



**The Faculty of Information and Communication Technology  
Mahidol University**

**Project Assignment; Phase II: From Design to Implementation  
M3, M4**

**Chem Ou Center tutorial school in Thailand  
Business Domain: Education**

**Members:**

6388038	Jirapat	Deesara
6488176	Jidapa	Moolkaew
6488179	Ponnnapassorn	Iamborisut
6488181	Thadeeya	Duangkaew

Section 1, Group 4

ITCS 241 Database Management Systems  
Semester 1, 2022

## Chem Ou Center tutorial school in Thailand

### Business Domain: Education

## Overview

### Business Process:

The enrollment process from Chem Ou Center tutorial school has responsibility for overseeing students who wish to enroll in subjects [1]. Students are able to study on their own in two ways: The first way is to enroll in a course at the counter and pay in cash at Krung Thai Bank. The second way is through the mobile app or on the website. In terms of payment, you can pay in many ways such as Krung Thai Bank, Counter Service, Online credit/debit card, or Mobile Banking.

### Activity:

#### 1. Enrollment course [2]:

##### 1.1. Register at the branch

first type of registration is to go to a branch to request a registration paper and then enroll by payment through Krung Thai Bank.

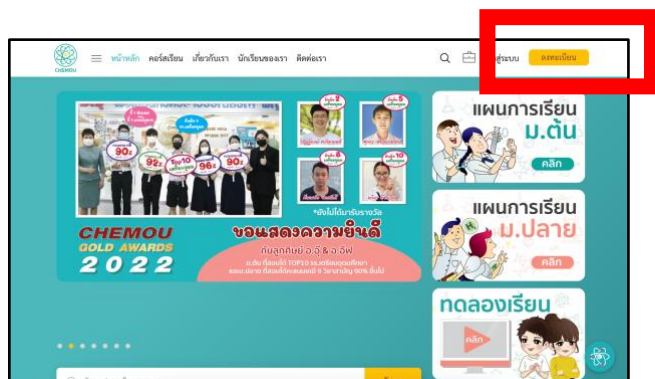


	<b>โรงเรียนกวดวิชารรณธรรม (เคมี อ.อู๋)</b> ใบแจ้งของทะเบียนเรียนผ่านธนาคารกรุงไทยด้วย เงินสด	
คอร์ส (Insti Code)	<input type="text"/>	จำนวนเงิน (Amount).....
ชื่อนักเรียน (Custo Name).....	เบอร์โทรศัพท์ (Ref No.1).....	
<p>คำเตือน 1. กรุณาเขียนด้วยตัวบรรจงให้อ่านได้ถูกต้องชัดเจน</p> <p>2. ต้องตรวจสอบคอร์สที่ลงทะเบียนก่อนและหลังลงทะเบียนว่าถูกต้องตามต้องการอย่างรอบคอบ เพราะหากผิดพลาดจะทำให้ผู้ลงทะเบียนเสียเวลาในการแก้ไขกับธนาคารภายหลัง ซึ่งโรงเรียนจะไม่รับผิดชอบ ต้องรีบแก้ไขกับธนาคารทันทีเมื่อพบว่าผิดพลาด</p> <p>3. ใบแจ้งของทะเบียนเรียนใช้สำกรับ 1 ใบ ต่อ 1 คอร์สเท่านั้น</p>		

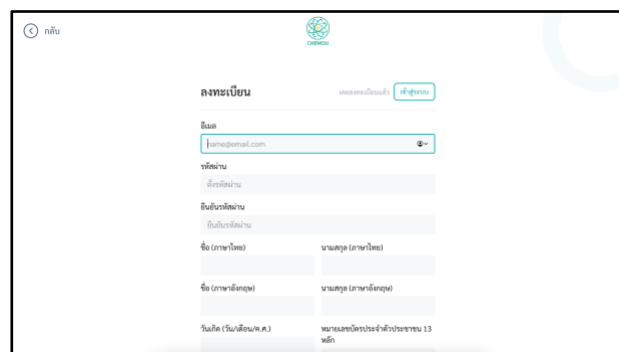
## 1.2. Online platform:

The second type of registration in the online platform can be done in two ways: through the ChemOu website or through the ChemOu application.

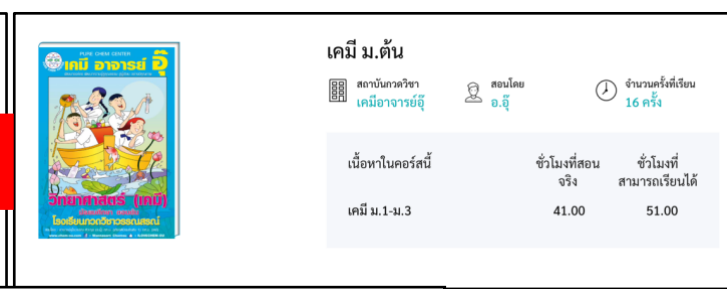
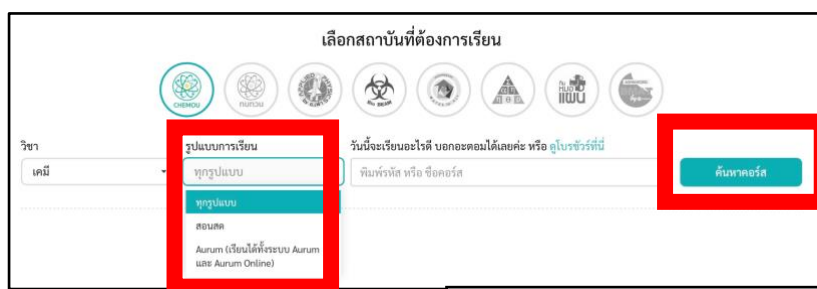
### 1.2.1. Website: [ChemOu](https://chemou.com) [3]



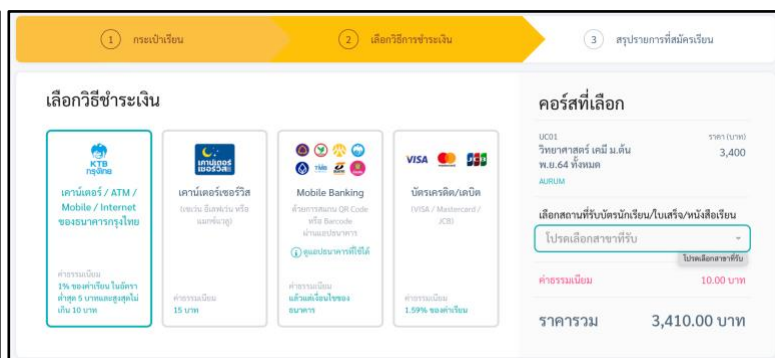
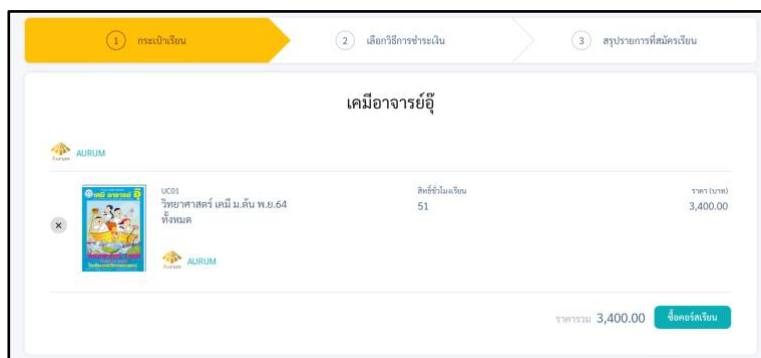
1. Go to Chem-Ou website



2. Register for new students to create a new account.



3. Select the course and types: live, DVD, or Aurum online



4. Select the branch and choose type that you want to pay for the course.

### 1.2.2. ChemOu application [4]:



1. Download CHEMOU Application



2.1. Log in with a student account that already has an account.



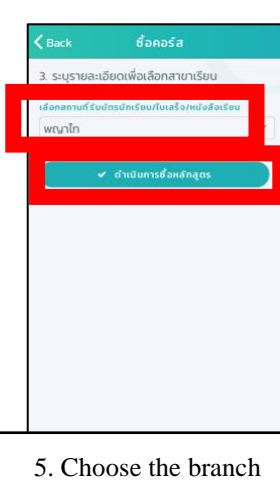
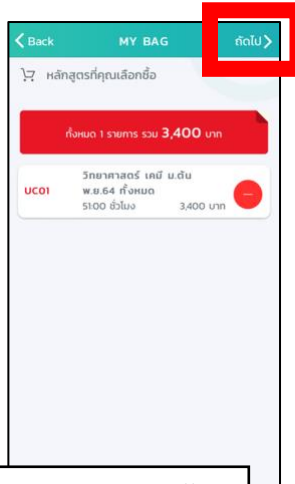
2.2 Register for new students to create a new account.



