# **Formative 1 – Data Quality and Performance in Action**

For this formative, I worked with a dataset containing 770 student records, which included demographic information, gaming habits, and academic performance. Before I could begin any meaningful analysis, I had to clean and prepare the data. I evaluated its quality based on five key dimensions: **accuracy**, **completeness**, **consistency**, **timeliness**, and **uniqueness**.

**Part 1 Data Quality Assessment and Improvement**

One of the first things I addressed was the **‘Sex’ column**, which originally used numeric codes (0 and 1) to represent gender. These codes weren’t very clear, so I replaced them with the actual labels “Male” and “Female” using a simple formula in Excel. This small change helped improve both readability and consistency in the dataset. (fig1)

Next, I focused on the **‘Percentage’ column**, which had a lot of formatting issues. Some values had percent signs, commas, or even double decimal points (e.g., “75..5%”). To handle this, I created two formulas. The first flagged entries with incorrect formatting (fig2), and the second attempted to clean and convert the values into a consistent decimal format. For example, “75.5%” became 0.755 (fig3) Any entries that were too messy to fix automatically were flagged for manual review.

I then checked for **missing values** using a regular expression pattern to identify empty cells (fig4). Fortunately, there weren’t any, so the dataset passed the completeness check. I also carried out several rounds of checks for **duplicate records**, and none were found, ensuring **uniqueness**. However, I wasn’t able to fully assess **accuracy** or **timeliness** because the dataset lacked source information and timestamps.

Raw Data:

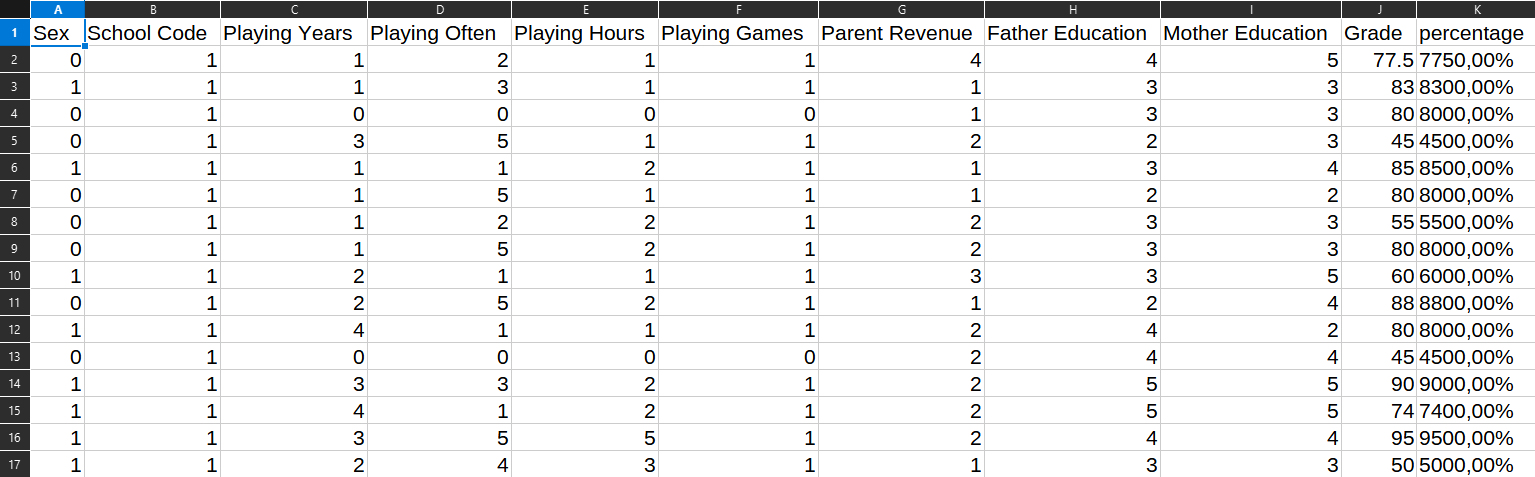


Fig 1:

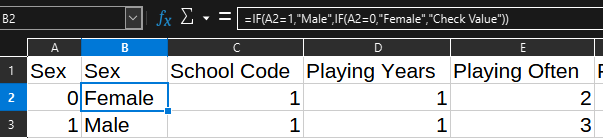


Fig 2:

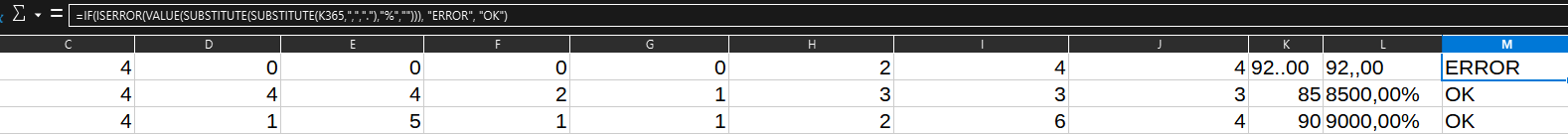
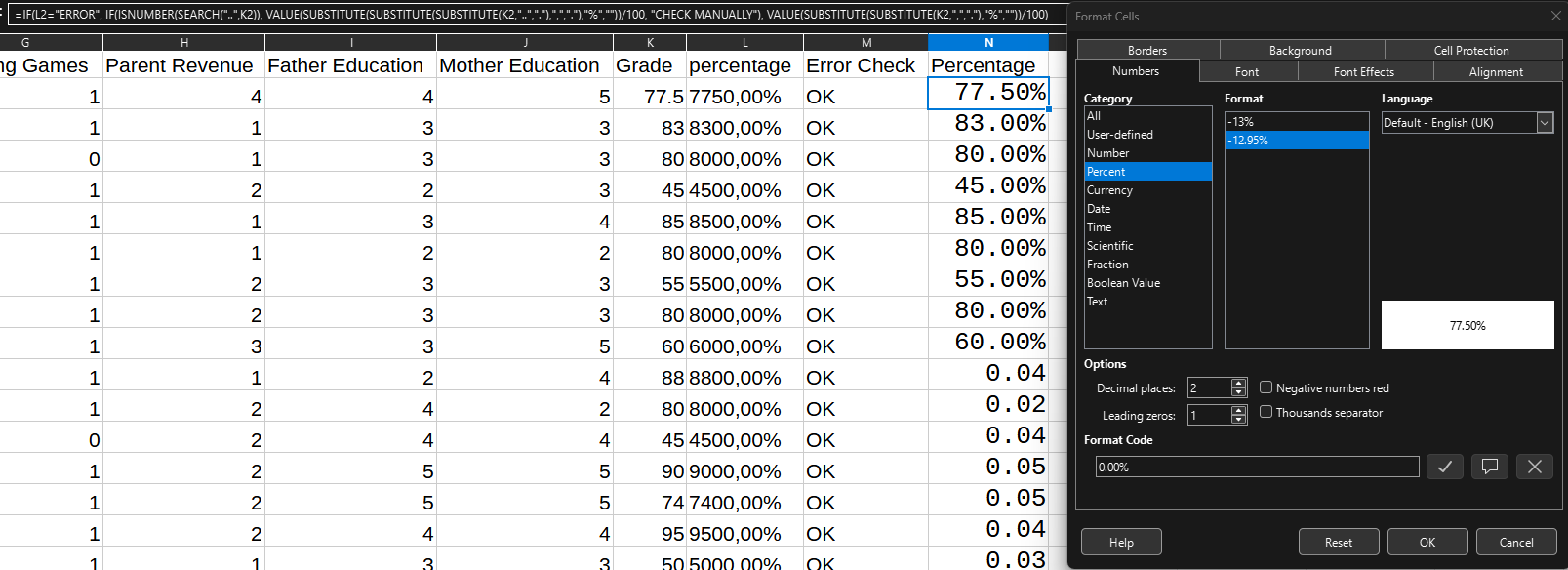
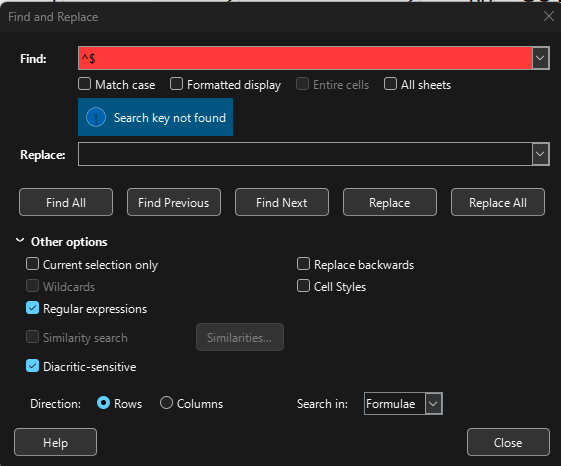


