

ANALYSIS OF UNIVERSITY STUDENT RECORDS

Introduction

In the education sector, analyzing student records is crucial for understanding academic performance, attendance patterns, and course effectiveness. This data analysis focuses on extracting valuable insights from university student records, including grades, courses, and attendance data.

By leveraging SQL queries, this analysis aims to identify top-performing students, examine the impact of attendance on grades, and evaluate trends in course performance. These insights can help educators and administrators make data-driven decisions to enhance academic outcomes.

Objectives

1. Analyze Student Performance Trends

- Identify high-performing students based on GPA or average grades.
- Compare student performance across different majors and courses.

2. Evaluate the Impact of Attendance on Academic Success:

- Determine if students with higher attendance rates tend to perform better academically.
- Identify students at risk due to low attendance percentages.

3. Assess Course Difficulty Levels:

- Find courses with the highest and lowest average grades.
- Identify courses where students struggle the most.

4. Generate Data-Driven Insights Using SQL Queries:

- Rank students within each major based on academic performance.
- Detect trends in course enrollments and grade distributions.

5. Support Decision-Making for Academic Improvement:

- Help educators identify students who need academic support.
- Assist university administrators in curriculum evaluation based on performance trends.

TOOLS

The following tools were used for data retrieval, analysis, and documentation:

1. SQL (Structured Query Language) – Data Extraction & Analysis
2. Microsoft Excel – Data Visualization & Reporting

DATA SOURCE

The data used for this analysis was manually created by designing structured tables that simulate a university student records system. The dataset includes key academic details such as student information, course enrollments, grades, and attendance records.

To ensure accuracy and relevance, the data was structured to reflect real-world academic scenarios, enabling meaningful insights into student performance, attendance trends, and course effectiveness.

Data Collection & Structure

The dataset consists of multiple interrelated tables representing different aspects of university records:

1. **Students Table:** Stores student details (e.g., Student ID, Name, Major, Enrollment Year).
2. **Courses Table:** Contains information on available courses (e.g., Course ID, Course Name, Credits, Department).
3. **Grades Table:** Records student performance in each course (e.g., Student ID, Course ID, Grade, Semester).
4. **Attendance Table:** Tracks student participation in classes (e.g., Date, Status: Present/Absent).

Each table was designed using **SQL**, ensuring proper relationships and **foreign key constraints** for data integrity.

Data Quality & Preparation

- **No duplicate records** were included to maintain accuracy.
- **No missing values** in key fields like Student ID, Course ID, and Grades.

Data Understanding:

The data analysis process involved several key steps, starting from database creation to data querying for insights. Below is a brief outline of the approach taken:

1. Creating the Database

- Designed a relational database in **SQL** to store student records.
- Used **CREATE DATABASE StudentRecords;** to initialize the database.

```
--Creating database UNIVERSITY STUDENT RECORD
Create Database StudentRecords;
use StudentRecords;

Commands completed successfully.

Completion time: 2025-02-26T13:40:49.8102952+01:00
```

2. Creating Tables

- **Structured tables to represent key university data:**
 - StudentDetails (StudentID, FirstName, LastName, Major, EnrollmentYear).
 - CoursesDetails (CourseID, CourseName, Credits, Department).
 - GradesDetails (GradeID, StudentID, CourseID, Grade, Semester).
 - AttendanceDetails (AttendanceID, StudentID, CourseID, Date, Status).
- **Ensured relationships using primary keys and foreign keys for data integrity.**

Example Table Creation:

STUDENT TABLE

```
--Creating Students tables
Create Table StudentDetails(
StudentID Varchar(10) PRIMARY KEY,
FirstName Varchar(20),
LastName Varchar(20),
DateofBirth date,
Gender Char(10),
EnrollmentYear int,
Major Char(50)
);

Select* from StudentDetails
```

Messages

```
Commands completed successfully.

Completion time: 2025-02-26T13:46:56.9538536+01:00
```

| Results | Messages |
|-----------|----------------|
| StudentID | FirstName |
| LastName | DateofBirth |
| Gender | EnrollmentYear |
| Major | |

COURSE TABLE

```

Select* from StudentDetails

-- Creating Courses Table
Create Table CoursesDetails(
CourseID Varchar(20) PRIMARY KEY,
CourseName VArchar(100),
Credits int,
Department Varchar(50)
)

Select* from CoursesDetails

```

Messages

Commands completed successfully.