3. Select the types: live, DVD, or Aurum online that you want to study.



4. Select the course that you want to enroll.



5. Choose the branch that you want to enroll.



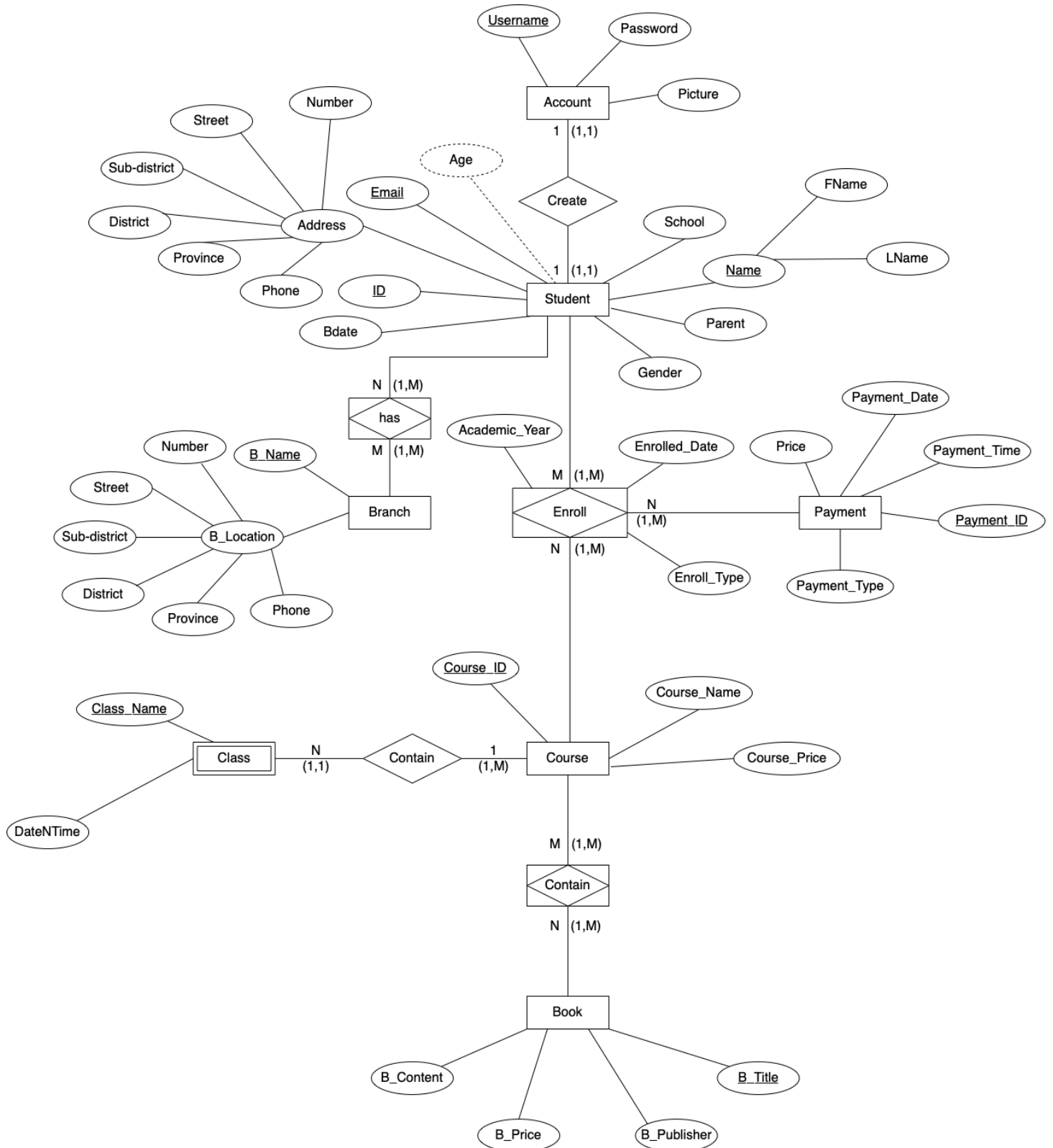
6. Choose type of payment that you want to pay for a course.

2. Apply for an enrollment at the company's branch.

### 3. Payment methods:

The process of paying for enrollment can be done through three ways: Krung Thai counter/ Counter service/ Online banking

## Entity-Relationship Diagram



## 8-steps to transform ERD to relational schema

### Step 1: Transform regular entity types

#### Student

<u>S_ID</u>	S_Email	S_Fname	S_Lname	S_Number	S_Street	S_Subdistrict	S_District
-------------	---------	---------	---------	----------	----------	---------------	------------

S_Province	S_Phone	S_Gender	S_School	S_Parent	S_BDate
------------	---------	----------	----------	----------	---------

#### Account

<u>A_Username</u>	A_Password	A_Picture
-------------------	------------	-----------

#### Branch

<u>B_Name</u>	B_Number	B_Street	B_Subdistrict	B_District	B_Province	B_Phone
---------------	----------	----------	---------------	------------	------------	---------

#### Class

<u>Class_Name</u>	DateNTime
-------------------	-----------

#### Course

<u>Course_ID</u>	Course_name	Course_price
------------------	-------------	--------------

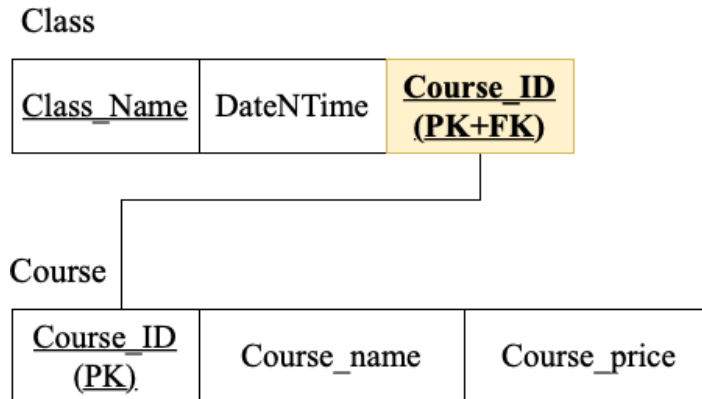
#### Book

<u>B_Title</u>	B_Content	B_Publisher	B_Price
----------------	-----------	-------------	---------

#### Payment

<u>Payment_ID</u>	Payment_Date	Payment_Time	Payment_Type	Price
-------------------	--------------	--------------	--------------	-------

### Step 2: Transforming weak entities



- Course ID, Course Name, and Course Price (in Entity Class) is a foreign key that refers to the Course ID, Course Name, and Course Price (primary) key in the relation Course.
- Course ID, Course Name, and Course Price (in Entity Class) must be a part of a combination of primary keys in the relation Class.

### Step 3: Transforming binary 1:1 relationship type

Student

<u>S_ID</u>	S_Email	S_Fname	S_Lname	S_Number	S_Street	S_Subdistrict	S_District
S_Province	S_Phone	S_Gender	S_School	S_Parent	S_BDate	<b>A_Username (FK)</b>	

Account

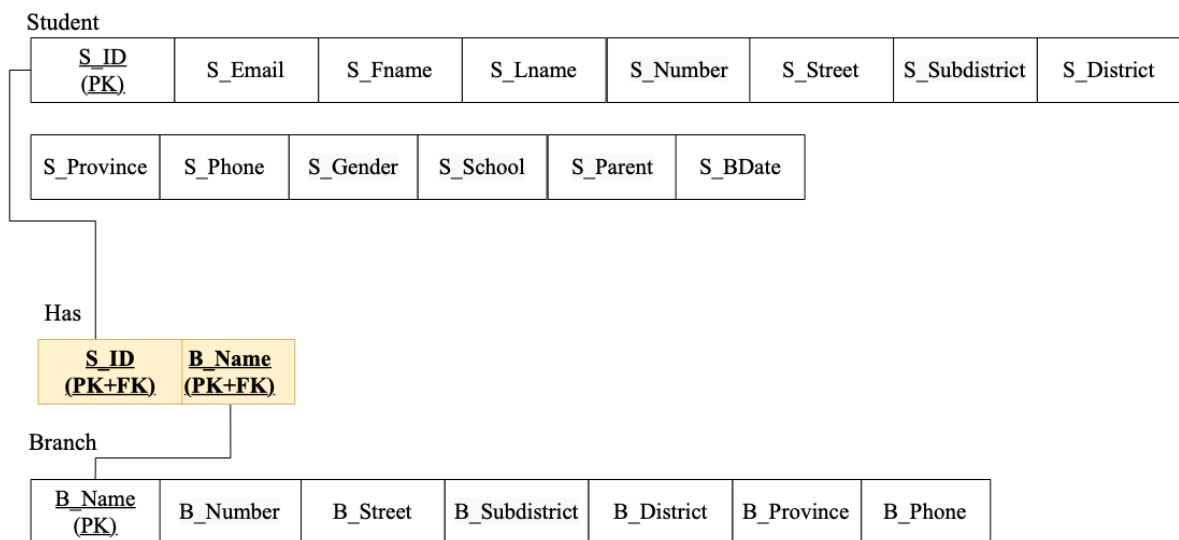
<b><u>A_Username</u> (PK)</b>	A_Password	A_Picture
-----------------------------------	------------	-----------

- AUsername (in Entity Student) is a foreign key that refers to the AUsername (primary) key in the relation Account.
- AUsername (in Entity Student) is not a part of the primary key in relation Student.

#### Step 4: Transforming binary 1:M relationship types

**Detail:** This step uses PK and FK in Entity Course and Entity Class. So, we have already added PK of Entity Course to FK in Entity Class in Step 2. Therefore, we don't do it again in Step 4 because it will be redundant.

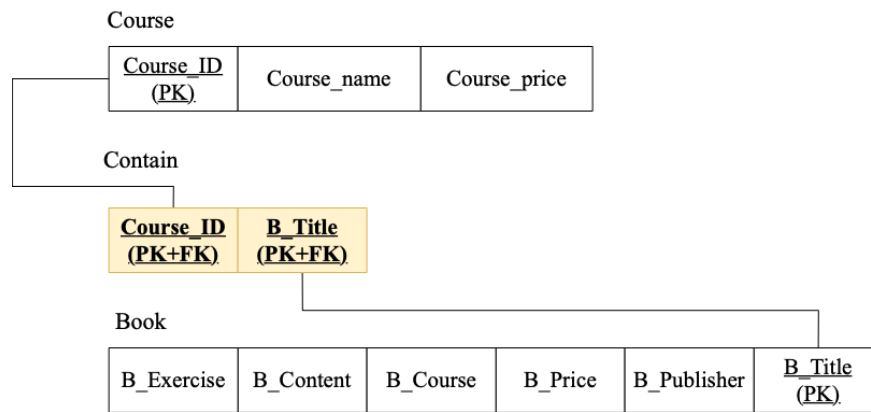
#### Step 5: Transforming binary M:N relationship types (Part1)



- S\_ID, and S Email (in Entity Has) is a foreign key that refers to the S\_ID, and S Email (primary) key in the relation Student.
- B\_Name (in Entity Has) is a foreign key that refers to the B\_Name (primary) key in the relation Branch.
- Both S ID, S Email, and B Name together must form a primary key in the relation Has.



