

信息与软件工程学院上机实验报告

(第 1 次)

一、实验名称

实验一：数据库 DDL 实验

二、实验目的及要求

2.1 知识能力目标

1. 掌握数据库系统的基本概念和基本知识，能够使用 SQL 完成数据库和基本表的定义、操纵与控制。
2. 理解数据库完整性机制，能够运用主键、唯一约束、外键和检查约束实现实体完整性、参照完整性和用户自定义完整性。
3. 围绕 StudentCourse 与 SPJ 两个数据库，完成建库、建表、结构修改、索引处理和结果验证，形成数据库初步建立与设计能力。

2.2 素养目标

1. 按照工程认证“研究”要求，围绕实验目标设计执行步骤，分析运行结果，并据此形成清晰结论。
2. 按照工程认证“使用现代工具”要求，能够选用 Docker、MySQL 和 DataGrip 等工具完成数据库环境搭建、脚本执行与结果验证。
3. 按照工程认证“工程与可持续发展”要求，理解数据库对象设计应兼顾规范性、可维护性和后续复用需求，为后续实验继续使用同一数据库奠定基础。

三、实验环境

1. Docker 28.5.2, build ecc6942
2. MySQL 8.0
3. DataGrip 2026.1.1

四、实验设计

4.1 实验内容与步骤

4.1.1 建立 StudentCourse 数据库和基本表

目的：完成 StudentCourse 数据库及其三张基本表的定义，为后续完整性约束、索引处理和数据验证提供结构基础。

核心代码如下：

```
CREATE DATABASE IF NOT EXISTS StudentCourse
DEFAULT CHARACTER SET utf8mb4
```

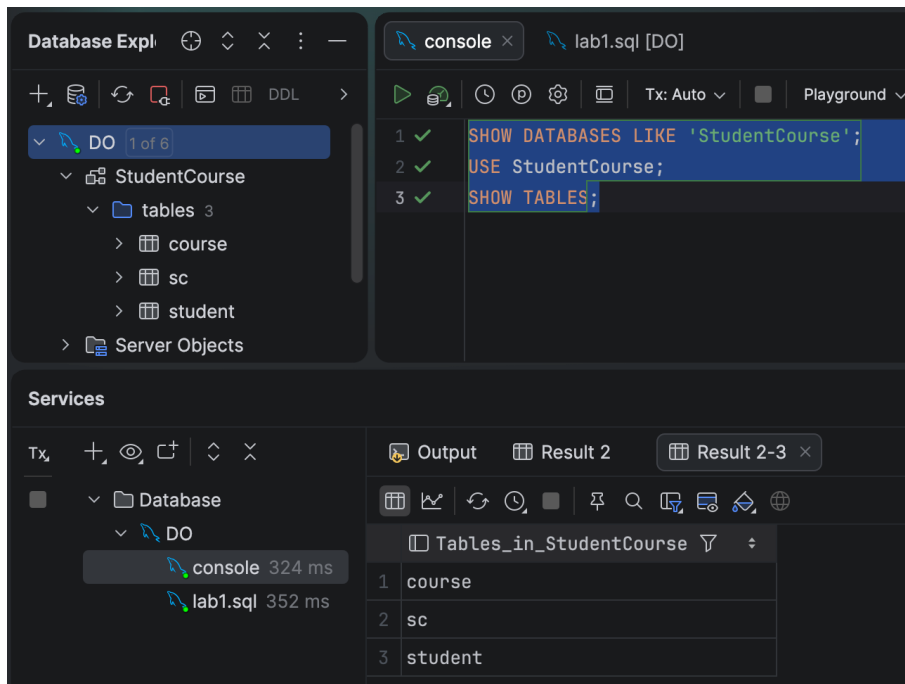
```
COLLATE utf8mb4_unicode_ci;

USE StudentCourse;

CREATE TABLE student (
    s_no CHAR(8),
    s_name VARCHAR(20),
    s_sex CHAR(6),
    s_birthdate DATE,
    s_major VARCHAR(40),
    CONSTRAINT pk_student PRIMARY KEY (s_no),
    CONSTRAINT uk_student_s_name UNIQUE (s_name),
    CONSTRAINT chk_student_s_sex CHECK (s_sex IN ('男', '女'))
);

CREATE TABLE course (
    c_no CHAR(5),
    c_name VARCHAR(40) NOT NULL,
    c_credit SMALLINT,
    c_pno CHAR(5),
    CONSTRAINT pk_course PRIMARY KEY (c_no),
    CONSTRAINT chk_course_c_credit CHECK (c_credit > 0),
    CONSTRAINT fk_course_c_pno FOREIGN KEY (c_pno) REFERENCES course(c_no)
);

CREATE TABLE sc (
    s_no CHAR(8),
    c_no CHAR(5),
    grade SMALLINT,
    semester CHAR(5),
    teaching_class CHAR(8),
    CONSTRAINT pk_sc PRIMARY KEY (s_no, c_no),
    CONSTRAINT chk_sc_grade CHECK (grade IS NULL OR (grade >= 0 AND grade <= 100)),
    CONSTRAINT fk_sc_s_no FOREIGN KEY (s_no) REFERENCES student(s_no),
    CONSTRAINT fk_sc_c_no FOREIGN KEY (c_no) REFERENCES course(c_no)
);
```



