实 验 报 告

课程名称： 数据库原理与应用实验

院系名称： 计算机科学与技术学院

专业班级： 计科22-3

学生姓名： 邹晨骏

学 号： 20222163

指导教师： 邓红

**黑龙江工程学院计算机科学与技术学院制**

**实验项目成绩评分表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **实验项目** | **评分标准（100%）** | | | **项目成绩**  **（百分）** | **成绩** |
| **结果准确性**  **60%** | **编码质量**  **20%** | **报告撰写20%** |
| 1 | MySQL工具的使用 |  |  |  |  |  |
| 2 | 数据库的定义 |  |  |  |  |
| 3 | 数据库的创建及维护 |  |  |  |  |
| 4 | 数据库的简单查询 |  |  |  |  |
| 5 | 数据库的复杂查询 |  |  |  |  |
| 6 | 数据库的综合查询 |  |  |  |  |
| 7 | 数据库的视图操作 |  |  |  |  |
| 8 | 存储过程\触发器 |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| 总成绩 | | | | |  | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验项目 | 数据库的复杂查询 | | | 实验日期 | 2024/10/18 |
| 实验地点 |  | 同组人数 | 1 | 实验地点 |  |
| 实验类型 | √验证性 □综合性 □设计性 □研究性 □其 他 | | | | |
| 一、实验目的  1、理解索引的的概念与类型。  2、掌握嵌套查询的方法。 | | | | | |
| 二、实验内容  1、查询“001”课程比"002"课程成绩高的所有学生的学号。  2、查询没学过“赵兰英“老师的课程的同学的学号、姓名。  3、查询所有课程有挂科的同学的学号、姓名。  4、查询至少一门课与学号为0003的同学所学相同的学生的学号和姓名。  5、统计列印各科成绩，各分数段人数：课程ID、课程名称，100-85,85-70,70-60,<60。  6、查询男生、女生的人数。  7、查询姓“王”的师生名单。  8、写出选修了数据结构的同学的学号和姓名。  9、统计每门课的选课人数，包括没有人选的课程，列出课程号及选课情况，其中选课情况为，如果此门课的选课人数超过100-- 人，则显示人多，40-100一般 1-40人正好，无人选。  10、查询计算机有哪些学生没有选课，列出姓名和学号（用外连接）。  11、成绩小于60的学生姓名,课程,成绩。  12、统计每个学生的选课人数和考试总成绩，并按照选课门数升序排列。  13、子查询的使用。  查询与“钟丽丽”在同一个系的学生。  查询选修了003号课程且成绩在85分以上的学生的学号、姓名。  查询所有学生的选课情况。  14、GROUP BY、ORDER BY和LIMIT子句的使用：  查找student中男生和女生的人数；  查找选修了2号课程的学生的学号及其成绩，查询结果按成绩降序排序；  返回student表中的前3位同学的信息。 | | | | | |
| 三、实验步骤及编码  1.  SELECT e1.student\_id  FROM enrollments e1  JOIN enrollments e2 ON e1.student\_id = e2.student\_id  WHERE e1.course\_id = '001' AND e2.course\_id = '002' AND e1.grade > e2.grade;  2.  SELECT s.student\_id, s.name  FROM students s  WHERE s.student\_id NOT IN (  SELECT e.student\_id  FROM enrollments e  JOIN courses c ON e.course\_id = c.course\_id  JOIN teachers t ON c.teacher\_id = t.teacher\_id  WHERE t.name = '赵兰英'  );  3.  SELECT DISTINCT s.student\_id, s.name  FROM students s  JOIN enrollments e ON s.student\_id = e.student\_id  WHERE e.grade < 60;  4.  SELECT DISTINCT s.student\_id, s.name  FROM students s  JOIN enrollments e ON s.student\_id = e.student\_id  WHERE e.course\_id IN (  SELECT course\_id  FROM enrollments  WHERE student\_id = '0002'  ) AND s.student\_id != '0002';  5.  