数据库原理实验报告

(2023-2024 年度第二学期)

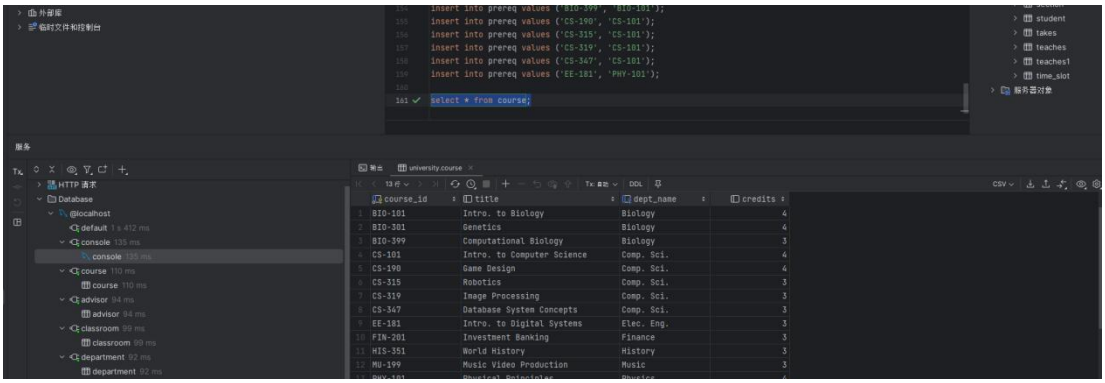
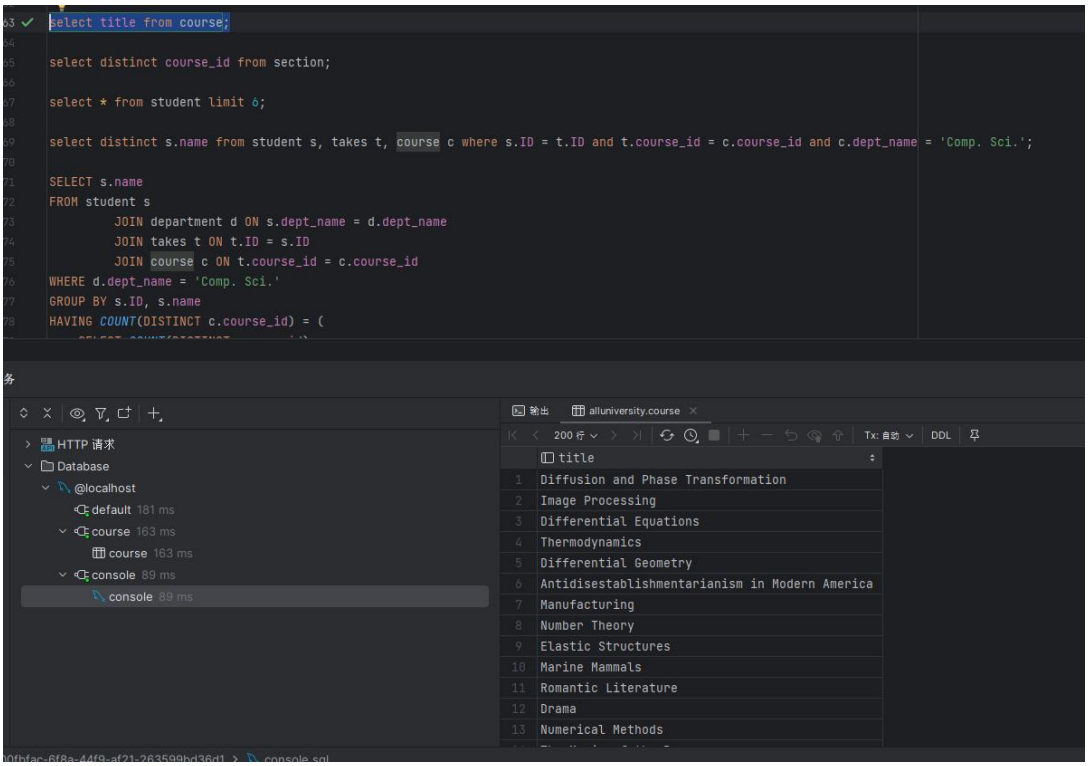
学院：_____软件学院_____

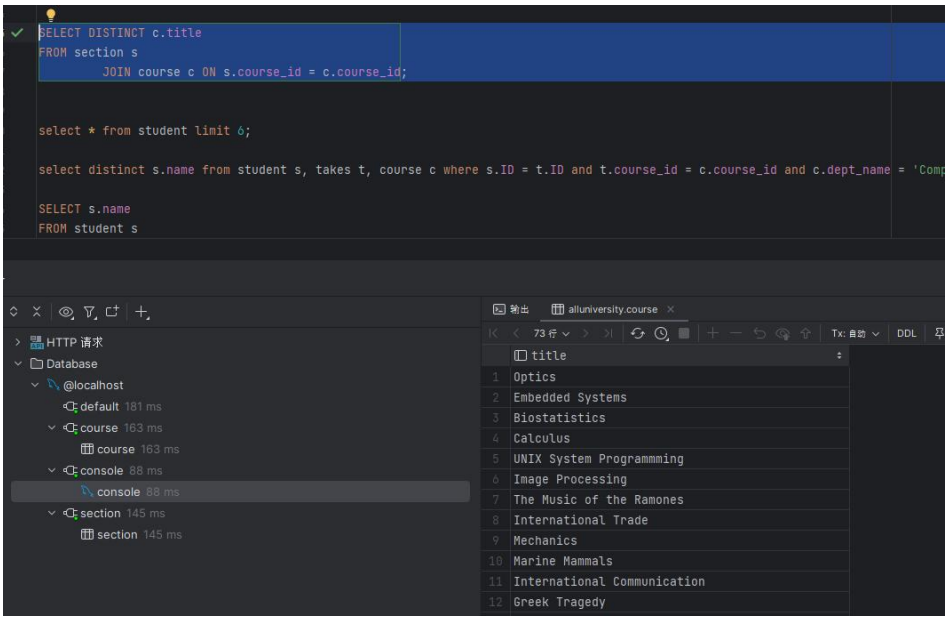
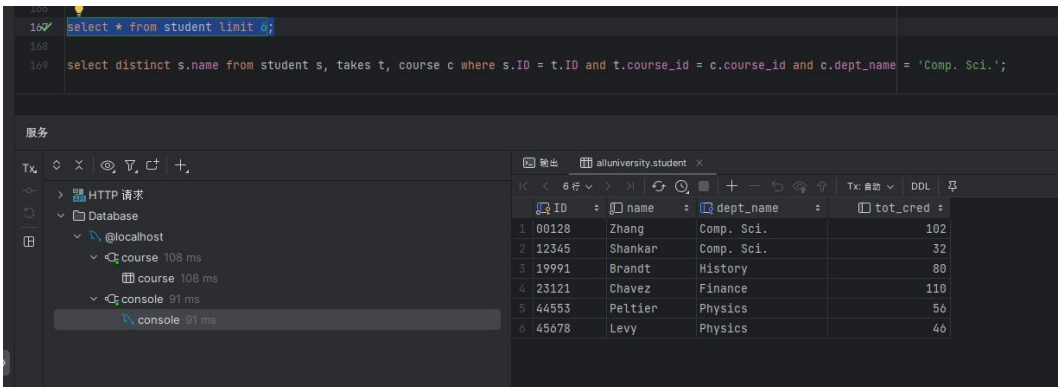
专业：_____软件工程_____

