Department of Information and Communication Technology

Faculty of Technology

University of Ruhuna

Database Management Systems Practicum
ICT 1222
Assignment 02

Group 08

Submitted to: Mr. P.H.P. Nuwan Laksiri

Submitted by: TG/2021/1057 – W.A.K.P. Lakshan

TG/2021/1064 – K.A.N. Perera

TG/2021/1052 – G.A.I. Rangajeewa

TG/2021/1035 – A.M.S.S. Kumara

Contents

01. Brief introduction about the problem/group project 1

02. Brief introduction to the solution 2

03. Proposed ER/EER diagram 3

04. Proposed Relational mapping diagram 4

05. Table structure of your solution 5

06. Architecture of your solution 9

07. Tools and technologies that you have used. 9

08. Security measures that you have taken to protect your DB. 9

09. Brief description about DB Accounts/Users and the reasons for creating such Accounts/Users 10

10. Code snippets to support your work 11

10.1 Query 01: (To view all attendance for all the subjects with percentage and eligibility) 11

10.2 Query 02: (To view attendance as individuals by giving the registration no) 11

10.3 Query 03: (To view attendance as individuals by giving the registration no, course code) 12

10.4 Query 04: (view_attendance_for_theory) 12

10.5 Query 05: (view_attendance_for_practical) 13

10.6 Query 06: (To view all student CA mark for given cource code) 13

10.7 Query 07: (To view all student CA mark for given cource code and Student id) 14

10.8 Query 08: (To view all CA mark for given Student id) 14

10.9 Query 09: (View All Final marks for whole batch 14

10.10 Query 10: (View all final marks for only student) 15

10.11 Query 11: (To view all student who are eligible?) 15

10.12 Query 12: (To view grades for all student) 15

10.13 Query 01: (View Grades for only student) 16

10.14 Query 14: (To view SGPA & CGPA for all students) 16

10.15 Query 05: (To view SGPA & CGPA for only students) 17

11. Problems that you faced during the development of the solution. 18

12. Solutions/how you have overcome the above identified problems. 18

13. If you are going to host your backend where are you going to host it and reasons for the selection 18

14. If you are going to host your backend in a cloud environment what are the things/changes that you have to do in your backend 19

15. Individual contribution to the backend development. 19

16. References 20

01. Brief introduction about the problem/group project

The Faculty of Technology faces a complex and multifaceted challenge in managing student information, academic performance, and administrative processes. In the traditional academic environment, the manual management of student records, attendance, and marks often leads to inefficiencies, data inconsistencies, and challenges in ensuring compliance with academic regulations. As the faculty grows and adapts to changing educational needs, these challenges become more pronounced.

In this context, the need for a comprehensive system that streamlines the management of student-related data, attendance, and academic performance has become evident. This project aims to address these challenges by developing a robust and user-friendly system that can be tailored to the faculty's specific requirements. The system is designed to provide multiple user roles with tailored permissions, ensuring that administrative tasks are efficiently executed, data is accurately maintained, and academic standards are upheld.

By centralizing student details, attendance records, and academic marks within a secure and well-structured system, the project seeks to enhance the overall academic experience for both students and faculty members. It aims to eliminate the redundancy of manual record-keeping, improve data accuracy, ensure compliance with faculty bylaws, and provide real-time access to essential academic information.

In summary, the problem at hand is the need for an integrated solution that simplifies and modernizes the management of student data, attendance, and academic performance within the Faculty of Technology, enhancing efficiency, data integrity, and the overall quality of academic services. This project is a proactive response to these challenges, designed to provide a comprehensive and effective solution for the faculty's administrative and academic needs.

