERE XA.ID = XB.SUB\_ORDER\_PRICE\_PLAN\_ID

AND EXISTS (SELECT 1

FROM SRC.DIM\_GW\_PRICE\_PLAN PP

WHERE PP.PRICE\_PLAN\_ID = XA.PRICE\_PLAN\_ID

AND PP.PRICE\_PLAN\_TYPE\_CD = 2)

AND XA.DEL\_FLAG = '1'

AND XB.DEL\_FLAG = '1'

AND XA.SO\_SUB\_ORDER\_ID = I\_PRI\_ID;

ELSIF I\_OPERATE\_TYPE IN (2, 7) THEN

--'2','销户','7','过户'

RETURN NULL;

ELSE

RETURN NULL;

END IF;

END;

ELSE

RETURN NULL;

END IF;

RETURN RESULT;

EXCEPTION

WHEN OTHERS THEN

RETURN NULL;

END GETMONTHFEECHGBEFORE;

### Hive中处理思路及方法（需补充）

## （游标类）GETOLDSERVICEBYSUBIDSERCODE

### Oracle数据库中对应OBJECT\_ID

350413

### 作用描述（需补充）

### 函数体内容

function getOldServiceBySubIdSerCode(s\_id number,

serviceCode VARCHAR2)

return varchar2 is

Result varchar2(4000);

c\_name varchar2(4000);

i number := 0;

cursor c\_t is

select xb.name

from NSRC.so\_service\_instance xa,

NSRC.product\_item xb,

NSRC.product\_item\_kind\_element xc,

NSRC.product\_item\_kind xd

where xa.so\_sub\_order\_id = s\_id

and xa.state != 'E'

and xa.del\_flag = '1'

and xa.service\_id = xb.id

and xb.state = 'U'

and xb.code = serviceCode

and rownum = 1;

begin

open c\_t;

loop

fetch c\_t

into c\_name;

exit when c\_t%notfound;

if i > 0 then

Result := Result || ';' || c\_name;

else

Result := c\_name;

end if;

i := i + 1;

end loop;

close c\_t;

return Result;

end getOldServiceBySubIdSerCode;

### Hive中处理思路及方法（需补充）

## （游标类）GETSERVICEBYPIIDSERCODE

### Oracle数据库中对应OBJECT\_ID

350416

### 作用描述（需补充）

### 函数体内容

function getServiceByPIIdSerCode(s\_id number,

serviceCode VARCHAR2)

return varchar2 is

Result varchar2(4000);

c\_name varchar2(4000);

i number := 0;

cursor c\_t is

select xb.name

from nSRC.service\_instance xa, nSRC.product\_item xb

where xa.product\_instance\_id = s\_id

and xa.state != 'E'

and xa.del\_flag = '1'

and xa.service\_id = xb.id

and xb.state = 'U'

and xb.code = serviceCode

and rownum = 1;

begin

open c\_t;

loop

fetch c\_t

into c\_name;

exit when c\_t%notfound;

if i > 0 then

Result := Result || ';' || c\_name;

else

Result := c\_name;

end if;

i := i + 1;

end loop;

close c\_t;

return Result;

end getServiceByPIIdSerCode;

### Hive中处理思路及方法（需补充）

## （游标类）GETONETIMEFEEBYSUBORDER

### Oracle数据库中对应OBJECT\_ID

350415

### 作用描述（需补充）

### 函数体内容

FUNCTION GETONETIMEFEEBYSUBORDER(S\_ID IN NSRC.SO\_SUB\_ORDER\_PRICE\_PLAN.SO\_SUB\_ORDER\_ID%TYPE)

RETURN NUMBER IS

RESULT NUMBER(12) := 0;

CHARGE NUMBER(12) := 0;

CURSOR C\_SO\_PRICE\_PLAN IS

SELECT \*

FROM NSRC.SO\_SUB\_ORDER\_PRICE\_PLAN

WHERE SO\_SUB\_ORDER\_ID = S\_ID

AND DEL\_FLAG = '1';

R\_SO\_SUB NSRC.SO\_SUB\_ORDER\_PRICE\_PLAN%ROWTYPE;

R\_PRICE\_PLAN SRC.DIM\_GW\_PRICE\_PLAN%ROWTYPE;

R\_SO\_SUB\_PARAM\_VALUE NSRC.SO\_SUB\_ORDER\_PARAM\_VALUE%ROWTYPE;

BEGIN

RESULT := 0;

BEGIN

/\*select sum(b.price) into Result

from price\_plan\_price\_item a,so\_price b

where a.price\_item\_id = b.so\_price\_id

and a.state = 'U'

and a.del\_flg = '1'

and b.state = 'U'

and b.del\_flag = '1'

and a.price\_plan\_id in(

select price\_plan\_id from price\_plan where price\_plan\_type\_cd = 9

and state = 'U' and del\_flag = '1'

and price\_plan\_id in(

select price\_plan\_id from so\_sub\_order\_price\_plan

where so\_sub\_order\_id = s\_id

and del\_flag = '1'

)

);\*/

SELECT SUM(XA.BILL\_AMOUNT)

INTO RESULT

FROM NSRC.BILL XA

WHERE XA.ROOT\_INDEX IN

(

SELECT SO\_ORDER\_ID FROM NSRC.SO\_SUB\_ORDER WHERE ID = S\_ID

)

AND XA.ROOT\_TYPE = 'SO'

AND XA.STATE <> 'L';

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RESULT := 0;

END;

IF RESULT = 0 OR RESULT IS NULL THEN

-- 2007-08-21增加or result is null 撤单一次性费用

RESULT := 0;

OPEN C\_SO\_PRICE\_PLAN;

LOOP

<<XXX>>

FETCH C\_SO\_PRICE\_PLAN

INTO R\_SO\_SUB;

EXIT WHEN C\_SO\_PRICE\_PLAN%NOTFOUND;

SELECT \*

INTO R\_PRICE\_PLAN

FROM SRC.DIM\_GW\_PRICE\_PLAN

WHERE PRICE\_PLAN\_ID = R\_SO\_SUB.PRICE\_PLAN\_ID

AND DEL\_FLAG = '1'

AND STATE = 'U';

IF R\_PRICE\_PLAN.PRICE\_PLAN\_TYPE\_CD <> 9 THEN

CHARGE := 0;

GOTO XXX;

END IF;

BEGIN

SELECT \*

INTO R\_SO\_SUB\_PARAM\_VALUE

FROM NSRC.SO\_SUB\_ORDER\_PARAM\_VALUE

WHERE SUB\_ORDER\_PRICE\_PLAN\_ID = R\_SO\_SUB.ID;

CHARGE := R\_SO\_SUB\_PARAM\_VALUE.FACTOR\_VALUE;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

CHARGE := 0;

END;

RESULT := RESULT + CHARGE;

END LOOP;

CLOSE C\_SO\_PRICE\_PLAN;

END IF;

RETURN RESULT;

EXCEPTION

WHEN OTHERS THEN

RETURN 0;

END GETONETIMEFEEBYSUBORDER;

### Hive中处理思路及方法（需补充）

## （游标类）GETOLDSERVICEBYSUBIDSERKIND

### Oracle数据库中对应OBJECT\_ID

350414

### 作用描述（需补充）

### 函数体内容

function getOldServiceBySubIdSerKind(s\_id number,

serviceItemCode VARCHAR2)

return varchar2 is

Result varchar2(4000);

c\_name varchar2(4000);

i number := 0;

cursor c\_t is

select xb.name

from nsrc.so\_service\_instance xa,

nsrc.product\_item xb,

nsrc.product\_item\_kind\_element xc,

nsrc.product\_item\_kind xd

where xa.so\_sub\_order\_id = s\_id

and xa.state != 'E'

and xa.del\_flag = '1'

and xa.service\_id = xb.id

and xb.state = 'U'

and xb.id = xc.product\_item\_id

and xc.del\_flag = '1'

and xc.product\_item\_kind\_id = xd.id

and xd.item\_kind\_code = serviceItemCode

and xd.del\_flag = '1'

and rownum = 1;

begin

open c\_t;

loop

fetch c\_t

into c\_name;

exit when c\_t%notfound;

if i > 0 then

Result := Result || ';' || c\_name;

else

Result := c\_name;

end if;

i := i + 1;

end loop;

close c\_t;

return Result;

end getOldServiceBySubIdSerKind;

### Hive中处理思路及方法（需补充）

## （游标类）GETALLCYCLEFEERATE

### Oracle数据库中对应OBJECT\_ID

350405

### 作用描述（需补充）

### 函数体内容

FUNCTION GETALLCYCLEFEERATE(S\_ID IN NSRC.SO\_SUB\_ORDER\_PRICE\_PLAN.SO\_SUB\_ORDER\_ID%TYPE)

RETURN NUMBER IS

RESULT NUMBER(12) := 0;

CHARGE NUMBER(12) := 0;

CURSOR C\_SO\_PRICE\_PLAN IS

SELECT \*

FROM NSRC.SO\_SUB\_ORDER\_PRICE\_PLAN

WHERE SO\_SUB\_ORDER\_ID = S\_ID

AND DEL\_FLAG = '1';

R\_SO\_SUB NSRC.SO\_SUB\_ORDER\_PRICE\_PLAN%ROWTYPE;

