2024~2025 春数据库及实现上机实验报告一

陈远洋

2025 年 4 月 12 日

1 实验环境说明

我的本地操作系统为 Windows11，64 位操作系统, 基于 x64 的处理器。安装的 MySQL 版本信息：`Ver 8.0.41 for Win64 on x86_64 (MySQL Community Server - GPL)`。实验中使用的数据库软件为 MySQL Workbench 8.0.41。

2 任务 1：概念模型（E-R 图）画法与逻辑模式转换实验

2.1 实验目的

1. 掌握 E-R 图的构成要素及图元。
2. 学习如何绘制图书馆管理系统的 E-R 图。
3. 掌握从概念模型（E-R 图）向逻辑模型（关系模式）的转换原则和步骤。

2.2 实验内容

题目 1：为一个图书馆设计一套图书管理的数据库，要求包括出版社和书籍的信息。出版社信息包括出版社名称、地址、电话；书籍信息包括书名、作者、ISBN、价格、出版日期。

1. 确定书籍实体和出版社实体的属性。（2 分）
   书籍（Book）：书名 BN, 作者 AT, 国际标准书号 ISBN, 价格 P。
   出版社（Press）：出版社名称 PN, 地址 AD, 电话 TEL。
2. 确定书籍和出版社之间的联系，给联系命名并指出联系的类型。（2 分）
   联系名称：出版
   联系类型：一对多（出版社对应书籍）
3. 确定联系本身的属性。（2 分）
   出版属性：出版日期 DATE
4. 画出书籍与出版社关系的 E-R 图。（4 分）
   见图一。

5. 将 E-R 图转化为关系模式，写出表的关系模式并标明各自的码。（2 分）
   实体“书籍”→ Book(ISBN, BN, AT, P, DATE, PN) PK: ISBN, FK: PN
   实体“出版社”→ Press(PN, AD, TEL) PK: PN

题目 2：设计一个图书馆员工和会员的管理系统。图书馆员工信息包括员工编号、姓名、电话等；会员信息包括会员编号、姓名、电话等。员工与会员之间存在“服务”联系，每个员工可服务多个会员，但每个会员只能由一个员工服务，员工服务会员有“开始服务日期”和“服务次数”两个属性；员工与书籍之间存在“管理”联系，每个员工可管理多本书籍，而每本书籍只能由一个员工管理；会员与书籍之间存在“借阅”联系，会员借阅书籍有“借阅日期”和“归还日期”两个属性，每个会员可借阅多本书籍，每本书籍可被多个会员借阅。

1. 确定实体和实体的属性。（2 分）
   员工：员工编号 SID, 姓名 SN, 电话 STEL
   会员：会员编号 VID, 姓名 VN, 电话 VTEL

2. 确定实体之间的联系，给联系命名并指出联系的类型。（2 分）
   员工 ↔ 会员，命名：服务类型：1 对多联系
   员工 ↔ 书籍，命名：管理类型：1 对多联系
   会员 ↔ 书籍，命名：借阅类型：多对多联系

3. 确定联系本身的属性。（2 分）
   服务：开始服务日期 SD，服务次数 ST
   管理：
   借阅：借阅日期 BD, 归还日期 ED

4. 画出 E-R 图。（4 分）
   见图二。

5. 将 E-R 图转化为关系模式，写出表的关系模式并标明各自的码。（3 分）
   实体“员工”→ Staff(SID, SN, STEL) PK: SID
   实体“书籍”→ Book(ISBN, BN, AT, P, DATE, PN, SID) PK: ISBN; FK: PN, SID
   实体“会员”→ VIP(VID, VN, VTEL, SID, SD, ST) PK:VID; FK:SID
   联系“借阅”→ Borrow(VID,ISBN,BD,ED) PK:VID+ISBN+BD; FK:VID,ISBN

