

**Department of Information and Communication Technology**

**Faculty of Technology**

**University of Ruhuna**

**Database Management Systems Practicum**

**ICT 1222**

**Assignment 02 – Mini Project**

Group 3

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

Submitted by:

**TG/2021/1055 GHDB Ashen**

**TG/2021/1066 K.A. Priyadarshana**

**TG/2021/1061 W.A. Anjana Kavidu**

**TG/2021/1023 K.S.K. Dilshanjith**

Contents

1. Brief introduction about the problem
2. Brief introduction to the solution
3. Proposed ER/EER diagram
4. Proposed Relational mapping diagram.
5. Table structure of the solution
6. Tools and technologies that we have used.
7. Security measures that we have taken to protect your DB.
8. Brief description about Users
9. Code snippets
10. Problems that we faced during the development of the solution.
11. How you have overcome the above identified problems
12. Where we going to host
13. What are the things, changes, that we must do in your backend?
14. Individual contribution to the backend development
15. References

**Brief introduction about the problem**

The context and specific challenges faced by the Faculty of Technology that led to the need for a student management system are multifaceted and typically stem from the complexity of managing many students, courses, and academic data. Here are some common challenges and context that necessitate a student management information system.

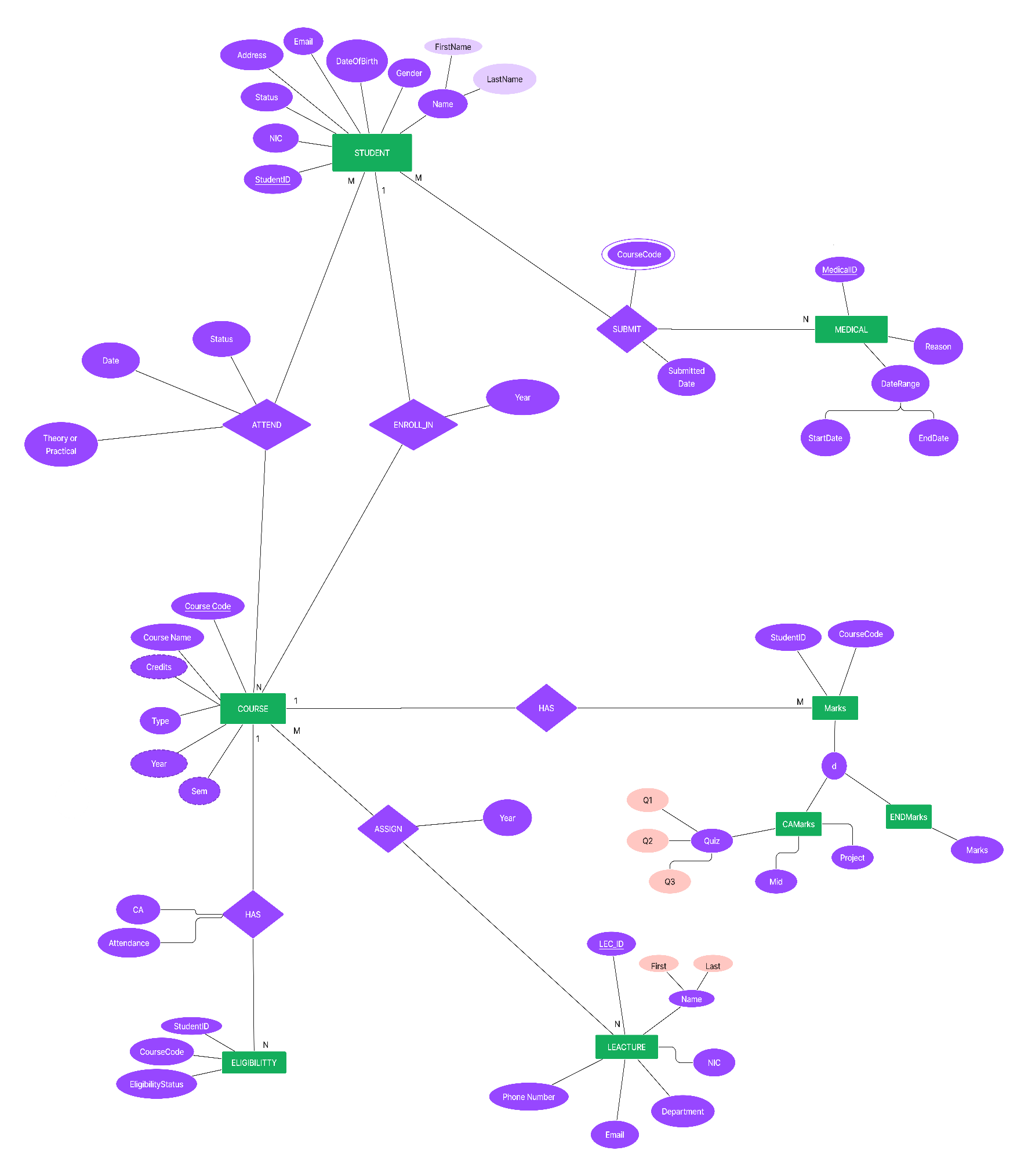
* **Student Enrollment and Records Management**
  + Our faculty deals with a significant number of students enrolling in various courses. Keeping track of student enrollment, personal information, academic progress, and attendance manually can be a daunting task, prone to errors, and time-consuming.
* **Course Management**
  + Managing numerous courses, their details, schedules, and course materials can become overwhelming without a systematic digital solution. Coordinating course offerings, faculty assignments, and tracking changes in course structure requires an efficient system.
* **Attendance Tracking**
  + Recording and monitoring student attendance in each course can be challenging, especially when multiple courses are running concurrently. An automated system can provide real-time attendance tracking, enabling prompt intervention for absent students.
* **Grading and Assessment**
  + Managing student assessment data, including grades for quizzes, assignments, midterms, and final exams, is labor-intensive when done manually. Calculating overall course.
* **Result Calculation and Reporting**
  + Preparing and releasing academic results for students in a timely and accurate manner is crucial. Manual calculations are prone to errors, and students and faculty depend on accurate and efficient systems for result processing.

**Brief introduction to the solution**

The proposed Student Management System is a comprehensive software solution designed to address the specific needs and challenges faced by the Faculty of Technology in managing students, courses, and academic data. This system is a versatile tool that integrates various modules to streamline administrative processes, enhance communication, and provide efficient data management for students and faculty.

**Key Features and Benefits of the Student Management System**

* Student Information Management
* Course Management
* Attendance Tracking
* Grading and Assessment
* Result Management

**Proposed ER/EER diagram**

**Proposed Relational mapping diagram**

A screenshot of a computer

Description automatically generated

**Table structure of the solution**

This provides an overview of the field names, data types, and relationships for each of the tables created.

Tables with the columns, data types, and lengths

**Departments**

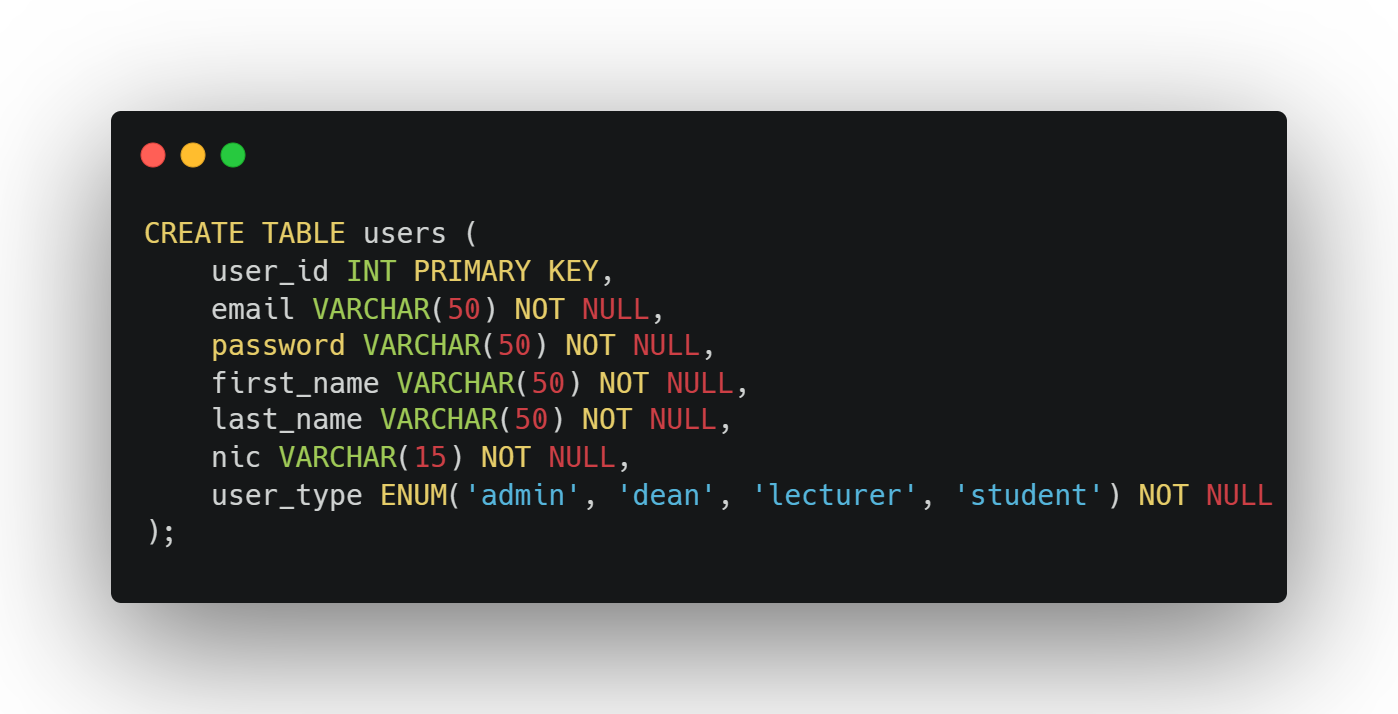
|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Data Type | Length | Extra |
| id | INT |  | PRI, AUTO\_INCREMENT |
| Title | VARCHAR | 100 |  |

A black rectangular with yellow and green text

Description automatically generated

**Users**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Data Type | Length | Extra |
| user\_id | INT |  | PRI |
| email | VARCHAR | 50 | NOT NULL |
| Password | VARCHAR | 50 | NOT NULL |
| first\_name | VARCHAR | 50 | NOT NULL |
| last\_name | VARCHAR | 50 | NOT NULL |
| Nic | VARCHAR | 15 | NOT NULL |
| user\_type | ENUM |  | NOT NULL |



