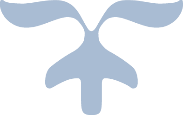


Attendance Management System



I3306 Database II

Prepared by   
Mohammad Dabaja

Lebanese University

Faculty of Sciences I

Department of computer Sciences

2019 - 2020

# Table of Contents

[Table of Contents 2](#_Toc36459365)

[Chapter I - Introduction 3](#_Toc36459366)

[I.1. Overview 3](#_Toc36459367)

[I.2. Plan of the document 3](#_Toc36459368)

[Chapter II - Architecture 4](#_Toc36459369)

[Chapter III - Database Modeling 5](#_Toc36459370)

[III.1. Conceptual Data Model 5](#_Toc36459371)

[III.2. Database Scheme 5](#_Toc36459372)

[III.3. Physical Data Model 6](#_Toc36459373)

[Chapter IV - Services & Interface Modeling 7](#_Toc36459374)

[IV.1. Services 7](#_Toc36459375)

[IV.2. Interface Modeling 8](#_Toc36459376)

[Chapter V - Implementation 12](#_Toc36459377)

[V.1. . Bat File 12](#_Toc36459378)

[V.2. Create Database 12](#_Toc36459379)

[V.3. DDL 13](#_Toc36459380)

[V.4. Triggers 15](#_Toc36459381)

[V.5. Stored Procedures 16](#_Toc36459382)

[V.6. Create User & grant privilege 17](#_Toc36459383)

[Chapter VI - Conclusion 18](#_Toc36459384)

[VI.1. Future Considerations 18](#_Toc36459385)

1. Introduction
   1. Overview

Attendance management system is made to manage the process of education in a university. It facilitates the election of the room were a teacher will give the lesson, and help the student and the teacher to track what was given I previous session. but the more important that it allow to take attendance of student in easy way.

* 1. Plan of the document

In this document, we propose the database architecture, Database Modeling, Services & Interface Modeling and implementation for the attendance management system application. We start by showing the 2-tier architecture used I this app then showing database engine, programming language and ODBC used. To come in the next chapter for the Database modeling (Conceptual Data Model, physical Data Model After that I chapter 3 we will introduce the services ad show images of interface. At the end we will show the implementation (create database, DDL, create indexes, create triggers, create user & give privileges, using transactions, procedures) and the .bat script used to build this database.

1. Architecture

This application uses 3-tier architecture. It has the following layers:

1. Presentation layer runs on a client (PC)
2. Data is stored on a Server.

The application interface which is called ODBC (Open Database Connectivity) an API which allows the client-side program to call the DBMS. The DBMS offers ODBC drivers for their DBMS. 2 tier architecture provides added security to the DBMS as it is not exposed to the end user directly. This architecture provides Direct and faster communication.

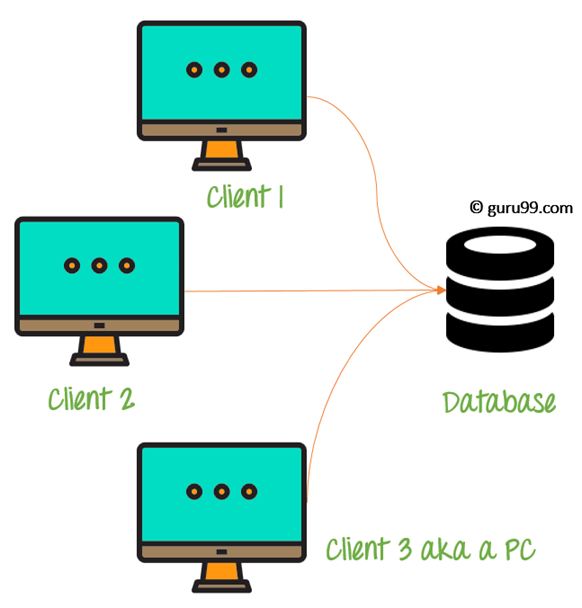


Figure : 2-tier Architecture Diagram

In this project the MySQL database engine was used as a database engine, java as programming language and J connector as ODBC.

1. Database Modeling
   1. Conceptual Data Model

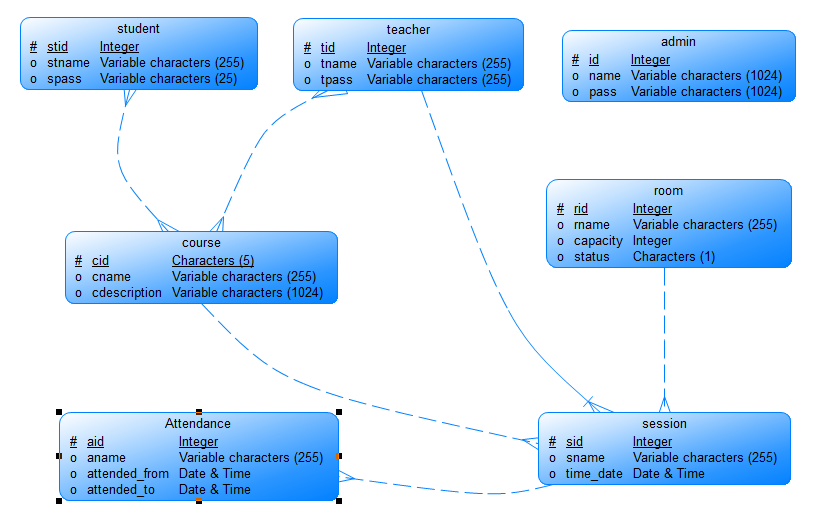


Figure : CDM

* 1. Database Scheme

Course (cid, came, description)

Student (stid, stname, pass)

Teacher (tid, tname, pass)

Admin (id, name, pass)

Room (rid, rname, capacity, status)

Session (sid, sname, time \_date)

Attendance (aid, aname, attended \_from, attended \_to)

Course \_Registration (#stid, #cid)

Teaches (#tid, #cid)

* 1. Physical Data Model

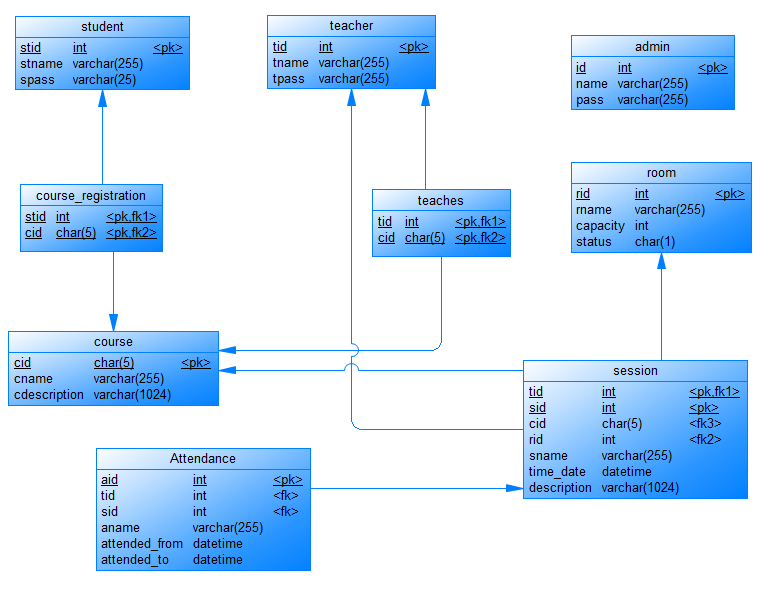


Figure : PDM

Database was modeled using Power Designer tool.