4.1.2 修改 StudentCourse 表结构并处理索引

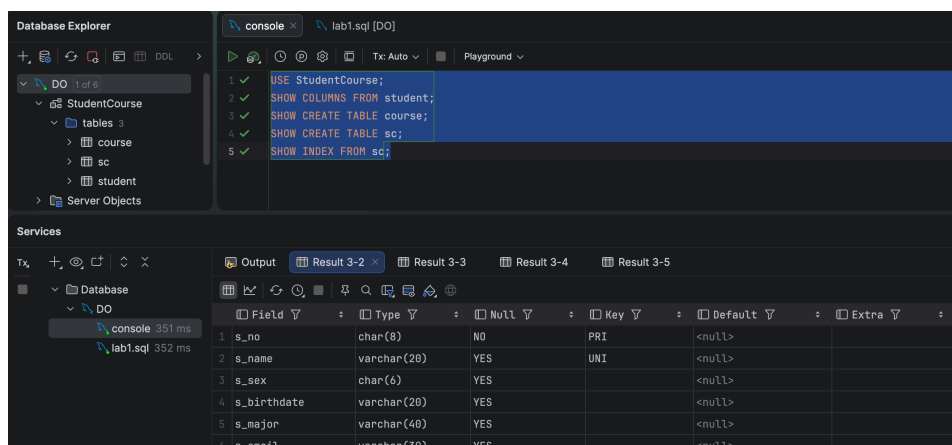
目的：通过 ALTER TABLE 与索引语句完成字段调整、补充约束和索引管理，体现数据库对象演化与性能辅助设计过程。

核心代码如下：

```
ALTER TABLE student ADD s_email VARCHAR(30);
ALTER TABLE student MODIFY s_birthdate VARCHAR(20);
ALTER TABLE course ADD CONSTRAINT uk_course_c_name UNIQUE (c_name);

CREATE UNIQUE INDEX idx_stu_s_name ON student(s_name);
CREATE UNIQUE INDEX idx_cou_c_name ON course(c_name);
CREATE UNIQUE INDEX idx_sc_c_co ON sc(s_no ASC, c_no DESC);

ALTER TABLE sc RENAME INDEX idx_sc_c_co TO idx_sc_sno_cno;
ALTER TABLE student DROP INDEX idx_stu_s_name;
```



The first screenshot shows the SQL script for creating the 'course' table and its indexes. The script is as follows:

```
USE StudentCourse;
SHOW COLUMNS FROM student;
SHOW CREATE TABLE course;
SHOW CREATE TABLE sc;
SHOW INDEX FROM sc;
```

The second screenshot shows the SQL script for creating the 'sc' table and its indexes. The script is as follows:

```
USE StudentCourse;
SHOW COLUMNS FROM student;
SHOW CREATE TABLE course;
SHOW CREATE TABLE sc;
SHOW INDEX FROM sc;
```

The third screenshot shows the SQL script for creating the 'sc' table and its indexes. The script is as follows:

```
USE StudentCourse;
SHOW COLUMNS FROM student;
SHOW CREATE TABLE course;
SHOW CREATE TABLE sc;
SHOW INDEX FROM sc;
```

The third screenshot also displays the output of the SQL script, showing the table structure and indexes for the 'sc' table.

