**《数据库原理及应用》实验报告**

题目: 数据定义与DML操作

院系：计算机科学与工程学院

班级：210402

姓名：谢鸿

学号：20212345

**实验报告（一）**

**一、实验题目**

实验一 数据定义与DML操作

**二、实验目的**

1. 掌握使⽤SQL语⾔创建数据库（模式） 的⽅法。

2. 掌握使⽤MySQL Workbench和SQL语⾔对表的操作⽅法。

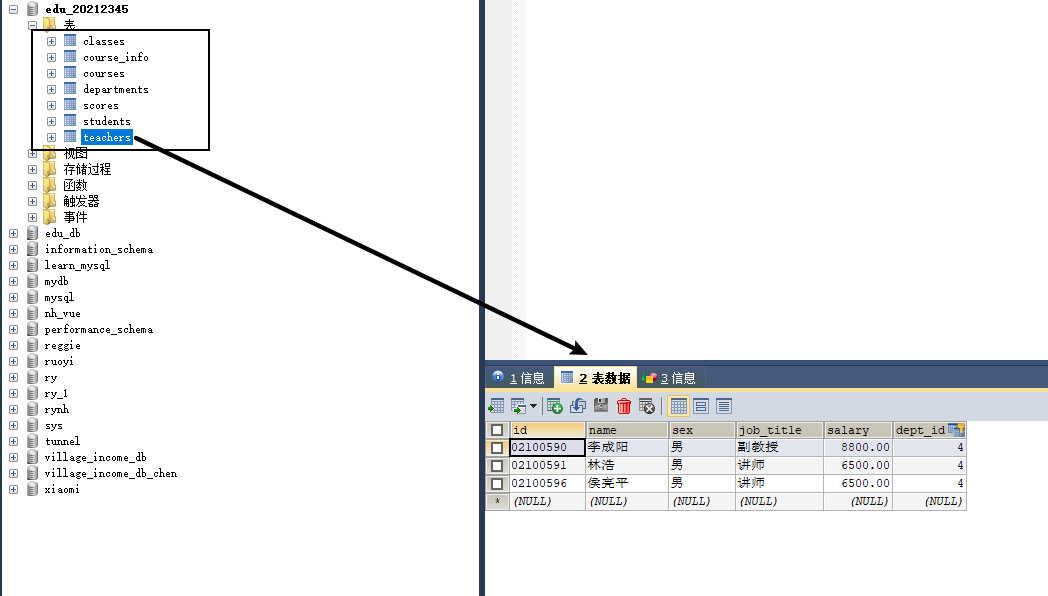
3. 掌握使⽤SQL语⾔插⼊、修改、删除数据 的⽅法。

4. 理解完成数据定义与DML操作可以有不同的解决⽅案（不同的⽅法、不同的SQL语句），要学会选择合理的⽅案。

**三、实验内容与实现**

**【实验要求一及实现】**

1. 创建 edu\_学号 数据库中的其余三张表：教师信息、开课信息 和 成绩信息。



2. 思考：

（1） 这三张表定义的先后顺序（考虑主/外键的关系）。

**【答】**：先创建“教师信息”表，因为它不依赖于其他表。

接下来创建“开课信息”表，因为它依赖于“教师信息”表的外键。

最后创建“成绩信息”表，因为它依赖于“学生信息”表（假设这个表已经存在）和“开课信息”表的外键。

（2）开课编号的⾃增主键如何设置。

**【答】**：可以使用AUTO\_INCREMENT属性来设置自增主键。这通常在创建表时，对主键列进行设置。

（3） 数据类型的不同⻓度对数据取值范围的影响。

**【答】**：不同长度的数据类型对性能的影响主要体现在存储空间的使用和检索效率上。较长的字段会消耗更多的存储空间，并且可能会降低检索和排序操作的效率。

（4） 如何定义三张表中的主键和外键约束。

**【答】**：在“教师信息”表中，教师ID可以设置为主键：

在“开课信息”表中，课程ID可以设置为主键，同时教师ID作为外键：

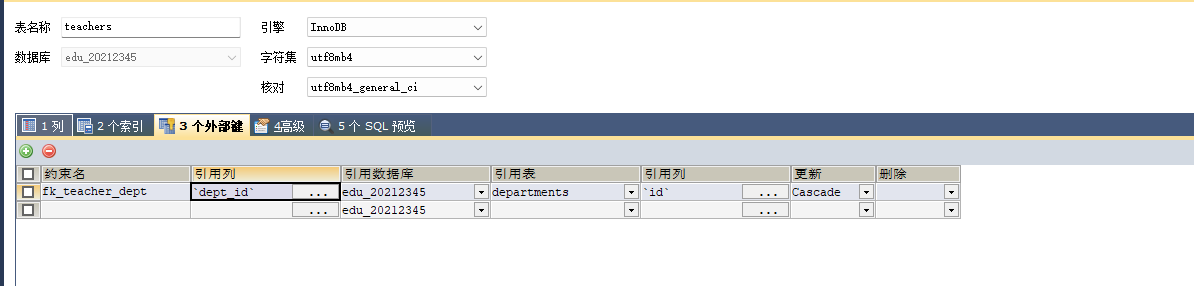
在“成绩信息”表中，可以使用复合主键，即学号和课程ID的组合：

3. 使⽤可视化⽅法定义 **教师信息**，要求：

（1）设置外键名为： fk\_teacher\_dept ；

（2）表外键：限制删除主键值且级联更新主键值；

（3）将设置好的表定义界⾯截图和⽣成的建表语句放⼊实验报告。



4. 使⽤SQL语句定义 **开课信息**，要求：

（1）定义时创建约束，⾃定义约束名；

（2）将**开课年份**列设置为字符型，占⽤4字节。

1. CREATE TABLE course\_info (
2. id INT AUTO\_INCREMENT,
3. course\_id CHAR(11) NOT NULL,
4. teacher\_id CHAR(8) NOT NULL,
5. year CHAR(4) NOT NULL, //开课年份设置为字符型，占用4字节
6. grade INT NOT NULL,
7. semester INT NOT NULL,
8. PRIMARY KEY (id),
9. CONSTRAINT fk\_courseinfo\_course FOREIGN KEY (course\_id) REFERENCES courses (id), // 自定义外键约束名
10. CONSTRAINT fk\_courseinfo\_teacher FOREIGN KEY (teacher\_id) REFERENCES teachers (id) //自定义外键约束名
11. );

