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 | 存储过程\触发器设计 | | | 实验日期 |  |
| 实验地点 | 线上 | 同组人数 | 1 | 实验台号 |  |
| 实验类型 | √验证性 □综合性 □设计性 □研究性 □其 他 | | | | |
| 一、实验目的  1.理解存储过程的概念。  2.掌握存储过程的使用方法。 | | | | | |
| 二、实验内容  （一）存储过程  1、定义一个存储函数，按性别计算所有学生的平均年龄。  2、编写存储过程，根据给定的学生学号，计算该生的平均成绩  3、创建存储过程，能够查询出指定课程（课程名）的平均成绩，并用OUT参数返回结果。  4、创建存储过程p\_Insertstu，实现向学生数据表中插入一个新同学，并提供相应的实参值（实参值由用户自定）。  （二）触发器  新建一个表：Teacher(Tno name Title money)  1、实现：定义一个before行级触发器，为教师表Teacher定义完整性规则＂教师的工资不得低于4000元，如果低于4000元，则自动的改为4000元＂）  ① 先查看有没有定义这样的触发器  ②定义这样的触发器  ③测试 （ 看Teacher表里面有什么，并且插入，进行测试）  ④删除该触发器 | | | | | |
| 三、实验步骤及编码  1.  DROP FUNCTION IF EXISTS avg\_age\_by\_gender;  DELIMITER //  CREATE FUNCTION avg\_age\_by\_gender(gender CHAR(1))  RETURNS FLOAT  DETERMINISTIC  BEGIN  DECLARE avg\_age FLOAT;  SELECT AVG(age) INTO avg\_age FROM Student WHERE gender = gender;  RETURN avg\_age;  END //  DELIMITER ;  2. （一）  DROP PROCEDURE IF EXISTS avg\_grade\_by\_student;  DELIMITER //  CREATE PROCEDURE avg\_grade\_by\_student(IN student\_id VARCHAR(10), OUT avg\_grade FLOAT)  BEGIN  SELECT AVG(grade) INTO avg\_grade FROM SC WHERE student\_id = student\_id;  END //  DELIMITER ; 3.  DROP PROCEDURE IF EXISTS avg\_grade\_by\_course;  DELIMITER //  CREATE PROCEDURE avg\_grade\_by\_course(IN course\_name VARCHAR(50), OUT avg\_grade FLOAT)  BEGIN  SELECT AVG(grade) INTO avg\_grade  FROM SC  JOIN Course ON SC.course\_id = Course.course\_id  WHERE Course.course\_name = course\_name;  END //  DELIMITER ; 4.  DROP PROCEDURE IF EXISTS p\_Insertstu;  DELIMITER //  CREATE PROCEDURE p\_Insertstu(IN student\_id VARCHAR(10), IN name VARCHAR(50), IN age INT, IN gender CHAR(1))  BEGIN  INSERT INTO Student (student\_id, name, age, gender) VALUES (student\_id, name, age, gender);  END //  DELIMITER ; （二） 1.  CREATE TABLE IF NOT EXISTS Teacher (  Tno VARCHAR(10) PRIMARY KEY,  name VARCHAR(50),  Title VARCHAR(50),  money FLOAT  );  2.  ① SHOW TRIGGERS LIKE 'Teacher';  ②DROP TRIGGER IF EXISTS before\_teacher\_insert;  DELIMITER //  CREATE TRIGGER before\_teacher\_insert  BEFORE INSERT ON Teacher  FOR EACH ROW  BEGIN  IF NEW.money < 4000 THEN  SET NEW.money = 4000;  END IF;  END //  DELIMITER ;  ③SELECT \* FROM Teacher; INSERT INTO Teacher (Tno, name, Title, money) VALUES ('T001', 'John Doe', 'Lecturer', 3500); SELECT \* FROM Teacher;ssssss ④DROP TRIGGER IF EXISTS before\_teacher\_insert; | | | | | |
| 四、实验结果及结论  1.  文本  描述已自动生成  2.  文本  描述已自动生成  3.  文本  描述已自动生成 4.    （二） 1.  文本  描述已自动生成  2.  ①  文本  低可信度描述已自动生成  ② 文本  描述已自动生成  ③ 文本  描述已自动生成 ④ | | | | | |
|  | | | | | |
| 五、备注 | | | | | |