Faculty Of Sciences Of Sfax

Rapport de Projet

Trendy Jobs & Tools Data Warehouse

Modern Azure Data Engineering Architecture

Auteur : Ayedi Yassin and Ben Slimene Nour El Houda
Encadrant : Bouaziz Senda
Technologies : Azure Data Lake, Azure Synapse, Azure Functions, PySpark, SQL
Année universitaire : 2024 – 2025

Contents

Chapter 1

Introduction

The technology job market evolves extremely fast. New tools, frameworks, and roles emerge every year, making it difficult for companies, students, and professionals to understand which skills are truly relevant.

This project proposes a **cloud-native data warehouse on Microsoft Azure** capable of:

- Capturing **real-time popularity** from developer communities
- Analyzing **historical job market demand**
- Comparing popularity versus real demand

The final goal is to provide a reliable analytical foundation to study trends in jobs and tools.

Chapter 2

Problematic and Challenges

The main challenges addressed in this project are:

- Popularity signals are noisy and volatile
- Historical datasets are static and delayed
- Data sources are heterogeneous (JSON, CSV, APIs)
- Large volumes of data must be processed incrementally
- Infrastructure cost must be minimized
- Data duplication must be avoided

A layered Data Lake architecture combined with incremental ETL pipelines was chosen to address these issues.

Chapter 3

Global Architecture

3.1 Architecture Overview

The architecture follows modern cloud data engineering best practices.

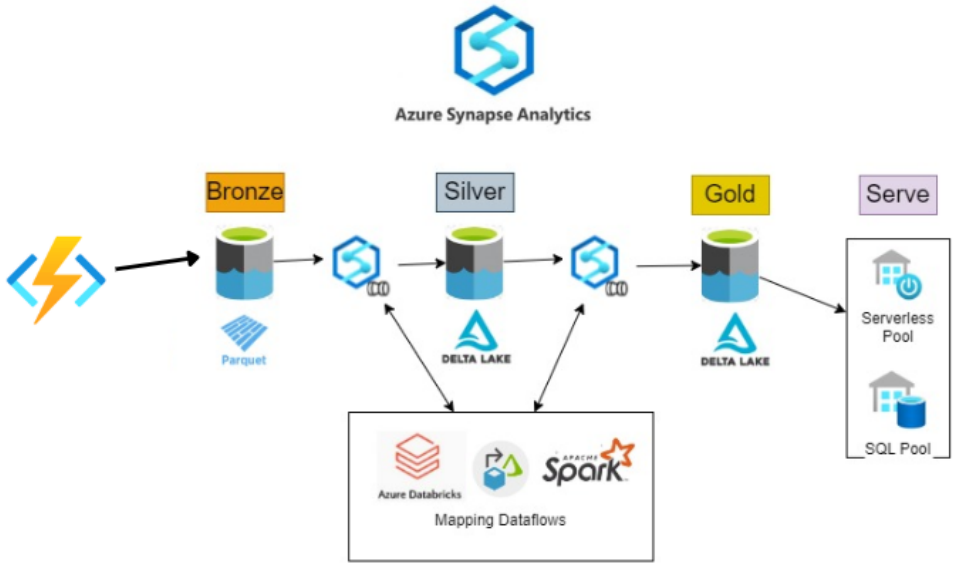


Figure 3.1: Global Azure Architecture (Extraction → Bronze → Silver → Gold → Serve)

Main components:

- Azure Functions for extraction
- Azure Data Lake Storage Gen2
- Azure Synapse Analytics (Spark + SQL)

Chapter 4

Azure Resource Creation

4.1 Azure Data Lake Storage Gen2

The storage account is created with:

- Hierarchical Namespace enabled
- Containers:
  - **bronze**: raw data
  - **silver**: cleaned and enriched data
  - **gold**: analytics-ready data

4.2 Azure Synapse Analytics

Steps:

1. Create Synapse Workspace
2. Link to Data Lake
3. Create Spark Pool (low-cost configuration)
4. Enable Serverless SQL Pool

Chapter 5

Bronze Layer

5.1 Why Azure Functions

Azure Functions were chosen because:

- Serverless execution
- Easy scheduling via TimerTrigger
- Native Azure integration

5.2 Project Structure

```
extract/
          function_app.py
          github_fetch.py
          reddit_fetch.py
          stackoverflow_fetch.py
          jobs_fetch.py
          utils.py
          host.json
          requirements.txt
          local.settings.json
          .funcignore
```

5.3 Function Orchestration

The main Azure Function is defined as:

```
@app.timer_trigger(schedule="0 */7 * * * *", arg_name="myTimer")
def ingestion_pipeline(myTimer):
    stages = [
        ("StackOverflow", fetch_stackoverflow),
        ("GitHub", fetch_github),
        ("Reddit", fetch_reddit),
    ]
    for stage_name, func_call in stages:
        func_call()
```

This design:

- Ensures sequential execution
- Isolates failures per source
- Allows easy extension

5.4 Local Execution

Azure Functions are executed locally using:

```
func start
```

5.5 Environment Variables

Required variables:

- `AZURE_STORAGE_CONNECTION_STRING`
- `GITHUB_PERSONAL_ACCESS_TOKEN`
- `REDDIT_CLIENT_ID`
- `REDDIT_CLIENT_SECRET`
- `REDDIT_USER_AGENT`

Each key is obtained from its respective platform (Azure Portal, GitHub Developer Settings, Reddit App Preferences).

