**Formative 1-**Data quality and performance in action****

**Part 1 – **Data Quality Assessment and Improvement:****

The purpose of this report is to conduct a data quality assessment on a dataset consisting of 770 student records, including demographics, gaming habits, and academic performance. The evaluation focused on five core data quality dimensions: **accuracy**, **completeness**, **consistency**, **timeliness**, and **uniqueness**.

A major goal of this task was to identify and resolve any issues in data formatting and structure, particularly in the percentage and Grade columns, to ensure the dataset is suitable for analysis or modelling.

As part of the cleaning process, several steps were taken:

**Step 1: Convert numbered values in ‘Sex’ Column**

The Sex column was encoded as 0 and 1. These values were mapped to "Female" and "Male" respectively, assuming 1 = Male 0 = Female

*=IF(A2=1,"Male",IF(A2=0,"Female","Check Value"))*

**Step 2: Percentage Column Cleaning:**

* A new column was created to detect formatting errors:

*=IF(ISERROR(VALUE(SUBSTITUTE(SUBSTITUTE(K2;",";".");"%";""))), "ERROR", "OK")*

* A second formula was applied to clean and normalize the values:

*=IF(L2="ERROR", IF(ISNUMBER(SEARCH("..",J2)), VALUE(SUBSTITUTE(SUBSTITUTE(SUBSTITUTE(J2,"..","."),",","."),"%",""))/100, "CHECK MANUALLY"), VALUE(SUBSTITUTE(SUBSTITUTE(J2,",","."),"%",""))/100)*

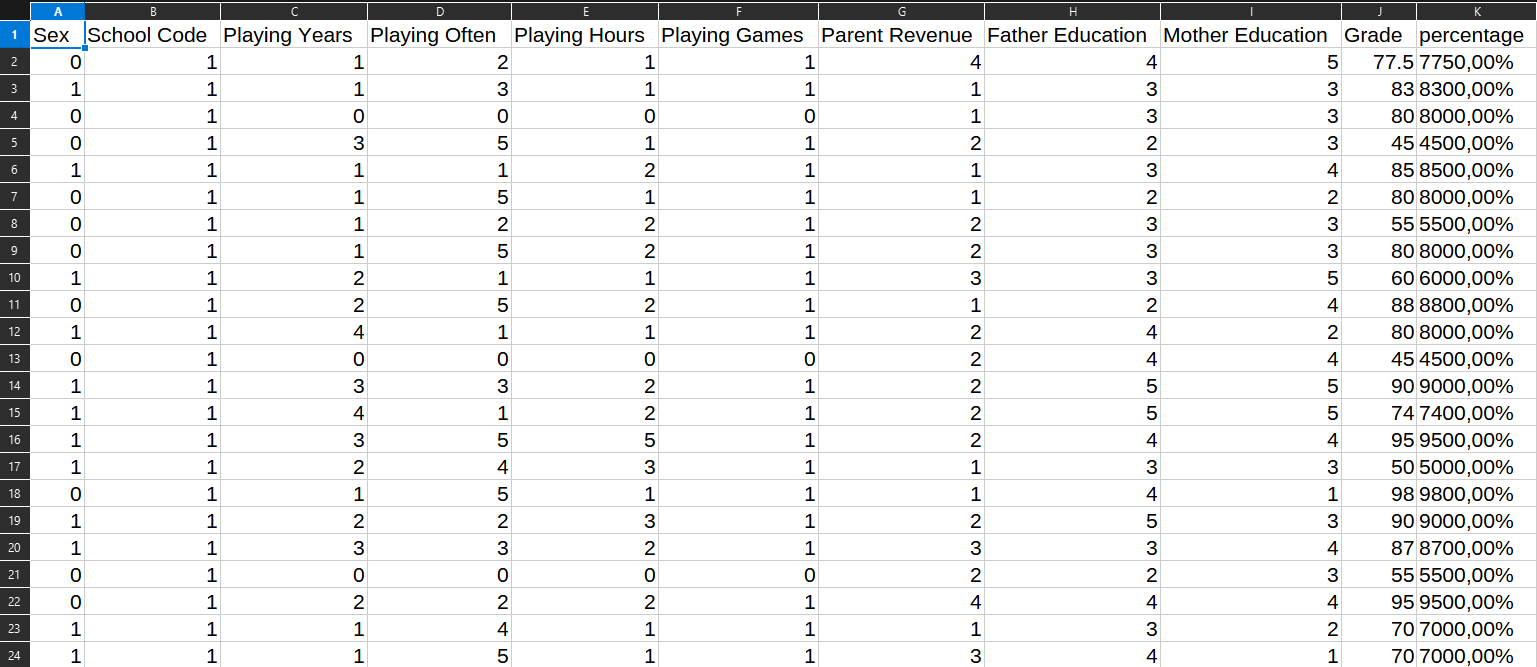
This formula checks if cell L2 contains "ERROR"; if so, it further checks if cell J2 contains a double dot, and if it does, cleans and converts J2 to a percentage, otherwise flags it for manual check; if L2 is not "ERROR", it cleans and converts J2 to a percentage.

