**A group of logos on a green background

Description automatically generated**

**DS6504 Business Intelligence and Big Data**

**Assignment 2: Student Attendance Dashboard**

**Tutor: Dr. Abubakar Siddique**

**Student: Adya Sinha**

**Student ID: 2230449**

**Date: 20 October 2024**

Table of Contents

[a. Executive Summary 3](#_Toc180354659)

[b. Entity Relationship Diagram 4](#_Toc180354660)

[c. SQL Views 5](#_Toc180354661)

[d. Stored Procedures 10](#_Toc180354662)

[e. Dashboard Screenshots 13](#_Toc180354663)

[f. User Guide for Student Attendance Dashboard 16](#_Toc180354664)

[Conclusion 17](#_Toc180354665)

# Executive Summary

This report presents the development and implementation of a Student Attendance Dashboard which is designed to monitor student attendance and analyse it effectively. This dashboard leverages SQL server and Microsoft Visual Studio SSRS to help provide comprehensive insights into the individual student attendance metric and their classes including the warning lists for at-risk students.

There are many key findings from the dashboard which include features like columns which display average attendance rates for lectures, labs, and overall class participation. These metrics enable Dr. Diana to quickly assess the engagement levels of her students. The user can also filter the attendance data by session ID and student ID which then provides them with detailed information on each of the students. The dashboard also automatically generates the lists of students whose attendance fall below a specified threshold. This further helps Dr. Diana identify the children who ned some special or additional support.

The aim of this student attendance dashboard is to enhance the ability of educators to support students proactively by providing actionable data and also help reduce manual workload.

# Entity Relationship Diagram

Given below is the Entity Relationship Diagram for the SQL Script: -

A screenshot of a computer

Description automatically generated

The ‘Attendance’ and ‘ColumnList’ tables are temporary and used for data import. They don't have direct relationships with the main tables. ‘StudentAttendance’ has a composite primary key (Student\_ID, S\_ID) and serves as the junction table between ‘Student’ and ‘ClassSession’. The foreign key relationships in ‘StudentAttendance’ are commented out in the original SQL but should be implemented for data integrity.

These are the relationships:

1. Student (1) --- (0..N) StudentAttendance
2. ClassSession (1) --- (0..N) StudentAttendance

# SQL Views

1. **vw\_ClassLectureAttendanceRate**
2. Purpose: This view calculates the attendance rate for class lectures which is done by counting the total sessions and attended sessions, which then provides the percentage of attendance.
3. Database Diagram:

Tables Used: ClassSession, StudentAttendance

Attributes: ClassSession(S\_ID, S\_Name, S\_Type), StudentAttendance(S\_ID, SS\_Attendance)

1. Keys Used :

Primary Key: ClassSession(S\_ID

Foreign Keys: StudentAttendance(S\_ID)

1. SQL Code:

CREATE VIEW vw\_ClassLectureAttendanceRate AS

SELECT

cs.S\_Name,

COUNT(sa.S\_ID) AS TotalSessions,

SUM(CASE WHEN sa.SS\_Attendance = '100.00%' THEN 1 ELSE 0 END) AS AttendedSessions,

CAST(SUM(CASE WHEN sa.SS\_Attendance = '100.00%' THEN 1 ELSE 0 END) AS FLOAT) / NULLIF(COUNT(sa.S\_ID), 0) \* 100 AS AttendancePercentage

FROM

ClassSession cs

LEFT JOIN StudentAttendance sa ON cs.S\_ID = sa.S\_ID

WHERE

cs.S\_Type = 0 -- Lecture

GROUP BY

cs.S\_Name;

1. **vw\_ClassLabAttendanceRate**
2. Purpose: This view calculates the attendance rate for all the lab sessions in a similar way to the one in lecture attendance rate.
3. Database Diagram:

Tables Used: ClassSession, StudentAttendance

Attributes: ClassSession(S\_ID, S\_Name, S\_Type), StudentAttendance(S\_ID, SS\_Attendance)

1. Keys Used :

Primary Keys: ClassSession(S\_ID)

Foreign Keys: StudentAttendance(S\_ID)

1. SQL Code:

CREATE VIEW vw\_ClassLabAttendanceRate AS

SELECT

cs.S\_Name,

COUNT(sa.S\_ID) AS TotalSessions,

SUM(CASE WHEN sa.SS\_Attendance = '100.00%' THEN 1 ELSE 0 END) AS AttendedSessions,

CAST(SUM(CASE WHEN sa.SS\_Attendance = '100.00%' THEN 1 ELSE 0 END) AS FLOAT) / NULLIF(COUNT(sa.S\_ID), 0) \* 100 AS AttendancePercentage

