VIETNAM NATIONAL UNIVERSITY, HANOI

**INTERNATIONAL SCHOOL**

**INS2055 – DATABASE SYSTEMS**

**PROJECT REPORT**

# STUDENT ENROLLMENT DATABASE AT VNU-IS


**Student Information**

**Nguyen Minh Hieu - 18071471**


**Lecturer: Dr. Tran Thi Oanh – International School of Vietnam National University, Ha Noi**

*Hanoi – 2021*

# CONTENTS

# ABSTRACT

This report uses and applies the knowledge learned from this course to analyze and design a student enrollment database for VNU-IS. Because VNU-IS's whole database is too intensive and takes a lot of work to cover, this report will only focus on the course registration process.

The report includes four main chapters. In the first chapter, I provide a brief introduction about VNU-IS, the sources of information, and specify the business rules along with the business narratives that will affect my process of analyzing and designing the database in the second chapter. Chapter three is where I devise 10 business questions and use SQL to find their answers to support the activities of my organization. Finally, I will point out the strong and weak points of the database and how it can be used inside the organization in chapter four.

# Chapter 1: INTRODUCTION

## 1. About the Organization

- Vietnam National University, Hanoi - International School (VNU - IS) was established in 2002.

- Mission: To provide undergraduate and graduate training entirely in foreign languages and conducting scientific research under international standards and based on basic sciences, applied sciences, and technology transfer, contributing to supplying high-quality human resources, scientific and technological products to serve the cause of national development.

- Vision: To become a VNU's pioneering member institution in higher education, towards interdisciplinary, multidisciplinary training and research, of high international integrity and standards, capable of exporting educational products.

- Programs of VNU-IS are full-time and intensive. VNU degree programs are designed according to the standards of prestigious foreign universities and are officially accredited and approved by VNU. Programs with foreign degrees are subject to and in line with the standards of the host institutions. All programs are approved and recognized by reputed, reliable accreditation agencies abroad and by the Ministry of Education and Training (MOET) in Vietnam.

## 2. Requirement Specification

### Methods for investigating business rules:

The information used in this report is gathered from the following sources

**Websites**

The database is designed mostly bases on the information taken from the VNU-IS students' portal

**Documents**

Before each registration period, emails with the projected timetables, subjects, and lecturers' information will be sent to each student. We use these references to create our tables.

**Observation**

Our experience with VNU_IS course enrollment also shapes the way we design this database.

## 3. Business Narrative

### 3.1. Business Rules

Students must achieve B2 English Certificate to be able to enroll in the subjects on the list.

Students have to enroll only for subjects that are included in the Progam curriculum for their major.

Students can retake a subject as many times as they want.

Students whose average score on the scale of 10 goes below 4 will fail a subject.

## 3.2. Business Narrative

Every student from VNU-IS has a unique Stu_Id and their personal information is stored.

Each subject contains some constant data with Sub_Code as an Identifier. Whether the subject is compulsory or elective will be noted in the Type column.

Each semester, several classes will be offered for each subject for students to choose from.

The information about the time and location at which the class is held, the time for the Final Exam, and the Online code in case of emergency would be changed depending on each Semester and school year.

Each class has one lecturer in charge but one lecturer can take charge of many classes. Lecturers can be changed depending on each Semester and school year.

Each student has 3 attempts to make changes to their enrollment. Only after the 3rd time will the final enrollment be recorded.

After the final timetables have been determined, students have to pay the tuition fees. The tuition fee is decided based on the student's major class. The amount paid for each credit of main subjects and general subjects is different and this may change depending on the semester and school year. These fees will be updated for each class/ subject registered.

Each student has an invoice with a specific In_Id instore, making it easier to manage. After the timetable is set for the particular semester and school year, the invoice will be updated to show the total amount of tuition fee that each student has to pay along with the debt and due date. Each row from the Enrollment table represents one line of the invoice.

Each student can gradually pay off the school fee online. Each payment is recorded with the date and amount paid. A corresponding amount will be subtracted from the Total_debt for each successful payment.

