

How to create Point-in-Time (PIT) in WhereScape RED

By Raphael Klebanov, WhereScape USA

1. Overview of Point-in-Time structure

Point-in-Time (PIT) considers a non-core Data Vault objects and used for one reason only – performance. It is defined as “A structure which sustains integrity of joins across time to all the Satellites that are connected to the Hub [or Link]”⁽¹⁾ WhereScape has successfully implemented Point-in-Time structures on customers’ sites.

PIT table is a specialized Satellite extension that is used to get the latest row “AS OF” a specific datetime without use of nested sub-queries in the main satellite query. Example of such “nested” query as follows with the use of *AS_OF_datetime* parameter:

```
SELECT count(*)
FROM HUB_TRADING_PARTNER hub,
     SAT_TRADING_PARTNER_DISTR sat1,
     SAT_TRADING_PARTNER_BILLING sat2,
     SAT_TRADING_PARTNER_SHIPPING sat3
WHERE hub.HUB_TRADING_PARTNER_KEY = sat1.HUB_TRADING_PARTNER_KEY
AND sat1.load_begin_datetime <=
     (select max(sat11.load_begin_datetime) from SAT_TRADING_PARTNER_DISTR sat11
      where sat1.HUB_TRADING_PARTNER_KEY = sat11.HUB_TRADING_PARTNER_KEY
      and sat11.load_begin_datetime <= CONVERT(datetime,'$PAS_OF_datetime$'))
AND hub.HUB_TRADING_PARTNER_KEY = sat2.HUB_TRADING_PARTNER_KEY
AND sat2.load_begin_datetime <=
     (select max(sat22.load_begin_datetime) from SAT_TRADING_PARTNER_DISTR sat22
      where sat2.HUB_TRADING_PARTNER_KEY = sat22.HUB_TRADING_PARTNER_KEY
      and sat22.load_begin_datetime <= CONVERT(datetime,'$PAS_OF_datetime$'))
AND hub.HUB_TRADING_PARTNER_KEY = sat3.HUB_TRADING_PARTNER_KEY
AND sat3.load_begin_datetime <=
     (select max(sat33.load_begin_datetime) from SAT_TRADING_PARTNER_DISTR sat33
      where sat3.HUB_TRADING_PARTNER_KEY = sat33.HUB_TRADING_PARTNER_KEY
      and sat33.load_begin_datetime <= CONVERT(datetime,'$PAS_OF_datetime$'));
```

Note that if no rows exist for any satellite table, and a date is given that is EARLIER than the first row made for that satellite, an equal-join will return no results. Also, performance of such query will suffer because of multiple tables being joined.

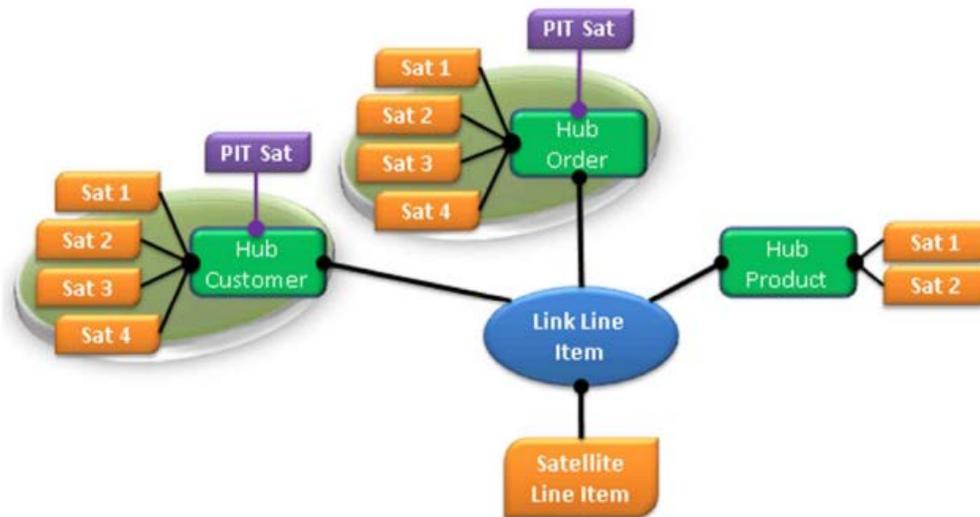


Figure 1 PIT Table Architecture Overview⁽¹⁾

While it is not shown on the figure above, one may add PIT tables to help join Link Satellites as well. In general, PIT tables are not added and not usually needed until there are three or more Satellites “hanging” off a parent -- Hub or Link -- that are loaded at different frequencies. PIT should not span over several Hubs or Links. For that Data Vault is using a Bridge table that can stretch over multiple Hubs or Links. While PIT structure is quite common in Data Vault, Bridge table (very similar to a high level factless fact table) is more difficult to maintain and rather rare in the Data Vault EDW.

The main functionality of the RED-generated store procedure will be the following INSERT statement:

```
INSERT INTO pit_trading_partner WITH (TABLOCK)
(
  hub_trading_partner_key,
  load_datetime,
  tp_shipping_load_beg_datetime,
  tp_billing_load_beg_datetime,
  tp_distr_load_beg_datetime
)
SELECT
  hub_trading_partner.hub_trading_partner_key,
  load_datetime,
  sat_trading_partner_shipping.load_beg_datetime,
  sat_trading_partner_billing.load_beg_datetime,
  sat_trading_partner_distr.load_beg_datetime
FROM hub_trading_partner
JOIN sat_trading_partner_shipping
  ON hub_trading_partner.hub_trading_partner_key =
    sat_trading_partner_shipping.hub_trading_partner_key
  AND @v_load_datetime BETWEEN sat_trading_partner_shipping.load_beg_datetime
    AND sat_trading_partner_shipping.load_end_datetime
JOIN sat_trading_partner_billing
  ON hub_trading_partner.hub_trading_partner_key =
    sat_trading_partner_billing.hub_trading_partner_key
  AND @v_load_datetime BETWEEN sat_trading_partner_billing.load_beg_datetime
    AND sat_trading_partner_billing.load_end_datetime
JOIN sat_trading_partner_distr
  ON hub_trading_partner.hub_trading_partner_key =
    sat_trading_partner_distr.hub_trading_partner_key
  AND @v_load_datetime BETWEEN sat_trading_partner_distr.load_beg_datetime
    AND sat_trading_partner_distr.load_end_datetime
```

Using WhereScape RED, one can schedule the snapshot process as frequently as you need it, e.g. hourly. Load of all PIT tables can be done in parallel and as the very last step in the WhereScape Scheduler Data warehouse processing job.

It is recommended to use variety of database means to increase efficiency of PIT table, such as: database columnar compression where available; applying partitioning on PIT tables, removing/archiving old records from the PIT tables and so on.

⁽¹⁾*Supercharge Your Data Warehouse* by Dan Linstedt 2010-2011