1. Services & Interface Modeling
   1. Services
2. Admin can view, insert, delete, update student, course, rooms and teacher to the system.
3. Admin and teacher can add, update or delete sessions for a specific time in an empty room.
4. Teacher can take attendance for student in a specific session.
   1. Interface Modeling

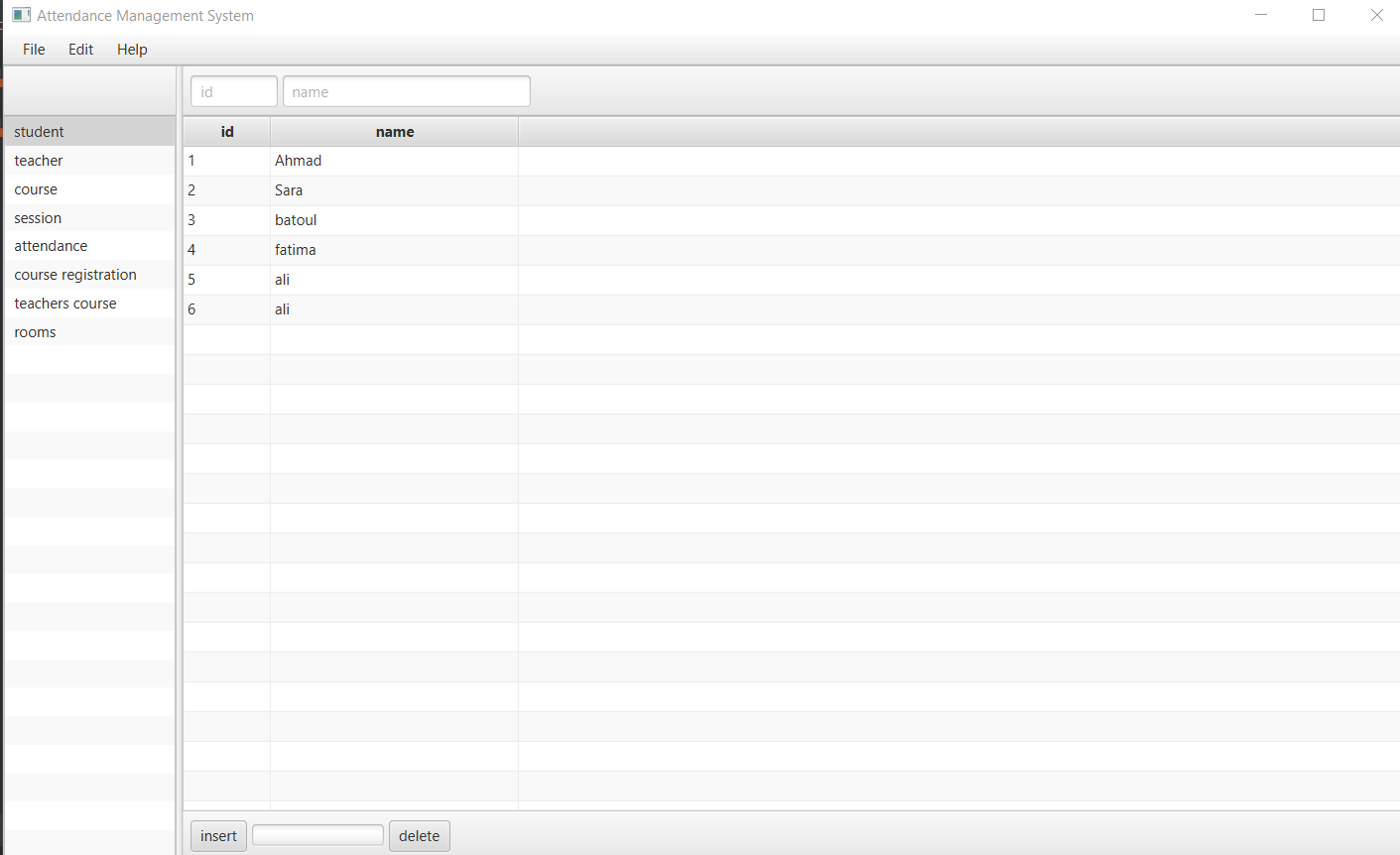


Figure : Student View