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality
sc	0	PRIMARY	1	s_no	A	
sc	0	PRIMARY	2	c_no	A	
sc	0	idx_sc_sno_cno	1	s_no	A	
sc	0	idx_sc_sno_cno	2	c_no	D	
sc	1	fk_sc_c_no	1	c_no	A	

4.1.3 装入 StudentCourse 基础数据并验证选课关系

目的：向三张基本表装入实验数据，并通过连接查询验证选课记录、课程信息和学生信息之间的参照关系。

核心代码如下：

```
INSERT INTO student (s_no, s_name, s_sex, s_birthdate, s_major, s_email)
VALUES (...);
```

```
INSERT INTO course (c_no, c_name, c_credit, c_pno) VALUES (...);
INSERT INTO sc (s_no, c_no, grade, semester, teaching_class) VALUES (...);
```

```
SELECT * FROM student ORDER BY s_no;
SELECT * FROM sc ORDER BY s_no, c_no;
```

```
SELECT
    s.s_no AS student_no,
    s.s_name AS student_name,
    c.c_name AS course_name,
    sc.semester,
    sc.grade
FROM sc
JOIN student AS s ON sc.s_no = s.s_no
JOIN course AS c ON sc.c_no = c.c_no
ORDER BY s.s_no, c.c_no;
```

console x lab1.sql [DO]

```

12 JOIN student AS s 1..n<->1: ON sc.s_no = s.s_no
13 JOIN course AS c 1..n<->1: ON sc.c_no = c.c_no
14 ORDER BY s.s_no, c.c_no;

```

Output StudentCourse.student StudentCourse.sc Result 4-4

	s_no	s_name	s_sex	s_birthdate	s_major	s_email
1	20240001	张晨	男	2005-03-15	信息管理与信息系统	zhangchen@sc.edu
2	20240002	李晴	女	2005-05-21	信息管理与信息系统	liqing@sc.edu
3	20240003	王越	男	2004-11-06	计算机科学与技术	wangyue@sc.edu
4	20240004	周宁	女	2005-01-18	软件工程	zhouning@sc.edu

console x lab1.sql [DO]

```

12 JOIN student AS s 1..n<->1: ON sc.s_no = s.s_no
13 JOIN course AS c 1..n<->1: ON sc.c_no = c.c_no
14 ORDER BY s.s_no, c.c_no;

```

Output StudentCourse.student StudentCourse.sc Result 4-4

SELECT * FROM sc ORDER BY s_no, c_no

	s_no	c_no	grade	semester	teaching_class
1	20240001	C0001	86	24101	DB240101
2	20240001	C0004	90	24102	DB240102
3	20240002	C0001	93	24101	DB240101
4	20240003	C0002	88	24101	CS240201
5	20240003	C0003	84	24102	CS240202
6	20240004	C0002	91	24101	SE240101

	s.s_no AS 学号	s.s_name AS 姓名	c.c_name AS 课程名称	sc.semester AS 学期	sc.grade AS 成绩
1	20240001	张晨	数据库原理	24101	86
2	20240001	张晨	数据库应用实验	24102	90
3	20240002	李晴	数据库原理	24101	93
4	20240003	王越	程序设计基础	24101	88
5	20240003	王越	数据结构	24102	84
6	20240004	周宁	程序设计基础	24101	91

