ETL, Cube Modeling & Analytics (SQL/MDX/Atoti)

Assignment 2 continues the AirQ workflow you started i**n Assignment 1.** The star schema for this part is given, and your job is to:

1. implement a clean **ETL** from the **OLTP snapshot** into this star,
2. build an **OLAP cube in Atoti** with **explicit hierarchies and measures**,
3. answer **business questions** using **SQL**, **MDX**, and the **Atoti web app** (via session.url).

Table of Contents

Preface 2
1. Warehouse schema 4
2. Measures 5
3. ETL from OLTP to OLAP 6
4. Build the cube in Atoti 6
5. Answer business questions (SQL, MDX, dashboard) 6
5.1 How to Execute & Save (SQL, MDX, Atoti) 6
5.2. Selecting and dividing questions 7
5.3. File naming (please follow carefully) 7
6. Business-question pool 7
7. Roles and deliverables 10
8. Report (short, 1–2 pages) 10
8.1. ETL summary 10
8.2. Answers to business questions 10
8.3. Reflection and lessons learned 10
9. What to Submit 11

Preface

Deadline: Upload all your results **by November 30, 2025.**

Submission guidelines: Please **pay attention to naming conventions and follow them carefully**, as this will ensure your code executes correctly in the test environment.

Compress your solution directory and upload it to TUWEL using the following filename:

`BI_2025_Assignment_2_Group_xxx.zip/.tgz`

Within this ZIP-file, the top-level directory is named BI_Projects. Inside this directory, there is a folder named DWH2_**xxx**, **please replace <xxx> with your actual three-digit group number**, (with the leading zeros if necessary).

This DWH2_xxx folder should contain all your deliverables as follows:

```
BI_Projects
  \-DWH2_xxx
    |
    +---csv
    |       15 original CSV files
    +---ddl
    |       airq_create_dwh2_xxx.sql, airq_create_stg2_xxx.sql,
    |       airq_reset_dwh2_xxx.sql, airq_reset_stg2_xxx.sql
    +---etl
    |       a2_etl01_dim_timemonth.sql
    |       a2_etl02_dim_city.sql
    |       a2_etl03_dim_param.sql
    |       a2_etl04_dim_alertpeak.sql
    |       a2_etl05_ft_param_city_month.sql
    +---mdx
    |       a2_q{NN}_{A|B}.mdx
    +---mdx_out
    |       a2_q{NN}_{A|B}.csv
    +---pdf
    |       a2_q{NN}.pdf
    +---sql
    |       a2_q{NN}_{A|B}.sql
    +---sqldump
    |       sqldump_airq_dwh2_xxx.sql
    |   AirQ_Part2_xxx.ipynb
    |   group_xxx.txt
    |   Report_Part2_Group_xxx.pdf


* Please replace xxx with Your Group Number
```

Assignment 2

In **Assignment 1**, you designed a star schema and implemented a SQL-first ETL, with a Jupyter notebook acting as an *orchestrator*. You also validated your loads and suggested business questions against your own mart. In **Assignment 2**, we continue the same workflow, but now the **star schema is given** and your focus shifts to (a) implementing a clean ETL from the **OLTP snapshot** into this provided star, and (b) building an **OLAP cube in Atoti**, defining **explicit hierarchies and measures**, and answering business questions via **SQL, MDX, and the Atoti web app**. The notebook remains minimal - primarily orchestration - so most of your work is still in SQL and MDX. Style, packaging and tone should follow A1.
A2 does **not** require a new setup guide. Use the same environment as A1. If you want a clean start, you may re-run the student bootstrap; otherwise, continue with your existing airq database.

Questions. Please post general questions and discuss issues in the TUWEL discussion forum. We appreciate it if you help other students (and may take this into account if you are short a few points for a better grade). For obvious reasons, however, please do not post any solutions in this forum.

