Student Data Pipeline & Analytics Project

Technical Report

By: Miranda Insights LLC

Developed: March 2026

1. Project Overview

This project demonstrates the development of an end-to-end data pipeline and analytics workflow using a simulated Student Information System (SIS) dataset. The objective was to design a scalable data architecture, perform data validation and transformation, and deliver actionable insights through dashboards, reports, and presentations.

The project integrates database design, API data extraction, statistical analysis, and business intelligence tools to showcase a full data lifecycle from raw data ingestion to final stakeholder communication.

---

2. Data Source & Simulation

The dataset was simulated to replicate a real-world SIS environment, including:

- Student demographic data
- Attendance records
- Assessment scores
- School-level metadata

This approach allowed for controlled testing of data engineering, validation, and reporting processes.

---

3. Database Design & Implementation (Supabase / SQL)

A relational schema was designed and implemented in Supabase using PostgreSQL.

Key Steps:

- Defined normalized tables: students, attendance, assessments, schools
- Established primary and foreign key relationships
- Imported structured CSV datasets into the database

Data Validation:

- Executed SQL queries to validate relationships and data integrity
- Identified duplicate records during validation
- Traced the issue back to the source dataset

Data Correction:

- Corrected duplicate data within the simulated SIS source
- Re-imported clean datasets after removing affected tables
- Revalidated data consistency post-import

Security:

- Implemented Row-Level Security (RLS) policies to control access at the record level

---

4. Data Extraction & Engineering (Python API Pipeline)

A Python-based data pipeline was developed to extract data from Supabase via API.

Tools:

- Python
- Pandas

Process:

- Authenticated using API URL and key
- Queried each table dynamically
- Loaded responses into Pandas DataFrames

- Performed joins and transformations for downstream analysis

This step enabled flexible data manipulation and integration across multiple tools.

---

5. Dashboard Development (Quarto + Python + R)

A multi-language Quarto document was developed to generate a **self-contained HTML dashboard**.

Tools:

- Quarto
- R
- Python (via reticulate)

Features:

- Integrated Python for data extraction and transformation
- Used R for statistical visualization and formatting
- Generated a fully portable HTML dashboard with embedded resources

This demonstrated the ability to combine multiple languages for optimal performance and presentation.

---

6. Excel Analytics & Dashboarding

To demonstrate business-facing analytics skills, data was exported into Excel using a custom Python script.

Enhancements:

- Created additional summary tables during export

- Structured datasets for pivot analysis

Excel Features Used:

- Pivot Tables
- Conditional Formatting (risk indicators)
- KPI Cards (average scores, attendance metrics)
- Interactive Dashboard with slicers

This step emphasized usability for non-technical stakeholders.

---

7. Data Visualization for Presentation (R)

An R script was developed to generate static, presentation-ready visualizations.

Features:

- Box plots with statistical indicators (Q1, median, Q3, min, max, mean)
- Color scaling for performance distribution
- Attendance comparison charts
- Summary statistic tables exported alongside visuals

This ensured clarity and consistency across presentation materials.

---

8. Presentation Development

A structured presentation was created to communicate findings effectively.

Components:

- Problem framing
- Data overview
- Subgroup analysis (EL, SPED, Grade, School)
- Visual insights and supporting statistics
- Technology stack overview

This stage focused on translating technical work into business-relevant storytelling.

---

9. Version Control & Project Management

Version control was maintained throughout the project using GitHub.

Practices:

- Regular commits and updates
- Organized repository structure
- Separation of scripts, outputs, and documentation

This ensured reproducibility and transparency.

---

10. Conclusion

This project demonstrates the ability to design and execute a full data pipeline, from database architecture to stakeholder-ready deliverables. It highlights strengths in data engineering, analytics, visualization, and communication across multiple platforms.

Q.E.D.

(*Quod Erat Demonstrandum — "Thus it has been demonstrated."*)