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 |  |
| 实验题目3 | 从课程段（section）表中查询课程名称，要求消除值相同的那些行。 |
| 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 |  |
| 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 |  |