**DATAFEST HACKATHON TEAM MEGABYTES**

**Problem Statement**

**Context:**Across many African countries, there has been a long-standing concern about the declining quality of elementary and secondary education. Recent statistics from JAMB, a Nigerian pre-tertiary examination body, reveal that 76% of students who sat for the 2024 UTME (Unified Tertiary Matriculation Examination) scored less than 200 out of 400 (50%). This significant underperformance highlights a critical issue that needs urgent attention: students are not adequately prepared for these crucial external exams, particularly starting from their Senior Secondary School (SSS) years.

**Key Concerns:**

* Poor Preparation: It has been observed that inadequate preparation from SS1 (Senior Secondary 1) onwards is a major contributor to students failing external exams such as the UTME. This indicates that there’s a need to monitor and support students right from the onset of their senior secondary education to improve their readiness for these critical assessments.
* Declining Interest in School: Alongside poor academic performance, there has been a growing trend among students to devalue education, popularized by the slogan "School na scam", which reflects a sense of disenchantment with the traditional educational system. Many students have lost interest in school, affecting their motivation and performance.

**Objective:**The goal is to create a monitoring and support system for students starting from SS1 to help track their performance, identify those who need extra support, and provide timely interventions to improve their readiness for external exams. In addition, there is a need to motivate and inspire students by bringing in successful professionals to demonstrate the value of education and provide encouragement. This will help rekindle interest in learning and change the negative mindset that education is unimportant.

**Solution:**The solution will involve:

1. Performance Tracking: Implementing a system to continuously track students’ academic performance from SS1 onwards across different subjects to identify early signs of struggles and provide necessary interventions.
2. Targeted Support: Based on performance data, offering tailored support for students (e.g., additional tutoring, counseling) to help them improve in their weak areas.
3. Motivation and Inspiration: Organizing motivational talks where successful professionals in different fields address students, emphasizing the importance of education as a foundation for their future careers.

By implementing this solution, we can better prepare students for external exams, restore their faith in the educational system, and ultimately improve the overall quality of education in schools**.**

**Data Collection Plan**

The project involves collecting and integrating data from multiple schools to analyze performance, attendance, financials, and staff information. The data has been merged into a CSV file, and an additional staff dataset was created by the team to simulate a real-world school data problem. This plan outlines the methods and tools used for data collection and cleaning.

**Data Sources**

The data was sourced from three different schools, and each school provided data related to:

* **Subject Scores**: Exam results for students across different subjects.
* **Students**: Demographic information, career aspirations, etc.
* **Attendance**: Records of student attendance.
* **Financial Data**: Details on school financial transactions, categorized by expenses and revenues.
* **Staff**: Information about staff members, including personal details, roles, and qualifications (created by the team).

These datasets were merged into CSV files to simplify the integration and processing steps. The five CSV files used for the project are:

* subject\_scores\_table.csv
* student\_table.csv
* attendance\_table.csv
* finance\_table.csv
* staff\_table.csv

**Data Collection Tools**

The data was collected, cleaned, and merged using the following tools and methods:

* **CSV Format**: The data from the three schools was provided in CSV format to ensure consistency and ease of access.
* **Python (Pandas)**: Python was used to read and clean the data from CSV files using Pandas. Data cleaning includes column name formatting, string casing adjustments, and validating score ranges.
* **PostgreSQL Database**: A PostgreSQL database was used to store the final processed data, ensuring easy querying and analysis. The SQLAlchemy library in Python was used to create the connection and load the cleaned datasets into the database.

**Data Preprocessing and Validation**

* **Column Cleaning**: All column names were standardized by stripping spaces, converting to lowercase, and replacing spaces with underscores to ensure consistency.
* **String Formatting**: Proper casing (title case) was applied to text columns such as student names, staff names, and career aspirations to maintain a professional and consistent look.
* **Score Validation**: Exam scores were validated to ensure that they fall within a range of 0-100.
* **Missing Data Handling**: Missing or incorrect values were either cleaned or handled using default values (e.g., missing scores replaced with zeros if needed).

**Loading Data into the Data Warehouse**

* **SQLAlchemy & PostgreSQL**: The processed data was loaded into a PostgreSQL database using SQLAlchemy. Each dataset was mapped to a corresponding table in the database (exam\_scores, students, attendance, financial, staff).
* **Verification**: After loading, queries were executed to verify that the data was successfully stored and correctly formatted in the database.

**Data Warehouse Dictionary**

The data warehouse for this project is designed to store and organize school data collected from three different schools, including scores, student demographics, attendance records, financial data, and staff information. The design ensures efficient querying, data integrity, and support for analytical models.

**Database**

* **Database Name**: school\_data\_warehouse
* **Database Type**: PostgreSQL
* **Purpose**: To centralize and store cleaned data from multiple sources (schools) in a structured manner, enabling efficient querying, reporting, and analysis.

**Schema**

* **Schema Name**: public (default schema)

**Tables**

The data is stored in five primary tables. Each table represents a distinct aspect of the school data.

