Database System: Machine Problem 2

2019010175 Ruiyang Kong

1 Connect to the database

We can use the following statement to connect to the database

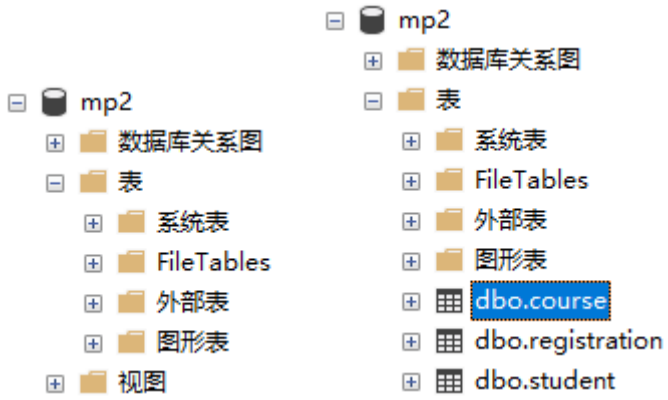
```
>>> import os
>>> os.chdir(r'C:\Users\asus\Desktop\Programming Test')
>>> from task import *
>>> connection = connect()
>>> cur = connection.cursor()
```

Before we create tables, there is no table in the database.

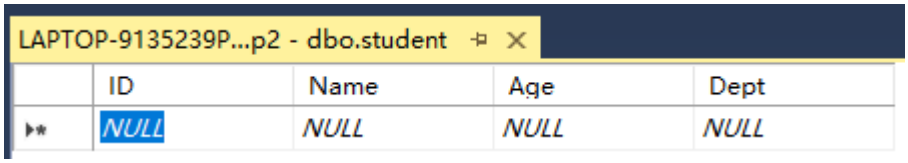
We can use the following statement to create tables.

```
>>> create_tables(cur)
```

After that, we can see the tables in the database.

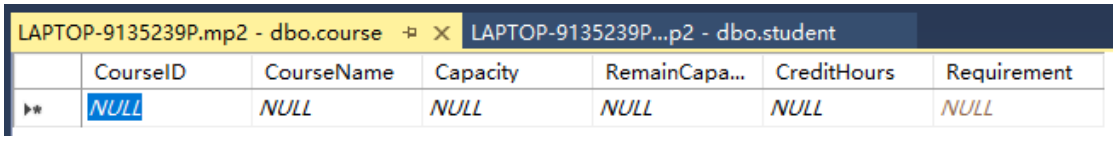


And now all tables are empty.



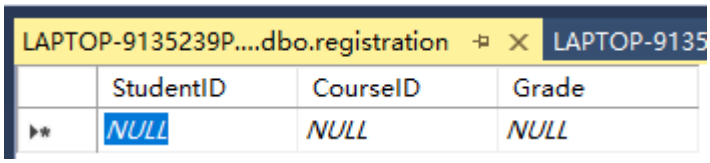
LAPTOP-9135239P...p2 - dbo.student

	ID	Name	Age	Dept
▶*	NULL	NULL	NULL	NULL



LAPTOP-9135239P.mp2 - dbo.course | LAPTOP-9135239P...p2 - dbo.student

	CourseID	CourseName	Capacity	RemainCapa...	CreditHours	Requirement
▶*	NULL	NULL	NULL	NULL	NULL	NULL



LAPTOP-9135239P....dbo.registration | LAPTOP-9135

	StudentID	CourseID	Grade
▶*	NULL	NULL	NULL

2 Insert data

Firstly we can insert the data in the given example.

```
>>> insert_data(cur)
```

LAPTOP-9135239P.mp2 - dbo.course

	CourseID	CourseName	Capacity	RemainCapa...	CreditHours	Requirement
▶	101	Introduction t...	4	3	3	<Req />
	102	C++	3	3	3	<Req><Prere...
*	NULL	NULL	NULL	NULL	NULL	NULL

LAPTOP-9135239P...p2 - dbo.student | LAPTOP-9135239P.mp2 - dbo.

