# 问题描述

平时日增量不大,类似双十一活动期间会出现日增量暴增,如何设计合适的物理模型表来存储历史数据

参考方案:

平时按照月份分区,特定的日子按照天分区

**仅供参考:**

# 方案一:

1.如果每天的增量不是很大,那么可以考虑按照月份分区,但是特殊的月份例如11月,数据量会比较大;

2.如果想让每个分区内的数据尽可能均匀,并且不影响下游访问,有一种方案是,将etl程序分两条线,

2.1 先准备两张表,A:按照日期做范围分区(每个月一个分区), B: 按照日期做list分区

--设置A表的跑数周期是每天, 但是遇到特殊活动的时候将A表的加载任务暂停

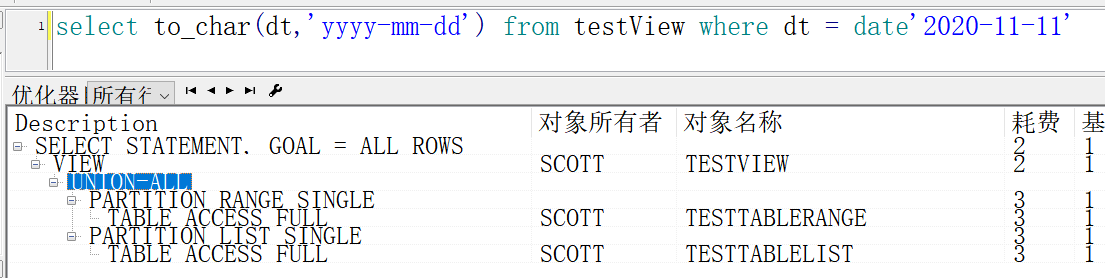
--设置B表的跑数周期是每天,但是平时不运行,只有在特殊活动的时候运行,用来替换A

2.2 这样A表的数据是按月分区, B表的数据是按天分区

2.3 创建一张视图,将A,B union all 起来,这样下游可以直接访问视图即可

缺点; 这样实现维护会比较麻烦,特别是还涉及到下游的作业依赖, 要进行切换. 但是高峰期每天的数据存在独立分区, 平时每个月的数据放在单独的分区,数据分布上应该会比较均匀,并且下游访问数据的时候比较方便

|  |
| --- |
| *平台: oracle11g, 开发工具 plsql-developer*  */\*drop table testTableList;*  *drop table testTableRange;\*/*  *--- A 表按照日期范围,每个月一个分区*  CREATE TABLE testTableRange  (dt date)  PARTITION BY RANGE (dt)  (  PARTITION p202009 VALUES LESS THAN (TO\_DATE('2020-10-01','YYYY-MM-DD')),  PARTITION p202010 VALUES LESS THAN (TO\_DATE('2020-11-01','YYYY-MM-DD')),  PARTITION p202011 VALUES LESS THAN (TO\_DATE('2020-12-01','YYYY-MM-DD')),  PARTITION pMax VALUES LESS THAN (MAXVALUE)  );  *--B 表按照日期,每天一个分区,主要用来处理业务高峰增量*  CREATE TABLE testTableList  (dt date)  PARTITION BY LIST (dt)  ( PARTITION p20201111 VALUES (date'2020-11-11'),  PARTITION p20201112 VALUES (date'2020-11-12'),  PARTITION p20201113 VALUES (date'2020-11-13'),  PARTITION p20201212 VALUES (date'2020-12-12'),  PARTITION p20200618 VALUES (date'2020-06-18')    );  *--测试数据*  insert into testTableList values (date'2020-11-11');  insert into testTableList values (date'2020-11-12');  insert into testTableList values (date'2020-06-18');  insert into testTableRange values (date'2020-11-01');  insert into testTableRange values (date'2020-11-02');  insert into testTableRange values (date'2020-11-10');  commit;  *--查看分区数据*  select \* from testTableList partition( p20201111);  select \* from testTableRange partition( p202011);  *-- 查看分区*  SELECT \* FROM USER\_TAB\_PARTITIONS WHERE TABLE\_NAME= upper('testTableList');  SELECT \* FROM USER\_TAB\_PARTITIONS WHERE TABLE\_NAME= upper('testTableRange');  *--创建视图*  create view testView as  select dt from testTableRange union all  select dt from testTableList;  *-- 测试脚本,通过执行计划查看是否有通过分区筛选数据*  select to\_char(dt,'yyyy-mm-dd') from testView where dt >= date'2020-11-01' |



# 方案二:

如果不想来回切换表,可以用一张表实现

1.创建表A: 按照yearMonthFlag进行list分区,

1.1特殊活动日期 yearMonthFlag 格式: yyyyMMDD ,例如: 20201111--用来表示当天的数据分区

1.2淡季 yearMonthFlag 格式:yyyyMMDDxx, 例如 20201100 -- 用来表示整个月的数据(除了特殊活动日)

|  |
| --- |
| *-- 创建分区表,按照 yearMonthFlag list分区*  *-- drop table testTableList2;*  CREATE TABLE testTableList2  (dt date,  yearMonthFlag int)  PARTITION BY LIST (yearMonthFlag)  (  PARTITION p20201100 VALUES (20201100),  PARTITION p20201111 VALUES (20201111),  PARTITION p20201112 VALUES (20201112),  PARTITION p20201212 VALUES (20201212),  PARTITION p20200618 VALUES (20200618)  );  *--导入数据*  INSERT INTO testTableList2  SELECT SS.DT  ,(CASE WHEN SS.DT IN(DATE'2020-11-11', DATE'2020-11-12', DATE'2020-12-12',DATE'2020-06-18' )  THEN TO\_NUMBER(TO\_CHAR(DT, 'YYYYMMDD'))  ELSE to\_number( TO\_CHAR(DT, 'YYYYMM')||'00' )  end ) as yearMonthFlag  FROM  (  SELECT DATE'2020-06-18' AS DT FROM DUAL UNION ALL  SELECT DATE'2020-11-01' AS DT FROM DUAL UNION ALL  SELECT DATE'2020-11-02' AS DT FROM DUAL UNION ALL  SELECT DATE'2020-11-03' AS DT FROM DUAL UNION ALL  SELECT DATE'2020-11-11' AS DT FROM DUAL UNION ALL  SELECT DATE'2020-11-12' AS DT FROM DUAL) SS;  COMMIT;  *-- 查看分区*  SELECT \* FROM USER\_TAB\_PARTITIONS WHERE TABLE\_NAME= upper('testTableList2');    *--查看分区数据*  select \* from testTableList2 partition( P20201100);  select \* from testTableList2 partition( p20201111);  *-- 测试脚本,通过执行计划查看是否有通过分区筛选数据*  *--查看11.11 当天数据*  select \* from testTableList2 where yearMonthFlag = 20201111;    *--查看11月份除了 11.11/ 11.12 的数据*  select \* from testTableList2 where yearMonthFlag = 20201100;    *--查看整个11月的数据*  select \* from testTableList2 where yearMonthFlag >= 20201100 and yearMonthFlag < 20201201; |