### Step 5: Transforming binary M:N relationship types (Part2)

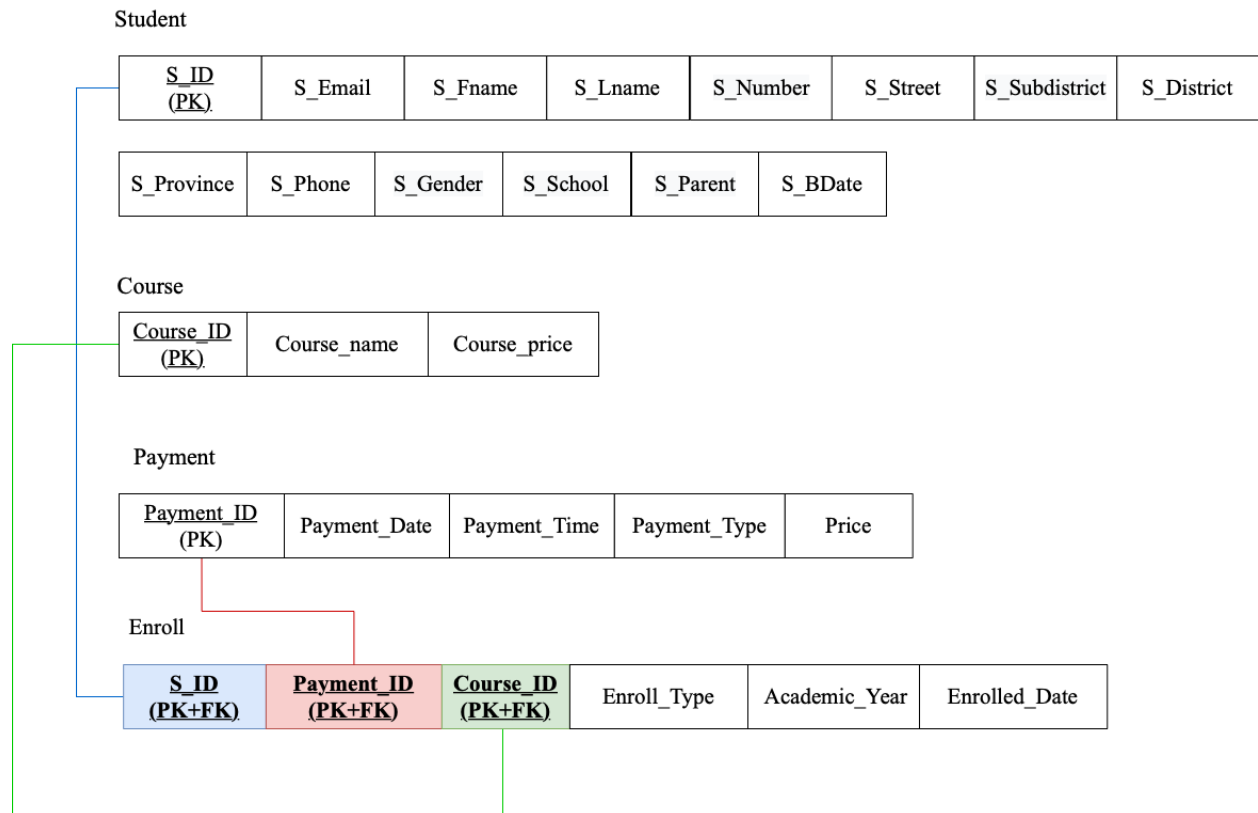


- Course ID, Course Name, and Course Price (in Entity Contain) is a foreign key that refers to the Course ID, Course Name, and Course Price (primary) key in the relation Course.
- B\_Title (in Entity Contain) is a foreign key that refers to the B\_ Title (primary) key in the relation Book.
- Both Course ID, Course Name, Course Price, and B\_ Title together must form a primary key in the relation Contain.

### Step 6: Transforming multi-valued attribute A in relation K

-

### Step 7: Transforming n-ary relationship type



- Each student enrolled several courses for any one of many payments.
- Each course has many payments from any one of many students.

### Step 8: Transforming specialization or generalization

-

### Final Relational Schema

# Student

<b>S_ID (PK)</b>	S_Email	S_Fname	S_Lname	S_Number	S_Street	S_Subdistrict	S_District
------------------	---------	---------	---------	----------	----------	---------------	------------

S_Province	S_Phone	S_Gender	S_School	S_Parent	S_BDate	<b>A_username (FK)</b>
------------	---------	----------	----------	----------	---------	------------------------

# Account

<b>A_username (PK)</b>	A_Password	A_Picture
------------------------	------------	-----------

# Branch

<b>B_Name (PK)</b>	B_Number	B_Street	B_Subdistrict	B_District	B_Province	B_Phone
--------------------	----------	----------	---------------	------------	------------	---------

# Class

Class_Name	DateNTIME	<b>Course_ID (PK+FK)</b>
------------	-----------	--------------------------

# Course

<b>Course_ID (PK)</b>	Course_name	Course_price
-----------------------	-------------	--------------

# Book

B_Exercise	B_Content	B_Course	B_Price	B_Publisher	<b>B_Title (PK)</b>
------------	-----------	----------	---------	-------------	---------------------

# Payment

<b>Payment_ID (PK)</b>	Payment_Date	Payment_Time	Payment_Type	Price
------------------------	--------------	--------------	--------------	-------

# Has

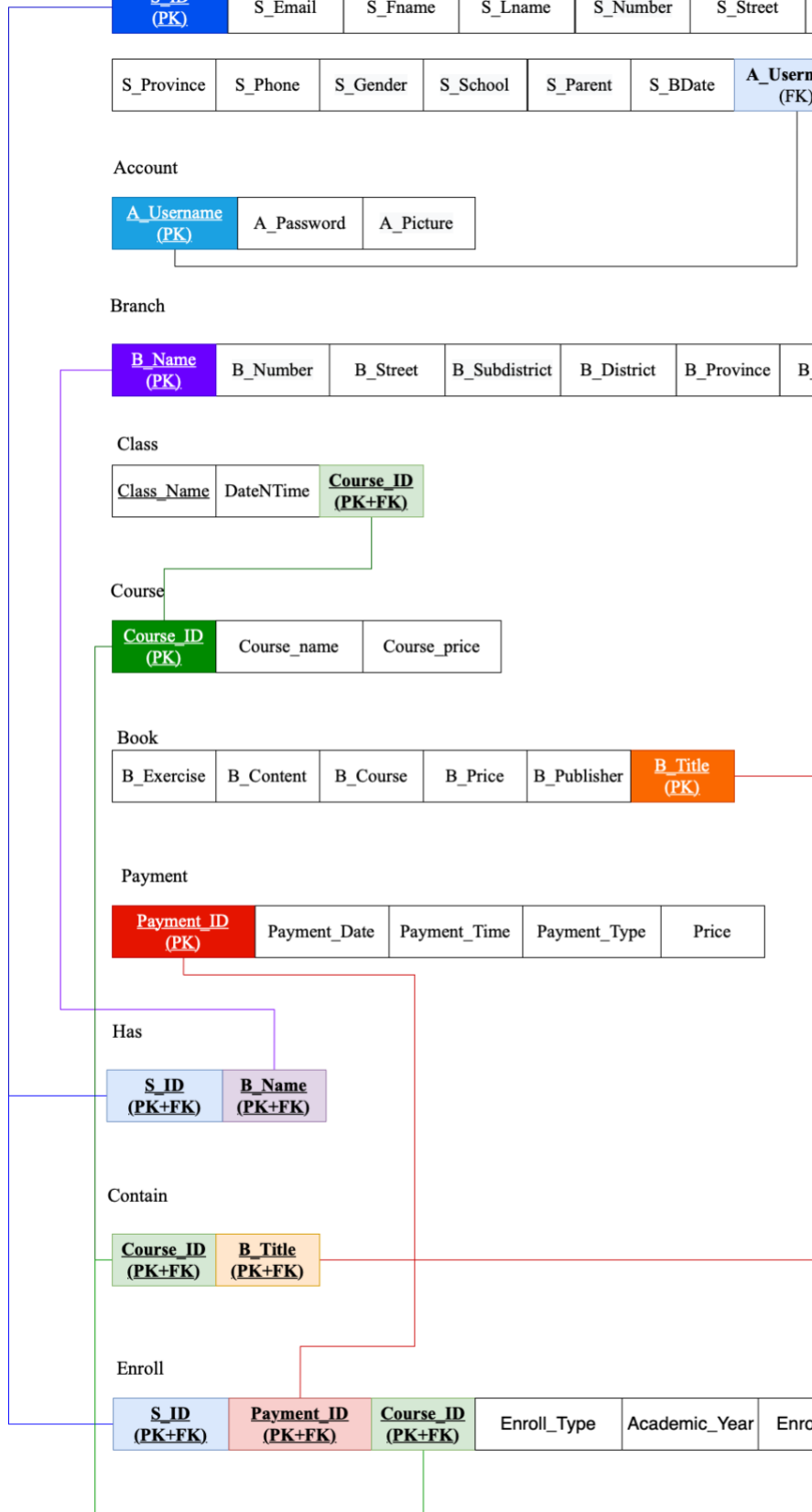
<b>S_ID (PK+FK)</b>	<b>B_Name (PK+FK)</b>
---------------------	-----------------------

# Contain

<b>Course_ID (PK+FK)</b>	<b>B_Title (PK+FK)</b>
--------------------------	------------------------

# Enroll

<b>S_ID (PK+FK)</b>	<b>Payment_ID (PK+FK)</b>	<b>Course_ID (PK+FK)</b>	Enroll_Type	Academic_Year	Enrolled_Date
---------------------	---------------------------	--------------------------	-------------	---------------	---------------



## Data Dictionary

Table Name	Attribute Name	Contents	Type	Format	Null able	Range	Key	FK Reference Table
Students	<u>SID</u>	Student Citizen's ID	CHAR (13)	xxxxxxxxx xxxx			PK	
	S_Email	Student's email	VARCHAR (30)	xxxxxx@x xxxx.com				
	S_Fname	Student's first name	VARCHAR (20)	XXXXXXXX x				
	S_Lname	Student's last name	VARCHAR (20)	XXXXXXXX x				
	S_Number	Student's number address	VARCHAR (20)	xx/xx				
	S_Street	Student's street address	VARCHAR (100)	Xxxxx	Y			
	S_Subdistrict	Student's subdistrict address	VARCHAR (100)	Xxxxx				
	S_District	Student's district address	VARCHAR (100)	Xxxxx				

	S_Province	Student's province address	VARCHAR (100)	XXXXXXXX				
	S_Phone	Student's phone number	CHAR (10)	XXXXXXXXX x				
	S_Gender	Student's gender	CHAR (1)	x		M or F		
	S_School	Student's school	VARCHAR (100)	XXXXXXXX				
	S_Parent	Student's parent phone number	CHAR (10)	XXXXXXXXX x				
	S_BD	Student's birth date	DATE	yyyy-mm-dd	Y			
	S_a_username	Student's account username	CHAR (7)	XXXXXXX	Y		FK	A_Username [Account]
Account	<u>A_Username</u>	Account's username	CHAR (7)	XXXXXXX			PK	
	A_Password	Account's password	CHAR (4)	XXXX				
	A_Picture	Account's picture	VARCHAR (20)	xx MB				