3 任务 2：关系的完整性、规范化理解与应用实验

3.1 实验目的

1. 了解关系模型的基本概念，掌握候选码和主码的确定。

2. 掌握并应用完整性规则。

3. 掌握关系规范化的定义和方法。

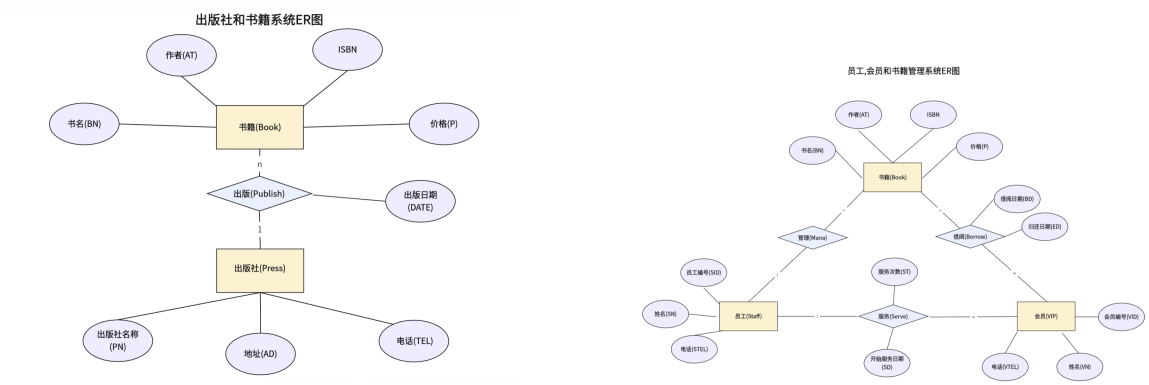


图 1: 图一、图二 E-R 图

3.2 设计性实验

某医院设计了药品库存管理系统，设计了如下药品订单表，请你用规范化理论将该表进行分解，使之满足 3NF 的规范化要求。（10 分）

表 1: 药品订单表

订单号	药品代码	药品名称	生产厂家	厂家地址	单价	订购数量	订购医院	医院地址
OD20190301	PH5002	阿莫西林	华北制药	河北省石家庄	12	100	人民医院	江苏省南京市
OD20190302	PH0038	感冒灵	仁和药业	浙江省杭州市	8	200	同仁医院	浙江省杭州市
OD20190303	PH1073	板蓝根	康恩贝集团	江苏省南通市	15	150	协和医院	北京市
OD20190304	PH2042	肠胃宁	修正药业	辽宁省沈阳市	20	80	同济医院	上海市
OD20190305	PH5002	阿莫西林	华北制药	河北省石家庄	12	120	中日医院	北京市

考虑到三范式的定义，即表中没有传递函数依赖。而此表中有药品代码 → 药品名称 + 生产厂家，生产厂家 → 厂家地址，订购医院 → 医院地址，订单号 → 药品代码 + 订购医院 + 订购数量 + 单价，等函数依赖可推出的部分函数依赖、传递函数依赖。所以需要进行分解。

将原表格分解为三个独立实体：药品，厂家，医院；以及一个联系关系：订单。分解如下：药品 (药品代码, 药品名称, **生产厂家**, 单价)，厂家 (厂家名称, 厂家地址)，医院 (医院名称, 医院地址)，订单 (订单号, **药品代码**, **订购医院**, 订购数量)。(画下划线的属性为主键，加粗的属性为外键) 见表 2.

3.3 观察与思考

1. 有如下所示的两张表：假设向关系 M 中插入新行，新行的数据分别如下。哪些行能够插入？若不能插入，为什么？（3 分）

A. (‘P004’, ‘银翘解毒片’, ‘S05’)

B. (‘P005’, ‘感冒灵’, null)

C. (‘P003’, ‘板蓝根’, ‘S03’)

D. (‘P006’, ‘藿香正气水’, ‘S01’)

E. (‘P007’, ‘牛黄解毒片’, ‘S07’)

表 2: 分解后的表格

药品代码	药品名称	生产厂家	单价
PH5002	阿莫西林	华北制药	12
PH0038	感冒灵	仁和药业	8
PH1073	板蓝根	康恩贝集团	15
PH2042	肠胃宁	修正药业	20

厂家名称	厂家地址
华北制药	河北省石家庄
仁和药业	浙江省杭州市
康恩贝集团	江苏省南通市
修正药业	辽宁省沈阳市

订单号	药品代码	订购医院	订购数量
OD20190301	PH5002	人民医院	100
OD20190302	PH0038	同仁医院	200
OD20190303	PH1073	协和医院	150
OD20190304	PH2042	同济医院	80
OD20190305	PH5002	中日医院	120