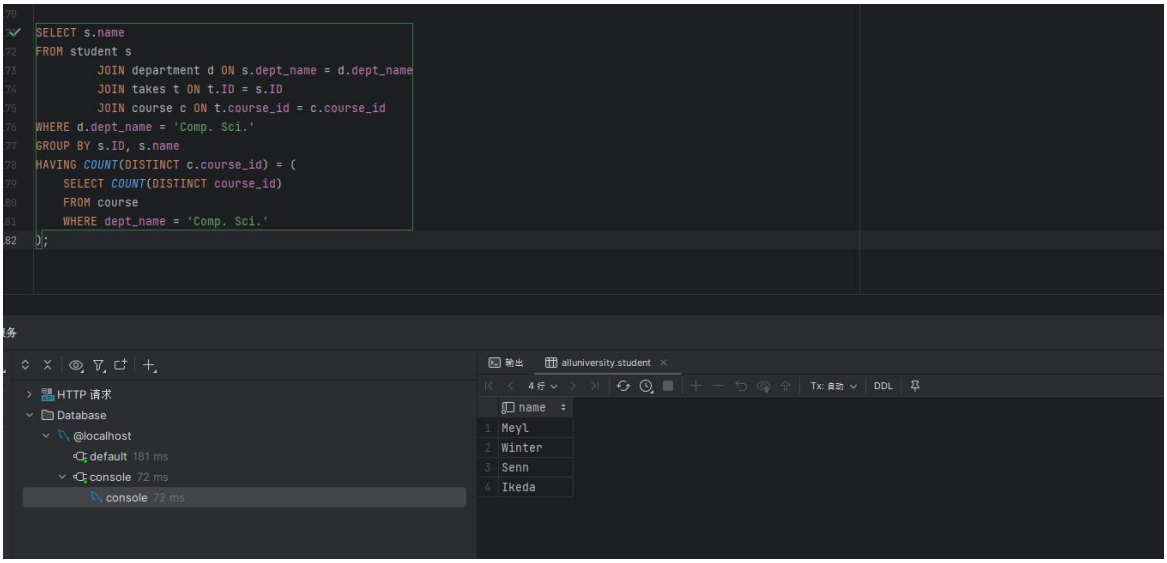
班级：_____552104_____

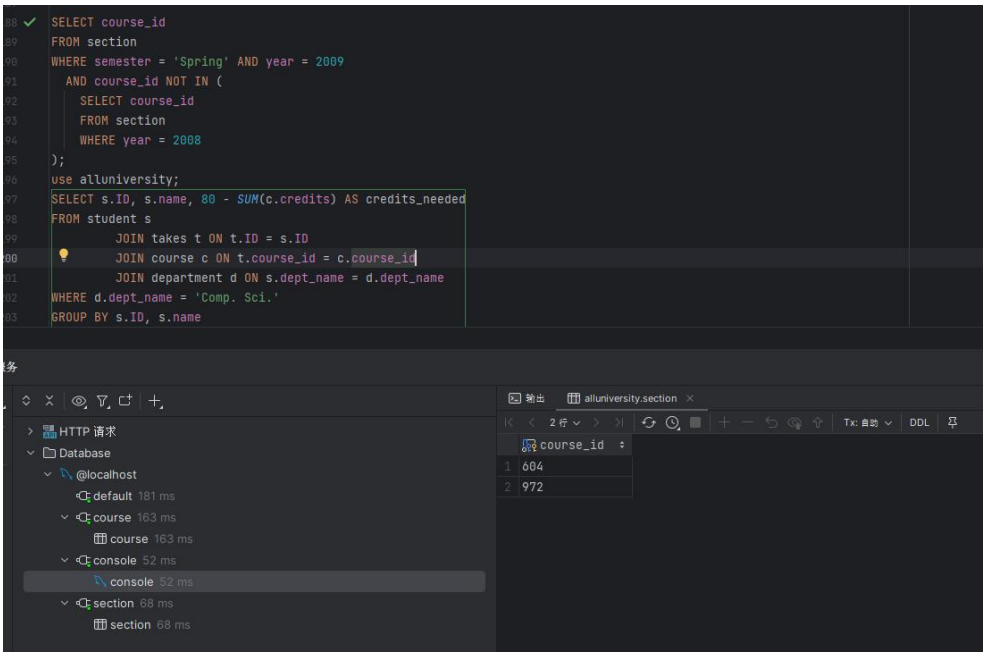
学号：_____55210425_____

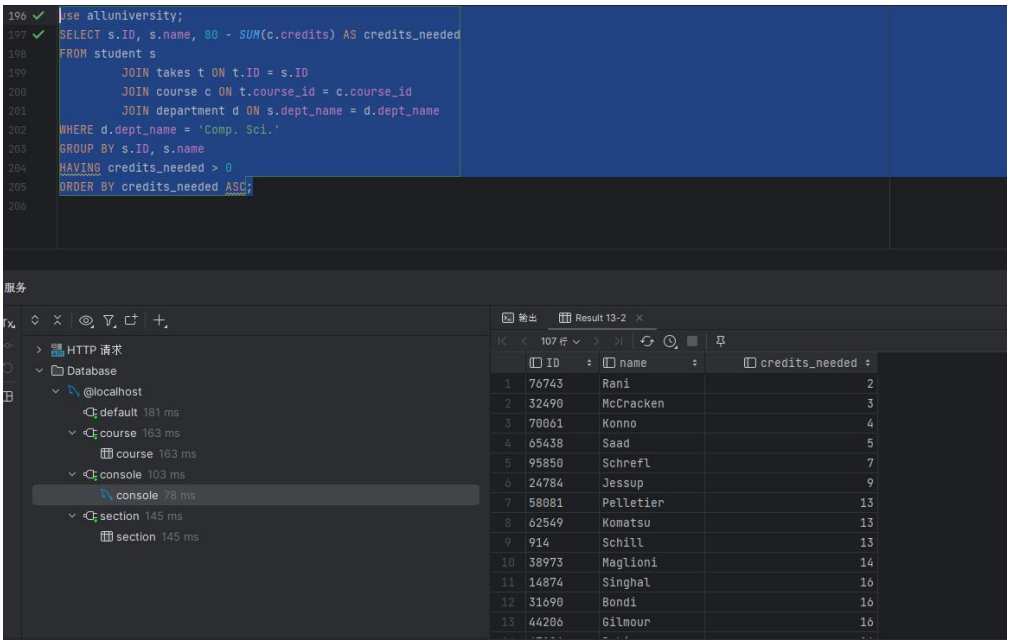
姓名：_____朱家顺_____

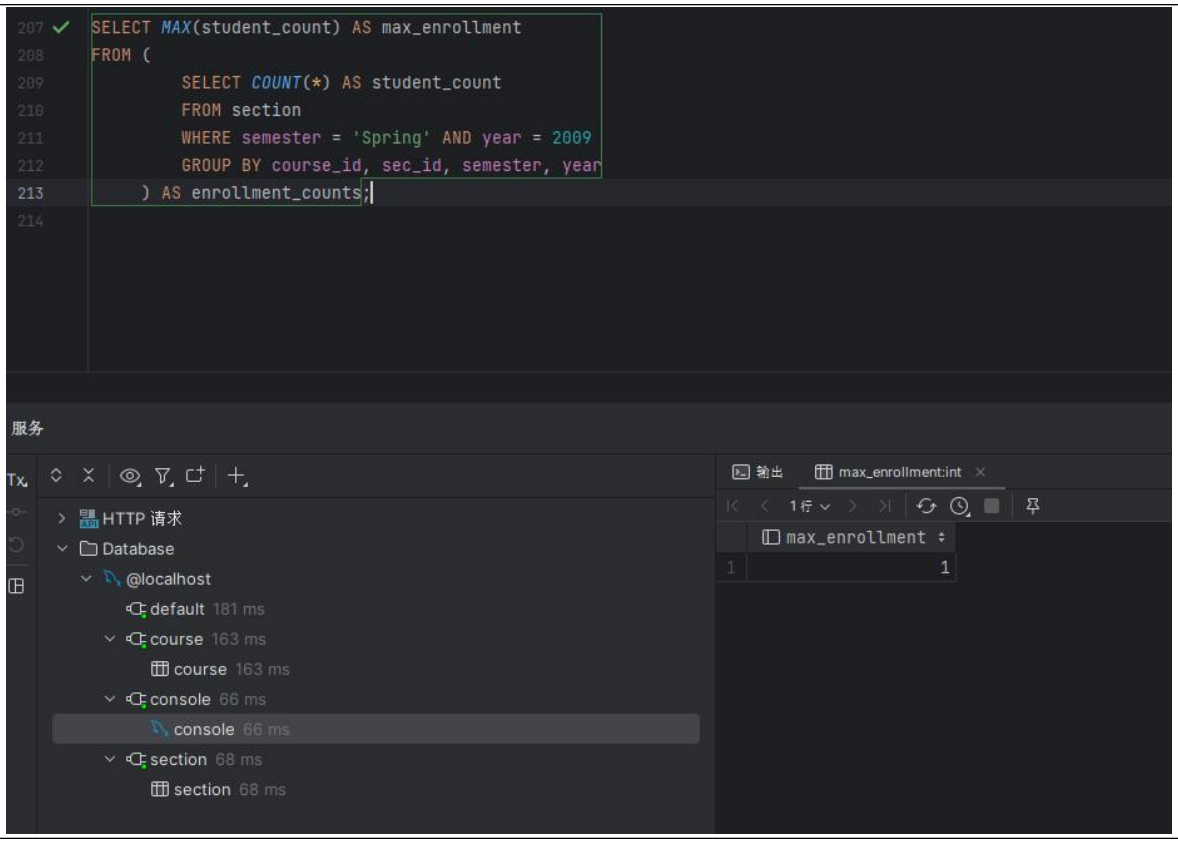
实验环境	Windows 11 家庭中文版（22000.1936）;mysql Ver 8.0.31 for Win64 on x86_64 (MySQL Community Server - GPL)
实验题目1	从课程表（course）中查询所有课程信息。
答案	select * from course;
效果截图	
实验题目2	从课程表中（course）查询课程名。
答案	SELECT title FROM course;
效果截图	
实验题	从课程段（section）表中查询课程名称，要求消除值相同的那些行。

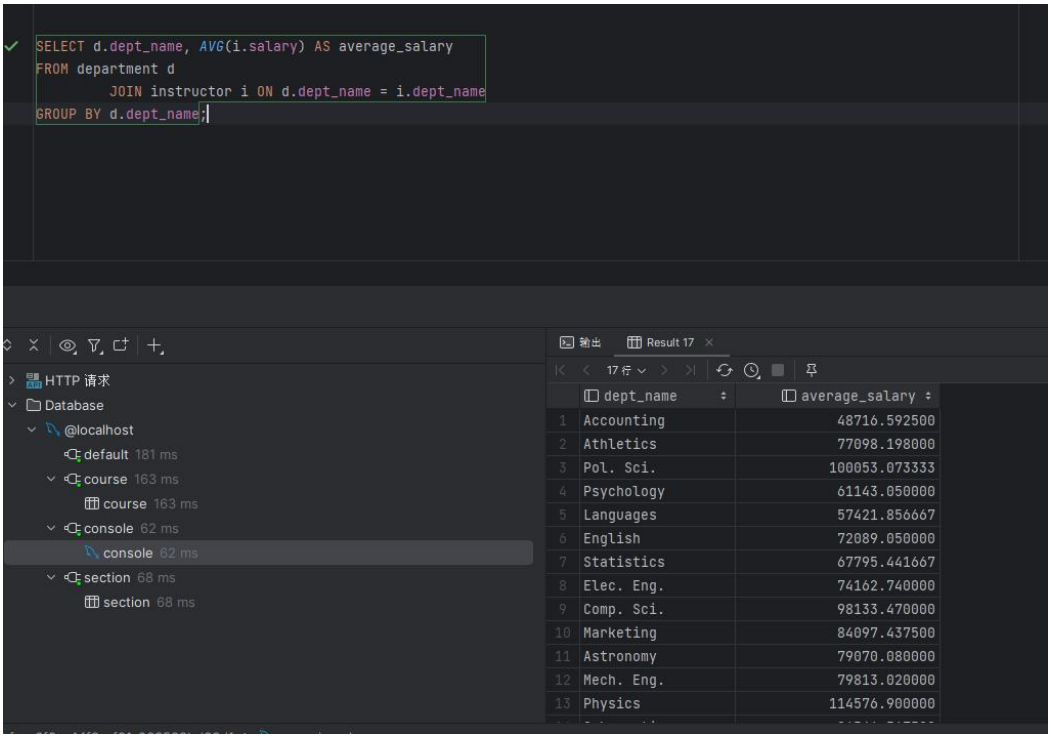
目 3	
答案	SELECT DISTINCT c.title FROM section s JOIN course c ON s.course_id = c.course_id;
效果截图	
实验题目 4	从学生（student）表中查询所有的信息，要求只显示查询结果的前 6 行数据。
答案	select * from student limit 6;
效果截图	
实验题目 5	查询选了所有计算机学院开设课程的学生的姓名。
答案	SELECT s.name FROM student s JOIN department d ON s.dept_name = d.dept_name

