**STUDENT PERFORMANCE ANALYSIS REPORT**

**INTRODUCTION:**

This report presents a detailed analysis of student performance data using Power BI, focusing on various relationships, hierarchical structures, and key views to extract actionable insights. The aim is to assist educators and administrators in identifying factors influencing academic outcomes, student engagement, and well-being.

**DATA RELATIONSHIPS IN POWER BI:**

**One-to-One Relationship:**

**Description:** Each record in Table A is directly related to one record in Table B, creating a one-to-one match in Power BI.

**Example:** *Student\_Demographics[Student\_ID] and Academic\_Info[Student\_ID]*, *Academic\_Info[Student\_ID] and Personal\_Factors[Student\_ID]*.

**One-to-Many Relationship:**

**Description**: A single record in Table A can link to multiple records in Table B, ideal for modeling relationships where one student has multiple scores.

**Example:** *Student\_Demographics[Student\_ID] to Academic\_Info[Exam\_Score]*.

**Many-to-One Relationship:**

**Description**: Many records in Table A relate to a single record in Table B, useful for understanding aggregated student data.

**Example:** *Academic\_Info[Exam\_Score] to Personal\_Factors[Student\_ID]*.

**Many-to-Many Relationship:**

**Description**: Records in both tables can relate to multiple records across each other, managed in Power BI with bridge or join tables.

**Hierarchies:**

Hierarchical structures in Power BI allow for multi-level data analysis, providing drill-down capabilities from broader to more granular insights. For this analysis, the following tables are used:

**Academic\_Info**:

|  |
| --- |
| Performance |
| attendance |
| hours studied |
| previous scores |
| exam score |

**Personal\_Factors**:

|  |
| --- |
| Internet access |
| family income |
| learning disabilities |
| motivational level |
| peer influence |
| parental involvement |

**School Impact Table**:

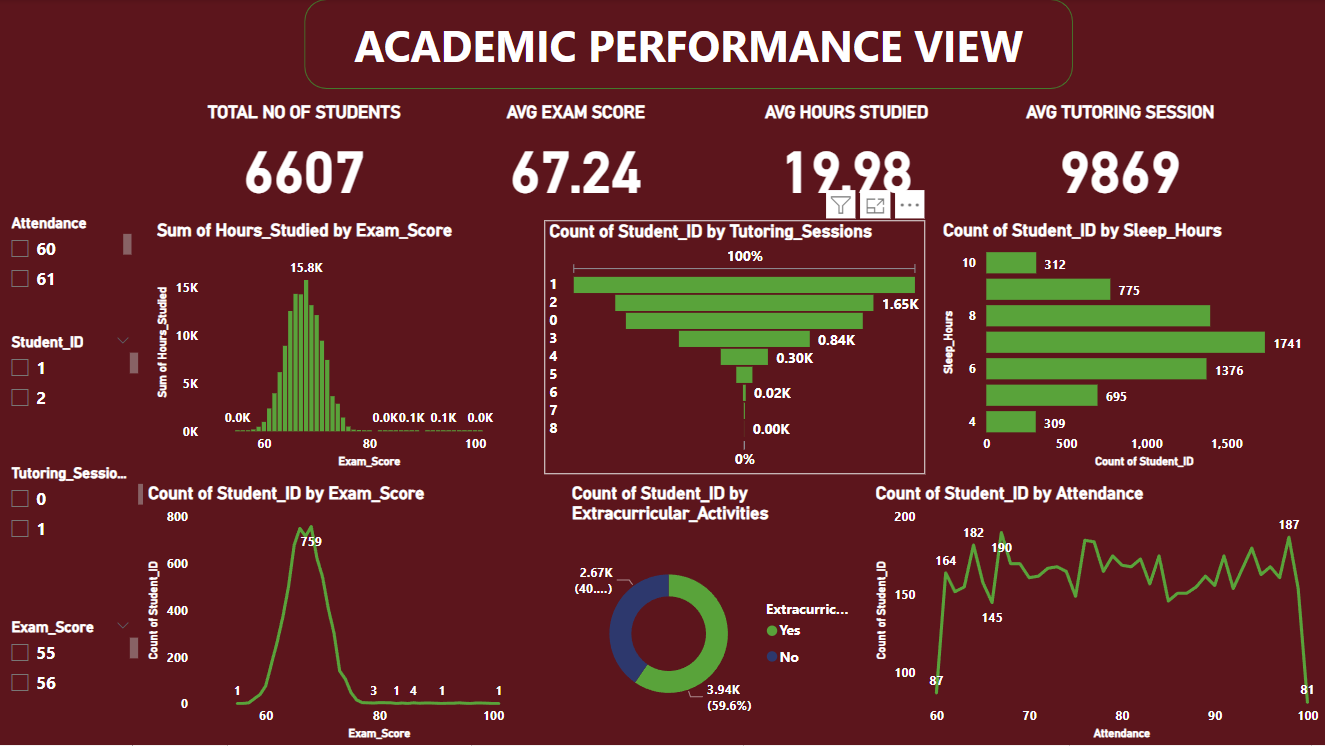
|  |
| --- |
| School impact |
| student ID |
| teacher quality |
| School type |