医院名称	医院地址
人民医院	江苏省南京市
同仁医院	浙江省杭州市
协和医院	北京市
同济医院	上海市
中日医院	北京市

表 3: 供应商关系 S（主码是“供应商 ID”）

供应商 ID	供应商名称	所在城市
S01	健康药业	南京
S02	万艾可	北京
S03	长生堂	广州
S04	九芝堂	成都

表 4: 药品关系 M（主码是“药品 ID”，外码是“供应商 ID”）

药品 ID	药品名称	供应商 ID
P001	阿司匹林	S01
P002	复方感冒胶囊	S02
P003	维生素 C	S03

A, C, E 不能。A, E 是因为外键在 S 中取值不存在；C 是因为主键冲突。B, D 可以。因为外键取值为 S 中的值或者空。

2. 非规范化数据表可能带来哪些不利影响？（3 分）

较多的数据冗余，较低的数据结构化程度以及可能带来的数据更新异常较低的数据共享度以及可能带来的数据一致性较差。

4. 在规范化过程中，为什么要避免传递依赖？（3 分）

若存在传递依赖可能导致数据冗余：对一些传递依赖的列可能会被多次重复。更新，插入，删除异常：在修改某个规则时，可能需要对表中成千上万的键进行修改，如此带来大量的工作量以及可能导致数据一致性被破坏。

5. 如果一个关系已经处于 2NF，它还可能存在哪些问题？（3 分）

可能仍然存在传递函数依赖，即不满足 3NF。此时可能遇到的问题同上即：

- 数据冗余
- 更新异常
- 插入异常

4 任务 3：MySQL 安装创建和维护数据库实验

4.1 实验目的

1. 熟悉在 Windows、Mac 或 Linux 平台下安装与配置 MySQL 的方法。
2. 掌握启动服务并登录 MySQL 数据库的多平台方法和步骤。
3. 了解手工配置 MySQL 的跨平台方法。
4. 深入理解 MySQL 数据库的相关概念。
5. 掌握使用 MySQL Workbench/Navicat 等客户端工具和 SQL 语句在多平台创建数据库的方法。
6. 掌握使用 MySQL Workbench/Navicat 等客户端工具和 SQL 语句在多平台删除数据库的方法。

4.2 实验内容

1. 在 Windows、Mac 或 Linux 平台下安装 MySQL。（2 分）
   从 MySQL 官网下载 MySQL Installer。运行安装程序，选择“Developer Default”选项。根据提示完成安装。我这里选择`Ver 8.0.41 for Win64 on x86_64 (MySQL Community Server - GPL)`。
2. 在服务（Windows）/系统偏好设置（Mac）/系统服务（Linux）中，手动启动或者关闭 MySQL 服务。（2 分）
   打开“服务”管理器（Win + R -> 输入 services.msc）。找到“MySQL”服务，右键选择“启动”或“停止”。见图 3
3. 使用命令行在各平台启动或关闭 MySQL 服务。（2 分）
   打开终端管理员模式下的 powershell，分别输入

   ```
   net start mysql80
   net stop mysql80
   ```

   以开关 mysql 服务。
4. 分别用 MySQL Workbench/Navicat 等客户端工具和命令行方式在各平台登录 MySQL。（2 分）
   下载 MySQL Workbench 按照提示完成安装，新建连接。对命令行模式输入`mysql -u root -p`，输入密码，进入 MySQL 命令行。见图 3

5. 在配置文件（my.ini 或 my.cnf）中将数据库的存储位置改为不同的路径，例如 `D:\MYSQL\DATA` 或 `/var/mysql/data`，然后重启服务，并验证路径更改的有效性。（2 分）
   找到配置文件 my.ini，修改 [mysqld] 部分的 datadir 参数，并将 data 目录下的文件拷贝到目标地址，重启服务。见图 3

6. 创建数据库。

   (a) 使用 MySQL Workbench/Navicat 等客户端工具在各平台创建教学管理数据库 JXGL。（2 分）
   在 MySQL Workbench 中，点击“创建数据库”。输入名称“JXGL”，选择字符集，点击确定。

   (b) 使用 SQL 语句在各平台创建数据库 MyTestDB。（2 分）
   在 SQL 语句查询栏中输入`CREATE DATABASE MyTestDB;`，然后点击运行。见图 3。

7. 查看数据库属性。

   (a) 在 MySQL Workbench/Navicat 等客户端工具中查看创建后的 JXGL 数据库和 MyTestDB 数据库的状态，及其文件所在的文件夹。（2 分）
   在 MySQL Workbench 中，右键数据库“JXGL”或“MyTestDB”，查看其状态和存储路径。或者键入 SQL 语句

```
SELECT * from information_schema.SCHEMATA
WHERE SCHEMA_NAME IN ('JXGL','MyTestDB');
```

