

**Department of Information and Communication Technology**

**Faculty of Technology**

**University of Ruhuna**

**Database Management Systems Practicum**

**ICT 1222**

**Assignment 02 – Mini Project Group**

**Group Number 08**

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

TG/2022/1407 – D.M.D.M Disanayaka

TG/2022/1405 – K.M.M.S CHATHURANGA

TG/2022/1406 – R.M.N.C Eshan

TG/2022/1420 – T.M.I Sivmal

Submitted by:

Contents

Brief introduction about the problem/group project……………………………………………………..

Brief introduction to the solution…………………………………………………………………………………..

Proposed ER/EER diagram……………………………………………………………………………………………..

Proposed Relational mapping Diagram………………………………………………………………………….

Table structure of solution…………………………………………………………………………………………….

Tools and technologies that you have used.……………………………………………………………….…..

Security measures that you have taken to protect your Database………………………………………………………………………………………………………………………..

DB Accounts/Users and the reasons for creating such Accounts/Users………………………….

Code snippets to support your work………………………………………………………………………………..

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

Solutions/how you have overcome the above identified problems…………………………………

New database technologies/trends that you have used to develop the backend…………….

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

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……………………………………………………………………………......

Individual contribution to the backend development………………………………………………………..

References………………………………………………………………………………………………………………………..

# Brief introduction about the problem/group project

# The adoption of a Library Management System (LMS) has greatly improved data management within the library, addressing many issues of traditional manual systems.

# • Easier Data Entry and Updates: The LMS reduces the need for manual entry and updating, automating cataloging and record-keeping to lower human error and increase accuracy.

# • Eliminating Redundancy: The LMS prevents duplicate entries, keeping the database clean and organized, which boosts data integrity.

# • Quick Data Access: Staff and patrons can now easily find books, member records, and transaction histories in seconds, replacing slow manual searches with simple clicks.

# • Optimized Space: Digital records reduce the need for physical file storage, freeing up space for study areas and displays.

# • Increased Data Security and Backup: With features like role-based access control, the LMS restricts access to sensitive information and includes regular backups to protect against data loss, ensuring continuity even in case of system issues.

# Brief introduction to the solution

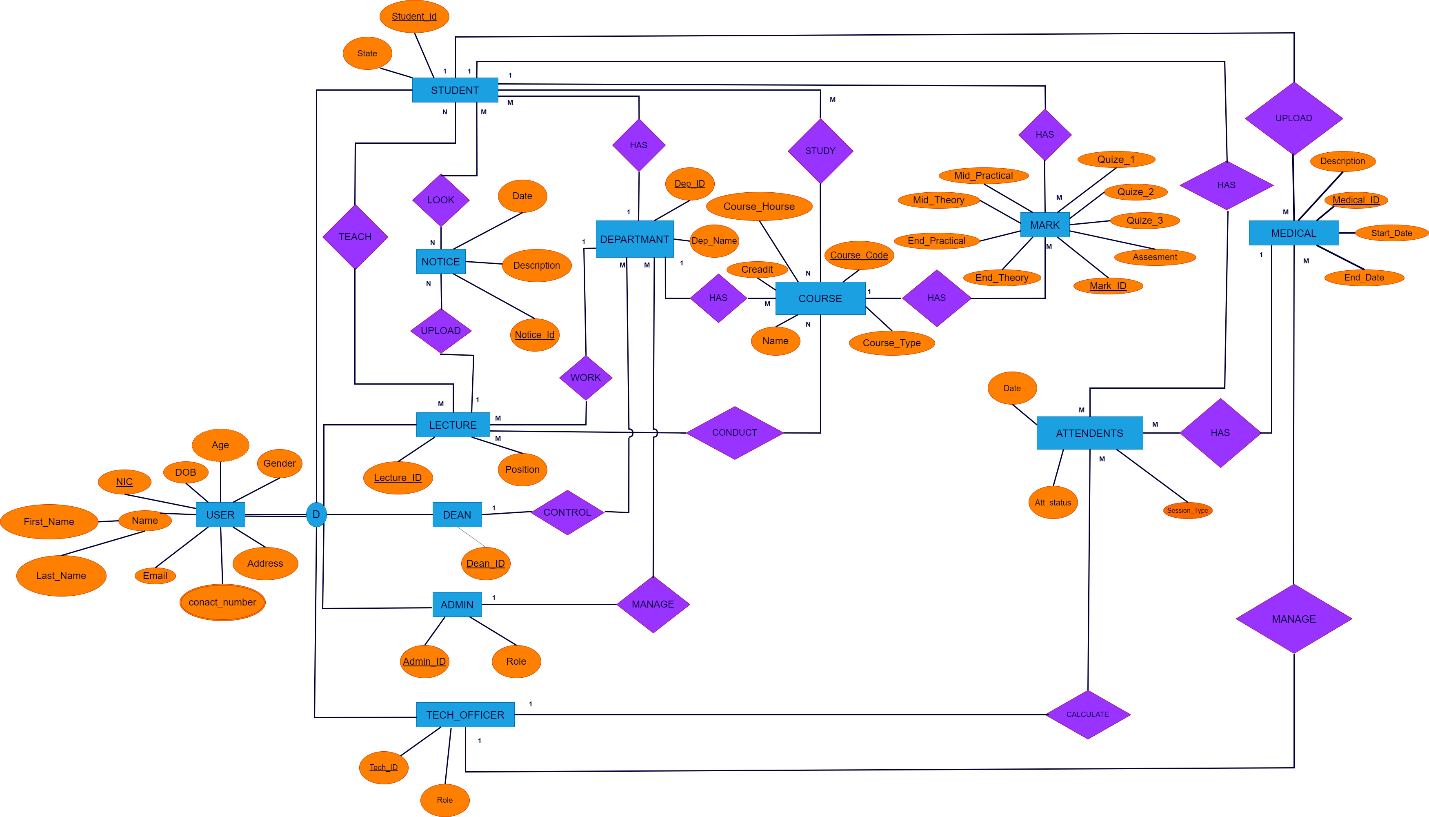
To solve these issues, we created an automated Library Management System (LMS). This system makes it easy to enter, update, and find data compared to the old manual methods.