### Dashboard Views

#### 1. Academic Performance View

The "Academic Performance View" dashboard displays core metrics that influence student outcomes, such as:

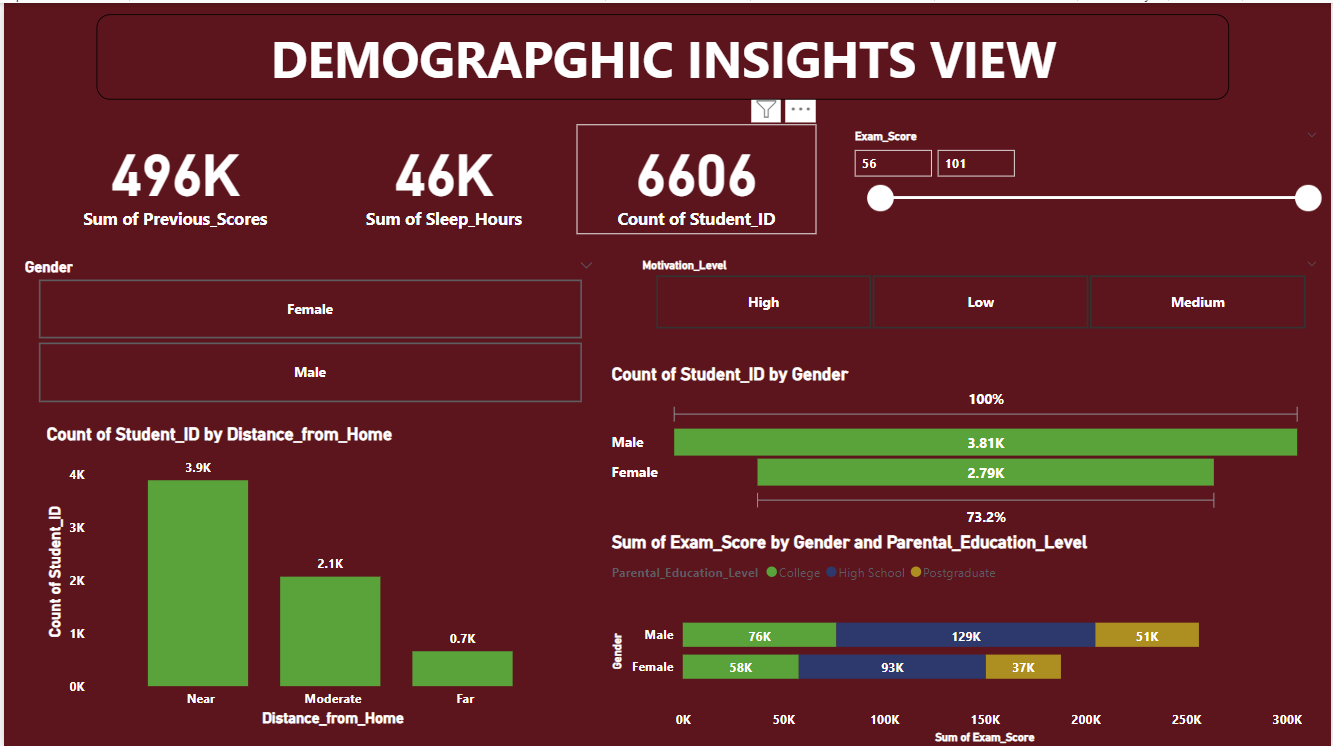
* **Metrics**: Total students, average exam scores, hours studied, tutoring sessions.
* **Visuals**: Study hours vs. exam scores, tutoring attendance, sleep hours distribution, extracurricular participation.
* **Objective**: To uncover patterns in academic engagement and performance factors



