

**VIETNAM NATIONAL UNIVERSITY   
HO CHI MINH CITY INTERNATIONAL UNIVERSITY**

**Attendance Management System**

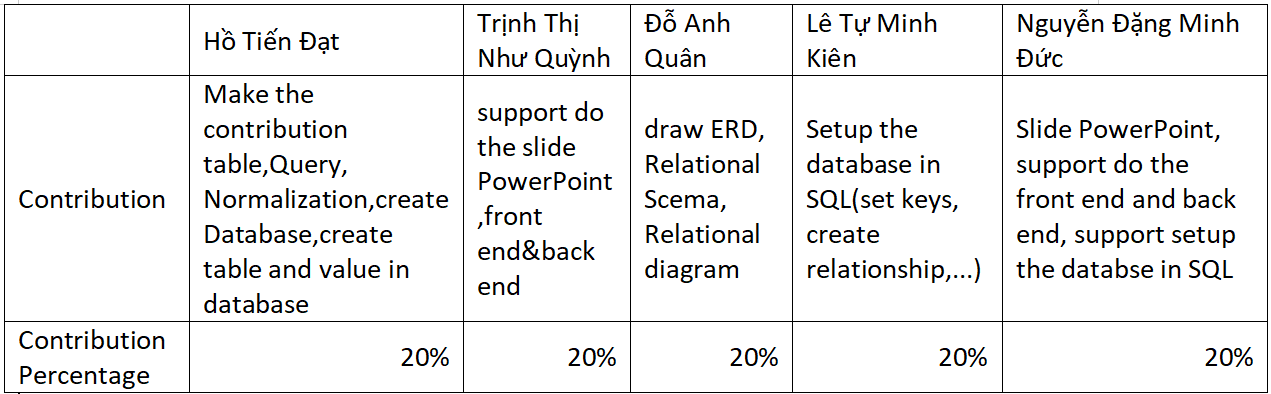
GROUP 4

**Instructor’s name:** Nguyen Thi Thuy Loan

**Group Members:**

|  |  |
| --- | --- |
| Name | Student\_ID |
| Hồ Tiến Đạt | ITCSIU21047 (Leader) |
| Đỗ Anh Quân | ITCSIU21027 |
| Nguyễn Đặng Minh Đức | ITCSIU21050 |
| Lê Tự Minh Kiên | ITDSIU21092 |
| Trịnh Thị Như Quỳnh | ITITIU20291 |

## Table 1.

****

## Table 2.

|  |
| --- |
| **Note:** Write report is the contribution of the whole team |

**Table of content:**

**[I. Abstract and Introduction](#_jg3fbaro5pv) 4**

[Problem statement:](#_66unsw5i5g2) 6

**[II. Diagram and schema](#_q9lb05y92vme) 6**

[Introduction:](#_6ocpiqiqud5z) 6

[ERD:](#_72vvet1bktv7) 6

[Figure 2.1. Entity Relationship Diagram ( ERD )](#_ym54ei6o29rt) 7

[Relational Schema:](#_fl5vd7gqsvbk) 8

[Figure 2.2. Relational Schema](#_xympa5dnpla) 9

**[III. Database](#_dxepq0148fam) 9**

[Figure 3.1.](#_ye05z58hmhj7) 9

[Figure 3.2.](#_9o3tfzih6t11) 10

[Figure 3.3.](#_91pww1xryi0w) 11

[Figure 3.4.](#_uny81bsbv61y) 12

**[IV. SQL Query](#_iokcqsxq4mm1) 13**

[Figure 4.1.](#_6gjptrav05yj) 23

[Figure 4.2.](#_jw3p2ngz916n) 24

[Figure 4.3.](#_s44o73sfdgc9) 25

[Figure 4.4.](#_ro3jb583xqi) 25

[Figure 4.5.](#_9n8u6bmqa9gm) 26

[Figure 4.6.](#_ijf0gzayl8fg) 26

[Figure 4.7.](#_qa1ohjd0659r) 27

[Figure 4.8.](#_q5s2e4lz5dds) 27

[Figure 4.9.](#_h54ye1v378u3) 28

**[V. Normalization](#_bmicw6nj5kmg) 28**

[Figure 5.1. Initial erd](#_52xssfr9xlkv) 28

[First Normal Form-1NF:](#_bwhx1p9uuabq) 28

[Figure 5.2.](#_pquntll3uffd) 29

[Second Normal Form-2NF:](#_qzzk64w154tu) 29

[Third Normal Form-3NF:](#_tlvq4ghi0dl2) 30

[Figure 5.3.](#_6wud74v2xi5y) 30

**[VI. Software:](#_mhcfjg8qtfov) 30**

[Figure 6.1. error image](#_hugeqks6f4e) 31

[Technologies :](#_w8dfbz8zcse9) 31

[Figure 6.2. pom.xml](#_yp49bgcbkcwe) 32

[Figure 6.3.](#_hgipi77gmgdm) 32

[Figure 6.4. AdminSecurityConfig.java](#_8mikc9a8w0af) 33

[Figure 6.5. AdminLoginController.java](#_5h7ehcqcf5wl) 34

[Figure 6.6. AttendanceEntity.java](#_bfppxktmrfil) 35

[Figure 6.7. AttendanceRepository.java](#_4kop9j8zcrsx) 36

[Figure 6.8. admin-login.html](#_1cjtihxmtkgg) 37

**[VII. Reference:](#_ym1auds5qtjt) 37**

**[VIII. Conclusion:](#_88l41rk99ar4) 37**

# I. Abstract and Introduction

Abstract: In an effort to replace the pen-and-paper roll call system, which has been widely used in all education systems around the world, our team suggests utilizing current technology to develop an online attendance monitoring application that can assist instructors and tutors. Since the 5.0 era has begun, everything must be modernized in order to be appropriate for contemporary living. There are numerous facets of contemporary existence that involve technology. Therefore, we will need to discover a high-tech alternative to the conventional method.

This technique also benefits the mentees and students because it makes it simpler to track their progress and increases their awareness of the number of days remaining before they are disqualified from taking the exam.

This project's objective was to create a Check Attendance System for an educational institution. The system is designed to assist instructors and administrators in maintaining student attendance records and generating reports to analyze attendance data.

The five-person team that completed the project enabled continuous feedback and iterative development. The team was accountable for defining the scope of the project, devising functional and non-functional requirements, designing the database, and implementing the system.

It offers a comprehensive summary of the endeavor and its results. It begins with a description of the project's goals, objectives, and methodology, as well as the development instruments. The report then describes the system's functional and non-functional requirements, as well as the use cases and user stories, developed to encapsulate these requirements.

The report outlines the system's design, including the Entity-Relationship Diagram (ERD) and Relational Schema, which define the database structure and entity relationships. Additionally, the report describes the system's architecture and the front-end and back-end components used in its construction.

The report concludes with an assessment of the project's success and recommendations for future system enhancements. The project's accomplishments and difficulties are discussed, along with the lessons learned.