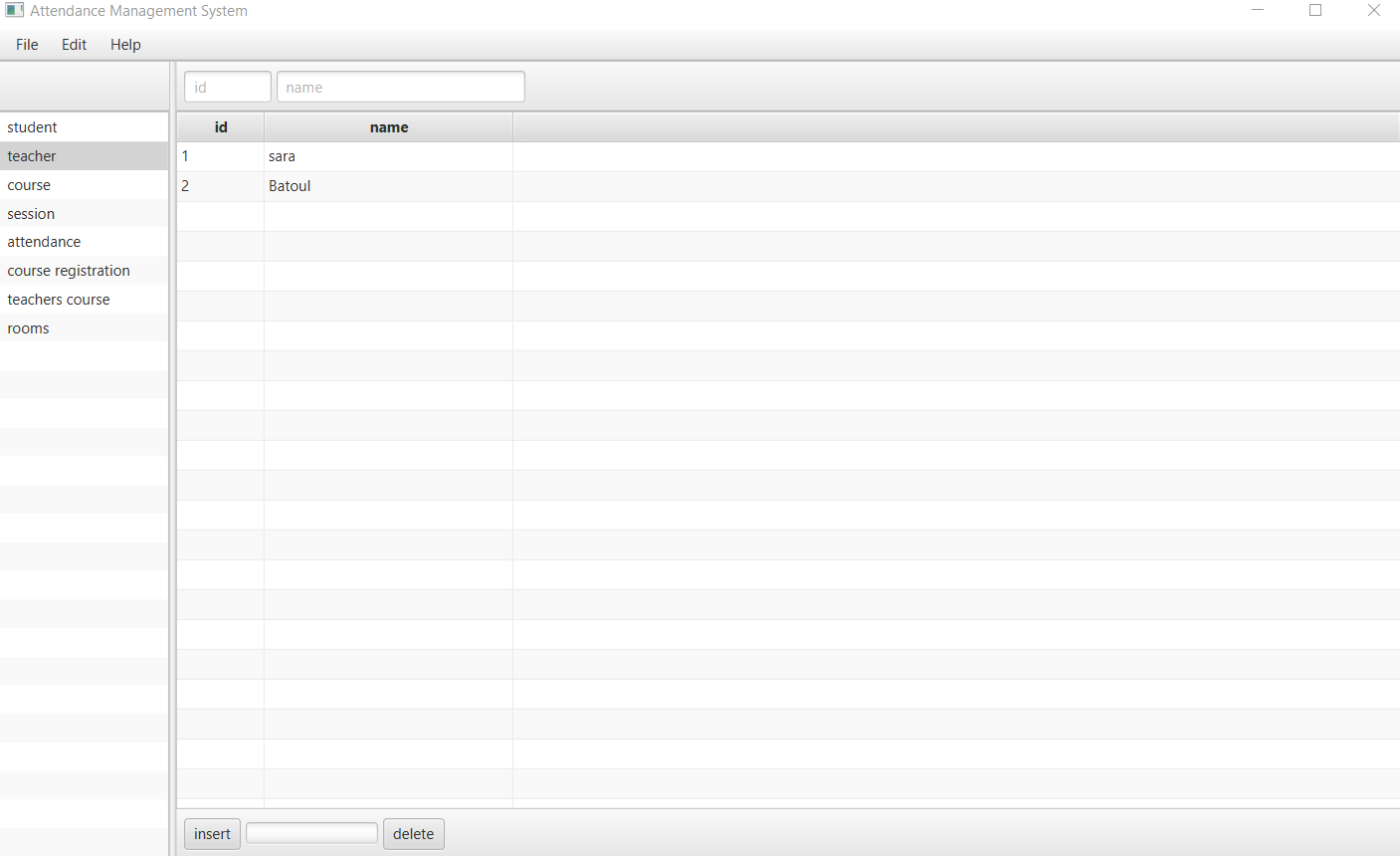


Figure : Teacher View

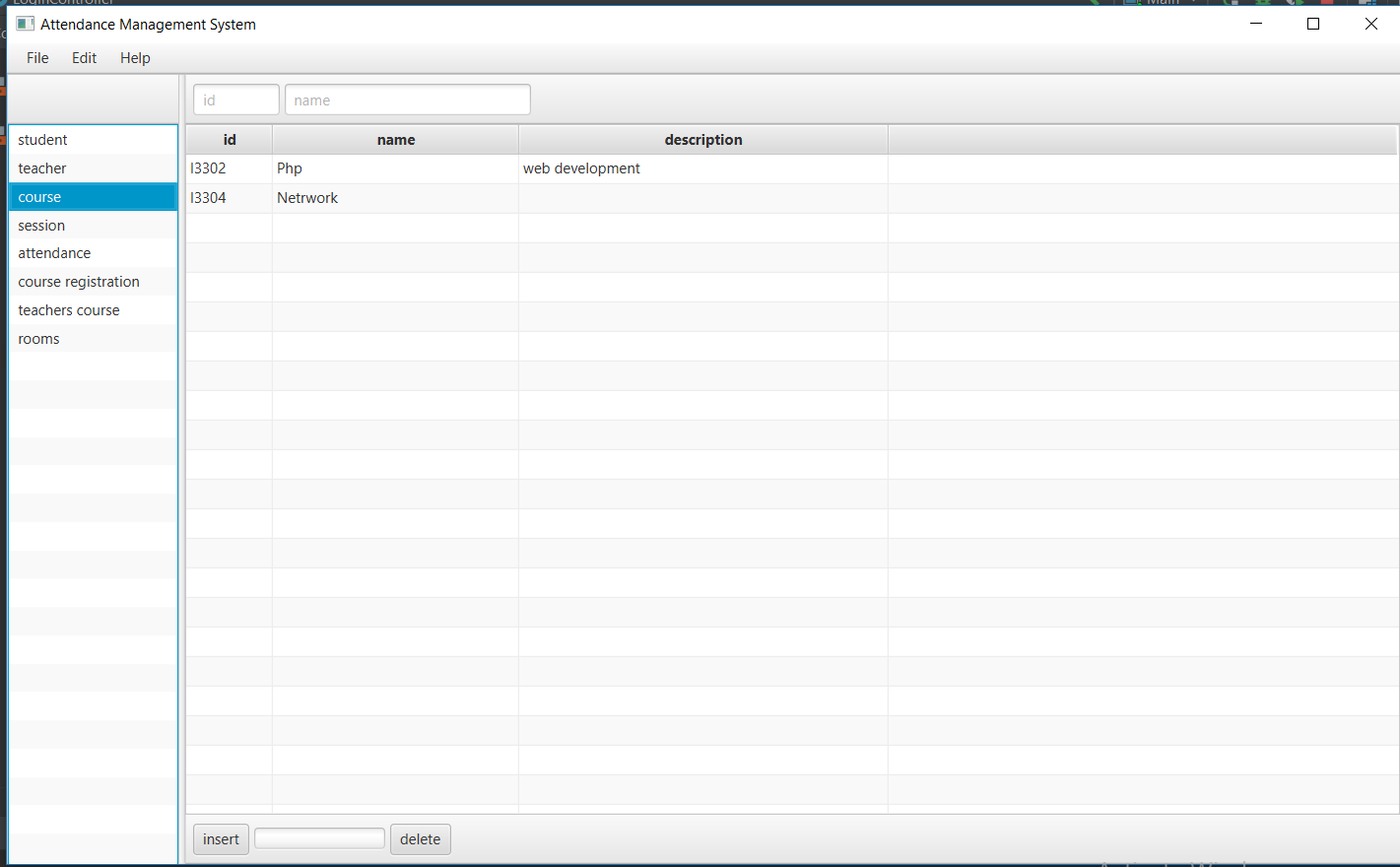


Figure : Course View

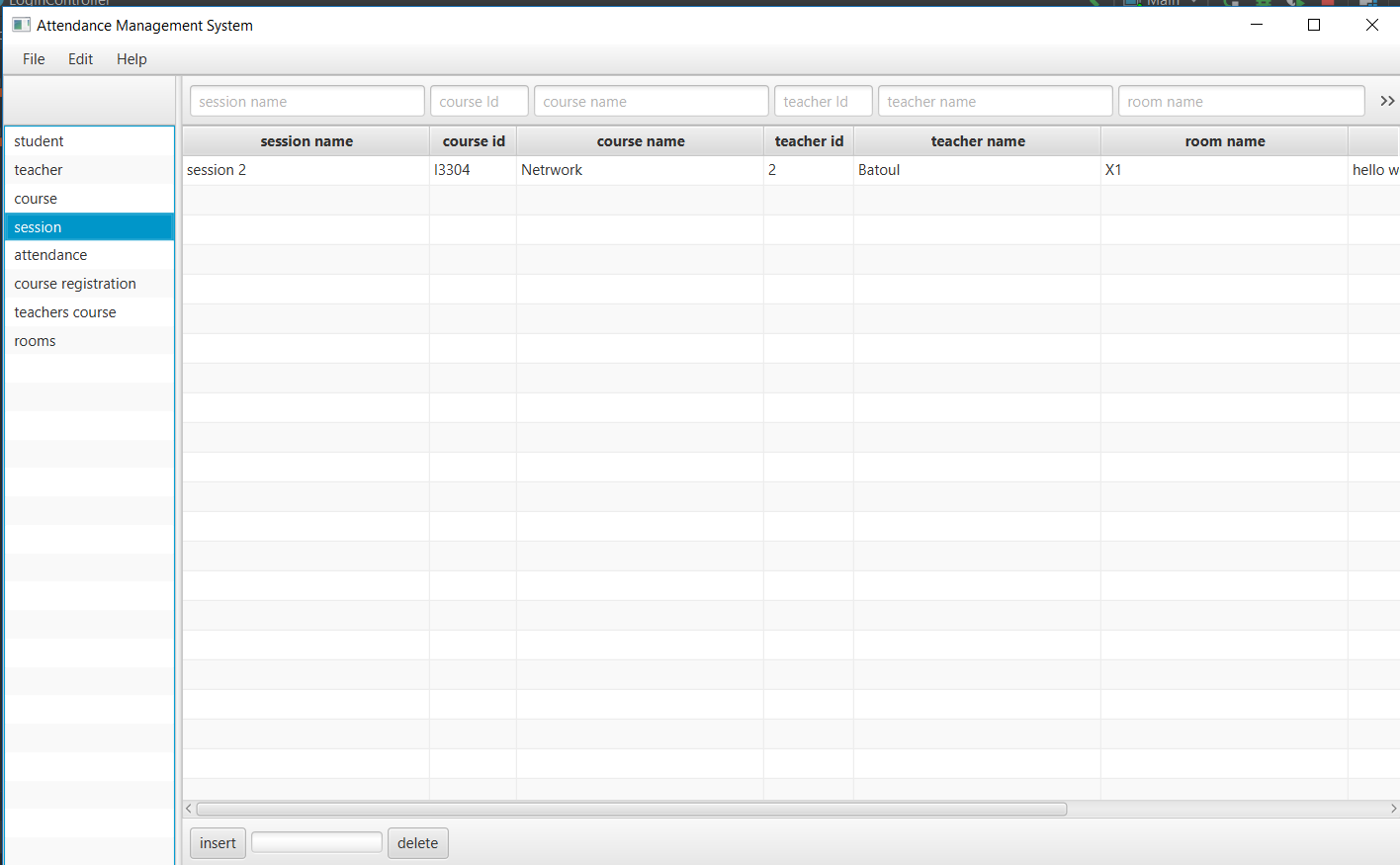


Figure : Session View