Completion time: 2025-02-26T14:27:54.5553689+01:00

| Results | Messages |
|----------|------------|
| CourseID | CourseName |
| Credits | Department |


GRADE TABLE

```

-- Creating Grades Table
Create Table GradesDetails(
GradeID Varchar(10) PRIMARY KEY,
StudentID Varchar(10),
CourseID VArchar(10),
Grade float,
Semester Varchar (20)
);
ALTER TABLE GradesDetails
ADD CONSTRAINT FK_Grades_Students
FOREIGN KEY (StudentID) REFERENCES StudentDetails(StudentID);
ALTER TABLE Grades
ADD CONSTRAINT FK_Grades_Courses
FOREIGN Key (CourseID) REFERENCES CoursesDetails(CourseID)

Select* from GradesDetails

```

 Messages


Commands completed successfully.

Completion time: 2025-02-26T14:27:54.5553689+01:00

| Results | Messages |
|----------|-----------|
| GradeID | StudentID |
| CourseID | Grade |
| Semester | |

ATTENDANCE TABLE

```
-- Creating Attendance Table
Create Table AttendanceDetails(
AttendanceID Varchar(10) PRIMARY KEY,
StudentID Varchar(10),
CourseID VArchar(10),
Date date,
Status Varchar (20)
);
Select* from AttendanceDetails
```

 Messages

Commands completed successfully.

Completion time: 2025-02-26T14:33:26.0468412+01:00

| Results | Messages |
|--------------|-----------|
| AttendanceID | StudentID |
| CourseID | Date |
| Status | |

INSERTING VALUES INTO THE TABLE

```
--ADDING DATA TO THE STUDENT TABLES
```

```
INSERT INTO StudentDetails(  
StudentID,FirstName,LastName,DateofBirth,Gender,EnrollmentYear,Major)  
VALUES  
( 'RUG001', 'JOHN', 'DOE', '2000-04-05', 'Male', 2019, 'COMPUTER SCIENCE'),  
( 'RUG002', 'DAVID', 'ESTHER', '2002-02-05', 'Female', 2021, 'MATHEMATICS'),  
( 'RUG003', 'LAWRENCE', 'MATTEW', '2001-07-05', 'Male', 2020, 'COMPUTER SCIENCE'),  
( 'RUG004', 'VINCENT', 'PEACE', '2005-05-12', 'Female', 2022, 'PHYSIS'),  
( 'RUG005', 'PAUL', 'SAMUEL', '2002-09-12', 'Male', 2019, 'CHEMISTRY'),  
( 'RUG006', 'JOHNSON', 'RUTH', '2000-05-03', 'Female', 2019, 'MATHEMATICS'),  
( 'RUG007', 'LOVE', 'ALICE', '2002-07-02', 'Female', 2018, 'COMPUTER SCIENCE'),  
( 'RUG008', 'IKECHI', 'HAPPINESS', '2005-10-08', 'Female', 2020, 'MECHANICAL ENGINEERING'),  
( 'RUG009', 'JOVITA', 'BLESSING', '2002-08-10', 'Male', 2022, 'MATHEMATICS'),  
( 'RUG010', 'ALABI', 'DAVID', '2004-01-11', 'Male', 2021, 'PHYSIS')  
  
Select* from StudentDetails
```

Messages

(10 rows affected)