Overall, this report functions as a thorough resource for educational institutions and software development teams seeking to create similar attendance management systems.

## Problem statement:

The online reviewing system simplifies the traditional procedures, which previously relied on roll calls but have since transitioned to using applications and devices. From this vantage point, we are in a better position to assist instructors with attendance and make it easier for students to track their own development.

This system also includes a feedback system through which students can report any errors in their progression, as well as a hotline through which tutees can contact their instructors if they require assistance.

The management team is responsible for updating customer service and maintaining an accurate customer database.

# 

# II. Diagram and schema

## Introduction:

This academic report provides an overview of the entity-relationship diagram (ERD) and relational schema for a group project on a Check Attendance System. The project aims to design and develop a database management system for an educational institution to manage the attendance records of students and teachers.

## ERD:

An entity relationship diagram (ERD) is a graphical representation of entities and their relationships. Students, teachers, classes, and attendance records can be defined as entities within the Check Attendance System. Following is a description of the connections between these entities:

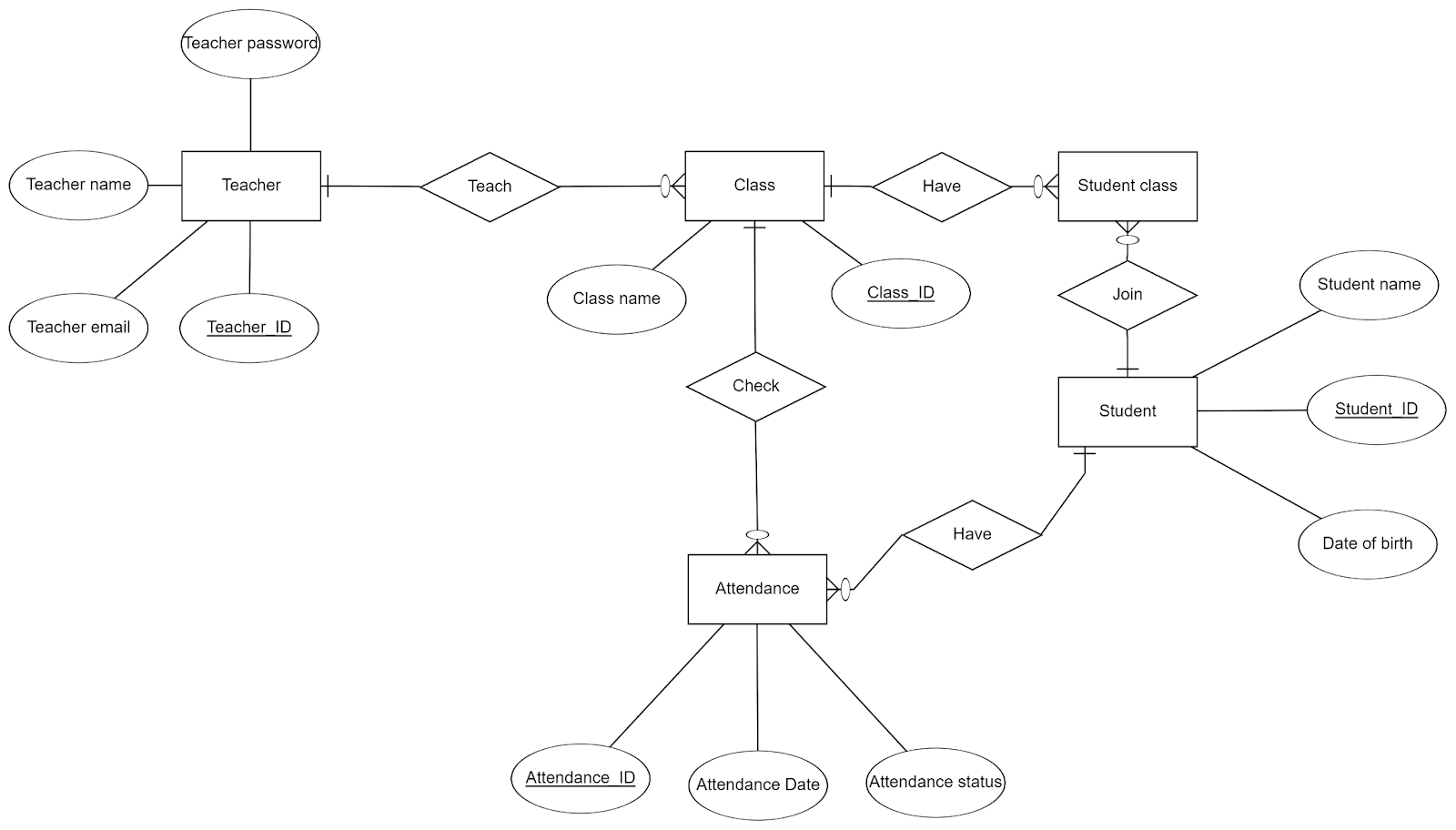
A student can attend multiple classes, and multiple students can attend a class. A diamond symbol in the ERD indicates that this relationship is a many-to-many relationship.

A teacher can instruct multiple classes, but a class can only be instructed by one teacher. A diamond on the ERD stands in for this relationship, which is also a one-to-many relationship.

A student can have multiple attendance records for distinct classes. An attendance record can be associated with both a student and a class. The student entity can visualize this relationship as a many-to-one relationship using a diamond symbol.

Another entity included in the ERD is the student\_class, which is essentially a connection point between the student and class entities. The student\_class has a many-to-one relationship with the student and a many-to-one relationship with the diamond-shaped class entity.

The ERD for the Check Attendance System is shown below:



## Figure 2.1. Entity Relationship Diagram ( ERD )

## Relational Schema:

A Relational Schema is a collection of relations or tables used to represent entities and their relationships within a database management system. The Check Attendance System's relationships can be described as follows:

-Student\_ID, student\_name, and date\_of\_birth are attributes of the student relationship.

-Class\_ID and class\_name are attributes of the class relation.

-There are four attributes associated with the teacher relationship: teacher\_ID, teacher\_name, teacher\_email, and teacher\_password.

-The relationship between attendance records includes the attributes attendance\_ID, attendance\_status, and attendance\_date.

