数据库原理实践（2022年秋季学期）

序号:31

实 验 报 告

系别：计算机系
班级：计**20-**实验班
姓名:苏靖博
学号:**20105050110**

成绩：
评语：

指导教师签字：　　日期：

《数据库原理》实验报告

实验名称	数据库的建立及数据维护	实验序号	**1**
实验日期	Mon. Oct. 31 2022	实验人	苏靖博

一、实验目的

1.学会使用对象资源管理器创建数据库、创建基本表和查看数据库属性。

2. 学会使用对象资源管理器向数据库输入数据，修改数据，删除数据的操作。

3. 在SQL Server查询编辑器中完成复杂查询及视图定义。

二、实验任务

1、根据以上数据字典, 画出该数据库的ER图，完成数据库的概念结构设计。

2、将ER图转换成逻辑关系模式，判断逻辑数据库模式中的各个关系（表）是第几范式，如果没有达到第三范式或BC范式, 请进行规范化。完成数据库的逻辑结构设计。

3、通过对象资源管理器实现关系模式的存储，包括确定主码、外部码等。

4、按照给出的示例数据完成数据的录入。

5. 打开数据库SQL Server的查询编辑器，用SQL语句完成要求的查询。

（1） 求全体学生的学号、姓名和出生年份。

（2） 求每个系的学生总人数并按降序排列。

（3） 求选修了课程号为002或003的学生的学号、课程名和成绩。

（4） 检索选修某课程的学生人数多于3人的教师姓名。

（5） 查询所有未选课程的学生姓名和所在系。

（6） 求每个同学的课程成绩的最高分，查询结果项包括：学生姓名、课程号及最高分。

（7） 求所有讲授数据结构课程的教师姓名

（8） 查询所有选修了李正科老师的课程的学生信息

（9） 新建一个关系表S1(SNO,SNAME,SD,SA), 其字段类型定义与Student表中的相应字段(SNO,SNAME,SDEPT,SAGE)的数据类型定义相同。将表Student中在计算机系‘CS’的学生数据插入到表S1中。

