188.429 Business Intelligence

Group 033 Project Report - Part 1

Dimensional modelling and ETL exercise

Group 033

Frycz Mikolaj 12518516
Ajanidisz Andras 12533685

November 4, 2025

Contents

1 Synthetic tables (Table_X and Table_Y)

The original did not contain any information about mentorship. We only had tables for work events (tb_serviceevent) and basic employee roles (tb_employee). To extend this, we created two new tables so we could track the agency's mentorship programs and see how they affect technician performance.

1.1 Table X: tb_mentorship_program

This is an entity table that defines the master list of available mentorship programs.

Description of Fields:

id (INT, Primary Key): Unique ID for the program.
program_name (VARCHAR(255)): The program's official name (e.g., "New Hire Hardware Onboarding").
program_focus (VARCHAR(255)): The main skill it targets (e.g., "Hardware," "Software").

1.2 Table Y: tb_mentorship_link

This is a bridge table that connects mentors to mentees for a specific program. It has 12 rows, representing 12 different mentorship pairings.

Description of Fields:

id (INT, Primary Key): Unique ID for the link itself.
program_id (INT, Foreign Key): Links to tb_mentorship_program to show which program it is.
mentor_employeeid (INT, Foreign Key): The mentor's employee ID.
mentee_employeeid (INT, Foreign Key): The mentee's employee ID.

1.3 OLAP Schema Integration

We used an ETL script (a1_etl_07_dim_technician_scd2.sql) to pull data from these two new tables and update our main dim_technician dimension. This added three new columns to each technician's record in dim_technician:

mentorship_role (e.g., 'Mentor', 'Mentee', 'Both', 'None')
mentorship_program_name
mentorship_program_focus

This was the main goal, as it lets us answer business questions (like Q1, Q2, and Q3). Now we can slice the ft_service fact table using these new attributes. For example, we can compare the average service quality score for technicians who are 'Mentees' against those who are 'None'.

2 Business/analytic questions

Mikolaj Frycz:

- **Q1:** For each of the last three calendar years, provide a ranked list of cities, based on the total number of days that each city recorded alert level "Red" for three or more distinct air quality parameters.

- **Q2:** How do the total monthly service costs compare against the average data quality scores, rolled up by country?
- **Q3:** What is the month-over-month trend of "Red" and "Crimson" alert occurrences specifically for the "Particulate Matter" parameter category?
- **Q4:** What percentage of all service events were performed by under-qualified technicians, and which service group shows the highest rate of these compliance breaches?

Andras Ajanidisz:

- **Q1:** What is the average service duration and cost for tasks performed by technicians who held a "Junior" role at the time the service was performed, in contrast with "Senior" technicians?
- **Q2:** Is there a correlation between a sensor's manufacturer and the total number of "Yellow" alerts recorded by those devices, broken down by city?
- **Q3:** How does the average service quality score differ between technicians who are currently "Mentors", those who are "Mentees" and those who are currently not participating in any of the mentorship programs when performing tasks belonging to the "Hardware Intervention" category?
- **Q4:** Which sensor technology generates the most total data volume but has the lowest average data quality score?