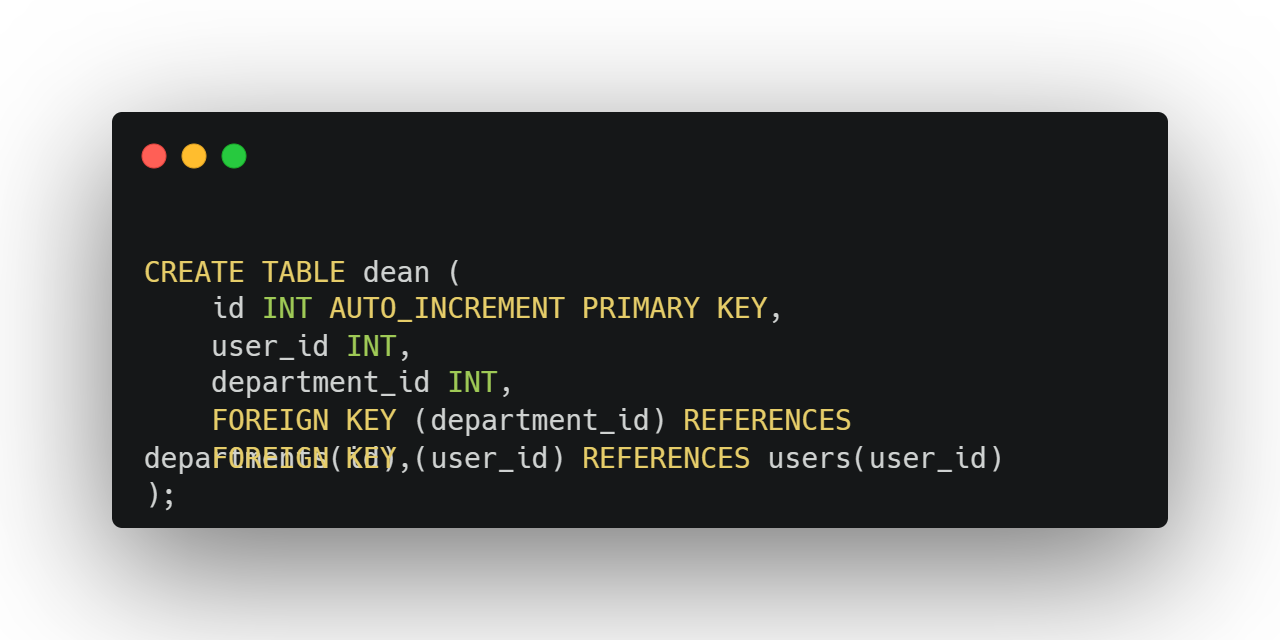
**admin**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Data Type | Length | Extra |
| admin\_id | INT |  | PRI, AUTO\_INCREMENT |
| user\_id | INT |  | FOREIGN KEY |



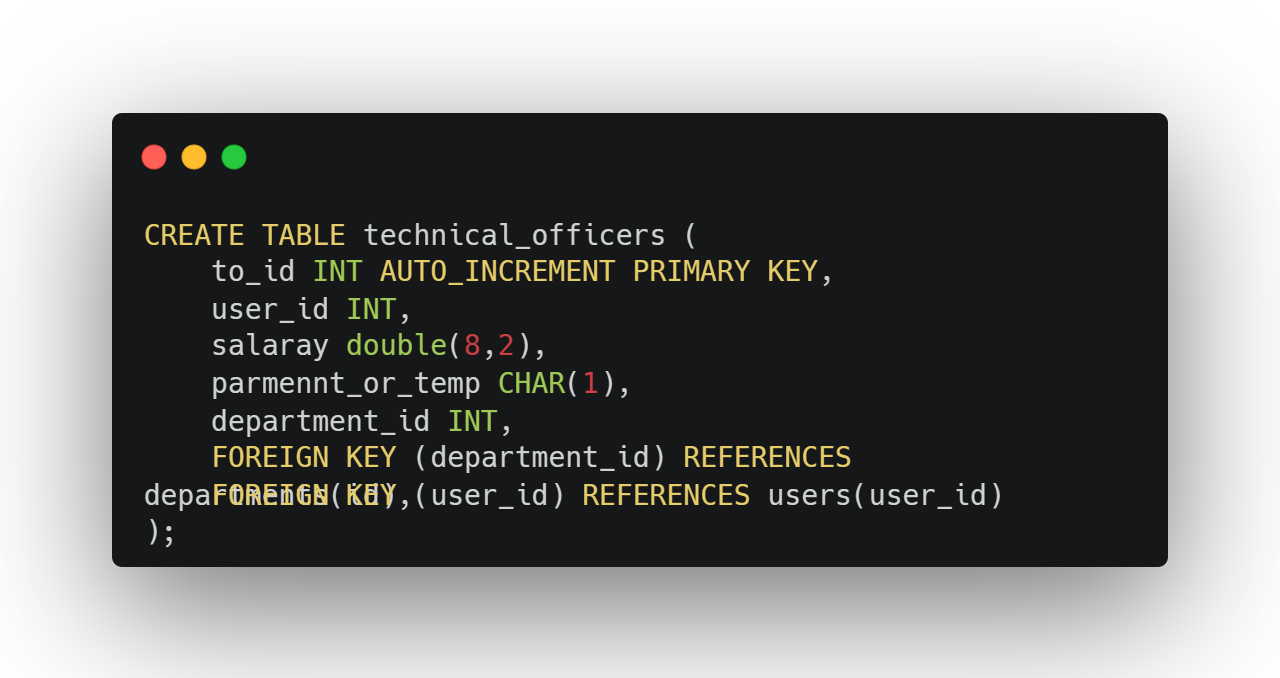
**dean**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Data Type | Length | Extra |
| id | INT |  | PRI, AUTO\_INCREMENT |
| user\_id | INT |  | FOREIGN KEY |
| department\_id | INT |  | FOREIGN KEY |

****

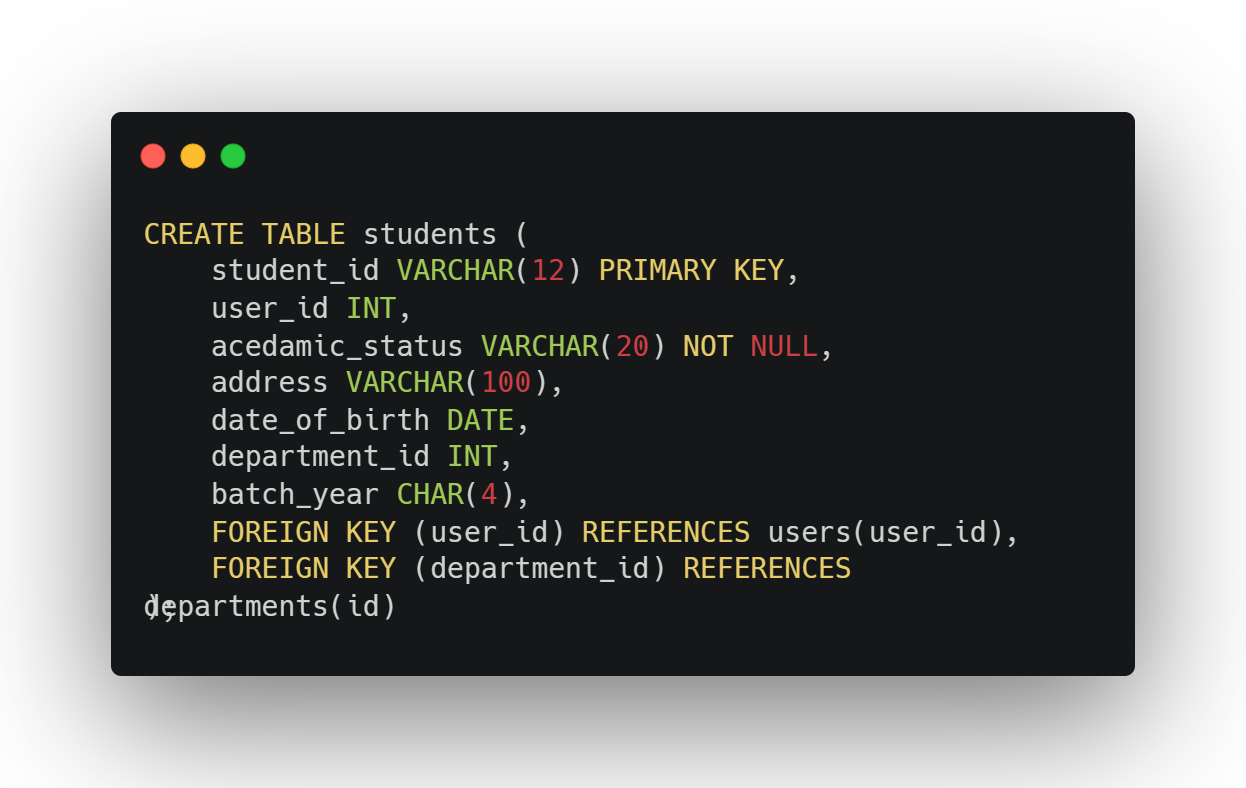
**technical\_officers**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Data Type | Length | Extra |
| to\_id | INT |  | PRI, AUTO\_INCREMENT |
| user\_id | INT |  | FOREIGN KEY |
| Salary | DOUBLE | 8,2 |  |
| parmennt\_or\_temp | CHAR | 1 |  |
| department\_id | INT |  | FOREIGN KEY |



**Students**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Data Type | Length | Extra |
| student\_id | VARCHAR | 12 | PRI |
| user\_id | INT |  | FOREIGN KEY |
| acedamic\_status | VARCAHR | 20 | NOT NULL |
| address | VARCAHR | 100 |  |
| date\_of\_birth | DATE |  |  |
| department\_id | INT |  | FOREIGN KEY |
| batch\_year | CHAR | 4 |  |



**lecturer**

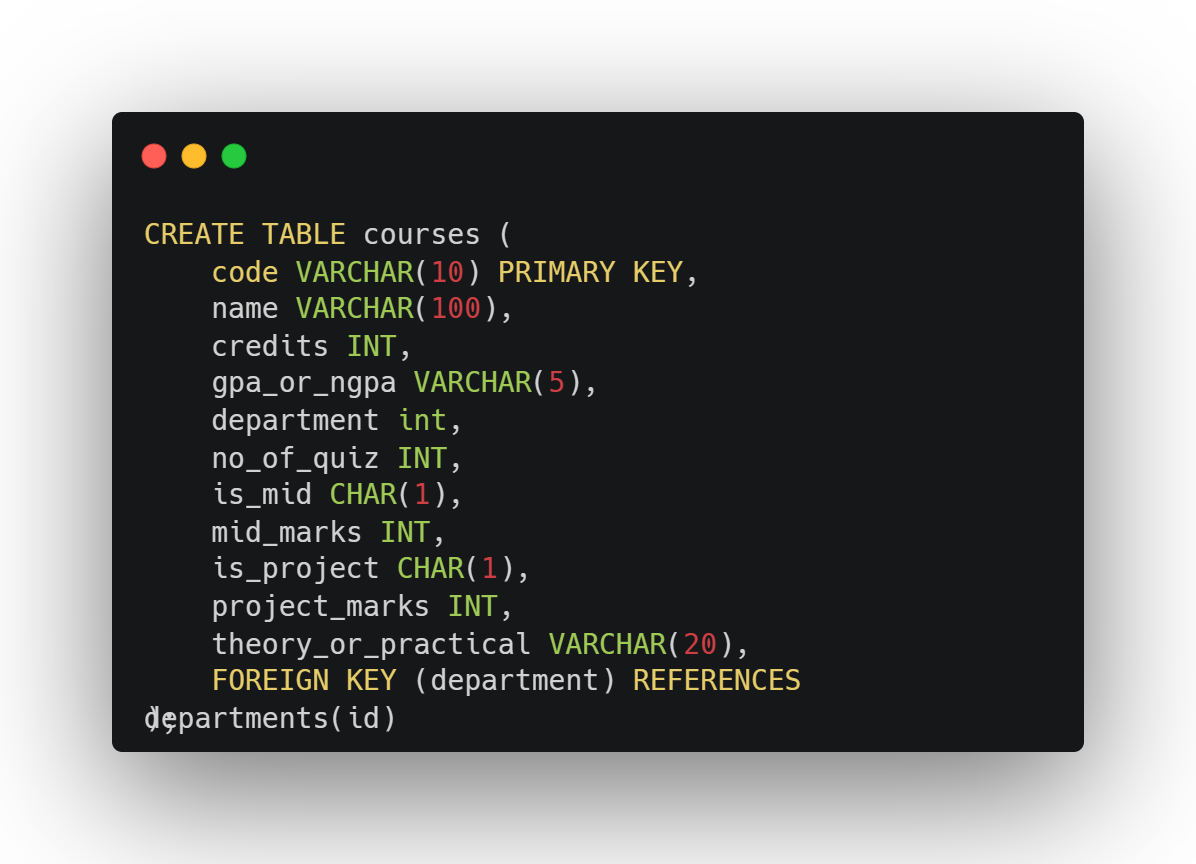
|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Data Type | Length | Extra |
| id | VARCHAR | 7 | PRI |
| user\_id | INT |  | FOREIGN KEY |
| phone\_number | VARCAHR | 15 |  |
| department\_id | INT |  | FOREIGN KEY |
| Email | VARCHAR | 100 |  |