For specific questions regarding the assignments, you can contact our tutors

- Ildar Fatkullin (ildar.fatkullin@tuwien.ac.at)
- Bosse Behrens (bosse.behrens@tuwien.ac.at)
- Alwin Krycha-Weilinger (alwin.krycha-weilinger@tuwien.ac.at)

Or in case of other issues

- Katja Hose (katja.hose@tuwien.ac.at).

1. Warehouse schema

Warehouse schema dwh2_xxx is given (as DDL script). **Do not alter columns or keys.**

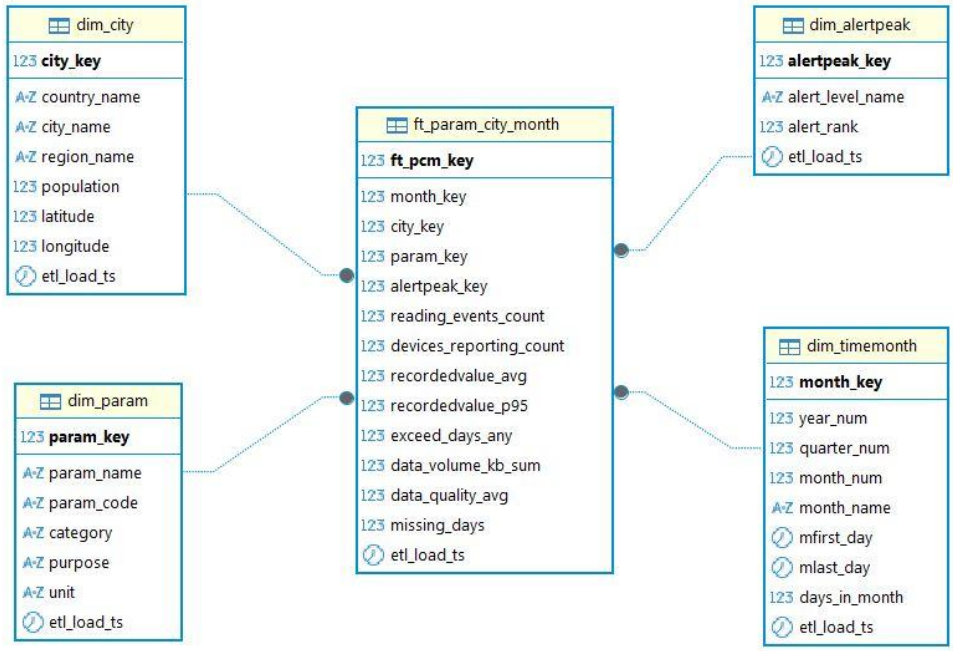

Figure 1. Star Schema

Dimensions (4):

- dim_timemonth (Month time): month_key (e.g., 202401), year_num, quarter_num, month_num, month_name, mfirst_day, mlast_day, days_in_month.
- dim_city (Geo): country_name, city_name, region_name and other attributes.
- dim_param (Parameters): param_name, category, purpose, unit.
- dim_alertpeak (Alert ladder): fixed keys 1000..1004 for None, Yellow, Orange, Red, Crimson with alert_rank for ordering.

Fact (1):

- ft_param_city_month at grain **Month × City × Param**, with FK to the four dims and the measures listed below. A uniqueness constraint enforces (month_key, city_key, param_key).

Staging schema stg2_xxx holds the read-only OLTP snapshot (readings, devices, parameters, alerts and thresholds, countries and cities, etc.). You should load from this snapshot into dwh2_xxx.

2. Measures

All measures live in ft_param_city_month at the **Month × City × Param** grain. Use the definitions below when writing your ETL.

