**ICT1222 - Database Management Systems Practicum Mini Project**



**Department of Information and Communication Technology**

**Faculty of Technology**

**University of Ruhuna**

**Database Management Systems Practicum**

**ICT 1222**

**Assignment 02 – Mini Project**

Group 02

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

Submitted by: ( TG/2023/1756 M.I.M. IRFAN), (TG/2023/1773 K.G.M.R.BANDARA), (TG/2023/1727 D.S.J.SILVA), (TG/2023/1736 U.G.A.N KAUSHALYA )

## **Brief introduction about the problem**

Managing academic data in a university manually or using unstructured spreadsheets is inefficient and prone to errors. Faculty members face challenges in tracking student attendance, recording marks, calculating eligibility for exams, and generating results. Additionally, managing user access for administrators, lecturers, technical officers, and students is cumbersome without a centralized system.

To address these challenges, we developed a database management system specifically for the Faculty of Technology at the University of Ruhuna. This system efficiently handles student details, attendance, marks, results, and eligibility verification, while providing secure access to different user roles. It ensures accurate record-keeping, reduces manual effort, and allows faculty and students to access necessary academic information quickly and reliably.

## **Brief introduction to the solution**

## 

To overcome the challenges of manual academic management, we designed and implemented a centralized database management system for the Faculty of Technology at the University of Ruhuna. The system uses MySQL as the backend and incorporates tables, views, stored procedures, and triggers to automate key processes.

* Views provide simplified access to aggregated data, such as attendance summaries, final marks, grades, and eligibility reports, reducing the need for manual calculations.
* Triggers ensure data consistency and automatically update related tables when certain actions occur, such as marking attendance or updating exam results.
* Stored procedures streamline repetitive tasks, like calculating final marks, checking eligibility for exams, and generating grade reports for students and batches.

By leveraging these database features, the system minimizes manual work, ensures data accuracy, and provides role-based access for administrators, lecturers, technical officers, and students. It allows faculty members to efficiently manage student information, attendance, assessments, and results, while students can securely view their own academic progress.

Proposed ER/EER diagram

| **Entity** | **Primary Key** | **Key Attributes** |
| --- | --- | --- |
| users | user\_id | name, email, role |
| student | user\_id | reg\_no, batch, status, department\_id |
| lecture | user\_id | specialization, designation |
| dean | lecture\_id | term\_start, term\_end |
| tech\_officer | user\_id | – |
| department | department\_id | name, faculty\_name |
| course | course\_id | name, credit, academic\_year, semester, total\_hours |
| session | session\_id | course\_id, session\_date, session\_hours, type |
| attendance | attendance\_id | student\_id, session\_id, status, medical, hours\_attended |
| marks | marks\_id | student\_id, course\_id, quiz1-3, assessment, mid, final, eligibility, grade |
| medical | medical\_id | student\_id, course\_id, exam\_type, date\_submitted, status |
| result | result\_id | student\_id, academic\_year, semester, sgpa, cgpa |

**Relationships:**

* **1:1:** users → student/lecture/tech\_officer, lecture → dean
* **1:N:** student → attendance, student → marks, course → session
* **M:N (junction tables):**
  + student\_course (students enrolled in courses)
  + lecture\_course (lecturers teaching courses)
  + department\_course (courses offered by departments)
  + lecture\_department (lecturers assigned to departments)

**Points:**

* attendance and marks are linked to students and courses, supporting medical handling.
* result tracks SGPA/CGPA per student per semester.
* Constraints enforce **referential integrity** and **data consistency**.

Proposed Relational Mapping Diagram

| **Table** | **FK References** | **Relationship Type** |
| --- | --- | --- |
| student | department(department\_id) | Many-to-One |
| lecture | users(user\_id) | One-to-One |
| dean | lecture(user\_id) | One-to-One |
| tech\_officer | users(user\_id) | One-to-One |
| session | course(course\_id) | Many-to-One |
| attendance | student(user\_id), session(session\_id) | Many-to-One |
| marks | student(user\_id), course(course\_id) | Many-to-One |
| medical | student(user\_id), course(course\_id) | Many-to-One |
| result | student(user\_id) | Many-to-One |
| student\_course | student(user\_id), course(course\_id) | Many-to-Many (junction) |
| lecture\_course | lecture(user\_id), course(course\_id) | Many-to-Many (junction) |
| department\_course | department(department\_id), course(course\_id) | Many-to-Many |
| lecture\_department | lecture(user\_id), department(department\_id) | Many-to-Many |