A screen shot of a computer code

Description automatically generated

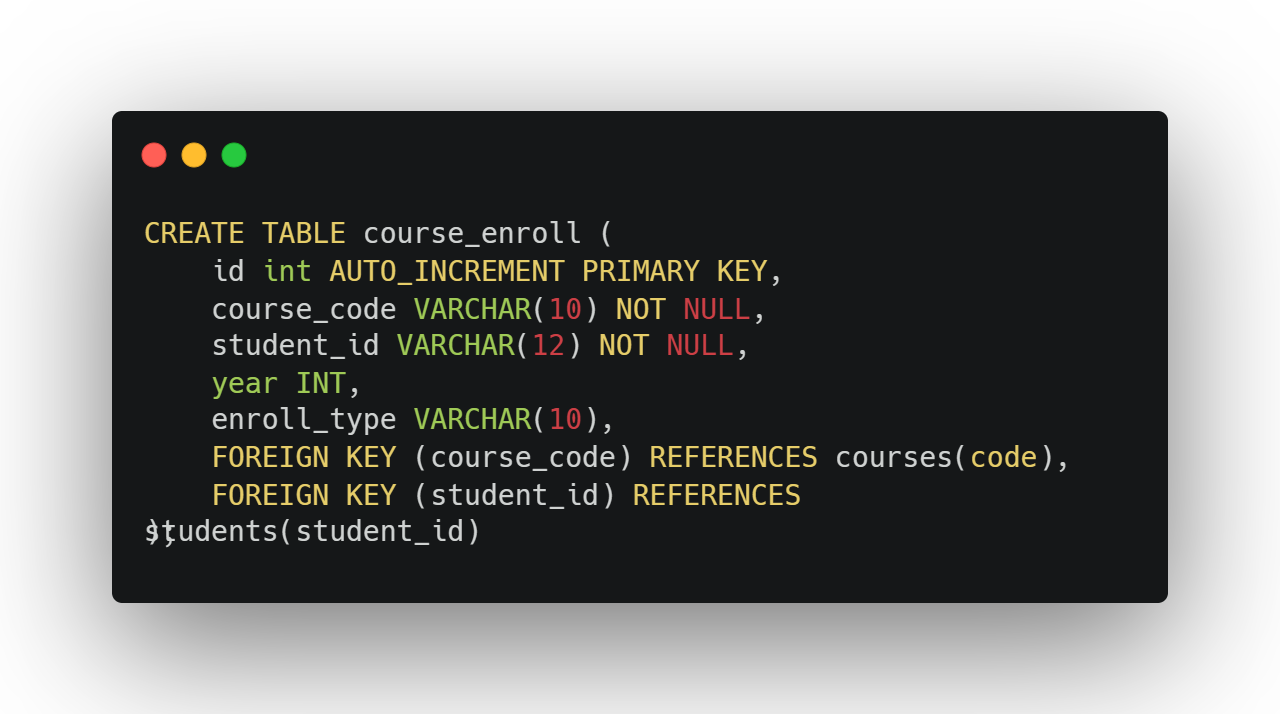
**courses**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Data Type | Length | Extra |
| Code | VARCHAR | 10 | PRI |
| Name | VARCHAR | 100 |  |
| Credits | INT |  |  |
| gpa\_or\_ngpa | VARCHAR | 5 |  |
| Department | INT |  | FOREIGN KEY |
| no\_of\_quiz | INT |  |  |
| is\_mid | CHAR | 1 |  |
| mid\_marks | INT |  |  |
| is\_project | CHAR | 1 |  |
| project\_marks | INT |  |  |
| theory\_or\_practical | VARCHAR | 20 |  |



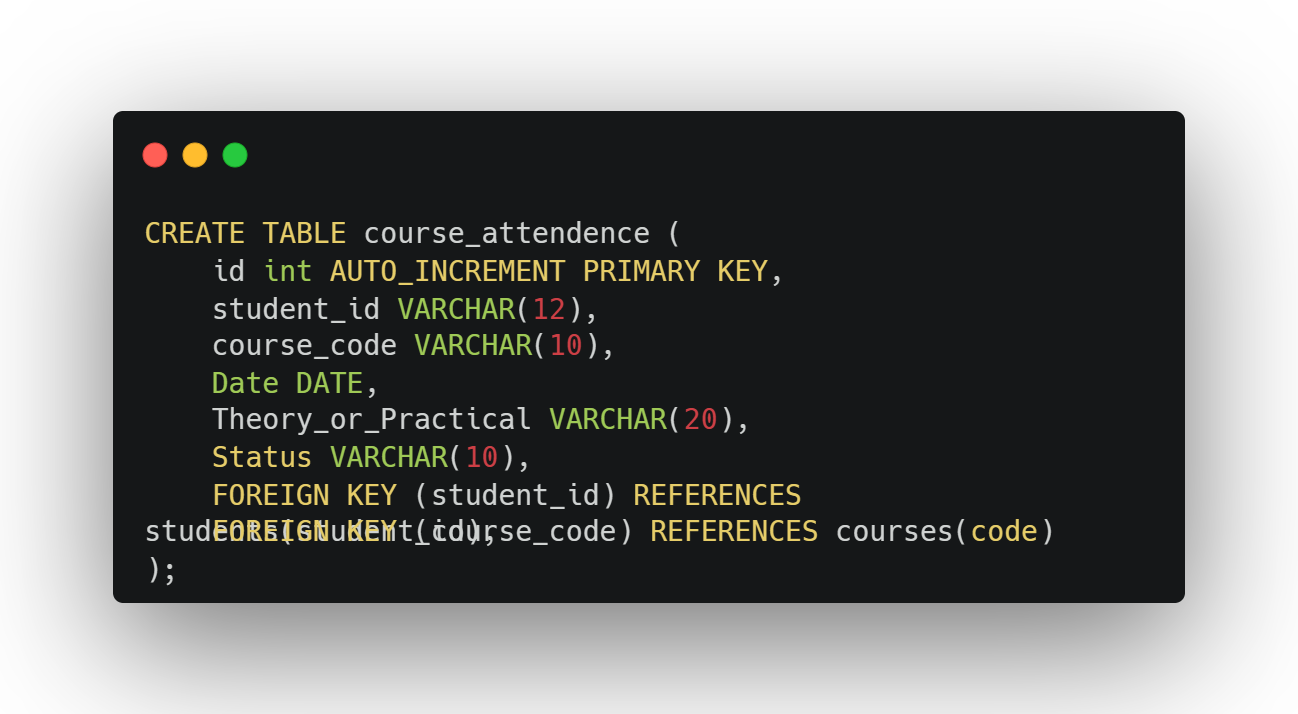
**course\_enroll**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Data Type | Length | Extra |
| Id | INT |  | PRI, AUTO\_INCREMENT |
| course\_code | VARCHAR | 10 | NOT NULL, FOREIGN KEY |
| student\_id | VARCHAR | 12 | NOT NULL, FOREIGN KEY |
| Year | INT |  |  |
| enroll\_type | VARCHAR | 10 |  |



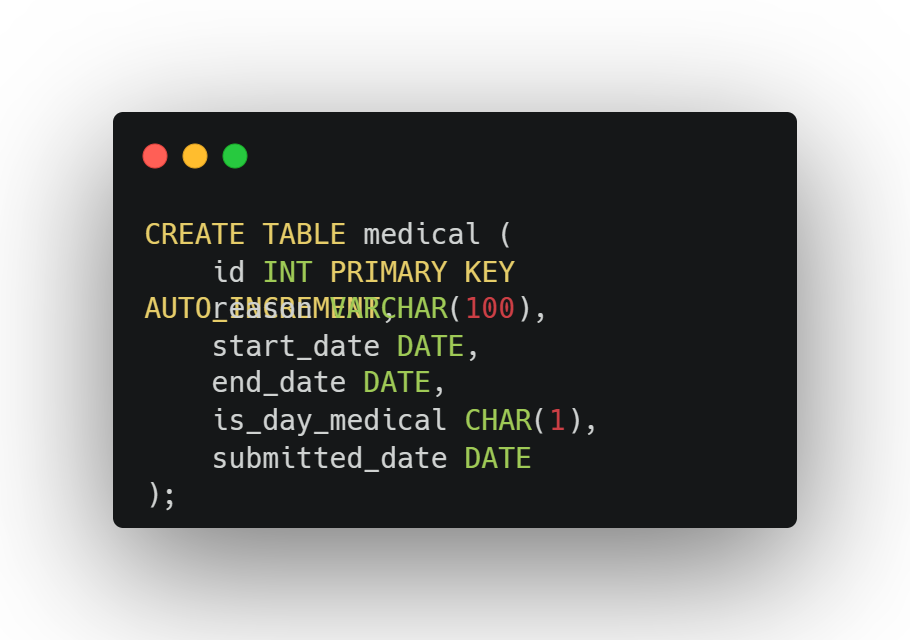
**course\_attendence**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Data Type | Length | Extra |
| Id | INT |  | PRI, AUTO\_INCREMENT |
| student\_id | VARCHAR | 12 | FOREIGN KEY |
| course\_code | VARCHAR | 10 | FOREIGN KEY |
| Datae | DATE |  |  |
| Theory\_or\_Practical | VARCHAR | 20 |  |
| Status | VARCHAR | 10 |  |