- **reading_events_count** — Number of distinct (device, day) pairs with at least one reading for this month, city, and param. **Aggregate**: SUM.
- **devices_reporting_count** — Number of distinct devices that produced at least one reading for this month, city, and param. **Aggregate**: SUM.
- **data_volume_kb_sum** — Sum of telemetry volume datavolumekb over all readings in the month. **Aggregate**: SUM.
- **exceed_days_any** — Count of distinct days in the month where the **daily peak alert** for the city & param reached **at least Yellow**. Compute as follows:
    1. For each **day**, derive a **daily rank** 0..4 by comparing readings to parameter-specific thresholds; take **only the highest** level exceeded that day (Yellow=1, Orange=2, Red=3, Crimson=4; None=0).
    2. Count days with daily_rank ≥ 1 to produce exceed_days_any. **Aggregate**: SUM.
- **missing_days** — days_in_month – days_with_readings, where days_with_readings is the count of distinct reading dates in that month for the city & param. **Aggregate**: SUM.
- **recordedvalue_avg** — Arithmetic mean of recordedvalue across all readings in the month. **Aggregate**: MEAN (semi-additive).
- **recordedvalue_p95** — The 95th percentile of recordedvalue across the month. **Aggregate**: MEAN (semi-additive).
- **data_quality_avg** — Mean of the per-reading quality score (1..5). **Aggregate**: MEAN (semi-additive).

In addition, ft_param_city_month contains the column alertpeak_key, which stores the monthly peak alert encoded as 1000..1004. Compute it via daily ranks:
monthly_peak_rank = MAX(daily_rank), then map 0→1000, 1→1001, ⋯, 4→1004
(do the math in 0..4 and map to 1000..1004 only at the end).
This column must be computed in your ETL and stored in ft_param_city_month because it is used to link to dim_alertpeak, it is not used as a separate aggregate measure.

All of the above columns are present in the provided DDL and should be populated by your ETL into dwh2_xxx.ft_param_city_month. The alert ladder keys and constraint 1000..1004 are part of the given design.

Note. The snapshot provides tb_paramalert and tb_alert for thresholds and level names; you should use them to derive daily ranks and monthly peaks.

3. ETL from OLTP to OLAP

- Run the given DDL to create stg2_xxx and dwh2_xxx . Do **not** modify the warehouse schema.
- Use the same 15 CSV files as in A1.
- Implement the ETL from stg2_xxx to dwh2_xxx. Place one SQL file per target table in etl/ and execute them in order from the notebook (dimensions first, then the fact). Keep the notebook as an orchestrator.
- You are encouraged to run post-ETL checks - any sanity checks you need, but you do not have to include them in the notebook or submission for A2. (A1 already covered documented validation.)

4. Build the cube in Atoti

The notebook creates a session and loads five stores from dwh2_xxx. Join the fact to dimensions on keys, create a cube (manual mode), and define hierarchies and measures explicitly.

- **Hierarchies**
  - **Time:** year_num → quarter_num → month_name (order months Jan→Dec).
  - **Geo:** region_name → country_name → city_name.
  - **Param:** purpose → category → param_name.
  - **Alert:** alert_level_name (order None→Yellow→Orange→Red→Crimson).
- **Measures**
  - SUM: reading_events_count, devices_reporting_count, data_volume_kb_sum, exceed_days_any, missing_days.
  - MEAN: recordedvalue_avg, recordedvalue_p95, data_quality_avg.

The cells containing hierarchies and measures to define are marked with **TODO notes**. The scaffolding (session, stores, joins, session.url) is provided.

5. Answer business questions (SQL, MDX, dashboard)

5.1 How to Execute & Save (SQL, MDX, Atoti)

Execute each chosen question in the correct tool and save queries into the files.

SQL

- Run in PostgreSQL (in DBeaver) against the given star schema.
- Save each query as a file in sql/ using the naming from §5.3 (e.g., a2_q03_A.sql).

MDX

- Run in the Jupyter notebook (using %%mdx magic cell).
- Save the MDX script in mdx/ (e.g., a2_q17_B.mdx).

Atoti PDFs

- Solve **4–6** questions interactively and export each dashboard to pdf/ (e.g., a2_q21.pdf).

5.2. Selecting and dividing questions

From the pool of 30 business questions, implement **10 in SQL** and **10 in MDX** per team.