For each class/subject enrolled, student's grades including Mid-term score (30%), Participation score (10%), Final exam score (60%), Average score on the scale of 10, Average score on the scale of 4, Letter grade, Result ("Pass" or "Fail") will be recorded. If the result is "Pass" for a particular class/subject, the number of Stu_Credits will increase an amount equivalent to that Sub_Credits, otherwise, the Student_Credits will remain unchanged.

# Chapter 2: ANALYZING AND DESIGNING THE DATABASE

## Analyzing and designing

### 1. Identify Entities

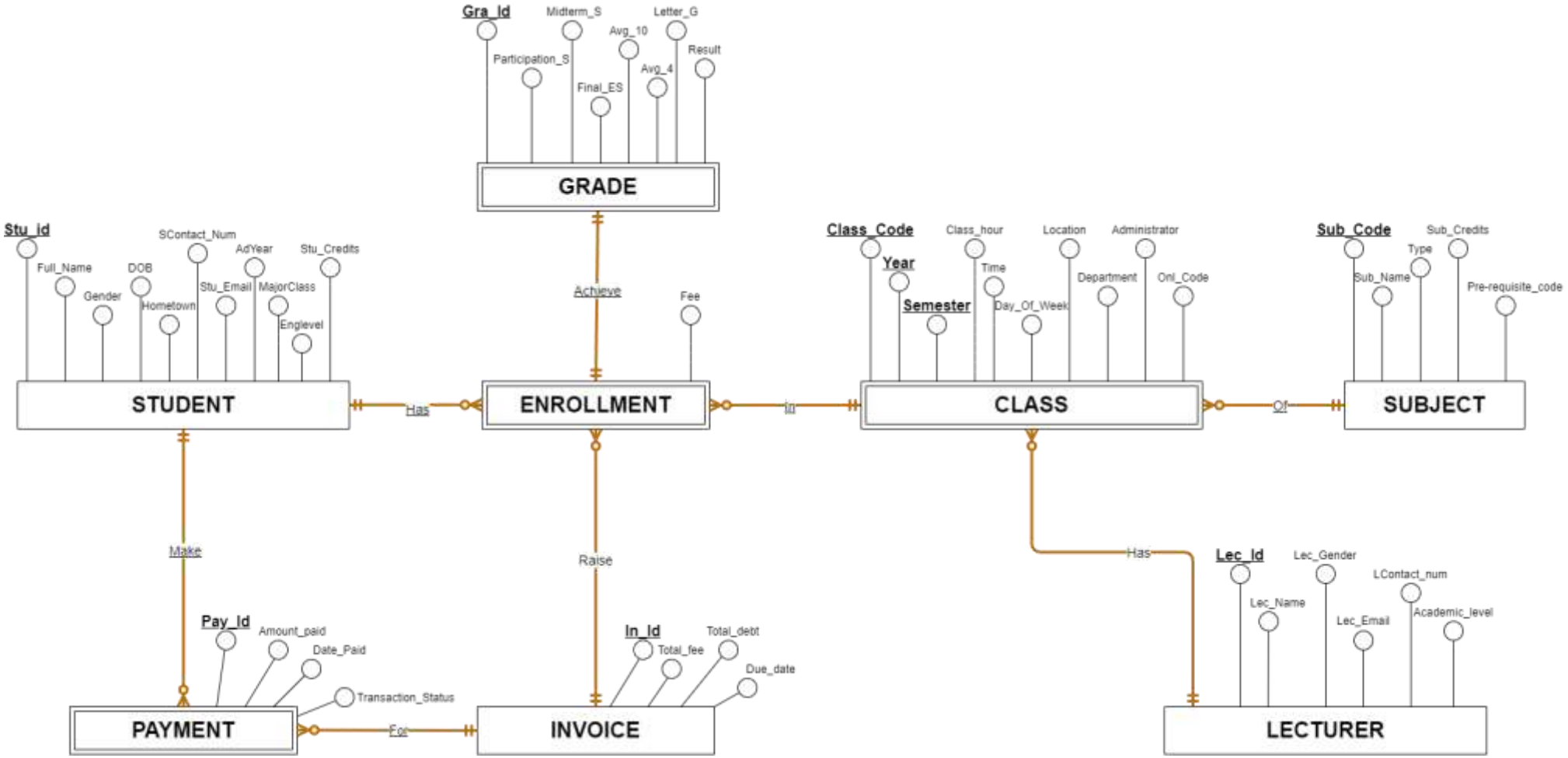
Strong entities: STUDENT, SUBJECT, LECTURER, INVOICE