Completion time: 2025-02-26T14:49:24.5669099+01:00

Results Messages

| | StudentID | FirstName | LastName | DateofBirth | Gender | EnrollmentYear | Major |
|----|-----------|-----------|-----------|-------------|--------|----------------|------------------------|
| 1 | RUG001 | JOHN | DOE | 2000-04-05 | Male | 2019 | COMPUTER SCIENCE |
| 2 | RUG002 | DAVID | ESTHER | 2002-02-05 | Female | 2021 | MATHEMATICS |
| 3 | RUG003 | LAWRENCE | MATTEW | 2001-07-05 | Male | 2020 | COMPUTER SCIENCE |
| 4 | RUG004 | VINCENT | PEACE | 2005-05-12 | Female | 2022 | PHYSIS |
| 5 | RUG005 | PAUL | SAMUEL | 2002-09-12 | Male | 2019 | CHEMISTRY |
| 6 | RUG006 | JOHNSON | RUTH | 2000-05-03 | Female | 2019 | MATHEMATICS |
| 7 | RUG007 | LOVE | ALICE | 2002-07-02 | Female | 2018 | COMPUTER SCIENCE |
| 8 | RUG008 | IKECHI | HAPPINESS | 2005-10-08 | Female | 2020 | MECHANICAL ENGINEERING |
| 9 | RUG009 | JOVITA | BLESSING | 2002-08-10 | Male | 2022 | MATHEMATICS |
| 10 | RUG010 | ALABI | DAVID | 2004-01-11 | Male | 2021 | PHYSIS |

```
-- ADDING DATA INTO COURSE TABLE
INSERT INTO CoursesDetails(
CourseID, CourseName, Credits,Department)
VALUES
('MAT101','INTRODUCTION TO MATHEMATICS', 3,'MATHEMATICS'),
('CSC102','INTRODUCTION TO COMPUTER', 4,'COMPUTER SCIENCE'),
('PHY103','INTRODUCTION TO PHYSIS', 3,'PHYSIS'),
('CHM101','INTRODUCTION TO CHEMISTRY', 2,'CHEMISTRY'),
('CSC104','DATA ANALYSIS', 5,'COMPUTER SCIENCE'),
('MAT105','QUADRATIC EQUATION', 4,'MATHEMATICS'),
('MEG101','INTRODUCTION TO ENGINEERING', 3,'MECHANICAL ENGINNERING'),
('CSC107','INTRODUCTION TO PROGRAMMING', 6,'COMPUTER SCIENCE'),
('PHY101','INTRODUCTION TO FORCE', 3,'PHYSIS'),
('CHM111','INTRODUCTION TO ORGANIC CHEMISTRY', 1,'CHEMISTRY')

Select* from CoursesDetails
```

Messages

(10 rows affected)

Completion time: 2025-02-26T14:51:40.5340981+01:00

Results Messages

| | CourseID | CourseName | Credits | Department |
|----|----------|-----------------------------------|---------|------------------------|
| 1 | CHM101 | INTRODUCTION TO CHEMISTRY | 2 | CHEMISTRY |
| 2 | CHM111 | INTRODUCTION TO ORGANIC CHEMISTRY | 1 | CHEMISTRY |
| 3 | CSC102 | INTRODUCTION TO COMPUTER | 4 | COMPUTER SCIENCE |
| 4 | CSC104 | DATA ANALYSIS | 5 | COMPUTER SCIENCE |
| 5 | CSC107 | INTRODUCTION TO PROGRAMMING | 6 | COMPUTER SCIENCE |
| 6 | MAT101 | INTRODUCTION TO MATHEMATICS | 3 | MATHEMATICS |
| 7 | MAT105 | QUADRATIC EQUATION | 4 | MATHEMATICS |
| 8 | MEG101 | INTRODUCTION TO ENGINEERING | 3 | MECHANICAL ENGINNERING |
| 9 | PHY101 | INTRODUCTION TO FORCE | 3 | PHYSIS |
| 10 | PHY103 | INTRODUCTION TO PHYSIS | 3 | PHYSIS |

```
--ADDING DATA INTO GRADE TABLE
INSERT INTO GradesDetails(
GradeID,StudentID,CourseID,Grade,Semester)
VALUES
(1,'RUG001','CSC102',3.75,'SECOND SEMESTER'),
(2,'RUG002','MAT101',3.98,'FIRST SEMESTER'),
(3,'RUG003','CSC104',4.45,'SECOND SEMESTER'),
(4,'RUG004','PHY101',3.88,'FIRST SEMESTER'),
(5,'RUG005','CHM101',3.99,'FIRST SEMESTER'),
(6,'RUG006','MAT105',4.50,'FIRST SEMESTER'),
(7,'RUG007','CSC107',3.30,'SECOND SEMESTER'),
(8,'RUG008','MEG101',3.67,'FIRST SEMESTER'),
(9,'RUG009','CHM102',3.45,'SECOND SEMESTER'),
(10,'RUG010','CHM111',4.70,'SECOND SEMESTER')

Select* from GradesDetails
```

Messages

(10 rows affected)