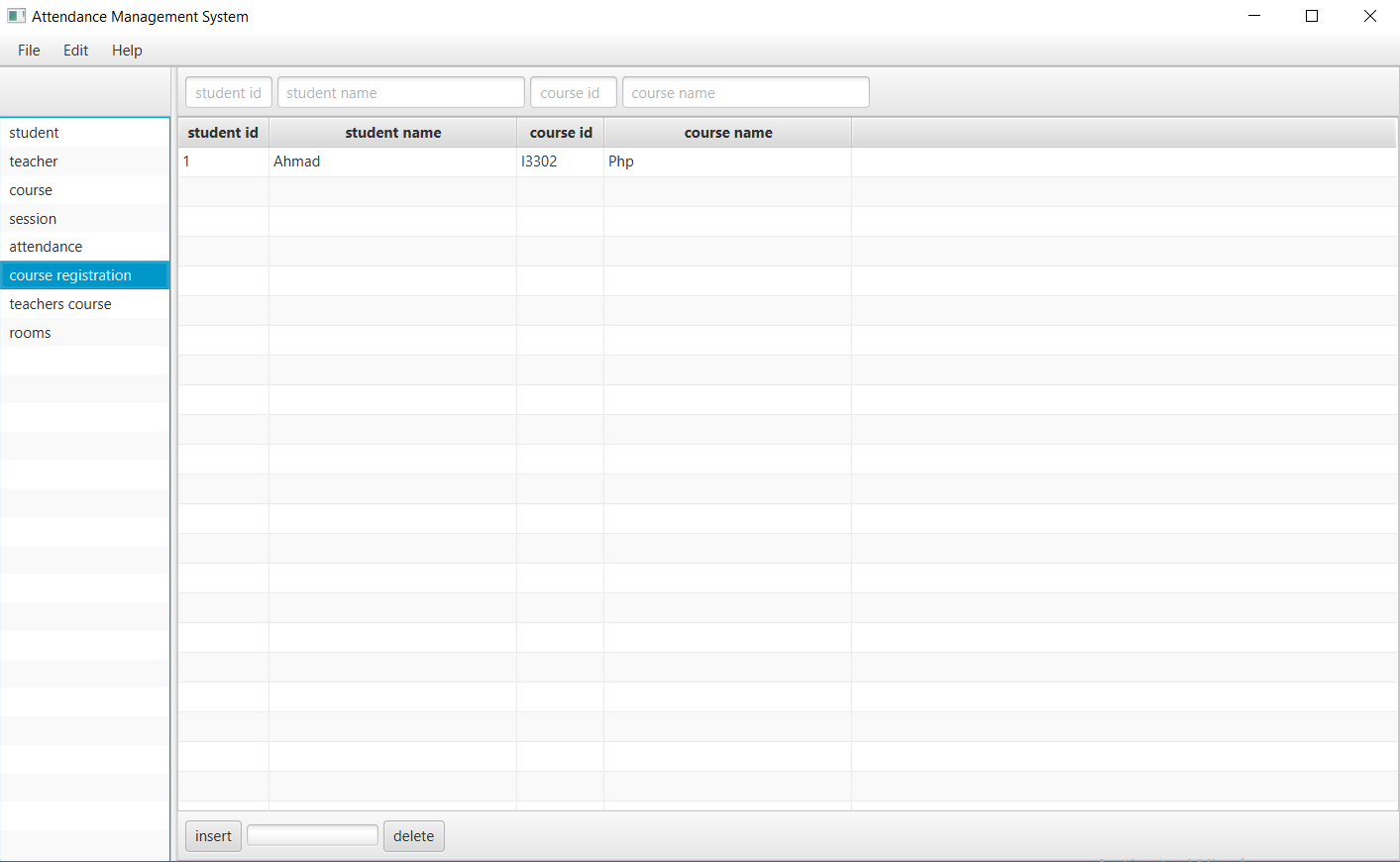


Figure : Course Registration View

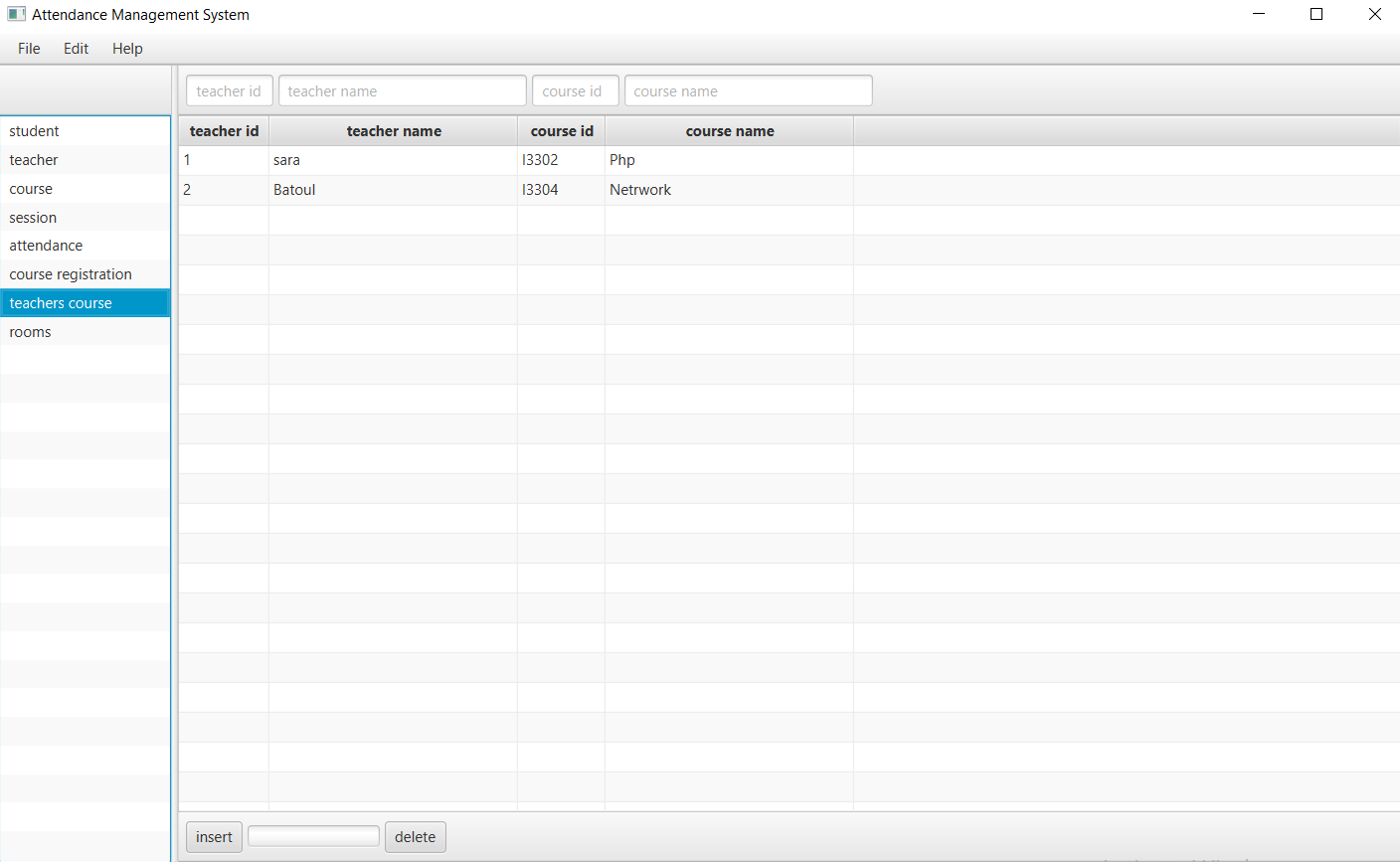


Figure : Teacher Courses View

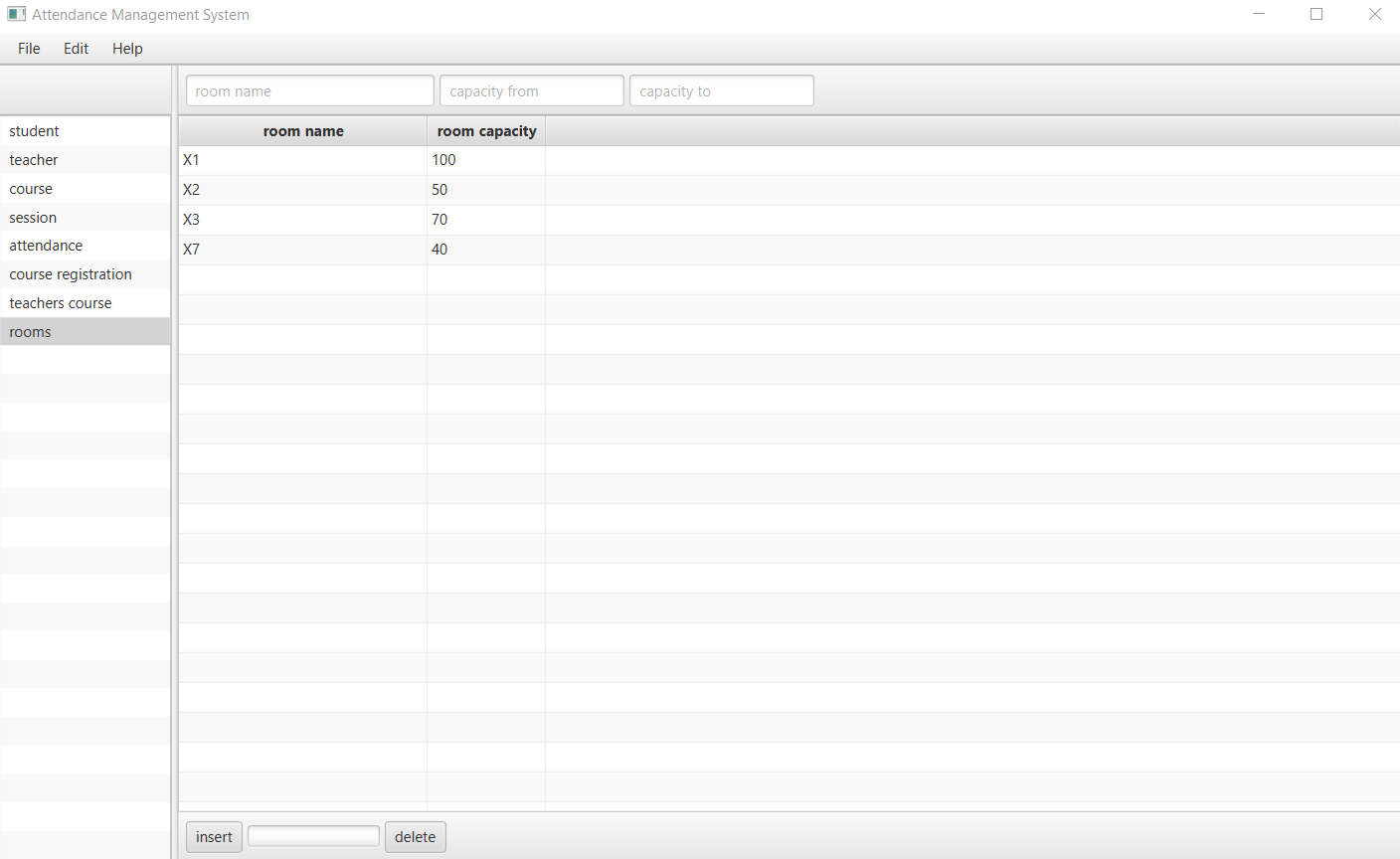