（10） 建立计算机系的学生的视图STUDENT_CS。利用视图STUDENT_CS，将学生的001号课程的成绩提高5分。

（11） 创建一个带参数的存储过程，将数据结构课程的成绩在Low与Hign分数段的学

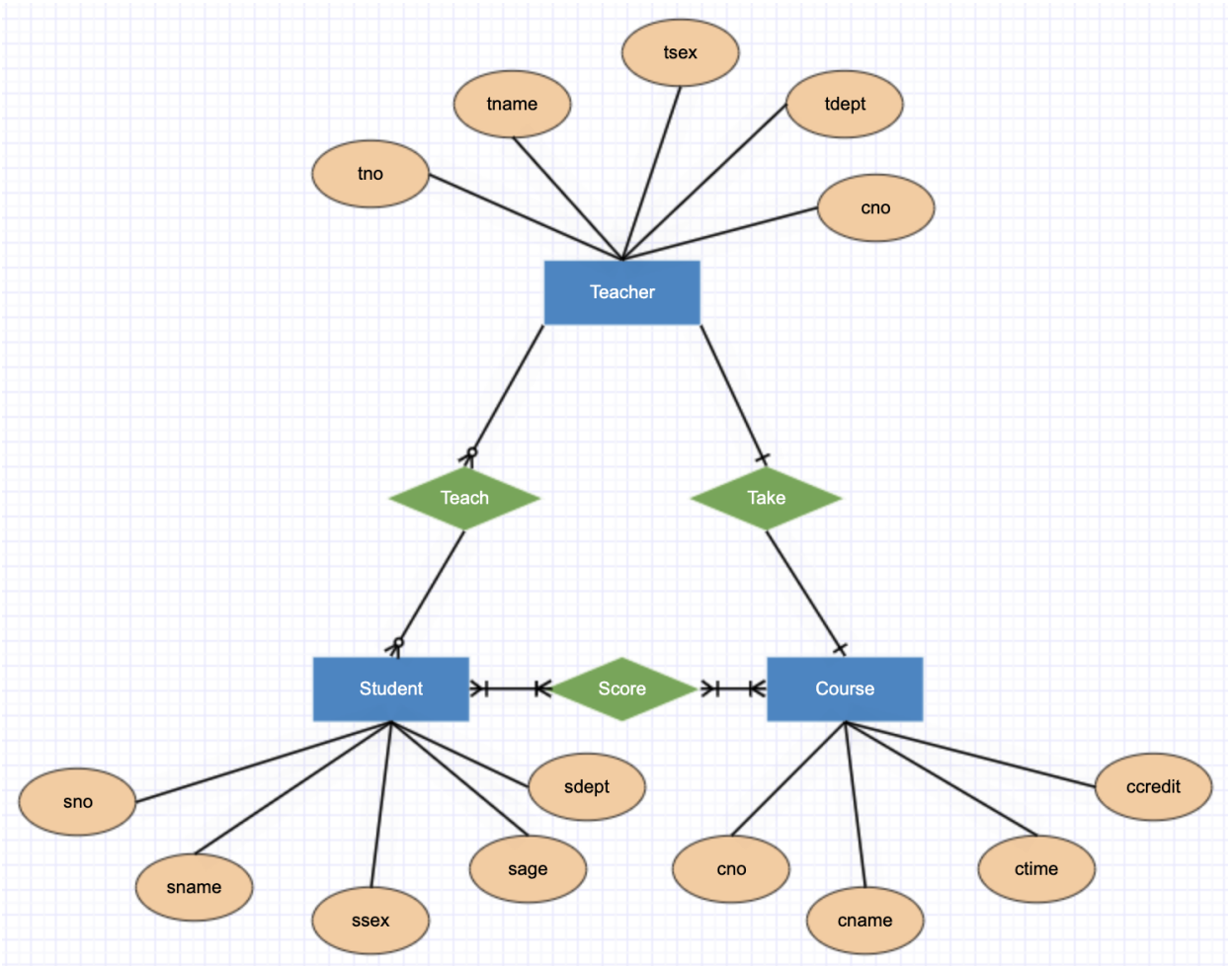
生信息提取出来，查询后的结果包括：姓名、所在的系、成绩。

6. 并通过实验结果验证查询语句的正确性，

7. 将每个SQL语句及结果存盘，以备老师检查。

3、 数据库的**E-R**图

根据以上数据字典，画出该数据库的E-R图，完成数据库的概念结构设计。ER图如下所示：



4、 数据库关系模型的设计与截图

将E-R图转换成关系模型，确定主码、外部码等，填写在下面，并给出SQL Server中表的设计的截图。判断关系模型中的各个关系（表）是第几范式，如果没有达到第三范式或BC范式，请进行规范化。完成数据库的逻辑结构设计。

(1) Student Table

a. Relational model

Student (sno, sname, ssex, sage, sdept)

b. Shortcut

```
CREATE TABLE `Student` (
    sno char(20) NOT NULL,
    sname char(20) NOT NULL,
    ssex char(10) NOT NULL,
    sage int NOT NULL,
    sdept char(30) NOT NULL,
    PRIMARY KEY (sno)
);
SHOW COLUMNS FROM Student;
```

Field	Type	Null	Key	Default	Extra
sno	char(20)	NO	PRI	NULL	
sname	char(20)	NO		NULL	
ssex	char(10)	NO		NULL	
sage	int	NO		NULL	
sdept	char(30)	NO		NULL	

c. Normalization

- **Primary key: sno**
- **Not-null field: sname, ssex, sage, sdept**

(2) Course Table

a. Relational model

Course (cno, cname, ctime, ccredit)

b. Shortcut

```
CREATE TABLE `Course` (
    cno char(20) NOT NULL,
    cname char(20) NOT NULL,
    ctime int NOT NULL,
    ccredit int NOT NULL,
    PRIMARY KEY (cno)
);
SHOW COLUMNS FROM Course;
```

