TECHNISCHE UNIVERSITÄT WIEN

TECHNISCHE UNIVERSITAT WIEN

Business Intelligence VU

188.429

Assignment 1: Dimensional Modelling and ETL

Group: 006

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

Date of Submission: October 30, 2025

1 Synthetic Tables (Table_X and Table_Y)

To enrich the provided OLTP data, we introduced two synthetic tables to model the impact of environmental campaigns on air quality and operational metrics.

1.1 Table_X: `tb_environmental_campaign`

This table stores the details of all environmental, regulatory, or awareness campaigns. Each row represents a unique initiative, defining its scope, duration, and budget.

Table 1: Field Descriptions for `tb_environmental_campaign`

Column Name	Data Type	Description
campaign_id	INTEGER	Surrogate primary key.
campaign_name	TEXT	The official name of the campaign.
campaign_type	TEXT	Category (e.g., 'Awareness', 'Regulation').
start_date	DATE	The date the campaign officially began.
end_date	DATE	The date the campaign ended (nullable).
responsible_agency	TEXT	The organization leading the campaign.
budget_million_eur	NUMERIC(10,2)	The allocated budget in millions of Euros.

1.2 Table_Y: `tb_campaign_city`

This bridge table creates a many-to-many relationship between campaigns (`tb_environmental_campaign`) and the existing OLTP table `tb_city`. This allows a single campaign to target multiple cities, and a city to participate in multiple campaigns.

1.3 Integration into OLAP Schema

Data from `tb_environmental_campaign` is loaded directly into the `dim_campaign` dimension. The bridge table, `tb_campaign_city`, is used during the ETL process for the fact tables to determine which campaign (if any) was active in a given city on a specific date.

2 Business/Analytic Questions

The following business questions were formulated to guide the design of our star schema.

2.1 Student A - Kerim Halilović

1. How did average pollution levels change in cities participating in campaigns compared to those that did not?
2. Which campaign types (Awareness vs. Regulation) achieved the highest effectiveness scores?
3. Is there a correlation between campaign budget and observed improvement in air quality?
4. During which months do cities most frequently launch environmental campaigns?

2.2 Student B - Nikola Lukić

1. Do cities with higher campaign activity require fewer maintenance services per device?
2. Which sensor manufacturers operate most frequently in cities with active campaigns?
3. How does service cost vary between cities with and without ongoing campaigns?
4. Are underqualified technician assignments more common during campaign periods?

3 Star Schema Diagram

The figure below illustrates the final star schema, consisting of two fact tables and eight dimension tables. The diagram shows the relationships and keys for all tables in the `dwh_006` schema.

