**实验报告**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 学 号 | 17150011001 | | 姓 名 | 陈扬 | 专业班级 | | 17 计算机科学与技术 | |
| 课程名称 | **数据库系统** | | | | 学期 | | **2020年春季学期** | |
| 任课教师 | 黄磊 | | 完成日期 | 4 月 28 日 | | 上机课时间 | | 周三12节（双） |
| 实验名称 | | 使用商用数据库MYSQL,实现对教务系统数据的存储、查询和更新 | | | | | | |
| 一、实验内容和要求：  1. 实验项目名称 使用商用数据库 SQL Server 或 MySQL，实现对教务系统数据的存储、查询 和更新。  2. 实验内容  （1）用 SQL 语句建立学生选课数据库，并对每张表录入实验数据。  （2）结合教材所学的 SQL 语法，进行实际操作，练习各类查询语句，增 加、删除、修改等语句。  （3）学习、掌握直至熟练使用 SQL。  （4）将自己认为重要的 20 个例题写到实验报告里。  （5）将各个 SQL 查询及其结果屏幕截图附在实验报告中。  3. 实验要求  （1）完成实验内容。  （2）记录实验过程和实验结果，撰写实验报告  二、实验过程（实验记录、代码、操作等）和结果：  开发工具：Navicat for MYSQL  开发语言：SQL  开发环境: MACOS  E-R图设计  E-R图设计主要包含概念设计、E-R图、以及将E-R图转化为物理模型  概念设计：  数据库系统中共创建了3个实体 学生,课程,学生选课表  学生的属性:学号、姓名、性别、年龄、系别  课程的属性: 选课号,课程名,先修课,学分  学生选课表:学生号,选课号,学分  设计教务系统,创建三个基本表,分别是 student 表,course 表和 SC 表.   1. 设计教务系统,创建三个基本表,分别是 student 表,course 表和 SC 表.    1. 创建 student 表:  * （1）在列级定义主码 * CREATE TABLE Student(Sno CHAR(20) PRIMARY KEY,  Sname CHAR(20) NOT NULL,  Ssex CHAR(2),  Sage SMALLINT,  Sdept CHAR(20) ); * 其中 SNO 学号为主码,SNAME非 NULL 满足实体完整性.   1. 建立 Course 表 * CREATE TABLE Course  (Cno CHAR(4) PRIMARY KEY, Cname CHAR(40), Cpno CHAR(4), Ccredit SMALLINT, FOREIGN KEY (Cpno) REFERENCES Course(Cno) ); * Cpno是外码,被参照表是Course,被参照列是Cno * Cpno是表示先修课   1. 创建 SC 表: * CREATE TABLE SC ( Sno CHAR(20),  Cno CHAR(4),  Grade SMALLINT,  PRIMARY KEY (Sno,Cno),  FOREIGN KEY (Sno) REFERENCES Student(Sno),  FOREIGN KEY (Cno)REFERENCES Course(Cno) ); * /\* 主码由两个属性构成，必须作为表级完整性进行定义\*/ * /\* 表级完整性约束条件，Sno是外码，被参照表是Student \*/ * /\* 表级完整性约束条件， Cno是外码，被参照表是Course \*/  1. 建立专业表DEPT，要求专业名称Dname列取值唯一且不能取空值，专业编号Deptno列为主码。  * CREATE TABLE DEPT ( Deptno NUMERIC(2),  Dname CHAR(9) UNIQUE NOT NULL,/\*要求Dname列值唯一, 并且不能取空值\*/  PRIMARY KEY (Deptno) );  1. insert student信息  * INSERT INTO student(Sno,sname,ssex,sage,sdept) VALUES(17150011001,'Marcus','MA',21,'CS'); INSERT INTO student(Sno,sname,ssex,sage,sdept) VALUES(17150011002,'Mike','MA',21,'CS'); INSERT INTO student(Sno,sname,ssex,sage,sdept) VALUES(17150011003,'JIX','FA',19,'EE'); INSERT INTO student(Sno,sname,ssex,sage,sdept) VALUES(17150011004,'John','FA',22,'CS'); * insert