5. 使⽤SQL语句定义 **成绩信息**，要求：

（1）定义时不创建约束；

（2）通过 **修改表** 的⽅式添加约束，给出约束名。

1. CREATE TABLE scores (
2. student\_id CHAR(8) NOT NULL,
3. course\_info\_id INT NOT NULL,
4. score\_type CHAR(4) NOT NULL,
5. score DECIMAL(5, 2)
6. );

添加主键约束

1. ALTER TABLE scores
2. ADD CONSTRAINT pk\_scores PRIMARY KEY (student\_id, course\_info\_id, score\_type);

添加外键约束，假设students和course\_info表已经存在

1. ALTER TABLE scores
2. ADD CONSTRAINT fk\_scores\_student FOREIGN KEY (student\_id) REFERENCES students (id);
3. ALTER TABLE scores
4. ADD CONSTRAINT fk\_scores\_course\_info FOREIGN KEY (course\_info\_id) REFERENCES course\_info (id);

6. 修改**开课信息**的**开课年份**字段，将数据类型设置为⻓度为4字节的整型。

1. ALTER TABLE course\_info
2. MODIFY COLUMN year INT;

**【实验要求二及实现】**

将上述 **其余数据** 插⼊到表中，实验记录下数据录⼊的SQL语句（按表的数据插⼊顺序），包括开课但未批卷的成绩信息。

1. 思考 **数据插⼊的顺序（考虑数据完整性要求）**，给出你认为正确的插⼊表的顺序并说明原因。

**【答】**：

插入顺序为：

departments teachers classes students courses course\_info scores

原因：

departments：这是其他表（如classes、courses和teachers）的外键依赖表，它应该首先被插入数据。

teachers：虽然它依赖于departments表，但它必须在courses和course\_info之前插入数据，因为那些表依赖于teachers表。

classes：它依赖于departments表，且students表将依赖于classes表。

students：它依赖于classes表，所以必须在classes表之后插入数据。

courses：它依赖于departments表，所以必须在departments表之后插入数据。

course\_info：它依赖于courses和teachers表，因此它们的数据必须先插入。

scores：最后插入，因为它依赖于students和course\_info表

1. 思考录⼊**开课信息**和**学⽣成绩**时表中各字段的取值。

**【答】**：

对于course\_info表，course\_id和teacher\_id必须是已经存在于courses和teachers表中的有效ID。year字段应该是一个有效的年份。

对于scores表，student\_id必须是students表中的一个有效ID，cs\_id必须是course\_info表中的一个有效ID，score可以为空，表示成绩未录入。