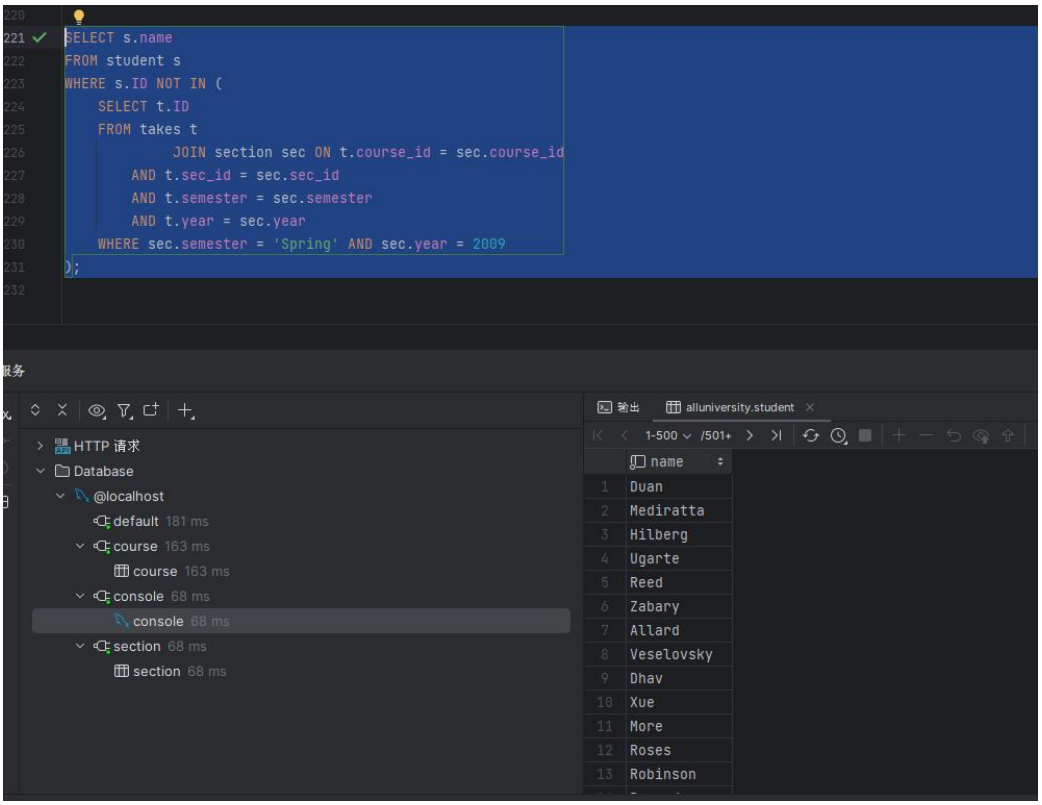
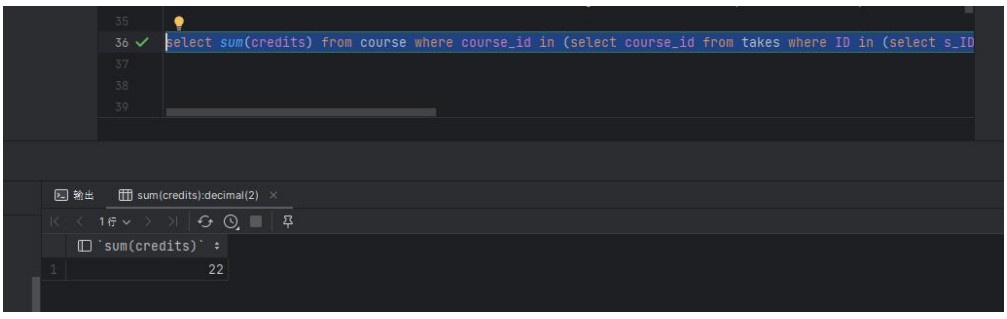
	JOIN takes t ON t.ID = s.ID JOIN course c ON t.course_id = c.course_id WHERE d.dept_name = 'Comp. Sci.' GROUP BY s.ID, s.name HAVING COUNT(DISTINCT c.course_id) = (SELECT COUNT(DISTINCT course_id) FROM course WHERE dept_name = 'Comp. Sci.');
效果截图	
实验题目6	查询2019年春季开课,但2018年不开课的课程的编号。(改为2009和2008)
答案	SELECT course_id FROM section WHERE semester = 'Spring' AND year = 2019 AND course_id NOT IN (SELECT course_id FROM section

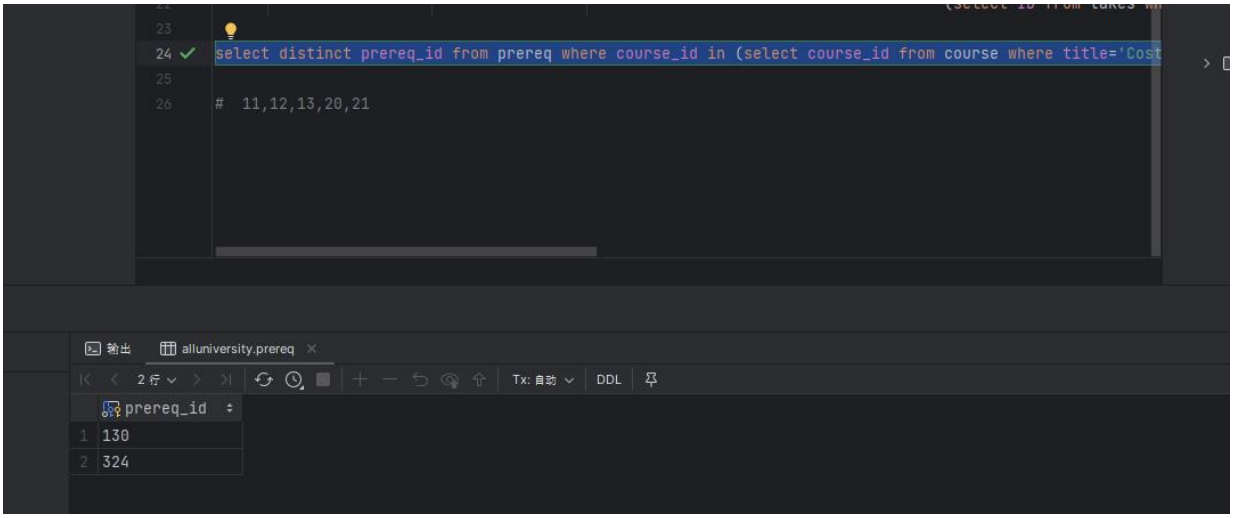
	WHERE year = 2018);
效果截图	
实验题目 7	假设毕业要求为修够 80 学分，请统计计算机学院内学生距离毕业要求还差多少学分，并按所差分数的升序排列。
答案	SELECT s.ID, s.name, 80 - SUM(c.credits) AS credits_needed FROM student s JOIN takes t ON t.ID = s.ID JOIN course c ON t.course_id = c.course_id JOIN department d ON s.dept_name = d.dept_name WHERE d.dept_name = 'Comp. Sci.' GROUP BY s.ID, s.name HAVING credits_needed > 0 ORDER BY credits_needed ASC;

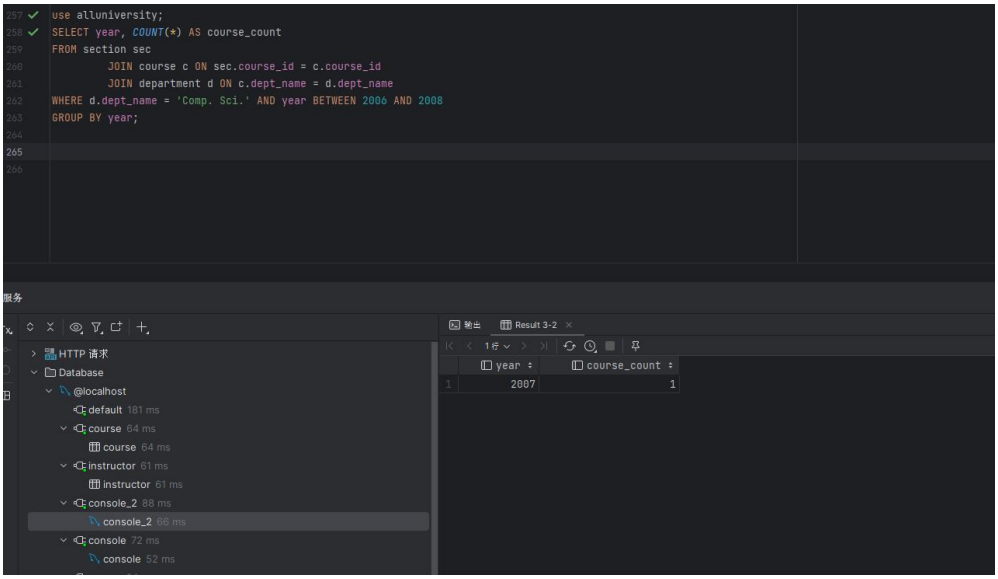
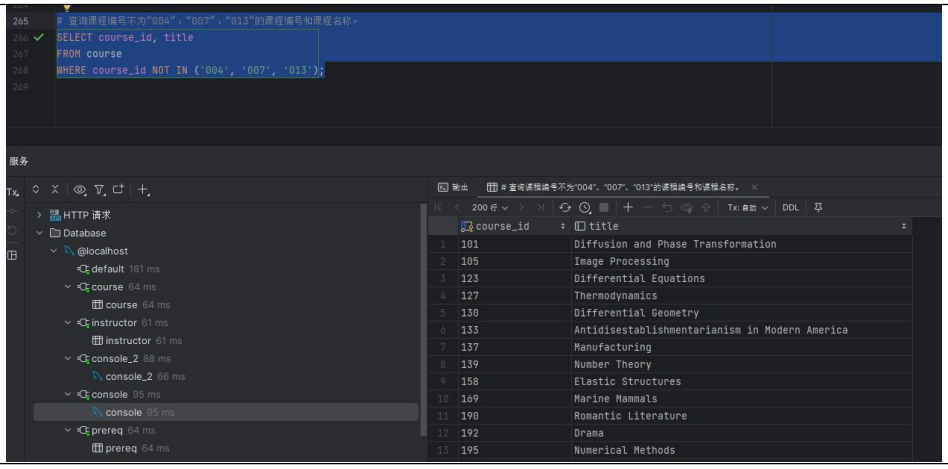
效果截图	
实验题目 8	统计 2019 年春季所开课程段选课人数的最大值。
答案	SELECT MAX(student_count) AS max_enrollment FROM (SELECT COUNT(*) AS student_count FROM section WHERE semester = 'Spring' AND year = 2019 GROUP BY course_id, sec_id, semester, year) AS enrollment_counts;