5.6 Uploading data

Each extraction function saves raw JSON files.

```
file_name = f"github_repos_{timestamp}.json"
upload_to_datalake(json_data, file_name, filesystem_name="bronze/
   github")
```

Key principles:

- No transformation
- Timestamped files
- Append-only

Chapter 6

Silver Layer

The Silver layer represents the first transformation stage of the data warehouse. Its main objective is to convert raw data from the Bronze layer into clean, standardized, and reusable datasets while preserving traceability and enabling incremental processing.

6.1 Incremental Ingestion Control

To avoid reprocessing already ingested data, an incremental ingestion mechanism is implemented. A control file is maintained in the Silver layer:

```
_control/processed_files.txt
```

This file stores the timestamp of the last successful ETL execution. During each run, only data files with an ingestion timestamp greater than the stored value are processed. This approach:

- Prevents duplicate records
- Reduces processing time
- Ensures idempotent ETL executions

6.2 Ingestion Time Extraction Using Regular Expressions

Each raw file stored in the Bronze layer follows a strict naming convention that includes the ingestion timestamp, for example:

```
github_repos_2025-12-06_12-29-23.json
```

To retrieve this timestamp programmatically, **regular expressions (regex)** are used. A regex is a pattern-matching mechanism that allows extracting structured information from strings.

In this project, a regex pattern is applied to extract the date and time portion of the filename. The extracted value is then converted into a timestamp and stored as the `ingestion_time` column.

6.3 Date Standardization

Dates originating from different sources are represented in heterogeneous formats (Unix timestamps, ISO strings). To unify time-based analysis, all dates are converted into a common structure composed of:

- `year`
- `month`
- `day`

This standardization enables efficient partitioning, aggregation, and time-series analysis in downstream layers.

6.4 Job Title Normalization

Job titles extracted from different sources often exhibit significant lexical variations. For example, the same role may appear as:

- `dataanalysis`
- `data-analysis`
- `Data Analyst`

To address this issue, a controlled normalization mapping is applied. A dictionary-based approach maps multiple variants to a canonical job title.

Jobs that do not match any predefined pattern are assigned a default value (`Unknown`), ensuring schema stability.

6.5 Tool Extraction Using Keyword Matching

Technology tools are extracted from textual fields such as descriptions, titles, and post bodies. This extraction is performed using keyword-based matching combined with regular expressions to ensure precise detection.

The result is a list of detected tools associated with each record, enabling fine-grained technology trend analysis.

6.6 Silver Layer Output Structure

After all transformations, the cleaned datasets are stored in the Silver container using a structured and partitioned layout.

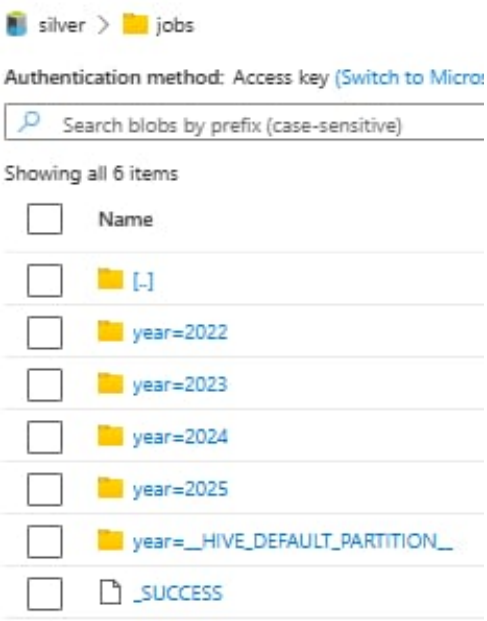


Figure 6.1: Sample of Raw Data Stored in the Bronze Layer

The Silver layer therefore acts as a stable and trusted foundation for advanced aggregations and analytics performed in the Gold layer.

Chapter 7

Gold Layer

7.1 Engagement Metrics

7.1.1 GitHub

$$Engagement = Stars + 2 \times Forks + Watchers$$

Forks are weighted higher as they indicate reuse.

7.1.2 Reddit

$$Engagement = Score + 2 \times Comments + 3 \times Awards$$

Awards represent strong community endorsement.

7.1.3 StackOverflow

$$Engagement = 0.5 \times Score + 5 \times Answers + 3 \times Favorites + \frac{Views}{1000}$$

Answers indicate problem-solving value.

7.2 Normalization

Min-Max normalization is applied to allow cross-source comparison.

7.3 Final Gold Dataset

All sources are merged and written to the Gold container.