R\_PRICE\_PLAN SRC.DIM\_GW\_PRICE\_PLAN%ROWTYPE;

R\_SO\_SUB\_PARAM\_VALUE NSRC.SO\_SUB\_ORDER\_PARAM\_VALUE%ROWTYPE;

BEGIN

RESULT := 0;

OPEN C\_SO\_PRICE\_PLAN;

LOOP

<<XXX>>

FETCH C\_SO\_PRICE\_PLAN

INTO R\_SO\_SUB;

EXIT WHEN C\_SO\_PRICE\_PLAN%NOTFOUND;

SELECT \*

INTO R\_PRICE\_PLAN

FROM SRC.DIM\_GW\_PRICE\_PLAN

WHERE PRICE\_PLAN\_ID = R\_SO\_SUB.PRICE\_PLAN\_ID

AND DEL\_FLAG = '1'

AND STATE = 'U';

IF R\_PRICE\_PLAN.PRICE\_PLAN\_TYPE\_CD <> 2 THEN

CHARGE := 0;

GOTO XXX;

END IF;

BEGIN

SELECT \*

INTO R\_SO\_SUB\_PARAM\_VALUE

FROM NSRC.SO\_SUB\_ORDER\_PARAM\_VALUE

WHERE SUB\_ORDER\_PRICE\_PLAN\_ID = R\_SO\_SUB.ID;

BEGIN

CHARGE := TO\_NUMBER(NVL(R\_SO\_SUB\_PARAM\_VALUE.FACTOR\_VALUE, '0'));

EXCEPTION

WHEN OTHERS THEN

CHARGE := 0;

END;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

BEGIN

SELECT A.RATE

INTO CHARGE

FROM SRC.DIM\_GW\_SUIT A, SRC.DIM\_GW\_PRICE\_PLAN\_PRICE\_ITEM B

WHERE A.SUIT\_ID = B.PRICE\_ITEM\_ID

AND B.PRICE\_PLAN\_ID = R\_SO\_SUB.PRICE\_PLAN\_ID

AND A.STATE = 'U'

AND B.STATE = 'U'

AND B.DEL\_FLG = '1';

EXCEPTION

WHEN OTHERS THEN

CHARGE := 0;

GOTO XXX;

END;

END;

RESULT := RESULT + CHARGE;

END LOOP;

CLOSE C\_SO\_PRICE\_PLAN;

RETURN RESULT;

EXCEPTION

WHEN OTHERS THEN

RETURN 0;

END GETALLCYCLEFEERATE;

### Hive中处理思路及方法（需补充）

## （游标类）GETALLCYCLEFEERATEBYPI

### Oracle数据库中对应OBJECT\_ID

350406

### 作用描述（需补充）

### 函数体内容

function GetAllCycleFeeRateByPi(pi\_id in Nsrc.product\_instance.id%type)

return number is

Result number(12) := 0;

charge number(12) := 0;

r\_pi Nsrc.product\_instance%rowtype;

s\_date Nsrc.product\_instance.product\_inst\_end\_dt%type;

priceplan\_id src.dim\_gw\_price\_plan.price\_plan\_id%type;

r\_price\_plan src.DIM\_GW\_PRICE\_PLAN%rowtype;

pipl\_id Nsrc.product\_instance\_price\_plan.id%type;

cursor c\_pi\_price is

select price\_plan\_id, id

from Nsrc.product\_instance\_price\_plan

where product\_instance\_id = pi\_id

and (begin\_date is null or begin\_date <= s\_date)

and (end\_date is null or end\_date >= s\_date)

and state = 'U'

and del\_flag = '1';

begin

select \* into r\_pi from Nsrc.product\_instance where id = pi\_id;

if r\_pi.state = 'E' then

s\_date := r\_pi.product\_inst\_end\_dt;

else

s\_date := sysdate;

end if;

open c\_pi\_price;

loop

<<xxx>>

fetch c\_pi\_price

into priceplan\_id, pipl\_id;

exit when c\_pi\_price%notfound;

select \*

into r\_price\_plan

from src.DIM\_GW\_PRICE\_PLAN

where price\_plan\_id = priceplan\_id

and del\_flag = '1'

and state = 'U';

if r\_price\_plan.price\_plan\_type\_cd <> 2 then

charge := 0;

goto xxx;

end if;

begin

select b.factor\_value

into charge

from (select \*

from Nsrc.product\_instance\_price\_plan a

where a.product\_instance\_id = pi\_id

and a.price\_plan\_id = priceplan\_id

and (a.begin\_date is null or a.begin\_date <= s\_date)

and (a.end\_date is null or a.end\_date >= s\_date)

and a.state = 'U'

and a.del\_flag = '1'

and a.id = pipl\_id) a,

(select \*

from Nsrc.product\_instance\_param\_value b

where b.state = 'U') b

where a.id = b.product\_instance\_price\_plan\_id

and rownum = 1;

exception

when no\_data\_found then

begin

select a.rate

into charge

from src.DIM\_GW\_SUIT a, src.DIM\_GW\_PRICE\_PLAN\_PRICE\_ITEM b

where a.suit\_id = b.price\_item\_id

and b.price\_plan\_id = priceplan\_id

and a.state = 'U'

and b.state = 'U'

and b.del\_flg = '1';

exception

when others then

charge := 0;

goto xxx;

end;

when others then

charge := 0;

goto xxx;

end;

Result := Result + charge;

end loop;

close c\_pi\_price;

return Result;

end GetAllCycleFeeRateByPi;

### Hive中处理思路及方法（需补充）

## （游标类）GETCEASETYPEBYSUBORDERID

### Oracle数据库中对应OBJECT\_ID

350409

### 作用描述（需补充）

### 函数体内容

function getCEASETYPEbysuborderid(s\_id in src.so\_sub\_order.id%type)

return varchar2 is

Result varchar2(4000);

c\_name varchar2(4000);

i number := 0;

cursor c\_t is

select xb.name

from src.so\_service\_order xa,

src.product\_item xb,

src.product\_item\_kind\_element xc,

src.product\_item\_kind xd

where xa.so\_sub\_order\_id = s\_id

and xa.state != 'E'

and xa.del\_flag = '1'

and xa.service\_id = xb.id

and xb.state = 'U'

and xb.id = xc.product\_item\_id

and xc.del\_flag = '1'

and xc.product\_item\_kind\_id = xd.id

and xd.item\_kind\_code = 'ZZ\_CEASE\_TYPE'

and xd.del\_flag = '1';

begin

open c\_t;

loop

fetch c\_t

into c\_name;

exit when c\_t%notfound;

if i > 0 then

Result := Result || ';' || c\_name;

else

Result := c\_name;

end if;

i := i + 1;

end loop;

close c\_t;

return Result;

end getCEASETYPEbysuborderid;

### Hive中处理思路及方法（需补充）

## （游标类）GETATTRBYSUBORDERIDATTRCODE

### Oracle数据库中对应OBJECT\_ID

350408

### 作用描述（需补充）

### 函数体内容

function getAttrBySubOrderIdAttrCode(s\_id in NSRC.so\_sub\_order.id%type,

attrCode VARCHAR2) --

return varchar2 is

Result varchar2(4000);

c\_name varchar2(4000);

i number := 0;

cursor c\_t is

select xc.attr\_value

from NSRC.so\_service\_order xa,

NSRC.service\_attr xb,

NSRC.so\_attr\_detail xc,

src.fl\_product\_item xd

where xa.so\_sub\_order\_id = s\_id

and xa.del\_flag = '1'

and xb.service\_id = xa.service\_id

and xb.base\_flag = 'B'

and xc.so\_service\_detail\_id = xa.id

and xc.attr\_type\_id = xd.id

and xd.code = attrCode

and rownum = 1;

begin

open c\_t;

loop

fetch c\_t

into c\_name;

exit when c\_t%notfound;

if i > 0 then

Result := Result || ';' || c\_name;

else

Result := c\_name;

end if;

i := i + 1;

end loop;

close c\_t;

return Result;

end getAttrBySubOrderIdAttrCode;

### Hive中处理思路及方法（需补充）

## （游标类）GETATTRBYPIIDATTRCODE

### Oracle数据库中对应OBJECT\_ID

350407

### 作用描述（需补充）

### 函数体内容

function getAttrByPIIdAttrCode(s\_id number,

attrCode VARCHAR2)

return varchar2 is

Result varchar2(4000);

c\_name varchar2(4000);

i number := 0;

cursor c\_t is

select xc.attr\_value

from src.fl\_service\_instance xa,

NSRC.service\_attr xb,

src.fl\_attr\_instance xc,

src.fl\_product\_item xd

where xa.product\_instance\_id = s\_id

and xa.del\_flag = '1'

and xa.state = 'U'

and xb.service\_id = xa.service\_id

and xb.base\_flag = 'B'

and xc.service\_instance\_id = xa.id

and xc.attr\_id = xd.id

and xd.code = attrCode

and rownum = 1;

begin

open c\_t;

loop

fetch c\_t

into c\_name;

exit when c\_t%notfound;

if i > 0 then

Result := Result || ';' || c\_name;

else

Result := c\_name;

end if;

i := i + 1;

end loop;

close c\_t;

return Result;

end getAttrByPIIdAttrCode;