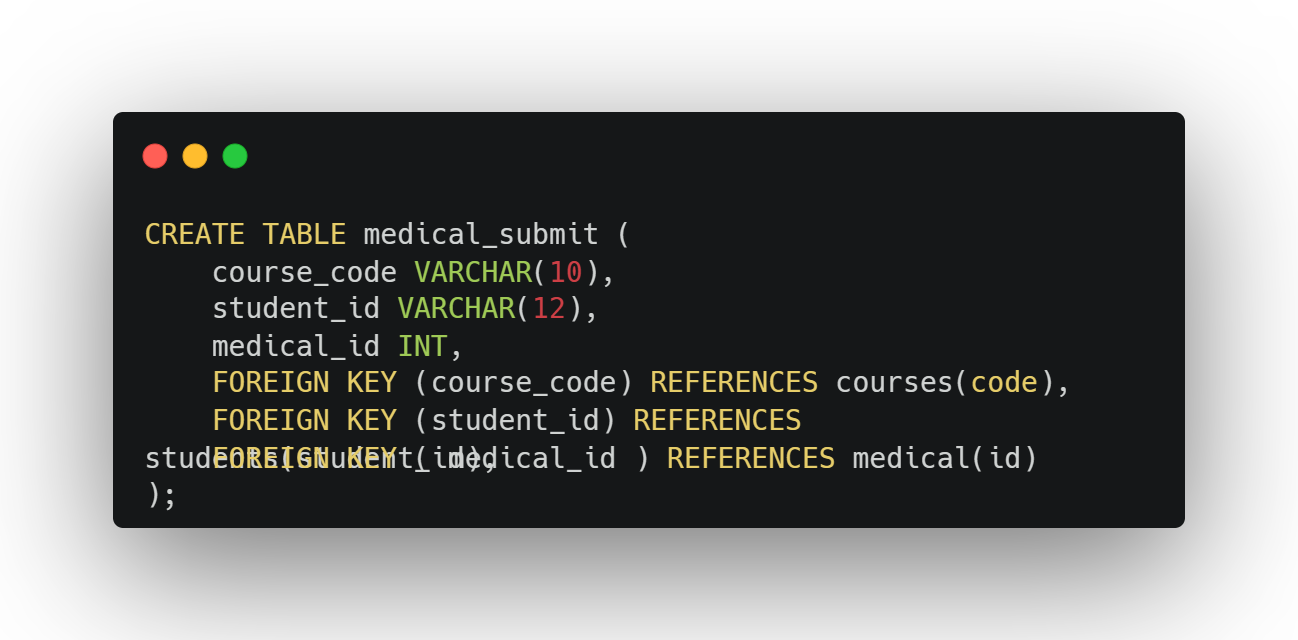
**Medical**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Data Type | Length | Extra |
| Id | INT |  | PRI, AUTO\_INCREMENT |
| Reason | VARCHAR | 100 |  |
| Start\_date | DATE |  |  |
| End\_date | DATE |  |  |
| is\_day\_medical | CHAR | 1 |  |
| submitted\_date | DATE |  |  |



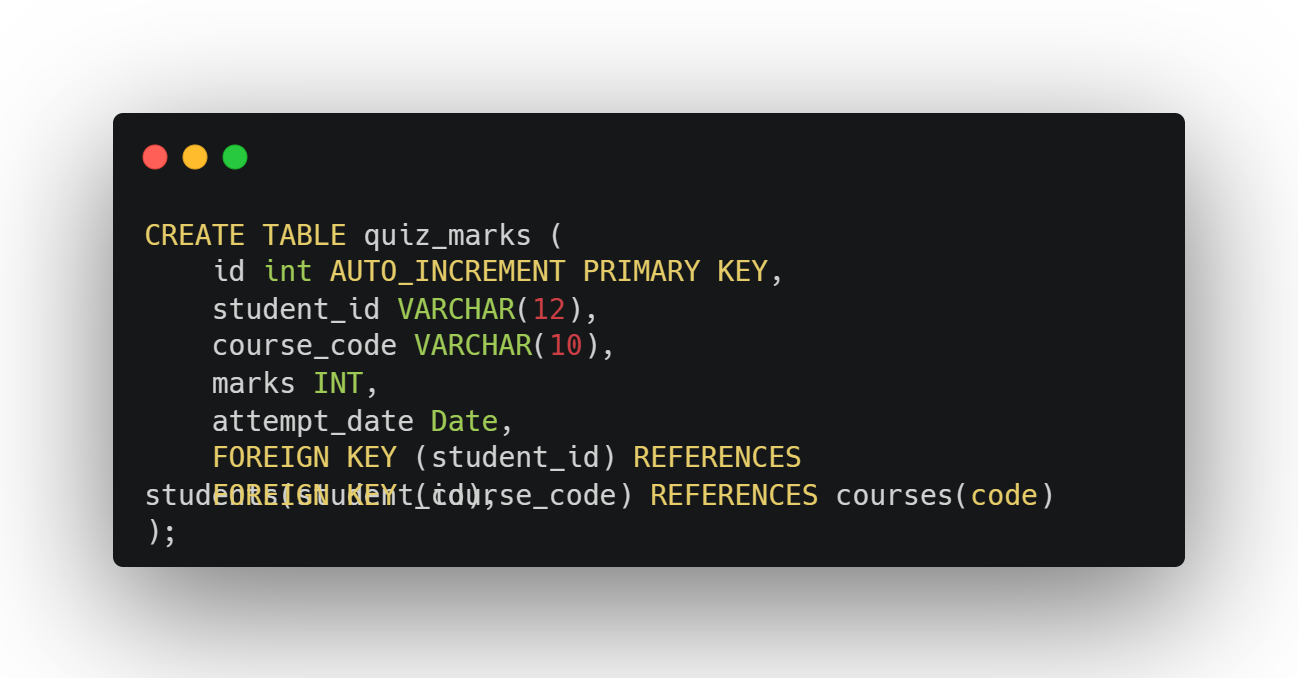
**medical\_submit**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Data Type | Length | Extra |
| course\_code | VARCHAR | 10 | FOREIGN KEY |
| student\_id | VARCHAR | 12 | FOREIGN KEY |
| medical\_id | INT |  | FOREIGN KEY |



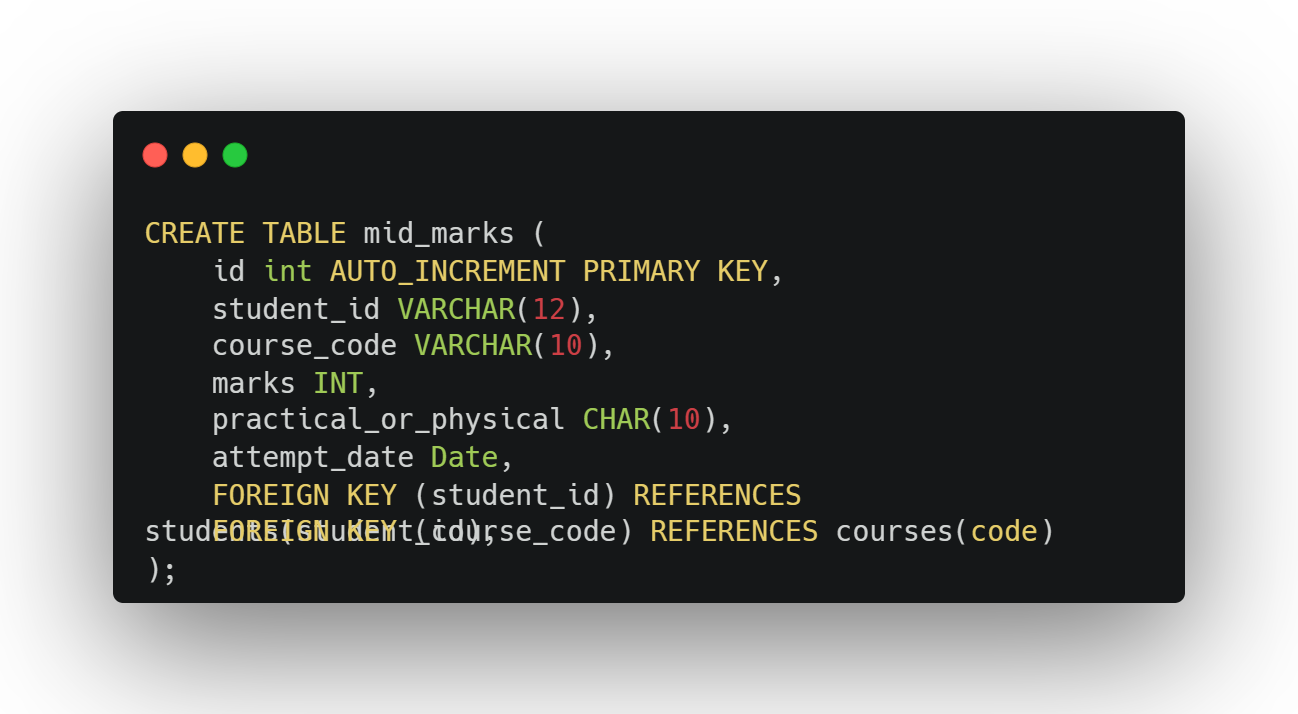
**quiz\_marks**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Data Type | Length | Extra |
| Id | INT |  | PRI, AUTO\_INCREMENT |
| student\_id | VARCHAR | 12 | FOREIGN KEY |
| course\_code | VARCHAR | 10 | FOREIGN KEY |
| Marks | INT |  |  |
| attempt\_date | DATE |  |  |



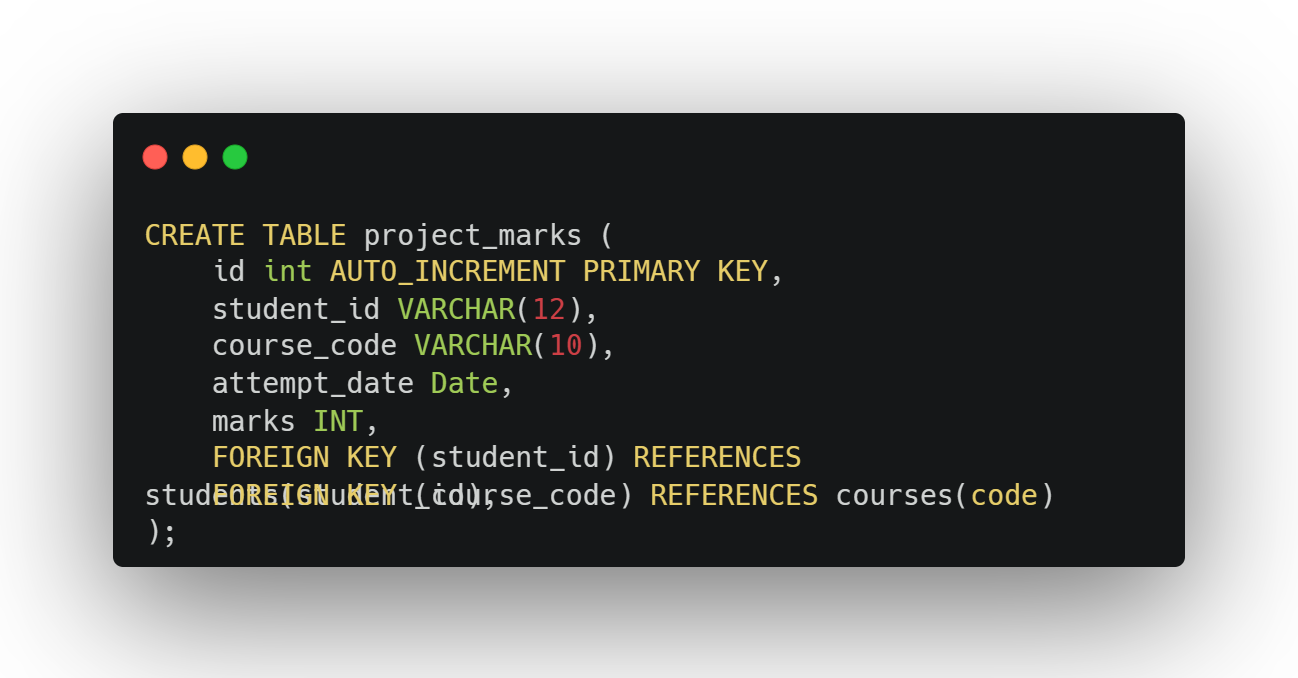
**mid\_marks**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Data Type | Length | Extra |
| Id | INT |  | PRI, AUTO\_INCREMENT |
| student\_id | VARCHAR | 12 | FOREIGN KEY |
| course\_code | VARCHAR | 10 | FOREIGN KEY |
| Marks | INT |  |  |
| practical\_or\_physical | CHAR | 10 |  |
| attempt\_date | DATE |  |  |



