**实验报告**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 学 号 | 19020011038 | | 姓 名 | 岳宇轩 | 专业班级 | | 19级计算机科学与技术慧与班 | |
| 课程名称 | **《数据库系统》** | | | | 学期 | | **2022年春季学期** | |
| 任课教师 | 刘洁 | | 完成日期 | 2022/3/14 | | 实验课时间 | | 2022/3/14 |
| 实验名称 | | 实验四：通过实验体会关系数据理论,函数依赖以及范式 | | | | | | |
| **一、实验要求（10%）**  实验要求  (1)  (a) 设计一个关系，使之满足1NF而不满足2NF。  (b) 设计一个关系，使之满足2NF而不满足3NF。  (c) 设计一个关系，使之满足3NF而不满足BCNF。  (2) 详细描述关系的语义，分析关系中存在的函数依赖。  (3) 使用商用数据库SQL Server设计实现（或其他DBMS），录入数据。  (4) 设计实验，体会数据冗余、增加异常、删除异常、修改复杂。  (5) 按照实验报告模板格式撰写实验报告，提交电子版实验报告。   1. **实验内容及步骤（80%）** 2. （a）设计一个关系，使之满足1NF而不满足2NF 3. 详细描述关系的语义，分析关系中存在的函数依赖   **C:/Users/yyx/AppData/Local/Temp/wps.ItxPfFwps**  上述关系模式属于1NF而不属于2NF，因为存在非主属性对码（Sno,Cno）的部分函数依赖：Sno→Sname，Cno→Cname。  在关系模式SC中，主码是（Sno,Cno）。所有能称得上是关系模式的一定是满足1NF的，因此首先SC∈1NF。  其次，对于非主属性学生姓名Sname来说，只需要主码中的一部分属性：学生学号Sno即可确定；对于非主属性课程名称Cname来说，只需要主码中的一部分属性：课程代号Cno即可确定。因此，存在非主属性对码的部分函数依赖，SC∉2NF。  sql语句如下：  CREATE TABLE SC(  Sno CHAR(9),  Cno CHAR(9),  Grade SMALLINT,  PRIMARY KEY (Sno,Cno),  Sname CHAR(20),  Cname CHAR(20)  );   1. （b）设计一个关系，使之满足2NF而不满足3NF 2. 详细描述关系的语义，分析关系中存在的函数依赖   SC(Sno,Cno,score,Sname,Cname)    设计关系模式Student(Sno,Sdept,Sloc).在该关系模式中，Sno的含义是学生学号，Sdept的含义是学生所属的院系，Sloc的含义是学生所属院系的教学楼位置。  根据上图可以看出，Sno是主码。由于主码的属性只有一个，因此是不会存在非主属性对码的部分函数依赖的，该关系模式属于2NF，  但该关系模式不属于3NF，因为存在非主属性对码的传递函数依赖：Sno→Sdept，Sdept→Sloc，也就是通过学号可以确定院系，通过院系可以确定位置。  sql语句如下：  CREATE TABLE Student(  Sno CHAR(9) PRIMARY KEY,  Sdept CHAR(20),  Sloc CHAR(30)  );   1. （c）设计一个关系，使之满足3NF而不满足BCNF 2. 详细描述关系的语义，分析关系中存在的函数依赖   在关系模式TCB中，T表示教师，C表示课程，B表示教材。  -函数依赖：（T，C）→B，（T，B）→C，C→B  -候选码：（T，C），（T，B）  -主属性：T，C，B  ※所有属性均是主属性，没有非主属性，故TCB∈3NF  ※TCB∉BCNF，因为C是决定因子，C不包含码  **sql语句如下：**  CREATE TABLE TCB(  Tno CHAR(9),  Cno CHAR(9),  Bno CHAR(9),  PRIMARY KEY (Tno,Cno)  );  3.使用商用数据库SQL Server设计实现（或其他DBMS），录入数据。  4.设计实验，体会数据冗余、增加异常、删除异常、修改复杂。   1. **数据冗余：**   向关系模式Student中录入数据：  INSERT INTO Student VALUES  (000000001,'计算机系','信息楼南楼'),  (000000002,'计算机系','信息楼南楼'),  (000000003,'计算机系','信息楼南楼'),  (000000004,'计算机系','信息楼南楼'),  (000000005,'计算机系','信息楼南楼'),  (000000006,'计算机系','信息楼南楼'),  (000000007,'计算机系','信息楼南楼'),  (000000008,'计算机系','信息楼南楼'),  (000000009,'计算机系','信息楼南楼'),  (000000010,'计算机系','信息楼南楼');  结果：    从实验结果中可以看到，由于存在非主属性对码的传递函数依赖，造成了数据冗余。  当我录入计算机系学生的信息时，会录入很多重复的Sloc信息。   