course信息 * INSERT INTO course(Cno,Cname,Cpno,Ccredit) VALUES(2,'数学',null,2); INSERT INTO course(Cno,Cname,Cpno,Ccredit) VALUES(6,'数据处理',null,2); INSERT INTO course(Cno,Cname,Cpno,Ccredit) VALUES(7,'PASCAL',6,4); INSERT INTO course(Cno,Cname,Cpno,Ccredit) VALUES(5,'数据结构',7,3); INSERT INTO course(Cno,Cname,Cpno,Ccredit) VALUES(1,'数据库',5,4); INSERT INTO course(Cno,Cname,Cpno,Ccredit) VALUES(3,'信息系统',1,4); INSERT INTO course(Cno,Cname,Cpno,Ccredit) VALUES(4,'操作系统',6,3); * INSERT SC 信息 * INSERT INTO SC(Sno,Cno,Grade) VALUES(17150011001, 1 , 92); INSERT INTO SC(Sno,Cno,Grade) VALUES(17150011001, 2 , 85); INSERT INTO SC(Sno,Cno,Grade) VALUES(17150011001, 3 , 88); INSERT INTO SC(Sno,Cno,Grade) VALUES(17150011002, 1 , 90); INSERT INTO SC(Sno,Cno,Grade) VALUES(17150011002, 2 , 90); INSERT INTO SC(Sno,Cno,Grade) VALUES(17150011002, 3 , 80);  1. 查询全体学生的详细记录  * SELECT Sno,Sname,Ssex,Sage,Sdept FROM Student;  1. 查全体学生的姓名及其出生年份  * SELECT Sname,2020-Sage FROM Student;  1. 查询计算机科学系全体学生的信息。  * SELECT \* FROM Student WHERE Sdept=‘CS’;  1. 查询年龄在20~23岁（包括20岁和23岁）之间的学生 的姓名、系别和年龄  * SELECT Sname, Sdept, Sage FROM Student WHERE Sage BETWEEN 20 AND 23;  1. 查询计算机科学系（CS）、数学系（MA）和信息 系（IS）学生的姓名和性别。  * SELECT Sname, Ssex FROM Student WHERE Sdept IN ('CS','MA','IS' );  1. 查询所有 M 开头学生的姓名、学号和性别。  * SELECT Sname, Sno, Ssex FROM Student WHERE Sname LIKE 'M%';  1. 查询计算机系年龄在20岁以上的学生姓名。  * SELECT Sname FROM Student WHERE Sdept= 'CS' AND Sage<20;  1. 查询全体学生情况，查询结果按所在系的系号升序 排列，同一系中的学生按年龄降序排列。  * SELECT \* FROM Student ORDER BY Sdept, Sage DESC;  1. 查询学生总人数。  * SELECT COUNT(\*) FROM Student;  1. ] 查询学生17150011001选修课程的总学分数。  * SELECT SUM(Ccredit) FROM SC,Course WHERE Sno='201215012' AND SC.Cno=Course.Cno;  1. 求各个课程号及相应的选课人数。  * SELECT Cno,COUNT(Sno) FROM SC GROUP BY Cno;  1. 查询选修了3门及以上课程的学生学号。  * SELECT Sno FROM SC GROUP BY Sno HAVING COUNT(Cno) =>3;  1. 查询每个学生及其选修课程的情况  * SELECT Student.\*, SC.\* FROM Student, SC WHERE Student.Sno = SC.Sno;  1. ]查询每一门课的直接先修课的名称  * SELECT FIRST.Cname , SECOND.Cname FROM Course FIRST, Course SECOND WHERE FIRST.Cpno = SECOND.Cno;  1. 