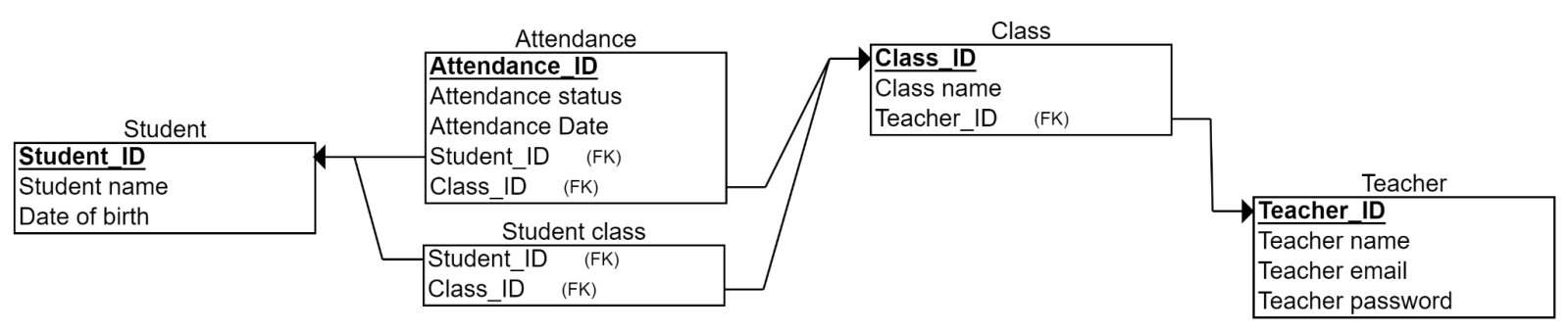
Following is a description of the relationships between these entities:

The junction or linking table student\_class is a representation of the one-to-many relationship between students and classes. This table contains the principal keys for the student and class relationships, student\_ID and class\_ID, respectively.

The teacher\_ID table contains the primary keys for the teacher and class relations, teacher\_ID, respectively.

Student\_ID and class\_ID are the two foreign keys for student and class associated with the attendance\_ID.

The Relational Schema for the Check Attendance System is shown below:



## Figure 2.2. Relational Schema

# III. Database

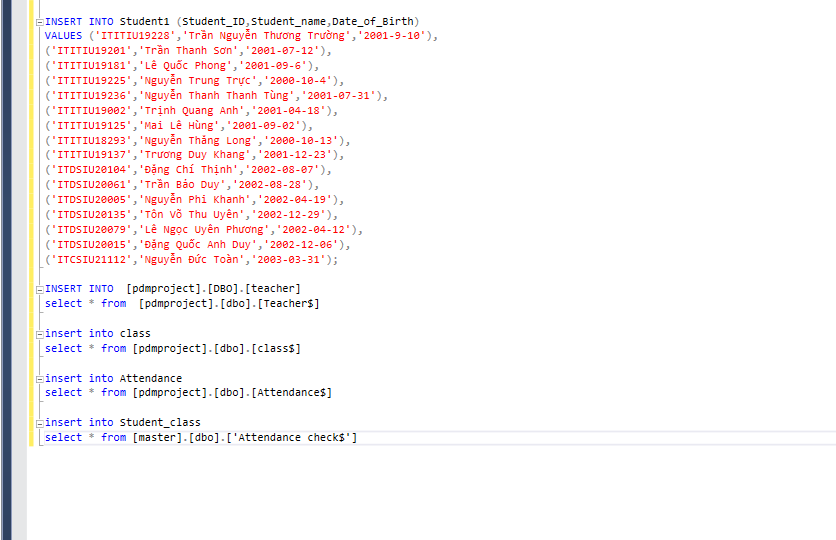
# 

## Figure 3.1.

After creating a database in Excel, we import data from an Excel file using this method: Click on the database we create in SQL Server (in this case, we call PDMPROJECT) ⇒ task ⇒ import Data.

## Figure 3.2.

Then, we construct tables in SQL Server and assign primary and foreign keys to each table (follow to ERD and schema).



## Figure 3.3.

Next, we insert the data from the imported Excel files into those tables in SQL Server. Note that the table names we assign to these SQL Server-created tables correspond to the column names in Excel files. Here, we demonstrate two methods for inserting data into SQL Server tables.

The first method is to manually insert data. It indicates we employ the structure

(in tables STUDENT1):

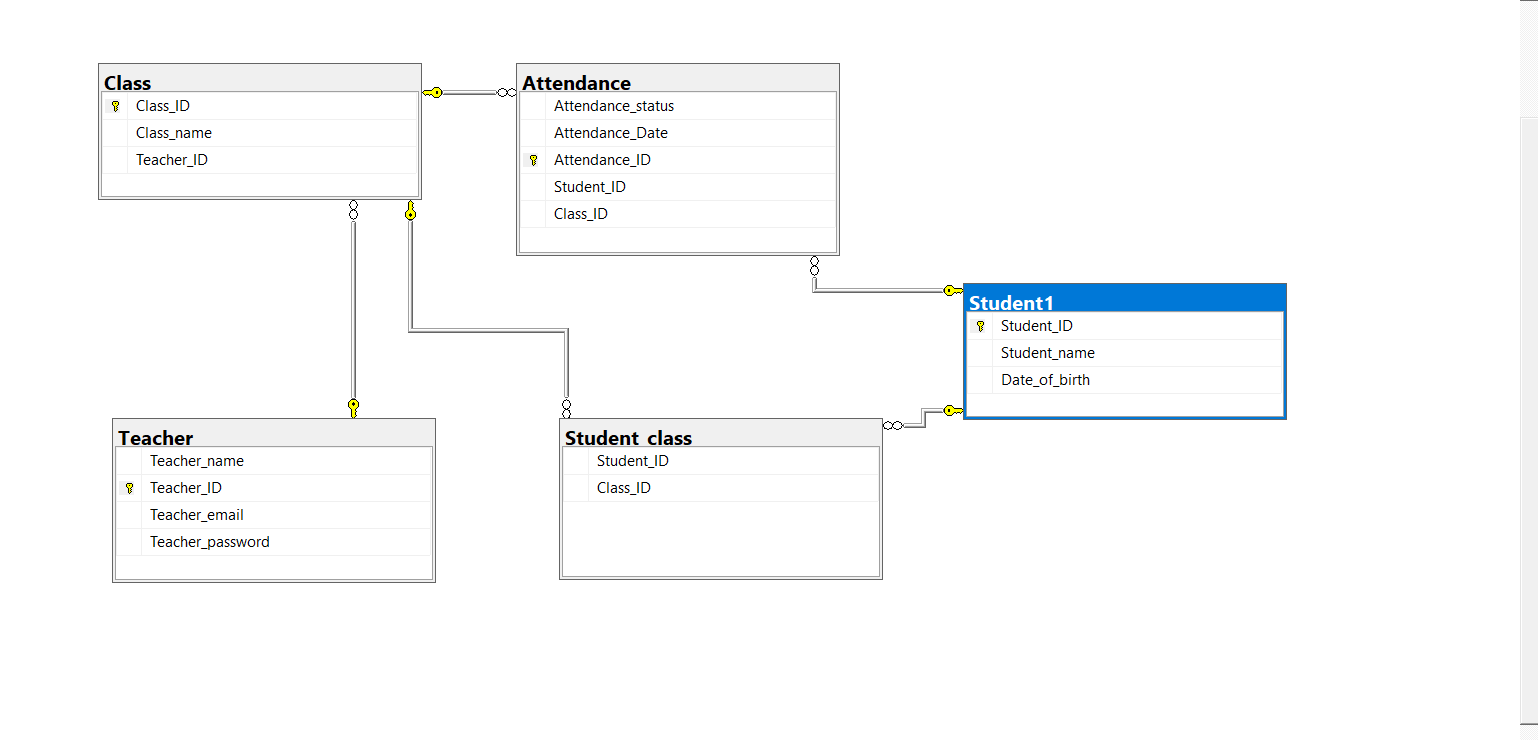
INSERT INTO table\_name (column1, column2, column3, ...)