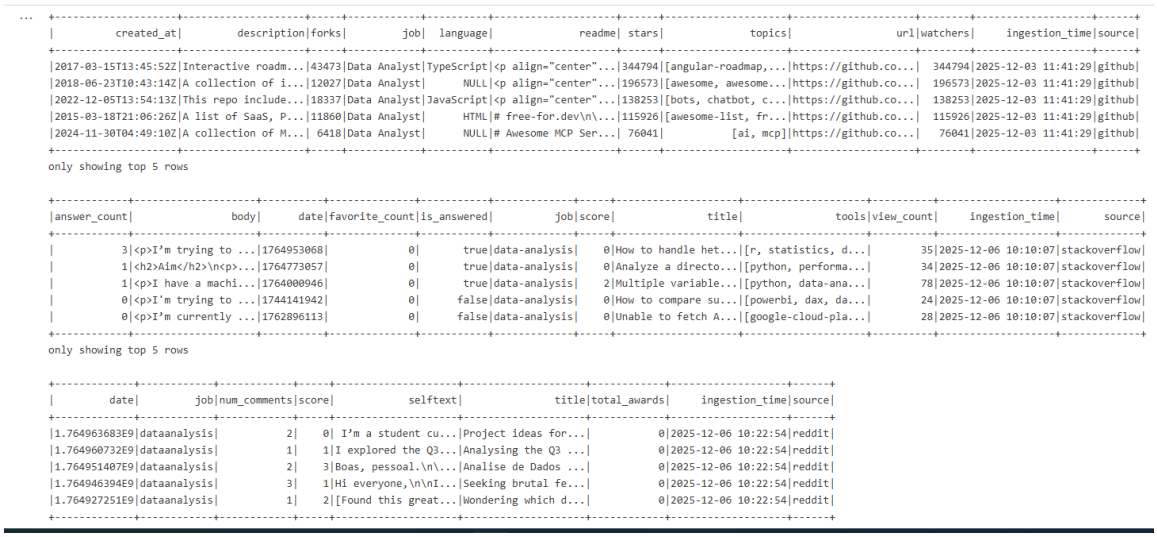
Chapter 8

Data Visualization Across Data Lake Layers

This chapter presents concrete examples of the data as it flows through the different layers of the Data Lake architecture. For each layer (Bronze, Silver, and Gold), a representative snapshot of the stored data is shown in order to illustrate the progressive refinement of the datasets.

8.1 Bronze Layer Data Sample

The Bronze layer contains raw data ingested directly from external sources without any transformation. Each record reflects the original structure returned by the APIs or datasets and is stored with an ingestion timestamp.



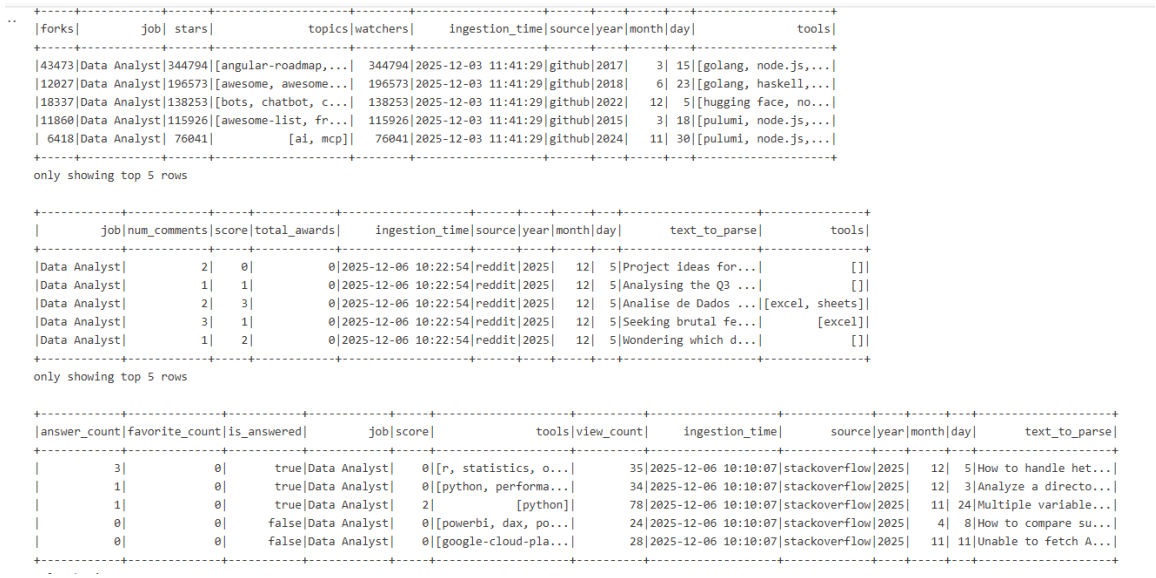
```
+--------------------+--------------------+-----+------------+----------+--------------------+------+--------------------+--------------------+--------+--------------------+------+
|         created_at|         description|forks|         job|  language|              readme| stars|              topics|                 url|watchers|     ingestion_time|source|
+--------------------+--------------------+-----+------------+----------+--------------------+------+--------------------+--------------------+--------+--------------------+------+
|2017-03-15T13:45:52Z|Interactive roadm...|43473|Data Analyst|TypeScript|<p align="center"...|344794|[angular-roadmap,...|https://github.co...|  344794|2025-12-03 11:41:29|github|
|2018-06-23T10:43:14Z|A collection of i...|12027|Data Analyst|      NULL|<p align="center"...|196573|[awesome, awesome...|https://github.co...|  196573|2025-12-03 11:41:29|github|
|2022-12-05T13:54:13Z|This repo include...|18337|Data Analyst|JavaScript|<p align="center"...|138253|[bots, chatbot, c...|https://github.co...|  138253|2025-12-03 11:41:29|github|
|2015-03-18T21:06:26Z|A list of SaaS, P...|11860|Data Analyst|      HTML|# free-for.dev\n\...|115926|[awesome-list, fr...|https://github.co...|  115926|2025-12-03 11:41:29|github|
|2024-11-30T04:49:10Z|A collection of M...| 6418|Data Analyst|      NULL|# Awesome MCP Ser...| 76041|          [ai, mcp]|https://github.co...|   76041|2025-12-03 11:41:29|github|
+--------------------+--------------------+-----+------------+----------+--------------------+------+--------------------+--------------------+--------+--------------------+------+
only showing top 5 rows

+------------+--------------------+---------+--------------+-----------+-------------+-----+--------------------+--------------------+----------+--------------------+------------+
|answer_count|                body|     date|favorite_count|is_answered|          job|score|               title|               tools|view_count|     ingestion_time|      source|
+------------+--------------------+---------+--------------+-----------+-------------+-----+--------------------+--------------------+----------+--------------------+------------+
|           3|<p>I'm trying to ...|1764953068|             0|       true|data-analysis|    0|How to handle het...|[r, statistics, d...|        35|2025-12-06 10:10:07|stackoverflow|
|           1|<h2>Aim</h2>\n<p>...|1764773057|             0|       true|data-analysis|    0|Analyze a directo...|[python, performa...|        34|2025-12-06 10:10:07|stackoverflow|
|           1|<p>I have a machi...|1764000946|             0|       true|data-analysis|    2|Multiple variable...|[python, data-ana...|        78|2025-12-06 10:10:07|stackoverflow|
|           0|<p>I'm trying to ...|1744141942|             0|      false|data-analysis|    0|How to compare su...|[powerbi, dax, da...|        24|2025-12-06 10:10:07|stackoverflow|
|           0|<p>I'm currently ...|1762896113|             0|      false|data-analysis|    0|Unable to fetch A...|[google-cloud-pla...|        28|2025-12-06 10:10:07|stackoverflow|
+------------+--------------------+---------+--------------+-----------+-------------+-----+--------------------+--------------------+----------+--------------------+------------+
only showing top 5 rows

+-------------+-----------+------------+-----+--------------------+--------------------+------------+--------------------+------+
|         date|        job|num_comments|score|            selftext|               title|total_awards|     ingestion_time|source|
+-------------+-----------+------------+-----+--------------------+--------------------+------------+--------------------+------+
|1.764963683E9|dataanalysis|           2|    0| I'm a student cu...|Project ideas for...|           0|2025-12-06 10:22:54|reddit|
|1.764960732E9|dataanalysis|           1|    1|I explored the Q3...|Analysing the Q3 ...|           0|2025-12-06 10:22:54|reddit|
|1.764951407E9|dataanalysis|           2|    3|Boas, pessoal.\n\...|Analise de Dados ...|           0|2025-12-06 10:22:54|reddit|
|1.764946394E9|dataanalysis|           3|    1|Hi everyone,\n\nI...|Seeking brutal fe...|           0|2025-12-06 10:22:54|reddit|
|1.764927251E9|dataanalysis|           1|    2|[Found this great...|Wondering which d...|           0|2025-12-06 10:22:54|reddit|
+-------------+-----------+------------+-----+--------------------+--------------------+------------+--------------------+------+
```

