

# 数据库原理及应用课程作业 #2

## 思政资源数据库创建与数据管理操作

杨睿妮 (2018011205014)  
yangruinii@foxmail.com

电子科技大学 — May 29, 2021

### 1 实验说明

- 采用 SQL 定义数据库模式以及操作数据和控制数据、采用数据操作语言 DML 对预备的思政素材进行增删改查、采用数据控制语言 DCL 进行安全权限许可控制
- 完成树状学科目录的前端展示

### 2 实验原理

SQL 的工作机理见图 1

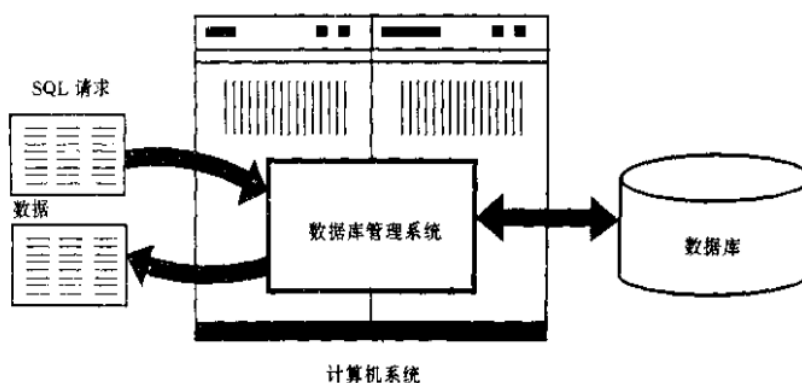


Figure 1: SQL 工作机理

SQL 的工作过程见图 2

### 3 实验环境

实验环境如表 1 所示。

### 4 实验步骤

#### 4.1 利用 Power Designer 的自动模式创建功能构建思政资源数据库

利用 Power Designer 的自动模式生成了lab\_1.sql脚本，将其导入 SQL Server 中，结果见图 3，具体的 SQL 语句见附录 A。

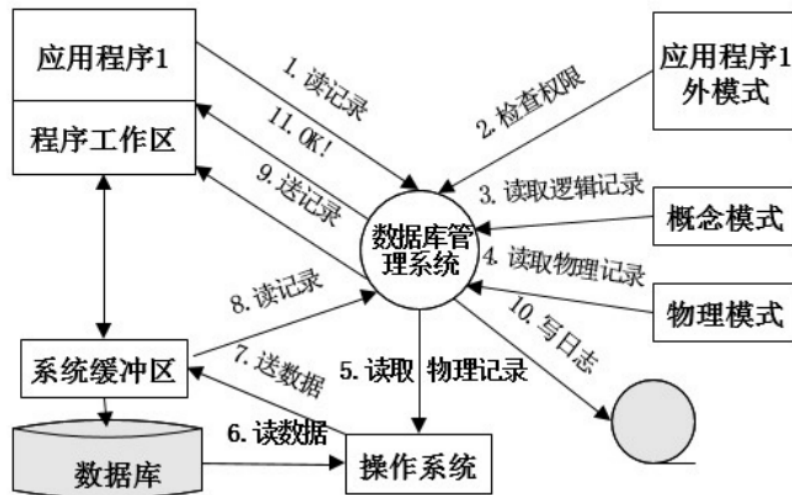


Figure 2: SQL 工作过程

Table 1: 实验环境

处理器	Intel(R) Core(TM) i5-8300H CPU @2.30GHz
内存	16GB
软件	Sybase PowerDesigner Version 16.5; SQL Server 2012
操作系统	Windows 10
语言	Python 3.8.5; HTML; JavaScript; CSS
框架	Flask 1.1.2

#### 4.2 采用 SQL 定义各个实体模式上的实体完整性、参照完整性和自定义完整性

在实验一建立数据库中时，已经对相关的属性进行了约束：满足关系的主码不能取空值；参照关系中每个元素的外码要么为空，要么等于被参照关系中某个元素的主码；对关系中每个属性的取值的限制做了具体定义。在后续的数据增删查改中也满足了三种完整性。