Field	Type	Null	Key	Default	Extra
cno	char(20)	NO	PRI	NULL	
cname	char(20)	NO		NULL	
ctime	int	NO		NULL	
ccredit	int	NO		NULL	

c. Normalization

- **Primary key: cno**
- **Not-null field: cname, ctime, ccredit**

(3) Teacher Table

a. Relational model

Teacher (tno, tname, tsex, tdept, cno)

b. Shortcut

```
CREATE TABLE `Teacher` (
    tno char(20) NOT NULL,
    tname char(20) NOT NULL,
    tsex char(10) NOT NULL,
    tdept char(30) NOT NULL,
    cno char(20) NOT NULL,
    PRIMARY KEY (tno)
);
SHOW COLUMNS FROM Teacher;
```

Field	Type	Null	Key	Default	Extra
tno	char(20)	NO	PRI	NULL	
tname	char(20)	NO		NULL	
tsex	char(10)	NO		NULL	
tdept	char(30)	NO		NULL	
cno	char(20)	NO		NULL	

c. Normalization

- **Primary key: tno**
- **Not-null field: tname, tsex, tdept, cno**

(4) Score Table

a. Relational model

Score (sno, cno, grade)

b. Shortcut

```
CREATE TABLE `Score` (
    sno char(20) NOT NULL,
    cno char(20) NOT NULL,
    grade int NOT NULL,
    PRIMARY KEY (sno, cno)
);
SHOW COLUMNS FROM Score;
```

Field	Type	Null	Key	Default	Extra
sno	char(20)	NO	PRI	NULL	
cno	char(20)	NO	PRI	NULL	
grade	int	NO		NULL	

c. Normalization

- **Primary key: <sno, cno>**
- **Not-null field: grade**

5、 各表录入数据的截图
通过对象资源管理器实现关系模式的存储，完成数据的录入，并截图。

Code:

Shortcut:

1. **Student Table**

sno	sname	ssex	sage	sdept
001	Jingbo Su	Male	20	Computer Science
002	Harry Potter	Male	22	Gryffindor
003	Ron Weasley	Male	22	Gryffindor
004	Hermione Granger	Female	21	Gryffindor
005	Draco Malfoy	Male	22	Slughorn
006	Tom Riddle	Male	96	Slughorn
NULL	NULL	NULL	NULL	NULL

2. **Course Table**

cno	cname	ctime	ccredit
050	Computer Intro.	32	3
193	IOS Development	48	2
221	Machine Learning	48	3
61b	Advanced Programming	48	4
75p	Data Base	64	6
828	Operation System	64	6
NULL	NULL	NULL	NULL

3. **Teacher Table**

tno	tname	tsex	tdept	cno
910	Albus Dumbledore	Male	Computer Science	050
911	Minerva McGonagall	Female	Gryffindor	61b
912	Remus Lupin	Male	Gryffindor	193
913	Rubeus Hagrid	Male	Gryffindor	221
914	Severus Snape	Male	Slughorn	828
915	Olivia Green	Female	Slughorn	75p
NULL	NULL	NULL	NULL	NULL

4. **Score Table**

sno	cno	grade
001	050	100
001	193	92
001	221	98
001	61b	96
002	050	90
002	193	88
002	221	90
002	61b	96
002	75p	80
003	050	80
003	193	82
003	221	78
003	61b	86
003	75p	90
004	050	100
004	193	98
004	221	98
004	61b	96
004	75p	99
004	828	95
005	050	90
005	193	82
005	221	82
005	828	83
006	050	99
006	193	99
006	221	99
006	61b	90
006	75p	99
006	828	90
NULL	NULL	NULL

6、 **SQL**语句及其查询编辑器与执行结果截图

（1） 求全体学生的学号、姓名和出生年份。

Code:

```
USE NCUT;
SELECT Student.sno, Student.sname, 2022 - Student.sage
FROM Student
```