The LMS uses unique keys to prevent duplicating records, saving space and keeping the database organized. Library staff and users can quickly find information by searching with keywords, without wasting time on manual searches. This system only needs a computer with enough storage, so there’s no need for large physical storage like in traditional systems.

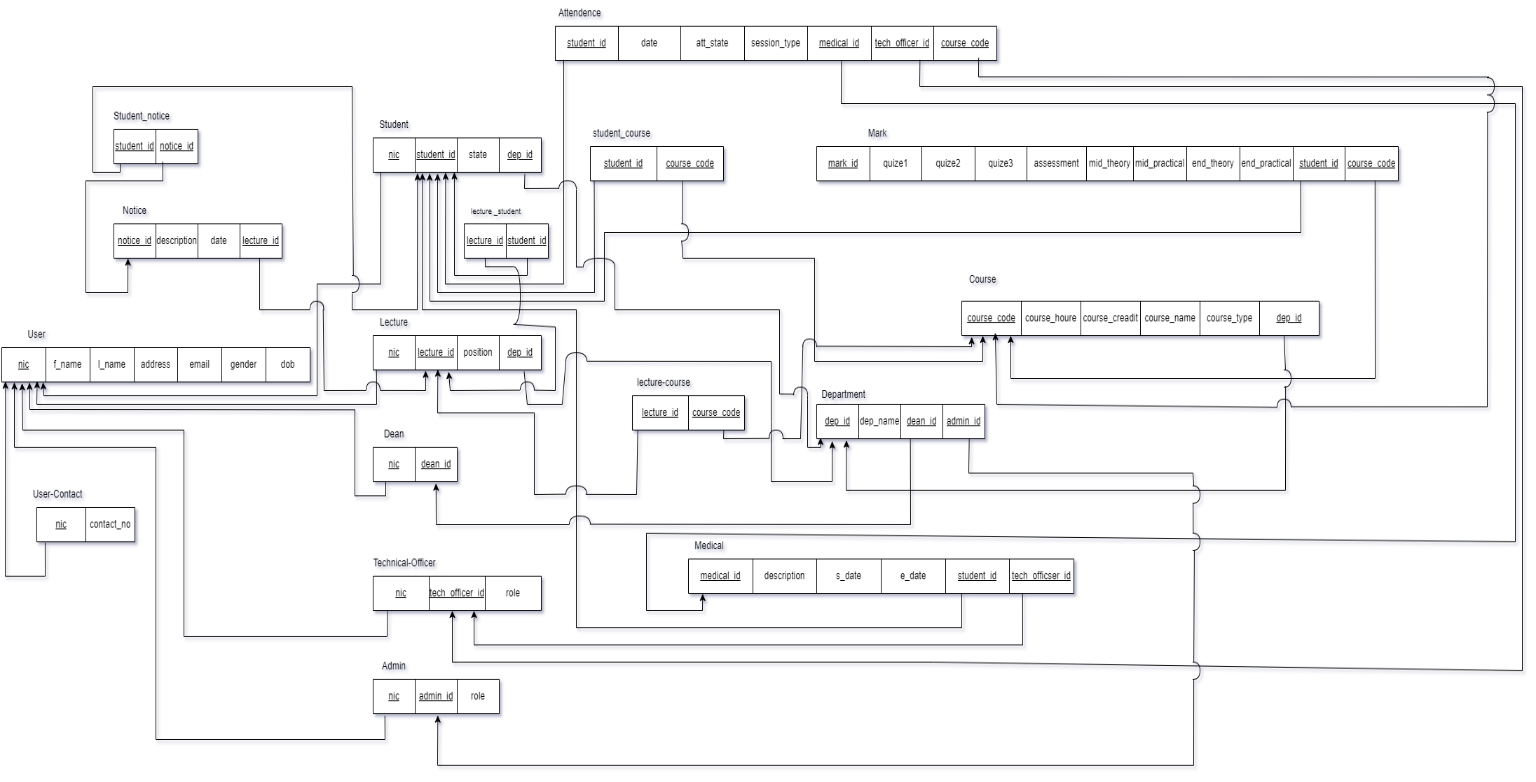
Data is backed up regularly, so it can be recovered easily if lost. The LMS also prevents incorrect data entry, making sure essential information isn’t missed. Automated processes reduce human error and improve data accuracy. Only authorized people can access the system, which boosts security and reduces the risk of data loss.