	ID	Name	Age	Dept
▶	1000113	Wade Williams	20	CS
	1000214	Dave Harris	20	CS
	1002079	Daisy White	21	EE
	1002098	Ivan Scott	18	EE
	1003342	Lucy Clark	21	CS
*	NULL	NULL	NULL	NULL

LAPTOP-9135239P....dbo.registration | LAPTOP-9135

	StudentID	CourseID	Grade
▶	1000113	101	99
*	NULL	NULL	NULL

Also we can insert single student/course by following statement.

```
>>> insert_student(cur, 1000000, 'Krydom', 13, 'IIIS')
```

LAPTOP-9135239P...p2 - dbo.student | LAPTOP-9135239P....dbo.regist

	ID	Name	Age	Dept
▶	1000000	Krydom	13	IIIS
	1000113	Wade Williams	20	CS
	1000214	Dave Harris	20	CS
	1002079	Daisy White	21	EE
	1002098	Ivan Scott	18	EE
	1003342	Lucy Clark	21	CS
*	NULL	NULL	NULL	NULL

```
>>> insert_course(cur, 100, 'Database System', 3, 3, 3, '<Req></Req>')
```

LAPTOP-9135239P.mp2 - dbo.course | LAPTOP-9135239P...p2 - dbo.student | LAPTOP-9135239P.

	CourseID	CourseName	Capacity	RemainCapa...	CreditHours	Requirement
▶	100	Database Syst...	3	3	3	<Req />
	101	Introduction t...	4	3	3	<Req />
	102	C++	3	3	3	<Req><Prere...
*	NULL	NULL	NULL	NULL	NULL	NULL

3 Delete data

We can use the following statement to delete a student/course from the database.

```
>>> delete_student(cur, 1000000)
```

LAPTOP-9135239P...p2 - dbo.student | LAPTOP-9135239P.mp2 - dbo.

ID	Name	Age	Dept
1000113	Wade Williams	20	CS
1000214	Dave Harris	20	CS
1002079	Daisy White	21	EE
1002098	Ivan Scott	18	EE
1003342	Lucy Clark	21	CS
NULL	*NULL*	*NULL*	*NULL*

```
>>> delete_course(cur, 100)
```

LAPTOP-9135239P.mp2 - dbo.course | LAPTOP-9135239P...p2 - dbo.student | LAPTOP-9135239P.

CourseID	CourseName	Capacity	RemainCapa...	CreditHours	Requirement
101	Introduction t...	4	3	3	<Req />
102	C++	3	3	3	<Req><Prere...
NULL	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*

4 Number of the students in a course

In the sample, only Wade Williams has taken the course Introduction to CS.
So the number of the students in course which ID=101 is 1, and ID=102, number is 0.

```
>>> count_course_students(cur, 101)
1
>>> count_course_students(cur, 102)
0
```

5 Whether a student satisfies the requirement of a course

Since only Wade Williams has taken the course which ID=101, and the prerequisite course of 102 is 101, so only he can take course 102. And everyone can take course 101.

```
>>> check_requirement(cur, 1000113, 102)
True
>>> check_requirement(cur, 1000214, 102)
False
>>> check_requirement(cur, 1000214, 101)
True
```

6 Register a student to a course

Firstly, if a student has not taken the prerequisite course, he cannot be registered.

```
>>> register_student(cur, 1000214, 102)
The student can not meet the course requirements.
False
```

Otherwise, the student can be added to a course if the capacity has remaining.surplus.

```
>>> register_student(cur, 1000214, 101)
Successfully Registered the student to the course.
True
>>> register_student(cur, 1002079, 101)
Successfully Registered the student to the course.
True
>>> register_student(cur, 1002098, 101)
Successfully Registered the student to the course.
True
```

And now the course/register table are as follow:

LAPTOP-9135239P.mp2 - dbo.course | LAPTOP-9135239P....dbo.registration | LAPTOP-9135239P..