Completion time: 2025-02-26T14:54:13.3613105+01:00

Results

| | GradeID | StudentID | CourseID | Grade | Semester |
|----|---------|-----------|----------|-------|-----------------|
| 1 | 1 | RUG001 | CSC102 | 3.75 | SECOND SEMESTER |
| 2 | 10 | RUG010 | CHM111 | 4.7 | SECOND SEMESTER |
| 3 | 2 | RUG002 | MAT101 | 3.98 | FIRST SEMESTER |
| 4 | 3 | RUG003 | CSC104 | 4.45 | SECOND SEMESTER |
| 5 | 4 | RUG004 | PHY101 | 3.88 | FIRST SEMESTER |
| 6 | 5 | RUG005 | CHM101 | 3.99 | FIRST SEMESTER |
| 7 | 6 | RUG006 | MAT105 | 4.5 | FIRST SEMESTER |
| 8 | 7 | RUG007 | CSC107 | 3.3 | SECOND SEMESTER |
| 9 | 8 | RUG008 | MEG101 | 3.67 | FIRST SEMESTER |
| 10 | 9 | RUG009 | CHM102 | 3.45 | SECOND SEMESTER |


```
--ADDING DATA INTO ATTENDANCE COLUMN
INSERT INTO AttendanceDetails(
AttendanceID,StudentID,CourseID,Date,Status)
VALUES
(1,'RUG001','MAT101','2023-09-01','PRESENT'),
(2,'RUG002','CSC101','2022-08-10','ABSENT'),
(3,'RUG003','PHY101','2024-05-12','PRESENT'),
(4,'RUG004','CHM101','2023-04-11','ABSENT'),
(5,'RUG005','MAT105','2024-01-01','LATE'),
(6,'RUG006','CSC104','2025-02-01','PRESENT'),
(7,'RUG007','MEG101','2024-03-12','ABSENT'),
(8,'RUG008','CSC107','2022-04-01','PRESENT'),
(9,'RUG009','GEO101','2024-09-06','LATE'),
(10,'RUG010','CHM102','2023-11-01','ABSENT')

Select* from AttendanceDetails
```

Messages

(10 rows affected)

Completion time: 2025-02-26T14:56:26.6454077+01:00

Results

Messages

| | AttendanceID | StudentID | CourseID | Date | Status |
|----|--------------|-----------|----------|------------|---------|
| 1 | 1 | RUG001 | MAT101 | 2023-09-01 | PRESENT |
| 2 | 10 | RUG010 | CHM102 | 2023-11-01 | ABSENT |
| 3 | 2 | RUG002 | CSC101 | 2022-08-10 | ABSENT |
| 4 | 3 | RUG003 | PHY101 | 2024-05-12 | PRESENT |
| 5 | 4 | RUG004 | CHM101 | 2023-04-11 | ABSENT |
| 6 | 5 | RUG005 | MAT105 | 2024-01-01 | LATE |
| 7 | 6 | RUG006 | CSC104 | 2025-02-01 | PRESENT |
| 8 | 7 | RUG007 | MEG101 | 2024-03-12 | ABSENT |
| 9 | 8 | RUG008 | CSC107 | 2022-04-01 | PRESENT |
| 10 | 9 | RUG009 | GEO101 | 2024-09-06 | LATE |

Analysis & Query Writing

The following are questions that will guide my analysis and how I use SQL to extract insights from my data:

1. List of students with GPA (Grade Point Average) above 3.5

```
--1 list of student with their GPA(Grade Point Average)
SELECT S.StudentID, S.FirstName, S.LastName,AVG(G.Grade) AS GPA
FROM StudentDetails AS S
JOIN GradesDetails AS G
ON S.StudentID=G.StudentID
GROUP BY S.StudentID, S.FirstName, S.LastName
HAVING AVG(G.Grade)>=3.5
ORDER BY GPA DESC
```

| | StudentID | FirstName | LastName | GPA |
|---|-----------|-----------|-----------|------|
| 1 | RUG010 | ALABI | DAVID | 4.7 |
| 2 | RUG006 | JOHNSON | RUTH | 4.5 |
| 3 | RUG003 | LAWRENCE | MATTEW | 4.45 |
| 4 | RUG005 | PAUL | SAMUEL | 3.99 |
| 5 | RUG002 | DAVID | ESTHER | 3.98 |
| 6 | RUG004 | VINCENT | PEACE | 3.88 |
| 7 | RUG001 | JOHN | DOE | 3.75 |
| 8 | RUG008 | IKECHI | HAPPINESS | 3.67 |

8 students listed have a GPA greater than 3.5, meaning they are performing above average in their academics. ALABI DAVID (GPA 4.7) is the highest-performing student