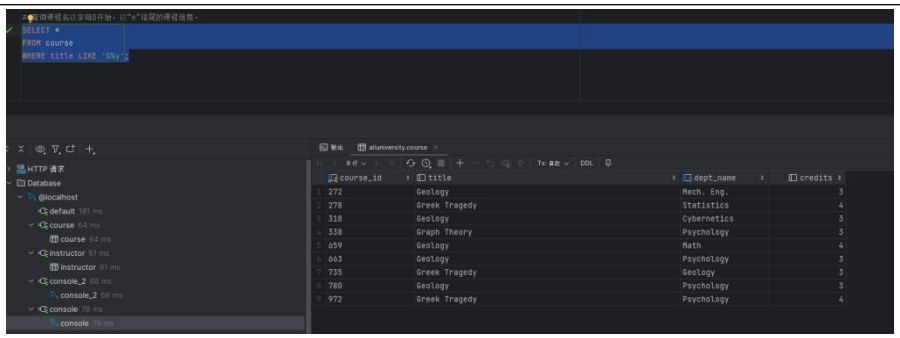
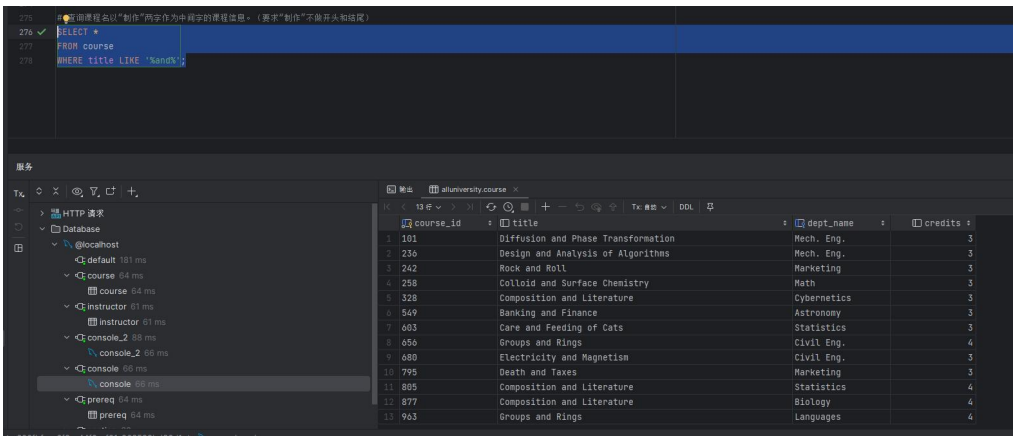
效果截图	
实验题目9	统计各个学院老师的平均年薪。
答案	SELECT d.dept_name, AVG(i.salary) AS average_salary FROM department d JOIN instructor i ON d.dept_name = i.dept_name GROUP BY d.dept_name;

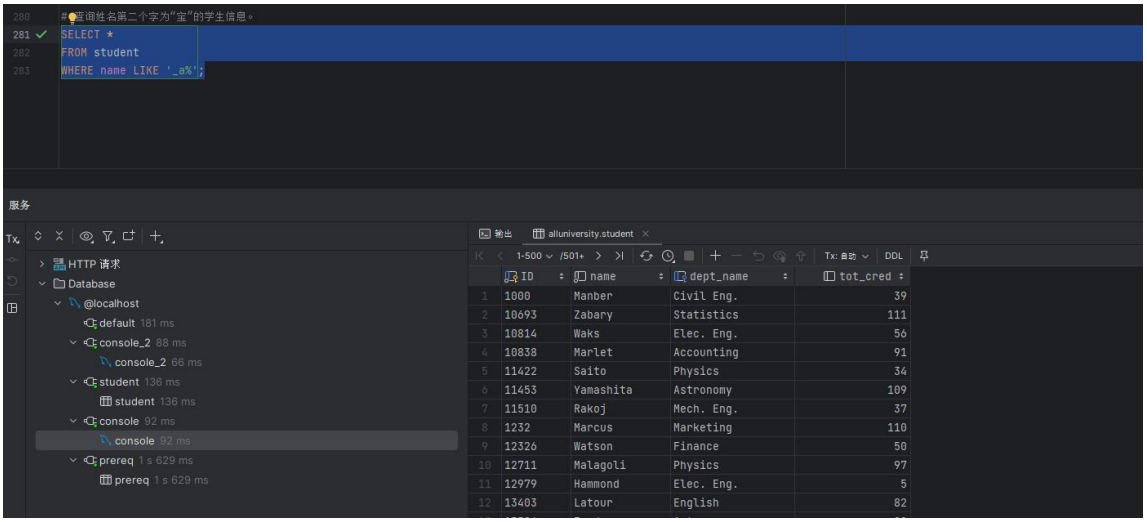
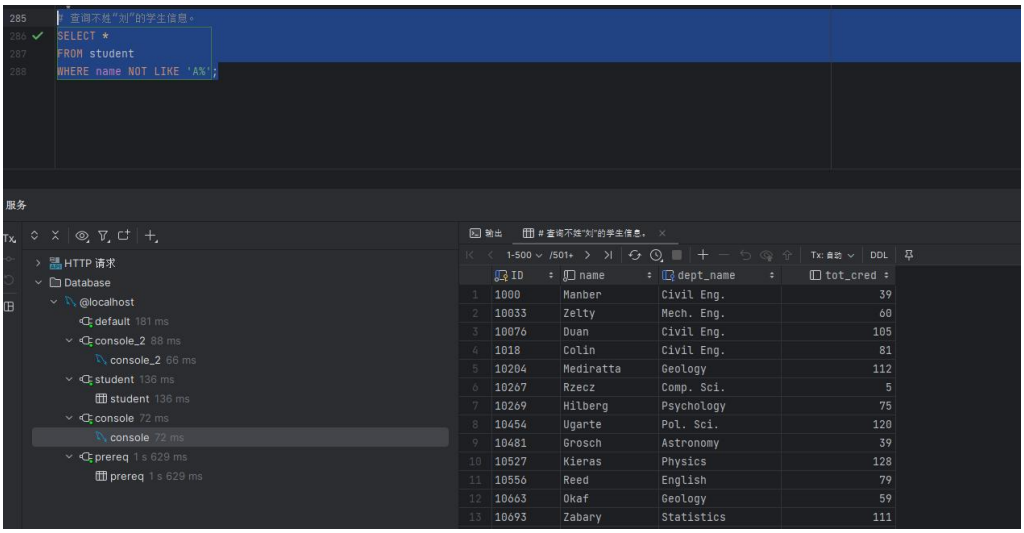
效果截图	
实验题目 10	请输出没有选择 2019 年春季开课的课程段的学生的姓名。（2009）
答案	SELECT s.name FROM student s WHERE s.ID NOT IN (SELECT t.ID FROM takes t JOIN section sec ON t.course_id = sec.course_id AND t.sec_id = sec.sec_id AND t.semester = sec.semester AND t.year = sec.year WHERE sec.semester = 'Spring' AND sec.year = 2009);

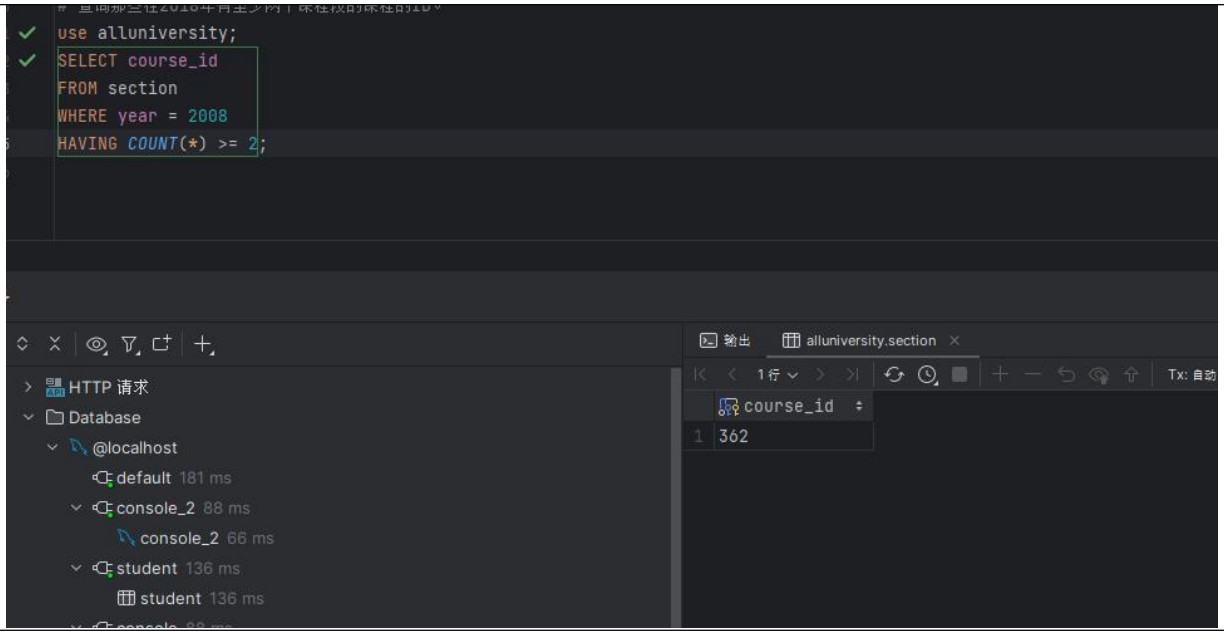
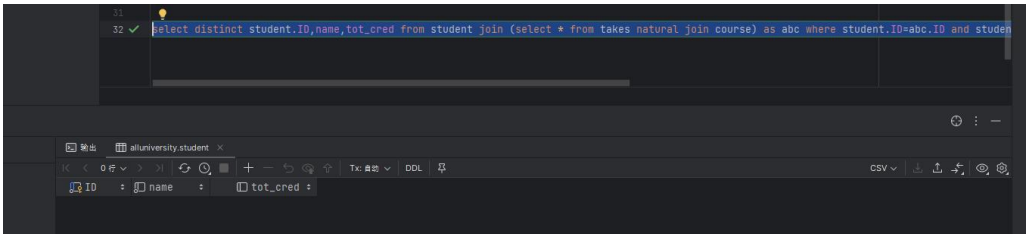
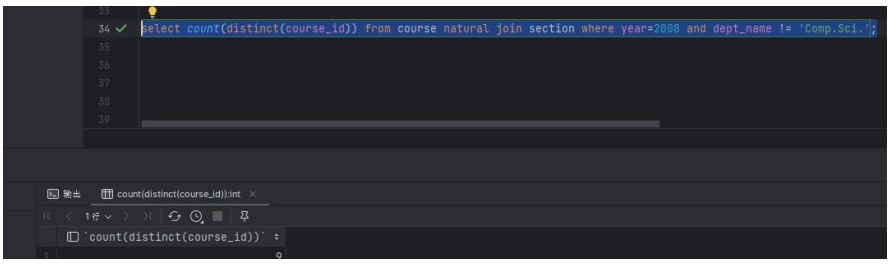
效果截图	
实验题目 11	请输出“张三”指导的学生在 2019 年春季开设课程段中所获得的总学分。（我们默认只要选择某个 course 的一个 section，就可以获得这个 course 的学分；若选择同一个 course 的多个 section，也只获得 1 次这个 course 的学分）。（Mingoz,2009,Fall）
答案	select sum(credits) from course where course_id in (select course_id from takes where ID in (select s_ID as ID from advisor where (s_ID in (select ID from takes where year=2009 and semester='Fall')) and (i_ID in (select ID from instructor where name='Mingoz'))) and semester='Fall' and year=2009);
效果截图	