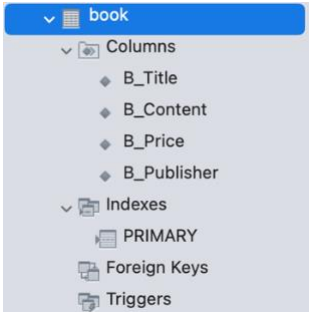
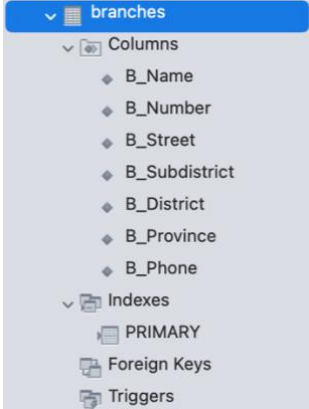
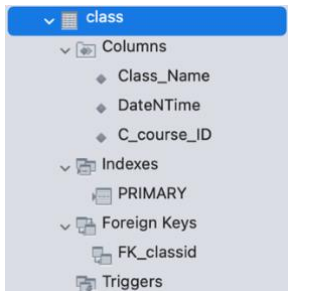
Branches	<b><u>B_Name</u></b>	Branch's name	VARCHAR (20)	xxxxxxxxx x			PK	
	B_Number	Branch's Number	VARCHAR (20)	Xx/xx				
	B_Street	Branch's Street	VARCHAR (50)	xxxxxxx	Y			
	B_Subdistrict	Branch's Subdistrict	VARCHAR (50)	Xxxxxxx				
	B_District	Branch's district	VARCHAR (50)	Xxxxxxx				
	B_Province	Branch's Province	VARCHAR (50)	Xxxxxxxx				
	B_Phone	Branch's phone	CHAR (9)	xxxxxxxxx				
Class	<b><u>Class_Name</u></b>	Classroom's name	VARCHAR (50)	RoomVx			PK	
	DateNTime	The date and time each room is taught.	VARCHAR (100)	Xxx 00:00-00:00 x.x.				
	<b><u>C_course_ID</u></b>	Course's ID	VARCHAR (20)	UCxx			PK+FK	Course_ID [Course]

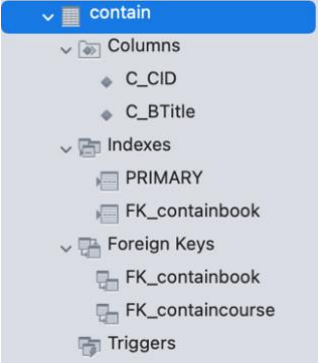
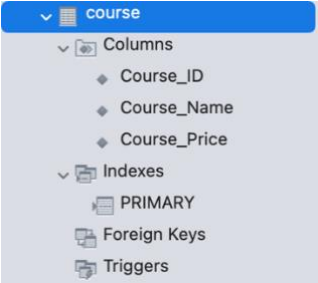

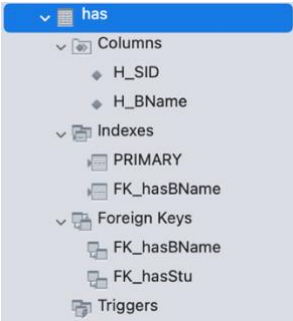
Course	<u>Course_ID</u>	Course's ID	VARCHAR (20)	UCxx			PK	
	Course_Name	Course's name	VARCHAR (200)	XXXXXXX				
	Course_Price	Course's price	INT	xxxxx				
Book	<u>B_Title</u>	Book's title	VARCHAR (100)	xxxxxxx			PK	
	B_Content	Book's content	VARCHAR (200)	xxxxxxxxx				
	B_Publisher	Book's publisher	VARCHAR (50)	Chem Ou				
	B_Price	Book's price	INT	Xxxxxx				
Payment	<u>Payment_ID</u>	Payment's ID	INT	XXXXXXX			PK	
	Payment_Date	Payment's date	DATE	yyyy-mm-dd				
	Payment_Time	Payment's time	TIME	00:00:00				
	Payment_Type	Type of payment	VARCHAR (50)	Xxxx				
	Price	Price paid in payment	INT	xxxx				

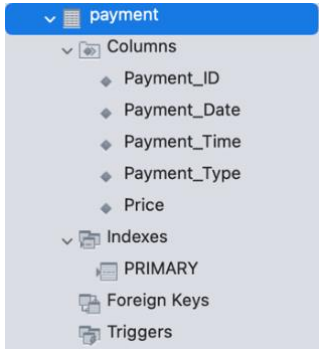
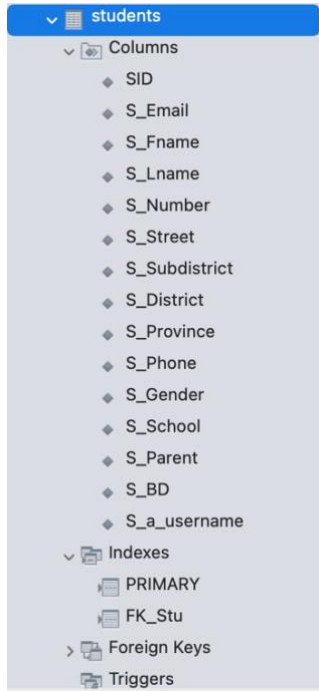
Has	<b><u>H_SID</u></b>	Student Citizen's ID	CHAR (13)	xxxxxxxxxx xxxx			PK+FK	SID [Student]
	<b><u>H_BName</u></b>	Branch's name	VARCH AR (50)	Xxxxxxx			PK+FK	B_Name [Branch]
Contain	<b><u>C_CourseID</u></b>	Course's ID	VARCH AR (50)	UCxx			PK+FK	Course_ID [Course]
	<b><u>C_BTitle</u></b>	Book's title	VARCH AR (50)	Xxxxxxx			PK+FK	B_Title [Book]
Enroll	<b><u>E_SID</u></b>	Student Citizen's ID	CHAR (13)	xxxxxxxxxx xxxx			PK+FK	SID [Student]
	<b><u>E_PayID</u></b>	Payment's ID	CHAR (9)	xxxxxxxxxx			PK+FK	Payment_ID [Payment]
	<b><u>E_CourseID</u></b>	Course's ID	VARCH AR (20)	UCxx			PK+FK	Course_ID [Course]
	Enroll_Type	Type of enrollment	VARCH AR (50)	Xxxx				
	Academic_year	The academic year in which the student is enrolled	CHAR (4)	xxxx				
	Enrolled_Date	Date and time of student enrollment	DATETIME	yyyy-mm-dd 00:00:00				



## Table description

Table	Description	Schema View	Information														
accounts	Account is kept data in terms of username, password and picture created by Student		<p><b>Table: accounts</b></p> <p><b>Columns:</b></p> <table><tr><td><u>A_Username</u></td><td>char(7) PK</td></tr><tr><td>A_Password</td><td>char(4)</td></tr><tr><td>A_Picture</td><td>varchar(20)</td></tr></table>	<u>A_Username</u>	char(7) PK	A_Password	char(4)	A_Picture	varchar(20)								
<u>A_Username</u>	char(7) PK																
A_Password	char(4)																
A_Picture	varchar(20)																
book	Book is kept data in terms of title, content, publisher, and price contained in Course		<p><b>Table: book</b></p> <p><b>Columns:</b></p> <table><tr><td><u>B_Title</u></td><td>varchar(100) PK</td></tr><tr><td>B_Content</td><td>varchar(200)</td></tr><tr><td>B_Price</td><td>int</td></tr><tr><td>B_Publisher</td><td>varchar(50)</td></tr></table>	<u>B_Title</u>	varchar(100) PK	B_Content	varchar(200)	B_Price	int	B_Publisher	varchar(50)						
<u>B_Title</u>	varchar(100) PK																
B_Content	varchar(200)																
B_Price	int																
B_Publisher	varchar(50)																
branches	Branches is kept data in terms of name, number, street, sub-district, district, province, and price, who's the student has study in which branch		<p><b>Table: branches</b></p> <p><b>Columns:</b></p> <table><tr><td><u>B_Name</u></td><td>varchar(20) PK</td></tr><tr><td>B_Number</td><td>varchar(20)</td></tr><tr><td>B_Street</td><td>varchar(50)</td></tr><tr><td>B_Subdistrict</td><td>varchar(50)</td></tr><tr><td>B_District</td><td>varchar(50)</td></tr><tr><td>B_Province</td><td>varchar(50)</td></tr><tr><td>B_Phone</td><td>char(9)</td></tr></table>	<u>B_Name</u>	varchar(20) PK	B_Number	varchar(20)	B_Street	varchar(50)	B_Subdistrict	varchar(50)	B_District	varchar(50)	B_Province	varchar(50)	B_Phone	char(9)
<u>B_Name</u>	varchar(20) PK																
B_Number	varchar(20)																
B_Street	varchar(50)																
B_Subdistrict	varchar(50)																
B_District	varchar(50)																
B_Province	varchar(50)																
B_Phone	char(9)																
class	Class is kept data in terms of name, and date-time contained in Course		<p><b>Table: class</b></p> <p><b>Columns:</b></p> <table><tr><td><u>Class_Name</u></td><td>varchar(50) PK</td></tr><tr><td>DateNTIME</td><td>varchar(100)</td></tr><tr><td><u>C_course_ID</u></td><td>varchar(20) PK</td></tr></table>	<u>Class_Name</u>	varchar(50) PK	DateNTIME	varchar(100)	<u>C_course_ID</u>	varchar(20) PK								
<u>Class_Name</u>	varchar(50) PK																
DateNTIME	varchar(100)																
<u>C_course_ID</u>	varchar(20) PK																

<b>contain</b>	Weak entity 'contain' is used to keep Book is contain by Course, which store course ID and book title		<b>Table: contain</b>  <b>Columns:</b> <u>C_CID</u> varchar(20) PK <u>C_BTitle</u> varchar(100) PK
<b>course</b>	Course is kept data in terms of course ID, name, price. And contained 'class' and 'book', which enroll by Student		<b>Table: course</b>  <b>Columns:</b> <u>Course_ID</u> varchar(20) PK Course_Name      varchar(200) Course_Price      int
<b>enroll</b>	Weak entity 'enroll' is used to keep 'course', 'payment' and 'student' that enroll which course, which stored academic year, and enrolled date		<b>Table: enroll</b>  <b>Columns:</b> <u>E_SID</u> char(13) PK <u>E_PayID</u> char(9) PK <u>E_CID</u> varchar(20) PK E_Type            varchar(50) E_AY                char(4) E_ED                datetime
<b>has</b>	Weak entity 'has' is used to keep the information in terms of student ID and branch name where student has study in which branch		<b>Table: has</b>  <b>Columns:</b> <u>H_SID</u> char(13) PK <u>H_BName</u> varchar(50) PK