Weak entities: ENROLLMENT, CLASS, PAYMENT, GRADE

### 2. Cardinality

- Each subject may offer many classes.
  Each class has to belongs to one subject.
- Each lecturer may teach many classes.
  Each class has to have one lecturer.
- Each student may have many enrollments.
  Each enrollment has to come from one student.
- Each class may be included in many enrollments.
  Each enrollment has to be in one class.
- For each registered class, the student has to achieve a grade result.
  Each grade stored has to belong to only one student from a certain class registered.
- Each invoice may contain many enrollment records.
  Each enrollment record has to raise only one invoice (belong to one invoice only).
- Each invoice may be paid many times.
  Each payment has to be for one invoice.
- Each student may make a payment many times.
  Each payment has to come from one student.

# 1. ERD


Gra_Id
Midterm_S
Letter_G
Avg_10
Result
Participation_S
Avg_4
Final_ES
GRADE
Achieve
Fee
Stu_id
SContact_Num
AdYear
Stu_Credits
Full_Name
DOB
Gender
Hometown
Stu_Email
MajorClass
Englevel
STUDENT
Has
ENROLLMENT
In
Class_Code
Class_hour
Location
Administrator
Year
Time
Department
Onl_Code
Semester
Day_Of_Week
CLASS
Of
Sub_Code
Sub_Credits
Type
Sub_Name
Pre-requisite_code
SUBJECT
Make
Raise
Has
Lec_Id
Lec_Gender
LContact_num
Lec_Name
Lec_Email
Academic_level
Pay_Id
Amount_paid
Date_Paid
Transaction_Status
In_Id
Total_debt
Total_fee
Due_date
PAYMENT
For
INVOICE
LECTURER

## 2. Relational Schema

**STUDENT** (Stu_Id, FullName, Stu_Gender, DOB, Hometown, MajorClass, AdYear, EngLevel, Stu_Email, SContact_Num, Stu_Credits)

PRIMARY KEY (Stu_Id)

**LECTURER** (Lec_Id, Lec_Name, Lec_Gender, Academic_Level, Lec_Email, LContact_Num)

PRIMARY KEY (Lec_Id)

**SUBJECT** (Sub_Code, Sub_Name, Sub_Credits, Type, Prerequisite_Code)

PRIMARY KEY (Sub_Code)

**CLASS** (Class_Code, Year, Semester, Sub_Code, Lec_Id, Class_Hour, Time, Day_Of_Week, Location, Department, Administrator, Onl_Code)

PRIMARY KEY (Class_Code, Year, Semester, Sub_code)

FOREIGN KEY (Lec_Id) references LECTURER (Lec_Id)

FOREIGN KEY (Sub_Code) references SUBJECT (Sub_Code)

**ENROLLMENT** (Stu_Id, Class_Code, Year, Semester, Sub_code, Fee)

PRIMARY KEY (Stu_Id, Class_Code, Year, Semester, Sub_code)

FOREIGN KEY (Stu_Id) references STUDENT (Stu_Id)

FOREIGN KEY (Class_Code, Year, Semester, Sub_code) references CLASS (Class_Code, Year, Semester, Sub_code)

**GRADE** (Gra_Id, Stu_Id, Class_Code, Year, Semester, Sub_code, Participation_S, Midterm_S, Final_ES, Avg_10, Avg_4, Letter_G, Result)

PRIMARY KEY (Gra_Id, Stu_Id, Class_Code, Year, Semester, Sub_code)

FOREIGN KEY (Stu_Id, Class_Code, Year, Semester, Sub_code) references ENROLLMENT (Stu_Id, Class_Code, Year, Semester, Sub_code)

**INVOICE** (In_Id, Total_Fee, Total_Debt, Due_Date)

PRIMARY KEY (In_Id)

**PAYMENT** (Pay_Id, Stu_Id, In_Id, Amount_Paid, Date_Paid, Transactions_Status)

PRIMARY KEY (Pay_Id, Stu_Id, In_Id)