FROM

ClassSession cs

LEFT JOIN StudentAttendance sa ON cs.S\_ID = sa.S\_ID

WHERE

cs.S\_Type = 1 -- Lab

GROUP BY

cs.S\_Name;

1. **vw\_ClassAverageLectureAttendanceRate**
   1. Purpose: This view calculates the average attendance rate for all the lecture classes.
   2. Database Diagram:

Tables Used: vw\_ClassLectureAttendanceRate

Attributes: vw\_ClassLectureAttendanceRate(AttendancePercentage)

* 1. Keys Used: No keys are used as it is an aggregate view.
  2. SQL Code:

CREATE VIEW vw\_ClassAverageLectureAttendanceRate AS

SELECT

AVG(AttendancePercentage) AS AverageLectureAttendanceRate

FROM

vw\_ClassLectureAttendanceRate;

1. **vw\_ClassAverageLabAttendanceRate**
2. Purpose: This view calculates the average attendance rate for all the lab classes.
3. Database Diagram:

Tables Used: vw\_ClassLabAttendanceRate

Attributes: vw\_ClassLectureAttendanceRate(AttendancePercentage)

1. Keys Used: No keys are used as it is an aggregate view.
2. SQL Code:

CREATE OR ALTER PROCEDURE sp\_GetClassAverageLabAttendanceRate

AS

BEGIN

SELECT \* FROM vw\_ClassAverageLabAttendanceRate;

END

GO

1. **vw\_ClassAverageAttendanceRate**

i. Purpose: This view calculates the overall average attendance rate by combining both the lecture and lab attendance rates.

ii. Database Diagram:

Tables Used: vw\_ClassLectureAttendanceRate, vw\_ClassLabAttendanceRate

Attributes: vw\_ClassLectureAttendanceRate(AttendancePercentage), vw\_ClassLabAttendanceRate(AttendancePercentage).

iii. Keys Used: No keys are used as it is an aggregate view.

iv. SQL Code:

CREATE VIEW vw\_ClassAverageAttendanceRate AS

SELECT

AVG(AttendancePercentage) AS AverageAttendanceRate

FROM

(SELECT AttendancePercentage FROM vw\_ClassLectureAttendanceRate

UNION ALL

SELECT AttendancePercentage FROM vw\_ClassLabAttendanceRate) AS SubQuery;

1. **vw\_LectureWarningList**

i. Purpose: This view shows all the students whose lecture attendance rate is lower than 70%.

ii. Database Diagram:

Tables Used: Student, StudentAttendance, ClassSession

Attributes: Student(Student\_ID, Student\_Name), StudentAttendance(Student\_ID, S\_ID, SS\_Attendance), ClassSession(S\_ID, S\_Type)

iii. Keys Used:

Primary Key**:** Student(Student\_ID), ClassSession(S\_ID)

Foreign Key**:** StudentAttendance(Student\_ID, S\_ID)

iv. SQL Code:

CREATE VIEW vw\_LectureWarningList AS

SELECT

s.Student\_ID,

s.Student\_Name,

SUM(CASE WHEN sa.SS\_Attendance = '100.00%' THEN 1 ELSE 0 END) AS AttendedLectures,

COUNT(sa.S\_ID) AS TotalLectures,

CAST(SUM(CASE WHEN sa.SS\_Attendance = '100.00%' THEN 1 ELSE 0 END) AS FLOAT) / NULLIF(COUNT(sa.S\_ID), 0) \* 100 AS LectureAttendanceRate

FROM

Student s

JOIN StudentAttendance sa ON s.Student\_ID = sa.Student\_ID

JOIN ClassSession cs ON sa.S\_ID = cs.S\_ID

WHERE

cs.S\_Type = 0 -- Lecture

GROUP BY

s.Student\_ID, s.Student\_Name

HAVING

CAST(SUM(CASE WHEN sa.SS\_Attendance = '100.00%' THEN 1 ELSE 0 END) AS FLOAT) / NULLIF(COUNT(sa.S\_ID), 0) \* 100 < 70;

1. **vw\_LabWarningList**

i. Purpose: This view shows all the students whose lab attendance rate is lower than 70%.

ii. Database Diagram:

Tables Used: Student, StudentAttendance, ClassSession

Attributes: Student(Student\_ID, Student\_Name), StudentAttendance(Student\_ID, S\_ID, SS\_Attendance), ClassSession(S\_ID, S\_Type)

iii. Keys Used:

Primary Key**:** Student(Student\_ID), ClassSession(S\_ID)

Foreign Key**:** StudentAttendance(Student\_ID, S\_ID)

iv. SQL Code:

CREATE VIEW vw\_LabWarningList AS

SELECT

s.Student\_ID,

s.Student\_Name,

SUM(CASE WHEN sa.SS\_Attendance = '100.00%' THEN 1 ELSE 0 END) AS AttendedLabs,

COUNT(sa.S\_ID) AS TotalLabs,

CAST(SUM(CASE WHEN sa.SS\_Attendance = '100.00%' THEN 1 ELSE 0 END) AS FLOAT) / NULLIF(COUNT(sa.S\_ID), 0) \* 100 AS LabAttendanceRate

FROM

Student s

JOIN StudentAttendance sa ON s.Student\_ID = sa.Student\_ID

JOIN ClassSession cs ON sa.S\_ID = cs.S\_ID

WHERE

cs.S\_Type = 1 -- Lab

GROUP BY

s.Student\_ID, s.Student\_Name

HAVING

CAST(SUM(CASE WHEN sa.SS\_Attendance = '100.00%' THEN 1 ELSE 0 END) AS FLOAT) / NULLIF(COUNT(sa.S\_ID), 0) \* 100 < 70;

1. **vw\_OverallWarningList**

i. Purpose: This view shows all the students whose overall attendance rate (lectures and labs both) is lower than 70%.

ii. Database Diagram:

Tables Used: Student, StudentAttendance

Attributes: Student(Student\_ID, Student\_Name), StudentAttendance(Student\_ID, S\_ID, SS\_Attendance)

iii. Keys Used:

Primary Key**:** Student(Student\_ID)

Foreign Key**:** StudentAttendance(Student\_ID)

iv. SQL Code:

CREATE VIEW vw\_OverallWarningList AS

SELECT

s.Student\_ID,

s.Student\_Name,

SUM(CASE WHEN sa.SS\_Attendance = '100.00%' THEN 1 ELSE 0 END) AS AttendedSessions,

COUNT(sa.S\_ID) AS TotalSessions,

CAST(SUM(CASE WHEN sa.SS\_Attendance = '100.00%' THEN 1 ELSE 0 END) AS FLOAT) / NULLIF(COUNT(sa.S\_ID), 0) \* 100 AS OverallAttendanceRate

FROM

Student s

JOIN StudentAttendance sa ON s.Student\_ID = sa.Student\_ID

GROUP BY

s.Student\_ID, s.Student\_Name

HAVING

CAST(SUM(CASE WHEN sa.SS\_Attendance = '100.00%' THEN 1 ELSE 0 END) AS FLOAT) / NULLIF(COUNT(sa.S\_ID), 0) \* 100 < 70;

# Stored Procedures

1. **sp\_GetStudentAttendance**

i. Purpose: This stored procedure retrieves a detailed attendance information for each student, including any sessions which they attended and the type of each session.

ii. Groupings and Aggregations

* **Groupings:** Group by Student\_ID and S\_ID
* **Aggregations:** There are no aggregations, as this procedure focuses on retrieving detailed records rather than aggregating data.

iii. SQL Code

CREATE OR ALTER PROCEDURE sp\_GetStudentAttendance

@StudentID NVARCHAR(10)

AS

BEGIN

SELECT

s.Student\_ID,

s.Student\_Name,

cs.S\_Name,

cs.S\_Date,

cs.S\_Type,

sa.SS\_Attendance

FROM

Student s

JOIN StudentAttendance sa ON s.Student\_ID = sa.Student\_ID

JOIN ClassSession cs ON sa.S\_ID = cs.S\_ID

WHERE

s.Student\_ID = @StudentID

ORDER BY

cs.S\_Date;

END

GO

1. **sp\_GetSessionAttendance**

i. Purpose: his stored procedure retrieves the attendance information for any specific session, which will list all the students who attended the sessions and their attendance status.

ii. Groupings and Aggregations

* **Groupings:** Group by S\_ID
* **Aggregations:** None, as this procedure focuses on retrieving detailed records rather than aggregating data.

iii. SQL Code

CREATE PROCEDURE sp\_GetSessionAttendance

@SessionID INT

AS

BEGIN

SELECT

s.Student\_Name,

s.Student\_ID,

sa.SS\_Attendance

FROM

StudentAttendance sa

JOIN Student s ON sa.Student\_ID = s.Student\_ID

WHERE

sa.S\_ID = @SessionID

ORDER BY

s.Student\_Name;

END

GO

1. **sp\_GetClassAverageLectureAttendanceRate**

i. Purpose: This stored procedure retrieves the average lecture attendance rate for all of the classes.

ii. Groupings and Aggregations

The query aggregates the attendance percentages to calculate the average lecture attendance rate.

iii. SQL Code

CREATE OR ALTER PROCEDURE sp\_GetClassAverageLectureAttendanceRate

AS

BEGIN

SELECT \* FROM vw\_ClassAverageLectureAttendanceRate;

END

GO

1. **sp\_GetClassAverageLabAttendanceRate**

i. Purpose: This stored procedure retrieves the average lab attendance rate for all of the classes.