**Step 3: Missing Value Detection**:

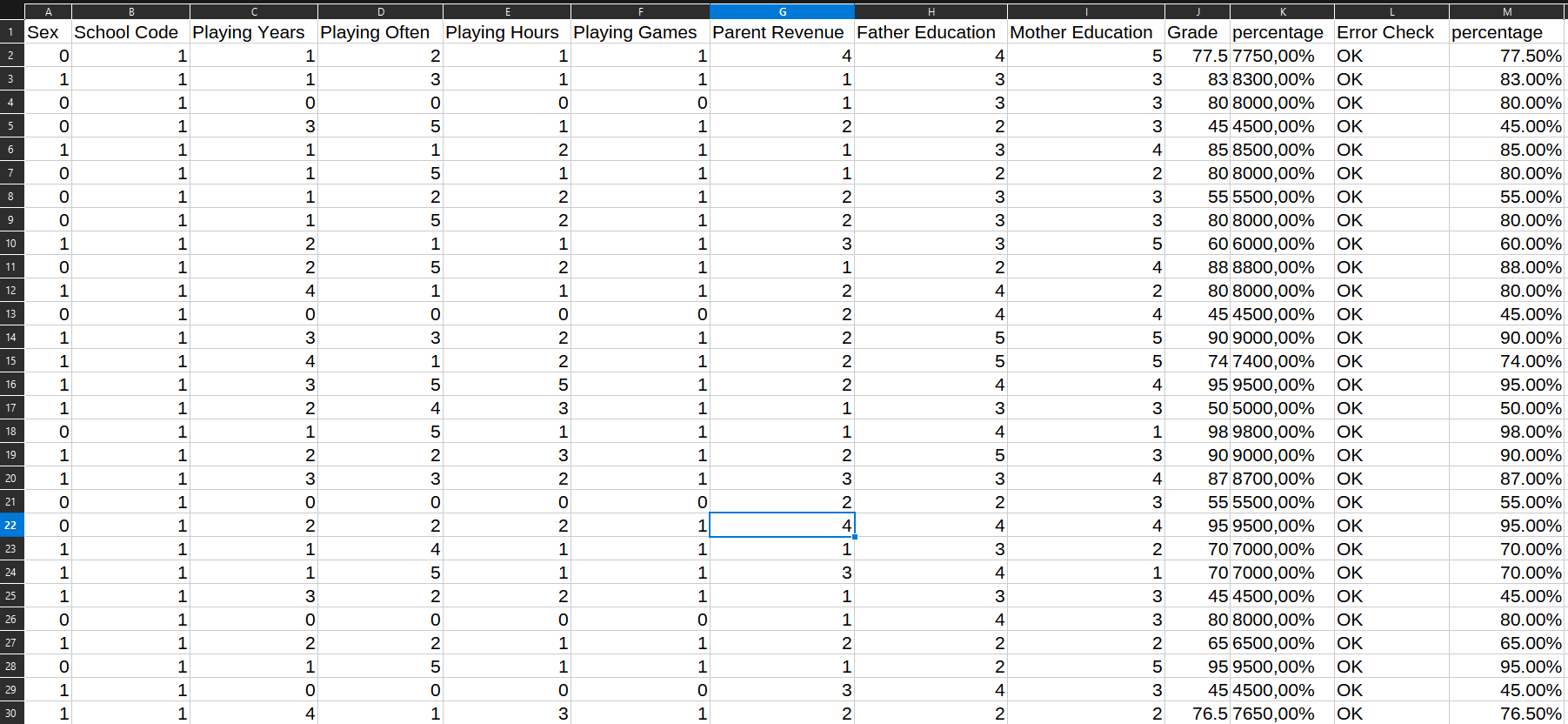
* Using **Find & Replace** with regular expressions (^$), empty cells were searched across the dataset. No missing values were found in the clean data.

