

Data Warehousing: Lab 1 Multidimensional Modelling

Maria Gkotsopoulou, Ricard Monge Calvo

October 2019

1. Conceptual design

To design our multidimensional model we look at our sources (data driven approach): AIMS for flight data, cancellations and delays; and AMOS for maintenance and logbook data. In addition, we consider the KPIs that need to be served using the model (demand driven approach).

We identify three main facts and model a star schema for each of them:

- Flights details with departure date and plane as dimensions, with takeoffs, cancellation, duration and delay/delay_duration as measures.
- Maintenance events with its associated type (un/scheduled), date and plane as dimensions and duration as a measure.
- Logs with each entry of the technical logbook with its associated personnel type, date and plane dimensions and count as a measure.

By assuming that each flight or maintenance fact will always have its associated log, and that cancelled or delayed flights also have an associated maintenance event, we conclude that the dates and planes dimensions of each fact have a large enough number of intersections and thus it is logical to join them, obtaining a constellation schema.

On the other hand, since the finest granularity of our KPIS is day we aggregate all the information of our fact tables at the day level. This means that measures such as the flight's takeoffs, maintenance's duration or logs' counts are aggregates for a given day and a given plane, which we assume our ETL provides.

2. Materialized views

First, we need to decide whether we would benefit from creating, if any, materialized views. Once we have written the queries needed to obtain all the KPIs specified (given our demand driven approach) we observe that there are some core similarities in terms of grouping attributes; namely month(or year) and aircraft (or model/ manufacturer). Specifically, we see a potential gain in creating materialized views for the KPIs associated with maintenance (ADOS, ADOSS, ADOSU and adding schedule_type) and logs (RR, PRR, MR and adding personnel_type). We consider that the tuple reduction of $365 * \#aircrafts$ to $12 * \#aircrafts$ is considerable and justifies the view creation.

On the other hand, it is just FH (flight hours) and TO (flight cycles) that are required on a daily level but they can be queried directly from the table. However, for the flight KPIs (cancellation, delays, delay_duration) we believe that we would obtain a significant tuple reduction by creating a view on a month aircraft level, as stated previously.

In our materialized views query definition we specify the `ENABLE QUERY REWRITE` so that the system decides to run a query against a view instead of a table. Based on the assumption that our ETL will run daily and feed into our DW any past modifications or new data for a given day, we believe that the choice of incremental updates is to be preferred over a complete update. In particular, we use Oracle's `FAST REFRESH ON COMMIT` with the associated materialized view log on each table used in the view creation.