3 Star schema diagram

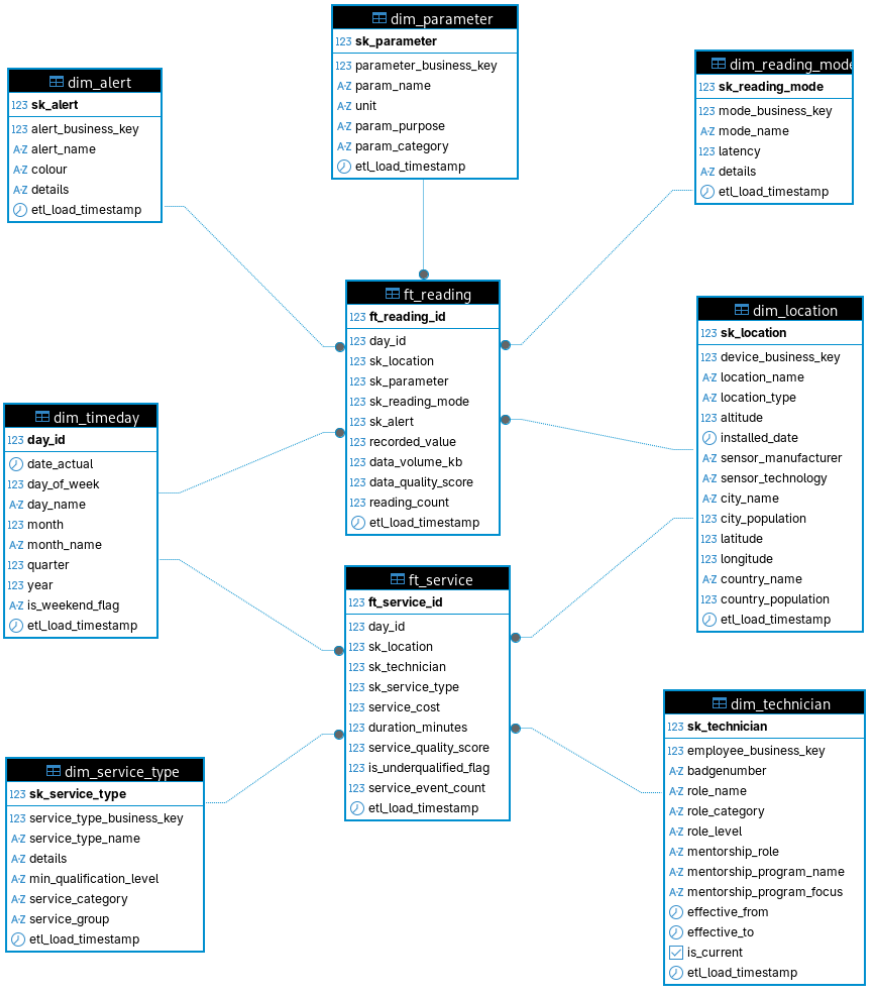

Figure 1: Star schema ER diagram

4 Fact tables

We built our data warehouse around two fact tables. This design keeps our processes separate but lets us link them using shared dimensions like dim_location and dim_timeday.

1. **ft_reading** (Mikolaj Frycz): Tracks all sensor and environmental data.
2. **ft_service** (Andras Ajanidisz): Tracks all operational and technician performance data.

4.1 Fact 1: ft_reading - Mikolaj Frycz

- **Business Motivation:** This table holds every sensor measurement. We use it to track air quality trends, monitor sensor data volume/quality, and find pollution hotspots. It's the key to answering business questions Q4, Q7, Q9, and Q11.
- **Grain:** The grain is one row per sensor reading event, as loaded from the tb_readingevent source table.
- **Measures:**
  - **recorded_value** (Rule: AVG, MIN, MAX): A semi-additive measure representing the scientific value recorded by the sensor.
  - **data_volume_kb** (Rule: SUM): A fully additive measure used to track the data load from each sensor.
  - **data_quality_score** (Rule: AVG): A semi-additive measure representing the quality (1-5) of the reading.
  - **reading_count** (Rule: SUM): A fully additive "helper" measure (always 1) used to count events.
- **Linked Dimensions:**
  - **dim_timeday** (Shared Dimension)
  - **dim_location** (Shared Dimension)
  - **dim_parameter**
  - **dim_reading_mode**
  - **dim_alert**

4.2 Fact 2: ft_service - Andras Ajanidisz

- **Business Motivation:** This is our main table for operations and HR. It logs every maintenance event so we can track costs, check technician compliance, and see if our new mentorship programs are effective. It answers questions Q1, Q3, Q6, Q8, and Q10.
- **Grain:** The grain is one row per technician service event, as loaded from the tb_serviceevent source table.
- **Measures:**
  - **service_cost** (Rule: SUM): A fully additive measure for the cost of the service.
  - **duration_minutes** (Rule: SUM): A fully additive measure for the time spent on the service.

  - **service_quality_score** (Rule: AVG): A semi-additive measure representing the quality (1-5) of the service.
  - **is_underqualified_flag** (Rule: SUM): A fully additive, calculated measure (1 or 0) that flags services performed by a technician whose role level was below the minimum requirement.
  - **service_event_count** (Rule: SUM): A fully additive "helper" measure (always 1) used to count events.

- **Linked Dimensions:**
  - **dim_timeday** (Shared Dimension)
  - **dim_location** (Shared Dimension)
  - **dim_technician**
  - **dim_service_type**

5 Dimension tables

- **dim_parameter:** The Parameters Dimension is a direct mapping of the Parameters table in the OLTP schema. There are no hierarchy levels.
- **dim_alert:** The Alert Dimension is a direct mapping of the Alert table in the OLTP schema. There are no hierarchy levels.
- **dim_reading_mode:** The Reading Mode Dimension is a direct mapping of the Reading Mode table in the OLTP schema. There are no hierarchy levels.
- **dim_location:** The Location Dimension is a conceptual table that comined data from four categories, arranged in the following hierarchy levels: country -> city -> sensor type -> sensor device.
- **dim_timeday:** The Time Dimension defines the following hierarchy levels: year, quarter, month, is_weekend, day of the week, date. Because both the Reading Event and Service Event facts share the same daily granularity, a single dimension table is used to provide consistent time context for both.
- **dim_technician:** The Technician Dimension is built on a hierarchy of role category, role level, mentorship program, mentorship role, role name, badgenumber, effective dates.

  This is a Slowly Changing Dimension of type 2. Because technicians change their roles or get promoted, multiple rows can refer to the same technician via a shared business key. The dimension table preserves role history, therefore, it is a Slowly Changing Dimension of Type 2.
- **dim_service_type:** The Service Type Dimension is a direct mapping of the Service Type table in the OLTP schema. There are no hierarchy levels.