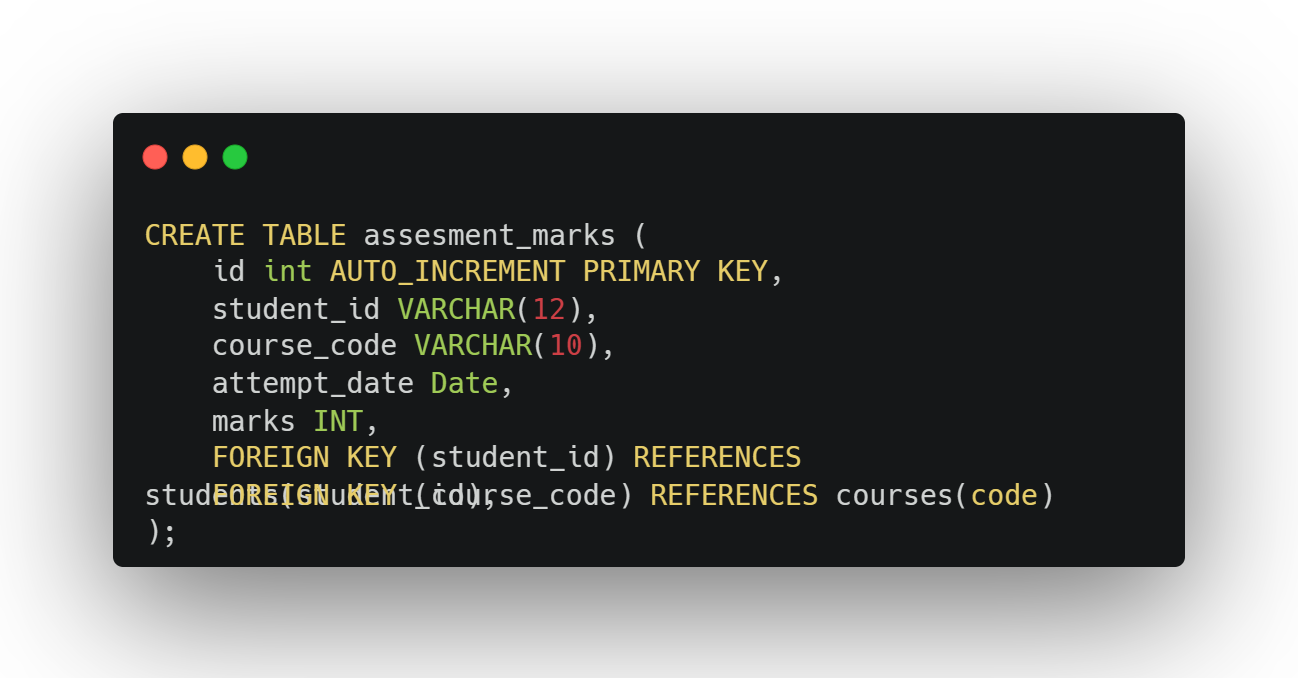
**project\_marks**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Data Type | Length | Extra |
| Id | INT |  | PRI, AUTO\_INCREMENT |
| student\_id | VARCHAR | 12 | FOREIGN KEY |
| course\_code | VARCHAR | 10 | FOREIGN KEY |
| attempt\_date | DATE |  |  |
| Marks | INT | 1 |  |



**assesment\_marks**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Data Type | Length | Extra |
| Id | INT |  | PRI, AUTO\_INCREMENT |
| student\_id | VARCHAR | 12 | FOREIGN KEY |
| course\_code | VARCHAR | 10 | FOREIGN KEY |
| attempt\_date | DATE |  |  |
| Marks | INT |  |  |



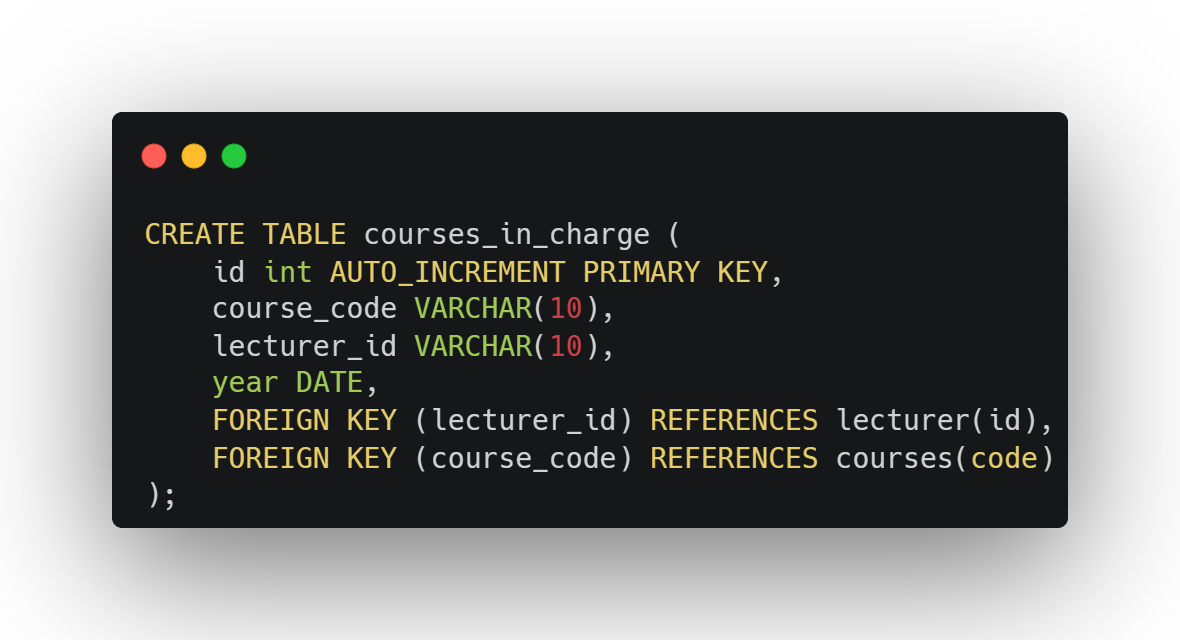
**final\_marks**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Data Type | Length | Extra |
| student\_id | VARCHAR | 12 | FOREIGN KEY |
| course\_code | VARCHAR | 10 | FOREIGN KEY |
| ca\_mark | INT |  |  |
| end\_mark | INT |  |  |
| total\_marks | INT |  |  |
| Grade | CHAR | 4 |  |
| updated\_at | DATE |  |  |



**courses\_in\_charge**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Data Type | Length | Extra |
| Id | INT |  | PRI, AUTO\_INCREMENT |
| course\_code | VARCHAR | 10 | FOREIGN KEY |
| lecturer\_id | VARCHAR | 10 | FOREIGN KEY |
| Year | DATE |  |  |