   (b) 使用 SHOW DATABASES 命令在各平台显示当前的所有数据库。（2 分）
   见图 3。

8. 删除数据库。

   (a) 使用 MySQL Workbench/Navicat 等客户端工具在各平台删除 JXGL 数据库。（2 分）
   在 MySQL Workbench 中，右键数据库“JXGL”，选择“删除”。

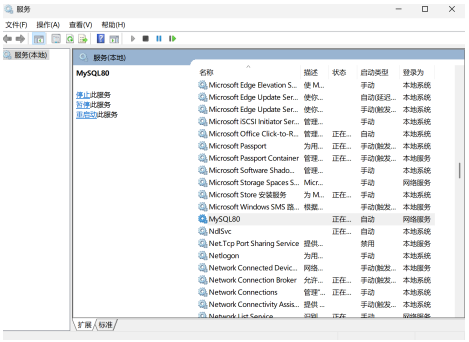
   (b) 使用 SQL 语句在各平台删除 MyTestDB 数据库。（2 分）
   键入 SQL 语句`DROP DATABASE MyTestDB`，然后点击运行。

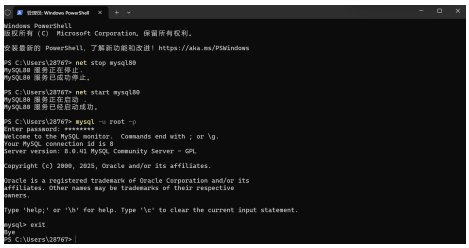
   (c) 使用 SHOW DATABASES 命令在各平台显示当前的所有数据库。（2 分）
   完整结果见图 3。

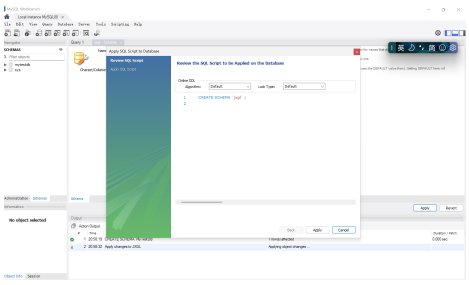
4.3 观察与思考

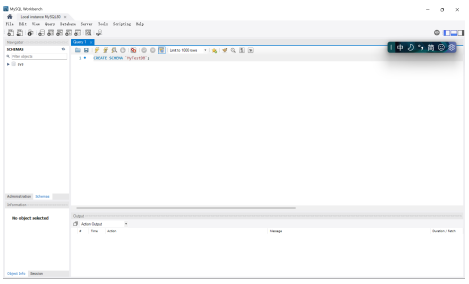
（1）如何在不同平台备份 MySQL 数据库？
根据查找到的资料，可以使用 ‘mysqldump’ 工具来备份 MySQL 数据库。基本命令格式如下：

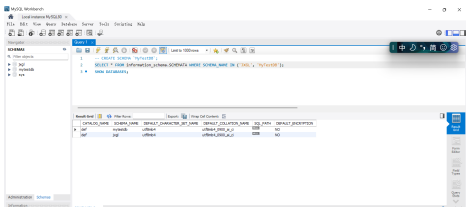
```
mysqldump -u [username] -p[password] [database_name] > backup.sql
```