Figure 8.1: Sample of Raw Data Stored in the Bronze Layer

This layer serves as a historical archive and guarantees full traceability and reproducibility of the ingestion process.

8.2 Silver Layer Data Sample

The Silver layer contains cleaned and enriched data. At this stage, schema normalization, date extraction, ingestion-time control, and tool identification have already been applied.



```
+-----+-----------+------+--------------------+--------+-------------------+------+----+-----+---+--------------------+
|forks|        job| stars|              topics|watchers|     ingestion_time|source|year|month|day|               tools|
+-----+-----------+------+--------------------+--------+-------------------+------+----+-----+---+--------------------+
|43473|Data Analyst|344794|[angular-roadmap,...|  344794|2025-12-03 11:41:29|github|2017|    3| 15|[golang, node.js,...|
|12027|Data Analyst|196573|[awesome, awesome...|  196573|2025-12-03 11:41:29|github|2018|    6| 23|[golang, haskell,...|
|18337|Data Analyst|138253|[bots, chatbot, c...|  138253|2025-12-03 11:41:29|github|2022|   12|  5|[hugging face, no...|
|11860|Data Analyst|115926|[awesome-list, fr...|  115926|2025-12-03 11:41:29|github|2015|    3| 18|[pulumi, node.js,...|
| 6418|Data Analyst| 76041|          [ai, mcp]|   76041|2025-12-03 11:41:29|github|2024|   11| 30|[pulumi, node.js,...|
+-----+-----------+------+--------------------+--------+-------------------+------+----+-----+---+--------------------+
only showing top 5 rows

+-----------+------------+-----+------------+-------------------+------+----+-----+---+--------------------+--------------+
|        job|num_comments|score|total_awards|     ingestion_time|source|year|month|day|       text_to_parse|         tools|
+-----------+------------+-----+------------+-------------------+------+----+-----+---+--------------------+--------------+
|Data Analyst|           2|    0|           0|2025-12-06 10:22:54|reddit|2025|   12|  5|Project ideas for...|            []|
|Data Analyst|           1|    1|           0|2025-12-06 10:22:54|reddit|2025|   12|  5|Analysing the Q3 ...|            []|
|Data Analyst|           2|    3|           0|2025-12-06 10:22:54|reddit|2025|   12|  5|Analise de Dados ...|[excel, sheets]|
|Data Analyst|           3|    1|           0|2025-12-06 10:22:54|reddit|2025|   12|  5|Seeking brutal fe...|       [excel]|
|Data Analyst|           1|    2|           0|2025-12-06 10:22:54|reddit|2025|   12|  5|Wondering which d...|            []|
+-----------+------------+-----+------------+-------------------+------+----+-----+---+--------------------+--------------+
only showing top 5 rows

+------------+--------------+-----------+-----------+-----+--------------------+---------+-------------------+----------+----+-----+---+--------------------+
|answer_count|favorite_count|is_answered|        job|score|               tools|view_count|     ingestion_time|    source|year|month|day|       text_to_parse|
+------------+--------------+-----------+-----------+-----+--------------------+---------+-------------------+----------+----+-----+---+--------------------+
|           3|             0|       true|Data Analyst|    0|[r, statistics, o...|       35|2025-12-06 10:10:07|stackoverflow|2025|   12|  5|How to handle het...|
|           1|             0|       true|Data Analyst|    0|[python, performa...|       34|2025-12-06 10:10:07|stackoverflow|2025|   12|  3|Analyze a directo...|
|           1|             0|       true|Data Analyst|    2|            [python]|       78|2025-12-06 10:10:07|stackoverflow|2025|   11| 24|Multiple variable...|
|           0|             0|      false|Data Analyst|    0|[powerbi, dax, po...|       24|2025-12-06 10:10:07|stackoverflow|2025|    4|  8|How to compare su...|
|           0|             0|      false|Data Analyst|    0|[google-cloud-pla...|       28|2025-12-06 10:10:07|stackoverflow|2025|   11| 11|Unable to fetch A...|
+------------+--------------+-----------+-----------+-----+--------------------+---------+-------------------+----------+----+-----+---+--------------------+
```