	CourseID	CourseName	Capacity	RemainCapa...	CreditHours	Requirement
▶	101	Introduction t...	4	0	3	<Req />
	102	C++	3	3	3	<Req><Prere...
*	NULL	NULL	NULL	NULL	NULL	NULL

LAPTOP-9135239P.mp2 - dbo.course | LAPTOP-9135

	StudentID	CourseID	Grade
▶	1000113	101	99
	1000214	101	0
	1002079	101	0
	1002098	101	0
*	NULL	NULL	NULL

But when the number of students has been at limit, the operation will fail.

```
>>> register_student(cur, 1003342, 101)
The course has no remainCapacity. Failed to register.
False
```

7 Remove a student from a course

We can use the following statement to remove a student from a course.

```
>>> remove_student(cur, 1002098, 101)
```

After the operation, the course/register table are as follow:

LAPTOP-9135239P....dbo.registration | LAPTOP-9135239P.mp2 - dbo.course

	CourseID	CourseName	Capacity	RemainCapa...	CreditHours	Requirement
▶	101	Introduction t...	4	1	3	<Req />
	102	C++	3	3	3	<Req><Prere...
*	NULL	NULL	NULL	NULL	NULL	NULL

LAPTOP-9135239P....dbo.registration | LAPTOP-9135

	StudentID	CourseID	Grade
▶	1000113	101	99
	1000214	101	0
	1002079	101	0
*	NULL	NULL	NULL

8 Update the capacity of a course

Reject this operation if the number of students already enrolled exceeds the new capacity.

```
>>> update_capacity(cur, 101, 2)
```

After this operation, the capacity of course 101 is not changed, still 4.

LAPTOP-9135239P.mp2 - dbo.course | LAPTOP-9135239P....dbo.registration

CourseID	CourseName	Capacity	RemainCapa...	CreditHours	Requirement
101	Introduction t...	4	1	3	<Req />
102	C++	3	3	3	<Req><Prere...
NULL	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*

But when if new capacity meet the condition, it will change.

```
>>> update_capacity(cur, 101, 3)
```

LAPTOP-9135239P.mp2 - dbo.course

CourseID	CourseName	Capacity	RemainCapa...	CreditHours	Requirement
101	Introduction t...	3	0	3	<Req />
102	C++	3	3	3	<Req><Prere...
NULL	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*

9 Retrieve the list of courses a student has attended in the past

We firstly register Wade Williams to course 102.

```
>>> register_student(cur, 1000113, 102)
Successfully Registered the student to the course.
True
```

Now he attended both two courses.

```
>>> retrieve_attended(cur, 1000113)
['C++', 'Introduction to CS']
```

Dave Harris and Daisy White only attended course 101.

```
>>> retrieve_attended(cur, 1000214)
['Introduction to CS']
```

Ivan Scott and Lucy Clark didn't attend any course.

```
>>> retrieve_attended(cur, 1002098)
[]
```

10 Update the grade of a student in a course

We can Dave Harris and Daisy White's grade in course 101 by 60, 59.

```
>>> update_grade(cur, 1000214, 101, 60)
>>> update_grade(cur, 1002079, 101, 59)
```

LAPTOP-9135239P....dbo.registration | LAPTOP-91352

StudentID	CourseID	Grade
1000113	101	99
1000214	101	60
1002079	101	59
1000113	102	0
NULL	*NULL*	*NULL*

11 Retrieve all the course failure records

All the failure records are (1002079,101,59) and (1000113,102,0).
With names, (Daisy White , Introduction to CS) and (Wade Williams , C++)

```
>>> retrieve_failure(cur)
( Daisy White , Introduction to CS )
( Wade Williams , C++ )
```

12 Compute the GPA of a student

Wade Williams' grade of 101 and 102 is 99 and 0, so GPA is 4.0 and 0.0.
So his final GPA is (4.0+0.0)/2=2.0

```
>>> compute_gpa(cur, 1000113)
Decimal('2.000000')
```

13 Compute the average grade of students in a course

For course 101, the average grade is (99+60+59)/3=72.667, 72 after rounding.

```
>>> course_avg(cur, 101)
72
```