查询每个学生及其选修课程的情况  * SELECT Student.Sno, Sname, Cname, Grade FROM Student,SC,Course  WHERE Student.Sno = SC.Sno AND SC.Cno = Course.Cno;  1. 查询与“Marcus”在同一个系学习的学生。  * 自身连接法: * SELECT S1.Sno, S1.Sname,S1.Sdept FROM Student S1,Student S2 WHERE S1.Sdept = S2.Sdept AND S2.Sname = 'Marcus'; * 查询选修了课程名为“信息系统”的学生学号和姓名 * SELECT Student.Sno,Sname FROM Student,SC,Course WHERE Student.Sno = SC.Sno AND  SC.Cno = Course.Cno AND  Course.Cname='信息系统';  1. 查询非计算机科学系中比计算机科学系任意一个 学生年龄小的学生姓名和年龄  * SELECT Sname,Sage FROM Student WHERE Sage <(SELECT MAX(Sage)  FROM Student  WHERE Sdept= 'CS')  AND Sdept <> 'CS ';  1. 询所有选修了1号课程的学生姓名。  * SELECT Sname FROM Student WHERE EXISTS(SELECT \*  FROM SC  WHERE Sno=Student.Sno AND Cno= ' 1 ');  1. **查询选修了全部课程的学生姓名。**  * SELECT Sname FROM Student WHERE NOT EXISTS(SELECT \*  FROM Course  WHERE NOT EXISTS(SELECT \*  FROM SC  WHERE Sno= Student.Sno  AND Cno= Course.Cno)  ); * 没有学生选了全部课  1. 查询选修了课程1或者选修了课程2的学生学号。  * SELECT Sno FROM SC WHERE Cno='1'  UNION  SELECT Sno FROM SC WHERE Cno= '2';  1. 查询计算机科学系的学生与年龄不大于19岁的学生 的交集。  * SELECT \* FROM Student WHERE Sdept='CS'  UNION ALL  SELECT \* FROM Student WHERE Sage<=19 * 和 Mooc 上的INTERSECT不同,mysql 用的是 UNION ALL  1. 查询计算机科学系的学生与年龄不大于19岁的学生的差集  * SELECT \* FROM SC WHERE Sdept='CS' EXCEPT SELECT \* FROM SC WHERE Sage<=19;  1. 建立CS系学生的视图。  * CREATE VIEW IS\_Student AS SELECT Sno,Sname,Sage FROM Student WHERE Sdept= 'CS';  1. 定义一个反映学生出生年份的视图。  * CREATE VIEW BT\_S(Sno,Sname,Sbirth) AS SELECT Sno,Sname,2020-Sage FROM Student;  1. 将学生的学号及平均成绩定义为一个视图  * [聚集函数] * CREATE VIEW S\_G(Sno,Gavg) AS SELECT Sno,AVG(Grade) FROM SC GROUP BY Sno;  1. 删除视图BT\_*S和IS\_*S1  * DROP VIEW BT\_S;/\*成功执行\*/  DROP VIEW IS\_S1;/\*拒绝执行\*/ DROP VIEW IS\_S1 CASCADE;/\*成功执行\*/  1. 在CS系学生的视图中找出年龄小于20岁的学生。  * SELECT Sno,Sage FROM IS\_Student WHERE Sage<20; * fig: 我们可以看到 CS 没有学生年级小于 20.  1. 将信息系学生视图IS\_Student中学号”17150011001”的学生姓名改为”老马”。  * UPDATE IS\_Student SET Sname= '老马' WHERE Sno= '17150011001';   我们可以看到马卡斯的名字改成了老马    三、心得总结（写出自己在完成实验过程中遇到的问题、解决方法，以及体会、收获等）：  我实际上在 MYSQL 上做了不止 20 个最经典的实验,实际上还有很多经典的例子,我基本上把 MOOC 上数据库基础的所有例子都实现了一篇,正如老师所说的,数据库是一门实践性很强的课程,要多通过课下练习才能更好地掌握这门课, | | | | | | | | |