**Architecture of solution**

**assesment\_marks**

A screen shot of a computer

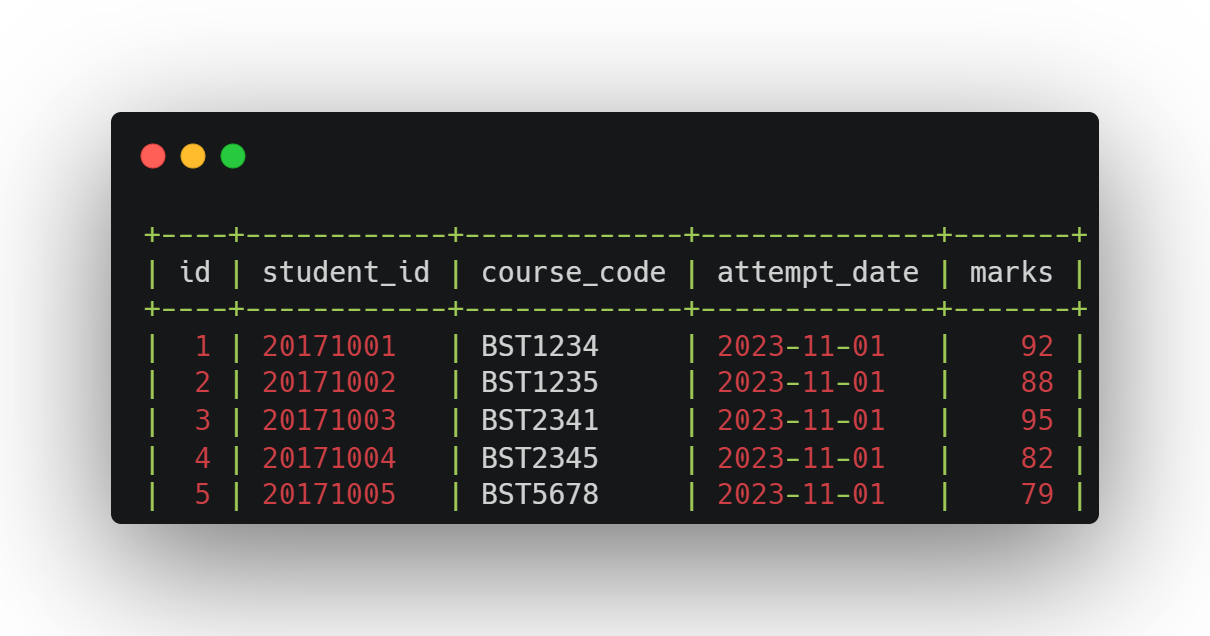
Description automatically generated

**attendance\_summary\_2022**

A screen shot of a computer

Description automatically generated

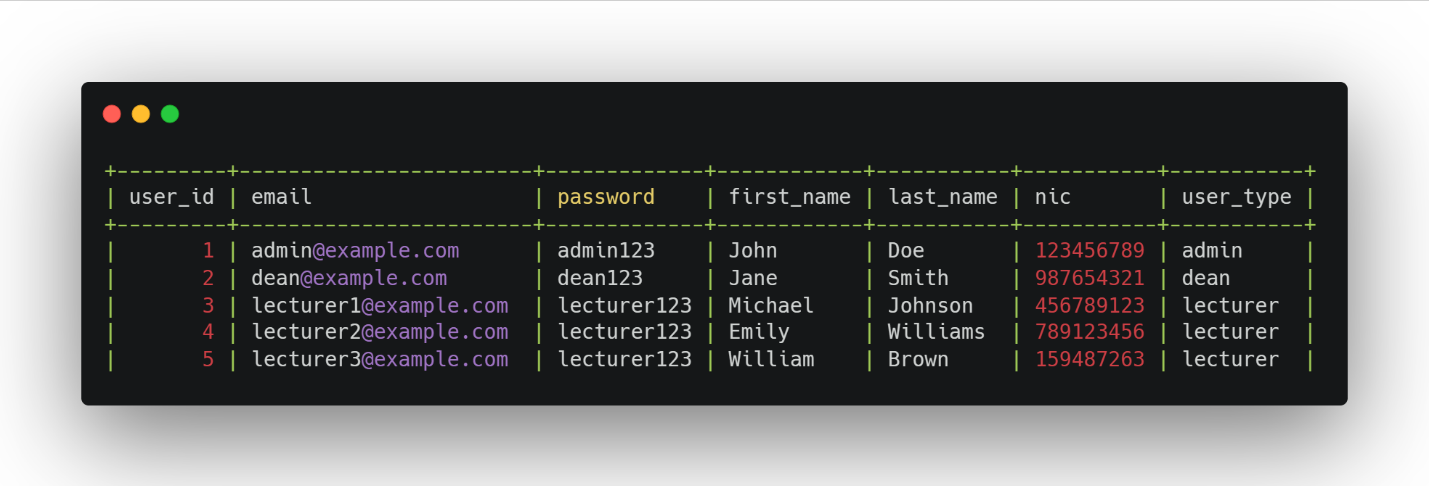
**course\_attendence**

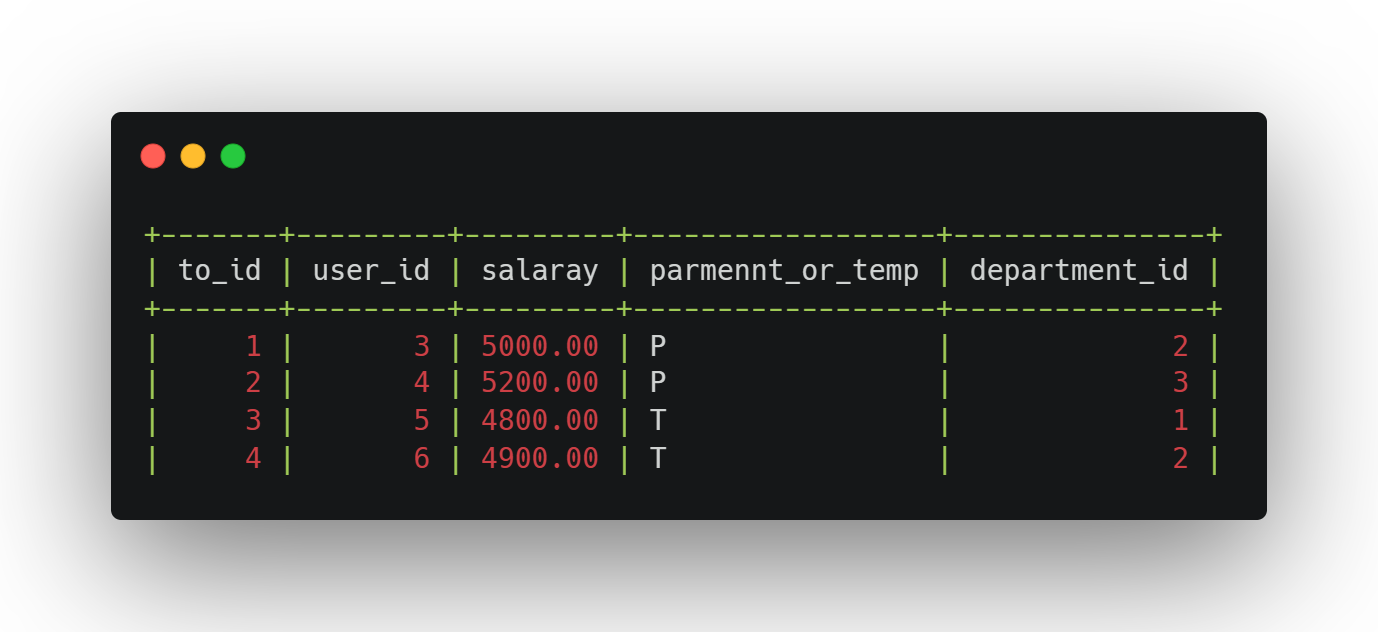


**Course\_enroll**

**A screen shot of a computer program

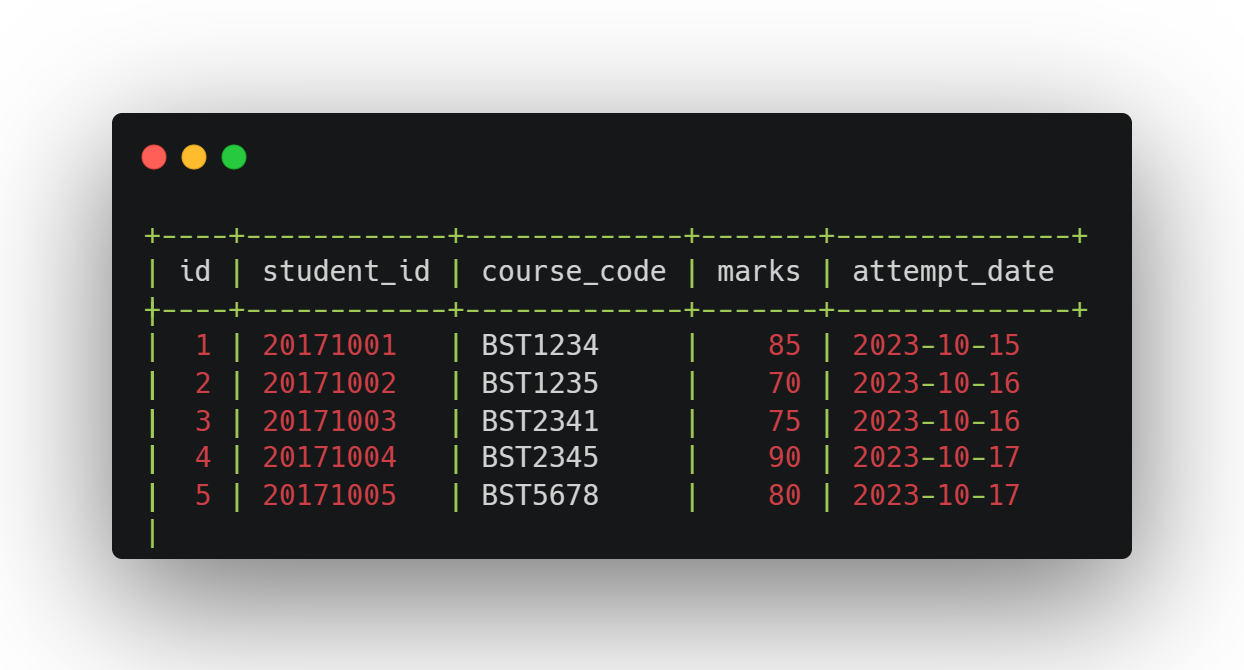
Description automatically generated**

**Users**

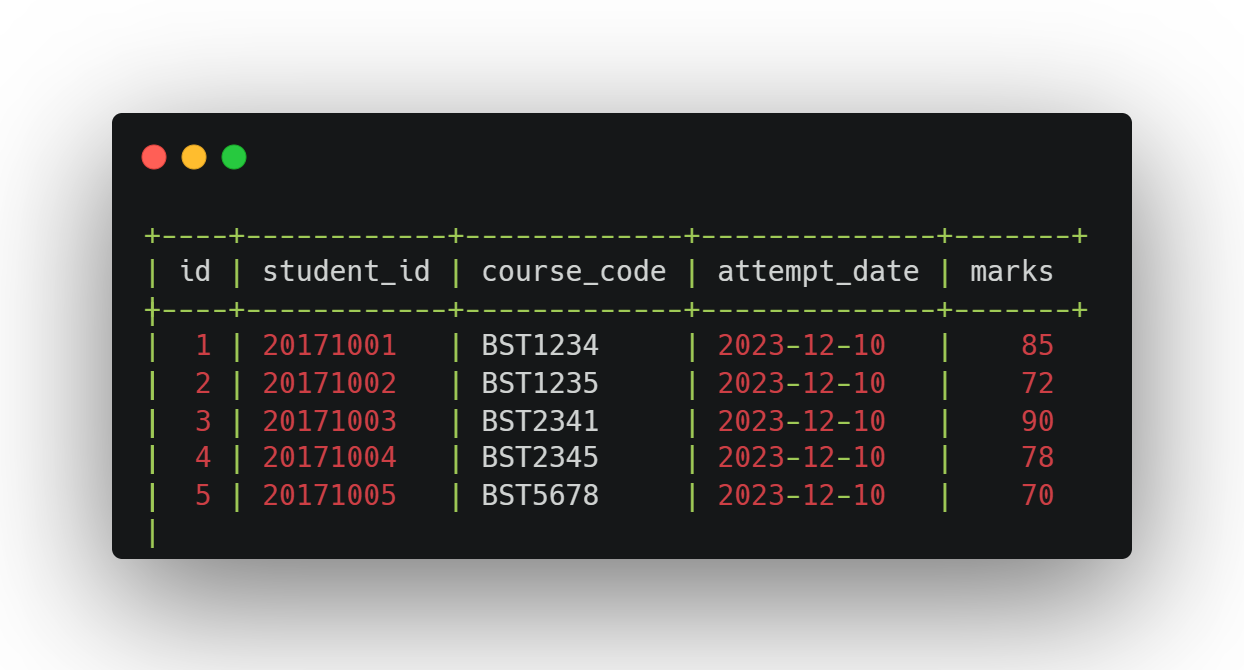
**technical\_officers**

**students**

****

**quiz\_marks**

**project\_marks**

****

**A black screen with white text

Description automatically generatedmid\_marks**

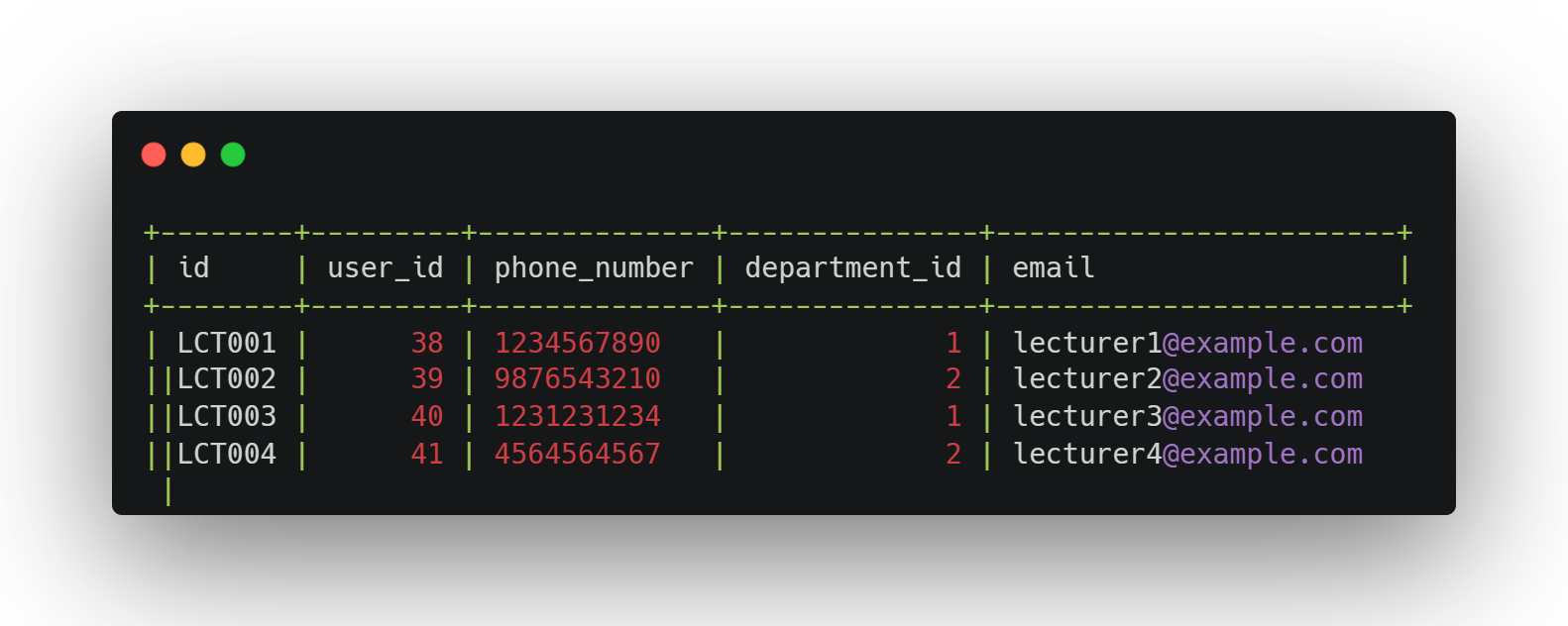
**medical\_submit**

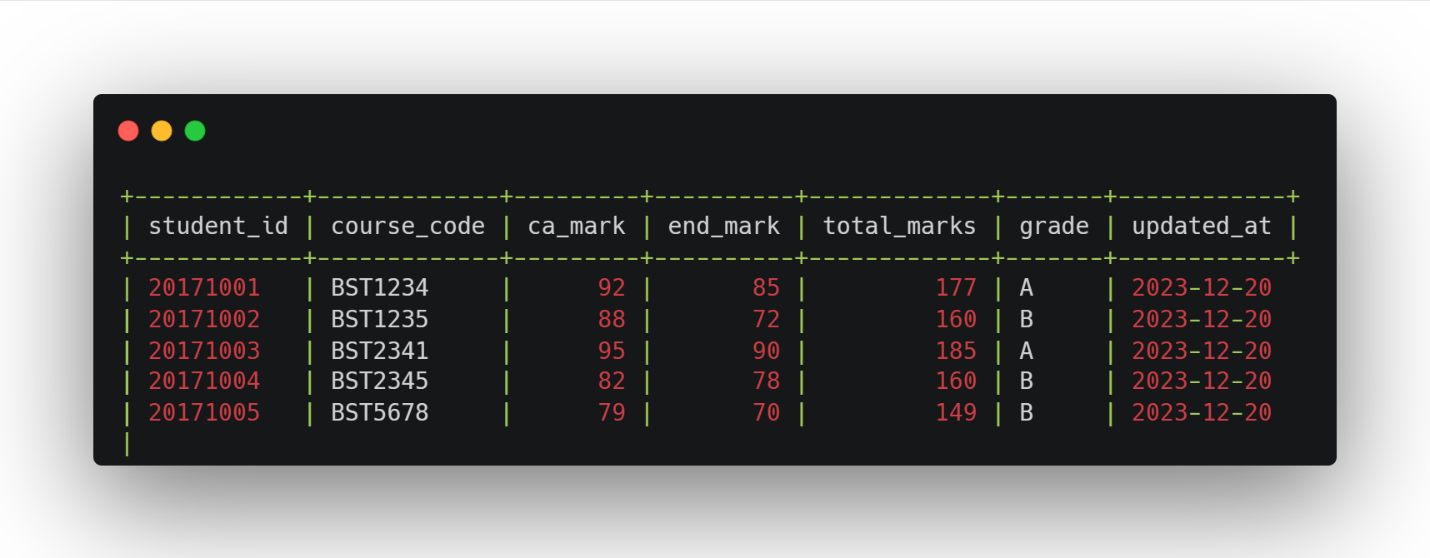
****

**A screen shot of a computer

Description automatically generatedmedical**

**lecturer**

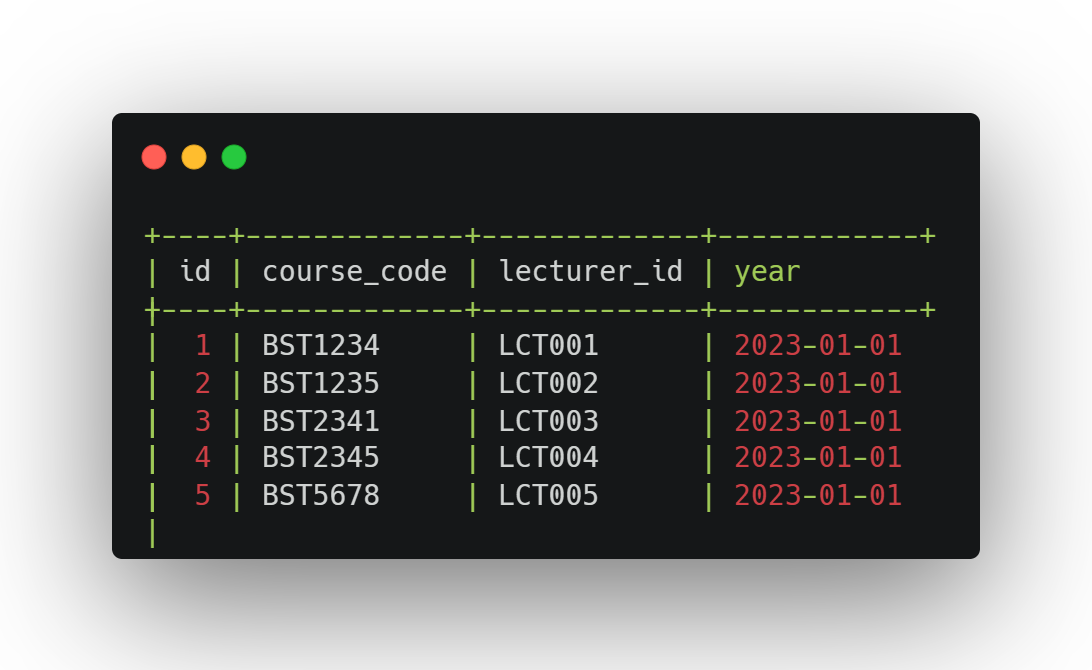
****

**final\_marks**

**departments**

**A screen shot of a computer

Description automatically generateddean**

**courses\_in\_charge**

**A screen shot of a black screen

Description automatically generatedcourses**

**A screen shot of a computer program

Description automatically generatedcourse\_enroll**

**Tools and technologies that we have used.**

In the development of our student management system project, we have used lot of tools and technologies to ensure efficient collaboration, code development, and database management.

* **Cloud Machine for MySQL Server**

We employed cloud-based infrastructure to host our MySQL database server. Our database was hosted in a virtual machine on a cloud platform, ensuring that it remained. This enabled our team to interact with the database.

* **SSH (Secure Shell) for Remote Access**

To connect with our cloud-based MySQL server, we use SSH. SSH provided us with encrypted and secure access to the virtual machine where the database was hosted.

* **MySQL Command-Line Interface (CLI)**

MySQL CLI is one of the primary tools for interacting with the database. It provided us with a powerful command-line interface to execute SQL queries, create and modify database structures, and manage the data efficiently.