#### 4.3 采用 SQL 语句创建视图，为思政资源素材检索提供外模式（用户模式）

在本次实验中，建立了名为 View\_Course\_Type 的视图，功能为选择出 type = '1' 的思政素材，单独组成一个视图，提供外模式。使用的语句见 create-view.sql。

```
create-view.sql

create view View_Course_Type
as
    select * from dbo.CourseItem where type = '1'
go

select * from View_Course_Type
```

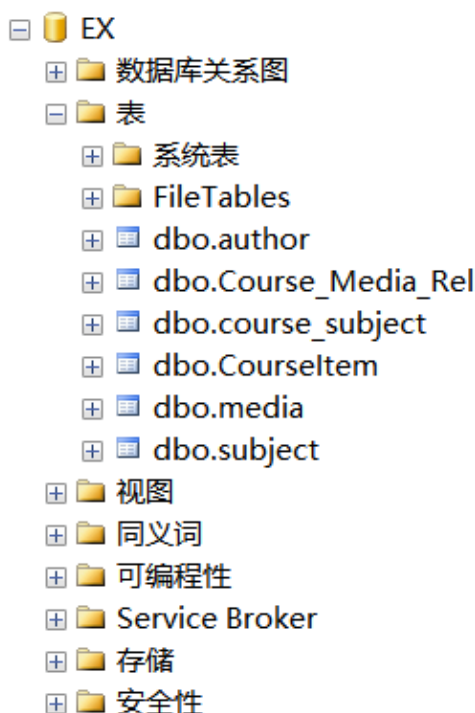


Figure 3: 数据库创建结果

#### 4.4 采用 SQL 在实体模式上创建单属性索引, 在视图上建立复合索引

为了提升思政资源素材的检索效率,在dbo.CourseItem上为keyword建立名为CourseKeyword的索引, 提高用户根据关键词查找素材的效率, 具体语句见create-index.sql, 结果见 4。

```
create-index.sql

CREATE INDEX CourseKeyword
ON dbo.CourseItem (keyword)
```

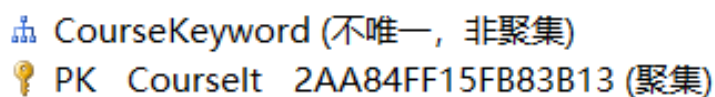


Figure 4: CourseKeyword 索引

为了提升思政资源素材在混合属性上的检索效率, 我们利用WITH SCHEMABINDING关键字建立新的视图, 以便于后续在该视图上建立复合索引。WITH SCHEMABINDING是防止视图所引用的表在视图未被调整的情况下发生改变的选项。

建立了名为View\_For\_Index的视图, 选择出表dbo.CourseItem中type为“1”的实体。并且在该视图上, 基于列author和keyword建立名为IDX\_V1的索引, 便于用户基于作者以及关键字查找思政素材。具体语句见create-view-index.sql, 结果见 5。

```
create-view-index.sql

create view View_For_Index
WITH SCHEMABINDING
AS
select courseDesc, courseID, keyword, author, source
from dbo.CourseItem where type = '1'
go

CREATE UNIQUE CLUSTERED INDEX IDX_V1
ON View_For_Index(author, keyword);
GO
```



Figure 5: 在视图上的复合索引” IDX\_V1”

#### 4.5 采用 Insert 语句将采编的思政资源素材录入到数据库中

插入采编者信息，采用的语句见insert-author.sql，具体结果见 6。

```
insert-author.sql

insert into dbo.author(authorID, authorName) values(1, 'Ruini Yang');
insert into dbo.author(authorID, authorName) values(2, 'Yifei Zhang');
insert into dbo.author(authorID, authorName) values(3, 'Culaccino');
insert into dbo.author(authorID, authorName) values(4, 'Komorebi');
insert into dbo.author(authorID, authorName) values(5, 'Goya');
insert into dbo.author(authorID, authorName) values(6, 'Ternura');
insert into dbo.author(authorID, authorName) values(7, 'Alkaid');
// ...
```