02. Brief introduction to the solution

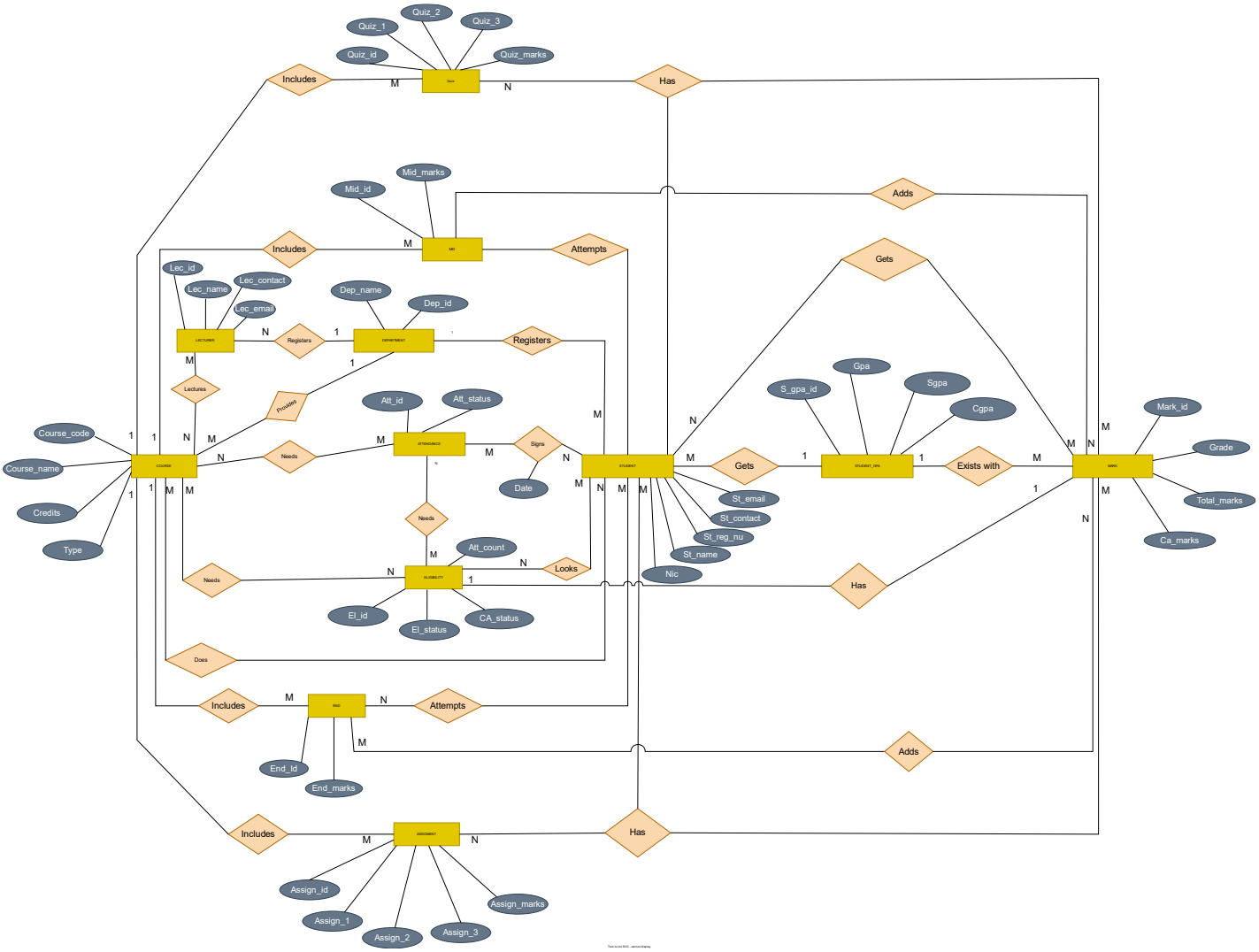
In response to the complex challenges faced by the Faculty of Technology in managing student data, academic performance, and administrative processes, this project proposes a comprehensive solution. The solution is designed to modernize and streamline the management of student-related data, attendance, and academic performance within the faculty.

Key elements of the solution include the development of a user-friendly and robust system that can be customized to meet the faculty's specific requirements. This system introduces a multi-tiered approach, providing different user roles with tailored permissions to ensure efficient execution of administrative tasks while maintaining data accuracy and upholding academic standards.

