

1. ODS🡪DW

根据DW模型建表，并建立分区，分区字段为PART\_SC，PART\_DT（PART\_SC = 'source\_code',PART\_DT = '${END\_DATE}'），数据类型为STRING类型，PART\_DT 格式为**yyyy-MM-dd**。

按照下面的SQL脚本来写DW：

set hive.exec.dynamic.partition=true;

set hive.exec.dynamic.partition.mode=nonstrict;

INSERT

OVERWRITE TABLE **DW.XXX**

PARTITION (**PART\_SC**, **PART\_DT**)

SELECT

from\_unixtime(unix\_timestamp('${END\_DATE}', 'yyyy-MM-dd'), **'yyyyMMdd'**) AS PERIOD\_WID

…,

,SOURCE\_CODE AS **PART\_SC**

,'${END\_DATE}' AS **PART\_DT**

FROM ODS表

WHERE **主表更新日期字段 >= '${START\_DATE}'**

AND **主表更新日期字段 < '${END\_DATE}'**

1. DW🡪DWD

建立与DW层相同的DWD模型，并建立分区，分区字段为PART\_SC（PART\_SC = 'source\_code'），可根据业务需求增加副分区，格式PART\_XX，数据类型为STRING类型；建表命名规范：”DWD\_XX\_XX\_F。

按照下面的SQL脚本来写DWD：

set hive.exec.dynamic.partition=true;

set hive.exec.dynamic.partition.mode=nonstrict;

INSERT

OVERWRITE TABLE **DWD.XXX**

PARTITION (**PART\_SC)**

SELECT **a.xx1,**

**…,**

**a.xxxx**  AS **PART\_SC**

FROM DW.XXX **a**,

(SELECT 唯一键, MAX(PERIOD\_WID) MPERIOD\_WID

FROM DW.XXX

GROUP BY 唯一键) **b**

WHERE **a.唯一键 = b. 唯一键**

AND **a.PERIOD\_WID = b.MPERIOD\_WID**

1. DWD🡪DMR

根据DM模型建表，编写脚本可根据业务需求建立分区，修改PERIOD\_WID值。