**【实验要求三及实现】**

实现如下操作，并将**语句**写⼊实验报告：

1. 19级学⽣选择的 **数据结构** 课程正考已经批改完毕，请根据如下表格更改分数。

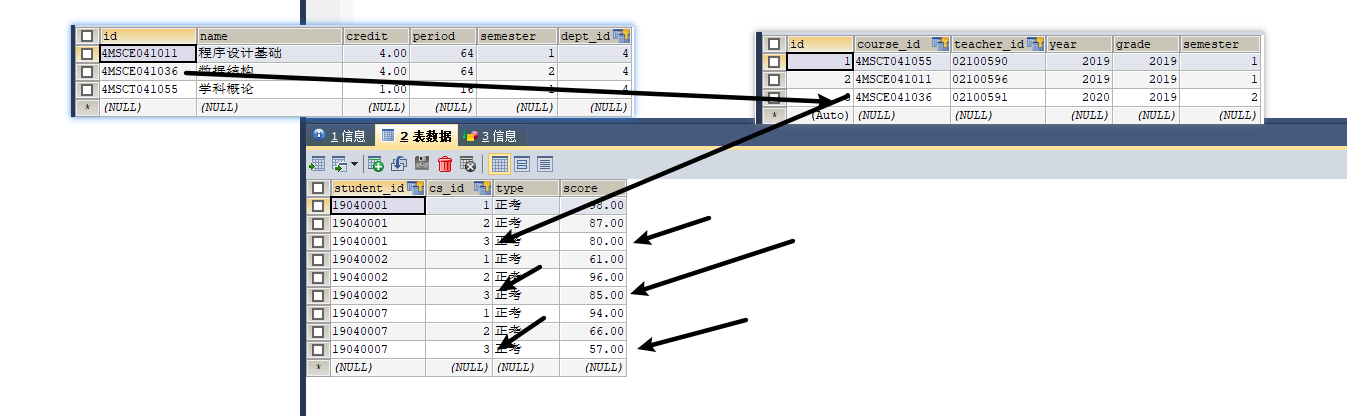
|  |  |
| --- | --- |
| 学号 | 分数 |
| 19040001 | 80.00 |
| 19040002 | 85.00 |
| 19040007 | 57.00 |

**【答】**：

SQL语句：

1. UPDATE scores
2. SET score = CASE student\_id
3. WHEN '19040001' THEN 80.00
4. WHEN '19040002' THEN 85.00
5. WHEN '19040007' THEN 57.00
6. END
7. WHERE student\_id IN ('19040001', '19040002', '19040007')
8. AND cs\_id = (SELECT id FROM course\_info WHERE course\_id = (SELECT id FROM courses WHERE NAME = '数据结构'))
9. AND TYPE = '正考';