- **Student A:** 5 SQL + 5 MDX
- **Student B:** 5 SQL + 5 MDX

Within **SQL**, all 10 must be **distinct** (no duplicates between A and B). Within **MDX**, all 10 must be **distinct** (no duplicates between A and B). It is allowed to implement the same question number once in SQL and once in MDX (e.g., Q14 in both SQL and MDX).

Additionally, solve 4–6 questions interactively in the Atoti web app and export as PDF.

5.3. File naming (please follow carefully)

Pattern: a2_q{NN}_{A|B}.{ext}

- {NN} = question number, **zero-padded to two digits** (01…30).
- {A|B} = who implemented it (**Student A** or **Student B**).
- {ext} = sql, mdx, csv, or pdf (Atoti export).

Where to place files

- sql/ → a2_q03_A.sql
- mdx/ → a2_q17_B.mdx
- mdx_out/ → a2_q17_B.csv (MDX result)
- pdf/ → a2_q21.pdf (Atoti export)

Examples

- Student A did Q03 in SQL → sql/a2_q03_A.sql
- Student B did Q17 in MDX → mdx/a2_q17_B.mdx
- Atoti export for Q21 → pdf/a2_q21.pdf

6. Business-question pool

1. For parameter PM2, show Exceed Days (any) by Country × Month for Q1 of 2024. Return Countries on rows and the first three months of 2024 (Jan–Mar) on columns.

2. For parameter O3, show Missing Days in Austria by City × Month for Q1 of 2023. Return Austrian Cities on rows and the first three months of 2023 (Jan–Mar) on columns.

3. For PM10 in 2024, show the total Exceed Days (any) by City. Return one row per city and a single column with the total number of exceedance days for that year.

4. For 2024, show total Data Volume (KB) by Region × Quarter. Return Regions on rows and the four quarters of 2024 on columns.

5. For 2023 and 2024, show total Data Volume (KB) by Param Category × Year. Return Param Categories on rows and the two years (2023, 2024) on columns.

6. For 2024, list the Top 10 Cities by total Missing Days (all parameters). Return the 10 cities with the highest totals on rows (highest → lowest) and one column with the 2024 total Missing Days.

7. For parameter PM10, show Avg Recorded Value and P95 Recorded Value by Country for 2023. Return Countries on rows and two columns—Avg Recorded Value and P95 Recorded Value—for the year 2023.

8. For 2024, show Avg Data Quality by Country, but only for countries whose 2024 year-total Devices Reporting is at least 2000. Return Countries on rows (filtered to Devices Reporting ≥ 2000) and one column with Avg Data Quality for the year 2024.

9. For 2024, show Reading Events by Country × Quarter (Top 10 countries). Return the four quarters on columns (Q1–Q4) and the Top 10 countries on rows, ranked by total Reading Events in 2024.

10. For 2024, list the Top 10 Countries by Avg Data Quality. Return the 10 countries with the highest values on rows (highest → lowest) and one column with Avg Data Quality for 2024.

11. For 2024, show Exceed Days (any) by Region for Param Category = ‘Gas’. Return Regions on rows and one column with Exceed Days (any) for the year 2024, filtered to Category = Gas.

12. For 2024, show Exceed Days (any) by City × monthly peak Alert Level (None, Yellow, Orange, Red, Crimson) for Eastern Europe. Return Cities in Eastern Europe on rows and the five Alert Levels on columns.

13. For Q1 of 2023, show Exceed Days (any) by City × Month where the monthly peak Alert Level is Yellow or None. Return Cities on rows and the first three months of 2023 (Jan–Mar) on columns, limited to months labelled Yellow or None.

14. For 2024, list the Top 10 City × Param pairs by Avg Data Quality. Return the 10 City–Param pairs with the highest values on rows (highest -> lowest) and one column with Avg Data Quality for 2024.