* **Visual Studio Code (VS Code)**

As IDE, Visual Studio Code provides central hub for our coding efforts. We used VS Code to write the project’s source code. It provides extensibility and version control with different extensions.

* **MySQL Workbench**

MySQL Workbench offered a user-friendly GUI environment.

* **GitHub**

To effectively manage our source code, track changes, and facilitate collaboration, we used GitHub, a web-based platform for version control and code hosting. GitHub allowed us to work on the project simultaneously, merge code changes, and maintain a comprehensive history of our development efforts.

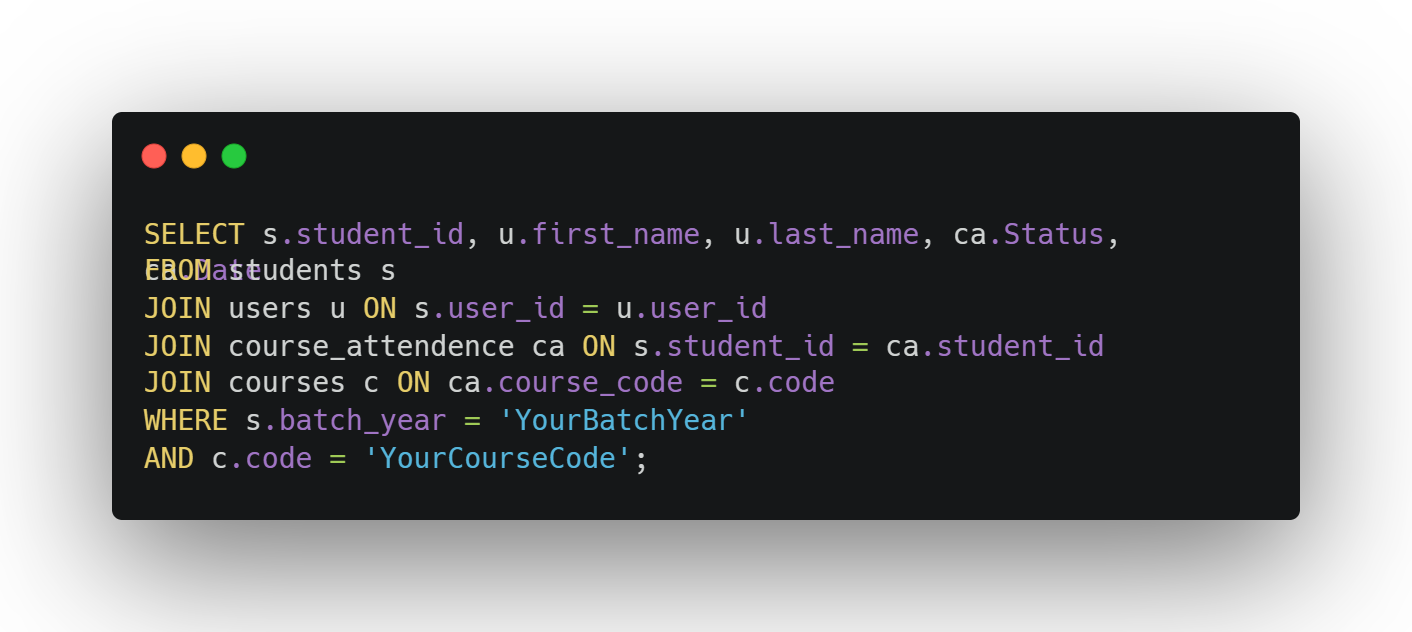
**Security measures that we have taken to protect your DB.**

We have set up a multi-user environment for database access, where each user is granted, specific permissions based on their role and responsibilities. This ensures that users can only access, modify, or view the data that is relevant to their tasks, significantly reducing the risk of unauthorized data breaches.

**Brief description about Users**

* **Admin**
  + Create, maintain, and delete courses.
  + Create, maintain, and delete user profiles.
  + Grant, maintain, and delete user permissions.
* **Lecturer**
  + Manage course materials, assignments, and assessments.
  + Create and upload course content, including lecture notes, assignments, and quizzes.
  + Monitor student progress, grade assignments, and interact with students.
  + Access and manage data related to their courses and students.
* **Technical officer**
  + Maintaining the technical infrastructure of the database system.
  + Include ensuring the smooth operation of the system, addressing technical issues, and providing technical support to users.
  + Maintenance tasks, update software, and troubleshoot technical problems.
* **Student**
  + Can learning and coursework.
  + Access course materials, submit assignments, and take assessments.
  + Can view their course schedules, grades, and interact with their instructors and peers.
  + Track their academic progress, participate in discussions, and access the resources they need.

