TECHNISCHE UNIVERSITÄT WIEN

TU Wien

Business Intelligence VU

188.429

Assignment 2: ETL, Cube Modeling Analytics (SQL/MDX/Atoti)

Group: 006

Student A: Kerim Halilović, 12434665
Student B: Nikola Lukić, 12127674

Date of Submission: November 30, 2025

1 ETL Summary

Implementation was straightforward as the schema was provided. We populated dimensions using the defined logic and handled the Fact table (`ft_param_city_month`) using a sequence of Common Table Expressions (CTEs):

- **Transformation:** We mapped text-based colour alerts to numeric ranks (1-4) and aggregated readings to the daily grain to identify daily peaks.
- **Aggregation:** Data was rolled up to the monthly grain to calculate measures like `exceed_days_any`.
- **Keys:** We generated the primary key using `ROW_NUMBER()` to ensure uniqueness across the Month-City-Param grain.

2 Answers to Business Questions

We implemented 10 questions in SQL and MDX for comparison, plus 5 interactive questions (Q11-Q15) in Atoti.

Selected Questions

- **SQL – Student A:** Q01, Q03, Q05, Q07, Q09
- **MDX – Student A:** Q02, Q04, Q06, Q08, Q10
- **SQL – Student B:** Q02, Q04, Q06, Q08, Q10
- **MDX – Student B:** Q01, Q03, Q05, Q07, Q09

Observations on Implementation

- **Pivoting Time (Q01, Q04, Q13):** SQL required verbose `CASE WHEN` statements to pivot months or quarters into columns. MDX was superior here; we simply placed the time hierarchy members (e.g., `[Jan]`, `[Feb]`) on the Columns axis.
- **Top-N Ranking (Q06, Q09, Q10, Q14):** Finding "Top 10" sets was significantly more efficient in MDX using `TOPCOUNT`. In SQL (specifically Q09), we had to write complex CTEs to rank countries before joining back to pivot the data, whereas MDX handled the ranking and projection in a single nested function.
- **Hierarchy vs. Flat Strings (Q05, Q11):** SQL treated the `category` column as flat strings, grouping "Gas" into a single row. MDX respected the hierarchy path defined in Atoti, splitting "Gas" by its parent Purpose (Comfort vs. Health Risk), which accurately reflected the multidimensional design.
- **Filtering and Sets (Q02, Q08, Q12, Q15):** SQL filtering is intuitive with `WHERE`. In MDX, filtering was tricky; we had to use `DESCENDANTS` to navigate from Country to City (Q02) and Sub-Selects to fix the Year (Q11, Q15) to avoid "hierarchy in multiple axes" errors.
- **Complex Aggregations (Q07, Q14):** Calculating percentiles (Q07) in SQL required specific window functions (`PERCENTILE_CONT`). In Atoti/MDX, this complexity was abstracted away in the cube definition, making the query itself much cleaner.

3 Reflection and Lessons Learned

Student A: Kerim Halilović

The ETL process showed why having a clear star schema (no pain of designing it like in Assignment 1) is important because it made writing the loading scripts easier. Manual tasks like fixing the region mappings were tedious.

I also found working with Atoti helpful. In SQL, calculating averages often requires careful grouping to avoid statistical errors. In Atoti, we only had to define the aggregation rule once during the setup, and the tool handled the calculations automatically. This showed me that setting up the cube correctly at the beginning saves time during the analysis.

Student B: Nikola Lukić

Learning MDX was difficult for me because I am used to SQL. Since I usually think in terms of rows and tables, concepts like tuples and sets were hard to grasp at first. I spent a lot of time fixing errors because I did not fully understand how axes worked.

However, I realized the value of MDX when working on Question 09. In SQL, I had to write a long query to rank the countries and then pivot the data. In MDX, the same task took only a few lines using the TopCount function. It is harder to learn than SQL, but I can see that it is more efficient for complex analytical queries.