FOREIGN KEY (Stu_Id) references STUDENT (Stu_Id)

FOREIGN KEY (In_Id) references INVOICE (In_Id)

## 3. Some real data and SQL scripts

INSERT INTO STUDENT VALUES (18071451, N'Đỗ Việt Anh', 'Male', '2000/06/03', N'Quảng Ninh', 'MIS2018A',3, 'B2', '18071451@vnu.edu.vn', 0388595898, 15);

INSERT INTO STUDENT VALUES (18071452, N'Hoàng Thị Ngọc Anh', 'Female', '2000/6/23', N'Thái Bình', 'MIS2018A', 3, 'B2', '18071452@vnu.edu.vn', 0977768543, 16);

INSERT INTO STUDENT VALUES (18071454, N'Nguyễn Hoàng Anh', 'Male', '2000/10/19', N'Tuyên Quang', 'MIS2018A', 3, 'B2', '18071454@vnu.edu.vn', 0859025735, 14);

INSERT INTO STUDENT VALUES (18071457, N'Nguyễn Tùng Bách', 'Male', '2000/5/14', N'Bắc Giang', 'MIS2018A', 3, Null, '18071457@vnu.edu.vn', 0355491959, Null);

INSERT INTO STUDENT VALUES (18071459, N'Phạm Trung Dũng', 'Male', '2000/9/5', N'Hà Nội', 'MIS2018A', 3, 'B2', '18071459@vnu.edu.vn', 0994489459, 17);

INSERT INTO LECTURER VALUES (1001, N'Nguyễn Thị Hồng Vân', 'Female', 'PhD', 'vannth@ftu.edu.vn', 0903471700);

INSERT INTO LECTURER VALUES (1002, N'Trần Thị Oanh', 'Female', 'PhD', 'oanhtt@isvnu.vn', 0362220684);

INSERT INTO LECTURER VALUES (1003, N'Lê Anh Ngọc', 'Male', 'PhD', 'anhngocle.vn@gmail.com', 0916880777);

INSERT INTO LECTURER VALUES (1004, N'Trần Đức Quỳnh', 'Male', 'PhD', 'quynhtd@isvnu.vn', 0902266018);

INSERT INTO LECTURER VALUES (1005, N'Đoàn Thu Trang', 'Female', 'PhD', 'trangdt@isvnu.vn', 0964404620);

INSERT INTO SUBJECT VALUES ('INS3066', 'Enterprise Business Solutions', 3, 'Compulsory', Null);

INSERT INTO SUBJECT VALUES ('INS2055', 'Database Systems', 4, 'Compulsory', 'INT1004');

INSERT INTO SUBJECT VALUES ('INS3062', 'Principles of Information Security', 3, 'Compulsory', 'INS2025');

INSERT INTO SUBJECT VALUES ('INS2051', 'Quantitative Methods for Management', 3, 'Compulsory', 'MAT1004');

INSERT INTO SUBJECT VALUES ('INS2024', 'Organizational Strategy', 3, 'Elective', 'INS2019');

INSERT INTO CLASS VALUES ('INS3066', '2020-2021', 2, 'INS3066', '1001', '10-12', 'From 3:50 PM to 6:30 PM', 'Thursday', N'Thanh Xuân', 'KHTN&CN', N'Việt Mỹ', '67oe56s');

INSERT INTO CLASS VALUES ('INS2055', '2020-2021', 2, 'INS2055', '1002', '3-6', 'From 8:50 AM to 12:30 PM', 'Thursday', N'Cầu Giấy', 'KHTN&CN', N'Việt Mỹ', 'r5k6ss1');

INSERT INTO CLASS VALUES ('INS3062', '2020-2021', 2, 'INS3062', '1003', '1-3', 'From 7:00 AM to 9:40 AM', 'Thursday', N'Cầu Giấy', 'KHTN&CN', N'Việt Mỹ', 'kf88fw4');

INSERT INTO CLASS VALUES ('INS2051', '2020-2021', 2, 'INS2051', '1004', '1-3', 'From 7:00 AM to 9:40 AM', 'Tuesday', N'Thanh Xuân', 'KHTN&CN', N'Việt Mỹ', 'ld9kgd2');