### Hive中处理思路及方法（需补充）

## （游标类）GETSERVICEBYPIIDSERKIND

### Oracle数据库中对应OBJECT\_ID

350417

### 作用描述（需补充）

### 函数体内容

function getServiceByPIIdSerKind(s\_id number,

serviceItemCode VARCHAR2)

return varchar2 is

Result varchar2(4000);

c\_name varchar2(4000);

i number := 0;

cursor c\_t is

select /\*+ rule \*/xb.name

from NSRC.service\_instance xa,

NSRC.product\_item xb,

NSRC.product\_item\_kind\_element xc,

NSRC.product\_item\_kind xd

where xa.product\_instance\_id = s\_id

and xa.state != 'E'

and xa.del\_flag = '1'

and xa.service\_id = xb.id

and xb.state = 'U'

and xb.id = xc.product\_item\_id

and xc.del\_flag = '1'

and xc.product\_item\_kind\_id = xd.id

and xd.item\_kind\_code = serviceItemCode

and xd.del\_flag = '1'

and rownum = 1;

begin

open c\_t;

loop

fetch c\_t

into c\_name;

exit when c\_t%notfound;

if i > 0 then

Result := Result || ';' || c\_name;

else

Result := c\_name;

end if;

i := i + 1;

end loop;

close c\_t;

return Result;

end getServiceByPIIdSerKind;

### Hive中处理思路及方法（需补充）

## （游标类）GETALLCYCLEFEERATEBYPI\_ALL

### Oracle数据库中对应OBJECT\_ID

3308550

### 作用描述（需补充）

### 函数体内容

function GetAllCycleFeeRateByPi\_ALL(pi\_id in src.product\_instance.id%type)

return number is

Result number(12) := 0;

charge number(12) := 0;

r\_pi src.product\_instance%rowtype;

s\_date src.product\_instance.product\_inst\_end\_dt%type;

priceplan\_id src.dim\_gw\_price\_plan.price\_plan\_id%type;

r\_price\_plan src.DIM\_GW\_PRICE\_PLAN%rowtype;

pipl\_id src.product\_instance\_price\_plan.id%type;

V\_NUM NUMBER;

cursor c\_pi\_price is

select price\_plan\_id, id

from src.product\_instance\_price\_plan

where product\_instance\_id = pi\_id

and (begin\_date is null or begin\_date <= s\_date)

and (end\_date is null or end\_date >= s\_date)

and state = 'U'

and del\_flag = '1';

begin

select \* into r\_pi from src.product\_instance where id = pi\_id;

if r\_pi.state = 'E' then

s\_date := r\_pi.product\_inst\_end\_dt;

else

s\_date := sysdate;

end if;

open c\_pi\_price;

loop

<<xxx>>

fetch c\_pi\_price

into priceplan\_id, pipl\_id;

exit when c\_pi\_price%notfound;

select \*

into r\_price\_plan

from src.DIM\_GW\_PRICE\_PLAN

where price\_plan\_id = priceplan\_id

and del\_flag = '1'

and state = 'U';

if r\_price\_plan.price\_plan\_type\_cd <> 2 then

charge := 0;

goto xxx;

end if;

begin

select b.factor\_value

into charge

from (select \*

from src.product\_instance\_price\_plan a

where a.product\_instance\_id = pi\_id

and a.price\_plan\_id = priceplan\_id

and (a.begin\_date is null or a.begin\_date <= s\_date)

and (a.end\_date is null or a.end\_date >= s\_date)

and a.state = 'U'

and a.del\_flag = '1'

and a.id = pipl\_id) a,

(select \*

from src.product\_instance\_param\_value b

where b.state = 'U') b

where a.id = b.product\_instance\_price\_plan\_id

and rownum = 1;

exception

when no\_data\_found then

begin

SELECT COUNT(\*)

INTO V\_NUM

FROM src.DIM\_GW\_SUIT a, src.DIM\_GW\_PRICE\_PLAN\_PRICE\_ITEM b

where a.suit\_id = b.price\_item\_id

and b.price\_plan\_id = priceplan\_id

and a.state = 'U'

and b.state = 'U'

and b.del\_flg = '1';

if V\_NUM > 0 then

select a.rate

into charge

from src.DIM\_GW\_SUIT a, src.DIM\_GW\_PRICE\_PLAN\_PRICE\_ITEM b

where a.suit\_id = b.price\_item\_id

and b.price\_plan\_id = priceplan\_id

and a.state = 'U'

and b.state = 'U'

and b.del\_flg = '1';

ELSE

---从C3未取到月租，转换至C1取月租

SELECT SUM(EA.BASE\_FEE / NVL(PERIOD\_NUM, 1) / 1000) FEE

INTO CHARGE

FROM SRC.PM\_COMPONENT\_MEMBERS\_INV1C AA,

SRC.PE\_PRODUCT\_DEF BA,

SRC.PE\_PRODUCT\_RATE CA,

SRC.PE\_RATE\_DEF DA,

SRC.PE\_RATE\_DETAIL EA,

SRC.UCS\_SUBS\_COMPONENT\_DAY FA

WHERE AA.COMPONENT\_ID = FA.COMPONENT\_ID

AND AA.MEMBER\_TYPE = 2

AND AA.MEMBER\_ID = BA.PRODUCT\_ID

AND BA.PRODUCT\_ID = CA.PRODUCT\_ID

AND CA.RATE\_ID = DA.RATE\_ID

AND DA.RATE\_ID = EA.RATE\_ID

AND CA.RANGE\_TYPE = 1

AND FA.SUBSCRIPTION\_ID = PI\_ID

AND EA.FLAG = 0

AND FA.ABS\_ACTIVE\_TIME < SYSDATE

AND FA.ABS\_INACTIVE\_TIME > SYSDATE;

end if;

exception

when others then

charge := 0;

goto xxx;

end;

when others then

charge := 0;

goto xxx;

end;

Result := Result + charge;

end loop;

close c\_pi\_price;

return Result;

end GetAllCycleFeeRateByPi\_ALL;

### Hive中处理思路及方法（需补充）

## （游标类）GETSERVATTRINFO

### Oracle数据库中对应OBJECT\_ID

350420

### 作用描述（需补充）

### 函数体内容

FUNCTION ""GETSERVATTRINFO"" (SO\_SUB\_ORDER\_ID NUMBER,

PI\_ID NUMBER,

INFO\_TYPE VARCHAR2,

INFO\_CODE VARCHAR2,

SOURCE\_TYPE VARCHAR2)

RETURN VARCHAR2 IS

RESULT VARCHAR2(4000);

C\_NAME VARCHAR2(4000);

I NUMBER := 0;

CURSOR C\_T IS

SELECT CASE

WHEN INFO\_TYPE = 'SERVICE\_KIND' AND SOURCE\_TYPE = 'SUB\_ORDER' THEN

GETSERVICEBYSUBORDERIDSERKIND(SO\_SUB\_ORDER\_ID, INFO\_CODE)

WHEN INFO\_TYPE = 'SERVICE\_KIND' AND SOURCE\_TYPE = 'INSTANCE' THEN

GETSERVICEBYPIIDSERKIND(PI\_ID, INFO\_CODE)

WHEN INFO\_TYPE = 'SERVICE\_KIND' AND SOURCE\_TYPE = 'OLD\_BAK' THEN

GETOLDSERVICEBYSUBIDSERKIND(SO\_SUB\_ORDER\_ID, INFO\_CODE)

WHEN INFO\_TYPE = 'SERVICE\_CODE' AND SOURCE\_TYPE = 'SUB\_ORDER' THEN

GETSERVICEBYSUBORDERIDSERCODE(SO\_SUB\_ORDER\_ID, INFO\_CODE)

WHEN INFO\_TYPE = 'SERVICE\_CODE' AND SOURCE\_TYPE = 'INSTANCE' THEN

GETSERVICEBYPIIDSERCODE(PI\_ID, INFO\_CODE)

WHEN INFO\_TYPE = 'SERVICE\_CODE' AND SOURCE\_TYPE = 'OLD\_BAK' THEN

GETOLDSERVICEBYSUBIDSERCODE(SO\_SUB\_ORDER\_ID, INFO\_CODE)

WHEN INFO\_TYPE = 'ATTR\_CODE' AND SOURCE\_TYPE = 'SUB\_ORDER' THEN

GETATTRBYSUBORDERIDATTRCODE(SO\_SUB\_ORDER\_ID, INFO\_CODE)

WHEN INFO\_TYPE = 'ATTR\_CODE' AND SOURCE\_TYPE = 'INSTANCE' THEN

GETATTRBYPIIDATTRCODE(PI\_ID, INFO\_CODE)

WHEN INFO\_TYPE = 'ATTR\_CODE' AND SOURCE\_TYPE = 'OLD\_BAK' THEN

GETOLDATTRBYSUBIDATTRCODE(SO\_SUB\_ORDER\_ID, INFO\_CODE)

ELSE

NULL

END

FROM DUAL;

BEGIN

OPEN C\_T;

LOOP

FETCH C\_T

INTO C\_NAME;

EXIT WHEN C\_T%NOTFOUND;

IF I > 0 THEN

RESULT := RESULT || ';' || C\_NAME;

ELSE

RESULT := C\_NAME;

END IF;

I := I + 1;

END LOOP;

CLOSE C\_T;

RETURN RESULT;

END GETSERVATTRINFO;