Overall, this LMS is easy to manage with fewer staff, cutting down on operating costs and labor. It’s a secure, efficient, and cost-effective solution for library management.

Proposed ER/EER diagram



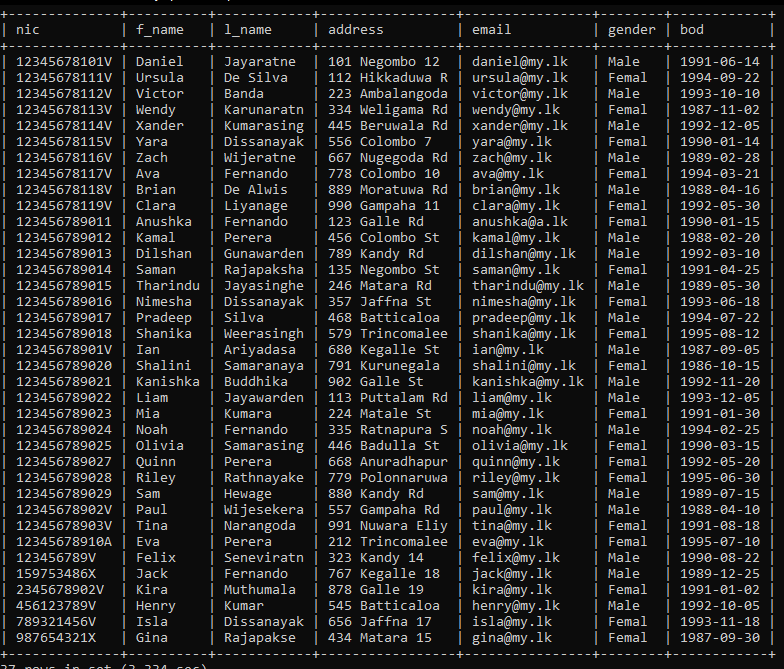
# Proposed Relational mapping Diagram.



# Table structure of solution

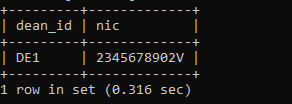
**User**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** |
| nic | char(12) | NO | PRI | NULL |
| f\_name | varchar(10) | YES |  | NULL |
| l\_name | varchar(10) | YES |  | NULL |
| address | varchar(15) | YES |  | NULL |
| email | varchar(15) | YES |  | NULL |
| gender | varchar(5) | YES |  | NULL |
| bod | date | YES |  | NULL |

****

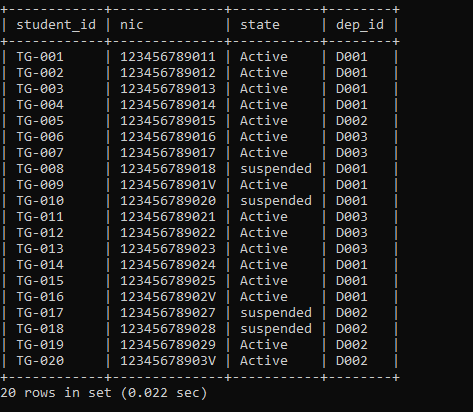
**Dean**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| dean\_id | char(5) | NO | PRI | NULL |  |
| nic | char(12) | YES | MUL | NULL |  |

****

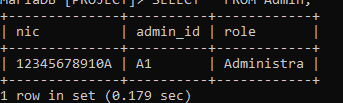
**Students**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Data Type** | **Null** | **Key** | **Default** |
| student\_id | varchar(6) | NO | PRI | NULL |
| nic | char(12) | YES | MUL | NULL |
| state | varchar(30) | YES |  | NULL |
| dep\_id | char(4) | YES | MUL | NULL |

****

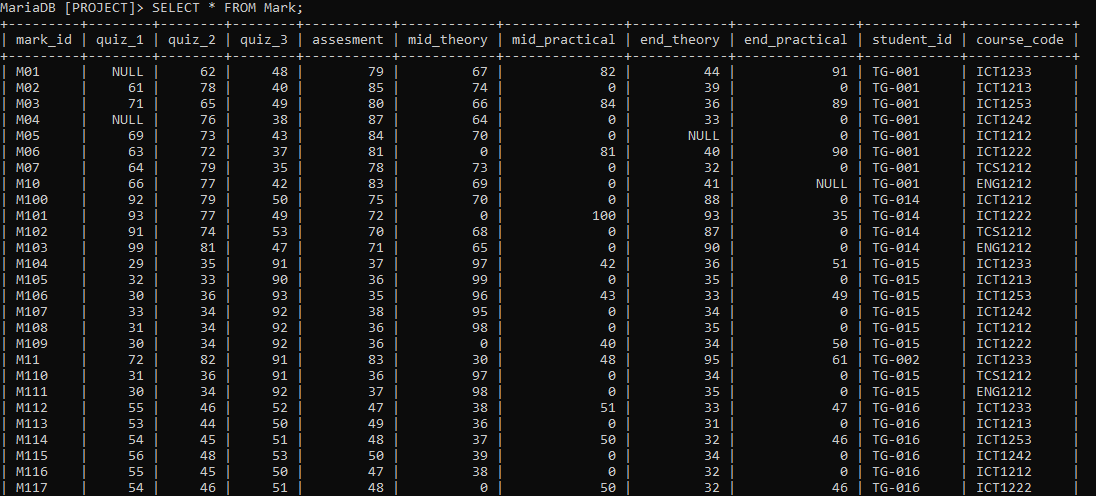
**Admin**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** |
| nic | char(12) | YES | MUL | NULL |
| admin\_id | varchar(10) | NO | PRI | NULL |
| role | varchar(10) | YES |  | NULL |