4.1.4 建立 SPJ 数据库和基本表

目的：完成 SPJ 经典供应商-零件-工程数据库的结构定义，突出联合主键、外键和检查约束在多表场景中的应用。

核心代码如下：

```
CREATE DATABASE IF NOT EXISTS SPJ
DEFAULT CHARACTER SET utf8mb4
COLLATE utf8mb4_unicode_ci;

USE SPJ;

CREATE TABLE S (
    SNO VARCHAR(10),
    SNAME VARCHAR(50) NOT NULL,
    STATUS INT NOT NULL,
    CITY VARCHAR(50) NOT NULL,
    CONSTRAINT pk_s PRIMARY KEY (SNO),
    CONSTRAINT chk_s_status CHECK (STATUS > 0)
);

CREATE TABLE P (
    PNO VARCHAR(10),
    PNAME VARCHAR(50) NOT NULL,
    COLOR VARCHAR(20) NOT NULL,
    WEIGHT INT NOT NULL,
    CONSTRAINT pk_p PRIMARY KEY (PNO),
    CONSTRAINT chk_p_weight CHECK (WEIGHT > 0)
);

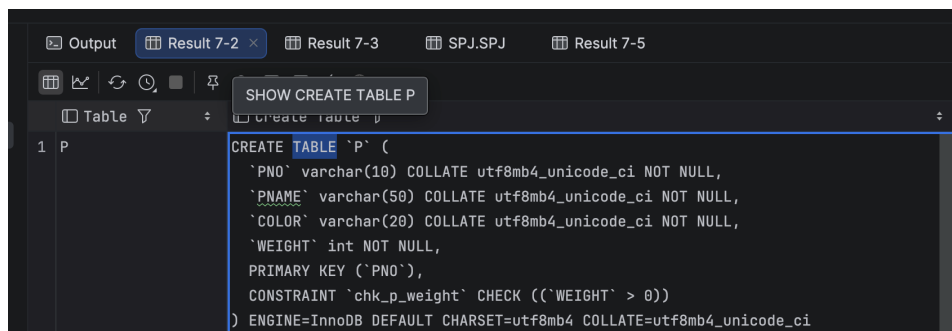
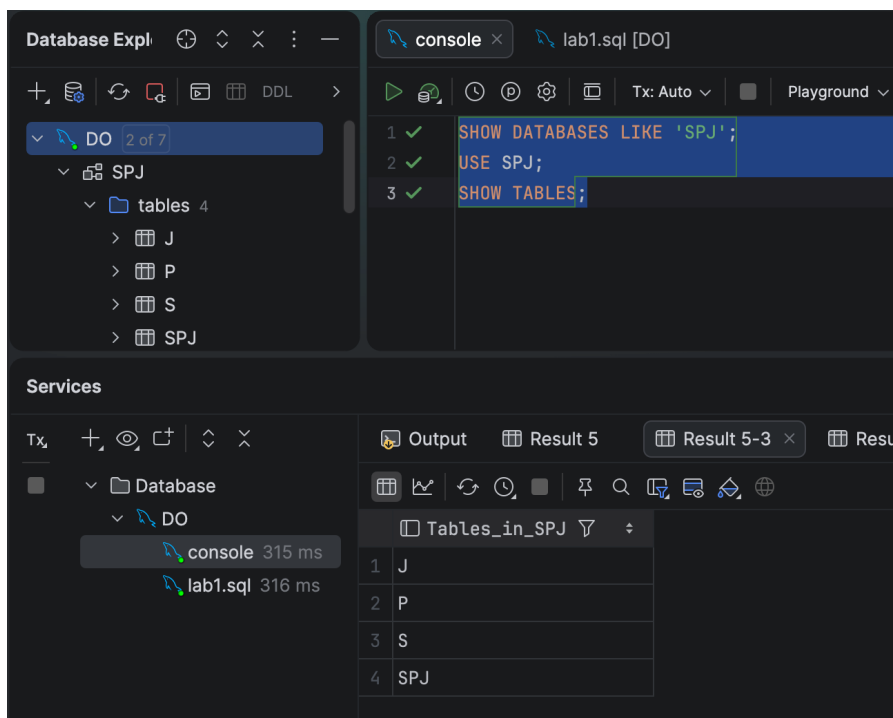
CREATE TABLE J (
    JNO VARCHAR(10),
    JNAME VARCHAR(50) NOT NULL,
    CITY VARCHAR(50) NOT NULL,
```

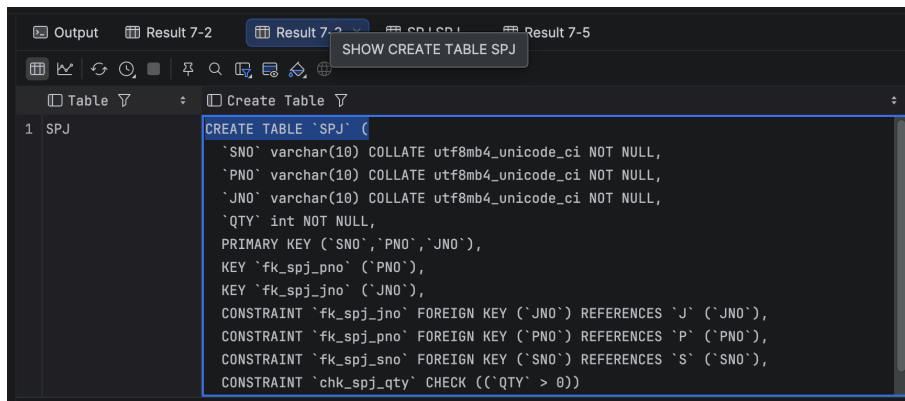
```

CONSTRAINT pk_j PRIMARY KEY (JNO)
);

CREATE TABLE SPJ (
    SNO VARCHAR(10) NOT NULL,
    PNO VARCHAR(10) NOT NULL,
    JNO VARCHAR(10) NOT NULL,
    QTY INT NOT NULL,
    CONSTRAINT pk_spj PRIMARY KEY (SNO, PNO, JNO),
    CONSTRAINT chk_spj_qty CHECK (QTY > 0),
    CONSTRAINT fk_spj_sno FOREIGN KEY (SNO) REFERENCES S(SNO),
    CONSTRAINT fk_spj_pno FOREIGN KEY (PNO) REFERENCES P(PNO),
    CONSTRAINT fk_spj_jno FOREIGN KEY (JNO) REFERENCES J(JNO)
);