### Hive中处理思路及方法（需补充）

## （游标类）GETSERVICEBYSUBORDERIDSERKIND

### Oracle数据库中对应OBJECT\_ID

350419

### 作用描述（需补充）

### 函数体内容

function getServiceBySubOrderIdSerKind(s\_id in Nsrc.so\_sub\_order.id%type,

serviceItemCode VARCHAR2)

return varchar2 is

Result varchar2(4000);

c\_name varchar2(4000);

i number := 0;

cursor c\_t is

select /\*+ rule \*/xb.name

from Nsrc.so\_service\_order xa,

Nsrc.product\_item xb,

Nsrc.product\_item\_kind\_element xc,

Nsrc.product\_item\_kind xd

where xa.so\_sub\_order\_id = s\_id

and xa.state != 'E'

and xa.del\_flag = '1'

and xa.service\_id = xb.id

and xb.state = 'U'

and xb.id = xc.product\_item\_id

and xc.del\_flag = '1'

and xc.product\_item\_kind\_id = xd.id

and xd.item\_kind\_code = serviceItemCode

and xd.del\_flag = '1'

and rownum = 1;

begin

open c\_t;

loop

fetch c\_t

into c\_name;

exit when c\_t%notfound;

if i > 0 then

Result := Result || ';' || c\_name;

else

Result := c\_name;

end if;

i := i + 1;

end loop;

close c\_t;

return Result;

end getServiceBySubOrderIdSerKind;

### Hive中处理思路及方法（需补充）

## （游标类）GETSERVICEBYSUBORDERIDSERCODE

### Oracle数据库中对应OBJECT\_ID

350418

### 作用描述（需补充）

### 函数体内容

function getServiceBySubOrderIdSerCode(s\_id in nSRC.so\_sub\_order.id%type,

serviceCode VARCHAR2)

return varchar2 is

Result varchar2(4000);

c\_name varchar2(4000);

i number := 0;

cursor c\_t is

select xb.name

from nSRC.so\_service\_order xa,

nSRC.product\_item xb

where xa.so\_sub\_order\_id = s\_id

and xa.state != 'E'

and xa.del\_flag = '1'

and xa.service\_id = xb.id

and xb.state = 'U'

and xb.code = serviceCode

and rownum = 1;

begin

open c\_t;

loop

fetch c\_t

into c\_name;

exit when c\_t%notfound;

if i > 0 then

Result := Result || ';' || c\_name;

else

Result := c\_name;

end if;

i := i + 1;

end loop;

close c\_t;

return Result;

end getServiceBySubOrderIdSerCode;

### Hive中处理思路及方法（需补充）

## （游标类）GETSYSFIELDDESCINFO

### Oracle数据库中对应OBJECT\_ID

350421

### 作用描述（需补充）

### 函数体内容

FUNCTION GETSYSFIELDDESCINFO(I\_TABLE\_NAME VARCHAR2,

I\_COLUMN\_NAME VARCHAR2,

COMINGVALUE VARCHAR2)

RETURN VARCHAR2 IS

RESULT VARCHAR2(4000);

C\_NAME VARCHAR2(4000);

I NUMBER := 0;

CURSOR C\_T IS

SELECT XA.DESCRIPTION

FROM src.SYS\_FIELD\_DESC XA

WHERE XA.TABLE\_NAME = I\_TABLE\_NAME

AND XA.COLUMN\_NAME = I\_COLUMN\_NAME

AND XA.DEL\_FLAG = '1'

AND XA.VALUE = COMINGVALUE

AND ROWNUM = 1;

BEGIN

OPEN C\_T;

LOOP

FETCH C\_T

INTO C\_NAME;

EXIT WHEN C\_T%NOTFOUND;

IF I > 0 THEN

RESULT := RESULT || ';' || C\_NAME;

ELSE

RESULT := C\_NAME;

END IF;

I := I + 1;

END LOOP;

CLOSE C\_T;

RETURN RESULT;

END GETSYSFIELDDESCINFO;

### Hive中处理思路及方法（需补充）

## （游标类）GET\_PROJECTCORRES\_BY\_ORDER\_ID

### Oracle数据库中对应OBJECT\_ID

350423

### 作用描述（需补充）

### 函数体内容

FUNCTION GET\_PROJECTCORRES\_BY\_ORDER\_ID(ORDER\_ID NUMBER)

RETURN VARCHAR2 DETERMINISTIC IS

SEL\_SQL VARCHAR2(2000);

RETURN\_STR VARCHAR2(3990);

TYPE CUR IS REF CURSOR;

CURSOR\_DATA CUR;

XML VARCHAR2(4000);

PROC\_REQUEST VARCHAR2(4000);

I NUMBER;

J NUMBER;

K NUMBER;

BEGIN

K := 0;

XML := '';

IF ORDER\_ID IS NULL THEN

RETURN\_STR := '';

ELSE

SEL\_SQL := ' select oms\_task\_instance\_xml\_1 from ( select a1.task\_id,a1.OMS\_TASK\_INSTANCE\_XML\_1

from src.OMS\_TASK\_INSTANCE a1

where a1.ORDER\_id=' || ORDER\_ID ||

' and a1.oms\_task\_type\_code like ''com%maintenanceconstruct%''

union

select a1.task\_id,a1.OMS\_TASK\_INSTANCE\_XML\_1

from src.OMS\_TASK\_INSTANCE\_his a1

where a1.ORDER\_id=' || ORDER\_ID ||

' and a1.oms\_task\_type\_code like ''com%maintenanceconstruct%''

) order by task\_id asc ';

OPEN CURSOR\_DATA FOR SEL\_SQL;

LOOP

FETCH CURSOR\_DATA

INTO XML;

EXIT WHEN CURSOR\_DATA%NOTFOUND;

I := 0;

J := 0;

PROC\_REQUEST := '';

I := INSTR(XML, 'assortsend');

IF I > 0 THEN

K := K + 1;

J := INSTR(XML, 'req\_content', I);

I := J + 33;

J := INSTR(XML, '""/>', I);

PROC\_REQUEST := SUBSTR(XML, I, J - I);

-- dbms\_output.put\_line(proc\_request);

IF RETURN\_STR IS NULL THEN

RETURN\_STR := '第' || K || '次请求协调信息为:' || PROC\_REQUEST;

ELSE

IF LENGTHB(RETURN\_STR) < 2000 THEN

RETURN\_STR := RETURN\_STR || ';第' || K || '次请求协调信息为:' ||

PROC\_REQUEST;

END IF;

END IF;

END IF;

END LOOP;

CLOSE CURSOR\_DATA;

END IF;

RETURN SUBSTR(TRIM(RETURN\_STR), 0, 3000);

END;

### Hive中处理思路及方法（需补充）

## （游标类）GET\_CORRES\_RTN\_BY\_ORDER\_ID

### Oracle数据库中对应OBJECT\_ID

350422

### 作用描述（需补充）

### 函数体内容

FUNCTION GET\_CORRES\_RTN\_BY\_ORDER\_ID(ORDER\_ID NUMBER)

RETURN VARCHAR2 DETERMINISTIC IS

SEL\_SQL VARCHAR2(2000);

RETURN\_STR VARCHAR2(3990);

TYPE CUR IS REF CURSOR;

CURSOR\_DATA CUR;

CURSOR\_DATA2 CUR;

XML VARCHAR2(4000);

V\_TASK\_TYPE\_CODE VARCHAR2(1000);

V\_TASK\_ID NUMBER;

K NUMBER;

BEGIN

K := 0;

XML := '';

V\_TASK\_ID := 0;

--DBMS\_OUTPUT.put\_line(ORDER\_ID);

IF ORDER\_ID IS NULL THEN

RETURN\_STR := '';

ELSE

SEL\_SQL := ' select task\_id,xpath\_get(''/rtn/proc\_result'',oms\_task\_instance\_xml\_1)||xpath\_get(''/rtn/msg'',oms\_task\_instance\_xml\_1)

from (select a1.task\_id,a1.OMS\_TASK\_INSTANCE\_XML\_1

from src.OMS\_TASK\_INSTANCE a1

where a1.ORDER\_id=' || ORDER\_ID ||

' and a1.oms\_task\_type\_code like ''com%projectcorres%''

) order by task\_id asc ';

OPEN CURSOR\_DATA FOR SEL\_SQL;

LOOP

FETCH CURSOR\_DATA

INTO V\_TASK\_ID, XML;

EXIT WHEN CURSOR\_DATA%NOTFOUND;

V\_TASK\_TYPE\_CODE := '';

OPEN CURSOR\_DATA2 FOR 'select c.task\_type\_code from src.work\_flow\_task\_instance c

where c.task\_id = (

select min(a.task\_id) from src.work\_flow\_task\_instance a,src.oms\_task\_instance b

where a.business\_order\_id = b.order\_id and b.task\_id = ' || V\_TASK\_ID || ' and a.task\_id > ' || V\_TASK\_ID || ' and a.task\_type\_code not like ''%exceptiondeal%'') ';

LOOP

FETCH CURSOR\_DATA2

INTO V\_TASK\_TYPE\_CODE;

EXIT WHEN CURSOR\_DATA2%NOTFOUND;

K := K + 1;

-- dbms\_output.put\_line(proc\_request);

IF RETURN\_STR IS NULL THEN

IF INSTR(V\_TASK\_TYPE\_CODE, 'maintenanceconstruct') > 0 THEN

K := K + 1;