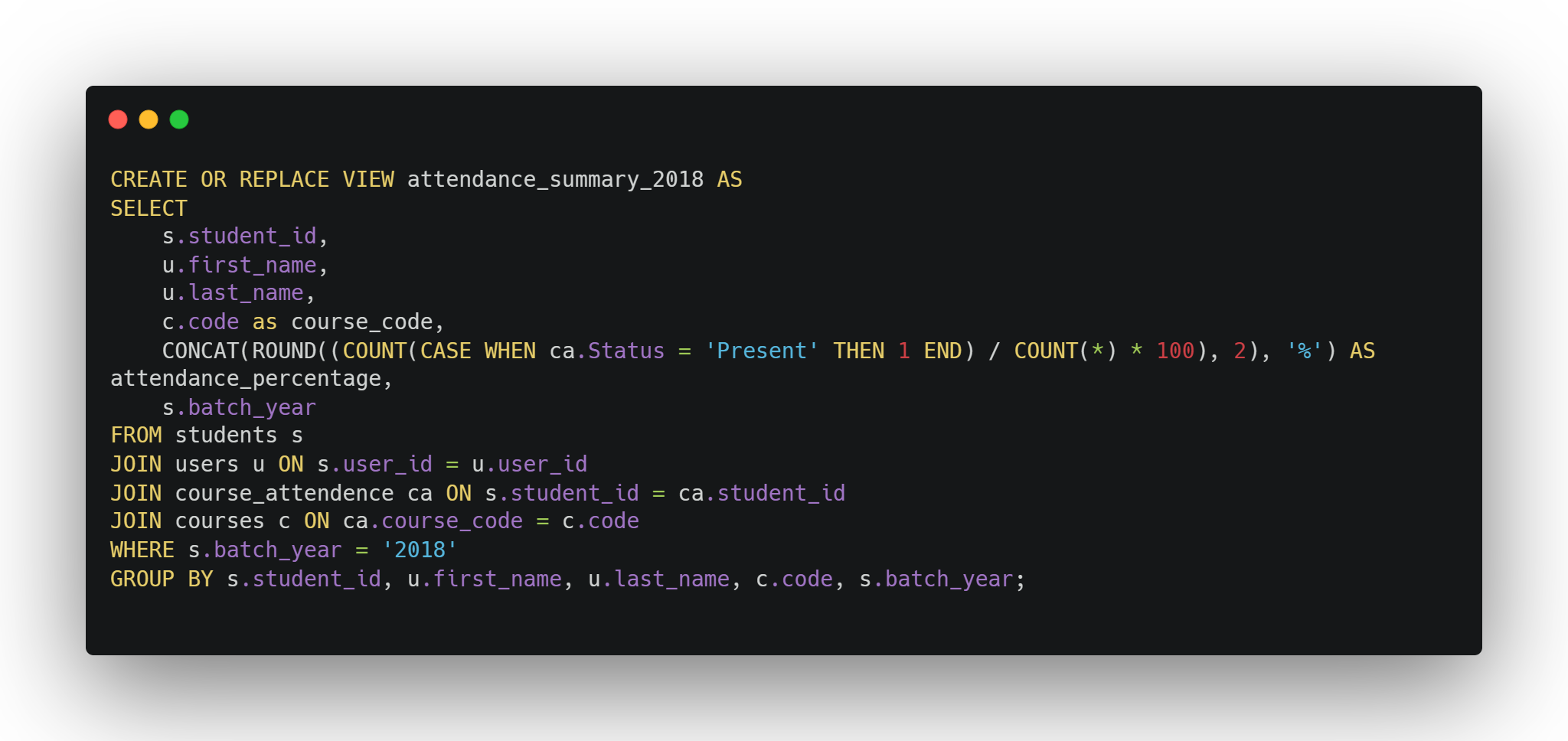
**A black screen with colorful text

Description automatically generated****Code snippets**

**Attendance summary**

A screen shot of a computer program

Description automatically generated

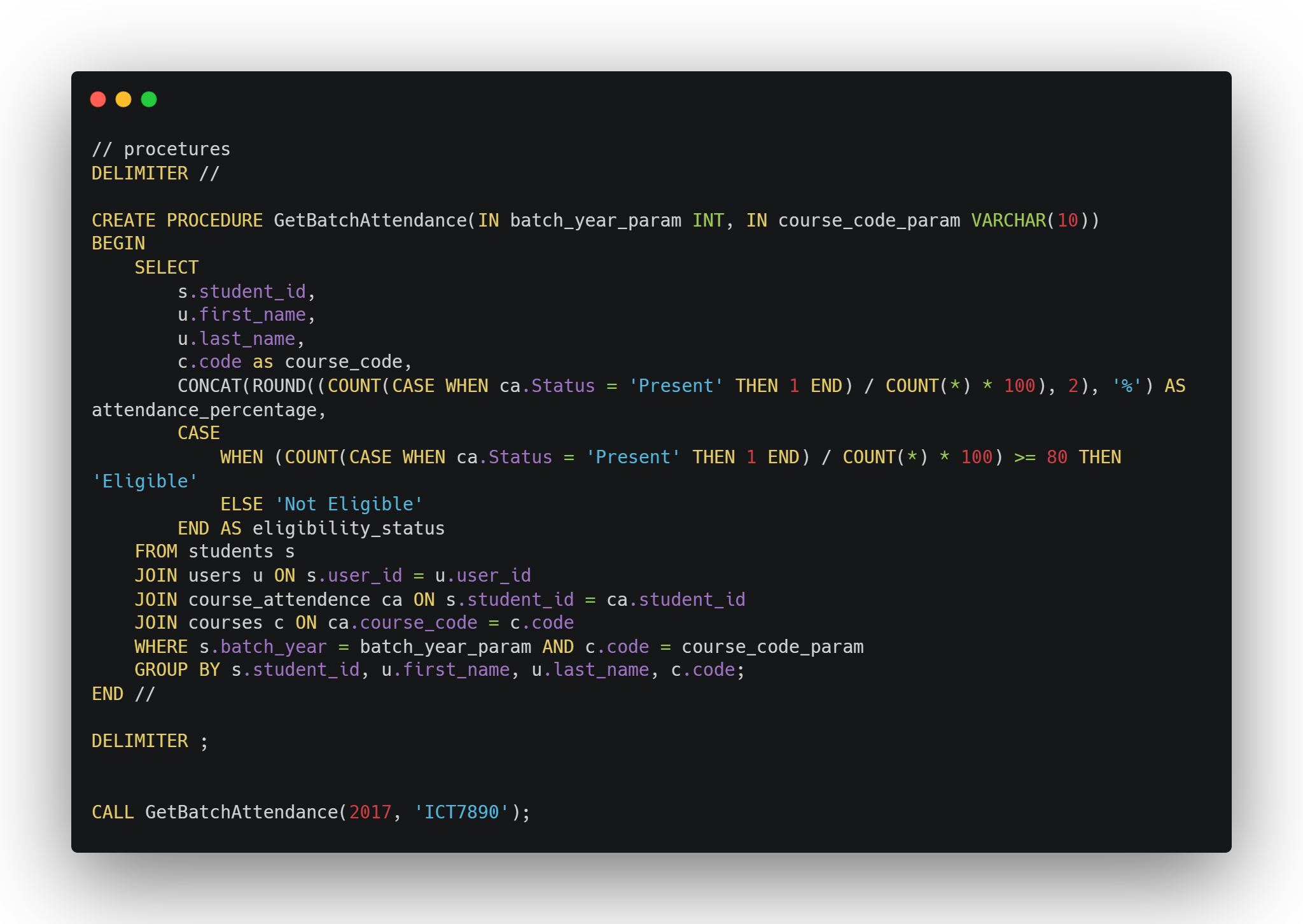
****

**A screen shot of a computer program

Description automatically generated**

**A screen shot of a computer program

Description automatically generated**

****

**Problems that we faced during the development of the solution.**

Regarding the challenges we encountered, we encountered several issues. We managed to devise solutions for some of them, while others remained unresolved. Here are the problems we confronted.

* Text Editor Selection
  + The primary challenge revolved around selecting a text editor environment that was user-friendly and conducive to efficient coding.
* Database Structure Planning
  + Another hurdle was deciding on the optimal structure for our database, defining how it should be organized and related.
* Table Attribute and Definitions
  + The following challenge was deciding on the appropriate attributes and data types to employ for our database tables.
* Compatibility Issues with MySQL Versions
  + We encountered difficulties due to variations in MySQL versions. Some older MySQL code was incompatible with our newer MySQL versions.
* Indentation Errors
  + An issue that arose was the occurrence of simple errors related to indentation marks.
* Problems with VS Code During Procedure Creation
  + We faced challenges with the functionality of VS Code when attempting to create procedures.

**How you have overcome the above identified problems**

* Find good editor.
* Dealing with the text editor challenge, we recognized that while there are numerous text editors available for programming languages, the options for MySQL are more limited. After careful consideration, we opted for Visual Studio Code as our coding platform. We further enhanced its functionality by incorporating a MySQL database plugin, allowing us to seamlessly connect to our MySQL database created using the command line (CMD).
* Database Structure
  + - In addressing the database structure challenge, we took a structured approach. We designed an Entity-Relationship (ER) diagram and established relational mappings. These tools provided us with a clear blueprint for defining the structure of our tables.
* Table Attributes and Definitions
  + - Resolving the issue concerning table attributes and their definitions was largely manageable. For certain attributes, we encountered the need to seek alternative definitions. For instance, when dealing with the "mobile number" attribute, we initially considered using the INT data type. However, due to character length restrictions, we pivoted to using VARCHAR to accommodate the data appropriately.
* MySQL Version Compatibility Issue
  + - Confronting compatibility issues arising from varying MySQL versions, we navigated this challenge by referencing the MySQL 8 Reference Manual. This resource enabled us to adapt our MySQL code to function seamlessly with the specific MySQL version we were using.

**Where we going to host**

We think to use for cloud hosting for several compelling reasons.

* Resource Optimization
  + Cloud hosting allows us to avoid the unnecessary consumption of our network connection and local resources for database hosting. By hosting our database in the cloud, we ensure efficient resource utilization.
* Resource Availability
  + Cloud hosting providers are well-equipped to deliver the essential resources needed for hosting, ensuring that we have access to the necessary infrastructure to support our website's requirements.

**What are the things, changes, that we must do in your backend?**

1. Database Schema and Structure

Define the database schema, including tables, columns, relationships, and constraints.

Ensure proper indexing on columns that are frequently used in queries for faster data retrieval.

2. Use user Authentication and Authorization

Implement user authentication to secure the system. Only authorized users should have access to specific parts of the database.

Create user roles (Admin, Lecturer, Technical Officer, Student) and define their permissions.

Implement role-based access control to restrict user actions based on their roles.

3. API Development

Create APIs to interact with the database from the frontend. These APIs should handle CRUD operations on tables.

4. Backup and Recovery:

Implement regular database backups to prevent data loss in the event of hardware failures or data corruption.

Create disaster recovery plans to restore the system to its last working state.

**Individual contribution to the backend development**

**TG-2021-1055 GHDB Ashen**

* Database table creation, including schema design.
* Entity-relationship (ER) modeling and mapping.
* Data insertion into the database.
* Development of database views.
* Report generation and integration.
* Contribution to the database normalization process.

**TG-2021-1061 W.A. Anjana Kavidu**

* Involvement in database table creation.
* Assisting with entity-relationship (ER) modeling and mapping.
* Contributing to data insertion tasks.
* Collaborating on the development of database views.
* Participating in the creation of reports.
* Contribution to the database normalization process.

**TG-2021-1066 K.A. Priyadarshana**

* Play pivotal role in database table creation.
* Contributing to entity-relationship (ER) modeling and mapping.
* Involvement in data insertion procedures.
* Coordinating with the team to develop database views.
* Participation in report generation.
* Confirm to the principles of database normalization.
* Coordinating as the team leader

**TG-2021-1023 K.S.K. Dilshanjith**

* Integral part of the database table creation process.
* Assisting in entity-relationship (ER) modeling and mapping tasks.
* Contributing to the insertion of data.
* Collaborating on the development of essential database views.
* Engaging in the creation of comprehensive reports.

**References**

Lecture notes

[*W3Schools MySQL*](https://www.w3schools.com/mysql/)

[*MySQL Documentation*](https://dev.mysql.com/doc/)