DataMart Design and Querying (OLAP)

Design a data warehouse to manage a hotel chain and to address the described issues.

# Problem specifications

A big hotel chain has over 500 hotels of different categories all over the world. Every day in the data base of each hotel, information on free, reserved, and unavailable rooms and corresponding customers is stored. Rooms may be unavailable due to maintenance.

The hotel chain managers would like to build a data warehouse to analyze room types and income. The hotel chain managers would like to know for each hotel every day:

* the percentage of reserved rooms
* the percentage of free rooms
* the percentage of unavailable rooms

The income and the percentage of rooms must be known according to:

* geographical location of the hotel (state, region, and province)
* hotel category (5 stars, 4 stars, …)
* room features (number of beds, TV, whirlpool bath, …)
* date, day of the week, month, year

The hotel chain managers would like to analyze the daily, holiday, monthly and yearly income. Some **frequent queries** the managers would like to execute are the following.

* 1. In 2008, for each state and month, analyze the portion of rooms which are reserved, free, and unavailable.
  2. In 2008, for each state, analyze the portion of rooms which are reserved.
  3. In 2008, for each state and month, analyze the income of 4-star hotels.

# Design and OLAP Tasks

1. Derive a conceptual data warehouse design (DFM) according to the specifications; in particular, the designed data warehouse must promptly answer to all the frequent queries.

The following information is also known:

* + the data warehouse has the information of 2005 and 2010
  + number of states: 40
  + number of cities: 400
  + number of hotels: 500
  + number of different room features: 8

1. Map the derived DFM to relational logical design (ROLAP).
2. Use the extended SQL language to answer frequent queries (a, b) and (c).
3. Considering the designed data warehouse and its cardinalities, decide whether and which materialized views are convenient to improve response time of the frequent queries (consider all the frequent queries, **a to c**). Explain the reasons for your choices.