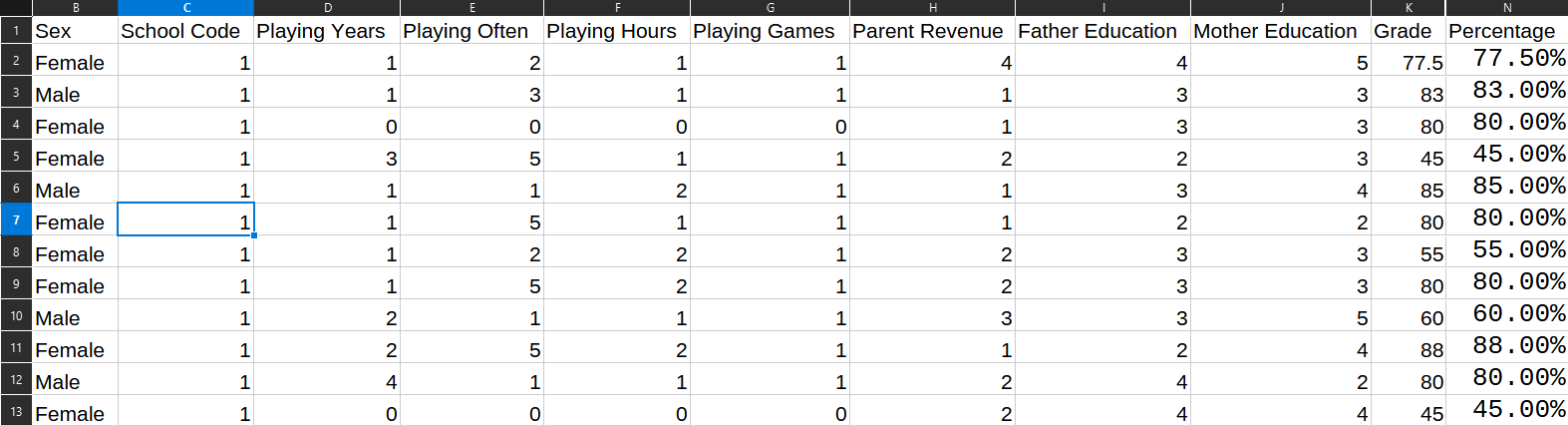
Fig 3:



**Fig 4:**



**Clean Data:**



**Part 2 Database Schema Design with SQL as a DDL**

After cleaning the data, I designed a **star schema (fig1)** to make future analysis easier and more efficient. The central **fact table** (fact\_grades, fig 2) stored the grades and an error flag to track any data issues. This table connected to four **dimension tables**:

* dim\_student: included demographic data such as sex.
* dim\_school: captured school identifiers.
* dim\_parent\_background: stored parental education (on a scale of 0–10) and income (scale of 1–4).
* dim\_gaming\_habits: recorded gaming behaviour, including how many years the student had played, how often, and for how long per session.

I enforced constraints to keep values within valid ranges, created indexes for faster joins, and enabled foreign key support in SQLite using PRAGMA foreign\_keys = ON(fig3);. During the data load , I encountered a formatting error in one column, which I had to fix manually — a reminder that even with automation, human oversight is still important.

