Université libre de Bruxelles
Faculté des Sciences Appliquées

January 26, 2024

INFO-H-419 – Data Warehouses

First session examination

Question 1: Data Warehouse Design (8 points)

Spotify wants to design a data warehouse to better understand the listening behaviour of its users. The data warehouse contain information describing the users, the artists, the music, the records, and the record labels. Information about the users include age, location, gender, etc. Artists can be creators (e.g., the composer Sergei Rachmaninoff) or interpreters (e.g., London Symphony Orchestra or the conductor Valery Gergiev) and they produce records, which can be albums or songs (e.g., Rachmaninoff Symphonies by London Symphony Orchestra conducted by Valery Gergiev). There are many music genres such as classical, jazz, latin, etc. In addition there are record labels such as Deutsche Grammophon for classical music or ECM for jazz.

1. Design a conceptual schema for the data warehouse. Propose dimension attributes and dimension hierarchies.
2. Translate the conceptual schema into a relational schema. Clearly indicate primary and foreign keys in your tables.
3. For the relational schema obtained in the previous question, write **five** different queries exploiting various characteristics of the information stored in the data warehouse. Express the queries first in English and then in SQL.

For this question you can base on your own experience of using Spotify or similar music services. Obviously, since the application domain is vast, focus on producing a minimal subset of functionality that you want to obtain in your solution. Please state clearly the hypothesis that you are taking in your solution.

Question 2: OLAP Querying (6 points)

The Research and Innovative Technology Administration (RITA) coordinates the US Department of Transportation's (DOT) research programs. It collects several statistics about many kinds of transportation means, including the information about flight segments between airports summarized by month. A logical schema of a data warehouse containing this information is given in Fig. 1.

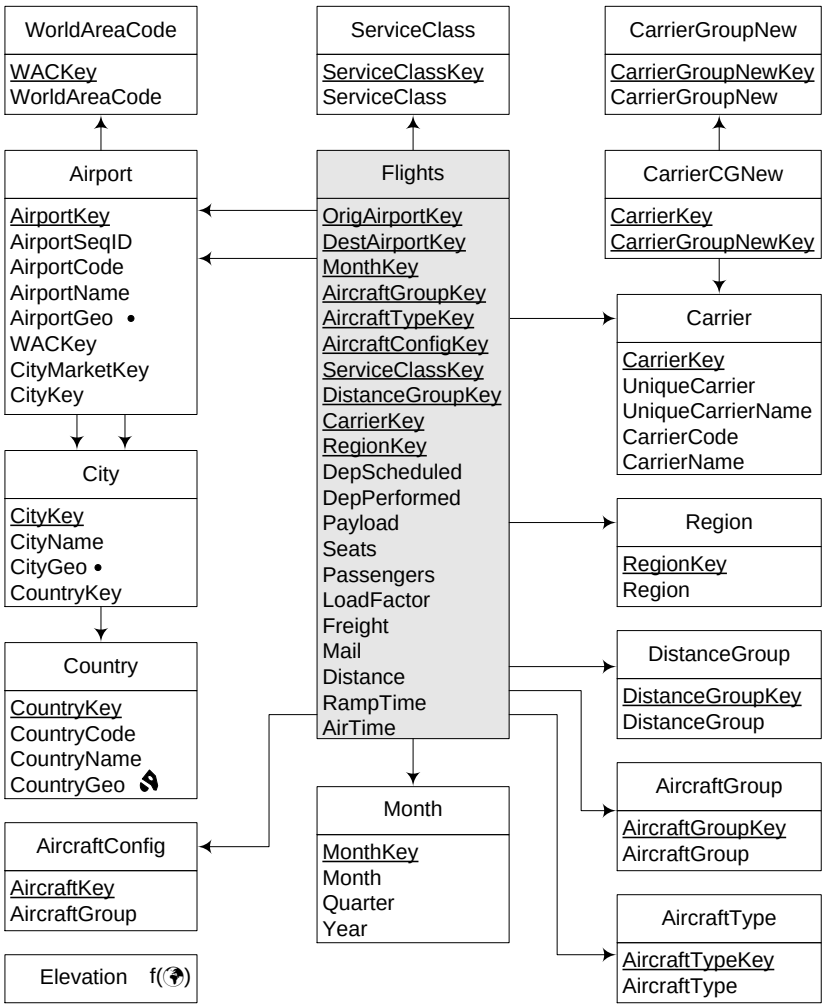

Figure 1: A logical multidimensional schema for the AirCarrier application

Write in SQL the following queries.

1. For each carrier and distance group, give the total number of seats sold in 2022.
2. Display for each city the three closest airports and their distance to the city independently of the country in which the city and the airport are located.
3. Give the total number of persons arriving to or departing from airports closer than 15 km from the city center in 2022.
4. Give for 2022 the ratio between the number of persons arriving to or departing from aiports closer than 15 km from the city center and the number of persons arriving to or departing from airports located between 15 and 40 km from the city center.
5. For cities operated by more than one airport, give the total number of arriving and departing passengers.
6. For cities operated by more than one airport, give the total number of arriving and departing passengers at the airport closest to the city center, and the ratio between this value and the city total.

Question 3: View Materialization (4 points)

Consider the AirCarrier data warehouse of Question 2.

1. Design a (partial) lattice of the data cube where each node represents a view.
2. Assign a realistic number of tuples to each node supposing that the fact table contains 1M rows.
3. Assuming that the bottom of the lattice (that is, the fact table) is materialized, determine using the View Selection Algorithm the **two** views to be materialized first.

Question 4: Indexing (4 points)

Consider the AirCarrier data warehouse of Question 2.

1. Propose a realistic indexing schema for the data warehouse that includes one index from each category that we have seen in the course.
2. For each of the index types proposed in (1), explain in detail the execution plan of a given query **with and without** the index. You can choose the queries given in Question 2 or any other query that you consider relevant for the application.