VALUES (value1, value2, value3, ...);

The second way is much faster (in other tables):

Insert into [database name].[dbo].[table1 name]

select \* from [database name].[dbo].[table2 name]



## Figure 3.4.

Finally, we create the database diagram to represent the relationship between the tables in SQL Server in this way: click on Database diagram⇒ new Database diagram. After that, SQL Server draws this diagram for us automatically.

# IV. SQL Query

SQL Query is the method to select values from the existing database. SQL Query uses multiple syntaxes to support search and pick out the value that we need from a manifold of data and values. For instance, the basic syntax of a query is:

“ SELECT [DISTINCT] <*target-list>*

FROM <*relation-list>*

WHERE <*qualification>*

**Or:**

SELECT DISTINCT column1, column2....columnN

FROM table\_name”

But before we can query, we must first connect SQL Server Management/MySQL to the NetBean environment. All the comprehensive information and instructions will be contained in the Practice file of Lab 4 of Lab PDM.

After completing the connection between the two software, we construct and style the query box using code. The codes include the design of the query box and the class responsible for generating and running the box. The code will be written using the programming language Java. Here is the code:

“package lab4pdm;

import java.awt.BorderLayout;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JTextField;

import javax.swing.JButton;

import javax.swing.JTextArea;

import java.awt.Font;

import javax.swing.SwingConstants;

import java.awt.event.ActionListener;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.ResultSetMetaData;

import java.sql.SQLException;

import java.sql.Statement;

import java.awt.event.ActionEvent;

public class form extends JFrame {

private JPanel contentPane;

private JTextField txtQuery;

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

form frame = new form();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

public form() {

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(400, 400, 866, 550);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(null);

JLabel lblQuery = new JLabel("Query");

lblQuery.setHorizontalAlignment(SwingConstants.LEFT);

lblQuery.setFont(new Font("Tahoma", Font.BOLD, 24));

lblQuery.setBounds(12, 30, 73, 35);

contentPane.add(lblQuery);

txtQuery = new JTextField();

txtQuery.setFont(new Font("Tahoma", Font.BOLD, 16));

txtQuery.setBounds(95, 13, 588, 91);

contentPane.add(txtQuery);

txtQuery.setColumns(10);

JTextArea txtResult = new JTextArea();

txtResult.setBounds(33, 140, 783, 326);

contentPane.add(txtResult);

JButton btnRun = new JButton("Run");

btnRun.setFont(new Font("Tahoma", Font.ITALIC, 21));

btnRun.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

if(txtQuery.getText().length() ==0){

JOptionPane.showMessageDialog(null,"Please input query string!", "Message", JOptionPane.WARNING\_MESSAGE);

return;

}

txtResult.selectAll();

txtResult.replaceSelection("");

Connection conn = null;

try {

DriverManager.registerDriver(new com.microsoft.sqlserver.jdbc.SQLServerDriver());

conn = DriverManager.getConnection("jdbc:sqlserver://localhost:1433;databaseName=Project PDM;user=sa;password=sa; encrypt = false");

} catch ( SQLException e) {

e.printStackTrace();

}

try {

Statement stmt = conn.createStatement();

String SQL = txtQuery.getText() ;

ResultSet rs = stmt.executeQuery(SQL);

StringBuilder results = new StringBuilder();

ResultSetMetaData metaData = rs.getMetaData();

int numberOfColumns = metaData.getColumnCount();

for (int i=1; i<= numberOfColumns; i++){

results.append(metaData.getColumnName(i)).append("\t");

}

results.append("\n");

//MetaData

while (rs.next()) {

for (int i = 1; i<= numberOfColumns;i++) {

results.append(rs.getObject(i)).append("\t");

}

results.append("\n");

}

txtResult.setText(results.toString());

}

catch(SQLException e) {

txtResult.setText(e.getMessage());

/\*

package lab4pdm;

import java.awt.BorderLayout;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JTextField;

import javax.swing.JButton;

import javax.swing.JTextArea;

import java.awt.Font;

import javax.swing.SwingConstants;

import java.awt.event.ActionListener;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.ResultSetMetaData;

import java.sql.SQLException;

import java.sql.Statement;

import java.awt.event.ActionEvent;

/\*\*

\*

\* @author HP

\*/

public class form extends JFrame {

private JPanel contentPane;

private JTextField txtQuery;

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

form frame = new form();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

public form() {

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(400, 400, 866, 550);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(null);

JLabel lblQuery = new JLabel("Query");

lblQuery.setHorizontalAlignment(SwingConstants.LEFT);

lblQuery.setFont(new Font("Tahoma", Font.BOLD, 24));

lblQuery.setBounds(12, 30, 73, 35);

contentPane.add(lblQuery);

txtQuery = new JTextField();

txtQuery.setFont(new Font("Tahoma", Font.BOLD, 16));

txtQuery.setBounds(95, 13, 588, 91);

contentPane.add(txtQuery);

txtQuery.setColumns(10);

JTextArea txtResult = new JTextArea();

txtResult.setBounds(33, 140, 783, 326);

contentPane.add(txtResult);

JButton btnRun = new JButton("Run");

btnRun.setFont(new Font("Tahoma", Font.ITALIC, 21));

btnRun.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

if(txtQuery.getText().length() ==0){

JOptionPane.showMessageDialog(null,"Please input query string!", "Message", JOptionPane.WARNING\_MESSAGE);

return;

}

txtResult.selectAll();

txtResult.replaceSelection("");

Connection conn = null;

try {

DriverManager.registerDriver(new com.microsoft.sqlserver.jdbc.SQLServerDriver());

conn = DriverManager.getConnection("jdbc:sqlserver://localhost:1433;databaseName=Project PDM;user=sa;password=sa; encrypt = false");

} catch ( SQLException e) {

e.printStackTrace();

}

try {

Statement stmt = conn.createStatement();

String SQL = txtQuery.getText() ;

ResultSet rs = stmt.executeQuery(SQL);

StringBuilder results = new StringBuilder();

ResultSetMetaData metaData = rs.getMetaData();

int numberOfColumns = metaData.getColumnCount();

for (int i=1; i<= numberOfColumns; i++){

results.append(metaData.getColumnName(i)).append("\t");

}

results.append("\n");

//MetaData

while (rs.next()) {

for (int i = 1; i<= numberOfColumns;i++) {

results.append(rs.getObject(i)).append("\t");

}

results.append("\n");

}

txtResult.setText(results.toString());

}

catch(SQLException e) {

txtResult.setText(e.getMessage());

}

}

});

btnRun.setBounds(723, 30, 93, 47);

contentPane.add(btnRun);

}

}

}

}

});