Figure 8.2: Cleaned and Enriched Data in the Silver Layer

This intermediate layer provides a reliable and standardized dataset that can be reused for multiple analytical use cases.

8.3 Gold Layer Data Sample

The Gold layer stores analytics-ready data. Here, engagement metrics have been calculated, normalized, and aggregated to support reporting and decision-making.

job	tool	year	month	day	source	job_engagement	tool_mentions	ingestion_time
Data Analyst	rust	2018	6	23	github	5.264220115964296	12	2025-12-06 12:29:23
Software Engineer	db2	2015	3	18	github	0.25208319437704263	1	2025-12-03 11:41:29
Business Analyst	python	2024	11	19	github	1.785723649862347	12	2025-12-06 12:29:23
Data Engineer	oracle	2015	3	18	github	0.25208319437704263	1	2025-12-03 11:41:29
Business Analyst	codecommit	2015	3	18	github	3.026877404902148	12	2025-12-06 12:29:23
Cloud Engineer	firebase	2015	3	18	github	3.026877404902148	12	2025-12-06 12:29:23
Cloud Engineer	codecommit	2015	3	18	github	3.026877404902148	12	2025-12-06 12:29:23
Data Scientist	haskell	2018	6	23	github	0.4379526620958672	1	2025-12-03 11:41:29
Data Engineer	typescript	2015	3	18	github	0.25208319437704263	1	2025-12-03 11:41:29
Machine Learning ...	huggingface	2015	3	18	github	0.25208319437704263	1	2025-12-03 11:41:29
Machine Learning ...	java	2024	11	19	github	1.785723649862347	12	2025-12-06 12:29:23
Machine Learning ...	ruby	2024	11	19	github	1.785723649862347	12	2025-12-06 12:29:23
Data Analyst	redis	2018	6	23	github	0.4379526620958672	1	2025-12-03 11:41:29
Data Scientist	mysql	2018	6	23	github	0.4379526620958672	1	2025-12-03 11:41:29
Software Engineer	docker	2018	6	23	github	0.4379526620958672	1	2025-12-03 11:41:29
Data Engineer	python	2019	10	3	github	2.2135196612609795	12	2025-12-06 12:29:23
Business Analyst	golang	2018	6	23	github	0.4379526620958672	1	2025-12-03 11:41:29
Cloud Engineer	java	2024	11	30	github	1.778347658289423	12	2025-12-06 12:29:23
Data Engineer	python	2018	6	23	github	0.4379526620958672	1	2025-12-03 11:41:29
Machine Learning ...	wire	2018	6	23	github	0.4379526620958672	1	2025-12-03 11:41:29

only showing top 20 rows

Figure 8.3: Analytics-Ready Data in the Gold Layer

The Gold layer represents the final output of the data warehouse and is optimized for querying through Azure Synapse SQL and analytical tools.

Chapter 9

Data Warehouse Modeling

9.1 Schema Overview

The data warehouse is modeled using a **Galaxy schema** (also known as a fact constellation schema). This design is chosen because the project contains **multiple fact tables** that share common dimensions.

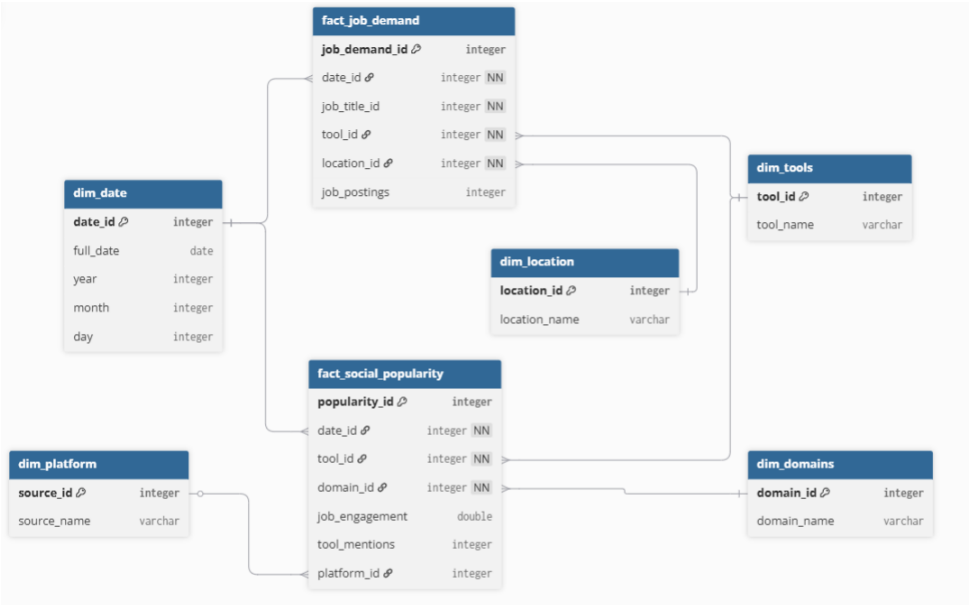


Figure 9.1: Galaxy Schema of the Data Warehouse

Each fact table can be analyzed independently, while shared dimensions enable cross-analysis between popularity and demand.