2. What is the Attendance Percentage for each Student

```
--2 ATTENDANCE PERCENTAGE FOR EACH STUDENT
SELECT S.StudentID,S.FirstName,S.LastName,ROUND(SUM(CASE WHEN A.Status='PRESENT' THEN 1 ELSE 0 END)*100/COUNT(*),2)
AS ATTENDANCEPERCENTAGE
FROM StudentDetails AS S
JOIN AttendanceDetails AS A
ON S.StudentID=A.StudentID
GROUP BY S.StudentID,S.FirstName,S.LastName
```

| | StudentID | FirstName | LastName | ATTENDANCEPERCENTAGE |
|----|-----------|-----------|-----------|----------------------|
| 1 | RUG001 | JOHN | DOE | 100 |
| 2 | RUG002 | DAVID | ESTHER | 0 |
| 3 | RUG003 | LAWRENCE | MATTEW | 100 |
| 4 | RUG004 | VINCENT | PEACE | 0 |
| 5 | RUG005 | PAUL | SAMUEL | 0 |
| 6 | RUG006 | JOHNSON | RUTH | 100 |
| 7 | RUG007 | LOVE | ALICE | 0 |
| 8 | RUG008 | IKECHI | HAPPINESS | 100 |
| 9 | RUG009 | JOVITA | BLESSING | 0 |
| 10 | RUG010 | ALABI | DAVID | 0 |

Students with Perfect Attendance (100%) are JOHN DOE (RUG001), LAWRENCE MATTEW (RUG003),JOHNSON RUTH (RUG006), IKECHI HAPPINESS (RUG008).These students have attended all their scheduled classes, which may correlate with better academic performance.

Students with 0% Attendance are DAVID ESTHER (RUG002), VINCENT PEACE (RUG004), PAUL SAMUEL (RUG005), LOVE ALICE (RUG007), JOVITA BLESSING (RUG009), ALABI DAVID (RUG010). These students have not attended any classes, which may indicate issues such as lack of engagement, absenteeism, or registration issues.

3. What is the most attended course

```
--3 MOST ATTENDED COURSE
SELECT C.CourseName, COUNT(*) AS ATTENDANCECOUNT
FROM AttendanceDetails AS A
JOIN CoursesDetails AS C
ON C.CourseID=A.CourseID
WHERE A.Status='PRESENT'
GROUP BY C.CourseName
ORDER BY ATTENDANCECOUNT
```

| | CourseName | ATTENDANCECOUNT |
|---|-----------------------------|-----------------|
| 1 | DATA ANALYSIS | 1 |
| 2 | INTRODUCTION TO FORCE | 1 |
| 3 | INTRODUCTION TO MATHEMATICS | 1 |
| 4 | INTRODUCTION TO PROGRAMMING | 1 |

No single course has higher attendance than the others, all courses have only 1 recorded attendance. Low overall attendance may indicate lack of student engagement or incomplete data

recording. Since the attendance count is the same across courses, there is no "most attended" course based on this dataset.

4. What is the average grade for each student

```
--4 AVERAGE GRADE FOR EACH STUDENT
SELECT S.FirstName,S.LastName,C.CourseName, AVG(G.Grade) AS AVERAGEGRADE
FROM GradesDetails AS G
JOIN CoursesDetails AS C
ON C.CourseID=G.CourseID
JOIN StudentDetails AS S
ON G.StudentID=S.StudentID
GROUP BY S.FirstName,S.LastName,C.CourseName
ORDER BY AVERAGEGRADE DESC
```

| | FirstName | LastName | CourseName | AVERAGEGRADE |
|---|-----------|-----------|-----------------------------------|--------------|
| 1 | ALABI | DAVID | INTRODUCTION TO ORGANIC CHEMISTRY | 4.7 |
| 2 | JOHNSON | RUTH | QUADRATIC EQUATION | 4.5 |
| 3 | LAWRENCE | MATTEW | DATA ANALYSIS | 4.45 |
| 4 | PAUL | SAMUEL | INTRODUCTION TO CHEMISTRY | 3.99 |
| 5 | DAVID | ESTHER | INTRODUCTION TO MATHEMATICS | 3.98 |
| 6 | VINCENT | PEACE | INTRODUCTION TO FORCE | 3.88 |
| 7 | JOHN | DOE | INTRODUCTION TO COMPUTER | 3.75 |
| 8 | IKECHI | HAPPINESS | INTRODUCTION TO ENGINEERING | 3.67 |
| 9 | LOVE | ALICE | INTRODUCTION TO PROGRAMMING | 3.3 |

ALABI DAVID has the highest average grade (4.7) in Introduction to Organic Chemistry. JOHNSON RUTH (4.5) and LAWRENCE MATTEW (4.45) also performed exceptionally well in their respective courses. Most students scored above 3.5, indicating a strong academic performance. LOVE ALICE has the lowest average grade (3.3) in Introduction to Programming.