<b>payment</b>	Payment is kept data in terms of payment ID, price, pay date, pay time, and payment type, which occur after enrolling by Student	 <p>The screenshot shows the 'payment' table structure in a database management tool. It lists columns: Payment_ID, Payment_Date, Payment_Time, Payment_Type, and Price. It also shows indexes, including a PRIMARY index on Payment_ID, and foreign keys and triggers.</p>	<b>Table: payment</b>  <b>Columns:</b> <table><tr><td><u>Payment_ID</u></td><td>char(9) PK</td></tr><tr><td>Payment_Date</td><td>date</td></tr><tr><td>Payment_Time</td><td>time</td></tr><tr><td>Payment_Type</td><td>varchar(20)</td></tr><tr><td>Price</td><td>int</td></tr></table>	<u>Payment_ID</u>	char(9) PK	Payment_Date	date	Payment_Time	time	Payment_Type	varchar(20)	Price	int																				
<u>Payment_ID</u>	char(9) PK																																
Payment_Date	date																																
Payment_Time	time																																
Payment_Type	varchar(20)																																
Price	int																																
<b>students</b>	Student is kept data in terms of address. And create 'account' and enroll 'course'	 <p>The screenshot shows the 'students' table structure in a database management tool. It lists columns: SID, S_Email, S_Fname, S_Lname, S_Number, S_Street, S_Subdistrict, S_District, S_Province, S_Phone, S_Gender, S_School, S_Parent, S_BD, and S_a_username. It also shows indexes, including a PRIMARY index on SID, and foreign keys and triggers.</p>	<b>Table: students</b>  <b>Columns:</b> <table><tr><td><u>SID</u></td><td>char(13) PK</td></tr><tr><td>S_Email</td><td>varchar(30)</td></tr><tr><td>S_Fname</td><td>varchar(20)</td></tr><tr><td>S_Lname</td><td>varchar(20)</td></tr><tr><td>S_Number</td><td>varchar(20)</td></tr><tr><td>S_Street</td><td>varchar(100)</td></tr><tr><td>S_Subdistrict</td><td>varchar(100)</td></tr><tr><td>S_District</td><td>varchar(100)</td></tr><tr><td>S_Province</td><td>varchar(100)</td></tr><tr><td>S_Phone</td><td>char(10)</td></tr><tr><td>S_Gender</td><td>char(1)</td></tr><tr><td>S_School</td><td>varchar(100)</td></tr><tr><td>S_Parent</td><td>char(10)</td></tr><tr><td>S_BD</td><td>date</td></tr><tr><td>S_a_username</td><td>char(7)</td></tr></table>	<u>SID</u>	char(13) PK	S_Email	varchar(30)	S_Fname	varchar(20)	S_Lname	varchar(20)	S_Number	varchar(20)	S_Street	varchar(100)	S_Subdistrict	varchar(100)	S_District	varchar(100)	S_Province	varchar(100)	S_Phone	char(10)	S_Gender	char(1)	S_School	varchar(100)	S_Parent	char(10)	S_BD	date	S_a_username	char(7)
<u>SID</u>	char(13) PK																																
S_Email	varchar(30)																																
S_Fname	varchar(20)																																
S_Lname	varchar(20)																																
S_Number	varchar(20)																																
S_Street	varchar(100)																																
S_Subdistrict	varchar(100)																																
S_District	varchar(100)																																
S_Province	varchar(100)																																
S_Phone	char(10)																																
S_Gender	char(1)																																
S_School	varchar(100)																																
S_Parent	char(10)																																
S_BD	date																																
S_a_username	char(7)																																

Basic part

1. List all the Payment detail.

**Expected 5 attributes, 11 rows as shown in the following output table.**

Statement:

```
SELECT Payment_ID, Payment_Date, Payment_TIME, Payment_Type, Price
FROM Payment
ORDER BY Payment_Date, Payment_TIME ASC;
```

Result:

Payment_ID	Payment_Date	Payment_TIME	Payment_Type	Price
367714631	2020-05-26	17:23:09	Mobile Banking	2200
771371440	2020-06-13	12:07:11	Krung Thai Bank	3800
209660058	2020-07-16	08:35:09	Counter Service	3800
773139342	2020-08-04	13:41:24	Krung Thai Bank	3400
777089264	2020-08-09	14:22:07	Krung Thai Bank	14000
363191350	2020-09-20	11:34:34	Mobile Banking	4500
362011933	2020-10-02	09:25:01	Mobile Banking	2000
202010309	2020-10-02	12:00:58	Counter Service	2800
770393711	2020-10-23	10:48:39	Krung Thai Bank	4800
209645320	2020-12-24	17:08:56	Counter Service	3000
362618177	2020-12-28	12:49:55	Mobile Banking	4800

2. Display List student that age less than 18 and parent phone number.

**Expected 2 attributes, 3 rows as shown in the following output table.**

Statement:

```
SELECT S_Phone AS "Student Phone", S_Parent AS "Parent Phone"
FROM students
WHERE YEAR(NOW())-YEAR(S_BD) < 18;
```

Result:

Student Phone	Parent Phone
0998761209	0889901116
0629259235	0629955859
0994742635	0616657859

3. List a branch that is located in the south part.

**Expected 3 attributes, 7 rows as shown in the following output table.**

Statement:

```
SELECT B_name AS "Branch's name in south", B_Province AS "Branch's province", B_District AS "Branch's district"
FROM branches
WHERE B_Phone LIKE '07%';
```

Result:

Branch's name in south	Branch's province	Branch's district
Chumphon	Chumphon	Mueang Chumphon
Hat Yai	Songkla	Hat Yai
Nakhon Si Thammarat	Nakhon Si Thammarat	Mueang Nakhon Si Thammarat
Phatthalung	Phatthalung	Mueang Phatthalung
Phuket	Phuket	Mueang Phuket
Surat Thani	Surat Thani	Mueang Surat Thani
Trang	Trang	Mueang Trang

4. List a student number that lives in the north province.

**Expected 3 attributes, 3 rows as shown in the following output table.**

Statement:

```
SELECT CONCAT(S_Fname, ' ', S_Lname) AS "Student name" , S_Phone, S_Province AS 'north province'
FROM students
WHERE S_Province = 'Chiang Mai' OR S_Province = 'Chiang Rai' OR S_Province = 'Lampang'
ORDER BY S_Province;
```

Result:

Student name	S_Phone	north province
Shaojun Wan	0629259235	Chiang Mai
Robert Hongthai	0831249990	Chiang Rai
Nippon Downey	0998761209	Lampang

5. Choose a junior's only course, and is a single course.

**Expected 3 attributes, 4 rows as shown in the following output table.**

Statement:

```
SELECT Course_ID, Course_Name, Course_Price
FROM course
WHERE Course_Name LIKE '%Junior%' AND Course_ID LIKE 'UC__';
```

Result:

Course_ID	Course_Name	Course_Price
UC01	Junior Chemistry	2800
UC02	Junior Chemistry for IJSO1	3000
UC03	Junior Chemistry for IJSO2	3400
UC04	Intensive exercise for Junior Chemistry	2200

6. Shows the type of payment and the maximum amount of each payment type.

**Expected 2 attributes, 3 rows as shown in the following output table.**

Statement:

```
SELECT Payment_Type AS "Type of payment", MAX(Price) AS "Max Price"
FROM payment
GROUP BY Payment_Type
ORDER BY MAX(Price) ASC;
```

Result:

Type of payment	Max Price
Counter Service	3800
Mobile Banking	4800
Krung Thai Bank	14000

7. List an enrolled date and student account username in 2022.

**Expected 2 attributes, 8 rows as shown in the following output table**

Statement:

```
SELECT DATE(E_ED) AS "Enrolled Date", COUNT(E_SID) AS "Student's enroll"
FROM enroll
WHERE YEAR(E_ED) = 2022
GROUP BY DATE(E_ED)
ORDER BY DATE(E_ED);
```

Result:

Enrolled Date	Student's enroll
2022-04-09	1
2022-04-20	1
2022-04-28	1
2022-05-02	1
2022-05-05	1
2022-05-09	1
2022-06-25	4
2022-10-02	1

8. List Course ID and book name belonging to Senior high school.

**Expected 2 attributes, 9 rows as shown in the following output table**

Statement:

```
SELECT C_CID AS "Course ID", C_BTitle AS "Book name"
FROM contain
WHERE C_CID LIKE 'UC1%'
ORDER BY C_BTitle DESC;
```

Result:

Course ID	Book name
UC14,UC15	Organic chemistry, polymer
UC10	Gas and gas properties
UC18,UC19	Exercise practice course
UC18,UC19	Entrance
UC14,UC15	Electrochemistry
UC11	Chemical reaction rate
UC12	Chemical equilibrium
UC17	Biomolecules
UC13	Acid-base I-II

9. List a branch phone that is located in the south.

**Expected 2 attributes, 7 rows as shown in the following output table**

Statement:

```
SELECT B_Province AS "Branch's province", B_Phone AS "Branch phone in south"
FROM branches
WHERE B_Phone LIKE '07%';
```

Result:

Branch's province	Branch phone in south
Chumphon	077510655
Songkla	074346165
Nakhon Si Thammarat	075431109
Phatthalung	074615571
Phuket	076234814
Surat Thani	077219255
Trang	075215359

10. List a branch name on the northeast.

**Expected 2 attributes, 9 rows as shown in the following output table**

Statement:

```
SELECT B_Phone AS "Branch phone in northeast", B_Province AS "Branch's province"
FROM branches
WHERE B_Phone LIKE '04%';
```

Result:

Branch phone in northeast	Branch's province
044621757	Buriram
043225752	Khonkaen
044268512	Nakhon Ratchasima
043516144	Roi Et
042732274	Sakon Nakhon
045262435	Ubon Ratchathani



11. List a total number of students who enroll in the 'UC14' course for all 3 academic years.

**Expected 2 attributes, 1 rows as shown in the following output table**

Statement:

```
SELECT E_CID AS "Course ID", COUNT(E_CID) AS 'Total number of Students'
FROM enroll
WHERE E_CID = 'UC14'
GROUP BY E_CID;
```

Result:

Course ID	Total number of Students
UC14	4

12. List a total number of students who enroll 'Live' type for the academic year 2021.

**Expected 2 attributes, 1 rows as shown in the following output table**

Statement:

```
SELECT E_Type, COUNT(E_Type) AS 'Total number of Students in 2021'
FROM enroll
WHERE E_Type = 'Live' AND E_AY = 2021
GROUP BY E_Type;
```

Result:

E_Type	Total number of Students in 2021
Live	8

13. List a course whose price is less than or equal to 2000.

**Expected 2 attributes, 8 rows as shown in the following output table**

Statement:

```
SELECT Course_Name AS 'Course Name', Course_Price AS 'Course Price'
FROM course
WHERE Course_Price <= 2000
ORDER BY Course_Price ASC;
```

Result:

Course Name	Course Price
Chemistry Grade 13, Volume 5 - Biomolecules	1000
Chemistry Grade 12, Volume 1 Periodic Table	1400
Chemistry Grade 12, Volume 3 - Gases and Gas ...	1400
Chemistry Grade 12, Volume 3 - Chemical Reacti...	1400
Chemistry Grade 12, Volume 3 - Chemical Equilib...	1400
Fundamentals of Lectures, Volume 1 (Grade 12)	1500
Fundamentals of Computing, Volume 2 (Grade 12)	1600
Chemistry Grade 13, Volume 4 - Electrochemistry	2000