RETURN\_STR := '第' || K || '次请求协调反馈信息为:';

END IF;

ELSE

IF INSTR(V\_TASK\_TYPE\_CODE, 'maintenanceconstruct') > 0 THEN

K := K + 1;

RETURN\_STR := RETURN\_STR || ' 第' || K || '次请求协调反馈信息为:';

END IF;

END IF;

END LOOP;

CLOSE CURSOR\_DATA2;

END LOOP;

CLOSE CURSOR\_DATA;

END IF;

RETURN SUBSTR(TRIM(RETURN\_STR || XML), 0, 3000);

END;

### Hive中处理思路及方法（需补充）

## （游标类）GETSERVICEBYSUBORDERIDSKVATE

### Oracle数据库中对应OBJECT\_ID

356103

### 作用描述（需补充）

### 函数体内容

function getServiceBySubOrderIdSKvate(s\_id in src.so\_sub\_order.id%type,

serviceItemCode VARCHAR2)

return varchar2 is

Result varchar2(4000);

c\_name varchar2(4000);

i number := 0;

cursor c\_t is

select xb.name

from (select /\*+parallel(xa,4)\*/

\*

from src.so\_service\_order xa

where xa.state != 'E'

and xa.del\_flag = '1'

and xa.so\_sub\_order\_id = s\_id) xa,

(select /\*+parallel(xb,4)\*/

\*

from src.product\_item xb

where xb.state = 'U') xb,

(select /\*+parallel(xc,4)\*/

\*

from src.product\_item\_kind\_element xc

where xc.del\_flag = '1') xc,

(select /\*+parallel(xd,4)\*/

\*

from src.product\_item\_kind xd

where xd.del\_flag = '1'

and xd.item\_kind\_code = serviceItemCode) xd

where xa.service\_id = xb.id

and xb.id = xc.product\_item\_id

and xc.product\_item\_kind\_id = xd.id

and rownum = 1;

begin

open c\_t;

loop

fetch c\_t

into c\_name;

exit when c\_t%notfound;

if i > 0 then

Result := Result || ';' || c\_name;

else

Result := c\_name;

end if;

i := i + 1;

end loop;

close c\_t;

return Result;

end getServiceBySubOrderIdSKvate;

### Hive中处理思路及方法（需补充）

## （游标类）GETSERVATTRINFOVATE

### Oracle数据库中对应OBJECT\_ID

356104

### 作用描述（需补充）

### 函数体内容

FUNCTION GETSERVATTRINFOVATE (SO\_SUB\_ORDER\_ID NUMBER,

PI\_ID NUMBER,

INFO\_TYPE VARCHAR2,

INFO\_CODE VARCHAR2,

SOURCE\_TYPE VARCHAR2)

RETURN VARCHAR2 IS

RESULT VARCHAR2(4000);

C\_NAME VARCHAR2(4000);

I NUMBER := 0;

CURSOR C\_T IS

SELECT CASE

WHEN INFO\_TYPE = 'SERVICE\_KIND' AND SOURCE\_TYPE = 'SUB\_ORDER' THEN

getServiceBySubOrderIdSKvate(SO\_SUB\_ORDER\_ID, INFO\_CODE)

WHEN INFO\_TYPE = 'SERVICE\_KIND' AND SOURCE\_TYPE = 'INSTANCE' THEN

GETSERVICEBYPIIDSERKINDvate(PI\_ID, INFO\_CODE)

WHEN INFO\_TYPE = 'SERVICE\_KIND' AND SOURCE\_TYPE = 'OLD\_BAK' THEN

GETOLDSERVICEBYSUBIDSERKIND(SO\_SUB\_ORDER\_ID, INFO\_CODE)

WHEN INFO\_TYPE = 'SERVICE\_CODE' AND SOURCE\_TYPE = 'SUB\_ORDER' THEN

GETSERVICEBYSUBORDERIDSERCODE(SO\_SUB\_ORDER\_ID, INFO\_CODE)

WHEN INFO\_TYPE = 'SERVICE\_CODE' AND SOURCE\_TYPE = 'INSTANCE' THEN

GETSERVICEBYPIIDSERCODE(PI\_ID, INFO\_CODE)

WHEN INFO\_TYPE = 'SERVICE\_CODE' AND SOURCE\_TYPE = 'OLD\_BAK' THEN

GETOLDSERVICEBYSUBIDSERCODE(SO\_SUB\_ORDER\_ID, INFO\_CODE)

WHEN INFO\_TYPE = 'ATTR\_CODE' AND SOURCE\_TYPE = 'SUB\_ORDER' THEN

GETATTRBYSUBORDERIDATTRCODE(SO\_SUB\_ORDER\_ID, INFO\_CODE)

WHEN INFO\_TYPE = 'ATTR\_CODE' AND SOURCE\_TYPE = 'INSTANCE' THEN

GETATTRBYPIIDATTRCODE(PI\_ID, INFO\_CODE)

WHEN INFO\_TYPE = 'ATTR\_CODE' AND SOURCE\_TYPE = 'OLD\_BAK' THEN

GETOLDATTRBYSUBIDATTRCODE(SO\_SUB\_ORDER\_ID, INFO\_CODE)

ELSE

NULL

END

FROM DUAL;

BEGIN

OPEN C\_T;

LOOP

FETCH C\_T

INTO C\_NAME;

EXIT WHEN C\_T%NOTFOUND;

IF I > 0 THEN

RESULT := RESULT || ';' || C\_NAME;

ELSE

RESULT := C\_NAME;

END IF;

I := I + 1;

END LOOP;

CLOSE C\_T;

RETURN RESULT;

END GETSERVATTRINFOVATE;

### Hive中处理思路及方法（需补充）

## （游标类）GETSERVICEBYPIIDSERKINDVATE

### Oracle数据库中对应OBJECT\_ID

356102

### 作用描述（需补充）

### 函数体内容

function getServiceByPIIdSerKindvate(s\_id number,

serviceItemCode VARCHAR2)

return varchar2 is

Result varchar2(4000);

c\_name varchar2(4000);

i number := 0;

cursor c\_t is

select xb.name

from (select /\*+parallel(xa,4)\*/

\*

from SRC.service\_instance xa

where xa.product\_instance\_id = s\_id

and xa.state != 'E'

and xa.del\_flag = '1') xa,

(select /\*+parallel(xb,4)\*/

\*

from SRC.product\_item xb

where xb.state = 'U') xb,

(select /\*+parallel(xc,4)\*/

\*

from SRC.product\_item\_kind\_element xc

where xc.del\_flag = '1') xc,

(select /\*+parallel(xd,4)\*/

\*

from SRC.product\_item\_kind xd

where xd.item\_kind\_code = serviceItemCode

and xd.del\_flag = '1') xd

where xa.service\_id = xb.id

and xb.id = xc.product\_item\_id

and xc.product\_item\_kind\_id = xd.id

and rownum = 1;

begin

open c\_t;

loop

fetch c\_t

into c\_name;

exit when c\_t%notfound;

if i > 0 then

Result := Result || ';' || c\_name;

else

Result := c\_name;

end if;

i := i + 1;

end loop;

close c\_t;

return Result;

end getServiceByPIIdSerKindvate;

### Hive中处理思路及方法（需补充）

## （游标类）FUN\_ONETIME\_FEE\_CHG

### Oracle数据库中对应OBJECT\_ID

350387

### 作用描述（需补充）

### 函数体内容

FUNCTION FUN\_ONETIME\_FEE\_CHG(S\_ID IN SRC.SO\_SUB\_ORDER\_PRICE\_PLAN.SO\_SUB\_ORDER\_ID%TYPE)

RETURN NUMBER IS

RESULT NUMBER(12) := 0;

CHARGE NUMBER(12) := 0;

CURSOR C\_SO\_PRICE\_PLAN IS

SELECT \*

FROM SRC.SO\_SUB\_ORDER\_PRICE\_PLAN

WHERE SO\_SUB\_ORDER\_ID = S\_ID

AND DEL\_FLAG = '1';

R\_SO\_SUB SRC.SO\_SUB\_ORDER\_PRICE\_PLAN%ROWTYPE;

R\_PRICE\_PLAN DIM.DIM\_GW\_PRICE\_PLAN%ROWTYPE;

R\_SO\_SUB\_PARAM\_VALUE SRC.SO\_SUB\_ORDER\_PARAM\_VALUE%ROWTYPE;

BEGIN

RESULT := 0;

BEGIN

SELECT SUM(XA.BILL\_AMOUNT)

INTO RESULT

FROM SRC.BILL /\*SRC.BILL\_ONETIMEFEE\*/ XA

WHERE XA.ROOT\_INDEX IN

(

SELECT SO\_ORDER\_ID FROM SRC.SO\_SUB\_ORDER WHERE ID = S\_ID

)

AND XA.ROOT\_TYPE = 'SO'

AND XA.STATE <> 'L';

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RESULT := 0;

END;

IF RESULT = 0 OR RESULT IS NULL THEN

-- 2007-08-21增加or result is null 撤单一次性费用

RESULT := 0;

OPEN C\_SO\_PRICE\_PLAN;

LOOP

<<XXX>>

FETCH C\_SO\_PRICE\_PLAN

INTO R\_SO\_SUB;

