Data Warehousing: Lab 3 Dashboarding

Physical Data Warehouse Design

Ricard Monge Calvo, Vicent Perez Gregori

January 2020

1. Proposed SQL queries

First KPI:

```
SELECT d1.monthID, SUM(f.flightHours) FH, SUM(f.flightCycles) FC
FROM TemporalDimension d1, AircraftDimension d2, AircraftUtilization f
WHERE f.aircraftID = d2.ID AND f.timeID = d1.ID AND d2.model = '737'
GROUP BY d1.monthID ORDER BY d1.monthID
```

Second KPI:

```
SELECT d3.y, SUM(f.scheduledOutOfService) ADOSS,
SUM(f.unScheduledOutOfService) ADOSU
FROM TemporalDimension d1, Months d3, AircraftUtilization f
WHERE f.aircraftID = 'XY-KKF' AND f.timeID = d1.ID AND d1.monthID = d3.ID
GROUP BY d3.y ORDER BY d3.y
```

Third KPI:

```
SELECT T_AIR.monthID,
    SUM(T_LOGS.CNT)/  SUM(T_AIR.FC) RRc,   SUM(T_LOGS.CNT)/  SUM(T_AIR.FH) RRh,
    SUM(T_LOGS.PIREP)/SUM(T_AIR.FC) PRRc,  SUM(T_LOGS.PIREP)/SUM(T_AIR.FH) PRRh,
    SUM(T_LOGS.MAREP)/SUM(T_AIR.FC) MRRc,  SUM(T_LOGS.MAREP)/SUM(T_AIR.FH) MRRh
FROM (SELECT t.monthID, SUM(f.flightHours) FH, SUM(f.flightCycles) FC
        FROM AircraftDimension d, AircraftUtilization f, TEMPORALDIMENSION t
        WHERE f.aircraftID = d.ID AND f.TIMEID = t.ID AND d.model = '737'
        GROUP BY t.monthID ORDER BY t.monthID) T_AIR,
    (SELECT L.monthID, SUM(L.COUNTER) AS CNT,
                SUM(CASE WHEN p.role='P' THEN L.COUNTER ELSE 0 END) AS PIREP,
                SUM(CASE WHEN p.role='M' THEN L.COUNTER ELSE 0 END) AS MAREP
        FROM LOGBOOKREPORTING L, PeopleDimension p, AircraftDimension d
        WHERE L.aircraftID = d.ID AND L.PERSONID = p.ID AND d.model = '737'
        GROUP BY L.monthID ORDER BY L.monthID) T_LOGS
WHERE T_AIR.monthID = T_LOGS.monthID GROUP BY T_AIR.monthID ORDER BY T_AIR.monthID
```

Fourth KPI:

```
SELECT T_AIR.model, SUM(T_LOGS.MAREP)/SUM(T_AIR.FC) MRRc,
    SUM(T_LOGS.MAREP)/SUM(T_AIR.FH) MRRh
FROM ( SELECT d.model, SUM(f.flightHours) FH, SUM(f.flightCycles) FC
        FROM AircraftDimension d, AircraftUtilization f
        WHERE f.aircraftID = d.ID GROUP BY d.model ORDER BY d.model) T_AIR,
    (SELECT d.model, SUM(CASE WHEN P.role='M' THEN L.COUNTER ELSE 0 END) AS MAREP
        FROM LOGBOOKREPORTING L, PeopleDimension P, AircraftDimension d
        WHERE L.aircraftID = d.ID AND L.PERSONID = P.ID AND P.airport = 'BSL'
        GROUP BY d.model ORDER BY d.model) T_LOGS
WHERE T_AIR.model = T_LOGS.model GROUP BY T_AIR.model ORDER BY T_AIR.model
```

2. Proposed Data Structures and Assumptions

Four additional data structures were created in order to improve the access cost given the KPIs specified in the previous section. Those data structures are:

1. Bitmap Join Index for AIRCRAFTUTILITZATION over AIRCRAFTDIMENSION.MODEL

   When creating a Bitmap over a table, having a unique id for each instance is a requirement. So, a unique constraint was added to the Dimensional table AircraftDimension on the variable ID, with the associated B+ tree (not used in the plans).

   This data structure was created with the purpose of improving the cost of the first KPI, by following the criterion that we would avoid performing the join between the tables AircraftUtilization and AircraftDimension and speed up the filtering by aircraft model.

2. Bitmap Index over the attribute AIRCRAFTID of the dimensional table AIRCRAFTUTILIZATION

   The KPI cost that carried to building this data structure was the improvement corresponding to the second KPI. The reason was to speed up the selection process of an aircraft of the fleet on the filtering step. For this step, we did not choose to build a B+ tree, that would be faster due to the high selectivity of taking a given aircraft, due to the 1900 disk blocks space constraint.

3. Bitmap Join Index for LOGBOOKREPORTING over AIRCRAFTDIMENSION.MODEL

   The purpose of creating this data structure is mainly due to the high cost related to the third KPI. It was considered to avoid the joins with dimension tables in the filtering of LOGBOOKREPORTING fact table. Moreover, the first mentioned Bitmap Join index for AIRCRAFTUTILIZATION fact table over AIRCRAFTDIMENSION.MODEL produces a speed up when performing the selection of that fact table.

4. Bitmap Join Index for LOGBOOKREPORTING over PeopleDimension.AIRPORT

   Another uniqueness constraint was needed in order to create this additional data structure. This time it was created over the attribute ID of the dimensional table PeopleDimension.

   This last data structure was originated due to the high cost related to the fourth KPI. We considered it interesting because it would save us of performing the join between the dimensional table PeopleDimension and the fact table LOGBOOKREPORTING.

Bitmaps turned out to be the best option to optimize the access plan due to their small disk cost. In particular, bitmap join indexes allowed us to avoid joins with dimension tables to filter by the dimensions' higher hierarchies. In addition, we could take advantage of them because of the high cardinality of the data in the fact tables for the attributes considered. Furthermore, although the selectivity factor of the model and aircraft selections is really high, we kept the bitmap indexes over B+ indexes due to the disk space constraints.

Contrarily to the Mondrain Schema used in the Dashboarding part, we decided not to use any FACTS_DRILLACROSS view of the Data Warehouse fact tables. The reason for this is that we achieved a lower cost with the previously mentioned queries and were able to better find structures to optimize the non-view versions of the KPIs.