14. List all students (first name and last name) who have the letter "N" in his/her first name and last name.

**Expected 2 attributes, 11 rows as shown in the following output table**

Statement:

```
FROM students
WHERE S_Fname LIKE "%N%" OR S_Lname LIKE "%N%"
ORDER BY S_Fname ASC;
```

Result:

Name	Surname
Dong	Eng
Duangxuanli	Tong
Hao	Wong
James	Mekmanee
Kongphop	Panyanirun
Linfeng	Zhun
Nippon	Downey
Ponnapassorn	Iamborisut
Robert	Hongthai
Shaojun	Wan

15. List the top 5 oldest students (first name and last name) with their gender, and their age in 2018.

**Expected 3 attributes, 5 rows as shown in the following output table**

```
SELECT CONCAT(S_Fname, ' ', S_Lname) AS 'Full Name', S_Gender AS 'Gender', (2020 - YEAR(S_BD)) AS 'Age'
FROM students
ORDER BY age DESC
LIMIT 5;
```

Result:

Full Name	Gender	Age
Robert Hongthai	M	19
Jirapat Deesara	M	19
Thadeeya Duangkaew	F	19
Kongphop Panyanirun	M	18
James Mekmanee	M	18

16. List all class time which you study in room 'V1'.

**Expected 3 attributes, 3 rows as shown in the following output table**

Statement:

```
SELECT Class_Name AS 'Class name', DateTime AS 'Date and Time', C_course_ID AS 'Course ID'
FROM class
WHERE Class_Name LIKE "%V1%"
ORDER BY C_course_ID;
```

Result:

Class name	Date and Time	Course ID
RoomV1	Sun 09:00 - 12:00	UC01
RoomV1	Sat 13:30 - 16:00	UC08,UC09,UC10
RoomV1	Thu 08:00 - 13:00	UC19

17. List all the book details.

**Expected 4 attributes, 21 rows as shown in the following output table**

Statement:

```
SELECT B_Title AS 'Book title', B_Content AS 'Content', B_Price AS 'Price', B_Publisher AS 'Publisher'  
FROM book  
ORDER BY B_Title ASC;
```

Result:

Book title	Content	Price	Publisher
Acid-base I-II	Acid-base	2600	Chem Ou
Atomic structure, safety and skills in chemistry ...	Basic Chemistry, Atomic Structure	2600	Chem Ou
Biomolecules	Biomolecules	1000	Chem Ou
Chemical bonding	Chemical bonding	2200	Chem Ou
Chemical equilibrium	Chemical equilibrium	1400	Chem Ou
Chemical reaction rate	Chemical reaction rate	1400	Chem Ou
Chemistry exercises for Junior high school	Focus on doing problems and analyzing for entr...	2200	Chem Ou
Electrochemistry	Electrochemistry	2000	Chem Ou
Entrance	Summary of advanced content, connecting the ...	14000	Chem Ou
Exercise practice course	Summarize important specific content and focus...	3800	Chem Ou
Fundamentals of Computing, Volume 2 (Grade 12)	Pre-prep the basics of junior high school chemis...	1600	Chem Ou
Fundamentals of Lectures, Volume 1 (Grade 12)	Pre-prep the basics of junior high school chemis...	1500	Chem Ou
Gas and gas properties	Gas and gas properties	1400	Chem Ou
IJSO - 1	Basic chemistry for Junior high school volume 1 ...	3000	Chem Ou
IJSO - 2	Basic chemistry for Junior high school volume 3 ...	3400	Chem Ou
Organic chemistry, polymer	Organic chemistry, polymer	4500	Chem Ou
Periodic table	periodic table	1400	Chem Ou
Pre-Entrance	Improve all basics in high school before going to...	3800	Chem Ou
Science (chemistry) - Junior high school	Chemistry for Grade 7 - Grade 9	2800	Chem Ou
Stoichiometry I	Moles and chemical formulas	2600	Chem Ou
Stoichiometry II	Solution, stoichiometry	2800	Chem Ou

18. List all branches where located in Bangkok.

**Expected 3 attributes, 5 rows as shown in the following output table**

Statement:

```
SELECT B_Name AS 'Branch name', B_Province AS 'Province', B_Phone AS 'Contact'
FROM branches
WHERE B_Province = 'bangkok'
ORDER BY B_Province ASC;
```

Result:

Branch name	Province	Contact
Bankapi	Bangkok	023701353
Phayathai	Bangkok	023060850
Pinklao	Bangkok	024095894
Srinakarin	Bangkok	020220035
Wongwian Yai	Bangkok	024378518

19. List a total number of students who enroll 'DVD' type for the academic year 2022.

**Expected 2 attributes, 1 row as shown in the following output table**

Statement:

```
SELECT E_Type, COUNT(E_Type) AS 'Total number of Students in 2022'
FROM enroll
WHERE E_Type = 'DVD' AND E_AY = 2021
GROUP BY E_Type;
```

Result:

E_Type	Total number of Students in 2022
DVD	3

20. List a total number of students who enroll each year (e.g.,2020, 2021, and 2022).

**Expected 2 attributes, 3 rows as shown in the following output table**

Statement:

```
SELECT E_AY AS 'Academic year', COUNT(E_AY) AS 'Total number of students'  
FROM enroll  
GROUP BY E_AY  
ORDER BY E_AY ASC;
```

Result:

Academic year	Total number of students
2020	11
2021	11
2022	11

## Advanced part

1. List all students who have enroll course 'UC18' and 'UC19'.

The result must contain four columns: "Student username", "Student name" (student first name and last name), Course ID, and "Course name", sorted by student's ID

**Expected 3 attributes, 9 rows as shown in the following output table**

Statement:

```
SELECT a.A_Username AS 'Student username', CONCAT(S_Fname, ' ', S_Lname) AS 'Student name', Course_ID AS 'Course ID', Course_Name AS 'Course name'
FROM accounts a
INNER JOIN students s ON s.S_a_username = a.A_Username
INNER JOIN enroll e ON e.E_SID = s.SID
INNER JOIN course crs ON crs.Course_ID = e.E_CID
WHERE crs.Course_ID LIKE 'UC18' OR crs.Course_ID LIKE 'UC19';
```

Result:

	Student userna...	Student name	Course ID	Course name
▶	2116548	Duangxuanli Tong	UC18	Chemistry Grade 14, Course Entrance
	2216984	Jirapat Deesara	UC18	Chemistry Grade 14, Course Entrance
	2903126	Kongphop Panyanirun	UC18	Chemistry Grade 14, Course Entrance
	2204102	Thadeeya Duangkaew	UC18	Chemistry Grade 14, Course Entrance
	2715649	Shaojun Wan	UC18	Chemistry Grade 14, Course Entrance
	2106473	Ponnapassorn Iamborisut	UC19	Intensive exercise for High Chemistry
	2204102	Thadeeya Duangkaew	UC19	Intensive exercise for High Chemistry
	2715649	Shaojun Wan	UC19	Intensive exercise for High Chemistry
	2057436	Hao Wong	UC19	Intensive exercise for High Chemistry

2. List all student which enroll in DVD Type on academic 2021 year.

The result must contain four columns: "Student username", "Name", "Course ID", and "Type", sorted by username's names alphabetically

**Expected 4 attributes, 13 rows as shown in the following output table**

Statement:

```
SELECT A_Username AS "Student's username", CONCAT(S_Fname, ' ', S_Lname) AS "Student's Name", E_CID AS "Course ID", E_Type AS "Type"
FROM accounts a
INNER JOIN students s ON s.S_a_username = a.A_Username
INNER JOIN enroll e ON e.E_SID = s.SID
INNER JOIN course crs ON crs.Course_ID = e.E_CID
WHERE E_Type = 'DVD';
```

Result:

	Student's usena...	Student's Name	Course ID	Type
▶	2116548	Duangxuanli Tong	UC23	DVD
	2116548	Duangxuanli Tong	UC18	DVD
	2216984	Jirapat Deesara	UC23	DVD
	2216984	Jirapat Deesara	UC18	DVD
	2115693	Jidapa Moolkeaw	UC04	DVD
	2903126	Kongphop Panyanirun	UC23	DVD
	2903126	Kongphop Panyanirun	UC18	DVD
	2901290	James Mekmanee	UC23	DVD
	2204102	Thadeeya Duangkaew	UC23	DVD
	2204102	Thadeeya Duangkaew	UC18	DVD
	2715649	Shaojun Wan	UC18	DVD
	2732481	Linfeng Zhun	UC01,UC04	DVD
	2057436	Hao Wong	UC23	DVD

3. List a total number of students who enroll 'Entrance' course for all 3 academic years.

The result must contain four columns: "Student ID", "Name", "Course ID" and "Course name"

**Expected 4 attributes, 6 rows as shown in the following output table**

Statement:

```
SELECT SID, CONCAT(S_Fname, ' ', S_Lname) AS "Student's name", course_ID, course_Name
FROM students s
INNER JOIN enroll e ON e.E_SID = s.SID
INNER JOIN course crs ON crs.Course_ID = e.E_CID
WHERE E_AY = 2022 AND course_Name LIKE "%Entrance%";
```

Result:

	SID	Student's name	course_ID	course_Name
▶	1102312034602	Duangxuanli Tong	UC23	Pre-Entrance
	1102735543510	Jirapat Deesara	UC23	Pre-Entrance
	1106652241503	Kongphop Panyanirun	UC23	Pre-Entrance
	1120012329012	James Mekmanee	UC23	Pre-Entrance
	1139214271920	Thadeeya Duangkaew	UC23	Pre-Entrance
	2600289182390	Hao Wong	UC23	Pre-Entrance

4. List of students enrolled course in each branch

The result must contain four columns: "Branch Name" and "freq of student "

**Expected 2 attributes, 31 rows as shown in the following output table**

Statement:

```
SELECT B_Name AS "Branch Name", COUNT(SID) AS 'freq of student'
FROM branches b
LEFT OUTER JOIN has h ON b.B_Name = h.h_BName
LEFT OUTER JOIN students s ON h.h_SID = s.SID
GROUP BY B_Name;
```

Result:

Branch Name	freq of student
▶ Bankapi	2
Buriram	0
Chachoengsao	1
Chiang Mai	1
Chiang Rai	2
Chonburi	1
Chumphon	0
Hat Yai	1
Khonkaen	1
Lampang	1
Nakhon Ratc...	1
Nakhon Sawan	2
Nakhon Si Th...	1
Nonthaburi	1
Phatthalung	0
Phayathai	7