Figure : Room View

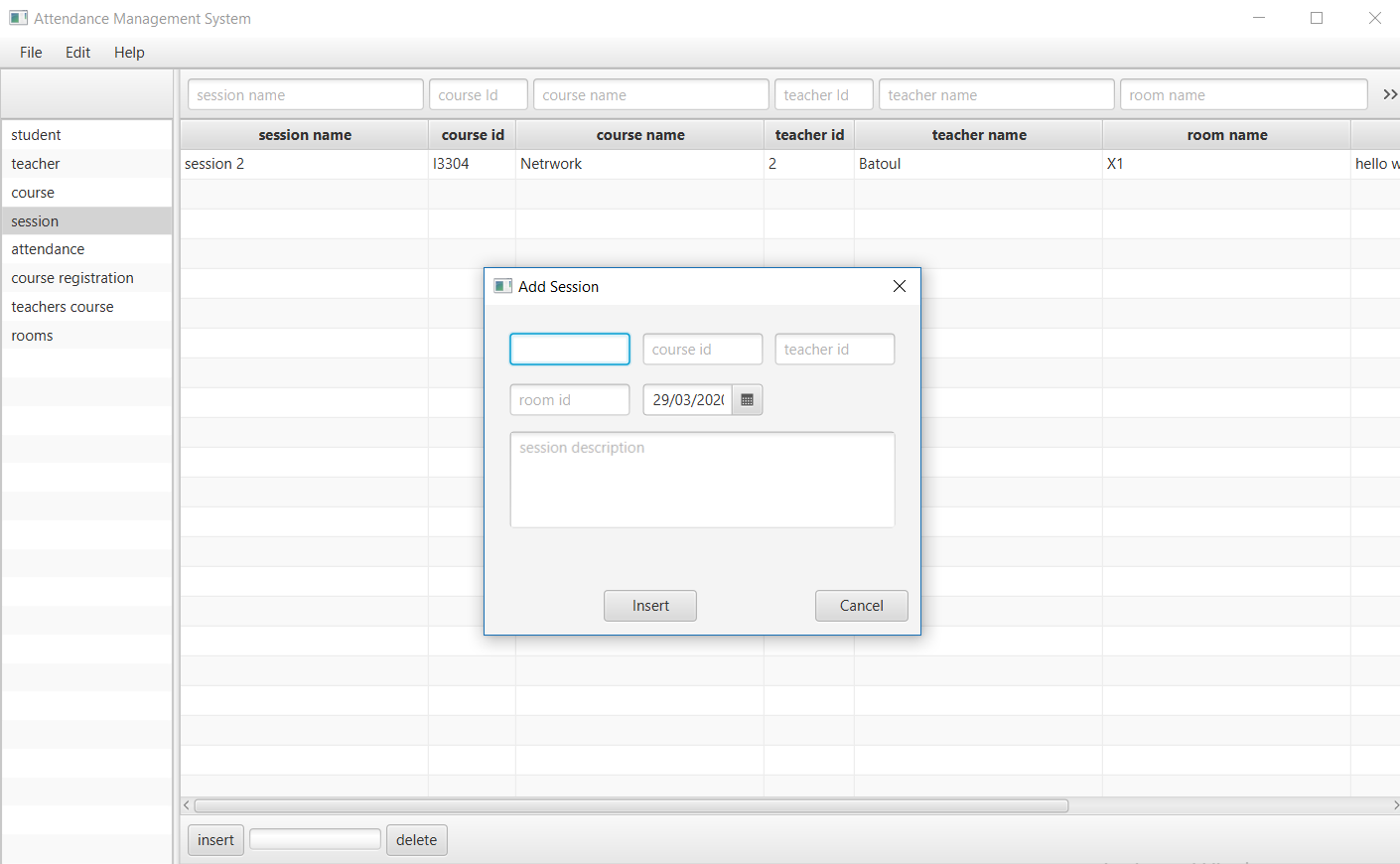


Figure : Insert Session

1. Implementation
   1. . Bat File

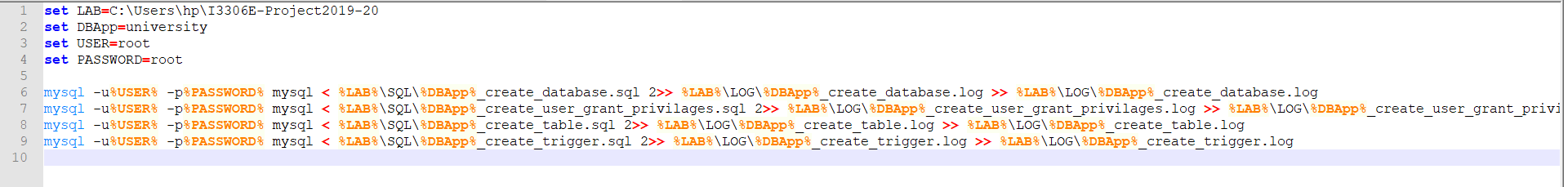
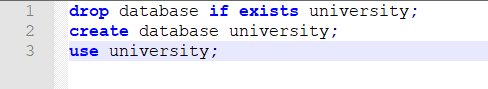