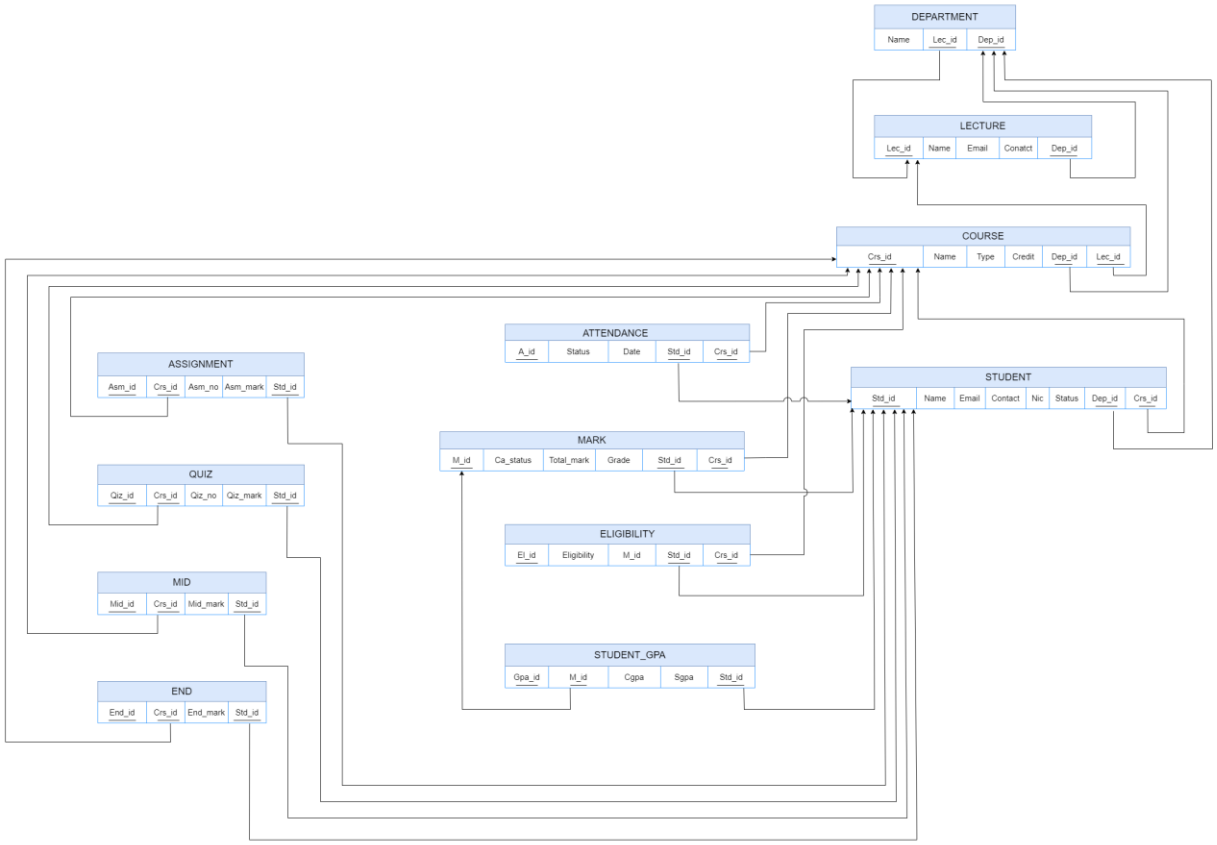
Central to the solution is the establishment of a secure and well-structured database that centralizes student details, attendance records, and academic marks. This centralization eliminates redundancy, enhances data accuracy, and ensures compliance with faculty regulations. Moreover, it offers real-time access to essential academic information for both students and faculty members.

The proposed solution represents a proactive response to the challenges faced by the Faculty of Technology. By implementing this comprehensive system, the project aims to enhance administrative efficiency, data integrity, and the overall quality of academic services. It offers a modernized and efficient approach to managing student data and academic processes, ensuring a more productive and effective educational environment.