9.2 Fact Tables

Two main fact tables are implemented:

- **Social Popularity Fact**: stores engagement metrics and tool mentions collected from social and developer platforms.
- **Job Demand Fact**: stores historical job market demand metrics extracted from structured datasets.

These facts represent different analytical perspectives but use a consistent dimensional structure.

9.3 Dimension Tables

The fact tables share multiple dimensions, including:

- **Date**: enables time-based analysis
- **Tools**: represents technologies and skills
- **Domains**: represents job roles and categories

Additional dimensions are used where relevant:

- **Source**: identifies the origin platform of popularity data
- **Location**: represents geographic information for demand data

9.4 Staging Tables

Staging tables are used as an intermediate logical layer between the Gold files stored in the Data Lake and the final warehouse tables. They provide a structured representation of the external Parquet data and act as a stable interface for loading dimensions and facts.

These tables:

- Reflect the schema of the Gold datasets
- Do not store data physically in Synapse
- Enable controlled and incremental data loading

9.5 External Data Access

The warehouse relies on external querying of Parquet files stored in the Gold layer of the Data Lake. This approach avoids data duplication and ensures scalability by separating storage from compute.

9.6 Incremental Loading

An incremental loading strategy is applied to ensure that only new data is inserted into the warehouse. This mechanism prevents duplication, reduces processing cost, and supports near real-time updates for popularity data.

9.7 Summary

The Galaxy schema provides a flexible and scalable modeling approach that supports multiple analytical use cases while maintaining consistency across shared dimensions.

Chapter 10

Data Visualization and Business Insights

This chapter presents the main analytical visualizations created using **Power BI**. These dashboards leverage the Gold layer of the data warehouse to analyze job market demand and compare it with real-time popularity indicators.

—

10.1 Job Demand During 2023

This visualization shows the evolution of job demand across different domains during the year 2023.

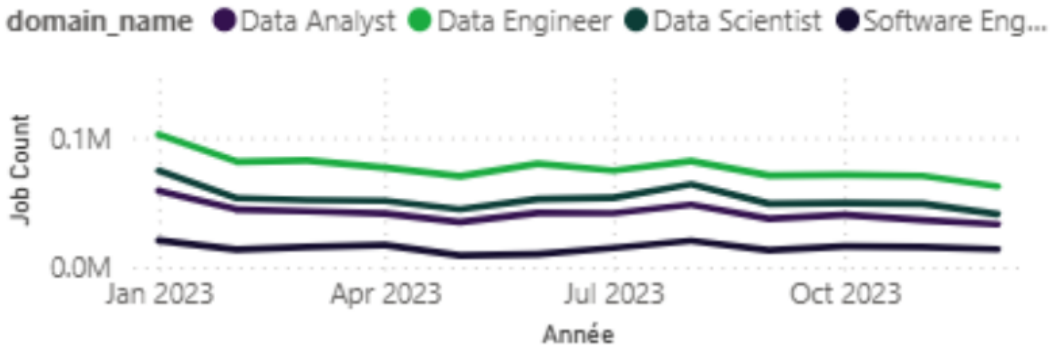


Figure 10.1: Job Demand Trends During 2023

—

10.2 Demand for Common Tools Across Four Job Roles

This dashboard highlights tools that are commonly demanded across four major job roles, illustrating shared skill requirements in the market.

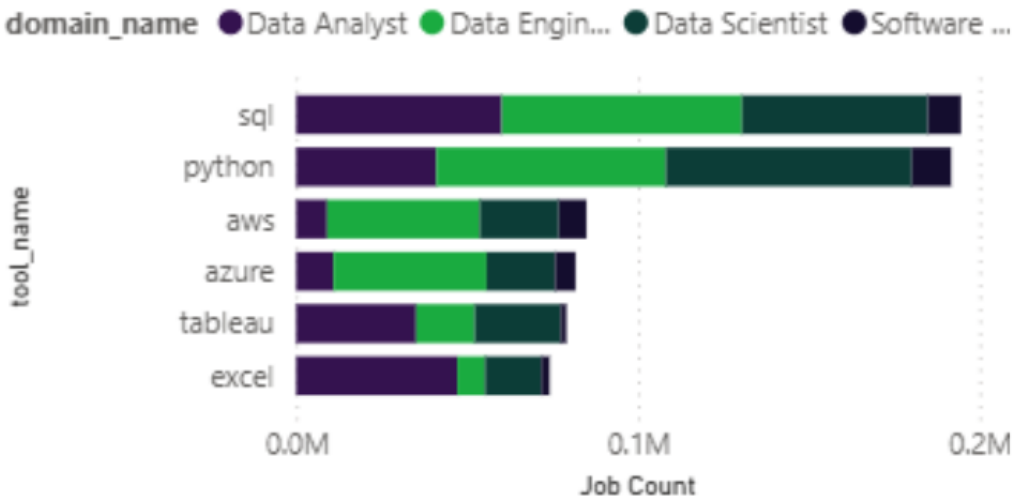


Figure 10.2: Demand for Common Tools Across Four Job Roles

—

10.3 Tool Demand for Data Analysts

This visualization focuses on the most demanded tools specifically for the Data Analyst role.