INSERT INTO CLASS VALUES ('INS2024_1', '2020-2021', 2, 'INS2024', '1005', '4-6', 'From 9:50 AM to 12:30 PM', 'Tuesday', N'Thanh Xuân', 'KHXH, KT&QL', N'Việt Mỹ', 'l4yuowe');

INSERT INTO ENROLLMENT VALUES (18071451, 'INS3066', '2020-2021', 2, 'INS3066', 3843000);

INSERT INTO ENROLLMENT VALUES (18071451, 'INS2024_2', '2020-2021', 2, 'INS2024' 3843000);

INSERT INTO ENROLLMENT VALUES (18071451, 'INS2051', '2020-2021', 2, 'INS2051', 3843000);

INSERT INTO ENROLLMENT VALUES (18071451, 'INS1004', '2020-2021', 2, 'INS1004', 5124000);

INSERT INTO ENROLLMENT VALUES (18071451, 'SOC1050', '2020-2021', 2, 'SOC1050', 2562000);

INSERT INTO GRADE VALUES ('1451_1', 18071451, 'INS3066', '2020-2021', 2, 'INS3066', 7, 6, 6.5, 6.4, 2, 'C', 'Passed');

INSERT INTO GRADE VALUES ('1451_2', 18071451, 'INS2024_2', '2020-2021', 2, 'INS2024', 10, 8, 6, 7, 3, 'B', 'Passed');

INSERT INTO GRADE VALUES ('1451_3', 18071451, 'INS2051', '2020-2021', 2, 'INS2051', 10, 7, 1.5, 4, 1, 'D', 'Passed');

INSERT INTO GRADE VALUES ('1451_4', 18071451, 'INS1004', '2020-2021', 2, 'INS1004', 10, 8, 8, 8.2, 3.5, 'B+', 'Passed');

INSERT INTO GRADE VALUES ('1451_5', 18071451, 'SOC1050', '2020-2021', 2, 'SOC1050', 9, 7, 7.5, 7.5, 3, 'B', 'Passed');

INSERT INTO INVOICE VALUES ('IN1451', 19215000, 6966000, '2021/5/9');

INSERT INTO INVOICE VALUES ('IN1452', 20496000, 0, '2021/5/9');

INSERT INTO INVOICE VALUES ('IN1454', 17934000, 0, '2021/5/9');

INSERT INTO INVOICE VALUES ('IN1459', 21777000, 10449000, '2021/5/9');

INSERT INTO INVOICE VALUES ('IN1464', 26901000, 0, '2021/5/9');

INSERT INTO PAYMENT VALUES (1451, 18071451, 'IN1451', 12249000, '2021/5/1', 'Successful ');

INSERT INTO PAYMENT VALUES (1452, 18071452, 'IN1452', 20496000, '2021/5/5', 'Successful ');

INSERT INTO PAYMENT VALUES (1454, 18071454, 'IN1454', 17934000, '2021/5/5', 'Successful ');

INSERT INTO PAYMENT VALUES (1459, 18071459, 'IN1459', 11328000, '2021/5/1', 'Successful ');

INSERT INTO PAYMENT VALUES (1464, 18071464, 'IN1464', 26901000, '2021/5/2', 'Successful ');

# Chapter 3: SOME USEFUL REPORTS AND INFORMATION

## 1. Design some useful reports and information for real usage

**1)** Get the information from the faculty to see what subjects each lecturer teaches during the semester?

**2)** Count the number of A or A+ grades of subjects in the semester.

**3)** Manage the amount of tuition paid from students (VIEW).

**4)** The student with student ID 18071451 pays the remaining tuition fees (TRANSACTION).

**5)** Which students will be eligible for a scholarship given the GPA (on a scale of 4) must be greater than or equal to 3.2?

**6)** Retrieve student information by hometown name (STORED PROCEDURE).

**7)** Count the number of tuition payment transactions between May 1st and May 3rd.

**8)** Updating student accumulated credits bases on the result of each registered subject (VIEW, TRIGGER, STORED PROCEDURE).