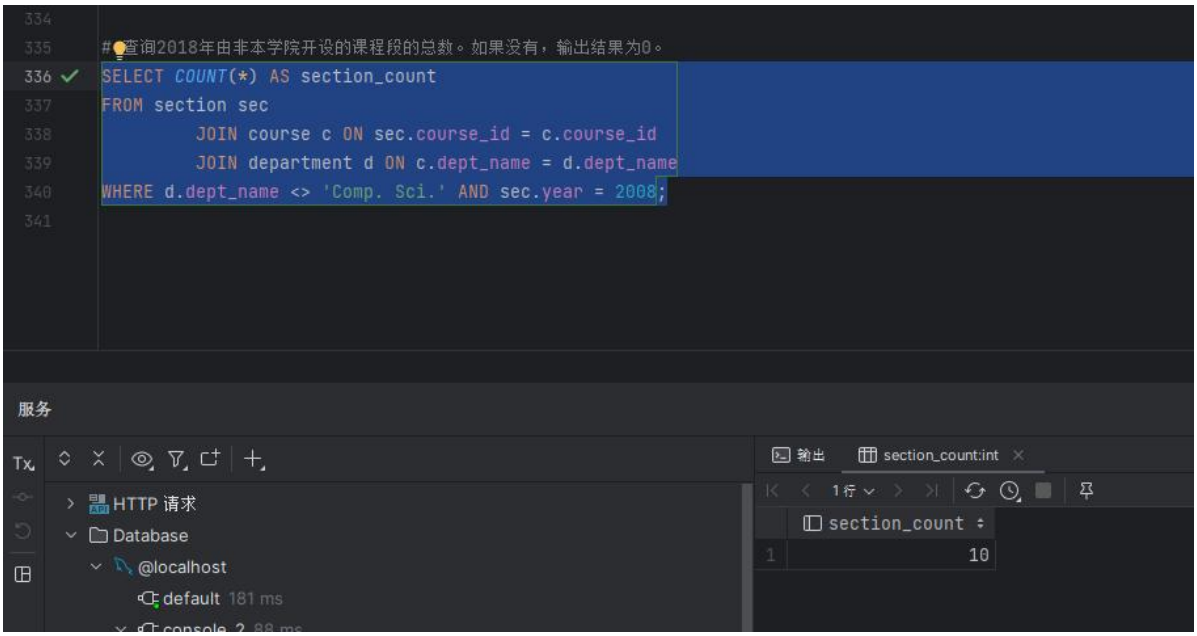
实验题目 12	请输出“数据库系统原理”和“离散数学”的共同的先修课程的 ID。（Cost Accounting 和 Environmental Law）
答案	select distinct prereq_id from prereq where course_id in (select course_id from course where title='Cost Accounting' or title='Environmental Law');
效果截图	
实验题目 13	请统计 2016 年-2018 年，计算机学院每年开设的课程段的数量。(2006-2008)
答案	SELECT year, COUNT(*) AS course_count FROM section sec JOIN course c ON sec.course_id = c.course_id JOIN department d ON c.dept_name = d.dept_name WHERE d.dept_name = 'Comp. Sci.' AND year BETWEEN 2006 AND 2008 GROUP BY year;

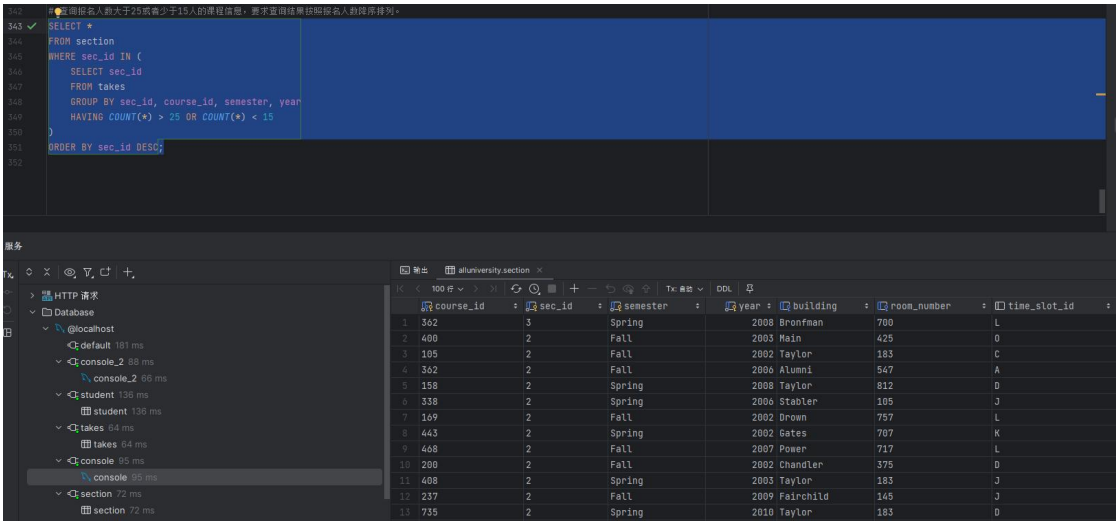
效果截图	
实验题目 14	查询课程编号不为“004”、“007”、“013”的课程编号和课程名称。
答案	SELECT course_id, title FROM course WHERE course_id NOT IN ('004', '007', '013');
效果截图	
实验题目 15	查询课程名以字母 D 开始，以“e”结尾的课程信息。(G y)
答案	SELECT * FROM course WHERE title LIKE 'G%y';

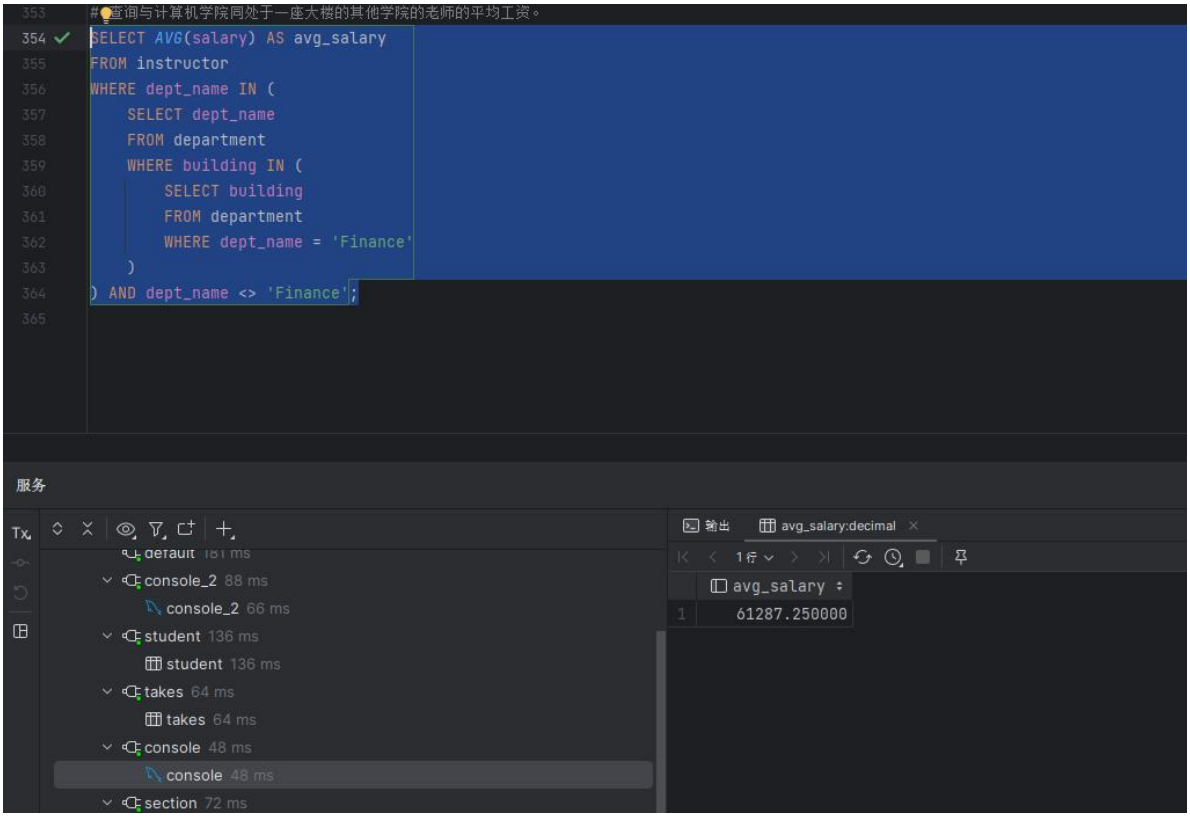
效果截图	
实验题目16	查询课程名以“制作”两字作为中间字的课程信息。（要求“制作”不做开头和结尾）(and)
答案	SELECT * FROM course WHERE title LIKE '%and%';
效果截图	
实验题目17	查询姓名第二个字为“宝”的学生信息。(a)
答案	SELECT * FROM student WHERE name LIKE '_a%';

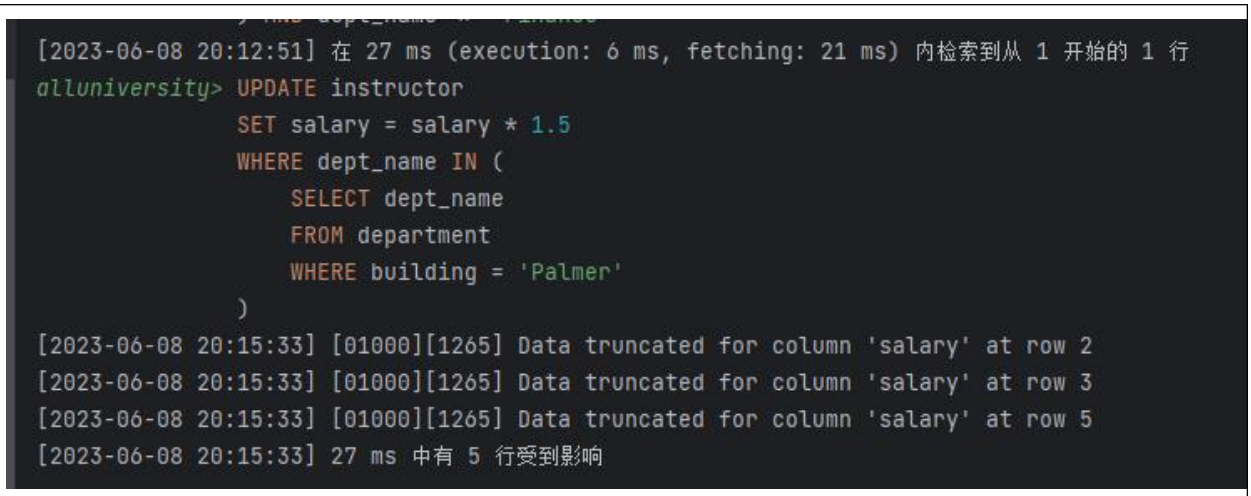
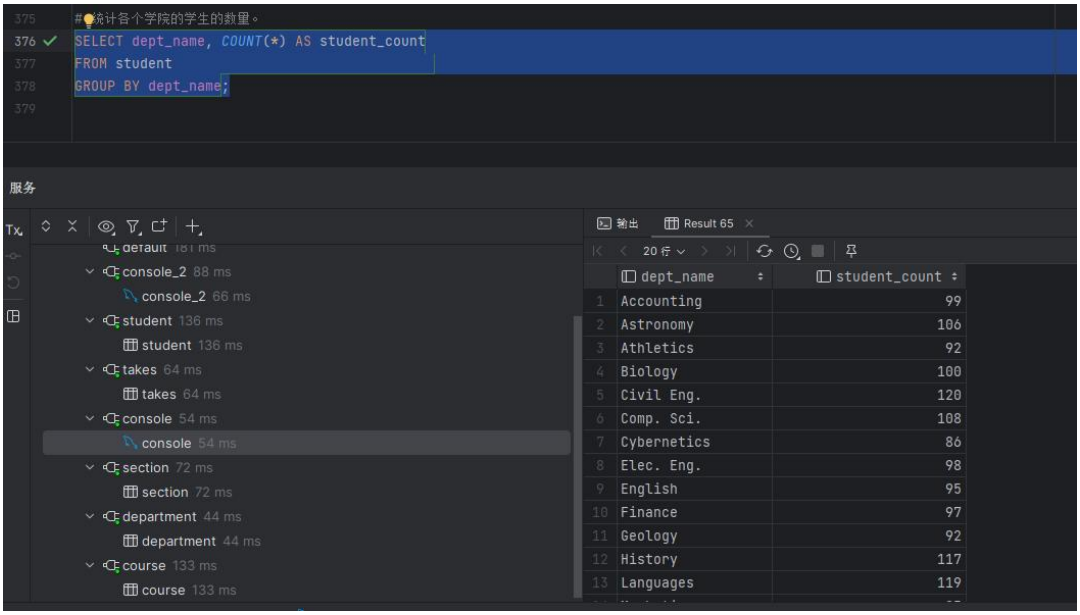
效果截图	
实验题目 18	查询不姓“刘”的学生信息。(A)
答案	SELECT * FROM student WHERE name NOT LIKE 'A%';
效果截图	
实验题目 19	查询那些在 2018 年有至少两个课程段的课程的 ID。(2008)
答案	SELECT course_id FROM section WHERE year = 2008 HAVING COUNT(*) >= 2;