Table structure of your solution

MariaDB [db\_project]> desc attendance;

+----------------+--------------------------+------+-----+---------+----------------+

| Field | Type | Null | Key | Default | Extra |

+----------------+--------------------------+------+-----+---------+----------------+

| attendance\_id | int(11) | NO | PRI | NULL | auto\_increment |

| student\_id | varchar(10) | NO | MUL | NULL | |

| session\_id | int(11) | NO | MUL | NULL | |

| status | enum('Present','Absent') | NO | | NULL | |

| medical | tinyint(1) | YES | | 0 | |

| hours\_attended | decimal(4,2) | YES | | 0.00 | |

+----------------+--------------------------+------+-----+---------+----------------+

6 rows in set (0.022 sec)

MariaDB [db\_project]> desc course;

+---------------+---------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+---------------+---------------+------+-----+---------+-------+

| course\_id | varchar(10) | NO | PRI | NULL | |

| name | varchar(100) | NO | | NULL | |

| credit | int(11) | NO | | NULL | |

| academic\_year | int(11) | YES | | NULL | |

| semester | enum('1','2') | NO | | NULL | |

| total\_hours | decimal(5,2) | YES | | 50.00 | |

| weekly\_hours | decimal(4,2) | YES | | 3.00 | |

+---------------+---------------+------+-----+---------+-------+

7 rows in set (0.013 sec)

MariaDB [db\_project]> desc dean;

+------------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+------------+-------------+------+-----+---------+-------+

| lecture\_id | varchar(10) | NO | PRI | NULL | |

| term\_start | date | NO | | NULL | |

| term\_end | date | YES | | NULL | |

+------------+-------------+------+-----+---------+-------+

3 rows in set (0.019 sec)

MariaDB [db\_project]> desc department;

+---------------+--------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+---------------+--------------+------+-----+---------+-------+

| department\_id | varchar(10) | NO | PRI | NULL | |

| name | varchar(100) | NO | | NULL | |

| faculty\_name | varchar(100) | YES | | NULL | |

+---------------+--------------+------+-----+---------+-------+

3 rows in set (0.017 sec)

MariaDB [db\_project]> desc department\_course;

+---------------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+---------------+-------------+------+-----+---------+-------+

| department\_id | varchar(10) | NO | PRI | NULL | |

| course\_id | varchar(10) | NO | PRI | NULL | |

+---------------+-------------+------+-----+---------+-------+

2 rows in set (0.010 sec)

MariaDB [db\_project]> desc marks;

+------------------+------------------------------------------------+------+-----+--------------+----------------+

| Field | Type | Null | Key | Default | Extra |

+------------------+------------------------------------------------+------+-----+--------------+----------------+

| marks\_id | int(11) | NO | PRI | NULL | auto\_increment |

| student\_id | varchar(10) | NO | MUL | NULL | |

| course\_id | varchar(10) | NO | MUL | NULL | |

| quiz1\_marks | decimal(5,2) | YES | | NULL | |

| quiz2\_marks | decimal(5,2) | YES | | NULL | |

| quiz3\_marks | decimal(5,2) | YES | | NULL | |

| assessment\_marks | decimal(5,2) | YES | | NULL | |

| mid\_marks | decimal(5,2) | YES | | NULL | |

| final\_theory | decimal(5,2) | YES | | NULL | |

| final\_practical | decimal(5,2) | YES | | NULL | |

| ca\_marks | decimal(5,2) | YES | | NULL | |

| final\_marks | decimal(5,2) | YES | | NULL | |

| ca\_eligible | enum('Eligible','Not Eligible','MC','WH') | YES | | Not Eligible | |

| final\_eligible | enum('Eligible','Not Eligible','MC','WH','E\*') | YES | | Not Eligible | |

| grade | char(10) | YES | | NULL | |

+------------------+------------------------------------------------+------+-----+--------------+----------------+

15 rows in set (0.014 sec)

MariaDB [db\_project]> desc medical;

+----------------+---------------------------------------+------+-----+---------+----------------+

| Field | Type | Null | Key | Default | Extra |

+----------------+---------------------------------------+------+-----+---------+----------------+

| medical\_id | int(11) | NO | PRI | NULL | auto\_increment |

| student\_id | varchar(10) | NO | MUL | NULL | |

| course\_id | varchar(10) | YES | MUL | NULL | |

| exam\_type | enum('Mid','Final','Attendance') | NO | | NULL | |

| date\_submitted | date | NO | | NULL | |

| status | enum('Pending','Approved','Rejected') | YES | | Pending | |

+----------------+---------------------------------------+------+-----+---------+----------------+

