Data Base and Data Mining Group of Politecnico di Torino

Data Science And Database Technology

Politecnico di Torino - School of Information Engineering
Master of Science in Computer Engineering

Design a data warehouse to analyze the business of an international parcels service with more than 200 branches all over the world, addressing the following issues.

Problem specifications

The branches manage dispatches of different products for many third party companies. The current information system is heterogeneous, thus each branch has its own data for its own business activities.

Each dispatch is handled by a single branch. The dispatch consists of one or more routes and every route has a departure and a destination place (city, province, region, and state).
The management needs to analyze the global flows of delivered goods to decide which branches to expand or reduce and to take strategic business decisions.

The analysis of the good flows is performed taking into account the income, the weight (expressed in kg), and the volume of the delivered goods. In particular, the management is interested in analyzing the income of each branch and in comparing it to the income of each district to which the branch belongs. A further analysis is performed on the profitability for different categories of goods and for different carrier types (air, rail, sea freight, etc).

To decide which branches to expand or reduce, the management needs to analyze the shipping routes in terms of income, volume, and weight of the delivered goods.

Eventually, the management needs to analyze the average income, the average weight, and the average volume of goods delivered in different years, semesters, 4-month periods, trimesters, 2-month periods, months, days of the month and days of the week.

The following are some of the frequent queries the management is interested in:

a) Considering only Italian routes, for each category of goods and for each year, select the average daily income for each month and the total monthly income since the beginning of the year.
b) Considering only air carriers, select the yearly average income per unit of volume for each destination province, and the percentage of such income compared to the yearly average income per unit of volume of the destination state.
c) In 2006 for each route, in terms of departure and destination region, select the monthly average income per unit of weight (in kg) of the delivered goods, and the average daily income for each month of the goods delivered on that route.
d) For each branch district and carrier type, select the total income for each month and the total volume of goods delivered in each month. Rank the results according to the total monthly volume (the highest is 1st).

e) In 2005, for each route, in terms of departure and destination city, select the yearly average income per unit of weight of the delivered goods, and the average daily income for each considered route.
f) In 2005 and 2006, considering only shipping by sea, select the half-year average income per unit of volume for each branch and for each departure region, and the half-year average income per unit of weight.
g) For each departure city and for each branch district, select the 4-month period income and the total volume of delivered goods for each 4-month period.

Design

The data warehouse will store information of 2004, 2005, 2006 and 2007. The following cardinalities are known (suppose data is uniformly distributed):

- Good categories: ~20
- Branches: ~200
- Branch districts: ~50
- Different carrier types: ~5
- Cities: ~1000
- Provinces: ~200
- Regions: ~100
- States: ~20

1. Design the data warehouse to address the described issues. In particular, the designed data warehouse must allow efficient execution of **all** the queries described in the specifications.
2. Write the frequent queries **(a)**, **(c)** and **(d)** of the "problem specifications" using the extended SQL language.
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). Explain reasons for your choices.