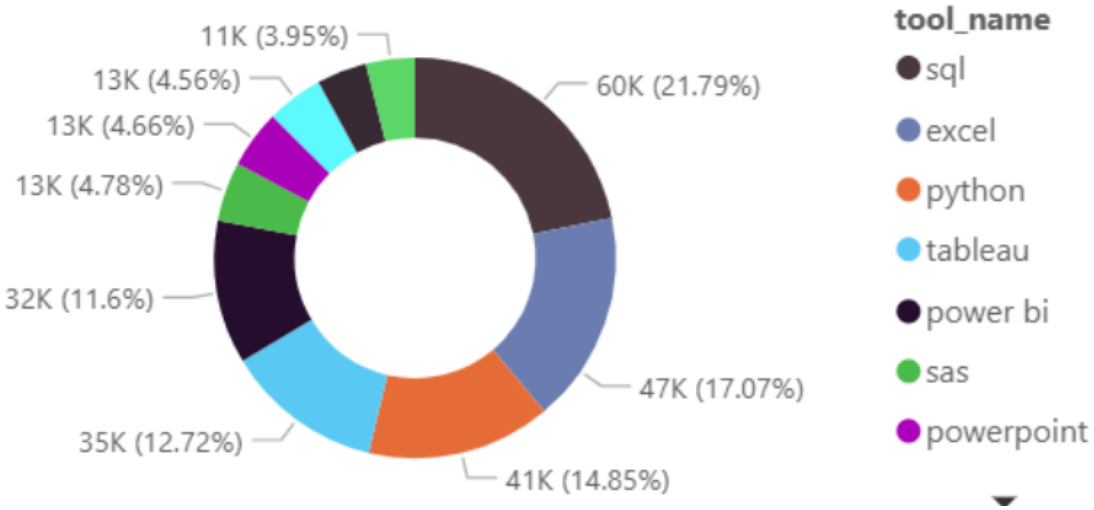


Figure 10.3: Tool Demand for Data Analysts

—

10.4 Data Engineering Demand by Country

This dashboard presents the geographic distribution of Data Engineering demand across different countries.

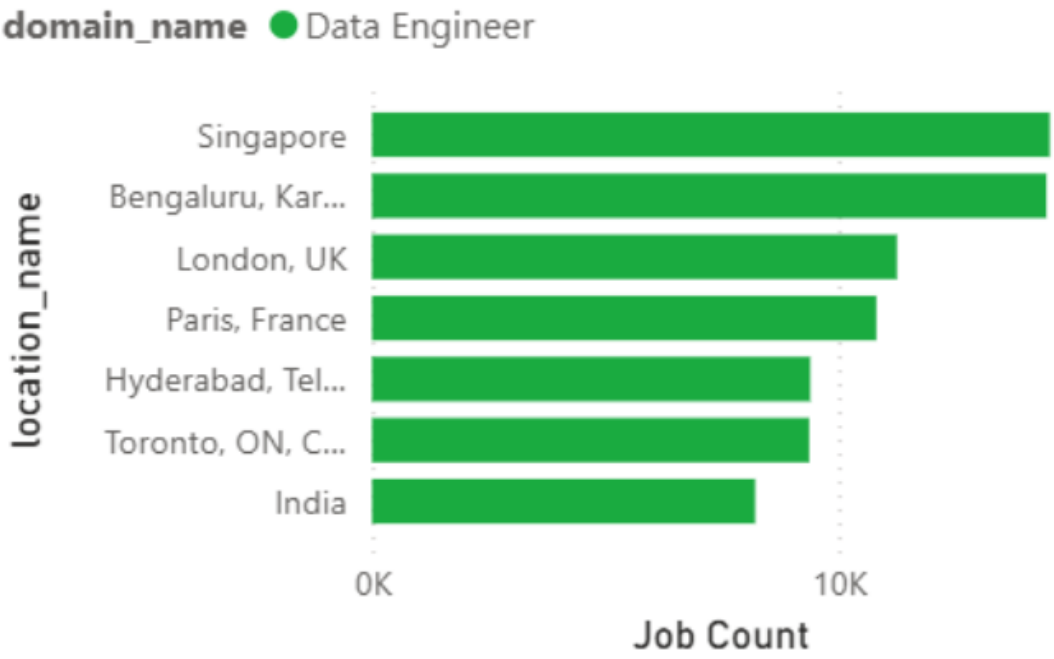


Figure 10.4: Data Engineering Demand by Country

—

10.5 Data Domain Demand on a Geographic Map

This map-based visualization shows how data-related domains are distributed geographically.