效果截图	
实验题目 20	查询计算机学院学生选了非本学院老师开设的属于本学院的课程的情况，统计这些学生的 ID,姓名，总学分。
答案	select distinct student.ID,name,tot_cred from student join (select * from takes natural join course) as abc where student.ID=abc.ID and student.dept_name!=abc.dept_name and student.dept_name='Comp.Sci.';
效果截图	
实验题目 21	查询 2019 年春季开设的，选课人数少于 25 并且多于 15 人的课程段信息。
答案	select count(distinct(course_id)) from course natural join section where year=2008 and dept_name != 'Comp.Sci.';
效果截图	

实验题目 22	查询 2018 年由非本学院开设的课程段的总数。如果没有，输出结果为 0。
答案	SELECT COUNT(*) AS section_count FROM section sec JOIN course c ON sec.course_id = c.course_id JOIN department d ON c.dept_name = d.dept_name WHERE d.dept_name <> 'Comp. Sci.' AND sec.year = 2008;
效果截图	
实验题目 23	查询报名人数大于 25 或者少于 15 人的课程信息，要求查询结果按照报名人数降序排列。
答案	SELECT * FROM section WHERE sec_id IN (SELECT sec_id FROM takes GROUP BY sec_id, course_id, semester, year HAVING COUNT(*) > 25 OR COUNT(*) < 15

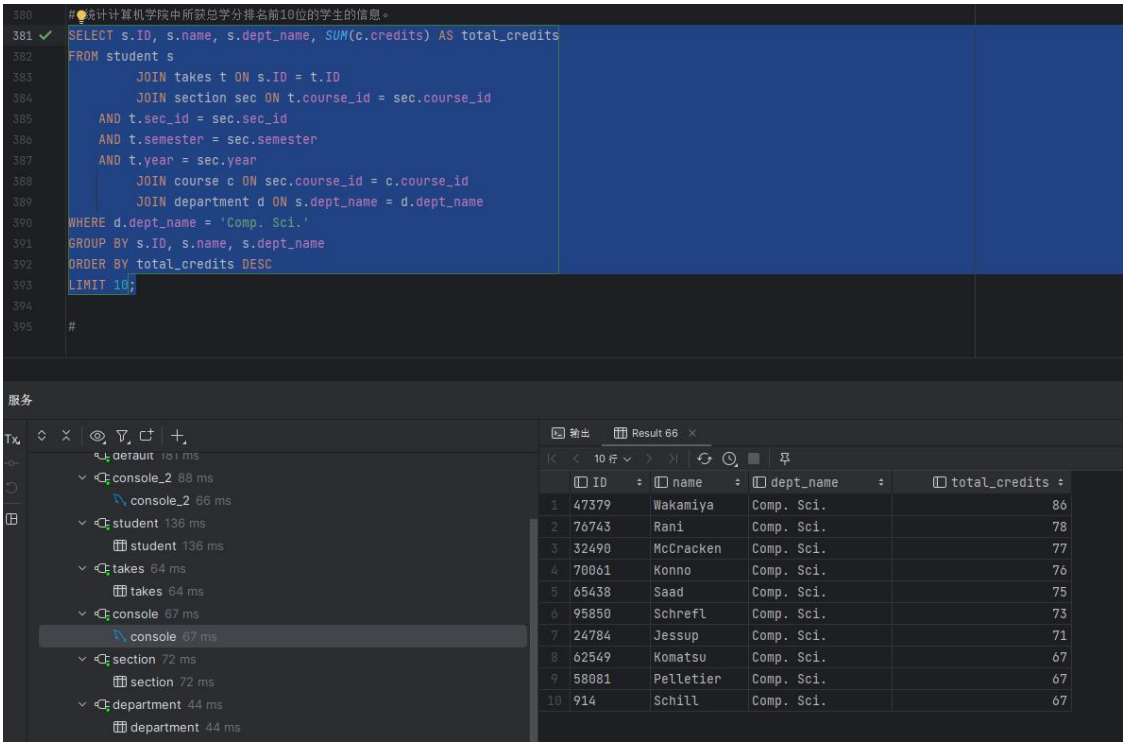
	) ORDER BY sec_id DESC;
效果截图	
实验题目 24	查询与计算机学院同处于一座大楼的其他学院的老师的平均工资。（Finance 经济学院）
答案	SELECT AVG(salary) AS avg_salary FROM instructor WHERE dept_name IN (SELECT dept_name FROM department WHERE building IN (SELECT building FROM department WHERE dept_name = 'Finance')) AND dept_name <> 'Finance';

效果截图	
实验题目 25	给在“匡亚明”大楼内办公的老师的工资增加到原来工资的 1.5 倍。(Palmer)
答案	UPDATE instructor SET salary = salary * 1.5 WHERE dept_name IN (SELECT dept_name FROM department WHERE building = 'Palmer');
效果截图	 # 给在“Palmer”大楼内办公的老师的工资增加到原来工资的1.5倍。 UPDATE instructor SET salary = salary * 1.5 WHERE dept_name IN (SELECT dept_name FROM department WHERE building = 'Palmer');

	
实验题目 26	统计各个学院的学生的数量。
答案	SELECT dept_name, COUNT(*) AS student_count FROM student GROUP BY dept_name;
效果截图	
实验题目 27	统计计算机学院中所获总学分排名前 10 位的学生的信息。
答案	SELECT s.ID, s.name, s.dept_name, SUM(c.credits) AS total_credits FROM student s JOIN takes t ON s.ID = t.ID JOIN section sec ON t.course_id = sec.course_id

```
    AND t.sec_id = sec.sec_id
    AND t.semester = sec.semester
    AND t.year = sec.year
        JOIN course c ON sec.course_id = c.course_id
        JOIN department d ON s.dept_name = d.dept_name
WHERE d.dept_name = 'Comp. Sci.'
GROUP BY s.ID, s.name, s.dept_name
ORDER BY total_credits DESC
LIMIT 10;
```