6 rows in set (0.012 sec)

MariaDB [db\_project]> desc session;

+---------------+----------------------------+------+-----+---------+----------------+

| Field | Type | Null | Key | Default | Extra |

+---------------+----------------------------+------+-----+---------+----------------+

| session\_id | int(11) | NO | PRI | NULL | auto\_increment |

| course\_id | varchar(10) | NO | MUL | NULL | |

| session\_date | date | NO | | NULL | |

| session\_hours | decimal(4,2) | YES | | 3.00 | |

| type | enum('Theory','Practical') | YES | | Theory | |

+---------------+----------------------------+------+-----+---------+----------------+

5 rows in set (0.015 sec)

MariaDB [db\_project]> desc student;

+---------------+-------------------------------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+---------------+-------------------------------------+------+-----+---------+-------+

| user\_id | varchar(10) | NO | PRI | NULL | |

| reg\_no | varchar(15) | NO | UNI | NULL | |

| batch | varchar(10) | YES | | NULL | |

| status | enum('Proper','Repeat','Suspended') | YES | | Proper | |

| department\_id | varchar(10) | YES | MUL | NULL | |

+---------------+-------------------------------------+------+-----+---------+-------+

5 rows in set (0.017 sec)

MariaDB [db\_project]> desc users;

+----------+----------------------------------------------------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+----------+----------------------------------------------------------+------+-----+---------+-------+

| user\_id | varchar(10) | NO | PRI | NULL | |

| name | varchar(50) | NO | | NULL | |

| email | varchar(50) | NO | UNI | NULL | |

| password | varchar(100) | NO | | NULL | |

| role | enum('Admin','Dean','Lecturer','Tech\_Officer','Student') | NO | | NULL | |

+----------+----------------------------------------------------------+------+-----+---------+-------+

5 rows in set (0.010 sec)

MariaDB [db\_project]> desc tech\_officer;

+---------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+---------+-------------+------+-----+---------+-------+

| user\_id | varchar(10) | NO | PRI | NULL | |

+---------+-------------+------+-----+---------+-------+

1 row in set (0.021 sec)

MariaDB [db\_project]> desc student\_course;

+------------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+------------+-------------+------+-----+---------+-------+

| student\_id | varchar(10) | NO | PRI | NULL | |

| course\_id | varchar(10) | NO | PRI | NULL | |

+------------+-------------+------+-----+---------+-------+

2 rows in set (0.020 sec)

MariaDB [db\_project]> desc result;

+---------------+---------------+------+-----+---------+----------------+

| Field | Type | Null | Key | Default | Extra |

+---------------+---------------+------+-----+---------+----------------+

| result\_id | int(11) | NO | PRI | NULL | auto\_increment |

| student\_id | varchar(10) | NO | MUL | NULL | |

| academic\_year | int(11) | YES | | NULL | |

| semester | enum('1','2') | NO | | NULL | |

| sgpa | decimal(3,2) | YES | | NULL | |

| cgpa | decimal(3,2) | YES | | NULL | |

| total\_credits | int(11) | YES | | 0 | |

+---------------+---------------+------+-----+---------+----------------+

7 rows in set (0.015 sec)

MariaDB [db\_project]> desc session;

+---------------+----------------------------+------+-----+---------+----------------+

| Field | Type | Null | Key | Default | Extra |

+---------------+----------------------------+------+-----+---------+----------------+

| session\_id | int(11) | NO | PRI | NULL | auto\_increment |

| course\_id | varchar(10) | NO | MUL | NULL | |

| session\_date | date | NO | | NULL | |

| session\_hours | decimal(4,2) | YES | | 3.00 | |

| type | enum('Theory','Practical') | YES | | Theory | |

+---------------+----------------------------+------+-----+---------+----------------+

5 rows in set (0.010 sec)

MariaDB [db\_project]> desc lecture \_department;

Empty set (0.011 sec)

MariaDB [db\_project]> desc lecture\_department;

+---------------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+---------------+-------------+------+-----+---------+-------+

| lecture\_id | varchar(10) | NO | PRI | NULL | |

| department\_id | varchar(10) | NO | PRI | NULL | |

+---------------+-------------+------+-----+---------+-------+

2 rows in set (0.009 sec)

MariaDB [db\_project]> desc lecture\_course;