插入学科信息，采用的语句见insert-subject.sql，具体结果见 7。

	authorID	authorName
1	7	Alkaid
2	3	Culaccino
3	5	Goya
4	4	Komorebi
5	1	Ruini Yang
6	6	Ternura
7	2	Yifei Zhang

Figure 6: 插入采编者

insert-subject.sql

```
insert into dbo.subject(subjectID, subjectName, parentID) values
(1,'基础学科', '0');
insert into dbo.subject(subjectID, subjectName, parentID) values
(2,'信息科技', '0');
insert into dbo.subject(subjectID, subjectName, parentID) values
(3,'医药卫生科技', '0');
insert into dbo.subject(subjectID, subjectName, parentID) values
(4,'哲学与人文科学', '0');
insert into dbo.subject(subjectID, subjectName, parentID) values
(5,'基础化学', '1');
// ...
```

	subjectID	subjectName	parentID
1	1	基础学科	0
2	2	信息科技	0
3	3	医药卫生科技	0
4	4	哲学与人文科学	0
5	5	基础化学	1

Figure 7: 插入学科信息

插入思政素材，采用的语句见insert-course.sql，具体结果见 8。

```
insert-course.sql

insert into dbo.CourseItem(courseID, authorID, courseDesc, courseName,
keyword, author, source, type, cnt) values(1, 8,'维护安全问题',
'Linux操作系统维护技术探析', 'Linux系统', '方木龙',
'中移互联网有限公司', '1', 0);
insert into dbo.CourseItem(courseID, authorID, courseDesc, courseName,
keyword, author, source, type, cnt) values(2, 9,'非易失主存构建存储系统
', '非易失主存的系统软件研究进展', '非易失主存', '舒继武', '中国科学：
信息科学', '2', 0);
insert into dbo.CourseItem(courseID, authorID, courseDesc, courseName,
keyword, author, source, type, cnt) values(3, 10,'混合教学模式',
'混合教学模式在“操作系统”中的研究和实践', '混合模式', '杨婷婷',
'无线互联科技', '1', 0);
insert into dbo.CourseItem(courseID, authorID, courseDesc, courseName,
keyword, author, source, type, cnt) values(4, 11,'课程教学内容重组',
'基于 O B E 理念的操作系统原理课程教学内容重组', '操作系统', '张秋红',
'兰州文理学院学报', '1', 0);
// ...
```

	courseID	authorID	courseDesc	courseName	keyword	author	source	type	cnt
1	1	8	维护安全问题	Linux操作系统维护技术探析	Linux系统	方木龙	中移互联网有限公司	1	3
2	2	9	非易失主存构建存储系统	非易失主存的系统软件研究进展	非易失主存	舒继武	中国科学：信息科学	2	1
3	3	10	混合教学模式	混合教学模式在“操作系统”中的研究和实践	混合模式	杨婷婷	无线互联科技	1	2
4	4	11	课程教学内容重组	基于 O B E 理念的操作系统原理课程教学内容重组	操作系统	张秋红	兰州文理学院学报	1	2

Figure 8: 插入思政素材

插入思政素材与学科之间的多对多关系，采用的语句见insert-rel.sql，具体结果见可以体现在前端界面显示上，点击学科可以显示出相应的思政素材。

```
insert-rel.sql

insert into dbo.course_subject(courseID, subjectID) values(1, 11)
insert into dbo.course_subject(courseID, subjectID) values(2, 11)
insert into dbo.course_subject(courseID, subjectID) values(3, 11)
insert into dbo.course_subject(courseID, subjectID) values(4, 11)
// ...
```