效果截图

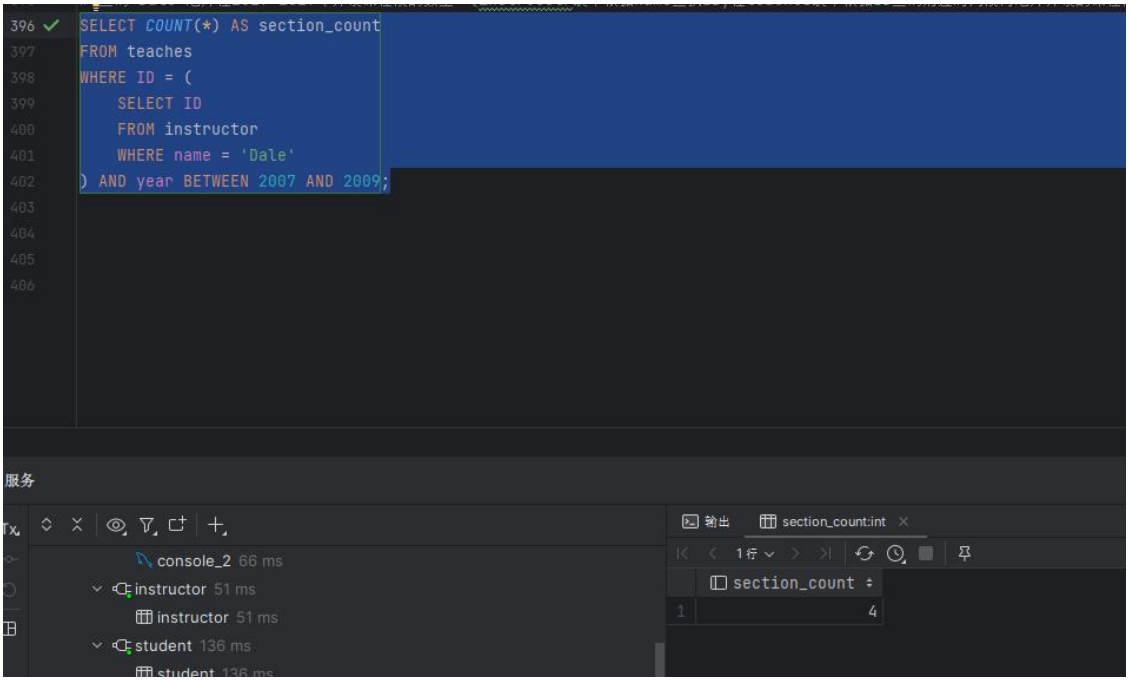


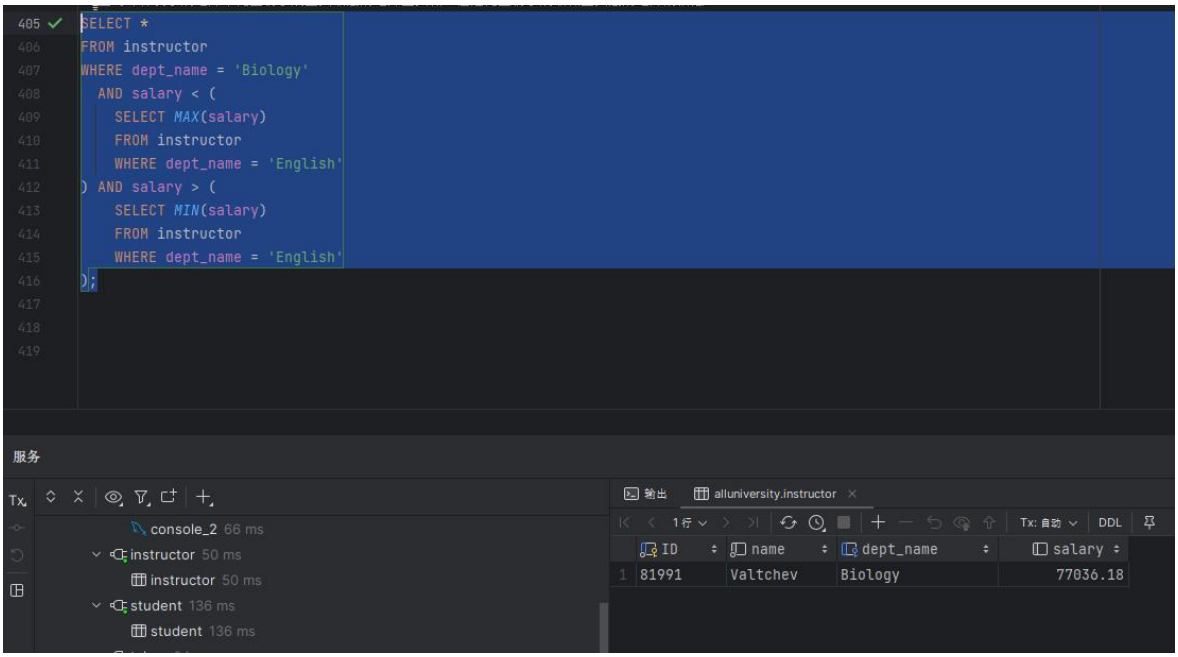
实验题目 28

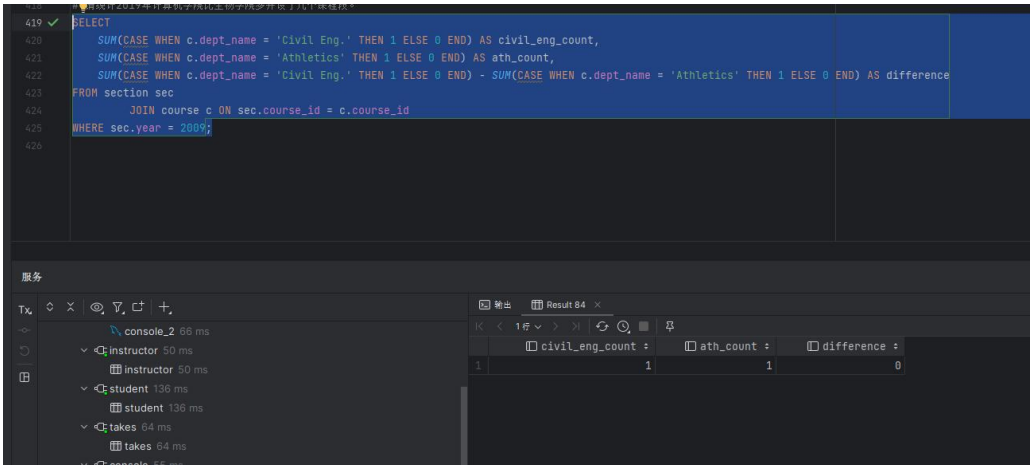
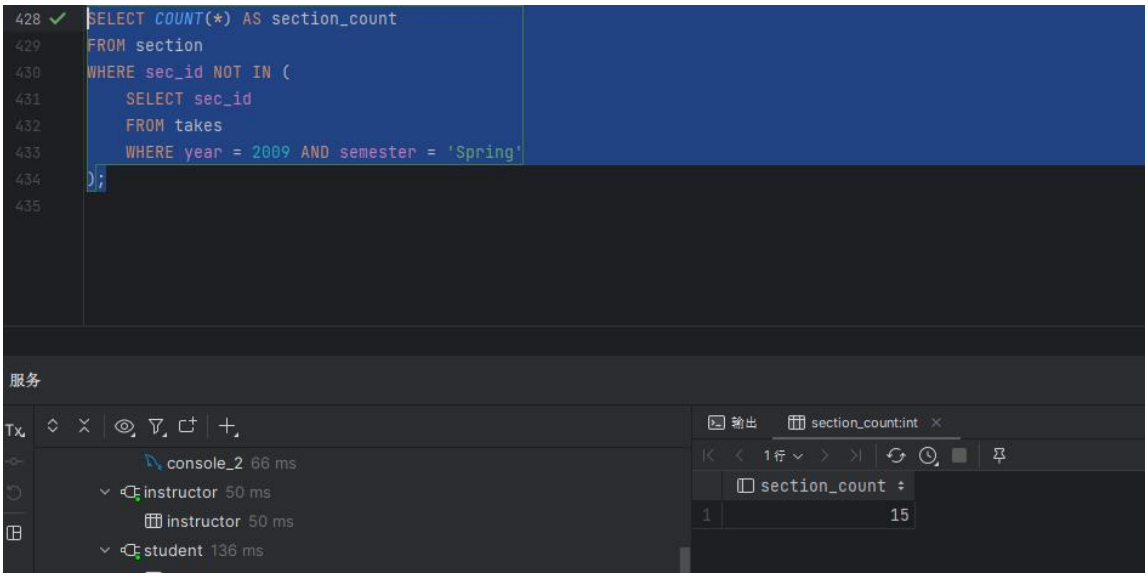
查询“李四”老师在 2017-2019 年开设课程段的数量。(2007-2009,”Dale”)

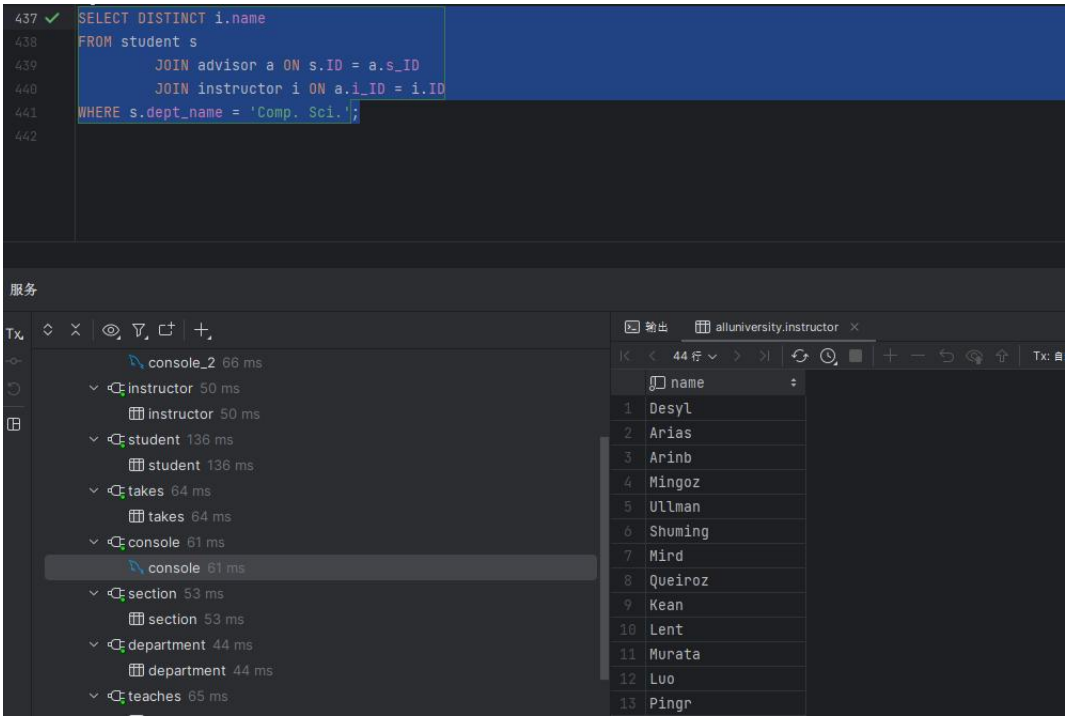
答案

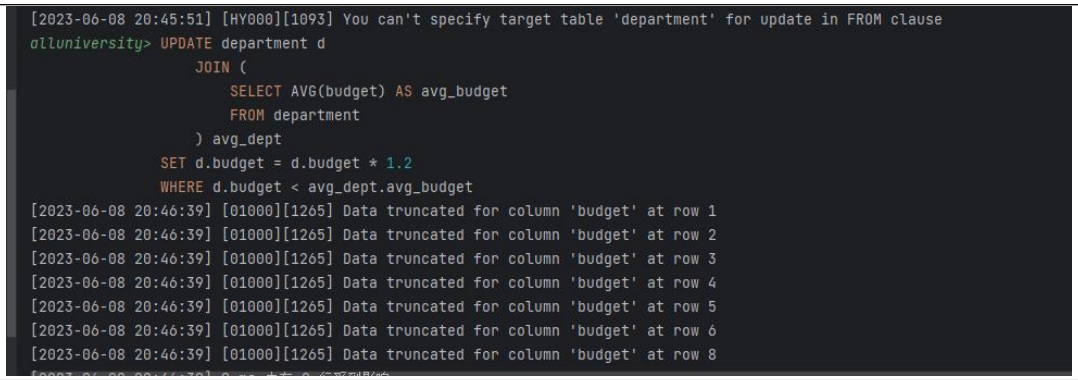
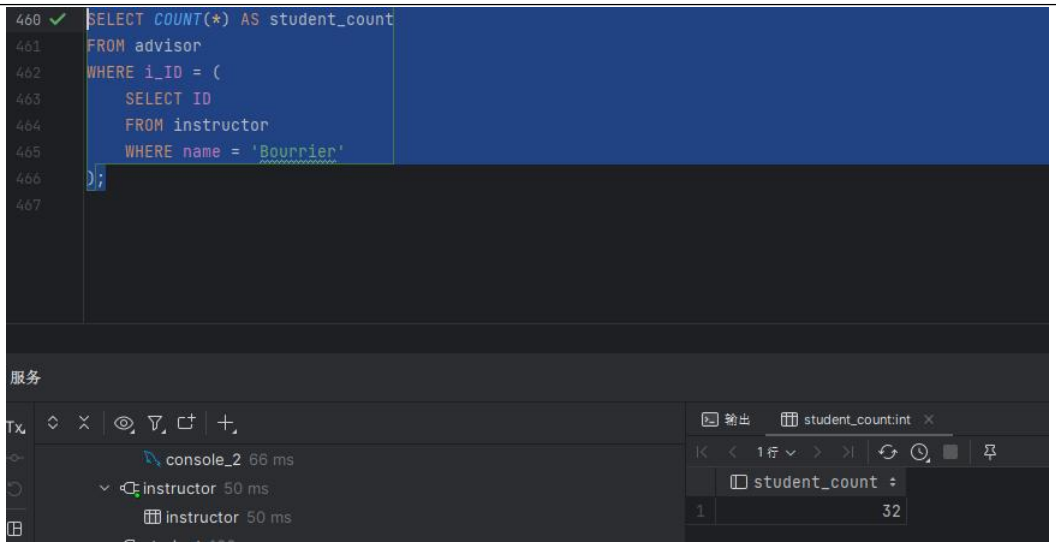
```
SELECT COUNT(*) AS section_count
FROM teaches
WHERE ID = (
    SELECT ID
```

	FROM instructor WHERE name = 'Dale') AND year BETWEEN 2007 AND 2009;
效果截图	
实验题目 29	查询计算机学院老师中比生物学院工资最高的老师工资低，但是比生物学院最低工资高的老师的信息。(生物学院，英语学院)
答案	SELECT * FROM instructor WHERE dept_name = 'Biology' AND salary < (SELECT MAX(salary) FROM instructor WHERE dept_name = 'English') AND salary > (SELECT MIN(salary) FROM instructor WHERE dept_name = 'English'

	);
效果截图	
实验题目 30	请统计 2019 年计算机学院比生物学院多开设了几个课程段。(2009,Civil Eng.,Athletics)
答案	SELECT SUM(CASE WHEN c.dept_name = 'Civil Eng.' THEN 1 ELSE 0 END) AS civil_eng_count, SUM(CASE WHEN c.dept_name = 'Athletics' THEN 1 ELSE 0 END) AS ath_count, SUM(CASE WHEN c.dept_name = 'Civil Eng.' THEN 1 ELSE 0 END) - SUM(CASE WHEN c.dept_name = 'Athletics' THEN 1 ELSE 0 END) AS difference FROM section sec JOIN course c ON sec.course_id = c.course_id WHERE sec.year = 2009;

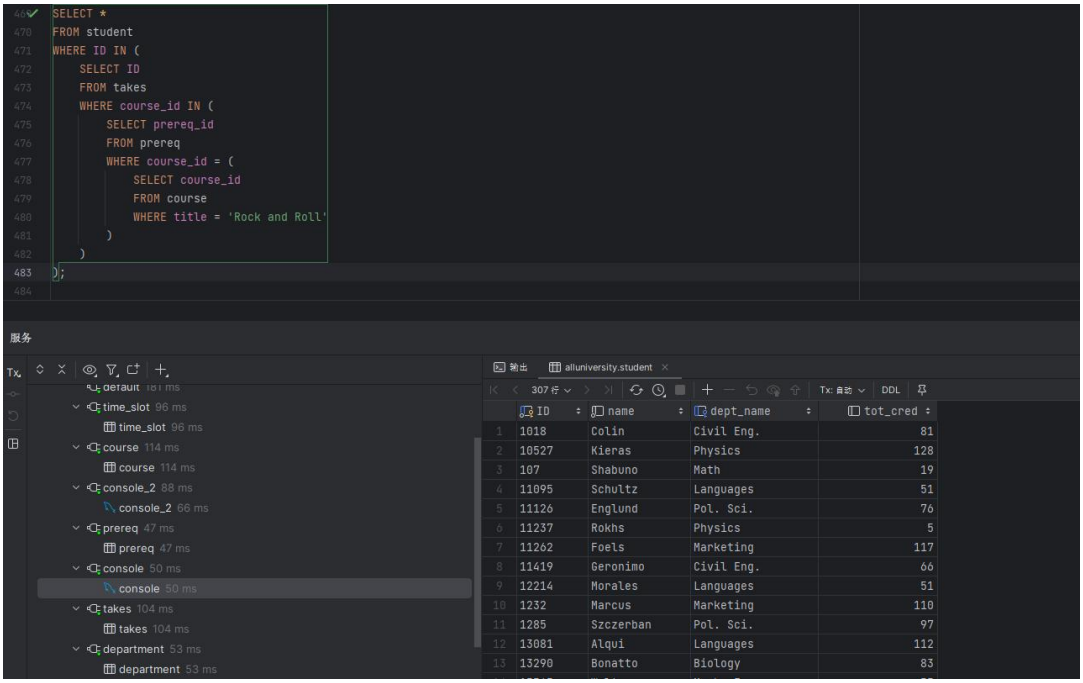
效果截图	
实验题目 31	请统计 2019 年春季，没有学生选课的课程段的数量。若没有，请输出 0。
答案	SELECT COUNT(*) AS section_count FROM section WHERE sec_id NOT IN (SELECT sec_id FROM takes WHERE year = 2009 AND semester = 'Spring');
效果截图	
实验题目 32	请输出计算机学院学生的指导老师的姓名，去掉重复信息。

答案	SELECT DISTINCT i.name FROM student s JOIN advisor a ON s.ID = a.s_ID JOIN instructor i ON a.i_ID = i.ID WHERE s.dept_name = 'Comp. Sci.';
效果截图	
实验题目 33	将低于整个学校的平均预算的各个学院的预算提高到原来的 1.2 倍。
答案	UPDATE department d JOIN (SELECT AVG(budget) AS avg_budget FROM department) avg_dept SET d.budget = d.budget * 1.2 WHERE d.budget < avg_dept.avg_budget;

效果截图	
实验题目 34	统计“李四”老师指导的学生数量。若没有，请输出 0。(Bourrier)
答案	SELECT COUNT(*) AS student_count FROM advisor WHERE i_ID = (SELECT ID FROM instructor WHERE name = 'Bourrier');
效果截图	
实验题目 35	查询已经选了“数据库原理“这门课的所有直接先修课程的学生的信息(Rock and Roll)
答案	SELECT *

```
FROM student
WHERE ID IN (
    SELECT ID
    FROM takes
    WHERE course_id IN (
        SELECT prereq_id
        FROM prereq
        WHERE course_id = (
            SELECT course_id
            FROM course
            WHERE title = 'Rock and Roll'
        )
    )
);
```

效果截图

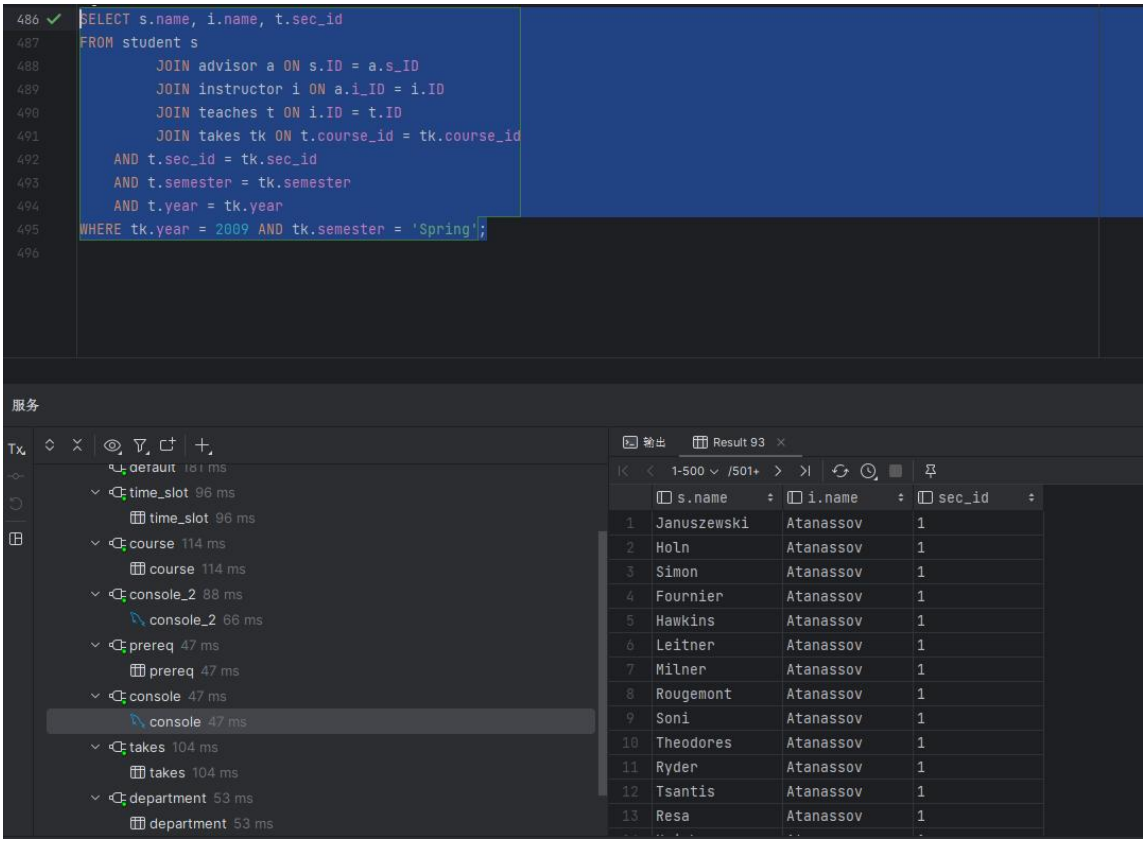


实验题目 36

查询 2019 年春季选了自己指导教师开设的课程段的学生的姓名、指导老师的姓名和课程段的 ID。(2009,)

答案	SELECT s.name, i.name, t.course_id FROM student s JOIN takes t ON s.ID = t.ID JOIN instructor i ON t.ID = i.ID WHERE t.course_id IN (SELECT course_id FROM teaches WHERE ID = (SELECT ID FROM instructor WHERE name = 'Atanassov') AND sec_id = (SELECT sec_id FROM section WHERE semester = 'Spring' AND year = 2009));

效果截图


SELECT s.name, i.name, t.sec_id
FROM student s
JOIN advisor a ON s.ID = a.s_ID
JOIN instructor i ON a.i_ID = i.ID
JOIN teaches t ON i.ID = t.ID
JOIN takes tk ON t.course_id = tk.course_id
AND t.sec_id = tk.sec_id
AND t.semester = tk.semester
AND t.year = tk.year
WHERE tk.year = 2009 AND tk.semester = 'Spring';
服务
time_slot 96 ms
course 114 ms
console_2 88 ms
prereq 47 ms
console 47 ms
takes 104 ms
department 53 ms
Result 93
s.name i.name sec_id
Januszewski Atanassov 1
Holn Atanassov 1
Simon Atanassov 1
Fournier Atanassov 1
Hawkins Atanassov 1
Leitner Atanassov 1
Milner Atanassov 1
Rougemont Atanassov 1
Soni Atanassov 1
Theodores Atanassov 1
Ryder Atanassov 1
Tsantis Atanassov 1
Resa Atanassov 1