Shortcut:

sno	sname	2022 - Student.
001	Jingbo Su	2002
002	Harry Potter	2000
003	Ron Weasley	2000
004	Hermione Granger	2001
005	Draco Malfoy	2000
006	Tom Riddle	1926

（2） 求每个系的学生总人数并按降序排列。

Code:

```
USE NCUT;
SELECT sdept, COUNT(sno)
FROM Student
GROUP BY sdept
ORDER BY COUNT(sno) DESC
```

Shortcut:

sdept	COUNT(sno)
Gryffindor	3
Slughorn	2
Computer Science	1

（3） 求选修了课程号为`**61b**`或`**828**`的学生的学号、课程名和成绩。

Code:

```
USE NCUT;
SELECT Student.sno, Course.cname, Score.grade
FROM Student, Course, Score
```

```
WHERE (Course.cno = '61b' OR Course.cno = '828')
      AND (Course.cno = Score.cno)
      AND (Student.sno = Score.sno)
```

Shortcut:

sno	cname	grade
001	Advanced Programming	96
002	Advanced Programming	96
003	Advanced Programming	86
004	Advanced Programming	96
004	Operation System	95
005	Operation System	83
006	Advanced Programming	90
006	Operation System	90

(4) 检索选修某课程的学生人数多于**3**人的教师姓名。

Code:

```
USE NCUT;
SELECT Teacher.tname
FROM Teacher, Score
WHERE (Teacher.cno = Score.cno)
GROUP BY Teacher.tname
HAVING COUNT(Score.sno) > 3
```

Shortcut:

tname
Albus Dumbledore
Remus Lupin
Rubeus Hagrid
Minerva McGonagall
Olivia Green

(5) 查询所有未选课程的学生姓名和所在系。

Code:

```
USE NCUT;
SELECT Student.sname, Student.sdept
FROM Student
WHERE Student.sno NOT IN (
     SELECT Student.sno
    FROM Student, Score
    WHERE Student.sno = Score.sno
    GROUP BY Score.cno
```

Shortcut:

sname	sdept

(6) 求每个同学的课程成绩的最高分，查询结果项包括：学生姓名、课程号及最高分。

Code:

```
USE NCUT;
SELECT Student.sname, Score.cno, Score.grade
FROM Student, Score
WHERE (Student.sno = Score.sno)
     AND Score.grade IN (
          SELECT MAX(Score.grade)
        FROM Score
        WHERE Student.sno = Score.sno
        GROUP BY Score.sno
     )
```

Shortcut:

sname	cno	grade
Jingbo Su	050	100
Harry Potter	61b	96
Ron Weasley	75p	90
Hermione Granger	050	100
Draco Malfoy	050	90
Tom Riddle	050	99
Tom Riddle	193	99
Tom Riddle	221	99
Tom Riddle	75p	99

(7) 求所有讲授数据库课程的教师姓名。

Code:

```
USE NCUT;
SELECT Teacher.tname
FROM Teacher, Course
WHERE (Teacher.cno = Course.cno)
     AND (Course.cname = 'Data Base')
```

Shortcut:

tname
Olivia Green

(8) 查询所有选修了 **Severus Snape** 老师的课程的学生信息。

Code:

```
USE NCUT;
SELECT Student.sno, Student.sname, Student.ssex, Student.sage,
Student.sdept
FROM Student, Teacher, Score
WHERE (Teacher.tname = 'Severus Snape')
     AND (Teacher.cno = Score.cno)
     AND (Student.sno = Score.sno)
```

Shortcut:

sno	sname	ssex	sage	sdept
004	Hermione Granger	Female	21	Gryffindor
005	Draco Malfoy	Male	22	Slughorn
006	Tom Riddle	Male	96	Slughorn

(9) 新建一个关系表 **S1 (SNO, SNAME, SD, SA)**, 其字段类型定义与 **Student** 表中的相应字段 **(SNO, SNAME, SDEPT, SAGE)** 的数据类型定义相同。将表 **Student** 中在计算机系 **'CS'** 的学生数据插入到表**S1**中。