## 2. Solutions for the above reports/information

**1)**
```
SELECT L.Lec_Id, L.Lec_Name, C.Class_Code ,S.Sub_Code, S.Sub_Name
FROM LECTURER L
INNER JOIN CLASS C
ON L.Lec_Id = C.Lec_Id
INNER JOIN SUBJECT S
ON S.Sub_Code = C.Sub_Code
ORDER BY Lec_Id ASC;
```

| Lec_Id | Lec_Name | Class_Code | Sub_Code | Sub_Name |
|---|---|---|---|---|
| 1001 | Nguyễn Thị Hồng Vân | INS3066 | INS3066 | Enterprise Business Solutions |
| 1002 | Trần Thị Oanh | INS2055 | INS2055 | Database Systems |
| 1002 | Trần Thị Oanh | INS2061 | INS2061 | Data Mining & Business Analytics |
| 1002 | Trần Thị Oanh | INS3063 | INS3063 | Enterprise Analytics for Decision Support |
| 1003 | Lê Anh Ngọc | INS3062 | INS3062 | Principles of Information Security |
| 1004 | Trần Đức Quỳnh | INS2051 | INS2051 | Quantitative Methods for Management |
| 1005 | Đoàn Thu Trang | INS2024_1 | INS2024 | Organizational Strategy |
| 1005 | Đoàn Thu Trang | INS2024_2 | INS2024 | Organizational Strategy |
| 1006 | Nguyễn Văn Tánh | INS1004 | INS1004 | Introduction to Computer Networks |
| 1007 | Trương Công Đoàn | INS3055 | INS3055 | Introduction to Object-Oriented Systems Devel... |
| 1007 | Trương Công Đoàn | INS3059 | INS3059 | Project Management |
| 1008 | Lê Duy Tiến | INS2025_1 | INS2025 | Information Systems in Organizations |
| 1008 | Lê Duy Tiến | INS2025_2 | INS2025 | Information Systems in Organizations |
| 1009 | Nguyễn Đình Văn | INS3056 | INS3056 | Information Systems Modeling and Design |
| 1010 | Trần Công Thành | INS2023_1 | INS2023 | Operations Management |
| 1010 | Trần Công Thành | INS2023_2 | INS2023 | Operations Management |
| 1011 | Nguyễn Đức Nam | PSY1050 | PSY1050 | Introduction to Psychology |
| 1012 | Nguyễn Thanh Tùng | INS2053 | INS2053 | Web Authoring and Web Management |
| 1013 | Vũ Xuân Đoàn | SOC1050 | SOC1050 | Introduction to Sociology |

**2)** SELECT C.Class_Code, S.Sub_Name, G.Letter_G, COUNT(*) AS NumberOfLetterGrade
FROM CLASS C
INNER JOIN SUBJECT S
ON C.Sub_Code = S.Sub_Code
INNER JOIN GRADE G
ON C.Class_Code = G.Class_Code
GROUP BY G.Letter_G, C.Class_Code
HAVING G.Letter_G = 'A' OR G.Letter_G = 'A+'
ORDER BY G.Letter_G DESC;

| Class_Code | Sub_Name | Letter_G | NumberOfLetterGrade |
|---|---|---|---|
| INS3055 | Introduction to Object-Oriented Systems Devel... | A+ | 1 |
| INS3059 | Project Management | A+ | 1 |
| INS3063 | Enterprise Analytics for Decision Support | A+ | 1 |
| SOC1050 | Introduction to Sociology | A+ | 2 |
| INS2053 | Web Authoring and Web Management | A | 1 |
| INS2061 | Data Mining & Business Analytics | A | 1 |
| INS3056 | Information Systems Modeling and Design | A | 1 |
| INS3066 | Enterprise Business Solutions | A | 1 |
| SOC1050 | Introduction to Sociology | A | 1 |

**3)** CREATE VIEW FeesPaymentFromStudent AS
SELECT S.Stu_Id, S.FullName, I.Total_Fee, P.Amount_Paid, I.Total_Debt
FROM STUDENT S
INNER JOIN PAYMENT P
ON S.Stu_Id = P.Stu_Id
INNER JOIN INVOICE I
ON I.In_Id = P.In_Id
ORDER BY Total_Debt ASC;