Through these steps, data types were corrected, formatting inconsistencies were resolved, and the dataset was verified for completeness and uniqueness. While **timeliness** and **accuracy** could not be fully assessed due to limited metadata, the dataset is now well-structured and ready for use.

Before:



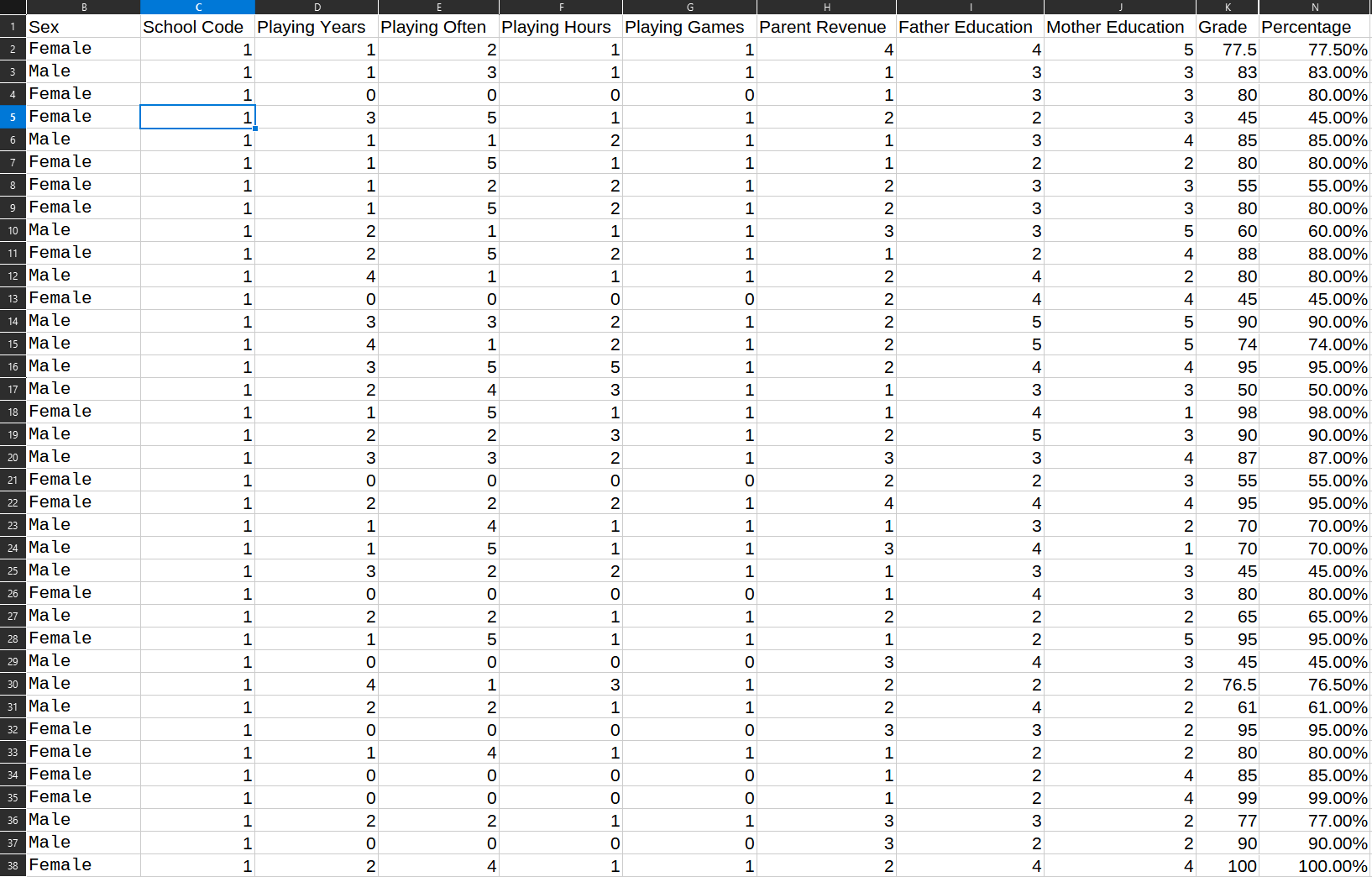
During Clean:



Error:



Clean:



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

This schema is designed to analyse the relationship between students' gaming habits and their academic performance. It uses a **star schema**, which is optimized for analytical queries, with one central fact table linked to multiple dimension tables.