1. **Table Name**: exam\_scores
   * **Description**: Stores exam scores for various subjects.
   * **Columns**:

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| student\_id | INT | Unique identifier for each student. |
| math\_score | FLOAT | Student’s score in Math (0-100). |
| history\_score | FLOAT | Student’s score in History (0-100). |
| physics\_score | FLOAT | Student’s score in Physics (0-100). |
| chemistry\_score | FLOAT | Student’s score in Chemistry (0-100). |
| biology\_score | FLOAT | Student’s score in Biology (0-100). |
| english\_score | FLOAT | Student’s score in English (0-100). |
| geography\_score | FLOAT | Student’s score in Geography (0-100). |
| school\_id | INT | Unique identifier for the school. |

* + **Primary Key**: (student\_id, school\_id)
  + **Foreign Key**: student\_id references students.student\_id.

1. **Table Name**: students
   * **Description**: Stores demographic information of students.
   * **Columns**:

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| student\_id | INT | Unique identifier for each student. |
| first\_name | VARCHAR | Student’s first name. |
| last\_name | VARCHAR | Student’s last name. |
| gender | VARCHAR | Student’s gender (e.g., Male, Female). |
| date\_of\_birth | DATE | Student’s date of birth. |
| career\_aspiration | VARCHAR | Student’s career goal. |
| school\_id | INT | Unique identifier for the school. |

* + **Primary Key**: student\_id
  + **Foreign Key**: school\_id references schools.school\_id.

1. **Table Name**: attendance
   * **Description**: Stores attendance records for students.
   * **Columns**:

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| attendance\_id | SERIAL | Unique identifier for each attendance record. |
| student\_id | INT | Unique identifier for each student. |
| date | DATE | Date of the attendance record. |
| present | BOOLEAN | Indicates if the student was present. |
| school\_id | INT | Unique identifier for the school. |

* + **Primary Key**: attendance\_id
  + **Foreign Key**: student\_id references students.student\_id.

1. **Table Name**: financial
   * **Description**: Stores financial data such as transactions and expenses.
   * **Columns**:

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| transaction\_id | SERIAL | Unique identifier for each transaction. |
| date | DATE | Date of the transaction. |
| amount | FLOAT | The amount of money involved in the transaction. |
| category | VARCHAR | Category of the transaction (e.g., Expense, Revenue). |
| description | TEXT | Description of the transaction. |
| remark | TEXT | Additional notes or remarks. |
| school\_id | INT | Unique identifier for the school. |

* + **Primary Key**: transaction\_id
  + **Foreign Key**: school\_id references schools.school\_id.

1. **Table Name**: staff
   * **Description**: Stores information about school staff, including their roles, subjects taught, and qualifications.
   * **Columns**:

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| staff\_id | SERIAL | Unique identifier for each staff member. |
| first\_name | VARCHAR | Staff member's first name. |
| last\_name | VARCHAR | Staff member's last name. |
| role | VARCHAR | Staff member's role (e.g., Teacher, Admin). |
| subject | VARCHAR | Subject the staff member teaches. |
| gender | VARCHAR | Gender of the staff member. |
| marital\_status | VARCHAR | Marital status of the staff member. |
| level\_of\_education | VARCHAR | Highest educational qualification. |
| school\_id | INT | Unique identifier for the school. |

* + **Primary Key**: staff\_id
  + **Foreign Key**: school\_id references schools.school\_id.

**Relationships**

The design ensures clear relationships between students, their scores, attendance, and the schools they belong to. The relationships between tables are defined as follows:

* **One-to-Many**:
  + Each student can have multiple exam scores (relationship between students and exam\_scores).
  + Each student can have multiple attendance records (relationship between students and attendance).
  + Each school can have multiple students, financial transactions, and staff members.
* **Foreign Keys**:
  + student\_id is a foreign key in exam\_scores and attendance, linking back to the students table.
  + school\_id is a foreign key in all tables, linking back to a potential schools table (not explicitly provided but implied).

**Data Types**

The data types for the columns are chosen based on the nature of the data:

* **INT**: For identifiers like student\_id, staff\_id, and school\_id.
* **VARCHAR**: For textual fields like names, roles, and descriptions.
* **FLOAT**: For numerical values like exam scores and financial transaction amounts.
* **BOOLEAN**: For logical fields like present (attendance).
* **DATE**: For date-related columns like date\_of\_birth, attendance date, and transaction date.
* **SERIAL**: Auto-incrementing identifier for unique keys like transaction\_id and attendance\_id.

**Indexing**

To improve performance, the following indexes are applied:

* **Primary keys**: Indexed by default for efficient retrieval of individual rows.
* **Foreign keys**: Indexed for efficient join operations between related tables (e.g., student\_id, school\_id).

**Example Query**

To retrieve all students who scored above 85 in Math and had perfect attendance, the query might look like this:

SELECT s.first\_name, s.last\_name, sc.math\_score, a.date

FROM students s

JOIN exam\_scores sc ON s.student\_id = sc.student\_id

JOIN attendance a ON s.student\_id = a.student\_id

WHERE sc.math\_score > 85

AND a.present = TRUE

GROUP BY s.student\_id, sc.math\_score, a.date;