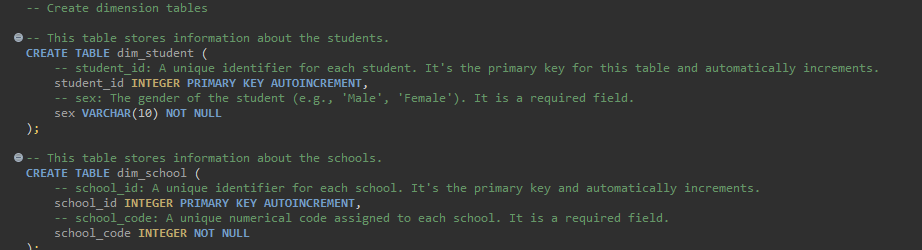
### **Example Use Case**

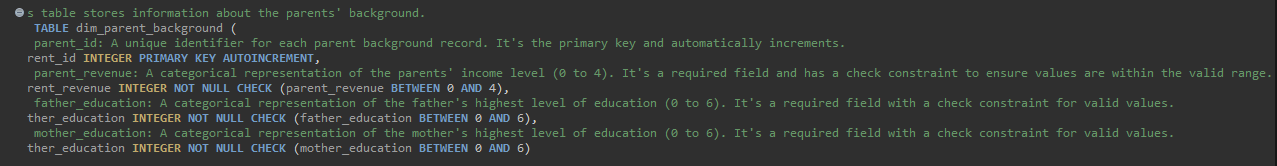
Using the schema, I compared academic performance between **frequent gamers** (those with a playing frequency of 4 or more) and **non-gamers** (students with zero years of gaming). I filtered the results by high parental education levels to explore how socioeconomic factors might influence the outcomes.(fig4)

### **Conclusion**

This project reinforced the importance of proper data cleaning and schema design. While there were some challenges — especially with formatting and missing metadata — the end result was a well-structured and reliable dataset that’s ready for deeper analysis.

Fig 1:





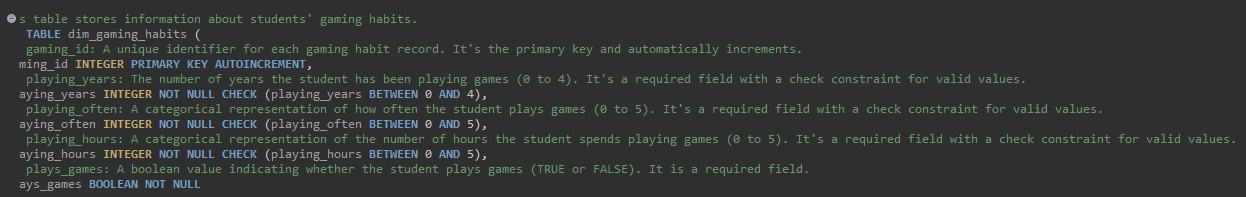


Fig 2:

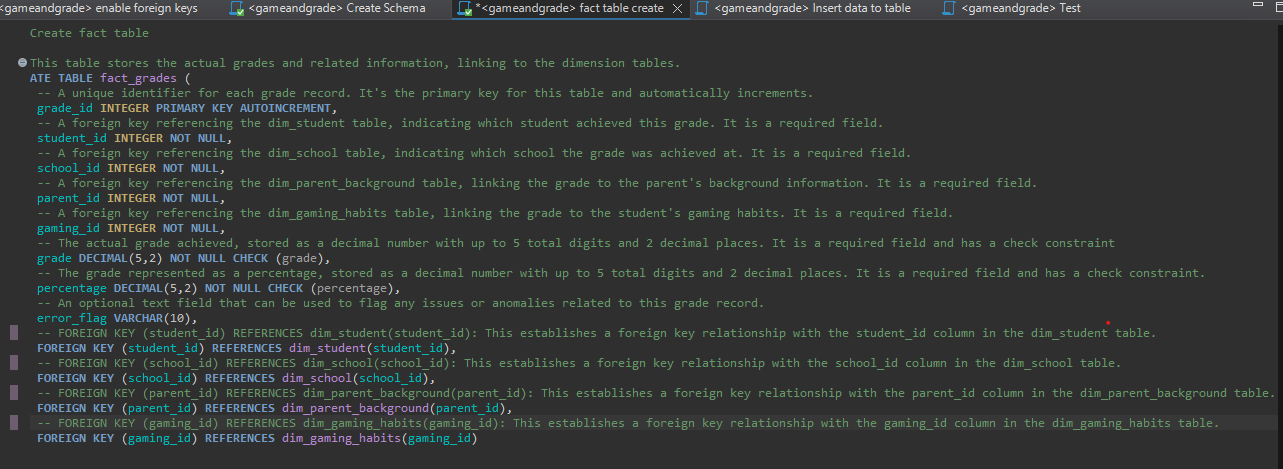


Fig 3:

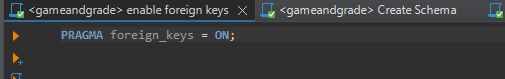


Fig 4:

