Technical Documentation: FakeSchoolData Project

Project completed by: Aymane RAMI

2025

Table des matières

[Project Objective 3](#_Toc201777740)

[Technologies Used 3](#_Toc201777741)

[Repository Structure 3](#_Toc201777742)

[Completed Steps 4](#_Toc201777743)

[Simulated Data Generation (Python script generate\_data.py) 4](#_Toc201777744)

[Snowflake Database Creation and Data Loading 4](#_Toc201777745)

[Data Transformation with dbt 4](#_Toc201777746)

[Analysis with Python Script analyze\_results.py 5](#_Toc201777747)

[Automation with GitHub Actions 5](#_Toc201777748)

[Next Steps 6](#_Toc201777749)

# Project objective

To build a simulated data pipeline for a fictional school, leveraging:

* A Python script to generate CSV files for students, courses, and results.
* A Snowflake database to store the data.
* Data transformations using dbt.
* Results analysis (statistics and visualizations) via another Python script.
* Automation with GitHub Actions.

# Technologies used

* Python 3.10
* Libraries: pandas, matplotlib, snowflake-connector-python, faker
* Database: Snowflake
* Transformation tool: dbt
* CI/CD: GitHub Actions

# Repository structure

pgsql

Copier le code

FakeSchoolData/

├── .github/

│ └── workflows/

│ └── run\_analysis.yml

├── analyze\_results.py

├── average\_grades\_chart.png

├── courses.csv

├── generate\_data.py

├── log/

├── results.csv

├── students.csv

├── venv/

├── .gitignore

├── fakeschool\_dbt/

│ └── models/

│ ├── average\_grades.py

│ ├── schema.yml

│ └── top\_students.py

# Completed steps

## Simulated Data Generation (Python script generate\_data.py)

* + Used Faker to generate:
    - 100 students with ID, first name, and last name.
    - 10 courses with ID and random names.
    - Random results (grades from 0 to 20) for each student across multiple courses.
  + Saved data into three CSV files: students.csv, courses.csv, results.csv.

## Snowflake database creation and data loading

* + Connected to Snowflake using Python (snowflake-connector-python).
  + Created the RAW schema and tables STUDENTS, COURSES, and RESULTS.
  + Loaded CSV files into Snowflake via a staging area and used COPY INTO commands to insert data.

## Data transformation with dbt

* + Initialized a dbt project named fakeschool\_dbt.
  + Configured Snowflake connection in profiles.yml.
  + Created models inside fakeschool\_dbt/models/:
    - average\_grades.py: calculates average grades per course.
    - top\_students.py: identifies the top 5 students with the highest averages.
    - schema.yml: documents and validates the data structure.
  + Compiled and executed transformations using dbt run to create transformed tables or views.

## Analysis with Python script analyze\_results.py

* + Connected to Snowflake and executed SQL queries to fetch enriched data.
  + Performed statistical calculations:
    - Mean, median, and standard deviation of grades per course.
    - Number of students per course.
    - Minimum and maximum grades per student.
    - Top 5 students by average grade.
  + Visualized results using matplotlib:
    - Histogram showing grade distribution (average\_grades\_chart.png).
    - Bar chart showing grade ranges (0-5, 6-10, etc.) (grade\_distribution\_chart.png).

## Automation with GitHub Actions

* + Configured workflow .github/workflows/run\_analysis.yml to:
    - Automatically run analyze\_results.py on every push to the main branch.
    - Schedule daily runs at 8:00 AM UTC via a cron job.
  + Setup includes:
    - Checking out the repository.
    - Installing Python and required dependencies.
    - Securely passing the SNOWFLAKE\_PASSWORD via GitHub Secrets.
    - Running the script from the correct working directory.
  + Generated charts are saved and uploaded as GitHub Actions artifacts for easy access.

# Next steps

* Add export of additional result files (e.g., CSV exports) if needed.
* Extend dbt models with tests and automated documentation.
* Optionally implement automatic email reporting.
* Enhance data visualization and reporting capabilities.