03. Proposed ER/EER diagram



04. Proposed Relational mapping diagram



05. Table structure of your solution

```
mysql> desc assignment;
+----------+------------+------+-----+---------+-------+
| Field    | Type       | Null | Key | Default | Extra |
+----------+------------+------+-----+---------+-------+
| Asm_id   | char(5)    | NO   | PRI | NULL    |       |
| Asm_no   | int        | NO   |     | NULL    |       |
| Asm_mark | int        | YES  |     | NULL    |       |
| Std_id   | varchar(6) | NO   | MUL | NULL    |       |
| Crs_id   | varchar(7) | NO   | MUL | NULL    |       |
+----------+------------+------+-----+---------+-------+
5 rows in set (0.43 sec)
```

```
mysql> desc mid;
+----------+------------+------+-----+---------+-------+
| Field    | Type       | Null | Key | Default | Extra |
+----------+------------+------+-----+---------+-------+
| Mid_id   | char(5)    | NO   | PRI | NULL    |       |
| Crs_id   | varchar(7) | NO   | MUL | NULL    |       |
| Mid_mark | int        | YES  |     | NULL    |       |
| Std_id   | char(6)    | NO   | MUL | NULL    |       |
+----------+------------+------+-----+---------+-------+
4 rows in set (0.11 sec)
```

```
mysql> desc quiz;
+----------+------------+------+-----+---------+-------+
| Field    | Type       | Null | Key | Default | Extra |
+----------+------------+------+-----+---------+-------+
| Quiz_id  | char(4)    | NO   | PRI | NULL    |       |
| Qiz_no   | int        | NO   |     | NULL    |       |
| Qiz_mark | int        | YES  |     | NULL    |       |
| Std_id   | varchar(6) | NO   | MUL | NULL    |       |
| Crs_id   | varchar(7) | NO   | MUL | NULL    |       |
+----------+------------+------+-----+---------+-------+
5 rows in set (0.07 sec)
```

```
mysql> desc end;
+-----------+------------+------+-----+---------+-------+
| Field     | Type       | Null | Key | Default | Extra |
+-----------+------------+------+-----+---------+-------+
| End_id    | char(5)    | NO   | PRI | NULL    |       |
| Crs_id    | varchar(7) | NO   | MUL | NULL    |       |
| End_marks | int        | YES  |     | NULL    |       |
| Std_id    | char(6)    | NO   | MUL | NULL    |       |
+-----------+------------+------+-----+---------+-------+
4 rows in set (0.05 sec)
```

```
mysql> desc attendance;
+--------+------------+------+-----+---------+-------+
| Field  | Type       | Null | Key | Default | Extra |
+--------+------------+------+-----+---------+-------+
| A_id   | char(4)    | NO   | PRI | NULL    |       |
| Count  | char(2)    | NO   |     | NULL    |       |
| Std_id | char(6)    | NO   | MUL | NULL    |       |
| Crs_id | varchar(7) | NO   | MUL | NULL    |       |
+--------+------------+------+-----+---------+-------+
4 rows in set (0.13 sec)
```

```
mysql> desc mark;
+------------+------------+------+-----+---------+-------+
| Field      | Type       | Null | Key | Default | Extra |
+------------+------------+------+-----+---------+-------+
| Std_id     | char(6)    | NO   |     | NULL    |       |
| Crs_id     | varchar(7) | NO   |     | NULL    |       |
| CA_Status  | varchar(4) | NO   |     |         |       |
| Total_Mark | bigint     | YES  |     | NULL    |       |
| Grade      | varchar(2) | NO   |     |         |       |
+------------+------------+------+-----+---------+-------+
5 rows in set (0.01 sec)
```

```
mysql> desc eligibility;
+------------------+------------+------+-----+---------+-------+
| Field            | Type       | Null | Key | Default | Extra |
+------------------+------------+------+-----+---------+-------+
| Std_id           | char(6)    | NO   |     | NULL    |       |
| Crs_id           | varchar(7) | NO   |     | NULL    |       |
| Total_Attendance | char(2)    | NO   |     | NULL    |       |
| CA_Mark          | bigint     | YES  |     | NULL    |       |
| Eligibility      | varchar(3) | NO   |     |         |       |
+------------------+------------+------+-----+---------+-------+
5 rows in set (0.01 sec)
```

```
mysql> desc  student_sgpa;
+--------+---------------+------+-----+---------+-------+
| Field  | Type          | Null | Key | Default | Extra |
+--------+---------------+------+-----+---------+-------+
| Std_id | char(6)       | NO   |     | NULL    |       |
| SGPA   | decimal(39,6) | YES  |     | NULL    |       |
+--------+---------------+------+-----+---------+-------+
2 rows in set (0.06 sec)

mysql> desc  student_cgpa;
+--------+---------------+------+-----+---------+-------+
| Field  | Type          | Null | Key | Default | Extra |
+--------+---------------+------+-----+---------+-------+
| Std_id | char(6)       | NO   |     | NULL    |       |
| CGPA   | decimal(39,6) | YES  |     | NULL    |       |
+--------+---------------+------+-----+---------+-------+
2 rows in set (0.01 sec)
```

```
mysql> desc department;
+--------+--------------+------+-----+---------+-------+
| Field  | Type         | Null | Key | Default | Extra |
+--------+--------------+------+-----+---------+-------+
| Dep_id | char(3)      | NO   | PRI | NULL    |       |
| Name   | varchar(150) | NO   |     | NULL    |       |
| Lec_id | char(5)      | NO   | MUL | NULL    |       |
+--------+--------------+------+-----+---------+-------+
3 rows in set (0.34 sec)
```