5. Find the average GPA by Major

```
--5 FIND AVERAGE GPA BY MAJOR
SELECT S.Major, ROUND(AVG(G.Grade),2) AS AVERAGEGPA
FROM StudentDetails AS S
JOIN GradesDetails AS G
ON S.StudentID=G.StudentID
GROUP BY S.Major
ORDER BY AVERAGEGPA DESC
```

| Results Messages | | |
|------------------|------------------------|------------|
| | Major | AVERAGEGPA |
| 1 | PHYSIS | 4.29 |
| 2 | CHEMISTRY | 3.99 |
| 3 | MATHEMATICS | 3.98 |
| 4 | COMPUTER SCIENCE | 3.83 |
| 5 | MECHANICAL ENGINEERING | 3.67 |

Physics has the highest average GPA of 4.29, indicating strong student performance in this field. Chemistry (3.99) and Mathematics (3.98) follow closely behind, showing consistent excellence in analytical and problem-solving disciplines. Computer Science (3.83) and Mechanical Engineering (3.67) have relatively lower GPAs, possibly due to the complexity of coursework or practical nature of these fields. All majors have an average GPA above 3.5, demonstrating a generally high level of academic achievement.

6. Identify top performing student in each Major

```
--6 IDENTIFY TOP PERFORMING STUDENT IN EACH MAJOR
SELECT S.Major,S.FirstName,S.LastName, AVG(G.Grade) AS GPA
FROM StudentDetails AS S
JOIN GradesDetails AS G
ON S. StudentID=G.StudentID
GROUP BY S.Major,S.FirstName,S.LastName
HAVING AVG(G.Grade) >=3.5
ORDER BY S.Major, GPA DESC
```

| Results Messages | | | | |
|------------------|------------------------|-----------|-----------|------|
| | Major | FirstName | LastName | GPA |
| 1 | CHEMISTRY | PAUL | SAMUEL | 3.99 |
| 2 | COMPUTER SCIENCE | LAWRENCE | MATTEW | 4.45 |
| 3 | COMPUTER SCIENCE | JOHN | DOE | 3.75 |
| 4 | MATHEMATICS | JOHNSON | RUTH | 4.5 |
| 5 | MATHEMATICS | DAVID | ESTHER | 3.98 |
| 6 | MECHANICAL ENGINEERING | IKECHI | HAPPINESS | 3.67 |
| 7 | PHYSIS | ALABI | DAVID | 4.7 |
| 8 | PHYSIS | VINCENT | PEACE | 3.88 |

ALABI DAVID (Physics) holds the highest GPA of 4.7, showcasing exceptional academic performance. Mathematics & Computer Science students excel, with JOHNSON RUTH (4.5) and LAWRENCE MATTEW (4.45) being top achievers in their fields.

The highest GPA in Mechanical Engineering (3.67) and Chemistry (3.99) is lower compared to other disciplines, possibly indicating challenging coursework or grading systems.

Physics, Mathematics, and Computer Science students have the highest GPAs, suggesting effective teaching methods, student dedication, or interest in these fields.

CONCLUSION

The analysis of university student records provided valuable insights into academic performance, attendance trends, and course engagement. By evaluating key metrics such as GPA, attendance percentage, course participation, and major-wise performance, the following conclusions can be drawn:

1. Top Performers Identified:

Certain students consistently achieve high GPAs, with ALABI DAVID (Physics) recording the highest GPA (4.7). Recognizing and rewarding these top performers can serve as motivation for others.

2. Major-Wise Academic Strength:

Physics, Mathematics, and Computer Science students tend to excel academically, indicating strong performance in science and technology disciplines.

3. Attendance Disparities Observed:

Several students exhibit low attendance percentages, potentially impacting their overall academic performance. This highlights the need for interventions such as attendance tracking, student engagement programs, and academic support initiatives.

4. Most Attended Courses:

Course participation is generally evenly distributed, with no single course significantly dominating in attendance. However, further analysis could help determine if certain courses need better student engagement strategies.

5. GPA Variation Across Majors:

While some majors like Physics (Avg GPA: 4.29) and Chemistry (Avg GPA: 3.99) perform well, others like Mechanical Engineering (Avg GPA: 3.67) have relatively lower averages, suggesting areas for academic improvement.

6. Performance Discrepancies Among Students:

Some students maintain high GPAs despite low attendance, while others with high attendance do not necessarily have the best grades. This suggests that academic success is influenced by factors beyond attendance alone, such as study habits, teaching quality, and individual aptitude.