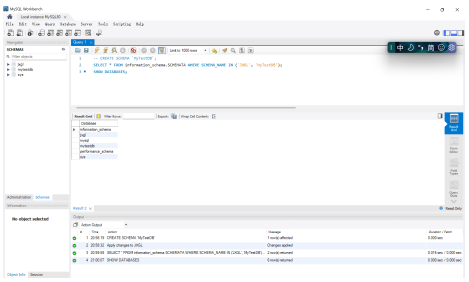


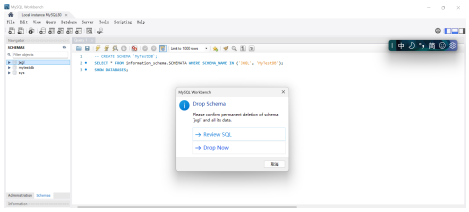


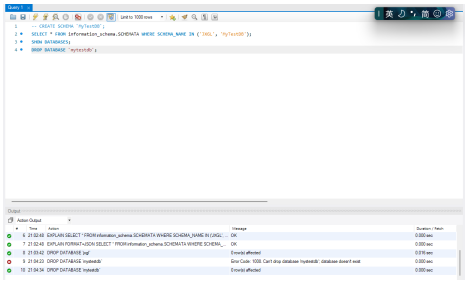


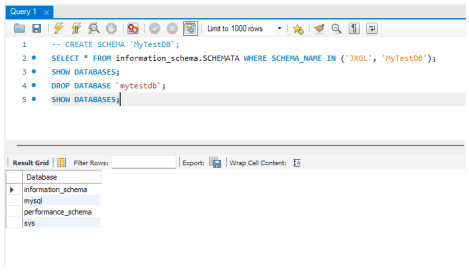












（2）讨论跨平台使用 MySQL 时的字符编码问题，如何配置以支持多语言？

可以将 MySQL 服务器的默认字符集设置为 ‘utf8mb4’，其包括 emoji 在内的所有 Unicode 字

符。

（3）如何在不同平台上通过配置文件定制 MySQL 的内存使用和连接限制？

如果需要定制内存使用和连接限制，可以调整 my.ini 文件中的以下参数：

- 内存使用:`innodb_buffer_pool_size`，`key_buffer_size`，`tmp_table_size`，`max_heap_table_size`
- 连接限制：`max_connections`

（4）在分布式部署时，MySQL 的数据同步有哪些方式？讨论它们的优势和适用场景。

- **主从复制:** 最常用的同步方式之一，适用于读写分离、负载均衡等场景。优点是可以增加读性能，但不能自动解决冲突。
- **半同步复制:** 在主从复制基础上增加了安全性，保证至少一个从库接收到了事务后再确认给客户端，适用于对数据一致性要求较高的场景。
- **组复制（MySQL Group Replication）:** 提供了高可用性和自动故障转移功能，适用于需要高可用性的场景，但它对网络延迟敏感。
- **Galera Cluster:** 一种多主同步复制解决方案，适合于需要多主写入的环境，但其性能可能受到节点数量的影响。

选择哪种方式取决于具体的业务需求，比如是否需要多主写入、对一致性的要求、以及预期的读写比例等。

（5）MySQL 的数据库文件在不同操作系统中存储位置可能有哪些不同？扩展名有哪些？

- Linux：默认情况下，MySQL 的数据文件存储在`/var/lib/mysql/`目录下。每个数据库对应一个子目录，表文件则直接存储在这个子目录内。
- Windows：默认存储路径可能是`C:\ProgramData\MySQL\MySQL Server X.X\Data\`，结构与 Linux 相似。
- MacOS：类似 Linux，具体位置依赖于安装方式，默认可能在`/usr/local/mysql/data/`。

关于文件扩展名：

- .frm：表结构定义文件。
- .MYD：MyISAM 存储引擎的数据文件。
- .MYI：MyISAM 存储引擎的索引文件。
- .ibd:InnoDB 存储引擎的表空间文件,每个表单独一个文件（当使用`innodb_file_per_table`选项时）。
- .opt：数据库级别的选项文件。

5 任务 4：数据表的创建与修改管理实验

5.1 实验目的

1. 掌握表的基础知识。
2. 掌握使用 MySQL Workbench 或其他第三方管理工具和 SQL 语句创建表的方法。
3. 掌握表的修改、查看、删除等基本操作方法。
4. 掌握表中完整性约束的定义。
5. 掌握完整性约束的作用。

5.2 实验内容

(一) 表定义与修改操作在 library 数据库中创建一个 bookInfo 表，表结构如下：按照下列

字段名	字段描述	数据类型	主键	外键	非空	唯一	自增
id	编号	INT(4)	是	否	是	是	是
isbn	国际标准书号	VARCHAR(13)	否	否	是	是	否
title	书名	VARCHAR(100)	否	否	是	否	否
author	作者	VARCHAR(100)	否	否	是	否	否
`publish_date`	出版日期	DATE	否	否	否	否	否
category	分类	VARCHAR(50)	否	否	是	否	否