SELECT c.course\_id, c.name,  SUM(CASE WHEN e.grade BETWEEN 85 AND 100 THEN 1 ELSE 0 END) AS '100-85',  SUM(CASE WHEN e.grade BETWEEN 70 AND 84 THEN 1 ELSE 0 END) AS '85-70',  SUM(CASE WHEN e.grade BETWEEN 60 AND 69 THEN 1 ELSE 0 END) AS '70-60',  SUM(CASE WHEN e.grade < 60 THEN 1 ELSE 0 END) AS '<60'  FROM courses c  JOIN enrollments e ON c.course\_id = e.course\_id  GROUP BY c.course\_id, c.name;  6.  SELECT gender, COUNT(\*) AS count  FROM students  GROUP BY gender;  7.  SELECT name FROM students WHERE name LIKE '王%';  UNION  SELECT name FROM teachers WHERE name LIKE '王%';  8.  SELECT s.student\_id, s.name  FROM students s  JOIN enrollments e ON s.student\_id = e.student\_id  JOIN courses c ON e.course\_id = c.course\_id  WHERE c.name = '数据结构';  9.  SELECT c.course\_id, c.name,  CASE  WHEN COUNT(e.student\_id) > 100 THEN '人多'  WHEN COUNT(e.student\_id) BETWEEN 40 AND 100 THEN '一般'  WHEN COUNT(e.student\_id) BETWEEN 1 AND 39 THEN '正好'  ELSE '无人选'  END AS 选课情况  FROM courses c  LEFT JOIN enrollments e ON c.course\_id = e.course\_id  GROUP BY c.course\_id, c.name;  10.  SELECT s.student\_id, s.name  FROM students s  LEFT JOIN enrollments e ON s.student\_id = e.student\_id  WHERE e.student\_id IS NULL AND s.department = '计算机科学与技术';  11.  SELECT s.name, c.name AS course\_name, e.grade  FROM students s  JOIN enrollments e ON s.student\_id = e.student\_id  JOIN courses c ON e.course\_id = c.course\_id  WHERE e.grade < 60;  12.  SELECT s.student\_id, s.name, COUNT(e.course\_id) AS 选课门数, SUM(e.grade) AS 总成绩  FROM students s  JOIN enrollments e ON s.student\_id = e.student\_id  GROUP BY s.student\_id, s.name  ORDER BY COUNT(e.course\_id) ASC;  13.  SELECT s.student\_id, s.name  FROM students s  WHERE s.department = (SELECT department FROM students WHERE name = '王晶');  SELECT s.student\_id, s.name  FROM students s  JOIN enrollments e ON s.student\_id = e.student\_id  WHERE e.course\_id = '003' AND e.grade > 85;  14.  SELECT gender, COUNT(\*) AS count  FROM students  GROUP BY gender;  SELECT s.student\_id, e.grade  FROM students s  JOIN enrollments e ON s.student\_id = e.student\_id  WHERE e.course\_id = '002'  ORDER BY e.grade DESC;  SELECT \* FROM students  ORDER BY student\_id  LIMIT 3;sssss | | | | | |
| 四、实验结果及结论  1.  文本  描述已自动生成  2.  图片包含 文本  描述已自动生成  3.  图示  描述已自动生成  4.  文本  描述已自动生成  5.  图片包含 图形用户界面  描述已自动生成  6.  文本  描述已自动生成  7.  文本  描述已自动生成  图形用户界面, 文本  描述已自动生成  8.  文本  中度可信度描述已自动生成  9.  文本  描述已自动生成  10.  图形用户界面, 文本  描述已自动生成  11.  文本  中度可信度描述已自动生成  12.  图形用户界面  中度可信度描述已自动生成  13.  文本  描述已自动生成  文本  低可信度描述已自动生成  图片包含 文本  描述已自动生成  14  文本  描述已自动生成  图片包含 文本  描述已自动生成  电脑屏幕的照片上有文字  中度可信度描述已自动生成 | | | | | |
| 五、备注 | | | | | |