EXIT WHEN C\_SO\_PRICE\_PLAN%NOTFOUND;

SELECT \*

INTO R\_PRICE\_PLAN

FROM DIM.DIM\_GW\_PRICE\_PLAN

WHERE PRICE\_PLAN\_ID = R\_SO\_SUB.PRICE\_PLAN\_ID

AND DEL\_FLAG = '1'

AND STATE = 'U';

IF R\_PRICE\_PLAN.PRICE\_PLAN\_TYPE\_CD <> 9 THEN

CHARGE := 0;

GOTO XXX;

END IF;

BEGIN

SELECT \*

INTO R\_SO\_SUB\_PARAM\_VALUE

FROM SRC.SO\_SUB\_ORDER\_PARAM\_VALUE

WHERE SUB\_ORDER\_PRICE\_PLAN\_ID = R\_SO\_SUB.ID;

CHARGE := R\_SO\_SUB\_PARAM\_VALUE.FACTOR\_VALUE;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

CHARGE := 0;

END;

RESULT := RESULT + CHARGE;

END LOOP;

CLOSE C\_SO\_PRICE\_PLAN;

END IF;

RETURN RESULT;

EXCEPTION

WHEN OTHERS THEN

RETURN 0;

END FUN\_ONETIME\_FEE\_CHG;

### Hive中处理思路及方法（需补充）

## （游标类）FUN\_CYCLE\_FEE\_RATE

### Oracle数据库中对应OBJECT\_ID

350383

### 作用描述（需补充）

### 函数体内容

FUNCTION FUN\_CYCLE\_FEE\_RATE(PI\_ID IN SRC.PRODUCT\_INSTANCE.ID%TYPE)

RETURN NUMBER IS

RESULT NUMBER(12) := 0;

CHARGE NUMBER(12) := 0;

R\_PI NSRC.PRODUCT\_INSTANCE%ROWTYPE;

S\_DATE NSRC.PRODUCT\_INSTANCE.PRODUCT\_INST\_END\_DT%TYPE;

PRICEPLAN\_ID DIM.DIM\_GW\_PRICE\_PLAN.PRICE\_PLAN\_ID%TYPE;

R\_PRICE\_PLAN DIM.DIM\_GW\_PRICE\_PLAN%ROWTYPE;

PIPL\_ID NSRC.PRODUCT\_INSTANCE\_PRICE\_PLAN.ID%TYPE;

CURSOR C\_PI\_PRICE IS

SELECT PRICE\_PLAN\_ID, ID

FROM NSRC.PRODUCT\_INSTANCE\_PRICE\_PLAN

WHERE PRODUCT\_INSTANCE\_ID = PI\_ID

AND (BEGIN\_DATE IS NULL OR BEGIN\_DATE <= S\_DATE)

AND (END\_DATE IS NULL OR END\_DATE >= S\_DATE)

AND STATE = 'U'

AND DEL\_FLAG = '1';

BEGIN

SELECT \* INTO R\_PI FROM NSRC.PRODUCT\_INSTANCE WHERE ID = PI\_ID;

IF R\_PI.STATE = 'E' THEN

S\_DATE := R\_PI.PRODUCT\_INST\_END\_DT;

ELSE

S\_DATE := SYSDATE;

END IF;

OPEN C\_PI\_PRICE;

LOOP

<<XXX>>

FETCH C\_PI\_PRICE

INTO PRICEPLAN\_ID, PIPL\_ID;

EXIT WHEN C\_PI\_PRICE%NOTFOUND;

SELECT \*

INTO R\_PRICE\_PLAN

FROM NDIM.DIM\_GW\_PRICE\_PLAN

WHERE PRICE\_PLAN\_ID = PRICEPLAN\_ID

AND DEL\_FLAG = '1'

AND STATE = 'U';

IF R\_PRICE\_PLAN.PRICE\_PLAN\_TYPE\_CD <> 2 THEN

CHARGE := 0;

GOTO XXX;

END IF;

BEGIN

SELECT B.FACTOR\_VALUE

INTO CHARGE

FROM (SELECT \*

FROM NSRC.PRODUCT\_INSTANCE\_PRICE\_PLAN A

WHERE A.PRODUCT\_INSTANCE\_ID = PI\_ID

AND A.PRICE\_PLAN\_ID = PRICEPLAN\_ID

AND (A.BEGIN\_DATE IS NULL OR A.BEGIN\_DATE <= S\_DATE)

AND (A.END\_DATE IS NULL OR A.END\_DATE >= S\_DATE)

AND A.STATE = 'U'

AND A.DEL\_FLAG = '1'

AND A.ID = PIPL\_ID) A,

(SELECT \*

FROM NSRC.PRODUCT\_INSTANCE\_PARAM\_VALUE B

WHERE B.STATE = 'U') B

WHERE A.ID = B.PRODUCT\_INSTANCE\_PRICE\_PLAN\_ID

AND ROWNUM = 1;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

BEGIN

SELECT A.RATE

INTO CHARGE

FROM SRC.DIM\_GW\_SUIT A, NDIM.DIM\_GW\_PRICE\_PLAN\_PRICE\_ITEM B

WHERE A.SUIT\_ID = B.PRICE\_ITEM\_ID

AND B.PRICE\_PLAN\_ID = PRICEPLAN\_ID

AND A.STATE = 'U'

AND B.STATE = 'U'

AND B.DEL\_FLG = '1';

EXCEPTION

WHEN OTHERS THEN

CHARGE := 0;

GOTO XXX;

END;

WHEN OTHERS THEN

CHARGE := 0;

GOTO XXX;

END;

RESULT := RESULT + CHARGE;

END LOOP;

CLOSE C\_PI\_PRICE;

RETURN RESULT;

END FUN\_CYCLE\_FEE\_RATE;

### Hive中处理思路及方法（需补充）

## F\_PARTITION\_EXIST

### Oracle数据库中对应OBJECT\_ID

350404

### 作用描述（需补充）

### 函数体内容

FUNCTION F\_PARTITION\_EXIST(OWNER# VARCHAR2,

TABLE\_NAME# VARCHAR2,

PARTITION\_NAME# VARCHAR2)

RETURN BOOLEAN IS

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*名称 --%NAME:判读表的分区是否存在

\*功能描述 --%COMMENT:

\*执行周期 --%PERIOD:日期 每天0:05执行

\*参数 --%PARAM:OWNER#对象名

\*参数 --%PARAM:TABLE\_NAME#表名

\*参数 --%PARAM:PARTITION\_NAME#分区名

\*创建人 --%CREATOR:杨顺利

\*创建时间 --%CREATED\_TIME:20090318

\*备注 --%REMARK:用于判断分区是否存在（应用范围并不大，仅针对以月为分区的情况，有分区模板的

也是以月为主分区）

\*修改记录 --%MODIFY:

\*来源表 --%FROM:

\*目标表 --%TO:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

RESULT BOOLEAN;

RECORDS NUMBER;

BEGIN

RECORDS := 0;

RESULT := FALSE;

SELECT DECODE(COUNT(\*), 0, 0, 1)

INTO RECORDS

FROM DUAL

WHERE EXISTS (SELECT \*

FROM ALL\_TAB\_PARTITIONS

WHERE TABLE\_OWNER = UPPER(OWNER#)

AND TABLE\_NAME = UPPER(TABLE\_NAME#)

AND PARTITION\_NAME = UPPER(PARTITION\_NAME#));

IF RECORDS = 1 THEN

RESULT := TRUE;

END IF;

RETURN(RESULT);

END;

### Hive中处理思路及方法（需补充）

## F\_OBJECT\_EXIST

### Oracle数据库中对应OBJECT\_ID

350403

### 作用描述（需补充）

### 函数体内容

FUNCTION F\_OBJECT\_EXIST(OBJ\_NAME VARCHAR2,

SUBOBJ\_NAME VARCHAR2,

OBJ\_TYPE VARCHAR2,

RETURN BOOLEAN IS

RECORD\_COUNTS NUMBER;

BEGIN

IF SUBOBJ\_NAME IS NULL THEN

SELECT 1

INTO RECORD\_COUNTS

FROM DUAL

WHERE EXISTS (SELECT \*

FROM ALL\_OBJECTS

WHERE OBJECT\_NAME = UPPER(OBJ\_NAME)

AND OWNER = UPPER(OBJ\_OWNER)

AND OBJECT\_TYPE = UPPER(OBJ\_TYPE));

ELSE

SELECT 1

INTO RECORD\_COUNTS

FROM DUAL

WHERE EXISTS (SELECT \*

FROM ALL\_OBJECTS

WHERE OBJECT\_NAME = UPPER(OBJ\_NAME)

AND SUBOBJECT\_NAME = UPPER(SUBOBJ\_NAME)

AND OWNER = UPPER(OBJ\_OWNER)

AND OBJECT\_TYPE = UPPER(OBJ\_TYPE));

END IF;

IF RECORD\_COUNTS = 1 THEN

RETURN TRUE;

ELSE

RETURN FALSE;

END IF;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN FALSE;

END F\_OBJECT\_EXIST;