要求进行表定义操作：

(1) 首先创建数据库 library。（1 分）

创建方法同任务三，在 MySQL Workbench 或其他第三方管理工具中，点击“创建数据库”按钮，输入名称“library”，点击确定。

(2) 创建 bookInfo 表。（1 分）

在右侧栏中点击刚刚创建完成的 library 数据库，右键下拉选项中的 ‘Tables’, 选择 ‘Create Table’，输入表名，按照要求键入每一列的属性以及约束条件，点击确定。

(3) 将 bookInfo 表的 title 字段的数据类型改为 VARCHAR(150)。（1 分）

在 ‘tables’ 栏目下，右键刚刚建好的表 ‘bookInfo’，选择 ‘Alter Table’。弹出的窗口界面与 (2) 相同，按照修改 title 字段的 ‘Datatype’ 为 VARCHAR(150)。点击确定，可以看到 MySQL 自动帮我们生成了 SQL 语句。

(4) 将`publish_date`字段的位置改到 author 字段的前面。（1 分）

同 (3)，在修改界面中右键该字段，选择 ‘Move Up’，将其移动到 author 字段的前面。点击确定，可以看到 MySQL 自动帮我们生成了 SQL 语句。

(5) 将 isbn 字段改名为`book_isbn`。（1 分）

同上，在修改界面双击 isbn 字段，修改名称为`book_isbn`，点击确定，可以看到 MySQL 自动帮我们生成了 SQL 语句。

(6) 将 bookInfo 表的 category 字段删除。（1 分）

同 (3)，在修改界面中右键 ‘category’，选择 ‘Delete selected’，点击确定，可以看到 MySQL 自动帮我们生成了 SQL 语句。

(7) 在 bookInfo 表中增加名为 price 的字段，数据类型为 DECIMAL(8,2)。（1 分）

同上，在修改页面新增一列，输入列名为 ‘price’，数据类型为 DECIMAL(8,2)，点击确定，可以看到 MySQL 自动帮我们生成了 SQL 语句。

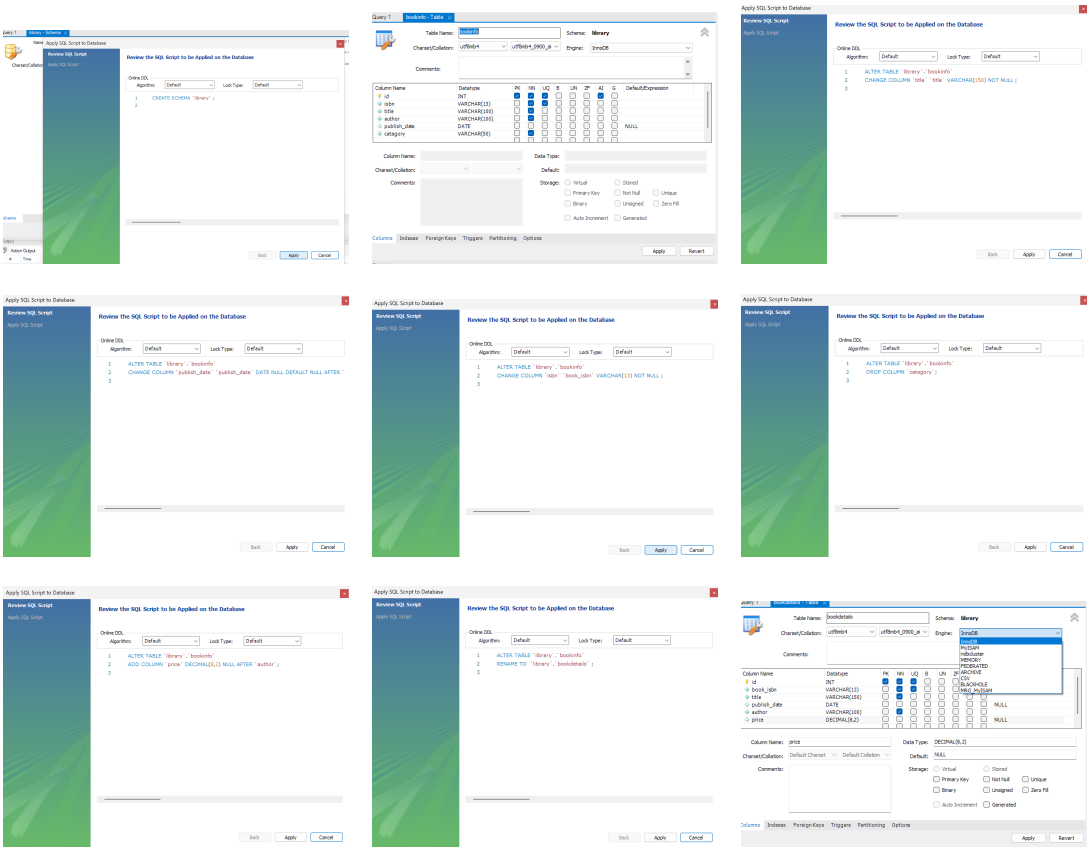
(8) 将 bookInfo 表改名为 bookDetails。（1 分）

同 (3)，在修改界面左上方输入新表名 ‘bookDetails’，可以看到 MySQL 自动帮我们生成了 SQL 语句。

(9) 将 bookDetails 表的存储引擎更改为 InnoDB 类型。（1 分）

在右上方的 ‘Engine’ 下拉框中选择 ‘InnoDB’，点击确定。这里我默认的 Engine 为 InnoDB。

上面操作流程对应的截屏如下：



（二）创建图书馆员工管理数据库 libraryStaff，并定义 department 表和 staff 表，完成两表之间的完整性约束。按照下列要求进行表操作：（基本操作大同小异，这里我在每一个小问里面给出对应的 SQL 语句）

