

In order to decide how to structure our multidimensional schema, we first checked which were the measures of interest and their respective levels of aggregation needed to answer the given queries.

Afterwards, we created a constellation schema with three fact tables: *AircraftsInfo*, *MaintenanceReports* and *FlightsInfo*, whose names correspond to the KPIs they contain. *AircraftsInfo* contains the measures: ADIS, ADOS, ADOSS, ADOSU, DYR, CNR, TDR, ADD, RRh, RRC, PRRh and PRRc.

*MaintenanceReports* has two measures: MRRc and MRRh, and an attribute, Airport, because we don't think it is justifiable to create an extra dimension just for Airport.

*FlightInfo* contains the measures: TO and FH.

Next, we created three dimensions, *Time*, *TimeDay* and *Aircraft*, allowing us to aggregate data depending on the month or year, the day, month or year and the aircraft registration, model or manufacturer, respectively.

The established relations in the schema were made to respond to the asked queries. The first fact schema is linked to dimensions of Time and Aircraft, the second to Aircraft only and the third one to TimeDay and Aircraft.

Once we had the constellation drawn in Indycos, we converted the multidimensional schema into a relational schema through the SQL statements found in the '*CreateStatements.sql*' file.

To do the project, we were given the frequencies of refreshment of the different KPIs. However, we were not provided with the frequencies of use of the queries (how often they are called). Therefore, we chose criteria that let us know which would be the most used. In particular, we focused on optimising aircraft usage to create the necessary materialised views.

Notice that only *AircraftsInfo* and *FlightsInfo* provide information about an aircraft's usage. Now, in the 'data flows frequency' table, we can see that TO and FH are updated weekly, implying the *FlightsInfo* table is refreshed weekly. On the other hand, the other measures belong to *AircraftsInfo*. This table is updated every week as it contains maintenance related measures that are updated weekly, despite aircraft usage measures being updated monthly.

Given that information, we have created a materialised view, *Flights*, that selects TO and FH for a given aircraft, day, month and year. Initially, it should be updated incrementally because there are no aggregate functions, meaning we won't modify any preexisting tuples of the view. However, it is set to FORCE update meaning the DBMS will choose the quickest update method. It is updated on commit, so each time the *FlightsInfo* is updated, so is *Flights*.

*AircraftsUsage* is the second view we have created, selecting ADIS, ADOS, ADOSS, ADOSU, DYR, CNR, TDR and ADD by aircraft, month and year. It also uses FORCE update. However, it is updated on demand because we are only interested in updating once the measures above have been added to the fact table, monthly.