### Hive中处理思路及方法（需补充）

## F\_WAIT

### Oracle数据库中对应OBJECT\_ID

4558205

### 作用描述（需补充）

### 函数体内容

function F\_WAIT(A in number, B in varchar2) return varchar2 is

--Result varchar2;

v\_out varchar2(100);

i varchar2(50);

v\_sql varchar2(200);

begin

if UPPER(b) = 'S' then

--dbms\_output.put\_line(to\_char(sysdate, 'yyyymmddhh24miss'));

v\_sql := 'select to\_char(sysdate+'||a||'/24/60/60,''yyyymmddhhmiss'') from dual';

execute immediate v\_sql into i;

loop

exit when to\_char(sysdate,'yyyymmddhhmiss') = i;

end loop;

v\_out := 'success';

--dbms\_output.put\_line(to\_char(sysdate, 'yyyymmddhh24miss'));

elsif UPPER(b) = 'M' then

v\_sql := 'select to\_char(sysdate+'||a||'/24/60,''yyyymmddhhmiss'') from dual';

execute immediate v\_sql into i;

loop

exit when to\_char(sysdate,'yyyymmddhhmiss') = i;

end loop;

v\_out := 'success';

elsif UPPER(b) = 'H' then

v\_sql := 'select to\_char(sysdate+'||a||'/24,''yyyymmddhhmiss'') from dual';

execute immediate v\_sql into i;

loop

exit when to\_char(sysdate,'yyyymmddhhmiss') = i;

end loop;

v\_out := 'success';

else

v\_out := 'fail:param B is wrong ';

END IF;

return(v\_out);

end F\_WAIT;

### Hive中处理思路及方法（需补充）

## F\_DAY\_PARTITION\_2

### Oracle数据库中对应OBJECT\_ID

350392

### 作用描述（需补充）

### 函数体内容

FUNCTION F\_DAY\_PARTITION\_2(F\_DATE VARCHAR2)

RETURN VARCHAR2 IS

V\_STANDARDDATE DATE;

V\_DATE DATE;

PART\_ID VARCHAR2(1);

BEGIN

V\_STANDARDDATE := TO\_DATE('19700101', 'YYYYMMDD');

V\_DATE := TO\_DATE(F\_DATE, 'YYYYMMDD');

PART\_ID := MOD(V\_DATE - V\_STANDARDDATE, 2);

RETURN(PART\_ID);

END F\_DAY\_PARTITION\_2;

### Hive中处理思路及方法（需补充）

## XPATH\_GET

### Oracle数据库中对应OBJECT\_ID

350425

### 作用描述（需补充）

### 函数体内容

FUNCTION XPATH\_GET(NODE VARCHAR2, XML VARCHAR2)

RETURN VARCHAR2 DETERMINISTIC IS

RESULT\_POS NUMBER(10);

POS\_XML\_BEGIN NUMBER(10);

POS\_XML\_END NUMBER(10);

RETURN\_STR VARCHAR2(3990);

BEGIN

RESULT\_POS := XPATH\_POS(NODE, XML);

IF RESULT\_POS = 0 THEN

RETURN NULL;

END IF;

POS\_XML\_BEGIN := FLOOR(RESULT\_POS / 10000);

POS\_XML\_END := MOD(RESULT\_POS, 10000);

RETURN\_STR := SUBSTR(XML,

POS\_XML\_BEGIN,

POS\_XML\_END - POS\_XML\_BEGIN + 1);

RETURN\_STR := REPLACE(RETURN\_STR, '<![CDATA[', '');

RETURN\_STR := REPLACE(RETURN\_STR, ']]>', '');

RETURN TRIM(RETURN\_STR);

END;

### Hive中处理思路及方法（需补充）

## XPATH\_POS

### Oracle数据库中对应OBJECT\_ID

350424

### 作用描述（需补充）

### 函数体内容

FUNCTION XPATH\_POS(NODE VARCHAR2, XML VARCHAR2)

RETURN NUMBER IS

POS\_NODE\_BEGIN NUMBER(10);

POS\_NODE\_END NUMBER(10);

POS\_XML\_BEGIN NUMBER(10);

POS\_XML\_END NUMBER(10);

RETURN\_STR VARCHAR2(3990);

IS\_LAST CHAR(1);

TEMP\_NODE\_BEGIN VARCHAR2(100);

TEMP\_NODE\_END VARCHAR2(100);

BEGIN

POS\_NODE\_BEGIN := 1;

POS\_NODE\_END := 1;

POS\_XML\_BEGIN := 1;

POS\_XML\_END := 1;

IS\_LAST := 'N';

LOOP

IF IS\_LAST = 'Y' THEN

EXIT;

END IF;

POS\_NODE\_BEGIN := POS\_NODE\_END;

POS\_NODE\_END := INSTR(NODE, '/', POS\_NODE\_BEGIN + 1);

IF POS\_NODE\_END = 0 THEN

POS\_NODE\_END := LENGTH(NODE) + 1;

END IF;

--为跳过标签属性去掉 temp\_node\_begin := '<'||substr(node,pos\_node\_begin+1,pos\_node\_end-pos\_node\_begin-1 )||'>';

--跳过标签的属性 begin

TEMP\_NODE\_BEGIN := '<' ||

SUBSTR(NODE,

POS\_NODE\_BEGIN + 1,

POS\_NODE\_END - POS\_NODE\_BEGIN - 1);

--跳过标签的属性 end

TEMP\_NODE\_END := '</' ||

SUBSTR(NODE,

POS\_NODE\_BEGIN + 1,

POS\_NODE\_END - POS\_NODE\_BEGIN - 1) || '>';

POS\_XML\_BEGIN := INSTR(XML, TEMP\_NODE\_BEGIN, POS\_XML\_BEGIN);

IF POS\_XML\_BEGIN = 0 THEN

RETURN 0;

END IF;

POS\_XML\_BEGIN := POS\_XML\_BEGIN + LENGTH(TEMP\_NODE\_BEGIN);

--跳过标签的属性 begin

LOOP

IF SUBSTR(XML, POS\_XML\_BEGIN, 1) <> '>' THEN

POS\_XML\_BEGIN := POS\_XML\_BEGIN + 1;

ELSE

POS\_XML\_BEGIN := POS\_XML\_BEGIN + 1;

EXIT;

END IF;

END LOOP;

--跳过标签的属性 end

IF POS\_NODE\_END = LENGTH(NODE) + 1 THEN

IS\_LAST := 'Y';

POS\_XML\_END := INSTR(XML, TEMP\_NODE\_END, POS\_XML\_END);

RETURN\_STR := SUBSTR(XML, POS\_XML\_BEGIN, POS\_XML\_END - POS\_XML\_BEGIN);

END IF;

END LOOP;

--返回的位置是(开始位置\*10000 + 结束位置)

RETURN POS\_XML\_BEGIN \* 10000 +(POS\_XML\_END - 1);

END XPATH\_POS;

### Hive中处理思路及方法（需补充）

# 自定义函数

## GetDayOfWeek

### 类名

org.apache.hadoop.hive.ql.udf. GetDayOfWeek.java

### HIVE函数名

f\_get\_day\_of\_week

### 使用方法

select f\_get\_day\_of\_week ('2013-03-12') from test\_pxg\_datax4 ;

**特殊说明：**传入参数为YYYY-MM-DD格式，返回值为1-7,其中1为星期日，其他依次后推。

（备注：结果有问题，返回值为1是应该是周六，以此类推）。

### Hive自定义函数代码

**package** org.apache.hadoop.hive.ql.udf;

**import** java.text.DateFormat;

**import** java.text.SimpleDateFormat;

**import** java.util.Calendar;

**import** java.util.Date;

**import** org.apache.hadoop.hive.ql.exec.UDF;

/\*

\* 函数功能：获取当天为星期中的第几天

\* Hive自定义函数创建：平晓刚

\* 创建时间：2015-07-06

\* 传入参数：传入参数为YYYY-MM-DD格式

\* 返回参数：返回值为1-7,其中1为星期日，其他依次后推

\* \*/

**public** **class** GetDayOfWeek **extends** UDF{

**public** **int** evaluate(String inputValue) **throws** Exception {

DateFormat fmt =**new** SimpleDateFormat("yyyy-MM-dd");

Date date = fmt.parse(inputValue);

Calendar calendar = Calendar.*getInstance*();

calendar.setTime(date);

**int** i =calendar.get(Calendar.*DAY\_OF\_WEEK*);

**int** j=0;

**if**(i==7){

j=1;

}

**if**(i==1){

j=2;

}

**if**(i==2){

j=3;

}

**if**(i==3){

j=4;

}

**if**(i==4){

j=5;

}

**if**(i==5){

j=6;

}

**if**(i==6){

j=7;

}

**return** j;

}

}

## ADD\_MONTH

### 类名

org.apache.hadoop.hive.ql.udf. AddMonths.java

### HIVE函数名

f\_add\_months

### 使用方法

实例：

select time ,To\_Char\_Date(AddMonths(To\_Date(time,'yyyyMMdd'),1),'yyyyMMdd') from test;

结果：

20131012 20131112

20131011 20131111

20131015 20131115

20131111 20131211

20131211 20140111

20131017 20131117

20131030 20131130

**特殊说明：**传入一个日期格式的数据，返回一个日期格式的数据。：

### Hive自定义函数代码

package org.apache.hadoop.hive.ql.udf;

import java.util.Calendar;

import java.util.Date;

import org.apache.hadoop.hive.ql.exec.UDF;