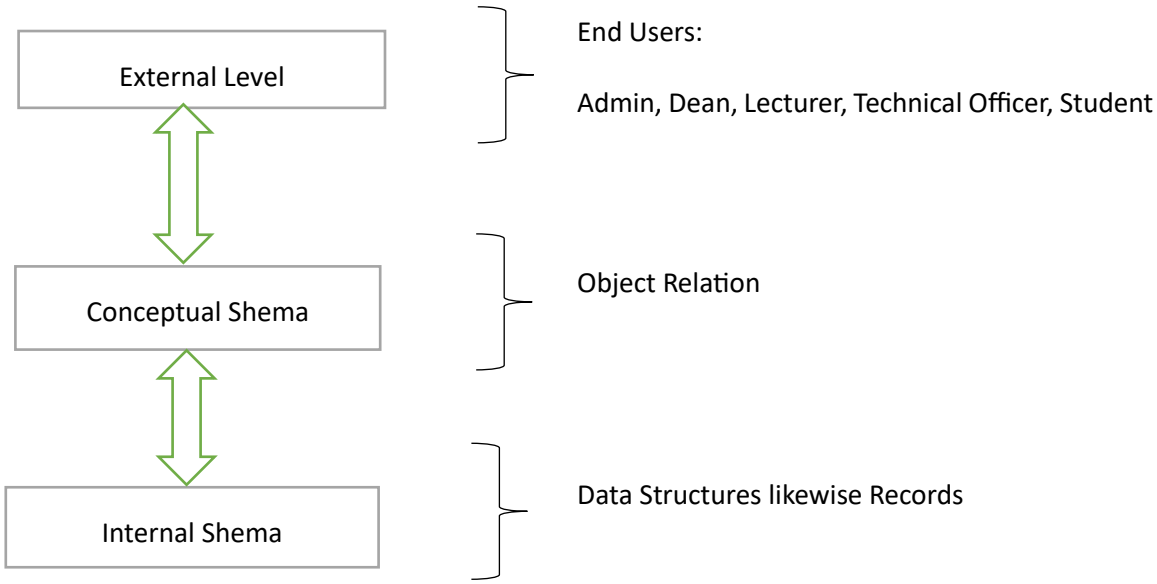
```
mysql> desc lecture;
+---------+--------------+------+-----+---------+-------+
| Field   | Type         | Null | Key | Default | Extra |
+---------+--------------+------+-----+---------+-------+
| Lec_id  | char(5)      | NO   | PRI | NULL    |       |
| Name    | varchar(250) | YES  |     | NULL    |       |
| Email   | varchar(150) | YES  |     | NULL    |       |
| Contact | varchar(15)  | YES  |     | NULL    |       |
| Dep_id  | char(3)      | NO   | MUL | NULL    |       |
+---------+--------------+------+-----+---------+-------+
5 rows in set (0.01 sec)
```

```
mysql> desc course;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| Crs_id | varchar(7)  | NO   | PRI | NULL    |       |
| Name   | varchar(40) | NO   |     | NULL    |       |
| Type   | varchar(10) | NO   |     | NULL    |       |
| Credit | int         | NO   |     | NULL    |       |
| Dep_id | char(3)     | NO   | MUL | NULL    |       |
| Lec_id | char(5)     | NO   | MUL | NULL    |       |
+--------+-------------+------+-----+---------+-------+
6 rows in set (0.01 sec)
```

```
mysql> desc student;
+---------+--------------+------+-----+---------+-------+
| Field   | Type         | Null | Key | Default | Extra |
+---------+--------------+------+-----+---------+-------+
| Std_id  | char(6)      | NO   | PRI | NULL    |       |
| Name    | varchar(250) | YES  |     | NULL    |       |
| Email   | varchar(150) | YES  |     | NULL    |       |
| Contact | varchar(15)  | YES  |     | NULL    |       |
| NIC     | varchar(12)  | NO   |     | NULL    |       |
| Status  | varchar(50)  | YES  |     | NULL    |       |
| Dep_id  | char(3)      | NO   | MUL | NULL    |       |
+---------+--------------+------+-----+---------+-------+
7 rows in set (0.01 sec)
```

```
mysql> desc student_course;
+--------+------------+------+-----+---------+-------+
| Field  | Type       | Null | Key | Default | Extra |
+--------+------------+------+-----+---------+-------+
| std_id | char(6)    | NO   |     | NULL    |       |
| Crs_id | varchar(7) | NO   |     | NULL    |       |
+--------+------------+------+-----+---------+-------+
2 rows in set (0.06 sec)
```

06. Architecture of your solution



07. Tools and technologies that you have used.

- draw.io: Used to draw ER diagram, Relational Mapping and Table structure.
- MySQL, Visual Studio Code, Notepad ++, WAMP Server: Used to Create, Modify and Maintain the Database which created.

08. Security measures that you have taken to protect your DB.

- Applying user privileges to users:
  1. Admin: All PRIVILEGES with GRANT OPTION for all tables
  2. Dean: All PRIVILEGES without GRANT OPTION for all tables
  3. Lecturer: All PRIVILEGES without GRANT and USER creation for all tables
  4. Technical Officer: Select, Write and Update permissions for attendance related tables/views.
  5. Student: Select permission for final attendance and final marks/Grades tables/views

09. Brief description about DB Accounts/Users and the reasons for creating such Accounts/Users

The Student Database Management System contain below user accounts:

- Admin: Create and maintain user accounts.
- Dean: Can perform SELECT, INSERT, UPDATE, DELETE…
- Lecturer: Only can SELECT, INSERT, DELETE, UPDATE tables related to QUIZ, Assignment, MID, END.
- Technical Officer: Only Can SELECT, INSERT, UPDATE, DELETE tables and views which related to attendance.
- Student: Only can SELECT operations on eligibility, foundation_of_mark, course_grade_points_view, attendance, student_cgpa, student_sgpa tables and views.