Our OLAP schema does not implement any junk or degenerate dimensions.

6 Snowflake vs. Star

Overall, implementing a snowflake architecture would have reduced the effectiveness of our OLAP schema to answer the business questions proposed above. We would not have preferred a Snowflake schema for this project, even if it were allowed.

A denormalized structure supports search for business questions that require us to aggregate data by city. If the Location Dimension were split into a city, a country and a device table, the above queries would have required multiple, complex joins. Similarly, aggregation of data along higher granuality level attributes is more efficient using the star schema. Furthermore, the primary benefit of a Snowflake is to save storage by reducing data redundacy. In our case, query speed and ease of use, the streghts of teh Star schema, are more important factors then data deduplicaiton.

7 ETL and validation summary

The ETL pipeline was implemented as a series of SQL scripts executed in lexicographical order. These scripts first populate the seven dimension tables, including handling the complex SCD2 logic for dim_technician, and subsequently populate the two fact tables. During the fact table load, all necessary business logic, such as alert level calculation and compliance checks, is performed.

Following the ETL run, a series of seven post-ETL validation checks were executed to verify the integrity, completeness, and correctness of the data warehouse.

7.1 Summary of the checks performed and their results:

1. **Referential Integrity Check:**
    - **Purpose:** To ensure no "orphan rows" exist. This check verified all 9 foreign keys across both ft_reading and ft_service.
    - **Result:** All 9 checks returned a count of **0 orphan rows**.
    - **Interpretation:** This confirms that 100% of our fact rows are successfully linked to their parent dimension tables, ensuring no data will be lost during analytical queries.
2. **Fact Row Count Check:**
    - **Purpose:** To compare the total row counts from the source event tables tb_readingevent, tb_serviceevent) against the final fact tables.
    - **Result:** The row counts for ft_reading and ft_service were **identical** to their respective source tables.
    - **Interpretation:** This validates that our ETL process loaded every source event exactly once, with no records being accidentally dropped or duplicated.
3. **SCD Type 2 Consistency Check:**
    - **Purpose:** To test the complex dim_technician SCD2 table for overlapping date ranges or multiple is_current=TRUE flags for the same technician.
    - **Result:** The query returned **0 rows**, indicating no such errors.
    - **Interpretation:** Our SCD2 logic is correct. This ensures that analysis of technician history is consistent and will not lead to double-counting of service events.
4. **Mentorship (Table X/Y) Integration Check:**
    - **Purpose:** To confirm that data from our synthetic tb_mentorship_program and tb_mentorship_link tables was successfully integrated into dim_technician.

- **Result:** The query showed **non-zero counts** for technicians with 'Mentor' and 'Mentee' roles.
- **Interpretation:** This validates that our custom tables were successfully processed, enriching dim_technician and enabling analysis of mentorship impact as intended.

5. **Measure Range Check:**

   - **Purpose:** To scan for "impossible" values in key measures, such as quality scores outside the 1-5 range or negative service costs.
   - **Result:** All checks returned **0** `bad_rows`.
   - **Interpretation:** This proves our data is clean and all measures fall within their expected business domains, ensuring the integrity of future analytical calculations.

6. **Business Logic Calculation Check:**

   - **Purpose:** To validate the custom logic created during the ETL, specifically the is_underqualified_flag and the sk_alert calculation.
   - **Result:** The query showed a **non-zero count** for 'Yes - Underqualified' events (as expected from the business case) and a **large count** for 'No Alert' events.
   - **Interpretation:** This confirms our CASE and COALESCE logic worked, and the DWH is correctly identifying both compliance issues and non-alerted readings.

7. **dim_location Denormalization Check:**

   - **Purpose:** To ensure the complex 4-table join used to build the dim_location conformed dimension resulted in exactly one row per source device.
   - **Result:** The row count for dim_location was **identical** to the source tb_sensordevice row count.
   - **Interpretation:** The denormalization was successful, with no devices lost or duplicated.

Finally, a prov_airq_dwh_033.jsonld file was generated to record the high-level provenance of the ETL run, linking the student agents, the OLTP source tables, and the generated DWH tables.

8 Reflection and lessons learned

After reflecting on and discussing our experiences, we agreed to conclude them in a single section, combining perspectives from both sides.

We think that the project has been a nice introduction to the core ideas of the Business Intelligence as a discipline. The real-world applicability of the subject was immediately clear. We can see how this knowdledge may integrate into and transform our careers, especially when considering roles such as data engineer or BI analyst. We gained useful insights into why conceptual distillation is necessary to enhance the business value of data. The project has also been a catalyst for further reflection on how technology development always begins at the abstract, conceptual layer before moving to tangible reality. Lastly, we have discovered new ways of incorporating Articifial Intelligence in par projects, which has amplified our abilities to reflect on our thought processes and elevated the workflow to a more conceptual level.