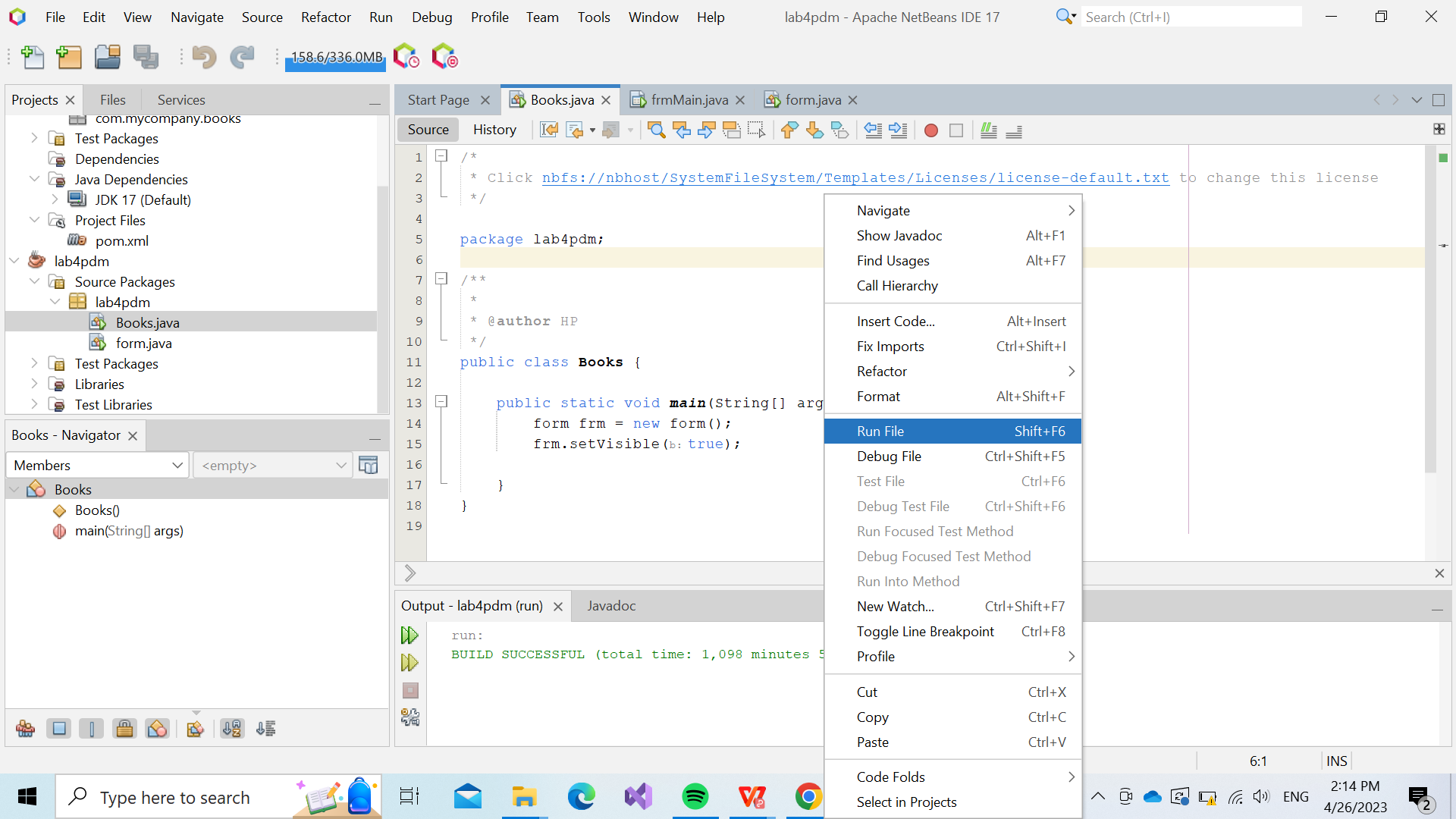
btnRun.setBounds(723, 30, 93, 47);

contentPane.add(btnRun);

}

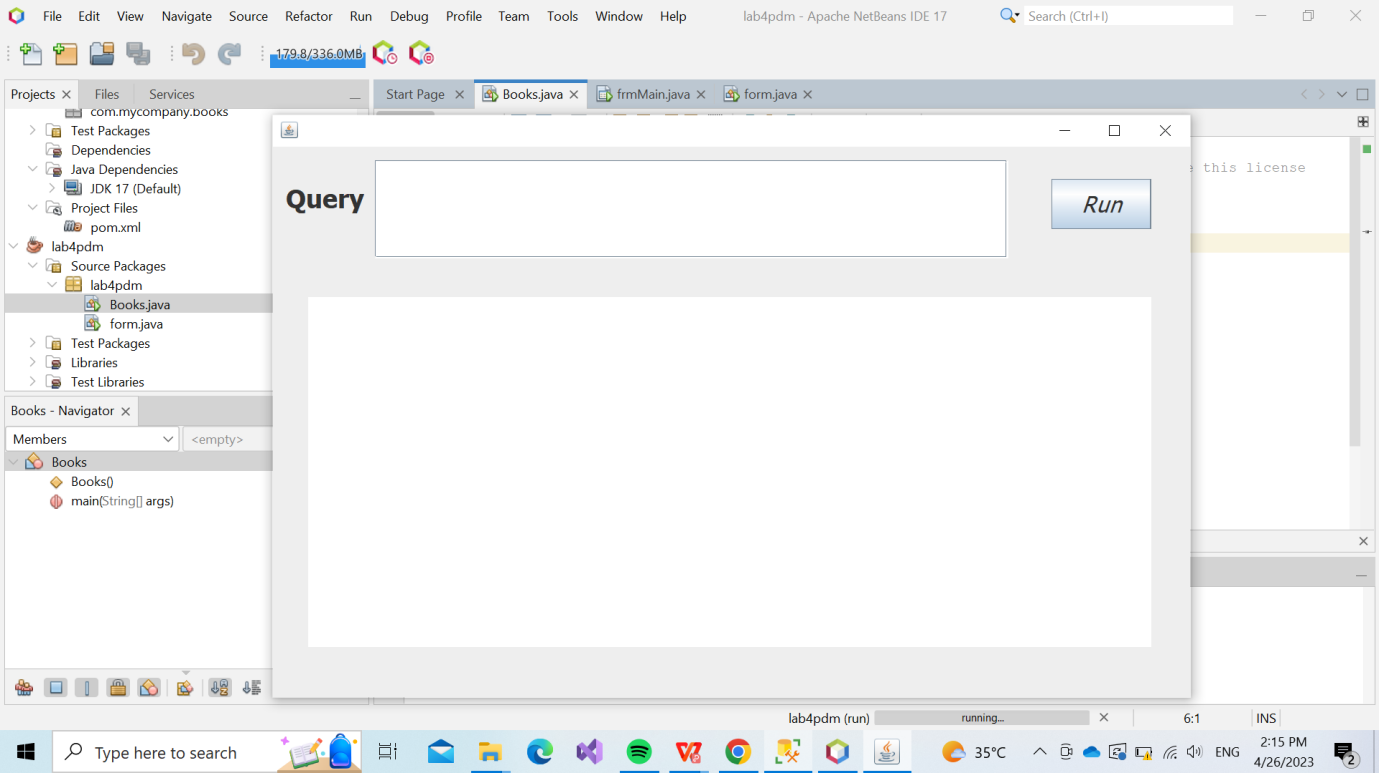
}”