10. Code snippets to support your work

10.1 Query 01: (To view all attendance for all the subjects with percentage and eligibility)

```
DELIMITER //
CREATE PROCEDURE view_attendance_for_subject()
BEGIN
SELECT
  Crs_id,
  CONCAT(Round((Total_Attendance * 6.66),2) ,'%') AS "Attendance",
  Eligibility
FROM ELIGIBILITY;
END//
DELIMITER ;

CALL view_attendance_for_subject();
```

10.2 Query 02: (To view attendance as individuals by giving the registration no)

```
DELIMITER //
CREATE PROCEDURE view_attendance_for_individualStd(IN id CHAR(6))
BEGIN
SELECT
  Crs_id,
  CONCAT(Round((Total_Attendance * 6.66),2) ,'%') AS "Attendance",
  Eligibility
FROM ELIGIBILITY
WHERE Std_id = id;
END//
DELIMITER ;

CALL view_attendance_for_individualStd();
```

10.3 Query 03: (To view attendance as individuals by giving the registration no, course code)

```
DELIMITER //
CREATE PROCEDURE view_attendance_for_StudentCource(IN sid CHAR(6), IN cid CHAR(7))
BEGIN
 SELECT
   Crs_id,
   CONCAT(ROUND((Total_Attendance * 6.66), 2), '%') AS "Attendance",
   Eligibility
 FROM ELIGIBILITY
 WHERE Std_id = sid AND Crs_id = cid;
END//
DELIMITER ;
```

CALL view_attendance_for_StudentCource(Std_id,Crs_id);

10.4 Query 04: (view_attendance_for_theory)

```
DELIMITER //
CREATE PROCEDURE view_attendance_for_theory()
BEGIN
SELECT
   e.Std_id,
   e.Crs_id,
   CONCAT(Round((e.Total_Attendance * 6.66),2) ,'%') AS "Attendance",
   e.Eligibility
   FROM ELIGIBILITY as e
   INNER JOIN course as c on e.Crs_id = c.Crs_id
   WHERE c.Type = 'Theory'
   ORDER BY Std_id;
END//
DELIMITER ;
```

CALL view_attendance_for_theory();

10.5 Query 05: (view_attendance_for_practical)

```
DELIMITER //
CREATE PROCEDURE view_attendance_for_practical()
BEGIN
SELECT
   e.Std_id,
   e.Crs_id,
   CONCAT(Round((e.Total_Attendance * 6.66),2) ,'%') AS "Attendance",
   e.Eligibility
   FROM ELIGIBILITY as e
   INNER JOIN course as c on e.Crs_id = c.Crs_id
   WHERE c.Type = 'practical'
   ORDER BY Std_id;
END//
DELIMITER ;
```

CALL view_attendance_for_practical();

10.6 Query 06: (To view all student CA mark for given cource code)

```
DELIMITER //
CREATE PROCEDURE view_all_CA_mark_for_cource(IN cid CHAR(7))
BEGIN
select Std_id,CA_Mark from foundation_of_mark where Crs_id = cid;
END//
DELIMITER ;
```

CALL view_all_CA_mark_for_cource(Crs_id);

10.7 Query 07: (To view all student CA mark for given cource code and Student id)

```
DELIMITER //
CREATE PROCEDURE view_CA_mark_for_CourceStudent(IN cid CHAR(7), IN sid CHAR(6))
BEGIN
select Std_id,CA_Mark from foundation_of_mark where Crs_id = cid AND Std_id = sid;
END//
DELIMITER ;
```

CALL view_CA_mark_for_CourceStudent(Crs_id,Std_id);

10.8 Query 08: (To view all CA mark for given Student id)

```
DELIMITER //
CREATE PROCEDURE view_CA_mark_for_Student(IN sid CHAR(6))
BEGIN
select Crs_id,CA_Mark from foundation_of_mark where Std_id = sid;
END//
DELIMITER ;
```

CALL view_CA_mark_for_Student(Std_id);

10.9 Query 09: (View All Final marks for whole batch)

```
DELIMITER //
CREATE PROCEDURE view_Final_mark_for_All()
BEGIN
select Std_id,Crs_id,Total_Mark from foundation_of_mark;
END//
DELIMITER ;
```

CALL view_Final_mark_for_All();