SELECT * FROM FeesPaymentFromStudent;

| Stu_Id | FullName | Total_Fee | Amount_Paid | Total_Debt |
|---|---|---|---|---|
| 18071452 | Hoàng Thị Ngọc Anh | 20496000 | 20496000 | 0 |
| 18071454 | Nguyễn Hoàng Anh | 17934000 | 17934000 | 0 |
| 18071464 | Trần Vũ Minh Hà | 26901000 | 26901000 | 0 |
| 18071465 | Phạm Thị Mỹ Hạnh | 23058000 | 23058000 | 0 |
| 18071468 | Nguyễn Thị Thu Hiền | 23058000 | 23058000 | 0 |
| 18071472 | Hà Quốc Huy | 20496000 | 20496000 | 0 |
| 18071476 | Cao Đình Khôi | 19215000 | 19215000 | 0 |
| 18071477 | Hà Thanh Lâm | 23058000 | 23058000 | 0 |
| 18071486 | Nguyễn Thảo My | 21777000 | 21777000 | 0 |
| 18071503 | Lê Thị Thoan | 23058000 | 23058000 | 0 |
| 18071493 | Nguyễn Nam Sơn | 19215000 | 16653000 | 2562000 |
| 18071481 | Nguyễn Hoàng Long | 23058000 | 17934000 | 5124000 |
| 18071484 | Nguyễn Thị Lý | 23058000 | 17934000 | 5124000 |
| 18071451 | Đỗ Việt Anh | 19215000 | 12249000 | 6966000 |
| 18071459 | Phạm Trung Dũng | 21777000 | 11328000 | 10449000 |

**4)** START TRANSACTION;

INSERT INTO PAYMENT

VALUES (2451, 18071451, 'IN1451', 6966000, '2021/05/08', 'Successful');

UPDATE INVOICE

SET Total_Debt = Total_Debt - 6966000

WHERE In_Id = 'IN1451';

COMMIT;

**5)** SELECT S.Stu_Id, Fullname , AVG(Avg_4) AS AverageScoreOnScale4
FROM STUDENT S
INNER JOIN ENROLLMENT E
ON S.Stu_Id = E.Stu_Id
INNER JOIN GRADE G
ON S.Stu_Id = G.Stu_Id
AND E.Class_Code = G.Class_Code
GROUP BY Stu_Id
HAVING AVG(Avg_4) >= 3.2;

| Stu_Id | Fullname | AverageScoreOnScale4 |
|---|---|---|
| 18071464 | Trần Vũ Minh Hà | 3.414285727909633 |
| 18071486 | Nguyễn Thảo My | 3.700000007947286 |

**6)** DELIMITER //
CREATE PROCEDURE GetStuInfoByHometownName (
IN HometownName NVARCHAR(15)
)
BEGIN
SELECT *
FROM STUDENT
WHERE Hometown = HometownName;
END //
DELIMITER ;
CALL GetStuInfoByHometownName('Hà Nội');

| Stu_Id | FullName | Stu_Gender | DOB | Hometown | MajorClass | AdYear | EngLevel | Stu_Email | SContact_Num | Stu_Credits |
|---|---|---|---|---|---|---|---|---|---|---|
| 18071459 | Phạm Trung Dũng | Male | 2000-09-05 | Hà Nội | MIS2018A | 3 | B2 | 18071459@vnu.edu.vn | 994489459 | 17 |
| 18071464 | Trần Vũ Minh Hà | Female | 2000-11-11 | Hà Nội | MIS2018A | 3 | B2 | 18071464@vnu.edu.vn | 387780003 | 21 |
| 18071484 | Nguyễn Thị Lý | Female | 2000-11-17 | Hà Nội | MIS2018A | 3 | B2 | 18071484@vnu.edu.vn | 972162821 | 18 |
| 18071486 | Nguyễn Thảo My | Female | 2000-07-18 | Hà Nội | MIS2018A | 3 | B2 | 18071486@vnu.edu.vn | 396581640 | 17 |
| 18071502 | Phùng Khắc Thiện | Male | 2000-03-07 | Hà Nội | MIS2018A | 3 | NULL | 18071502@vnu.edu.vn | 971127072 | NULL |