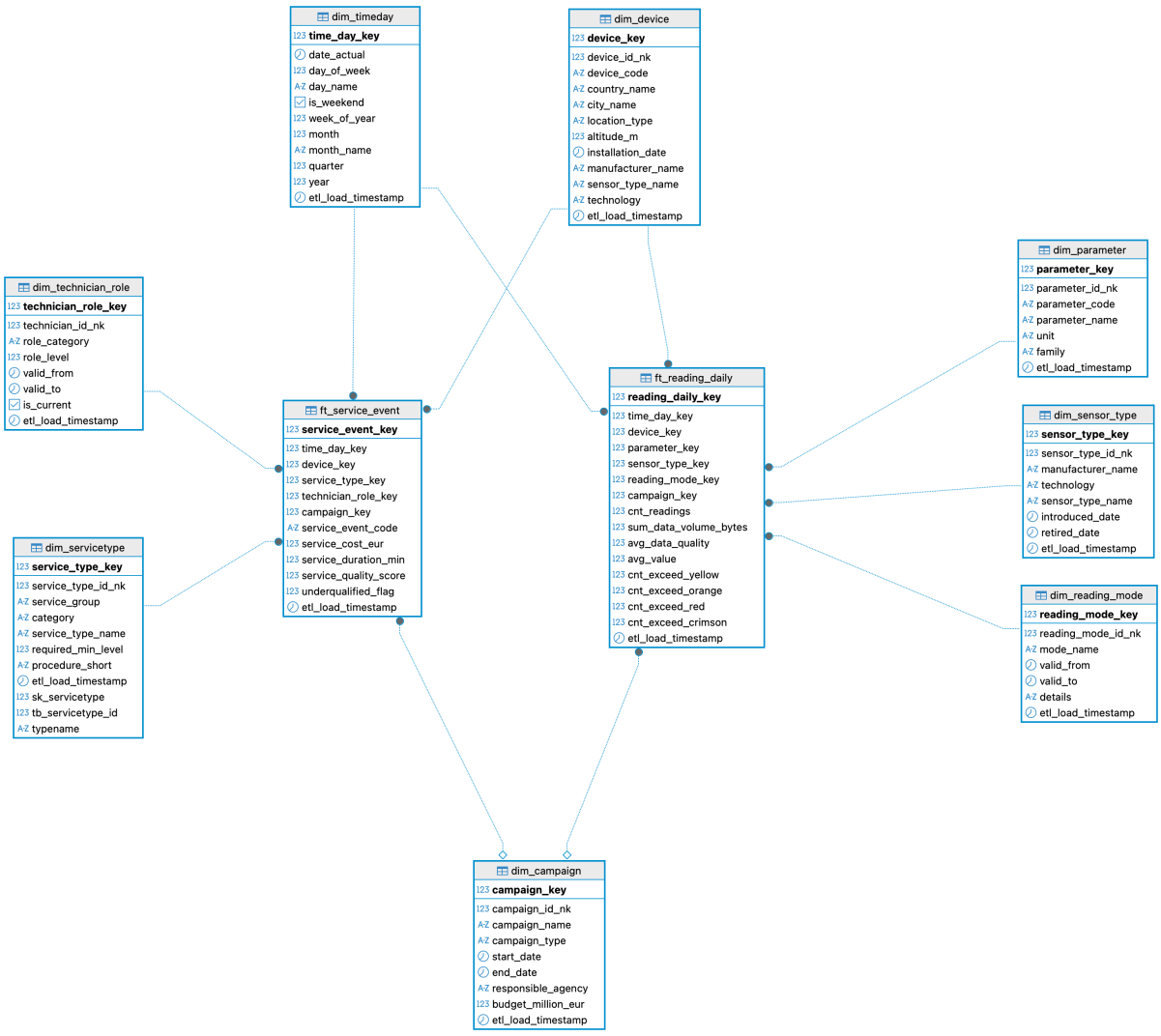

Figure 1: Star Schema for the Air Quality Data Warehouse.

4 Fact Tables

4.1 Fact 1: `ft_reading_daily`

- **Primary Responsibility:** Student A - Kerim Halilović
- **Business Motivation:** To analyze air quality trends and sensor telemetry load over time. This fact table enables comparisons of pollution metrics across different locations, device types, and during specific environmental campaigns.
- **Grain:** One row per sensor device, per measured parameter, per day.
- **Measures:**
  - `cnt_readings` (SUM)
  - `sum_data_volume_bytes` (SUM)
  - `avg_data_quality` (AVG)
  - `avg_value` (AVG)
  - `cnt_exceed_[level]` (SUM)
- **Linked Dimensions:** `dim_timeday`, `dim_device`, `dim_parameter`, `dim_sensor_type`, `dim_reading_mode`, `dim_campaign`.

4.2 Fact 2: `ft_service_event`

- **Primary Responsibility:** Student B - Nikola Lukić
- **Business Motivation:** To monitor operational efficiency, service costs, and technician compliance. This fact table helps track maintenance activities and analyze their cost and quality, particularly in relation to device types and campaign periods.
- **Grain:** One row per service event.
- **Measures:**
  - `service_cost_eur` (SUM)
  - `service_duration_min` (SUM)
  - `service_quality_score` (AVG)
  - `underqualified_flag` (SUM)
- **Linked Dimensions:** `dim_timeday`, `dim_device` (shared), `dim_service_type`, `dim_technician_role`, `dim_campaign`.