## Fact Table

* fact\_grades: Stores academic performance data (grades and percentages) along with foreign keys to all dimensions.
  + **Fields**: grade, percentage, error\_flag (data validation).
  + **Connections**: Links to student, school, parent background, and gaming habits.

## Dimension Tables

1. dim\_student: Contains student demographics (sex).
2. dim\_school: Lists schools (only school codes in the dataset).
3. dim\_parent\_background: Tracks parental education (0-10 scale) and income (1-4 scale).
4. dim\_gaming\_habits: Records gaming behavior:
   * playing\_years (0-4 years),
   * playing\_often (0-5 frequency scale),
   * playing\_hours (0-5 session length scale).

This schema allows for analysis of how gaming habits (frequency, duration) affect grades while controlling for factors like parental education and school differences.

## Key Features

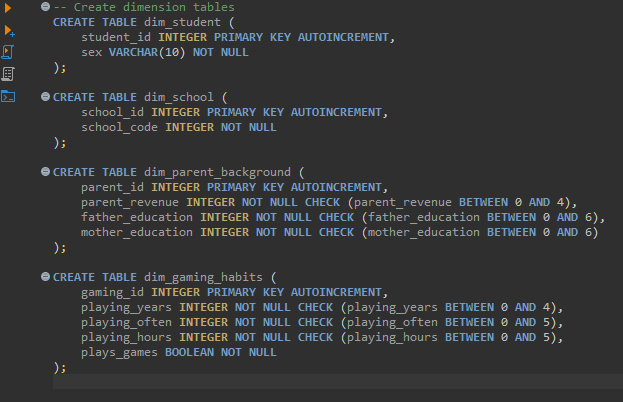
* **Star schema** for faster queries.
* **Constraints**: Ensures data validity (e.g., parental education between 0-10).
* **Indexes**: Improves query performance on joins.

During query execution, an unresolved error was identified in column 365. Due to the isolated nature of this instance, a manual correction was implemented. However, upon review, it is acknowledged that this issue should have been addressed during the data cleansing phase, particularly considering developing proficiencies in SQL.

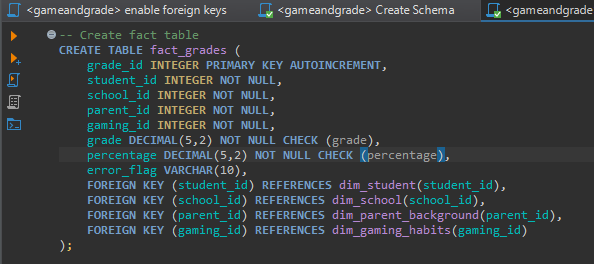
SQLite Steps:

Before creating the schema, enabling foreign key enforcement in SQLite is necessary as it's disabled by default: PRAGMA foreign\_keys = ON;

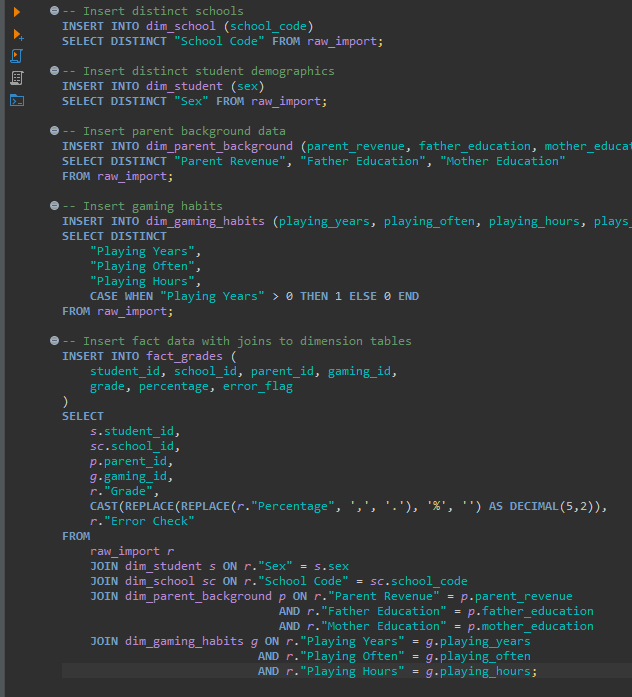
## Create Schema:



Create fact\_table:



Load and transform data:



## Example Use Case

A query could compare grades between frequent gamers (playing\_often ≥ 4) and non-gamers (playing\_years = 0) while filtering by parental education level.