Phitsanulok	0
Phuket	1
Pinklao	2
Rangsit	3
Ratchaburi	1
Rayong	1
Roi Et	0
Sakon Nakhon	0
Saraburi	0
Srinakarin	1
Surat Thani	0
Trang	1
Ubon Ratchat...	0
Udon Thani	0
Wongwian Yai	1



5. Display course ID, course name, that no student enrolled

**Expected 2 attributes, 19 rows as shown in the following output table**

Statement:

```
SELECT Course_ID, Course_Name
FROM course c
LEFT OUTER JOIN enroll e ON e.E_CID = c.Course_ID
LEFT OUTER JOIN students s ON s.SID = e.E_SID
GROUP BY c.Course_ID
HAVING COUNT(s.SID) = 0
ORDER BY c.Course_ID ASC;
```

Result

Course_ID	Course_Name
UC01	Junior Chemistry
UC01,UC02,UC03,UC04	Junior chemistry , IJSO 1 , IJSO 2 , Intensive exercise for Junior high school
UC05	Chemistry Grade 12, Volume 1 - Fundamentals of Chemistry + Atomic Structure
UC05,UC06,UC07	Chemistry Grade 12, Volume 1 - Atomic Structure, Periodic Table, Chemical Bonding
UC06	Chemistry Grade 12, Volume 1 Periodic Table
UC07	Chemistry Grade 12, Volume 1 Chemical Bonding
UC08	Chemistry Grade 12, Volume 2 - Moles and Chemistry Formulas stoichiometry I
UC09	Chemistry Grade 12, Volume 2 - Solution, Stoichiometry II
UC10	Chemistry Grade 12, Volume 3 - Gases and Gas Properties
UC11	Chemistry Grade 12, Volume 3 - Chemical Reaction Rate
UC11,UC12,UC13	Chemistry Grade 12, Volume 3 - ratio , equilibrium , acid base
UC12	Chemistry Grade 12, Volume 3 - Chemical Equilibrium
UC13	Chemistry Grade 13, Volume 4 - Acid-Bases I-II
UC14,UC15	Chemistry Grade 13, Volume 4, Electrochemistry, Organic Chemistry, Polymers
UC17	Chemistry Grade 13, Volume 5 - Biomolecules
UC18,UC19	Entrance, Intensive exercise for High Chemistry
UC21	Fundamentals of Lectures, Volume 1 (Grade 12)
UC21,UC22	Fundamentals of Lectures and Calculations, Volumes 1-2 (Grade 12)
UC22	Fundamentals of Computing, Volume 2 (Grade 12)

6. Displays book titles that are the same as course titles. Sorted by book titles alphabetically

**Expected 1 attributes, 21 rows as shown in the following output table**

Statement:

Result:

```
SELECT C_BTitle AS 'Book name'
FROM contain
WHERE C_CID IN (SELECT Course_ID FROM course)
GROUP BY C_BTitle
ORDER BY C_BTitle;
```

Book name
► Acid-base I-II
Atomic structure, safety and skills in chemistry operations
Biomolecules
Chemical bonding
Chemical equilibrium
Chemical reaction rate
Chemistry exercises for junior high school
Electrochemistry
Entrance
Exercise practice course
Fundamentals of Computing, Volume 2 (Grade 12)
Fundamentals of Lectures, Volume 1 (Grade 12)
Gas and gas properties
IJSO - 1
IJSO - 2
Organic chemistry, polymer
Periodic table
Pre-Entrance
Science (chemistry) - Junior high school
Stoichiometry I
Stoichiometry II

7. Displays the date and time of the classroom where each course is taught that the student has enrolled in. Sorted by Class Name alphabetically

**Expected 4 attributes, 13 rows as shown in the following output table**

Statement:

```
SELECT DISTINCT A.Username AS "Student username", Class_Name, DateTime AS "Date and Time", C_course_ID AS "Course ID", Course_Name AS "Course Name"
FROM class cl
INNER JOIN course crs ON crs.Course_ID = cl.C_course_ID
INNER JOIN enroll e ON e.E_CID = crs.Course_ID
INNER JOIN payment p ON p.Payment_ID = e.E_PayID
INNER JOIN students s ON e.E_SID = s.SID
INNER JOIN accounts a ON a.A_Username = s.S_a_username
WHERE TIME(E_ED) BETWEEN '09:00:00' AND '10:00:00'
ORDER BY Class_Name ASC;
```

Result:

Student userna...	Class_Name	Date and Time	Course ID	Course Name
▶ 2116548	RoomV1	Sat 13:30 - 16:00	UC08,UC09,UC10	Chemistry Grade 12 Volume 2, - Moles and Chemistry Formulas stoichiometry I,...
2901290	RoomV1	Sat 13:30 - 16:00	UC08,UC09,UC10	Chemistry Grade 12 Volume 2, - Moles and Chemistry Formulas stoichiometry I,...
2001892	RoomV1	Sat 13:30 - 16:00	UC08,UC09,UC10	Chemistry Grade 12 Volume 2, - Moles and Chemistry Formulas stoichiometry I,...
2057436	RoomV1	Sat 13:30 - 16:00	UC08,UC09,UC10	Chemistry Grade 12 Volume 2, - Moles and Chemistry Formulas stoichiometry I,...
2715649	RoomV1	Thu 08:00 - 13:00	UC19	Intensive exercise for High Chemistry
2115693	RoomV2	Wed 13:30 - 16:00	UC02	Junior Chemistry for IJSO1
2116548	RoomV2	Sun 13:30 - 16:00	UC08,UC09,UC10	Chemistry Grade 12 Volume 2, - Moles and Chemistry Formulas stoichiometry I,...
2901290	RoomV2	Sun 13:30 - 16:00	UC08,UC09,UC10	Chemistry Grade 12 Volume 2, - Moles and Chemistry Formulas stoichiometry I,...
2001892	RoomV2	Sun 13:30 - 16:00	UC08,UC09,UC10	Chemistry Grade 12 Volume 2, - Moles and Chemistry Formulas stoichiometry I,...
2057436	RoomV2	Sun 13:30 - 16:00	UC08,UC09,UC10	Chemistry Grade 12 Volume 2, - Moles and Chemistry Formulas stoichiometry I,...
2115693	RoomV3	Mon 13:00 - 15:30	UC03	Junior Chemistry for IJSO2
2715649	RoomV5	Mon 08:00 - 13:00	UC19	Intensive exercise for High Chemistry
2715649	RoomV6	Wed 08:00 - 13:00	UC19	Intensive exercise for High Chemistry

8. List of students living in Bangkok and enroll at Phayathai branch

**Expected 3 attributes, 4 rows as shown in the following output table**

Statement:

```
SELECT CONCAT(S_Fname, ' ', S_Lname) AS 'student name', S_Province AS 'student province', h_BName AS 'Branch enroll'
FROM students s
INNER JOIN has h ON h.h_SID = s.SID
INNER JOIN branches b ON b.B_Name = h.h_BName
WHERE S_Province = 'Bangkok' AND h_BName = 'Phayathai';
```

Result:

	student name	student province	Branch enroll
▶	Jirapat Deesara	Bangkok	Phayathai
	Jidapa Moolkeaw	Bangkok	Phayathai
	Kongphop Panyanirun	Bangkok	Phayathai
	Ponnapassorn Iamborisut	Bangkok	Phayathai

9. List a student ID and student email that has enroll in South branch.

The result must contain three columns: "S\_ID", "Student Email", "South's Branch"

**Expected 3 attributes, 4 rows as shown in the following output table**

Statement:

```
SELECT A_Username AS "Student username" , S_Email AS "Student Email", B_Name AS "South's branch"
FROM branches b
INNER JOIN has h ON b.B_Name = h.h_BName
INNER JOIN students s ON h.h_SID = s.SID
INNER JOIN accounts a ON a.A_Username = s.S_a_username
WHERE B_Phone LIKE '07%';
```

Result:

	Student userna...	Student Email	South's branch
►	2216984	Jirapatpui@gmail.com	Hat Yai
	2732481	Zhangfeng@hotmail.com	Nakhon Si Thammarat
	2732481	Zhangfeng@hotmail.com	Phuket
	2732481	Zhangfeng@hotmail.com	Trang

10. Displays courses ID, Course Name, and course prices that are less than the total course price of grade 12.

**Expected 3 attributes, 13 rows as shown in the following output table**

Statement:

```
SELECT Course_ID AS "Course ID" , Course_Name AS "Course Name", Course_Price AS "Course Price"
FROM course
WHERE Course_Price < (SELECT AVG(Course_Price)
FROM course
WHERE Course_Name LIKE '%Grade 12%')
ORDER BY Course_ID ASC;
```

Result:

Course ID	Course Name	Course Price
► UC04	Intensive exercise for Junior Chemistry	2200
UC05	Chemistry Grade 12, Volume 1 - Fundamentals of Chemistry + Atomic Structure	2600
UC06	Chemistry Grade 12, Volume 1 Periodic Table	1400
UC07	Chemistry Grade 12, Volume 1 Chemical Bonding	2200
UC08	Chemistry Grade 12, Volume 2 - Moles and Chemistry Formulas stoichiometry I	2600
UC10	Chemistry Grade 12, Volume 3 - Gases and Gas Properties	1400
UC11	Chemistry Grade 12, Volume 3 - Chemical Reaction Rate	1400
UC12	Chemistry Grade 12, Volume 3 - Chemical Equilibrium	1400
UC13	Chemistry Grade 13, Volume 4 - Acid-Bases I-II	2600
UC14	Chemistry Grade 13, Volume 4 - Electrochemistry	2000
UC17	Chemistry Grade 13, Volume 5 - Biomolecules	1000
UC21	Fundamentals of Lectures, Volume 1 (Grade 12)	1500
UC22	Fundamentals of Computing, Volume 2 (Grade 12)	1600

11. Displays the student's name, course code and course name that the student enrolled Course UC18 in 2021 but did not enroll Course UC19 in 2022.