5 Dimension Tables

- **Hierarchies:** Three dimensions include multi-level hierarchies:
  - `dim_device`: Country $\rightarrow$ City $\rightarrow$ Device
  - `dim_sensor_type`: Manufacturer $\rightarrow$ Technology $\rightarrow$ Type Name
  - `dim_service_type`: Service Group $\rightarrow$ Category $\rightarrow$ Type Name
- **Time Dimension:** We implemented a single time dimension, `dim_timeday`, as its daily granularity is enough to answer all formulated business questions and supports analysis at higher levels (month, quarter, year) through its attributes.

- **SCD Type 2:** The `dim_technician_role` dimension is implemented as an SCD Type 2 to track the history of roles held by each technician. Each row represents a specific role assignment with `valid_from` and `valid_to` dates, enabling accurate compliance checks for service events at any point in time.
- **Degenerate Dimension:** The `service_event_code` from the source system is stored directly in `ft_service_event`. This makes it possible to trace each fact record back to the original source transaction without creating a separate dimension table.

6 Snowflake vs. Star Schema

The star schema was required for this assignment and is the ideal choice for this project. By denormalizing hierarchies (such as geography into `dim_device`), we created a schema with fewer joins, which simplifies analytical queries and improves performance. All business questions can be answered efficiently with this structure.

A snowflake schema would have normalized the hierarchies into separate tables (e.g., `dim_city`, `dim_country`). While this might slightly reduce data redundancy in the dimension tables, it would increase query complexity by requiring additional joins. Given the performance goals of a data warehouse, the star schema's simplicity and speed are more advantageous.

7 ETL and Validation Summary

The ETL process was implemented using SQL scripts coordinated through a Jupyter Notebook. After the complete pipeline execution, 7 validation queries were executed on the `dwh_006` schema to verify data accuracy and integrity.

All 7 selected validation checks passed successfully, which confirms the ETL process loaded the data as expected. Some of the key findings from this test suite include:

- Dimension row counts (including custom `dim_campaign`) exactly matched their corresponding source tables.
- All foreign keys in both fact tables successfully referenced existing keys in their respective dimension tables (0 mismatches).
- The SCD Type 2 dimension (`dim_technician_role`) showed no overlapping time ranges and had exactly one current record per technician.
- All measures were within their expected logical ranges (e.g., no negative costs or invalid quality scores).

Finally, a provenance file (`prov_airq_dwh_006.jsonld`) was created to keep a detailed record of the ETL process.

8 Reflection and Lessons Learned

8.1 Student A: Kerim Halilović

This assignment I got hands on experience with the full data warehousing process, from theory to actual implementation. The biggest challenge for me was creating the ETL process for the ft_reading_daily fact table. Turning raw reading events into clean daily summaries required careful grouping and calculations. The hardest part was adding the business logic for exceedance counts, which meant joining the readings with alert thresholds and correctly labeling each case. This showed

me that ETL is not just about moving data but also about transforming it to create new and useful information.

I also realized how important it is to design a clear and detailed schema before writing any code. Having the star schema ready made it much easier to build the ETL scripts. The validation step was also very important, running checks on data links and value ranges helped confirm that our transformations were correct. Overall, this project taught me that the quality and reliability of a data warehouse depend entirely on how accurate and consistent its data is.

8.2 Student B: Nikola Lukić

My main lesson from this project is how important it is to model time and history correctly. When designing the ft_service_event fact table, I realized that linking each service only to the current employee role wasn't enough. To answer our business question about underqualified service, we needed to know what role the technician had at the time of the service. This led us to create the dim_technician_role table as a Slowly Changing Dimension (SCD) Type 2. Writing the ETL for this and joining it correctly in the fact table using the event date between valid_from and valid_to was the hardest but also the most rewarding part.

The project also showed me how valuable teamwork and shared design are. Since our dim_device and dim_campaign tables were shared between multiple fact tables, we had to agree early on how they would look and how the data would be loaded. This collaboration made combining our different parts much easier later on. Finally, building validation scripts, like checking the underqualified_flag logic taught me that validation is not just about checking record counts; it's about making sure the data warehouse truly answers the business questions it was designed for.