Figure : .Bat file

reasons for creating a MySQL database from a batch file:

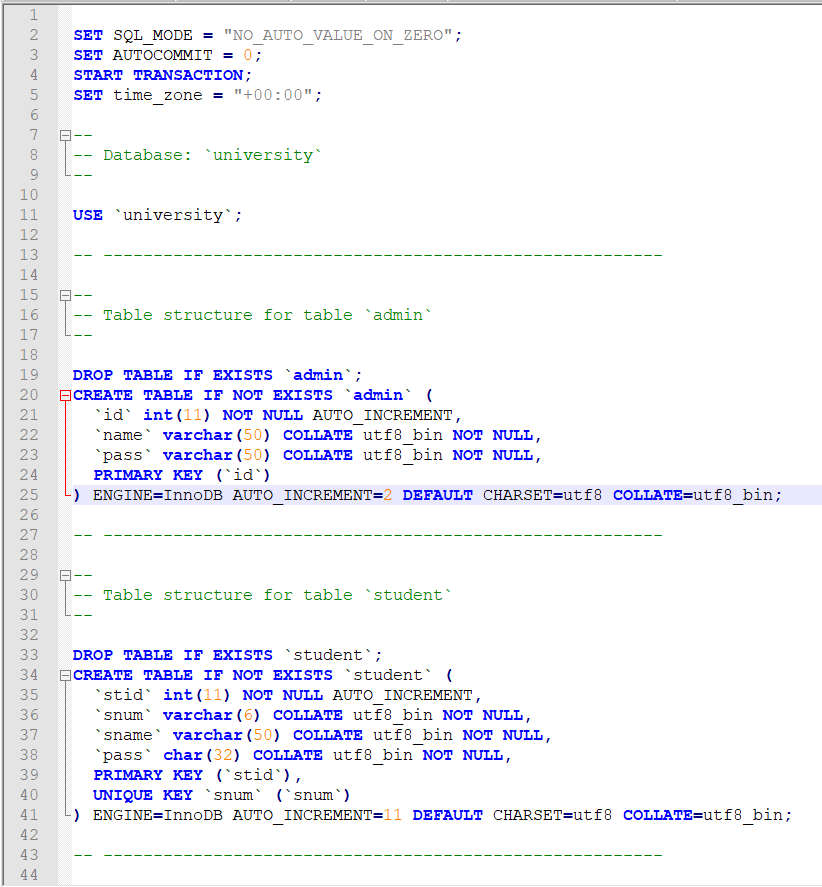
* the batch file provides an historical record of what's been done
* a batch file means that databases are reproducible - either at another time or in another place
  1. Create Database

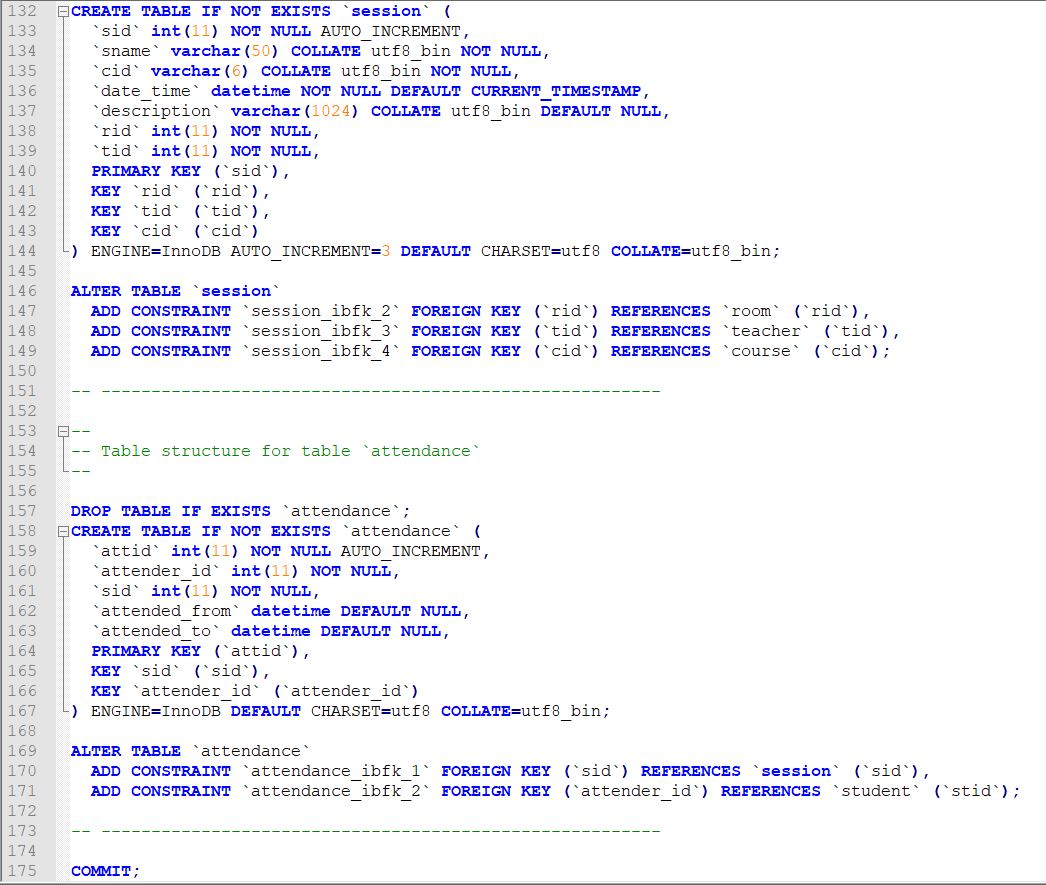


* 1. DDL

Tables of this database are created with the indexes on each primary key in university\_create\_database script.

All the DDL script is put I transaction.