+------------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+------------+-------------+------+-----+---------+-------+

| lecture\_id | varchar(10) | NO | PRI | NULL | |

| course\_id | varchar(10) | NO | PRI | NULL | |

+------------+-------------+------+-----+---------+-------+

2 rows in set (0.020 sec)

MariaDB [db\_project]>

Architecture of your solution

**Three-Tier Architecture:**

1. **Presentation Layer (Frontend):**
   * Web interface or CLI for users to enter/view data.
   * Role-based views: Student, Lecturer, Technical Officer, Dean, Admin.
2. **Application Layer (Business Logic):**
   * **Triggers**: Update attendance, marks, and eligibility automatically.
   * **Stored Procedures**: Calculate CA marks, final marks, grades, GPA/CGPA.
   * **Views**: Summarize attendance, marks, and results for batch or individual student.
3. **Data Layer (Backend Database):**
   * **MySQL** database with normalized tables (3NF).
   * Referential integrity via **foreign keys** and cascading updates/deletes.
   * Role-based access control for different users:
     + Admin: Full privileges
     + Dean: Full without grant
     + Lecturer: CRUD on course/student/marks
     + Technical Officer: CRUD on attendance
     + Student: Read-only on results

**Automation Highlights:**

* Automatic eligibility calculation for attendance & marks
* Medical handling integrated for CA, mid, final exams
* Dynamic GPA/CGPA calculation per semester/year
* Views for easy reporting and analytics

**Tools and Technologies Used**

In developing the **University Academic Management System**, we utilized the following tools and technologies:

1. **MySQL**
   * Relational database management system used as the backend.
   * Provides robust support for tables, relationships, constraints, triggers, stored procedures, and views.
   * Ensures data consistency, integrity, and normalization.
2. **MySQL Workbench**
   * Used for designing the **ER/EER diagrams**, creating database schema, and managing SQL scripts.
   * Supports visualization of relationships, forward engineering, and database maintenance.
3. **Navicat for MySQL**
   * Assisted in database development, data modeling, query execution, and user management.
   * Used for importing/exporting data, backup, and running administrative tasks.
4. **XAMPP**
   * Provided a local server environment to run MySQL and test database operations.
   * Helped integrate and test database connectivity for future backend applications.
5. **Draw.io**
   * Used to create clear and professional **ER/EER diagrams** and relational mapping diagrams.
   * Allowed easy visualization of entities, relationships, and cardinality for the project report.
6. **ChatGPT**
   * Assisted in **design guidance, query optimization, database design review**, and generating code snippets.
   * Used as a reference tool to clarify DBMS concepts and implement efficient database solutions.

## Security measures

**Security Measures Taken to Protect the Database**

To ensure **data integrity, confidentiality, and secure operations** in the University Academic Management System, the following security measures were implemented:

1. **Role-Based Access Control (RBAC)**
   * Users have specific roles with **tailored privileges**:
     + **Admin** – Full privileges with GRANT option
     + **Dean** – Full privileges without GRANT
     + **Lecturer** – CRUD privileges on academic tables but cannot create users
     + **Technical Officer** – CRUD on attendance tables/views only
     + **Student** – Read-only access to final marks and attendance
   * This ensures users only access data relevant to their responsibilities.
2. **User Authentication**
   * Each user has a **unique username and password**.
   * Ensures only authorized users can log in and access the database.
3. **Data Validation and Constraints**
   * Use of **NOT NULL, UNIQUE, ENUM, and CHECK constraints** to prevent invalid entries.
   * Example: Marks are restricted to 0–100; student status is restricted to proper, repeat, or suspended.
4. **Foreign Key Constraints and Referential Integrity**
   * Enforced relationships between tables using **foreign keys with cascading updates/deletes**.
   * Prevents orphaned records and maintains consistent data across the database.
5. **Stored Procedures and Triggers**
   * Automate sensitive operations such as:
     + Calculating attendance percentages
     + Updating eligibility for exams
     + Calculating final marks and grades
   * Minimizes human error and prevents unauthorized manual changes.
6. **Medical & Eligibility Handling**
   * Sensitive academic data like medical exemptions, CA eligibility, and final exam eligibility are **controlled via tables and triggers**.
   * Ensures accurate, secure handling of exceptions.
7. **Principle of Least Privilege**
   * Users only have the minimum permissions needed to perform their tasks.
   * Example: Students cannot update attendance or marks.
8. **Backup and Testing**
   * Regular backups were taken using **Navicat** and **MySQL Workbench**.
   * Testing in **XAMPP** ensured the database operates safely before deployment.

DB accounts /users