15. Show Exceed Days (any) by Country in Eastern Europe for 2023 and 2024. Return Countries (only those in Eastern Europe) on rows and two columns—2023 and 2024 totals of Exceed Days (any).

16. For 2024, show Data Volume (KB) by Param Category × Quarter. Return Param Categories on rows and the four quarters of 2024 (Q1–Q4) on columns.

17. Show Avg Data Quality by Country for 2023 and 2024. Return Countries on rows and two columns—2023 and 2024 values of Avg Data Quality.

18. For 2023, show Reading Events by Quarter for Vienna, Berlin, Moscow, and London (all parameters). Return the four cities on rows and the four quarters of 2023 (Q1–Q4) on columns.

19. For 2024, show Missing Days and Data Volume (KB) totals by City in Central Europe. Return Cities in Central Europe on rows and two columns - Missing Days and Data Volume (KB) - for the year 2024.

20. For 2024, show Avg Recorded Value by Country for all parameters with Purpose = Scientific Study, limited to Western Europe. Return Countries in Western Europe on rows and each Scientific Study parameter on columns (values = Avg Recorded Value).

21. Show Data Volume (KB) by Param Category for 2023 and 2024, limited to Purpose = Health Risk or Environmental Monitoring. Return Param Categories (only those under the two purposes) on rows and two columns—2023 and 2024 totals of Data Volume (KB).

22. For 2024, show Missing Days by Param Category × Quarter. Return Param Categories on rows and the four quarters of 2024 (Q1–Q4) on columns.

23. For 2024, for each Country, return the Month with the highest Data Volume (KB). Return one row per Country–Month (the best month per country) and one column with Data Volume (KB).

24. For parameter CO2, show P95 Recorded Value by Country for 2024. Return Countries on rows and one column with P95 Recorded Value for the year 2024.

25. For 2023 and 2024, show Exceed Days (any) by Purpose, plus the change from 2023 to 2024. Return Purposes on rows and three columns - Exceed Days 2023, Exceed Days 2024, and Change 2024–2023.

26. For parameters PM1 and NO2, show Avg Recorded Value by Country for 2024. Return Countries on rows and two columns - PM1 and NO2 (Avg Recorded Value) - for the year 2024.

27. For 2024, show Exceed Days (any) by Country × Quarter for Russia, Turkey, Austria, and Germany. Return the four countries on rows and the four quarters of 2024 (Q1–Q4) on columns.

28. For 2024, list the Top 10 Countries by Missing Days. Return the 10 countries with the highest totals on rows (highest → lowest) and one column with Missing Days for 2024.

29. Show Data Volume (KB) by Country in Eastern Europe for 2023 and 2024. Return Eastern European countries on rows and two columns—2023 and 2024 totals of Data Volume (KB).

30. For 2023–2024 (quarters), for Russia and Germany, return each country's quarter with the highest Data Volume (KB), and show both Data Volume (KB) and Avg Recorded Value for that quarter. Return one row per Country × Quarter (the top quarter per country across 2023–2024) and two columns: Data Volume (KB) and Avg Recorded Value.

Additional (solved) examples

31. For parameter O3, list the Top 10 Cities by P95 Recorded Value for 2023. Return the 10 cities with the highest values on rows (highest → lowest) and one column with P95 Recorded Value for 2023.

32. For 2024, show Data Volume (KB) by City for category 'Volatile Organic Compound', and list the Top 10 cities. Return the Top 10 cities on rows (highest → lowest) and one column with Data Volume (KB) for 2024, limited to the Volatile Organic Compound category.

33. For parameter PM4 in 2024, return for each Country the Month with the highest Avg Data Quality. Return one row per Country × Month (the month with the highest Avg Data Quality in 2024) and one column with Avg Data Quality.

All 30 given business questions are compatible with the given grain and hierarchies. They are phrased so you can implement them either in pure SQL over dwh2_xxx or in MDX over the cube.