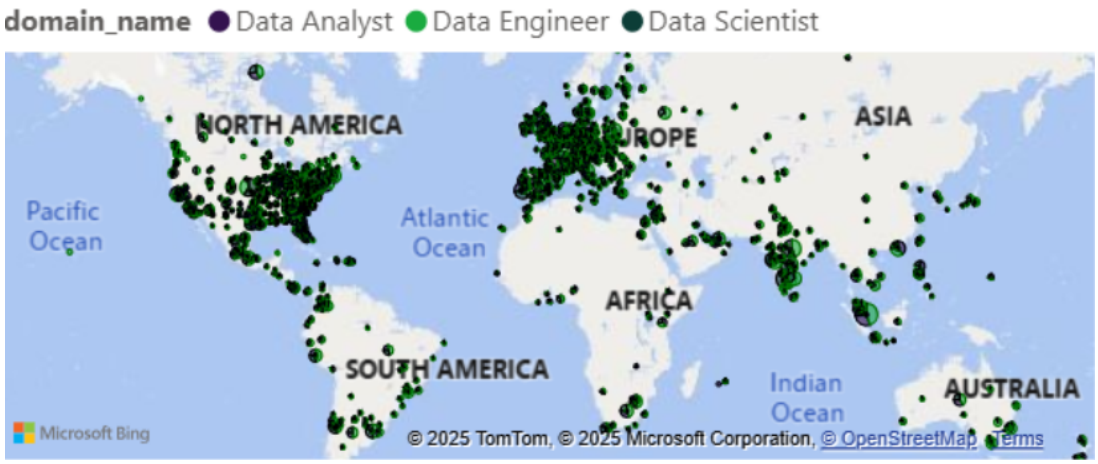


Figure 10.5: Geographic Distribution of Data Domains

—

10.6 AWS vs Azure Demand for Cloud Engineers

This visualization compares the demand for AWS and Azure skills among Cloud Engineers.

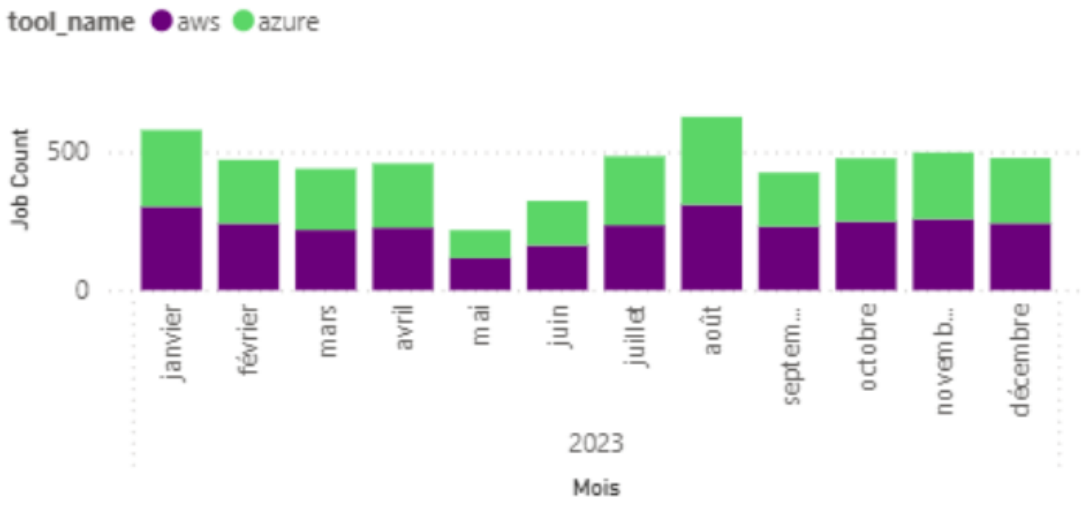


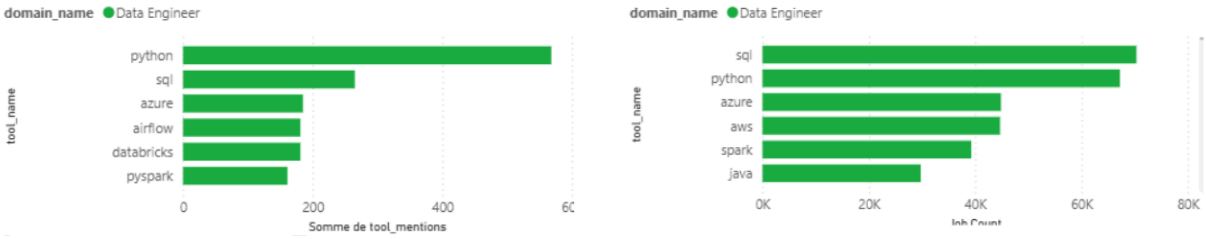
Figure 10.6: AWS vs Azure Demand for Cloud Engineers

—

—

10.7 Data Engineer Tools: Popularity vs Demand

This comparison highlights the gap between real-time popularity and actual market demand for Data Engineer tools.



Popularity Demand

Figure 10.7: Popularity vs Demand of Data Engineer Tools

Chapter 11

Conclusion

This project demonstrates the design and implementation of a modern, scalable data warehouse on Microsoft Azure for analyzing technology job trends. By combining real-time popularity signals from developer communities with historical job market demand, the solution provides a comprehensive view of how tools and roles evolve over time.

The adopted architecture, based on a layered Data Lake approach (Bronze, Silver, and Gold), ensures data traceability, incremental processing, and analytical reliability. Azure Functions enable automated and near real-time data ingestion, while Azure Synapse provides both distributed processing and analytical modeling capabilities.

The use of a Galaxy schema allows multiple analytical perspectives to coexist while sharing common dimensions, facilitating cross-analysis between popularity and demand. Power BI visualizations built on top of the Gold layer transform raw data into actionable insights, supporting decision-making for professionals, organizations, and learners.

Overall, this project highlights how cloud-native services and modern data engineering practices can be combined to build an efficient, cost-aware, and extensible analytical platform. Future improvements may include advanced forecasting, real-time dashboards, and automated orchestration to further enhance the system's analytical value.