

Database   
Experiment Report

Faculty 计算机全英联合班

Name 邓璟、徐晨舒

Student ID 邓璟 201730700165

徐晨舒201730614370

Lecturer 王昊翔

Experiment Title Task 2. Design a MIS for Computer Science college of SCUT

|  |
| --- |
| **Description** |
| 【Objective and Requirement】  Objective：  1. Design and implement a student information system  2. Info about student, course, teacher, taking course  3. Queries  Requirement：  No constraint on DBMS or programming language.  【Environment】  Operating System：Windows 10 |
| **ER Diagram & Relation Model** |
|  |
| **SQL commands** |
| In this experiment, we first build a database named Lab\_2 on MYSQL, then we use a library of python, pymysql to connect MYSQL with the interface.   * The code that we build Lab\_2 and create tables in MYSQL are shown as follows:   CREATE DATABASE if not exists Lab\_2;  USE Lab\_2;  create table if not exists Student (  primary key(ID),  ID char(10) not null,  S\_Name varchar(20),  Gender varchar(10),  age int,  school\_year int,  class int  );  insert into Student values('0000000001', 'Tony', 'Male', 20, 2016, 1);  insert into Student values('0000000002', 'Tom', 'Male', 19, 2017, 2);  insert into Student values('0000000003', 'Jane', 'Female', 20, 2016, 3);  insert into Student values('0000000004', 'Nick', 'Male', 20, 2015, 4);  insert into Student values('0000000005', 'Mike', 'Male', 19, 2017, 3);  insert into Student values('0000000006', 'Sam', 'Male', 20, 2015, 4);  insert into Student values('0000000007', 'Marry', 'Female', 19, 2016, 5);  insert into Student values('0000000008', 'Pat', 'Female', 20, 2015, 6);  insert into Student values('0000000009', 'Shurjo', 'Male', 21, 2016, 6);  insert into Student values('0000000010', 'Vital', 'Male', 20, 2015, 6);  insert into Student values('0000000011', 'Trump', 'Male', 18, 2017, 7);  insert into Student values('0000000012', 'Melania', 'Female', 20, 2015, 7);  insert into Student values('0000000013', 'Pompeo', 'Male', 20, 2015, 8);  insert into Student values('0000000014', 'Matisse', 'Male', 19, 2016, 8);  insert into Student values('0000000015', 'Bush', 'Male', 19, 2016, 9);  create table if not exists Class (  primary key(ID),  ID char(7) not null,  C\_Name varchar(20),  Teacher\_ID char(5),  Credit int,  Suitable\_Grade int,  Cancel\_Year int  );  insert into Class values('0000001', 'Chinese', '00001', 1, 2016, 2020);  insert into Class values('0000002', 'Math', '00002', 2, 2017, 2020);  insert into Class values('0000003', 'English', '00003', 1, 2016, 2020);  insert into Class values('0000004', 'Database', '00004', 2, 2016, 2020);  insert into Class values('0000005', 'CPP', '00005', 2, 2016, 2020);  insert into Class values('0000006', 'AI', '00006', 1, 2016, 2020);  insert into Class values('0000007', 'Algorithm', '00007', 2, 2016, 2020);  create table if not exists Teacher (  primary key(ID),  ID char(5) not null,  T\_Name varchar(20),  Course varchar(20),  Administrator int  );  insert into Teacher values('00001', 'Professor\_Chan', 'Chinese', 1);  insert into Teacher values('00002', 'Professor\_Yu', 'Database', 1);  insert into Teacher values('00003', 'Professor\_Liu', 'Math', 2);  insert into Teacher values('00004', 'Professor\_Xu', 'English', 1);  insert into Teacher values('00005', 'Professor\_Zheng', 'CPP', 1);  insert into Teacher values('00006', 'Professor\_Ng', 'AI', 2);  insert into Teacher values('00007', 'Professor\_Yan', 'Algorithm', 1);  create table if not exists Choose (    Student\_ID char(10) not null,  Course\_ID char(7) not null,  Teacher\_ID char(5) not null,  Choose\_Year int,  Grade int,  foreign key(Student\_ID) references Student(ID),  foreign key(Course\_ID) references Class(ID),  foreign key(Teacher\_ID) references Teacher(ID)  );  insert into Choose values('0000000001', '0000001', '00001', 2016, 95);  insert into Choose values('0000000002', '0000002', '00002', 2017, 92);  insert into Choose values('0000000003', '0000003', '00003', 2016, 95);  insert into Choose values('0000000004', '0000004', '00004', 2016, 89);  insert into Choose values('0000000005', '0000002', '00002', 2016, 87);  insert into Choose values('0000000006', '0000004', '00004', 2016, 94);  insert into Choose values('0000000007', '0000001', '00001', 2015, 67);  insert into Choose values('0000000008', '0000005', '00005', 2016, 95);  insert into Choose values('0000000009', '0000005', '00005', 2015, 78);  insert into Choose values('0000000010', '0000002', '00002', 2016, 67);  insert into Choose values('0000000011', '0000005', '00005', 2016, 59);  insert into Choose values('0000000012', '0000006', '00006', 2016, 86);  insert into Choose values('0000000013', '0000006', '00006', 2016, 87);  insert into Choose values('0000000014', '0000007', '00007', 2016, 95);  insert into Choose values('0000000015', '0000007', '00007', 2016, 92);   * Then we use pymysql, a library of python to execute sql code in python.  1. Search and Query: 2. Student’s information can be queried by inputting students’ ID or name or both.  1. Score information can be queried with students’ name or ID or both.  1. Course choosing information can be queried with course ID or name or both.  1. Teachers’ information can be queried with teachers’ name or ID or both.  * Average score of a student can be queried with student’s ID. * Average score of all students can be queried by pressing the search button without inputting any information. * Average score of students taking the same course can be queried with student’s ID and course ID. * Average score of students in the same class can be queried with class number.  