（1）创建 libraryStaff 数据库。（1 分）

```
CREATE DATABASE libraryStaff;
```

（2）创建 department 表，并附上 SQL 语句。（1 分）

Department 表的结构

字段名	字段描述	数据类型	主键	外键	非空	唯一	自增
dept_id	部门号	INT(4)	是	否	是	是	否
dept_name	部门名	VARCHAR(50)	否	否	是	是	否
functions	部门职能	VARCHAR(100)	否	否	否	否	否
location	部门位置	VARCHAR(100)	否	否	否	否	否

Staff 表的结构

字段名	字段描述	数据类型	主键	外键	非空	唯一	自增
id	员工号	INT(4)	是	否	是	是	否
staff_num	员工编号	VARCHAR(10)	否	否	是	是	否
dept_id	部门号	INT(4)	否	是	否	否	否
name	姓名	VARCHAR(50)	否	否	是	否	否
sex	性别	VARCHAR(10)	否	否	否	否	否
birth_date	出生日期	DATE	否	否	否	否	否
address	家庭住址	VARCHAR(100)	否	否	否	否	否

```
USE libraryStaff;
CREATE TABLE department(
    dept_id INT(4) PRIMARY KEY,
    dept_name VARCHAR(50) NOT NULL,
    functions VARCHAR(100),
    loacation VARCHAR(100),
    CONSTRAINT unique_dept UNIQUE(dept_name)
);
```

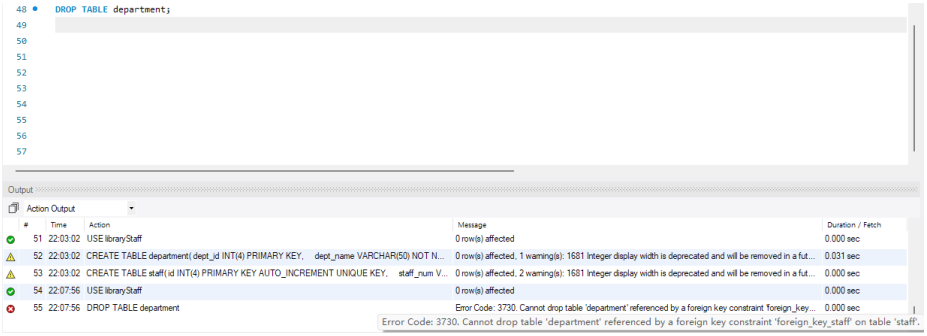
（3）创建 staff 表（注意外键），并附上 SQL 语句。（1 分）

```
USE libraryStaff;
CREATE TABLE staff(
    id INT(4) PRIMARY KEY AUTO_INCREMENT UNIQUE KEY,
    staff_num VARCHAR(10) NOT NULL,
    dept_id INT(4),
    name VARCHAR(50) NOT NULL,
    sex VARCHAR(10) NOT NULL,
    birth_date DATE,
    address VARCHAR(100),
    CONSTRAINT unique_staff UNIQUE(staff_num),
    CONSTRAINT foreign_key_staff
    FOREIGN KEY(dept_id) REFERENCES department(dept_id)
);
```