10.10 Query 10: (View all final marks for only student)

```
DELIMITER //
CREATE PROCEDURE view_All_Final_mark_for_student(IN sid CHAR(6))
BEGIN
select Std_id,Crs_id,Total_Mark from foundation_of_mark where Std_id = sid;
END//
DELIMITER ;
```

CALL view_All_Final_mark_for_student(Std_id);

10.11 Query 11: (To view all student who are eligible?)

```
DELIMITER //
CREATE PROCEDURE view_All_Eligibile_student()
BEGIN
SELECT Std_id,Crs_id,Eligibility FROM eligibility;
END//
DELIMITER ;
```

CALL view_All_Eligibile_student();

10.12 Query 12: (To view grades for all student)

```
DELIMITER //
CREATE PROCEDURE view_Grades_for_All()
BEGIN
select Std_id,Crs_id,Grade from mark ;
END//
DELIMITER ;
```

CALL view_Grades_for_All();

10.13 Query 01: (View Grades for only student)

```
DELIMITER //
CREATE PROCEDURE view_Grades_for_student(IN sid CHAR(6))
BEGIN
select Std_id,Crs_id,Grade from mark where Std_id = sid;
END//
DELIMITER ;
```

CALL view_Grades_for_student(Std_id);

10.14 Query 14: (To view SGPA & CGPA for all students)

```
DELIMITER //
CREATE PROCEDURE view_SGPA_CGPA_For_All()
BEGIN
SELECT
   s.Std_id,
   s.SGPA,
   c.CGPA
FROM
   student_sgpa AS s
INNER JOIN student_cgpa AS c ON s.Std_id = c.Std_id;
END//
DELIMITER ;
```

CALL view_SGPA_CGPA_For_All();

10.15 Query 05: (To view SGPA & CGPA for only students)

```
DELIMITER //
CREATE PROCEDURE view_SGPA_CGPA_For_Student(IN sid CHAR(6))
BEGIN
SELECT
   s.Std_id,
   s.SGPA,
   c.CGPA
FROM
   student_sgpa AS s
INNER JOIN student_cgpa AS c ON s.Std_id = c.Std_id
WHERE s.Std_id = sid;
END//
DELIMITER ;
```

CALL view_SGPA_CGPA_For_All();

11. Problems that you faced during the development of the solution.

- We didn't understand the question of our mini project and it was very difficult to do our work.
- When we draw the ER diagram, we faced some problems like wise wrong connections to Entities and Relations, there were issues of some cardinality notations, inconsistent naming for some attributes and placing entities.
- When we create some tables, sometimes there were errors with data types.
- There were some relations missing in relational mapping.

12. Solutions/how you have overcome the above identified problems.

- We discussed with our group members and got Idea about whole question.
- We built the ER diagram again with patience.
- We used another data types for some tables.
- Missing tables are added to the database.
- Several times the tables and database were dropped and recreated.

13. If you are going to host your backend where are you going to host it and reasons for the selection

➢ Cloud environment

Because,

- Clouds can support relational databases very well.
- Security.
- We can expand our database capacity on runtime.
- Cost efficiency
- Data backup and disaster recovery

14. If you are going to host your backend in a cloud environment what are the things/changes that you have to do in your backend

We have to mod our current database structure. In the current database structure, there are some little bit issues in our database. Also, we need to add more security features to our database. In current database system we have added privileges for our database security.

15. Individual contribution to the backend development.

TG Number	Contribution
TG/2021/1052	• Data insert • Requirement Document • Er-diagram • Table creation • Queries (Store Procedures)
TG/2021/1064	• Data insert • Table creation • Report • Relational Schema • Queries (Store Procedure)
TG/2021/1057	• Data insert • Table creation • Report • Queries (Store Procedure) • Relational Schema • Assign primary keys & foreign keys
TG/2021/1035	• Data insert • Table creation • Requirement Document • Er-diagram • User privilages

16. References

- Lecture Notes and practical Sessions.
- W3 School