INFS3200 Sem 1, 2025

Prac2 Data Warehouses

Introduction

Learning Objectives:

- Learn how to create a cube and its dependent components in OLAP. The tasks include:
  - To identify and build the dimensions,
  - To define measures (both stored and calculated facts).
- Learn how to view an OLAP dataset.
- Understand how data is organized and stored in dimensions and data cubes,
- Learn how to query a data cube using SQL queries.
- Understand how materialized views are used in OLAP.

Part 1: Data Warehouse Setup

You are given a Data Warehouse, namely GO which has a star schema. GO was sourced from a transactional database system, which contains data for an outdoor equipment store. The followings are the description of the tables (including two dimension tables and one fact table), intended for analysis that have gone through the ETL (Extraction, Transformation, and Loading of heterogeneous data) process:

Table	**Description**
Go_products	This table stores the products information, including hierarchical attributes (Product line -> Product type -> Product) and prices.
Go_retailers	This table stores retailer information, including hierarchical attributes (Retailer name -> Country).
Sales_Fact	This is fact table that stores sales quantity, retailer, and products.

Download the prac2 material from blackboard and upload to your zone.

Identifying Dimensions

Using the source data tables as the primary input, the following dimensions are identified as requirements for the OLAP data model:

- Product
- Retailer

Each of the dimensions contains hierarchical structure, the description is presented in table 1.

Identifying Measures

You can have two types of measures in a cube: *Stored* (or Base) measures, and *Calculated* measures. In this prac, we have

- Quantity (Stored measures)
- Total Gross Margin (Calculated measures)
  - Formula: Quantity * (Unit price – Unit cost)

The calculated measure will be defined during the creation of cubes, which will be introduced later. After all dimensions are defined, we are ready to create our cube. Execute the provided script "create_cube.sql. The results will be stored in a materialized view named "sales_cube".

Consider the hierarchical structure in dimension table 'go retailer', perform rollup to summarize total sales quantity information for store level and country level. Execute the provided script "retailer_sales_rollup.sql." The results will be stored in a materialized view named "retailer_sales".

Part 2: Query Data Cube via SQL

The views you created for data cube is similar to traditional table-based star models. However, there are two key differences:

- Fact tables in a star schema store detailed data, while the cube views reveal many summary levels.
- In addition to the facts in fact tables, the cube provides additional measures and calculations, which are calculated and materialized as columns in the cube view.

These differences impact the way you query data. With star queries, you aggregate the data by combining aggregation functions (such as sum) and the GROUP BY clause. With cube queries, if the cube has been fully calculated (cube view is fully materialized), you simply select the data you want (either stored or calculated measures) as a column. Typically, no aggregation function is necessary since the data has already been summarized by the cube.

Task:

1. Find the top-3 country with the highest gross margin.
   a. Note: Please compare the cost between querying from tables and querying from the materialized view named "retailer_sales".
2. Find the country that sold the highest number of watch ("product type").
   a. Note: Please construct a cube that contains the required product, retailer, and sales information.

b. Note2: Please compare the cost between querying from tables and querying from the materialized view of the created cube.