插入多媒体素材，采用的语句见insert-media.sql，具体结果见 9。

```
insert-media.sql

insert into dbo.media(mediaID, authorID,mediaName,url) values(1,2,
'王道考研-操作系统','https://www.bilibili.com/video/BV1YE411D7nHfrom
=search
&seid=13181938279769288528')
insert into dbo.media(mediaID, authorID,mediaName,url) values(2,2,
'谭浩强老师C语言教程程序设计','https://www.bilibili.com/video/
BV1VW41187zq?from=search&seid=12389279456804192620')
insert into dbo.media(mediaID, authorID,mediaName,url) values(3,1,
'计算机软件技术基础','https://www.bilibili.com/video/BV1xa4y1j7ZN?
from=search&seid=1409768092267906681')
// ...
```

	mediaID	authorID	mediaName	url
1	1	2	王道考研-操作系统	https://www.bilibili.com/video/BV1YE411D7nHfrom...
2	2	2	谭浩强老师C语言教程程序设计	https://www.bilibili.com/video/BV1VW41187zq?fro...
3	3	1	计算机软件技术基础	https://www.bilibili.com/video/BV1xa4y1j7ZN?fro...

Figure 9: 插入多媒体素材

#### 4.6 采用 Update 语句更新思政资源素材的点击阅读次数

以将每个type为”1”的思政素材点击次数加 1 为例，具体语句见update-cnt.sql，结果见图 10。

```
update-cnt.sql

update dbo.CourseItem set cnt = cnt + 1
where type = '1'
```

author	source	type	cnt
方木龙	中移互联网有限公司	1	1
杨婷婷	无线互联科技	1	1
张秋红	兰州文理学院学报	1	1
应沈静	科技风	1	1
衷宜	黑龙江教育	1	1
刘婷	成人教育	1	1
智爱娟	电脑知识与技术	1	1
黄驰	中国管理信息化	1	1

Figure 10: 更新操作

#### 4.7 采用 Select 语句实现思政资源素材的关键词模糊检索

以对关键词中含有“系统”进行模糊检索为例，具体语句见select.sql，结果见图 11。

```
select.sql

select * from dbo.CourseItem
where keyword like '%系统%'
```

	courseID	authorID	courseDesc	courseName	keyword	author	source	type	cnt
1	1	8	维护安全问题	Linux操作系统维护技术探析	Linux系统	方木龙	中移互联网有限公司	1	1
2	4	11	课程教学内容重组	基于 OBE理念的操作系统原理课程教学内容重组	操作系统	张秋红	兰州文理学院学报	1	1
3	5	12	自主可控操作系统	自主可控操作系统的探索与研究	操作系统	程宇	技术	2	0

Figure 11: 检索操作

#### 4.8 采用 Select 语句实现思政资源素材的分类别、分目录按点击阅读次数降序统计

以将一级目录（即没有上级学科，parentID = 0）中，且type = 1的学科按点击阅读次数降序统计。具体语句见desc.sql，结果见图 12。

```
desc.sql

select subject.parentID, cnt, CourseItem.courseName
from dbo.subject, dbo.CourseItem, dbo.course_subject
where CourseItem.courseID = course_subject.courseID
and subject.parentID = 0
group by subject.parentID, cnt, CourseItem.courseName
order by cnt desc
```

	parentID	cnt	courseName
1	0	3	Linux操作系统维护技术探析
2	0	2	混合教学模式在“操作系统”中的研究和实践
3	0	2	基于 OBE理念的操作系统原理课程教学内容重组
4	0	1	《独立学院汇编语言程序设计课程教学研究与探索
5	0	1	《计算机组成原理》课程中“DMA方式”知识点的教学设计
6	0	1	H1806BM2 微处理器指令系统分析破译研究
7	0	1	OBE理念下药用基础化学教学效果评价研究
8	0	1	翻转课堂在基础物理专业实验教学中的应用研究
9	0	1	非易失主存的系统软件研究进展
10	0	1	基于CNKI数据库和CiteSpace突变词分析
11	0	1	基于STM32微处理器的人眼虹膜识别系统设计
12	0	1	基于遍历求二叉树的程序设计与探讨