Code:

```
USE NCUT;
CREATE TABLE `S1` (
     sno char(20) NOT NULL,
    sname char(20) NOT NULL,
    sd char(30) NOT NULL,
    sa int NOT NULL,
    PRIMARY KEY (sno)
);
INSERT INTO `S1`
SELECT Student.sno, Student.sname, Student.sdept, Student.sage
FROM Student
WHERE Student.sdept = 'Computer Science';
SELECT * FROM S1;
```

Shortcut:

sno	sname	sd	sa
001	Jingbo Su	Computer Science	20
NULL	NULL	NULL	NULL

(10) 建立计算机系的学生的视图**STUDENT_CS**。利用视图**STUDENT_CS**，将学生的**193**号课程的成绩提高**5**分。**(if score > 95, then score = 100 finally…)**

Code:

```
USE NCUT;
SET SQL_SAFE_UPDATES = 0;

CREATE VIEW `STUDENT_CS`
AS
```

```
SELECT *
FROM Student
WHERE Student.sdept = 'Computer Science';

UPDATE Score
SET Score.grade =
CASE WHEN Score.grade <= 95 THEN Score.grade + 5
ELSE 100 END
WHERE Score.sno IN (
     SELECT Score.sno
    FROM STUDENT_CS
);
```

Shortcut:

sno	sname	grade
001	Jingbo Su	100
001	Jingbo Su	100
001	Jingbo Su	100
001	Jingbo Su	100

(11) 创建一个带参数的存储过程，将数据库课程的成绩在 **low** 与 **high** 分数段的学生信息提取出来, 查询后的结果包括：姓名、所在的系、成绩。

Code:

```
USE NCUT;
DELIMITER //
DROP PROCEDURE IF EXISTS `__PROCEDURE_TASK__`;
CREATE PROCEDURE __PROCEDURE_TASK__(IN LOW int, IN HIGH int)
BEGIN
    SET @LB = LOW;
    SET @HB = HIGH;
    SELECT Student.sname, Student.sdept, Score.grade
    FROM Student, Course, Score
    WHERE Course.cname = 'Data Base'
       AND Student.sno = Score.sno
```

```
        AND Course.cno = Score.cno
        AND Score.grade BETWEEN @LB AND @HB;
END //
DELIMITER ;
CALL __PROCEDURE_TASK__(85, 100);
```

Shortcut:

sname	sdept	grade
Harry Potter	Gryffindor	90
Ron Weasley	Gryffindor	100
Hermione Granger	Gryffindor	100
Tom Riddle	Slughorn	100

7、 实验中的问题及解决方法

1. https://github.com/Sue217/NCUT/issues/2

Maximum Grade of each Student:

```
SELECT Student.sname, Score.cno, Score.grade
FROM Student, Score
WHERE (Student.sno = Score.sno)
    AND Score.grade IN (
        SELECT MAX(Score.grade)
        FROM Score
        WHERE Student.sno = Score.sno
        GROUP BY Score.sno
    );
```

Maximum Grade of each Course:

```
SELECT Student.Sname, Course.cno, Score.grade
    FROM Score, Student, Course
    WHERE (Student.sno = Score.sno)
        AND (Course.cno = Score.cno)
        AND Score.grade IN (
            SELECT MAX(Score.grade)
            FROM Score
            WHERE (Course.cno = Score.cno)
            GROUP BY Score.cno
        );
```

2. Bug https://github.com/Sue217/NCUT/issues/3

如果一个同学多门不同课有相同最高得分，那么这些最高成绩都会被选择出来！

八、实验的总结与收获

1. 掌握了根据数据字典创建E-R图, 再将E-R图转换为关系模式, 并最终使用sql 语言创建数据表的过程

2. 掌握了基本SQL数据查询、定义、控制、操纵等语句, 并在此基础上完成了相对复杂的查询

3. 熟练使用mySQL图形化界面, 并为后续实验数据库的连接做好前期准备工作