Student Performance Dashboard

Project Description

Build a scalable data engineering and analytics pipeline that ingests, cleans, stores, and visualizes student performance data. The system will process ~10 million records, produce a cleaned/partitioned dataset, populate a queryable database, and deliver an interactive dashboard that surfaces trends and actionable insights for educators and administrators.

---

2. Group Members & Roles

- Abdulrahman Mustafa — Data Engineer (ETL, storage, performance tuning)
- Malak Mohamed — Data Engineer / Analyst (preprocessing, validation, SQL)
- Khaled Ehab — Data Analyst / Visualization Specialist (SQL analysis, charts, dashboard) — *Team Leader*
- Eman Wassem — Documentation & Presentation Lead
- Ahmed Mohamed Galal — Documentation & Presentation Lead

All members will assist cross-functionally as needed.

---

3. Team Leader

Khaled Ehab

---

4. Objectives

1. Implement an efficient ETL pipeline that reliably cleans and standardizes the raw data (10M+ records).

2. Design and populate a normalized SQL schema that supports fast analytical queries.

3. Create clear, interactive visualizations and a dashboard that covers the project's KPIs.

4. Deliver full project documentation and a stakeholder-ready presentation.

---

5. Tools & Technologies (primary + potential extensions)

Python (Pandas, PySpark), Parquet, DuckDB / PostgreSQL, Matplotlib / Seaborn / Plotly, Git/GitHub. (Optional / future: Airflow / Prefect, dbt, Snowflake / BigQuery)

---

6. Milestones & Deadlines

Milestone	Deliverables	Deadline
M1 — Data Preprocessing	Cleaned dataset (Parquet), ETL scripts, data dictionary	Oct 31, 2025

M2 — SQL Integration	Database schema, loaded tables, tested queries (notebook)	Nov 7, 2025
M3 — Visualization	Visualization notebook + interactive dashboard (Streamlit/Plotly)	Nov 20, 2025
M4 — Presentation	Final report (PDF) + slide deck + demo	Dec 1, 2025

---

7. KPIs

A. Data Preprocessing

- Data quality: % of missing/duplicate records handled → Target: 100% (no unhandled duplicates; explanation recorded).
- ETL throughput: Time to process and write cleaned Parquet for full dataset → Target: ≤ 5 minutes.
- Storage efficiency: Parquet compression target reduces raw CSV size by ≥ 60%.

B. SQL Integration (Schema & Queries)

- Schema quality: Normalized schema (≥ 3NF) and clear foreign-key relationships → Target: 100% design coverage.
- Query accuracy: Test queries must match expected results on validation set → Target: ≥ 95%.

- Query performance: Representative analytical queries (joins, group-bys) execute on indexed tables of 10M+ rows → Target: average < 5 seconds (local test with dataset; exact target will be reported per query).

C. Visualization (Charts & Dashboard)

- Coverage: Dashboard visualizes ≥ 90% of defined KPIs/metrics (grades, attendance trends, top performers, cohort analysis).
- Load performance: Dashboard initial load & core view rendering → Target: < 5 seconds on a reasonable local machine; aim for < 3 seconds on trimmed views.
- Usability: Internal reviewer score for clarity and usefulness → Target: ≥ 4/5.

D. Presentation & Documentation

- Completeness: Final report contains all required sections (methodology, results, limitations, reproducibility) → Target: 100%.
- Stakeholder clarity: Presentation receives feedback score ≥ 4/5 from evaluators.

Validation methods: execution time measured with Python profiling, ETL logs; query timing via `EXPLAIN ANALYZE` (Postgres) or DuckDB timing; dashboard load-time measured with browser dev tools / simple timing script; reviewer feedback from at least two independent evaluators.

Short note on realism & scope

Targets above are adjusted to the dataset scale (≈10M records). We prioritize reproducibility and measurable validation (timings, test queries, logs). If infrastructure or resource constraints appear (e.g., limited CPU/memory), we will

document them and provide scaled performance targets or a cloud-run alternative.