**7)** SELECT Date_Paid, COUNT(*) AS NumberOfTransaction
FROM PAYMENT
WHERE Date_Paid
BETWEEN ('2021-05-01') AND ('2021-05-03')
GROUP BY Date_Paid;

| | Date_Paid | NumberOfTransaction |
|---|---|---|
| ▶ | 2021-05-01 | 2 |
| | 2021-05-02 | 4 |
| | 2021-05-03 | 1 |

**8)** DROP VIEW IF EXISTS Class_credits;
CREATE VIEW Class_credits AS
SELECT G.Gra_Id, G.Class_Code, Sub_Credits
FROM GRADE G
INNER JOIN ENROLLMENT E
ON G.Stu_Id = E.Stu_Id
AND G.Class_Code = E.Class_Code
INNER JOIN CLASS C
ON G.Class_Code = C.Class_Code
INNER JOIN SUBJECT S
ON C.Sub_Code = S.Sub_code;

---

DROP PROCEDURE IF EXISTS Cre_update;
DELIMITER //
CREATE PROCEDURE Cre_update (
IN GradeID CHAR(6),
IN StudentID CHAR(8)
)
BEGIN
DECLARE Finalresult CHAR(6);

```
DECLARE Creditsnum INT;

SELECT Result
INTO Finalresult
FROM GRADE
WHERE Gra_Id = GradeID;

SELECT Sub_Credits
INTO Creditsnum
FROM Class_credits
WHERE Gra_Id = GradeID;

IF Finalresult = 'Passed'  THEN
        UPDATE student
        SET Stu_Credits = Stu_Credits + Creditsnum
        WHERE Stu_Id = StudentID;
END IF;
END //
DELIMITER ;
```

---

```
DROP TRIGGER IF EXISTS Credit_update;
DELIMITER //
CREATE TRIGGER Credit_update
AFTER INSERT ON GRADE
FOR EACH ROW
BEGIN
CALL Cre_update ( NEW.Gra_Id, NEW.Stu_Id);
END //
DELIMITER ;
```

---

```
INSERT INTO GRADE VALUES ('1503_6', 18071503, 'SOC1050', '2020-2021', 2,
'SOC1050', 10, 9, 9, 9.1, 4, 'A+', 'Passed');
SELECT * FROM student WHERE Stu_Id = '18071503';
```

*Student '18071503' 's number of accumulated credits before her class 'SOC1050' grade is updated.*

| Stu_Id | FullName | Stu_Gender | DOB | Hometown | MajorClass | AdYear | EngLevel | Stu_Email | SContact_Num | Stu_Credits |
|---|---|---|---|---|---|---|---|---|---|---|
| 18071503 | Lê Thị Thoan | Female | 2000-09-12 | Bắc Ninh | MIS2018A | 3 | B2 | 18071503@vnu.edu.vn | 375483348 | 18 |

*Student '18071503' 's number of accumulated credits after her class 'SOC1050' grade* is updated. *(SOC1050's credits earned is 2)*

| Stu_Id | FullName | Stu_Gender | DOB | Hometown | MajorClass | AdYear | EngLevel | Stu_Email | SContact_Num | Stu_Credits |
|---|---|---|---|---|---|---|---|---|---|---|
| 18071503 | Lê Thị Thoan | Female | 2000-09-12 | Bắc Ninh | MIS2018A | 3 | B2 | 18071503@vnu.edu.vn | 375483348 | 20 |

# Chapter 4: CONCLUSION

This report has given a representative picture of the student enrollment process in VNU-IS. I have applied the business rules and business narratives in the designed database and come up with some solutions to practical business tasks using a wide range of statements from MySQL such as SELECT FROM WHERE ORDER BY GROUP BY HAVING, TRIGGER, VIEW, PROCEDURE, TRANSACTION, etc.

This report is created with my own experience as VNU-IS students after thorough research. Nevertheless, I am well aware of the shortcomings and limited scope of my study since I am only undergraduates.

This database can help VNU-IS store and manage data about their students, subjects, classes offered, lecturers, student enrollment, their grades, and tuition payment for each semester from each school year.

# REFERENCES

VNU - International School-Vietnam National University, Hanoi - International School

VIETNAM NATIONAL UNIVERSITY (vnu.edu.vn)

Cổng thông tin sinh viên (isvnu.vn)

diagrams.net