/\*

\* 函数功能：获取传入时间增加几个月后的日期

\* Hive自定义函数创建：平晓刚

\* 创建时间：2015-07-06

\* 传入参数1：传入参数为YYYY-MM-DD格式

\* 传入参数2：增加月份

\* 返回参数：返回参数为YYYY-MM-DD格式

\* \*/

public class AddMonths extends UDF{

public Date evaluate(Date date,int amount) {

Calendar calendar=Calendar.getInstance();

calendar.setTime(date);

calendar.add(Calendar.MONTH, amount);

return calendar.getTime();

}

}

## MONTHS\_BETWEEN

### 类名

org.apache.hadoop.hive.ql.udf. Months\_Between.java

### HIVE函数名

f\_months\_between

### 使用方法

实例：

select startdate,enddate,Months\_Between(To\_Date(startdate,'yyyy-MM-dd HH:mm:ss'),To\_Date(enddate,'yyyy-MM-dd HH:mm:ss')) from test;

结果：

OK

1970-02-01 00:00:00 2013-01-01 00:00:00 44.95

1970-01-01 00:00:00 2013-01-01 05:00:00 45.0

1970-05-01 00:00:00 2013-01-01 06:00:00 41.0

1970-06-01 00:00:00 2013-01-01 00:00:00 40.0

1970-01-01 00:00:00 2013-01-01 12:00:00 45.0

1970-07-01 00:00:00 2013-01-09 00:00:00 39.0

1970-08-01 00:00:00 2013-01-10 00:00:00 38.5

**特殊说明：**现在只能大的月份减去小的月份。就是开始月份一定要小于结束月份才能计算准确。传入两个参数开始日期和结束日期：

### Hive自定义函数代码

package org.apache.hadoop.hive.ql.udf;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Calendar;

import java.util.Date;

import org.apache.hadoop.hive.ql.exec.UDF;

import org.apache.hadoop.io.Text;

public class Months\_Between extends UDF {

public Text evaluate(Date start, Date end) throws ParseException {

Date start\_m = lastDayOfMonth(start);

long l = 0;

l = start\_m.getTime() - start.getTime();

long day = l / (24 \* 60 \* 60 \* 1000);

float months = 0;

System.out.println(day);

months = day / (float)getDays(start);

l = end.getTime() - getFirstDay(end.getMonth()).getTime();

day = l / (24 \* 60 \* 60 \* 1000);

months = months + day / (float)getDays(end)

+ (end.getMonth() - start.getMonth() + 1)

+ (end.getYear() - start.getYear());

return new Text(String.valueOf(months));

}

// 计算一个月的最后一天

public Date lastDayOfMonth(Date date) {

Calendar cal = Calendar.getInstance();

cal.setTime(date);

cal.set(Calendar.DAY\_OF\_MONTH, 1);

cal.roll(Calendar.DAY\_OF\_MONTH, -1);

return cal.getTime();

}

// 计算一个月的天数

public int getDays(Date date) {

Calendar rightNow = Calendar.getInstance();

rightNow.setTime(date);

int days = rightNow.getActualMaximum(Calendar.DAY\_OF\_MONTH);

return days;

}

// 计算一个月的第一天

public Date getFirstDay(int month) throws ParseException {

Calendar c = Calendar.getInstance();

int year = c.get(Calendar.YEAR);

String str = year + "-" + month + "-1";

Date d = null;

d = new SimpleDateFormat("yyyy-MM-dd").parse(str);

return d;

}

}

## LASTDAY

### 类名

org.apache.hadoop.hive.ql.udf. LastDay.java

### HIVE函数名

f\_last\_day

### 使用方法

实例：

Select startdate,

To\_Char\_Date(f\_last\_day(To\_Date(startdate,'yyyy-MM-dd HH:mm:ss')),'yyyyMMdd')

from test;

结果：

1970-02-01 00:00:00 19700228

1970-01-01 00:00:00 19700131

1970-05-01 00:00:00 19700531

1970-06-01 00:00:00 19700630

1970-01-01 00:00:00 19700131

1970-07-01 00:00:00 19700731

1970-08-01 00:00:00 19700831

**特殊说明：**传入一个日期格式的数据，返回一个日期格式的数据。：

### Hive自定义函数代码

package org.apache.hadoop.hive.ql.udf;

import java.util.Calendar;

import java.util.Date;

import org.apache.hadoop.hive.ql.exec.UDF;

/\*

\* 函数功能：获取传入时间所在月最后一天的日期

\* Hive自定义函数创建：平晓刚

\* 创建时间：2015-07-06

\* 传入参数：传入参数为YYYY-MM-DD格式

\* 返回参数：返回参数为YYYY-MM-DD格式

\* \*/

public class LastDay extends UDF{

public Date evaluate(Date date) {

Calendar cal = Calendar.getInstance();

cal.setTime(date);

cal.set(Calendar.DAY\_OF\_MONTH, 1);

cal.roll(Calendar.DAY\_OF\_MONTH, -1);

return cal.getTime();

}

}

## CharToDate

### 类名

org.apache.hadoop.hive.ql.udf. CharToDate.java

### HIVE函数名

f\_char\_to\_date

### 使用方法

实例：

Select f\_char\_to\_date(‘2015-05-06’,'yyyy-MM-dd') from test;

**特殊说明：**传入一个日期格式的数据，返回一个日期格式的数据。

日期格式化参数如下：

|  |  |  |
| --- | --- | --- |
| **字母** | **日期或时间元素** | **备注** |
| G | Era 标志符 |  |
| y | 年 | 常用 |
| M | 年中的月份 | 常用 |
| w | 年中的周数 |  |
| W | 月份中的周数 |  |
| D | 年中的天数 |  |
| d | 月份中的天数 | 常用 |
| F | 月份中的星期 |  |
| E | 星期中的天数 |  |
| a | Am/pm 标记 |  |
| H | 一天中的小时数（0-23） | 常用 |
| k | 一天中的小时数（1-24） |  |
| K | am/pm 中的小时数（0-11） |  |
| h | am/pm 中的小时数（1-12） |  |
| m | 小时中的分钟数 | 常用 |
| s | 分钟中的秒数 | 常用 |
| S | 毫秒数 |  |
| z | 时区 |  |
| Z | 时区 |  |

### Hive自定义函数代码

**package** org.apache.hadoop.hive.ql.udf;

**import** java.io.PrintWriter;

**import** java.text.ParseException;

**import** java.text.SimpleDateFormat;

**import** java.util.Calendar;

**import** java.util.Date;

**import** org.apache.hadoop.hive.ql.exec.UDF;

/\*

\* 函数功能：将字符串转换成日期类型

\* Hive自定义函数创建：平晓刚

\* 创建时间：2015-07-16

\* 传入参数1：字符串

\* 传入参数2：与参数1对应的字符串格式

\* 返回参数：日期

\* \*/

**public** **class** CharToDate **extends** UDF {

**public** Date evaluate(String date, String format) **throws** ParseException {

SimpleDateFormat sdfs = **new** SimpleDateFormat(format.toString());

Date date\_in = sdfs.parse(date.toString());

**return** date\_in;

}

}

## DateToChar

### 类名

org.apache.hadoop.hive.ql.udf. DateToChar.java

### HIVE函数名

f\_date\_to\_char

### 使用方法

实例：

Select f\_date\_to\_char(f\_char\_to\_date(‘2015-05-06 12:56:08’,'yyyy-MM-dd HH:mm:ss'),’dd’) from test;

**特殊说明：**传入一个日期格式的数据，返回一个日期格式的数据。

日期格式化参数如下：

|  |  |  |
| --- | --- | --- |
| **字母** | **日期或时间元素** | **备注** |
| G | Era 标志符 |  |
| y | 年 | 常用 |
| M | 年中的月份 | 常用 |
| w | 年中的周数 |  |
| W | 月份中的周数 |  |
| D | 年中的天数 |  |
| d | 月份中的天数 | 常用 |
| F | 月份中的星期 |  |
| E | 星期中的天数 |  |
| a | Am/pm 标记 |  |
| H | 一天中的小时数（0-23） | 常用 |
| k | 一天中的小时数（1-24） |  |
| K | am/pm 中的小时数（0-11） |  |
| h | am/pm 中的小时数（1-12） |  |
| m | 小时中的分钟数 | 常用 |
| s | 分钟中的秒数 | 常用 |
| S | 毫秒数 |  |
| z | 时区 |  |
| Z | 时区 |  |

### Hive自定义函数代码

**package** org.apache.hadoop.hive.ql.udf;

**import** java.text.ParseException;

**import** java.text.SimpleDateFormat;

**import** java.util.Calendar;

**import** java.util.Date;

**import** org.apache.hadoop.hive.ql.exec.UDF;

/\*

\* 函数功能：将日期转换成字符串

\* Hive自定义函数创建：平晓刚

\* 创建时间：2015-07-16

\* 传入参数1：日期内容

\* 传入参数2：与参数1对应的字符串格式

\* 返回参数：字符串

\* \*/

**public** **class** DateToChar **extends** UDF {

**public** String evaluate(Date str, String format) **throws** ParseException {

Calendar calendar = Calendar.*getInstance*();

calendar.setTime(str);

SimpleDateFormat sdfs = **new** SimpleDateFormat(format.toString());

String output = sdfs.format(calendar.getTime());

**return** output;

}

}