**Expected 3 attributes, 2 rows as shown in the following output table**

Statement:

```
SELECT CONCAT(S_Fname, ' ', S_Lname) AS "Student Name", E_CID AS "Course's enroll", Course_name AS "Course's name"
FROM students s
LEFT OUTER JOIN accounts a ON a.A_Username = s.S_a_username
LEFT OUTER JOIN enroll e ON e.E_SID = s.SID
LEFT OUTER JOIN course crs ON crs.Course_ID = e.E_CID
WHERE E_CID = "UC18" AND E_AY = '2020' AND E_CID NOT IN (SELECT E_CID FROM enroll
WHERE E_CID = "UC19" AND E_AY = '2021');
```

Result:

	Student Name	Course's enroll	Course's name
►	Kongphop Panyanirun	UC18	Chemistry Grade 14, Course Entrance
	Thadeeya Duangkaew	UC18	Chemistry Grade 14, Course Entrance

12. Displays the books title and course name that contain the same word 'Entrance'.

**Expected 1 attributes, 4 rows as shown in the following output table**

Statement:

```
SELECT Course_Name AS "Name" FROM course WHERE Course_Name LIKE "%Entrance%"
UNION
SELECT B_Title AS "Name" FROM book WHERE B_Title LIKE "%Entrance%";
```

Result:

	Name
►	Chemistry Grade 14, Course Entrance
	Entrance, Intensive exercise for High Chemistry
	Pre-Entrance
	Entrance

13. display the total price of the junior course whose course price is greater than the minimum course price of the entrance course

**Expected 1 attributes, 1 rows as shown in the following output table**

Statement:

```
SELECT SUM(Course_Price) AS "Course Price that "  
FROM course  
WHERE Course_Name LIKE "%Junior%" AND Course_Price > (SELECT MIN(Course_Price)  
FROM course  
WHERE Course_Name LIKE "%Entrance%");
```

Result:

	Sum of course price in Junior
▶	15200

14. Display the province of the branch that matches the province of the student that starts with the letter 'c'.

**Expected 1 attributes, 5 rows as shown in the following output table**

Statement:

```
SELECT B_Province AS "Province"  
FROM branches  
WHERE B_Province LIKE "C%"  
UNION  
SELECT S_Province AS "Province"  
FROM students  
WHERE S_Province LIKE "C%";
```

Result:

	Province
▶	Chachoengsao
	Chiang Mai
	Chiang Rai
	Chonburi
	Chumphon

15. Displays the lowest and highest prices for each payment type. Sort by lowest price

**Expected 2 attributes, 3 rows as shown in the following output table**

Statement:

```
SELECT Payment_Type, CONCAT('[',MIN(Price), ', ', MAX(Price),']') AS "Price range"
FROM payment p
INNER JOIN enroll e ON e.E_PayID = p.Payment_ID
INNER JOIN course crs ON crs.Course_ID = e.E_CID
INNER JOIN students s ON s.SID = e.E_SID
GROUP BY Payment_Type
ORDER BY MIN(Price) ASC;
```

Result:

	Payment_Type	Price range
►	Mobile Banking	[2000, 4800]
	Counter Service	[3000, 3800]
	Krung Thai Bank	[3400, 14000]

16. List a student ID and student email that has enroll in North branch.

The result must contain three columns: "S\_ID", "Student Email", "North's Branch"

**Expected 3 attributes, 6 rows as shown in the following output table**

Statement:

```
SELECT A.Username AS "Student username" , S_Email AS "Student Email", B_Name AS "North's branch"
FROM branches b
INNER JOIN has h ON b.B_Name = h.h_BName
INNER JOIN students s ON h.h_SID = s.SID
INNER JOIN accounts a ON a.A_Username = s.S_a_username
WHERE B_Phone LIKE '05%';
```

Result:

	Student userna...	Student Email	North's branch
►	2715649	ShaojunW@gmail.com	Chiang Mai
	2106473	Paulaponn@gmail.com	Chiang Rai
	2715649	ShaojunW@gmail.com	Chiang Rai
	2715649	ShaojunW@gmail.com	Lampang
	2001892	Wangwang@gmail.com	Nakhon Sawan
	2057436	Henhao@hotmail.com	Nakhon Sawan

17. List a student ID and student email that has enroll in Northeast branch.

The result must contain three columns: "S\_ID", "Student Email", "Northeast's Branch"

**Expected 3 attributes, 2 rows as shown in the following output table**

Statement:

```
SELECT A.Username AS "Student username" , S.Email AS "Student Email", B.Name AS "Northeast's branch"
FROM branches b
INNER JOIN has h ON b.B_Name = h.h_BName
INNER JOIN students s ON h.h_SID = s.SID
INNER JOIN accounts a ON a.A_Username = s.S_a_username
WHERE B.Phone LIKE '04%';
```

Result:

	Student username	Student Email	Northeast's branch
►	2116548	TangD@gmail.com	Khonkaen
	2116548	TangD@gmail.com	Nakhon Ratchasima

18. List a student ID and student email that has enroll in central branch.

-- The result must contain three columns: "S\_ID", "Student Email", "Central's Branch"

**Expected 3 attributes, 27 rows as shown in the following output table**

Statement:

```
SELECT A.Username AS "Student username" , S.Email AS "Student Email", B.Name AS "Central's branch"
FROM branches b
INNER JOIN has h ON b.B_Name = h.h_BName
INNER JOIN students s ON h.h_SID = s.SID
INNER JOIN accounts a ON a.A_Username = s.S_a_username
WHERE B.Phone LIKE '02%' OR B.Phone LIKE '03%' OR B.Phone LIKE '05%';
```

Result:

	Student username	Student Email	Central's branch
►	2901290	Jamespotter@gmail.com	Bankapi
	2204102	Thadeeya00@hotmail.com	Bankapi
	2116548	TangD@gmail.com	Chachoengsao
	2715649	ShaojunW@gmail.com	Chiang Mai
	2106473	Paulaponn@gmail.com	Chiang Rai
	2715649	ShaojunW@gmail.com	Chiang Rai
	2001892	Wangwang@gmail.com	Chonburi
	2715649	ShaojunW@gmail.com	Lampang
	2001892	Wangwang@gmail.com	Nakhon Sawan
	2057436	Enhao@hotmail.com	Nakhon Sawan
	2901290	Jamespotter@gmail.com	Nonthaburi
	2216984	Jirapatpui@gmail.com	Phayathai
	2115693	JidapaM@gmail.com	Phayathai
	2903126	Kongphopppp@hotmail.c...	Phayathai

2106473	Paulaponn@gmail.com	Phayathai
2901290	Jamespotter@gmail.com	Phayathai
2204102	Thadeeya00@hotmail.com	Phayathai
2057436	Enhao@hotmail.com	Phayathai
2216984	Jirapatpui@gmail.com	Pinklao
2115693	JidapaM@gmail.com	Pinklao
2903126	Kongphopppp@hotmail.c...	Rangsit
2106473	Paulaponn@gmail.com	Rangsit
2204102	Thadeeya00@hotmail.com	Rangsit
2057436	Enhao@hotmail.com	Ratchaburi
2001892	Wangwang@gmail.com	Rayong
2903126	Kongphopppp@hotmail.c...	Srinakarin
2115693	JidapaM@gmail.com	Wongwian Yai



19. Display student emails with payment after noon (12:00 noon) and enrollment dates of students enrolled in each course and branch. Sort by time of payment

**Expected 5 attributes, 21 rows as shown in the following output table**

Statement:

```
SELECT S_Email AS "Student email", Course_ID AS "Course ID", Payment_Time AS "Payment time", DATE(E_ED) AS "Enroll date", B_Name AS "Branch's name"
FROM class cl
INNER JOIN course crs ON crs.Course_ID = cl.C_course_ID
INNER JOIN enroll e ON e.E_CID = crs.Course_ID
INNER JOIN payment p ON p.Payment_ID = e.E_PayID
INNER JOIN students s ON e.E_SID = s.SID
INNER JOIN has h ON s.SID = h.H_SID
INNER JOIN branches b ON b.B_Name = h.h_BName
WHERE TIME(Payment_Time) > ("12:00:00")
GROUP BY B_Name
ORDER BY Payment_Time;
```

Result:

Student email	Course ID	Payment time	Enroll date	Branch's name
TangD@gmail.com	UC08,UC09,UC10	12:49:55	2020-10-01	Chachoengsao
TangD@gmail.com	UC08,UC09,UC10	12:49:55	2020-10-01	Khonkaen
TangD@gmail.com	UC08,UC09,UC10	12:49:55	2020-10-01	Nakhon Ratchasima
Jirapatpui@gmail.com	UC08,UC09,UC10	12:49:55	2020-10-01	Hat Yai
Jamespotter@gmail.com	UC08,UC09,UC10	12:49:55	2020-10-01	Bankapi
Jamespotter@gmail.com	UC08,UC09,UC10	12:49:55	2020-10-01	Nonthaburi
Wangwang@gmail.com	UC08,UC09,UC10	12:49:55	2020-10-01	Chonburi
Wangwang@gmail.com	UC08,UC09,UC10	12:49:55	2020-10-01	Nakhon Sawan
Wangwang@gmail.com	UC08,UC09,UC10	12:49:55	2020-10-01	Rayong
ShaojunW@gmail.com	UC08,UC09,UC10	12:49:55	2020-10-01	Chiang Mai
ShaojunW@gmail.com	UC08,UC09,UC10	12:49:55	2020-10-01	Chiang Rai
ShaojunW@gmail.com	UC08,UC09,UC10	12:49:55	2020-10-01	Lampang
Henhao@hotmail.com	UC08,UC09,UC10	12:49:55	2020-10-01	Ratchaburi
JidapaM@gmail.com	UC03	13:41:24	2022-10-02	Phayathai
JidapaM@gmail.com	UC03	13:41:24	2022-10-02	Pinklao
JidapaM@gmail.com	UC03	13:41:24	2022-10-02	Wongwian Yai
Zhangfeng@hotmail.com	UC03	13:41:24	2022-04-09	Nakhon Si Thammarat
Zhangfeng@hotmail.com	UC03	13:41:24	2022-04-09	Phuket
Zhangfeng@hotmail.com	UC03	13:41:24	2022-04-09	Trang
Kongphopppp@hotmail...	UC14	17:08:56	2021-01-16	Rangsit
Kongphopppp@hotmail...	UC14	17:08:56	2021-01-16	Srinakarin

20. Displays the total number of students who paid through each payment.

**Expected 2 attributes, 3 rows as shown in the following output table**

Statement:

```
SELECT Payment_Type, COUNT(E_PayID) AS "Total student in each payment"
FROM payment p
LEFT OUTER JOIN enroll e ON e.E_PayID = p.Payment_ID
INNER JOIN students s ON e.E_SID = s.SID
GROUP BY Payment_Type;
```

Result:

Payment_Type	Total student in each payment
Counter Service	6
Mobile Banking	13
Krung Thai Bank	14



## Appendix

### References

- [1] U. Sivakul, "ChemOu," [Online]. Available: <https://www.chem-ou.com/School>. [Accessed 24 November 2022].
- [2] U. Sivakul, "ChemOu (Course)," [Online]. Available: <https://www.chem-ou.com/Course/index>. [Accessed 24 November 2022].
- [3] U. Sivakul, "ChemOu (Home)," [Online]. Available: <https://www.chem-ou.com/Home>. [Accessed 24 November 2022].
- [4] U. Sivakul, "ChemOu (App)," [Online]. Available: <https://www.chem-ou.com/Site/download>. [Accessed 24 November 2022].