（4）删除 department 表，并附上 SQL 语句。（1 分）

```
USE libraryStaff;
DROP TABLE department;
```

欸，但是此时我们发现右下方的输出栏提示我们 “Cannot drop table ‘department’ referenced by a foreign key constraint ‘foreign_key_staff’ on the table ‘staff’” 也就是由于外键的存在，删除失败了！说明在删除 department 表之前，我们需要先删除 staff 表的外键约束，否则将导致剩下的 ‘staff’ 表出问题！



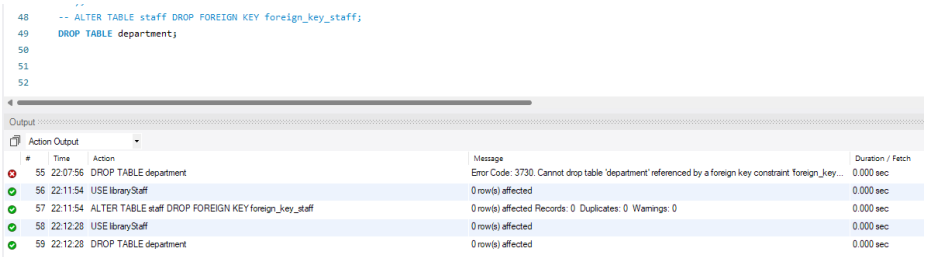
（5）删除 staff 表的外键约束，代码如下：（1 分）

```
USE libraryStaff;
ALTER TABLE staff DROP FOREIGN KEY foreign_key_staff;
```

（6）重新删除 department 表，并附上 SQL 语句。（1 分）

```
USE libraryStaff;
DROP TABLE department;
```

这时我们发现，右下方输出信息提示成功删除！



5.3 观察与思考

1、关于 NOT NULL

(1) 在定义基本表语句时，NOT NULL 参数的作用是什么？（1 分）

NOT NULL 参数的作用是约束该字段不能为 NULL，即该字段的值必须非空。在定义某字段为 NOT NULL 时，如果更新或者插入的目标会使某一行的该字段为空数据库会报错并拒绝执行该操作。这时常见的数据完整性约束之一。

(2) 主码列修改成允许 NULL 能否操作？为什么？（2 分）

不允许。主码（Primary Key）是用来唯一标识表中每一行记录的字段或字段组合。主码的两大特性是唯一性和非空性就保证了主码一定不能为 NULL，否则会导致数据库出现错误。

2、关于外码

(1) 根据下面设计的表结构，Staff 表的外键能否设置成功？思考外码设置需要注意哪些问题？（3 分）

不能成功，因为`dept_no`字段在 Staff 表中定义时并未被设置成主键，而外键的定义依赖于主键。

Department 表的结构

字段名	字段描述	数据类型	主键	外键
`dept_no`	部门号	INT(4)		
`dept_name`	部门名	VARCHAR(50)		

Staff 表的结构

字段名	字段描述	数据类型	主键	外键
`staff_no`	员工编号	VARCHAR(10)		
`dept_no`	所在部门号	INT(4)		√
name	姓名	VARCHAR(50)		

外码的设置需要注意以下几点：

1. 是依赖于数据库中已存在的表的主键。
2. 数据类型和长度必须匹配：外键列与目标列的数据类型、长度、字符集等必须完全一致。
3. 外键列不能有主表的目标列中没有的值。
4. 注意设置级联操作以保证数据一致性

(2) 如果主表无数据，从表的数据能输入吗？（1 分）

不行。外键约束要求从表的外键列中的值必须存在于主表的目标列中。如果主表中没有数据，从表的外键列就无法引用有效的值，因此插入操作会失败。但是在从表该列未被设置为 NOT NULL 时，可以输入 NULL 值。

(3) 先创建从表，再创建主表是否可以？（1 分）

不可以。在创建从表时，如果设置了外键约束，数据库会检查主表是否存在以及主表的目标列是否有效。如果主表尚未创建，从表的外键约束将无法验证，导致创建失败。

3、关于主码和唯一约束

(1) 唯一约束列是否允许 NULL 值？（1 分）

唯一约束列允许 NULL 值，并且在同一个表格中可以有多个行为 NULL 值。但该字段不为 NULL 的行之间的值一定不相同。

(2) 一张表可以设置几个主码，可以设置几个唯一约束？（1 分）

一张表只能设置一个主码，但可以设置多个唯一约束。