数据视图：



1. 将**19级**学⽣的信息从**学⽣表**中删除。

提示：需要关联删除哪些表中的数据？删除的顺序是什么？

**【答】**：

删除19级学生的信息时，考虑到表间的外键依赖关系，需要确保删除操作不会违反数据完整性约束。在删除学生信息之前，必须先删除所有依赖于学生表主键的外键记录。

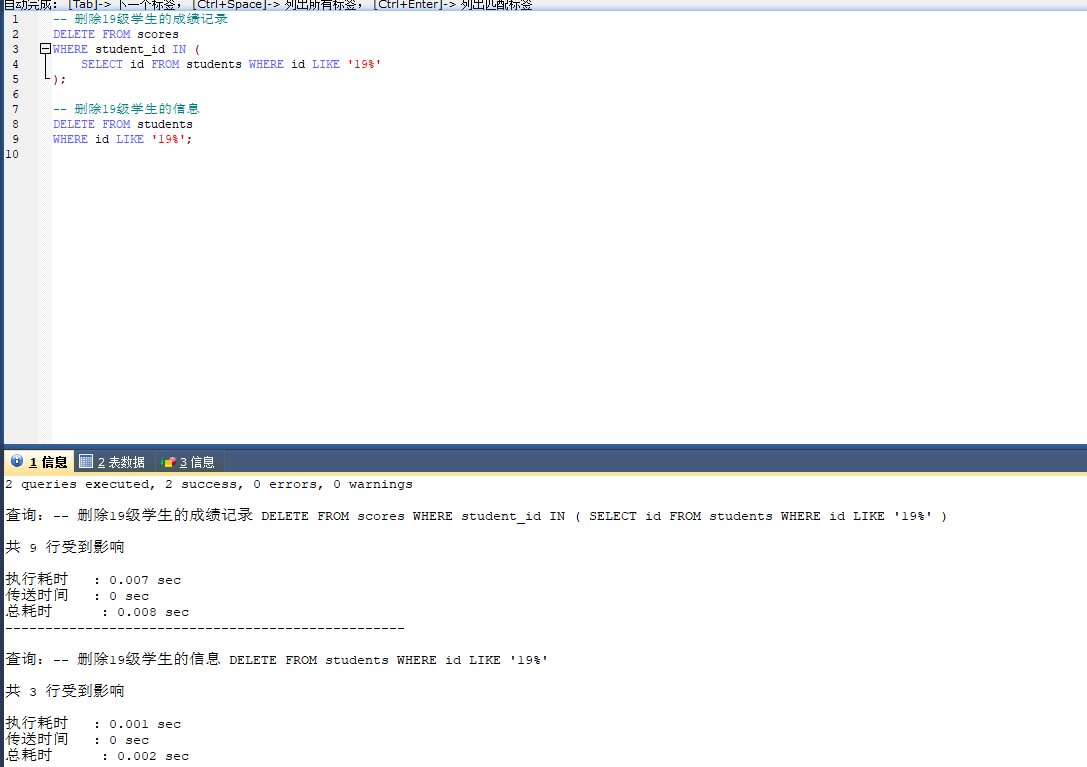
首先删除scores表中与19级学生相关的记录。

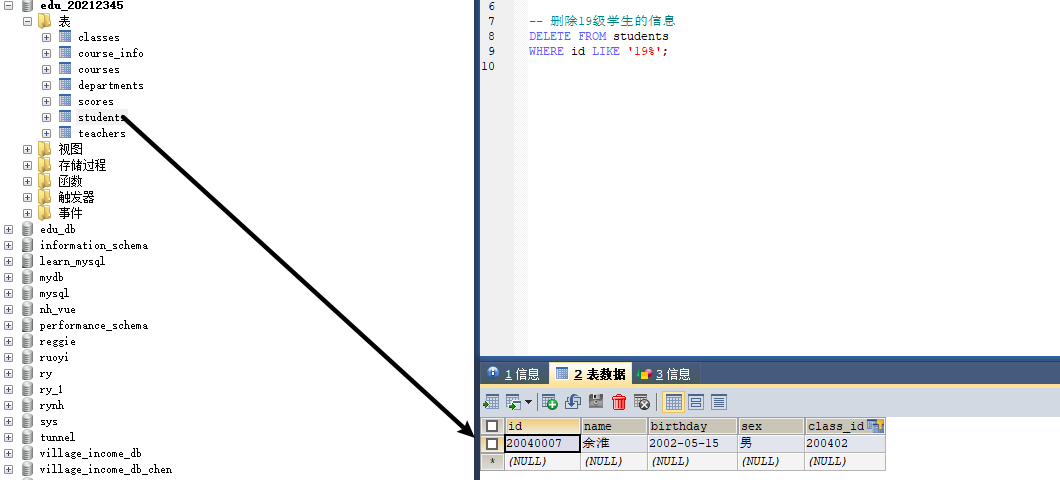
然后删除students表中的19级学生记录。

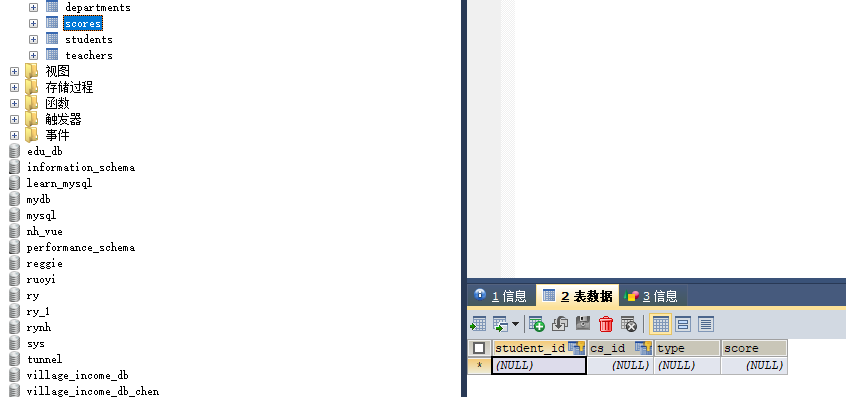
SQL语句：

1. // 删除19级学生的成绩记录
2. DELETE FROM scores
3. WHERE student\_id IN (
4. SELECT id FROM students WHERE id LIKE '19%'
5. );
6. // 删除19级学生的信息
7. DELETE FROM students
8. WHERE id LIKE '19%';

数据视图：







**四、实验问题总结与心得**

在这次实验中，我深入学习了使用SQL语言创建和管理数据库的技能。通过实际操作，我不仅掌握了创建数据库的基本命令，还了解了如何使用MySQL Workbench进行更高级的数据库操作。我发现，虽然初始时对于各种SQL语句和选项感到困惑，但通过实践和不断尝试，我逐渐掌握了它们的用法。

对我来说，学习如何正确插入、修改和删除数据是特别重要的。这不仅是数据库管理的基础，也是维护数据完整性和准确性的关键。通过在不同的数据集上实践这些操作，我更加了解了这些命令在实际应用中的重要性。

另外，我也意识到了在进行数据定义和数据操作时，存在多种不同的方法和SQL语句。根据数据的特点和操作的目标选择最合适的方法，对于提高工作效率和确保数据质量至关重要。这次实验不仅增强了我的技术技能，也提高了我的问题解决能力和决策能力。