1. Insert information: 2. insert a student’s information:   There will be a warning popup message when age is not in range [10, 50].   1. insert a course information:  1. insert a course choosing information:   Before inserting a course choosing information, two constraints will be checked. First, whether the teacher is teaching the corresponding course. Second, whether the students’ school year is less than the suitable grade of the course. Either of them is not satisfied, the insertion will fail.   1. Update information: 2. Update a student’s information:   Any single attribution can be updated separately except the student’s ID.   1. Update a course information:   Any single attribution can be updated separately except the course ID.   1. Update a course choosing information:   Before update the information, the relationship between teacher and course will be checked, if it is not corresponding, a warning message will popup.  If you are a teacher, after updating a student’s course choosing information, the score of the new course will be set to “null”, which is very reasonable.   1. Delete information: 2. Delete student’s information:   Before deleting a piece of student’s information, the corresponding course choosing information will be deleted first, which make sure that the student to be deleted choose no course.   1. Delete course information:   Before deleting a piece of course information, the corresponding course choosing information will be deleted first, which make sure that no one choose the course to be deleted.   1. Delete course choosing information: |
| **Program manual** |
| * Login surface  * Login as an administrator:   An administrator not only can modify information of student, course and course choosing, but also query all the mentioned information:   1. Modify students’ information:   An administrator can insert, update and delete a student’s information by first searching the student with his ID or name or both.  Message will be popup when the operation is successful.  After deleting, search again, the student is not found.  When insert or update, if the student’s age is not in range [10, 50], a warning will popup.   1. Modify course information:   An administrator can insert, update and delete a course information by first searching the course with its ID or name or both.  Message will be popup when the operation is successful.  After deleting, search again, the course is not found.   1. Modify course choosing information:   An administrator can insert, update and delete a course choosing information by first searching with students’ ID and course ID or.  Message will be popup when the operation is successful.  The score of the currently inserted course choosing information will be set to none.  After deleting, search again, the course choosing information is not found.  If the student’s grade is larger than the suitable grade of the course, the student can not choose the course.  If the relationship between teacher and course is not corresponding, a warning message will popup.   1. Query:  * Query student’s information:   Search with the student name or ID or both:     * Query score:   Search with the student ID or course ID or both:     * Query course choosing:   Search with the student ID or name or both:     * Query teacher information:   Search with the teacher ID or name or both:     * Query average score information:     Average score for all students Average score of students in the same class  Average score of one student   * Login as a teacher:   A teacher can modify students’ score and query.     * Login as a student:   Go to the query surface directly. A student can only query information, but no modifying any. This surface is the same as administrator’s query surface, except the number of radio button at top. A student can only query information about student, score, course choosing. |
| **Assignment** |
| 邓璟  I am responsible for the SQL code and the connection between SQL and python. At the very beginning, I write and run SQL code on MySQL to build a database named *Lab\_2,* which contains four tables and their own initial records. After debugging each SQL command separately in MySQL, I tried to connect the SQL and python. I used python to connect the database *Lab\_2* and execute SQL command through default surface function in *pymysql*, a python library. Then, for each surface in UI, I use four functions, *search*, *update*, *delete and* *insert,* to satisfy all the requirements about information of students, courses, teachers and course choosing. Finally, input constraints for some data such as student’s age is checked in insert and update operations. I’m also responsible for the report writing.  徐晨舒  I am responsible for the UI design. Firstly, there is a login interface different for administrators, students and teachers. Administrators can access the modifying and query part, while students can only access the query part, which is realized by the option interface. The modifying interface is of three kinds, including student, course and coursing choosing. In course choosing part, teachers can modify a student’s score while administrators cannot. In query interface, there are five query options chosen by radio buttons. And the query result is shown in a table. At last, I reserved interfaces for the SQL command. |
| **Teacher’s Comments and Score** |
| Comment：  Score： Signature：  Date： |