#### 2. Demographic Insights View

The "Demographic Insights View" focuses on student demographics and related academic performance influences:

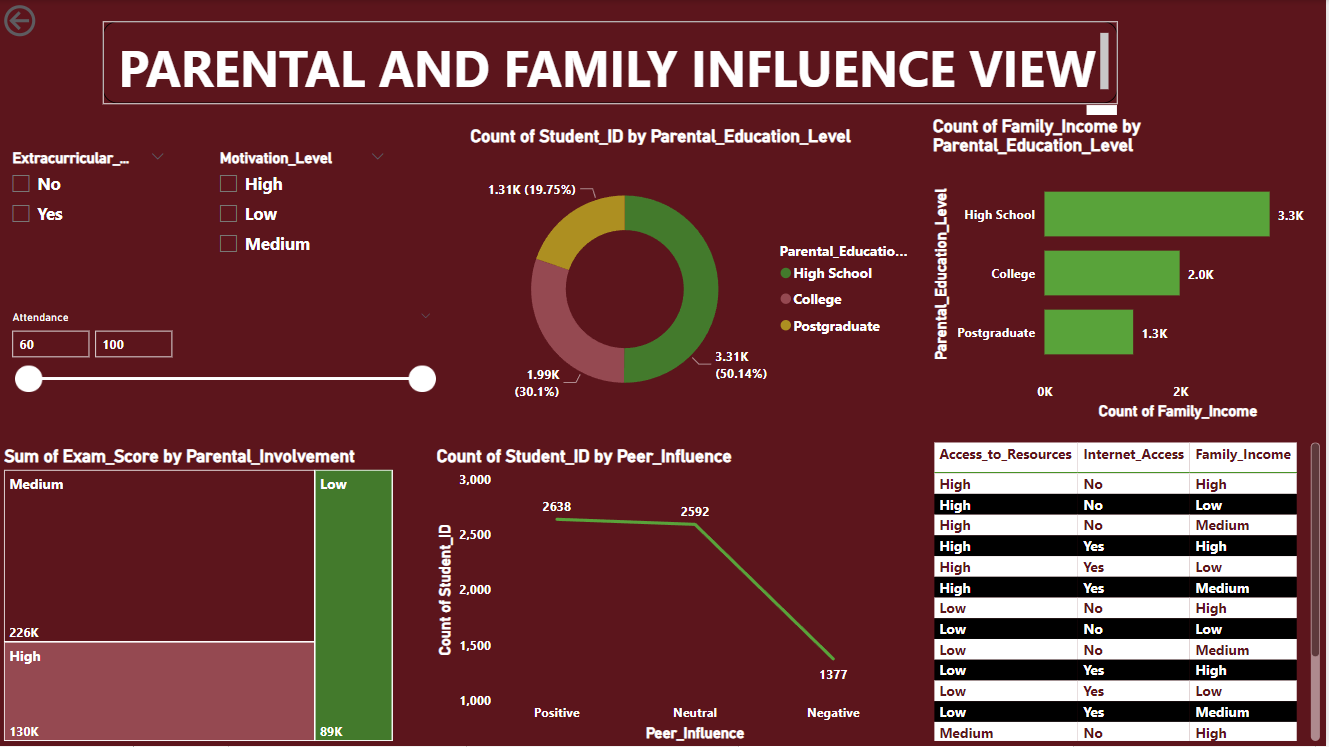
* **Metrics**: Total previous scores (496K), sleep hours (46K), total students (6607).
* **Visuals**: Gender distribution, motivation levels, distance from home, parental education, and gender-based exam scores.
* **Objective**: To understand demographic factors impacting academic success



#### 3. Parental and Family Influence View

This dashboard, titled "Parental and Family Influence View," analyzes family dynamics on academic outcomes:

* **Key Filters**: Extracurricular participation, motivation level, attendance.
* **Visuals**: Parental involvement, family income by education level, peer influence, internet access, and exam scores.
* **Objective**: To assess how family conditions affect student engagement and performance.