****

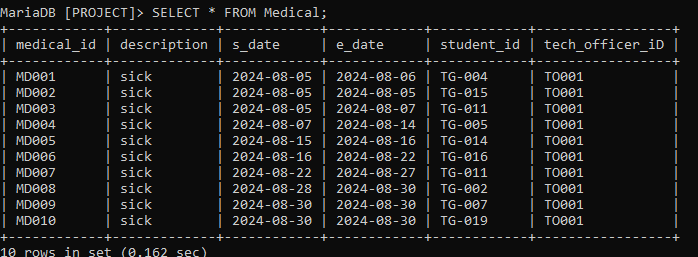
**Mark**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** |
| mark\_id | char(10) | NO | PRI | NULL |
| quiz\_1 | int(11) | YES |  | NULL |
| quiz\_2 | int(11) | YES |  | NULL |
| quiz\_3 | int(11) | YES |  | NULL |
| assesment | int(11) | YES |  | NULL |
| mid\_theory | int(11) | YES |  | NULL |
| mid\_practical | int(11) | YES |  | NULL |
| end\_theory | int(11) | YES |  | NULL |
| end\_practical | int(11) | YES |  | NULL |
| student\_id | varchar(6) | YES | MUL | NULL |
| course\_code | char(8) | YES | MUL | NULL |

****

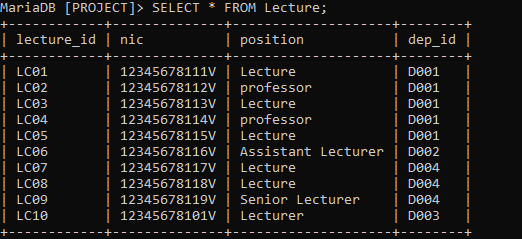
**Medical**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** |
| medical\_id | char(10) | NO | PRI | NULL |
| description | varchar(50) | YES |  | NULL |
| s\_date | date | YES |  | NULL |
| e\_date | date | YES |  | NULL |
| student\_id | varchar(6) | YES | MUL | NULL |
| tech\_officer\_iD | char(10) | YES | MUL | NULL |

****

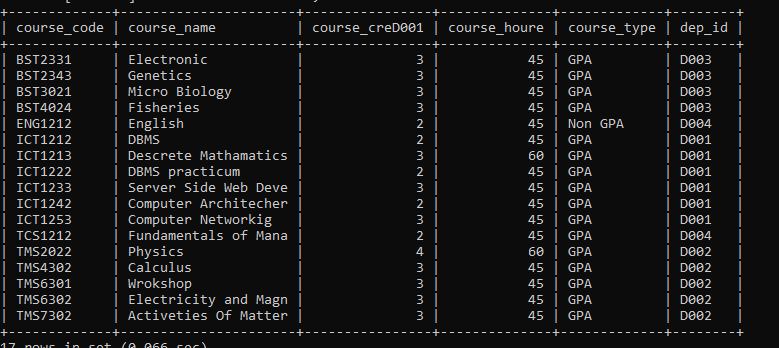
**Lecture**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** |
| lecture\_id | varchar(5) | NO | PRI | NULL |
| nic | char(12) | YES | MUL | NULL |
| position | varchar(20) | YES |  | NULL |
| dep\_id | char(4) | YES | MUL | NULL |

****

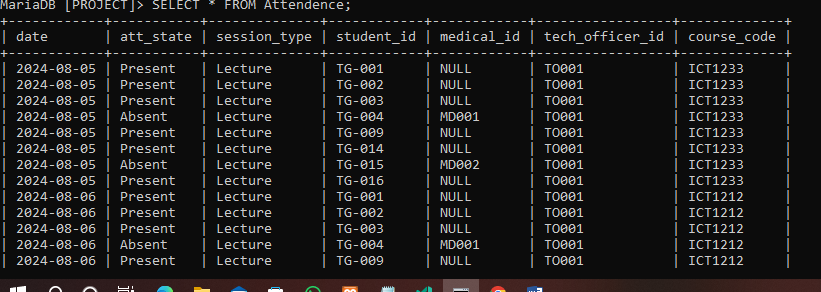
**Course**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** |
| course\_code | char(8) | NO | PRI | NULL |
| course\_name | varchar(20) | YES |  | NULL |
| course\_credit | int(11) | YES |  | NULL |
| course\_houre | int(11) | YES |  | NULL |
| course\_type | varchar(15) | NO |  | NULL |
| dep\_id | char(4) | YES | MUL | NULL |