**Right-click on the mouse and choose “Run file”:**



## Figure 4.1.

**A query box will appear:**

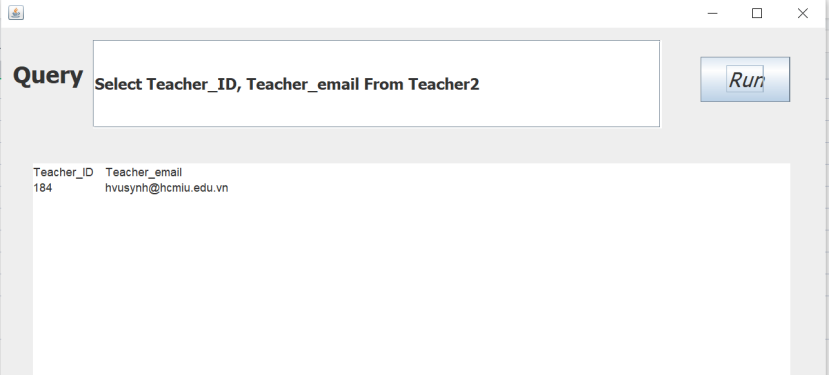


*Figure 4.2.*

In the blank next to the word Query, we enter some syntax (mentioned above or otherwise) to retrieve all information and data from the SQL Server Management database. Our team has compiled a list of questions that range in difficulty from simple to difficult, and the queries of them:

## 1)Select id and email of teacher

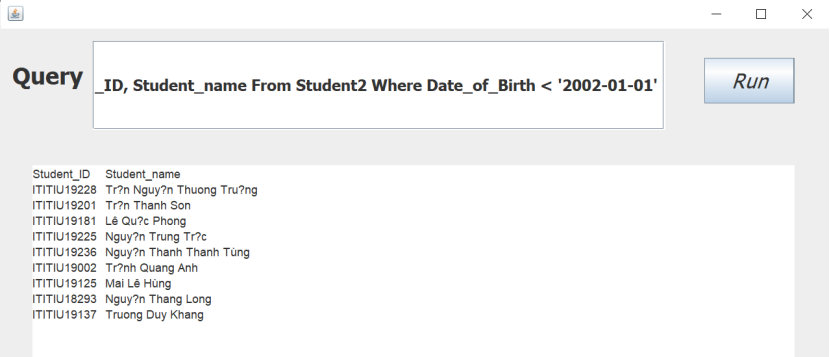
Code: Select Teacher\_Id, Teacher\_email From Teacher2



## Figure 4.3.

## 2) Select all the student\_ID and name that born in 2001

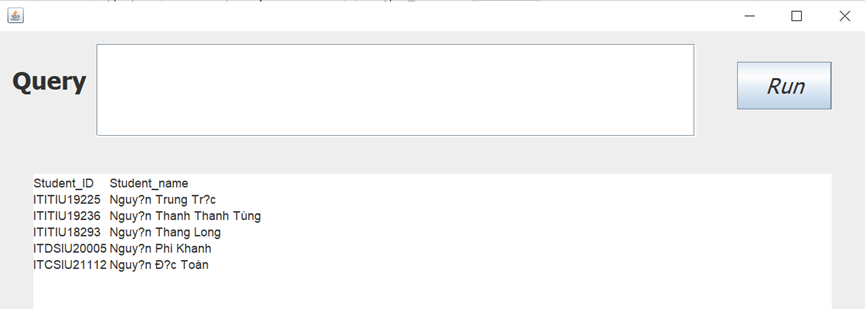
Code: Select Student\_ID, Student\_name From Student2 Where Date\_of\_Birth < '2002-01-01'



## Figure 4.4.

## 3) Select all the student name and ID have sir name is Nguyễn

Code: Select Student\_ID, Student\_name From Student2 Where Student\_name LIKE 'Ng% '

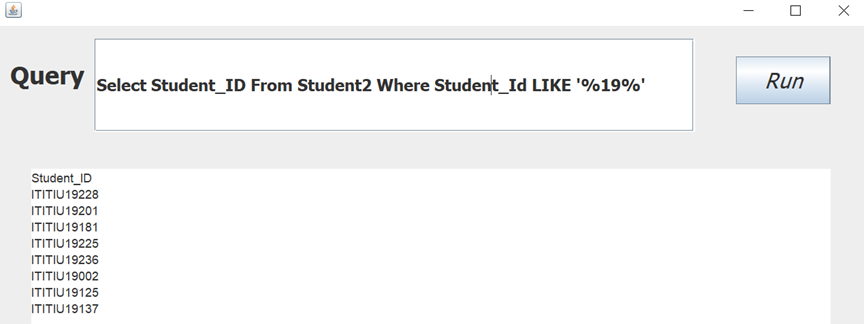


## Figure 4.5.

## 

## 4) Select all the Student\_ID that is K19

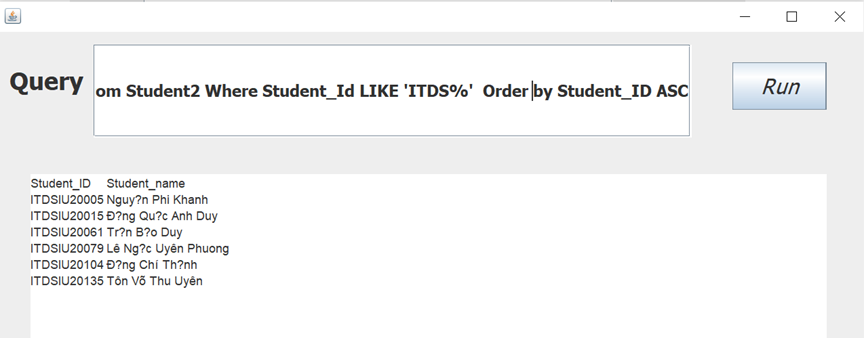
Code: Select Student\_ID From Student2 Where Student\_Id LIKE '%19%'



## Figure 4.6.

## 5) Select all the student name and ID that study DS major in ascending order by ID

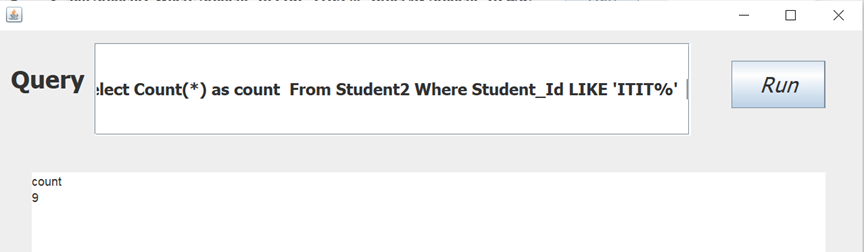
Code: Select Student\_ID,Student\_name From Student2 Where Student\_Id LIKE 'ITDS%' Order by Student\_ID ASC