#### 4. Health and Motivational Level

The "Health and Motivational Level" dashboard examines well-being and motivation:

* **Metrics**: Total students (6607), sleep hours by exam score, motivation level.
* **Visuals**: Physical activity participation, sleep hours distribution, school type, teacher quality, and peer influence.
* **Objective**: To correlate health, motivation, and academic outcomes.

### 

**ADVANCED DAX FUNCTION:**

**SUMX:**

The weight is based on factors like Attendance or Motivation\_Level. We can use SUMX to iterate over a table and apply a custom calculation for each row.

WeightedExamScore = SUMX( 'Academic\_Info', 'Academic\_Info'[Exam\_Score] \* ( 1 + CALCULATE( SUM('Personal\_Factors'[Motivation\_Level]) / 100 ) ) )

**SUMX**: Iterates over each row in the Academic\_Info table.

**AVERAGEX:**

**AVERAGEX**: Similar to SUMX, but instead of summing, it calculates the average of an expression for each row in a table.

AverageScore =

AVERAGEX(

'Academic\_Info',

'Academic\_Info'[Exam\_Score] \* (1 + 'Academic\_Info'[Attendance] / 100)

)

This calculates the average of Exam\_Score, adjusting each score based on Attendance percentage as a weight.

**FILTER:**

**FILTER**: Works in conjunction with CALCULATE or X functions (like SUMX) to dynamically filter rows based on specific criteria.

FilteredExamScoreSum =

SUMX(

FILTER(

'Academic\_Info',

'Academic\_Info'[Previous\_Scores] >= 70

),

'Academic\_Info'[Exam\_Score]

)

This calculates the sum of Exam\_Score only for students who had Previous\_Scores of 70 or higher.

**RELATED:**

**RELATED**: Useful in scenarios with relationships between tables, allowing you to bring in columns from related tables.

TotalScoreWithIncomeFactor =

SUMX(

'Academic\_Info',

'Academic\_Info'[Exam\_Score] \*

RELATED('Personal\_Factors'[Family\_Income])

)

This calculates the sum of Exam\_Score, adjusting each score by the student’s Family\_Income factor from a related table.

**SUMMARIZE:**

**SUMMARIZE**:Creates a summary table for grouping and aggregation, which can be used within other functions.

SummaryTable =

SUMMARIZE(

'Academic\_Info',

'Academic\_Info'[Student\_ID],

"Total\_Exam\_Score", SUM('Academic\_Info'[Exam\_Score])

)

This creates a summary table showing each student’s total Exam\_Score, grouped by Student\_ID.

**SWITCH:**

Whether we want to categorize students Exam\_Score into performance levels (e.g., "Excellent," "Good," "Average," "Needs Improvement"). You could use the SWITCH function to create these categories based on score ranges.

Exam\_Performance =

SWITCH(

TRUE(),

'Academic\_Info'[Exam\_Score] >= 85, "Excellent",

'Academic\_Info'[Exam\_Score] >= 70, "Good",

'Academic\_Info'[Exam\_Score] >= 50, "Average",

"Needs Improvement"

)

**RANKX:**

To rank students by their Exam\_Score in descending order (highest score receives Rank 1):

ExamScore\_Rank =

RANKX(

ALL('Academic\_Info'),

'Academic\_Info'[Exam\_Score],

,

DESC,

DENSE

)

### Conclusion

The Power BI report provides comprehensive insights into student performance, emphasizing the impact of demographic, familial, health, and motivational factors. This analysis aids in identifying areas for targeted interventions, helping educators foster a more supportive academic environment for students.