****

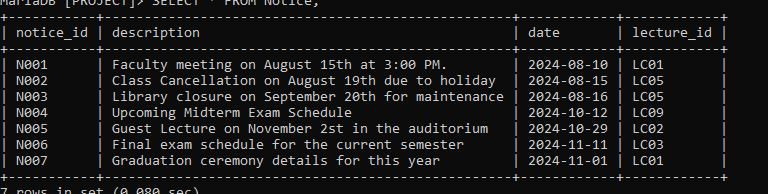
**Attendence**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** |
| date | date | YES |  | NULL |
| att\_state | varchar(20) | YES |  | NULL |
| session\_type | varchar(25) | YES |  | NULL |
| student\_id | varchar(6) | YES | MUL | NULL |
| medical\_id | char(10) | YES | MUL | NULL |
| tech\_officer\_id | varchar(6) | YES | MUL | NULL |
| course\_code | char(8) | YES | MUL | NULL |

****

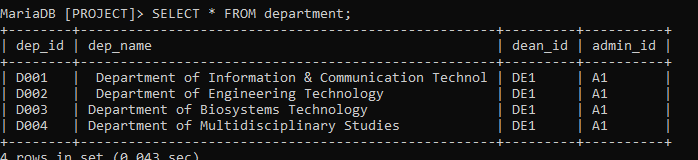
**Notice**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** |
| notice\_id | varchar(6) | YES |  | NULL |
| description | varchar(30) | YES |  | NULL |
| date | date | YES |  | NULL |
| lecture\_id | varchar(5) | YES | MUL | NULL |

****

**Department**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** |
| dep\_id | char(4) | NO | PRI | NULL |
| dep\_name | varchar(50) | YES |  | NULL |
| dean\_id | char(5) | YES | MUL | NULL |
| admin\_id | varchar(10) | YES | MUL | NULL |



# Tools and technologies that you have used.

Draw.io:

* Used to draw ER diagram, relational schema, and table structure.

MySQL SERVER, VS code, Notepad:

* Used to create database and maintain.

GitHub and GitHub Desktop:

* Version Control

# Security measures that you have taken to protect your Database.

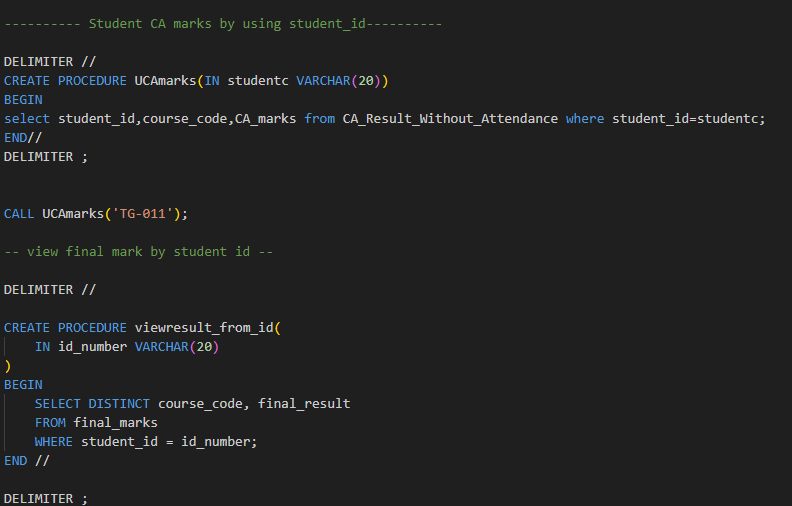
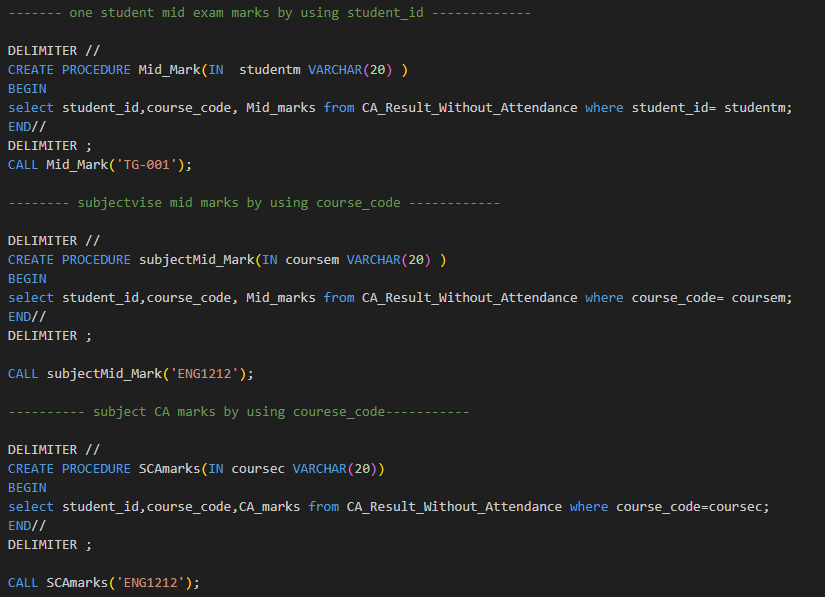
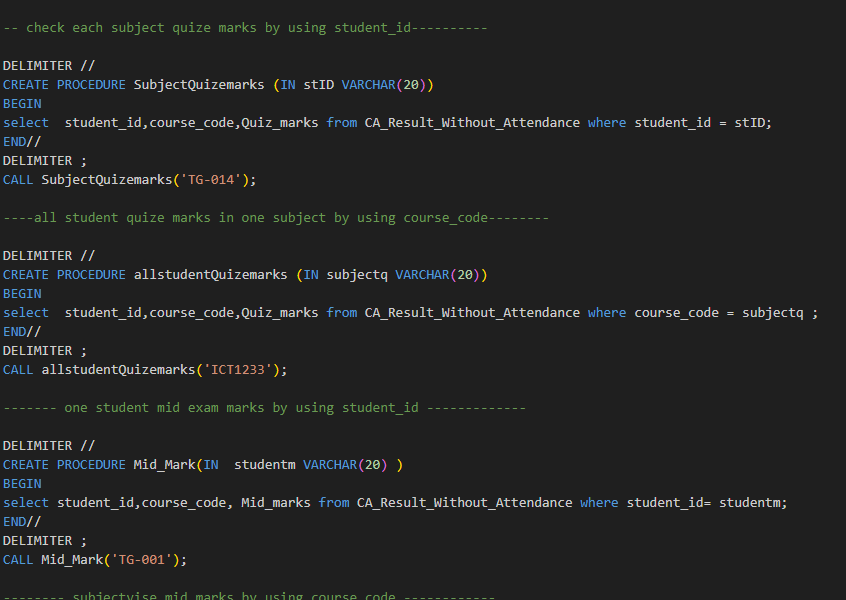
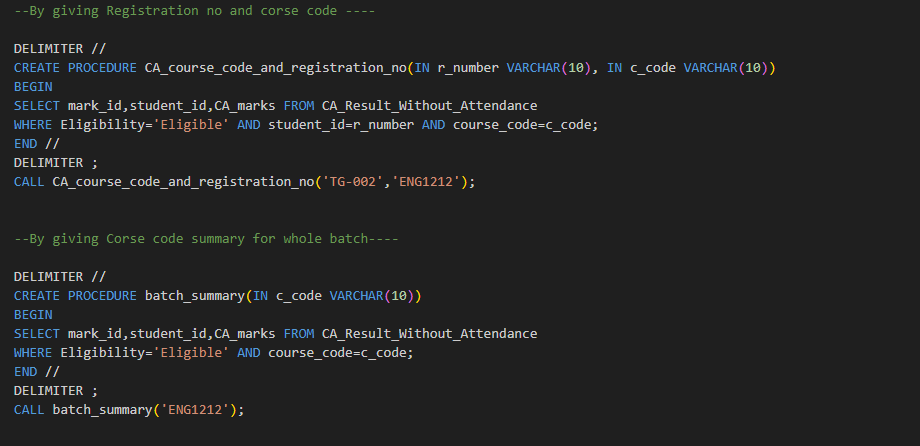
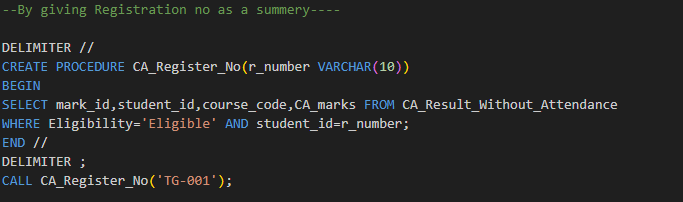
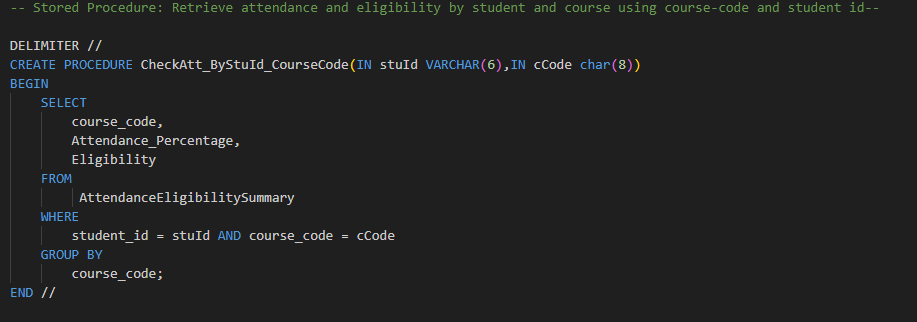
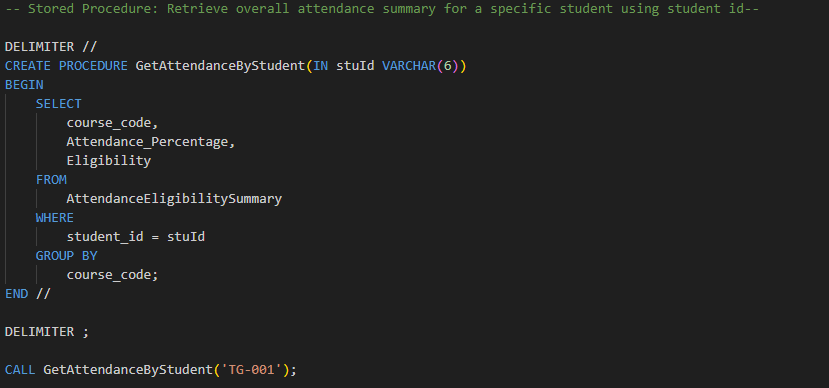
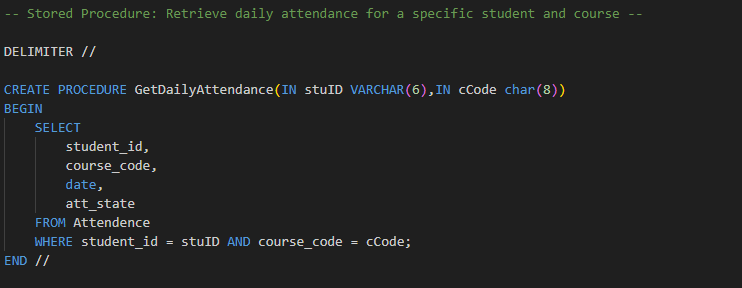
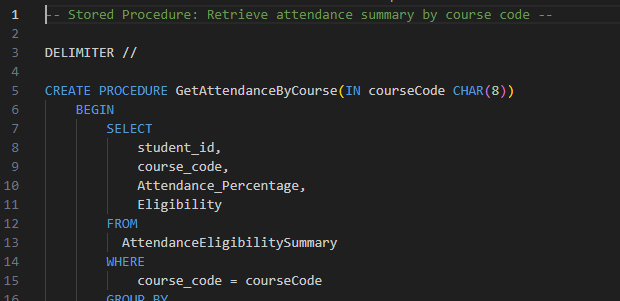
* Admin - With All privileges with Grant Option for all the tables in the database
* Dean - With All privileges without Grant for all the tables in the database
* Lecturer – All privileges without Grant and user creation for all the tables in the database
* Technical Officer - Read, write, and update permissions for attendance related tables/views.
* Student - Read permission for final attendance and final marks/Grades tables/views.