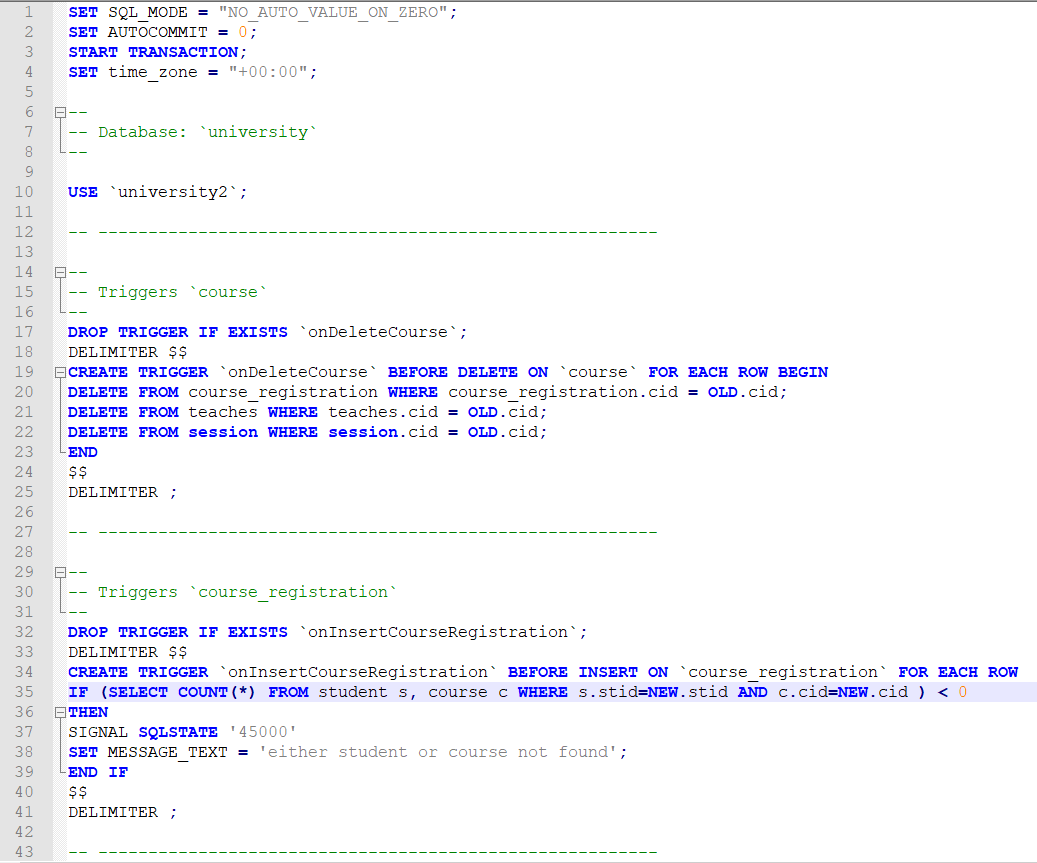


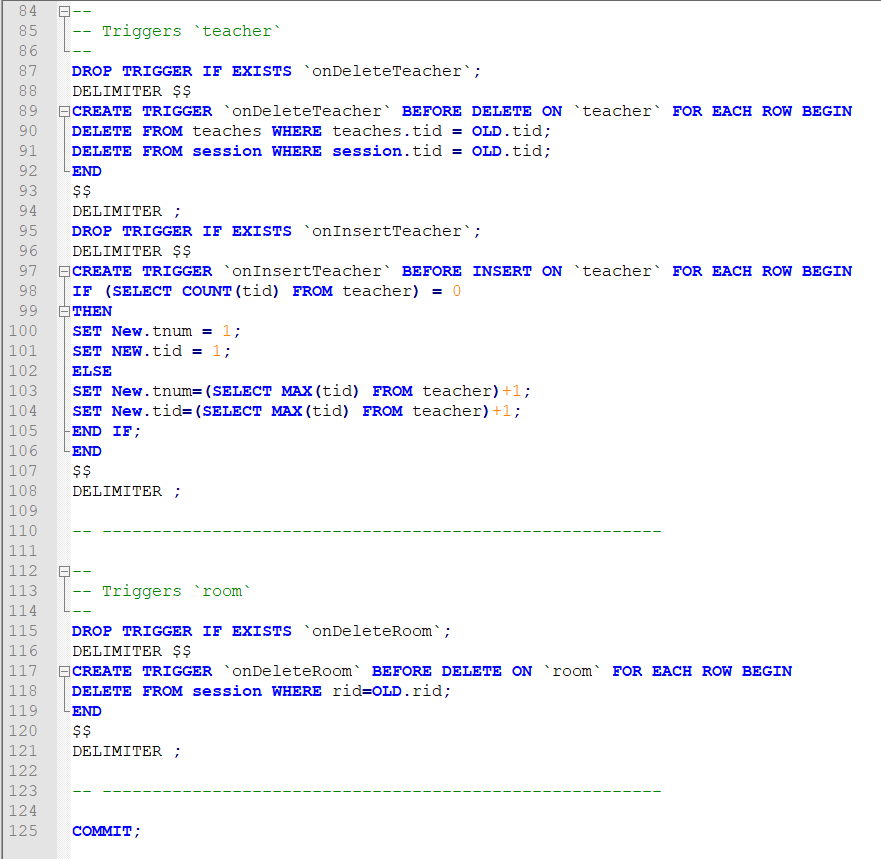


* 1. Triggers

Triggers are added on each tale to materialize the referential-integrity constraints in university\_create\_trigger script.

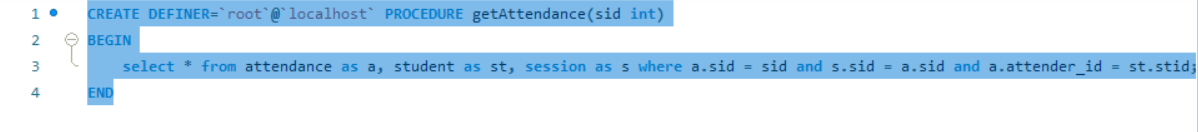
All the triggers scripts are put I transaction.





* 1. Stored Procedures

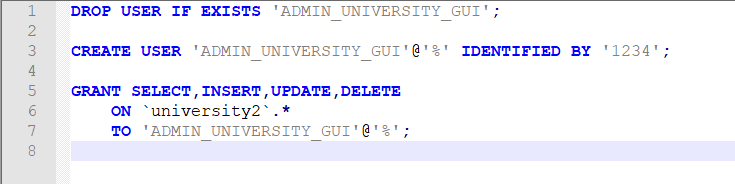
Plenty of Procedures are made for facilitating the application operation that are found in university\_create\_stored\_procedure script.





* 1. Create User & grant privilege

In university\_create\_user\_grant\_privilages script a specific user for our system is created having all privilege needed.



1. Conclusion
   1. Future Considerations

We are looking in the future to turn our architecture to 3-tier architecture so we can add server to upload and download content to it. We are looking to add some features so the teacher can upload lectures for a session such as video or pdf ad homework. We are looking for implementing the student interface so the student can download and see the lectures and upload his homework to the teacher.