## Figure 4.7.

## 6) How many students that study\_ID major

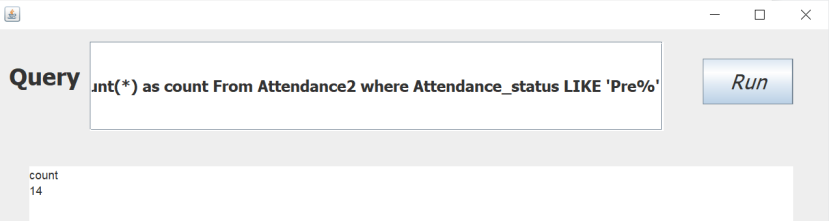
Code: Select Count(\*) as count From Student2 Where Student\_Id LIKE 'ITIT%'



## Figure 4.8.

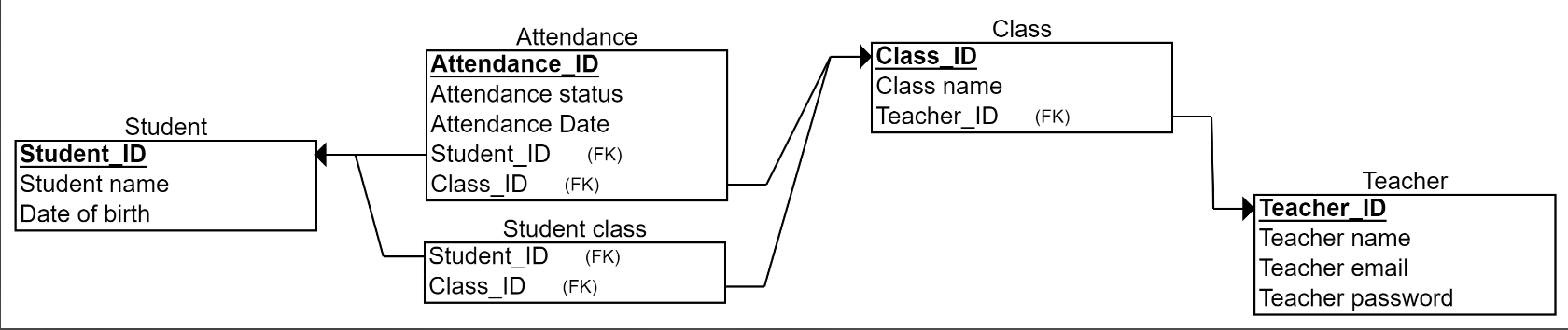
## 7) Count all the Attendance\_ID that is attended in that class.

Code: select count(\*) as count From Attendance2 where Attendance\_status LIKE 'Pre%'



## Figure 4.9.

# V. Normalization



*Figure 5.1. Initial ERD*

## **First Normal Form-1NF:**

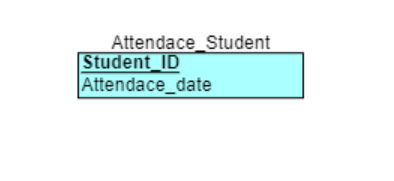
**Definition:** The domain of an attribute must include only atomic values, and the value of any attribute in a tuple must be a single value from the domain of that attribute.

• No relations within relations or relations as attribute values within tuples

• Considered part of the formal definition of a relation in the basic (flat) relational model

In other words, an *implicit* constraint

—----------------------

Since students can attend classes on multiple days, the Attendance Date attribute can contain multiple values in this database. To obtain the first normal form, we must therefore partition it into a separate table. The record Attendance\_Student contains the attribute Attendance Date and the primary key Student's ID. Here is the result:

## Figure 5.2.

## **Second Normal Form-2NF:**

**Definition:** A relation is in second normal form if it is in 1NF and every non key attribute is fully functionally dependent on the primary key.

—----------------------------

Observing the initial ERD, we can see that every table has a single primary key and no set of attributes for primary keys. Moreover, each table's attributes are interrelated and dependent on a single primary key. Consequently, this database conforms to the second normal form.

## **Third Normal Form-3NF:**

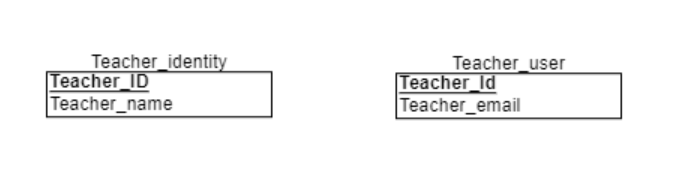
**Definition**: A table design is said to be in 3NF if both the following conditions hold:

• The table must be in 2NF.

• The transitive functional dependency of non-prime attributes on any superkey should be removed.

An attribute that is not part of any candidate key is known as a non-prime attribute.

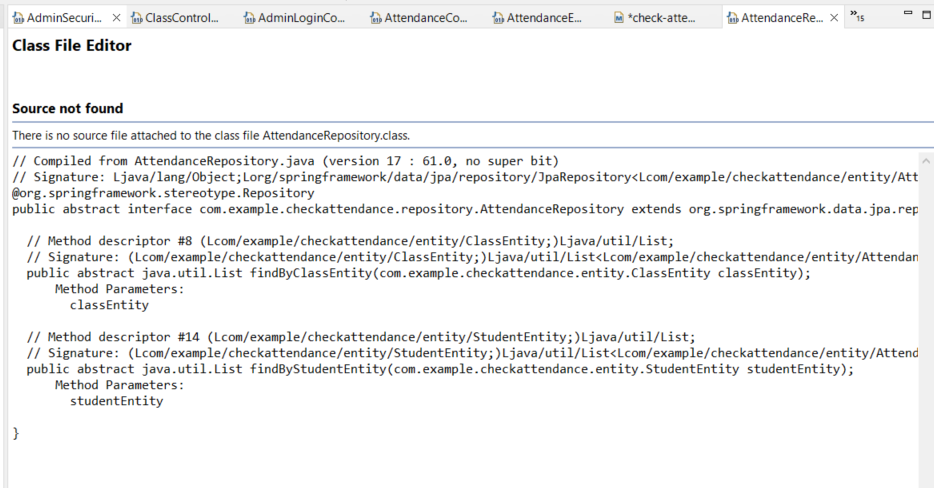
From the initial ERD, we can see that some attributes depend on each other, and they also depend on a primary key (tính chất bắc cầu).In the teacher table, for instance, the attribute email depends on the attribute Teacher\_name. Thus, they all depend on the Teacher\_ID primary key. Therefore, we must separate them into two tables in order to accomplish the Third Normal Form. One name, Teacher\_identity, which includes the Teacher\_name and Teacher\_ID as the primary key. Teacher\_user comprises an email attribute and Teacher\_ID as the primary key in the second table. Here is the result:



## Figure 5.3.

# **VI. Software:**

*Due to the deletion of the incorrect file, we were unable to complete the web component on schedule.*