ii. Groupings and Aggregations

The query aggregates the attendance percentages to calculate the average lab attendance rate.

iii. SQL Code

CREATE OR ALTER PROCEDURE sp\_GetClassAverageLabAttendanceRate

AS

BEGIN

SELECT \* FROM vw\_ClassAverageLabAttendanceRate;

END

GO

1. **sp\_GetClassAverageAttendanceRate**

i. Purpose: This stored procedure retrieves both the average lab and lecture attendance rate for all the classes.

ii. Groupings and Aggregations

The query aggregates the attendance percentages from both lecture and lab views to calculate the overall average attendance rate.

iii. SQL Code

CREATE OR ALTER PROCEDURE sp\_GetClassAverageAttendanceRate

AS

BEGIN

SELECT \* FROM vw\_ClassAverageAttendanceRate;

END

GO

1. **sp\_LabWarningList**

i. Purpose: This procedure retrieves a list of students whose lab attendance rate is below a specified threshold.

ii. Groupings and Aggregations

* **Groupings:** Group by student ID and name
* **Aggregations:** The query aggregates the attendance records to calculate the lab attendance rate for each student.

iii. SQL Code

CREATE OR ALTER PROCEDURE sp\_GetLabWarningList

@Threshold DECIMAL(5,2)

AS

BEGIN

SELECT

s.Student\_ID,

s.Student\_Name,

SUM(CASE WHEN sa.SS\_Attendance = '100.00%' THEN 1 ELSE 0 END) AS AttendedLabs,

COUNT(sa.S\_ID) AS TotalLabs,

CAST(SUM(CASE WHEN sa.SS\_Attendance = '100.00%' THEN 1 ELSE 0 END) AS FLOAT) / NULLIF(COUNT(sa.S\_ID), 0) \* 100 AS LabAttendanceRate

FROM

Student s

JOIN StudentAttendance sa ON s.Student\_ID = sa.Student\_ID

JOIN ClassSession cs ON sa.S\_ID = cs.S\_ID

WHERE

cs.S\_Type = 1 -- Lab

GROUP BY

s.Student\_ID, s.Student\_Name

HAVING

CAST(SUM(CASE WHEN sa.SS\_Attendance = '100.00%' THEN 1 ELSE 0 END) AS FLOAT) / NULLIF(COUNT(sa.S\_ID), 0) \* 100 < @Threshold;

END

GO

# Dashboard Screenshots

Given below are the screenshots of the dashboard that I have created. The home page shows an overview of the dashboard that links to all the various sections.

The screenshot below shows the sections and all the elements I have used such as columns and tables. I have also used colours to make it more appealing.

A screenshot of a computer

Description automatically generated

Once you deploy the dashboard or click on the URL, you can put the metrics in the Student ID, Session ID and the Threshold as shown in the screenshot below. I have put Sam’s student ID below and as you can see and we can see the results.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

# User Guide for Student Attendance Dashboard

Th Student Attendance Dashboard is designed to help provide Dr. Diana with a comprehensive view of student attendance in her course DS6504. One can access it by using the provided URL in the web browser. This dashboard has been sectioned into 3 main parts to help her monitor all the metrics: -

* **Individual Student Performance**:

This section allows users to filter and view attendance data for specific students. The parameters Student ID and Session ID are used. A table populates with details showing Student Name, Session Name, Date of Session, Type (Lecture/Lab), Attendance Status (Yes/No). It also visualizes Attendance Over Time with a column chart that displays the student's attendance over time, providing insights into their engagement.

* **Warning Lists**:

This section highlights the students who are at risk of falling behind, based on all their attendance rates. One can adjust the threshold percentage using the parameter provided. A table displays students with attendance rates below the selected threshold. The table includes columns for: Student ID, Student Name, Attended Labs, Total Labs, and Lab Attendance Rate.

* **Overall Class Attendance Metrics**:

This section displays three columns representing**-** Class Average Lecture Attendance Rate, Class Average Lab Attendance Rate, and Class Average Overall Attendance Rate. Each column provides a visual representation of the average attendance rates. The values displayed indicate the percentage of attendance for lectures, labs, and overall.

**Interactivity Features**

* **Dynamic Filtering**: The dashboard allows users to filter data dynamically based on selected parameters.
* **Drill-Through Actions**: Users can click on specific entries in warning lists to drill down into detailed reports for individual students.

# Conclusion

The Student Attendance dashboard provides a powerful tool for Dr. Diana to help monitor and analyse student attendance data. It serves as critical tool for enhancing the academic success of students in the DS6504 course. The insights gained from this dashboard will help contribute to improved student performance and retention rates. The integration of drill-through and dynamic filtering functionalities further enhances its usability, ensuring that the tutor of the respective course, Dr. Diana has the necessary resources at her fingertips to support her students effectively.