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

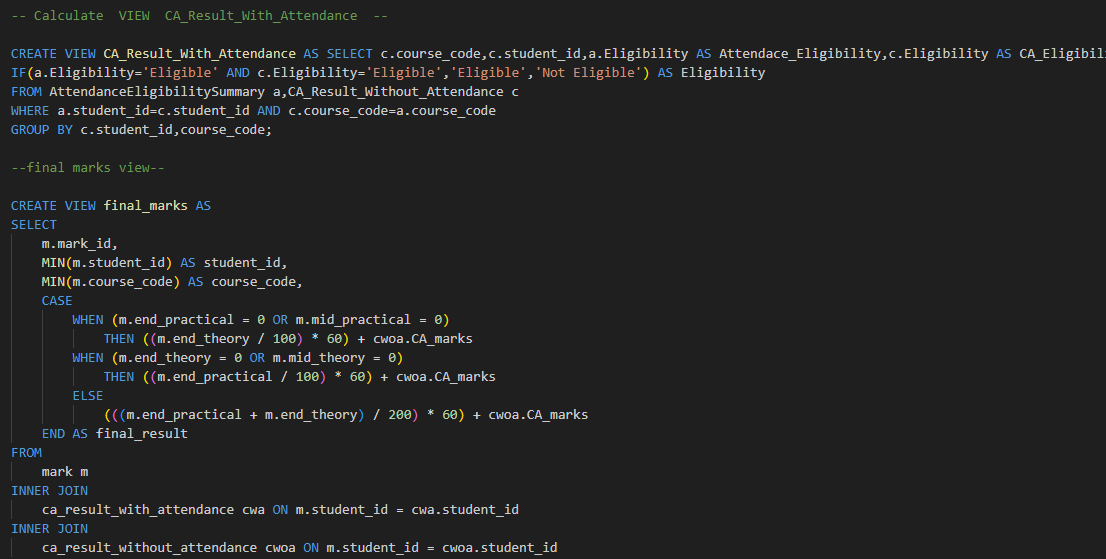
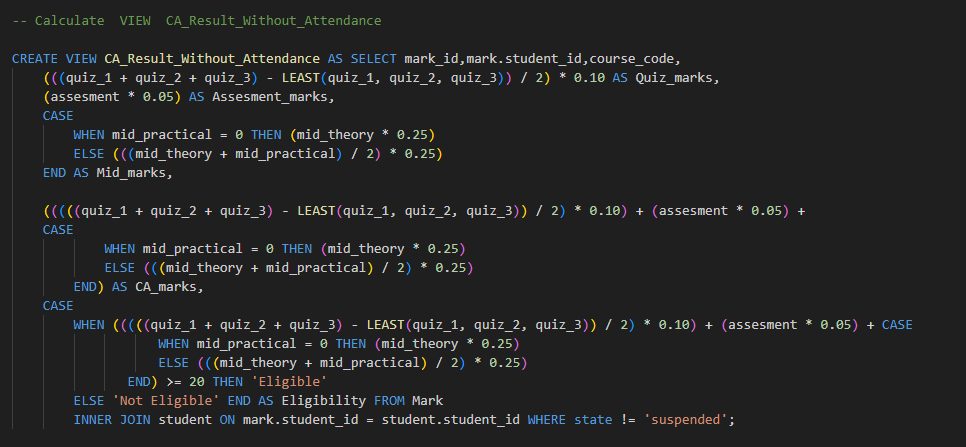
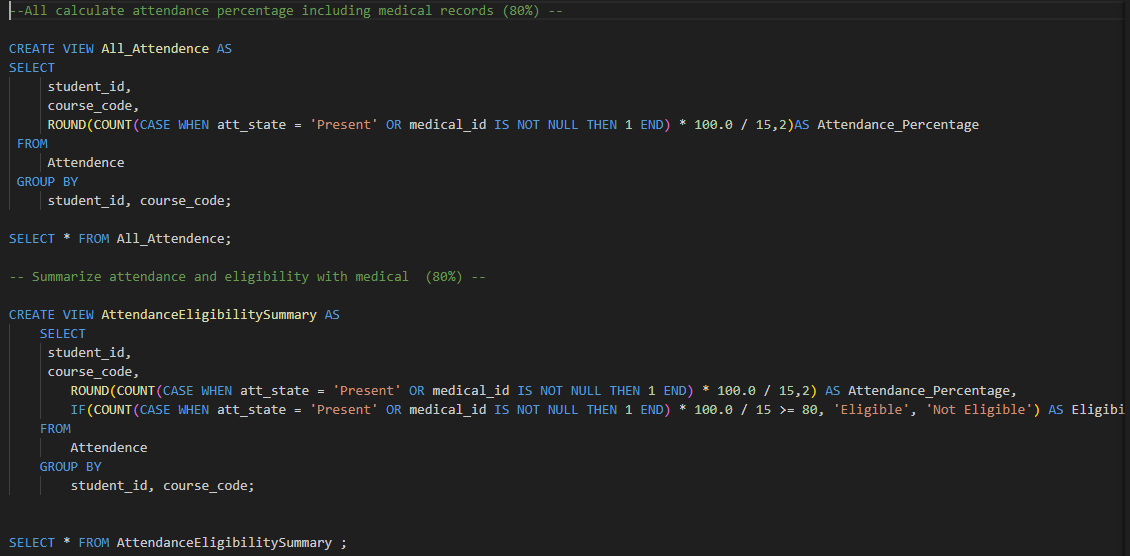
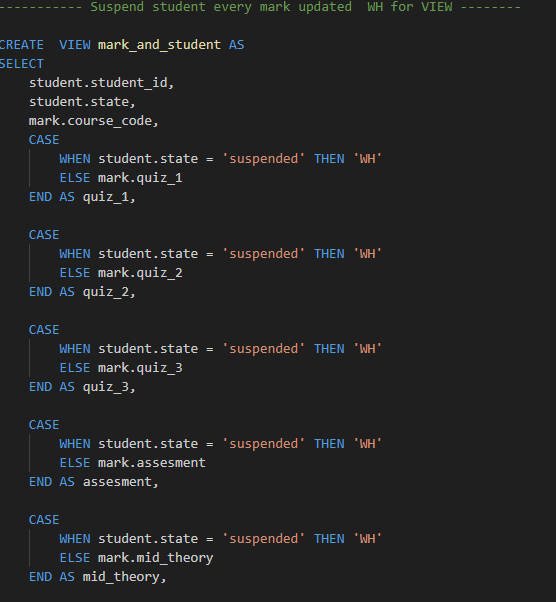
Our Learning Management System contain below user accounts:

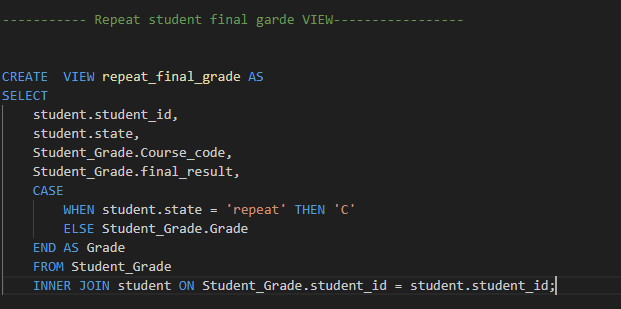
* Admin –
  + Admin can access full system.
  + Maintain database.
* Dean –
  + Can Access any tables and review.
  + Update tables and data.
* Lecturer –
  + Access course table and exammarks table and maintain.
  + Can Update table.
* Technical Officer –
  + maintain Read, write and update permissions for attendance related tables/views.
* Student –
  + Access course tables and can only view marks.

# Code snippets to support your work.



VIEW





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

* Difficult to creation of attendance table, exam mark table, hard to find suitable foreign keys.
* Maintain attendance table data and marks table data.
* We want to change ER diagram and Relational map.
* When we add data to tables there are some errors of data types.

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

* Attendance table data set is separate to weekday files.
* Attendance data set type repetitive tying.

# New database technologies/trends that you have used to develop the backend.

* MySQL
* MySQL Workbench
* Visual Studio Code
* Notepad ++
* GitHub

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

* + Anyone can access the database in any place.
  + Real time accessing.
  + Easy to maintain database.

# 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 want to buy a database server and creating data base into the online server.
  + Change some data types.
  + We have a lots of data in our database.

# Individual contribution to the backend development.

GIT\_Repositories=https://github.com/dilushamadushan/TeamZero\_TECMIS.git

**Clone using the wed URL**

# References

Lecture Notes

W3Schools.com