## Figure 6.1. error image

## **Technologies :**

* Language: JAVA
* Framework: Spring-tool-suite
* Library: Spring Boot, Spring Data JPA, Spring Security
* Database Management System: SQL Server

**Install the library via Maven, the dependencies include:**



The code directory includes:

## Figure 6.2. pom.xml

## Figure 6.3.

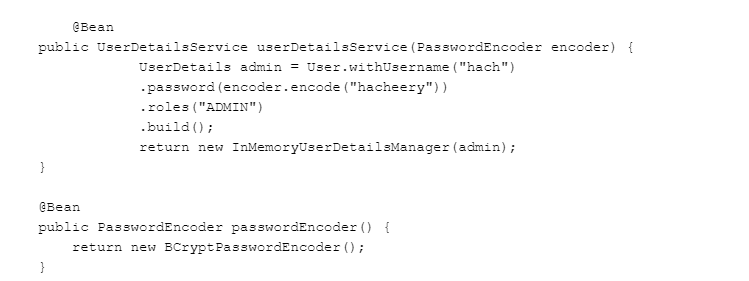
There is also a jwtTokenProvider.

*Note: This direction is old and has not been fixed.*

**Functions of each package:**

-**Configure**: Create an account and assign access rights.

Ex:





## Figure 6.4. AdminSecurityConfig.java

-**Controller**: Receive the request and return the response to the client. The controller can be understood as an intermediary layer between the server and the outside.

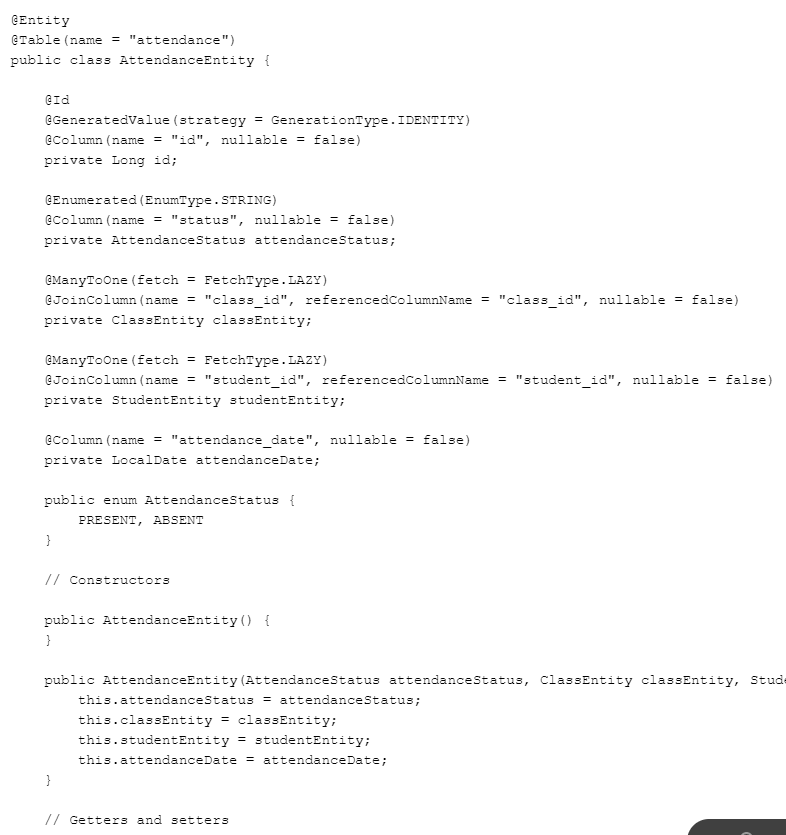
Ex:



## Figure 6.5. AdminLoginController.java

-**Entity**: The domain model corresponds to the table in the DB, which can be mapped to the DB.

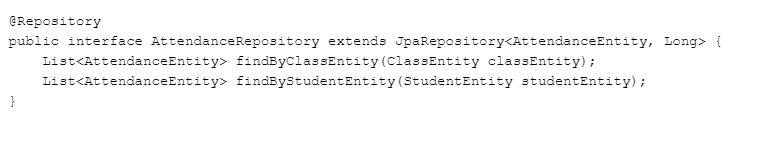
Ex:



## Figure 6.6. AttendanceEntity.java

-**Repository**: data retrieval service

Ex:



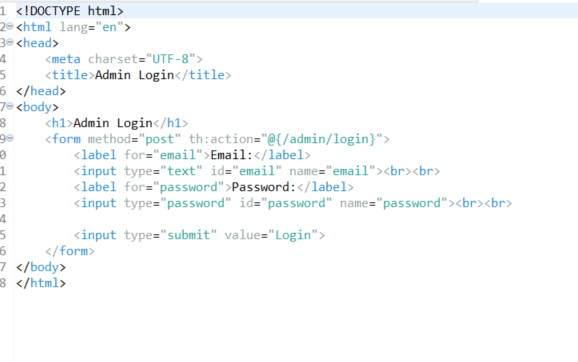
## Figure 6.7. AttendanceRepository.java

-**Service**: serving business logic

*This code is not available in the old file.*

**Template: front-end**

Ex:

*Figure 6.8. admin-login.html*

# **VII. Reference:**

-<https://www.spiceworks.com/hr/workforce-management/articles/features-of-attendance-management-systems/>

-Refer to the data table here: <https://www.slideshare.net/ManojKumar1530/attendance-management-system-project-report>

-Refer to the software:

<https://www.codejava.net/frameworks/spring-boot/connect-to-microsoft-sql-server-examples>

<https://viblo.asia/p/phan-1-spring-boot-30-va-spring-security-60-GAWVpdBYV05>

<https://www.javaguides.net/2019/01/spring-boot-microsoft-sql-server-jpa-hibernate-crud-restful-api-tutorial.html>

# 

# **VIII. Conclusion:**

The development of the Check Attendance System was a challenging but fruitful endeavor. The system is designed to aid educational institutions in managing student attendance records and generating attendance reports. The system provides instructors and administrators with an efficient tool for monitoring attendance and identifying students who may require additional assistance.

Continuous feedback and iterative development were made possible through the project's design. The team successfully designed and implemented the system, meeting all functional and non-functional project requirements and objectives.

The database design, which consisted of the Entity-Relationship Diagram (ERD) and Relational Schema, was an integral part of the project. The database design ensured that the system's data were properly organized and that entity relationships were precisely defined. In addition, the system architecture, front- and back-end components were meticulously designed to guarantee scalability and efficiency.

Overall, the group's Check Attendance System initiative was a tremendous success. This endeavor allowed the team members to refine their skills in project administration, database design, and software development. The project team was able to overcome obstacles and attain achievement by utilizing their knowledge and experience.

In conclusion, the Check Attendance System is a valuable tool for educational institutions to manage attendance records and analyze attendance data. The system offers instructors and administrators an efficient and effective method for tracking attendance and identifying students who may need additional support. The project team successfully designed and implemented the system, meeting all functional and non-functional requirements and project objectives. Our team believes this report will be a valuable resource for educational institutions and software development teams.