```





4.1.5 装入 SPJ 基础数据并验证供应关系

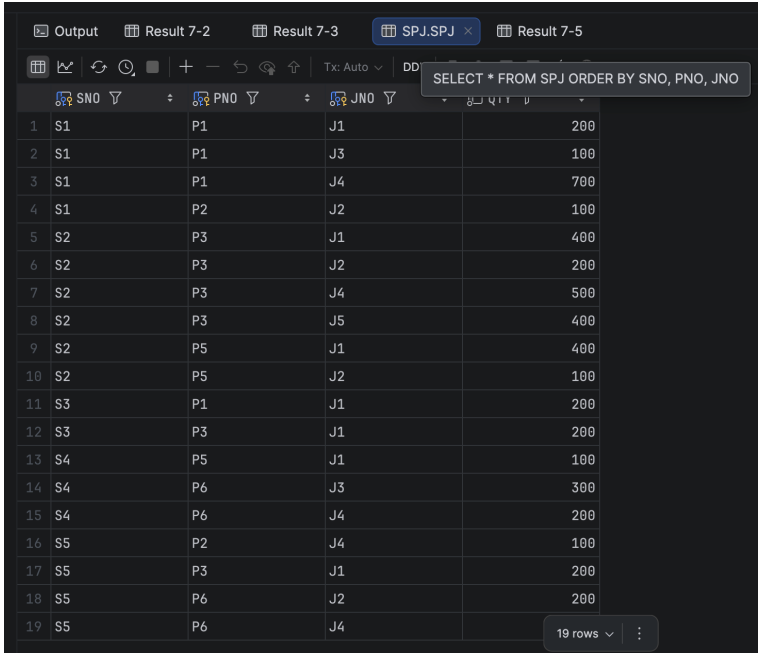
目的：向 SPJ 数据库装入基础数据，通过多表连接验证供应商、零件和工程项目之间的供应关系是否建立正确。

核心代码如下：

```
INSERT INTO S (SNO, SNAME, STATUS, CITY) VALUES (...);
INSERT INTO P (PNO, PNAME, COLOR, WEIGHT) VALUES (...);
INSERT INTO J (JNO, JNAME, CITY) VALUES (...);
INSERT INTO SPJ (SNO, PNO, JNO, QTY) VALUES (...);
```

```
SELECT * FROM SPJ ORDER BY SNO, PNO, JNO;
```

```
SELECT
    SPJ.SNO,
    S.SNAME,
    SPJ.PNO,
    P.PNAME,
    SPJ.JNO,
    J.JNAME,
    SPJ.QTY
FROM SPJ
JOIN S ON SPJ.SNO = S.SNO
JOIN P ON SPJ.PNO = P.PNO
JOIN J ON SPJ.JNO = J.JNO
ORDER BY SPJ.SNO, SPJ.PNO, SPJ.JNO;
```



SNO	SNAME	PNO	PNAME	JNO	JNAME	QTY
1	S1	P1	螺母	J1	三建	200
2	S1	P1	螺母	J3	弹簧厂	100
3	S1	P1	螺母	J4	造船厂	700
4	S1	P2	螺栓	J2	一汽	100
5	S2	P3	螺丝刀	J1	三建	400
6	S2	P3	螺丝刀	J2	一汽	200
7	S2	P3	螺丝刀	J4	造船厂	500
8	S2	P3	螺丝刀	J5	机车厂	400
9	S2	P5	凸轮	J1	三建	400
10	S2	P5	凸轮	J2	一汽	100
11	S3	P1	螺母	J1	三建	200
12	S3	P3	螺丝刀	J1	三建	200
13	S4	P5	凸轮	J1	三建	100
14	S4	P6	齿轮	J3	弹簧厂	300
15	S4	P6	齿轮	J4	造船厂	200
16	S5	P2	螺栓	J4	造船厂	100
17	S5	P3	螺丝刀	J1	三建	200
18	S5	P6	齿轮	J2	一汽	200
19	S5	P6	齿轮	J4	造船厂	500

4.2 实验总结与心得体会

本次实验围绕数据库 DDL 操作完成了两个典型数据库的建立与验证。通过 StudentCourse 数据库的设计与调整，进一步掌握了数据库、基本表、主键、唯一约束、外键、检查约束以及索引的创建与修改方法；通过 SPJ 数据库的建立与关联验证，进一步理解了多表结构下联合主键和参照完整性的实现方式。

实验过程中可以看出，数据库设计不仅要保证对象能够成功创建，还要关注约束是否完整、结构修改是否规范、查询结果是否与预期一致。只有把建库、建表、结构调整、数据装入和结果验证作为一个完整过程来处理，才能为后续 DML、视图、授权和备份恢复等实验提供稳定基础。