1. **增加异常**   对于关系模式SC：  假设要开设一门新的课程，暂时还没有人选修。这样，由于还没有"学号"关键字，课程名称和学分也无法记录入数据库。  执行如下sql插入语句：  INSERT INTO SC (Cno,Cname) VALUES (123456789,'数据库系统');  结果显示：     1. **删除异常：**   对于关系模式Student来说，如果删除了一个院校的所有学生，那么该院校也将被删除掉。  首先进行数据插入：  INSERT INTO Student VALUES  (000000001,'计算机系','信息楼南楼'),  (000000002,'计算机系','信息楼南楼'),  (000000003,'计算机系','信息楼南楼'),  (000000004,'计算机系','信息楼南楼'),  (000000005,'计算机系','信息楼南楼'),  (000000006,'计算机系','信息楼南楼'),  (000000007,'计算机系','信息楼南楼'),  (000000008,'计算机系','信息楼南楼'),  (000000009,'计算机系','信息楼南楼'),  (100000001,'物理系','信息楼北楼'),  (100000002,'物理系','信息楼北楼'),  (100000003,'物理系','信息楼北楼'),  (100000004,'物理系','信息楼北楼'),  (100000005,'物理系','信息楼北楼'),  (100000006,'物理系','信息楼北楼');  执行如下sql语句，删除所有计算机系学生：  DELETE FROM Student WHERE Sdept = '计算机系';  然后查看结果：  SELECT \* FROM Student;  结果：    从上述过程可以看出，虽然我只是进行了删除计算机系学生的操作，但是计算机系的信息（Sloc）也丢失了。   1. **修改复杂：**   对于关系模式Student来说，若更改了某个院校的地点，数据表中院校所属行的地点都要更新，否则就会出现不一致的情况。  首先对关系模式Student插入数据：  INSERT INTO Student VALUES  (000000001,'计算机系','信息楼南楼'),  (000000002,'计算机系','信息楼南楼'),  (000000003,'计算机系','信息楼南楼'),  (000000004,'计算机系','信息楼南楼'),  (000000005,'计算机系','信息楼南楼'),  (000000006,'计算机系','信息楼南楼'),  (000000007,'计算机系','信息楼南楼'),  (000000008,'计算机系','信息楼南楼'),  (000000009,'计算机系','信息楼南楼');  展示结果：    假设计算机系的位置发生改变，变化到了西海岸校区，那么就要对所有的条目进行更新。  UPDATE Student SET Sloc = '西海岸校区' WHERE Sdept = '计算机系';  SELECT \* FROM Student;  结果：    可以看到，受影响的行数很多。假设一个院系有好几千名学生，那么就要更新好几千行的数据，这无疑是不合理的、费事的，并且在更新的过程中容易出错，是非常复杂的修改过程。   1. **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）（10%）**   通过本次实验目的一二，我对于各种范式以及各种依赖关系有了更加清楚的认知。1NF表示的是没有重复列，一般所有的关系模式都是属于1NF的；2NF表示的关系模式消除了非主属性对码的部分函数依赖。在这里我温习了三个概念，主属性非主属性、码、部分函数依赖。主属性是候选码的属性，码是被选做主属性的候选码，部分函数依赖是指，决定因子中的部分属性同样可以确定属性值；3NF消除了非主属性对码的传递函数依赖，所谓传递函数依赖是指A→B,B→C，那么责成A→→C；BCNF是指关系模式中所有的决定因子都包含码，所谓决定因子是指依赖关系中的箭头左边的部分。  通过本次实验目的三四，我重新熟悉了sql语句的编写，对于数据冗余、增加异常、删除异常、修改复杂这四个错误有了更加清楚的认知。在实验一开始，我对于sql语句的使用，尤其是关系表的创建和数据的增删改查感到有些陌生，通过复习前面的实验知识以及编写本次实验的过程，我对于sql的使用更加的熟练了。在进行录入数据的时候，我深感由于不合理的关系模式设计导致的数据冗余是有多么麻烦。虽然上课的时候听得有些模模糊糊，但在亲身实践的过程中，我对于数据冗余、增加异常、删除异常、修改复杂这几个问题也有了更加清楚的认知。数据冗余是指由于不合理的关系模式设计导致有许多重复数据；增加异常是指想要插入数据,结构因为表设计的问题，导致不能成功插入；删除异常是指只想删除其中的某些数据 ，结果把不该删的也删了；修改复杂是指想更新一条数据,结果工作量大，还容易出错。  本次实验结束之后，我认为在进行数据库关系结构设计时，应当充分明确其之间的依赖关系，设计符合BCNF关系的模型，避免数据冗余、增加异常、删除异常、修改复杂。 | | | | | | | | |
|  | | | | | | | | |