7. Roles and deliverables

Within the team of two, keep the Student A / Student B split from A1.

- **ETL (joint effort):** Both students co-author and review a single ETL pipeline and submit one set of ETL scripts. There is no A/B split for ETL; both are equally responsible for a working load into dwh2_xxx.
- **SQL**: 10 total (5 by A; 5 by B), all **distinct** within SQL.
- **MDX**: 10 total (5 by A; 5 by B), all **distinct** within MDX.
- **Dashboard**: 4–6 joint effort interactive answers exported as **PDF** from Atoti.

8. Report (short, 1–2 pages)

A2 has a **much smaller report** than A1. Please aim for **1–2 pages** (plus a cover with names).

8.1. ETL summary

How you populated each dimension; how you computed each fact measure. Note any notable decisions, pitfalls, or difficulties.

8.2. Answers to business questions

List your selected 10 SQL and 10 MDX question numbers. Then provide 4-6 short bullets that reflect on your query implementations (not on the output). Each bullet should reference specific question numbers and briefly discuss the code. For example, which query was easier in SQL vs. MDX, how you expressed the business intent in each language (joins, groupings, hierarchies, filters), or any interesting difficulties, optimisations, or design choices in your queries.

Example:

SQL – Student A: Q03, Q07, Q12, Q16, Q25

SQL – Student B: Q04, Q10, Q15, Q18, Q29

MDX – Student A: Q05, Q09, Q17, Q22, Q24

MDX – Student B: Q06, Q11, Q14, Q20, Q30

8.3. Reflection and lessons learned

A short section where each student contributes 1-2 paragraphs on what they personally learned from Assignment 2 beyond the individual queries. Possible themes: teamwork and division of labour, ETL practices and debugging, building and using the OLAP cube in Atoti, working across tools (DBeaver, Jupyter, Atoti), or how the overall workflow changed your understanding of data warehousing and analytics. This section is not graded heavily, but we do expect an honest and thoughtful reflection on your experience.

9. What to Submit

Submit a single archive containing the following:

- **CSV snapshot:** csv/ (all original CSVs).
- **DDL:** ddl/ (staging & warehouse DDL).
- **ETL:** etl/ (one SQL per target table).
- **SQL answers:** sql/ (10 files total; distinct within SQL; use naming from §5.3).
- **MDX answers:** mdx/ (10 files total; distinct within MDX; use naming from §5.3).
- **MDX outputs:** mdx_out/ (CSV results for your MDX queries).
- **Dashboard exports:** pdf/ (4–6 Atoti exports as PDF).
- **SQL dump:** sqldump/ (schema + data via pg_dump).
- **Notebook:** AirQ_Part2_xxx.ipynb (must run start-to-finish with no manual steps).
- **Names:** group_xxx.txt (team members' names, optional single-line note if necessary)
- **Report:** Report_Part2_Group_xxx.pdf (1–2 pages + cover).

Use ASCII only (underscores, no spaces/Umlauts/special chars). Replace **xxx** with your three-digit group number. Follow the file-naming rules in §5.3 exactly.

```
BI_Projects/
  DWH2_xxx/
    csv/       # original CSV files
    ddl/       # DDL only (staging, warehouse)
    etl/       # ETL steps: a2_etl01_...sql, a2_etl02_...sql, …
    mdx/       # MDX-queries in .mdx files: a2_q{NN}_{A|B}.mdx
    mdx_out/   # CSV files with the results of MDX-queries
    pdf/       # PDF files with dashboard exports: a2_q{NN}.pdf
    sql/       # SQL-queries in .sql files: a2_q{NN}_{A|B}.sql
    sqldump/   # Export produced by pg_dump
    AirQ_Part2_xxx.ipynb
    group_xxx.txt
    Report_Part2_Group_xxx.pdf
```

Place all deliverables in a single folder, compress it into a .zip:

BI_2025_Assignment_2_Group_**xxx**.zip/.tgz

and upload it to TUWEL.