Figure 12: 降序统计

#### 4.9 采用 Grant 语句为一组用户授予不同的增删改查权限许可

创建三个登录账号以及对应的数据库用户，具体语句见create-login.sql。



create-login.sql

```
create login dba1 with password = '123', default_database = EX
create user Wang for login dba1 with default_schema = dbo
create login dba2 with password = '123', default_database = EX
create user Li for login dba2 with default_schema = dbo
create login dba3 with password = '123', default_database = EX
create user Zhang for login dba3 with default_schema = dbo
```

赋予三个用户不同的权限，具体语句见grant.sql。

grant.sql

```
grant select on CourseItem to Wang
grant insert on author to Li
grant update(author, type) on CourseItem to Zhang
```

登录dba1账号，输入select \* from dbo.语句，在EX数据库中只会出现CourseItem的提示，与我们之前设置的权限相符合。见图 13。

```
select * from dbo.
```

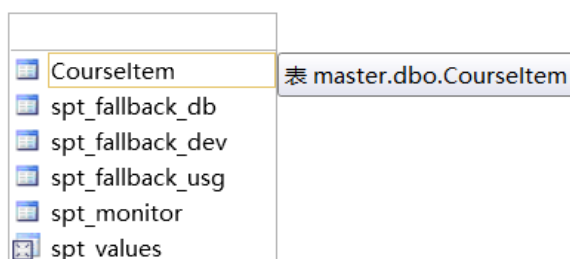


Figure 13: dba1 权限

登录dba2账号，输入select \* from dbo.author语句，会报错。这是由于dba2对于表author只有insert权限而没有select的权限。结果见图 14。

登录dba3账号，输入update dbo.CourseItem set type = '2' where author = '方木龙'语句，但是会报错“拒绝了对对象‘CourseItem’（数据库‘master’，架构‘dbo’）的 SELECT 权限。”通过sa用户对dba3用户用语句grant select on dbo.CourseItem to Zhang赋予select权限，再次执行该指令不再报错。

## 5 Flask 框架实现数据库操作

本次实验后端对数据库的操作采用了Flask框架中调用pyodbc库对数据库数据进行调取与操作。

以@app.route('/user/get/<int:id>')路由为例，实现了在表author中的select操作。具体代码如下：

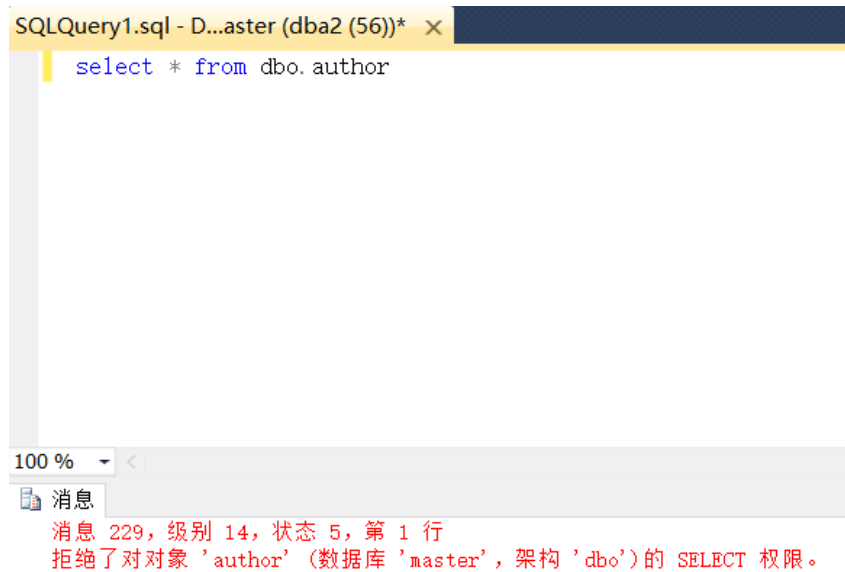
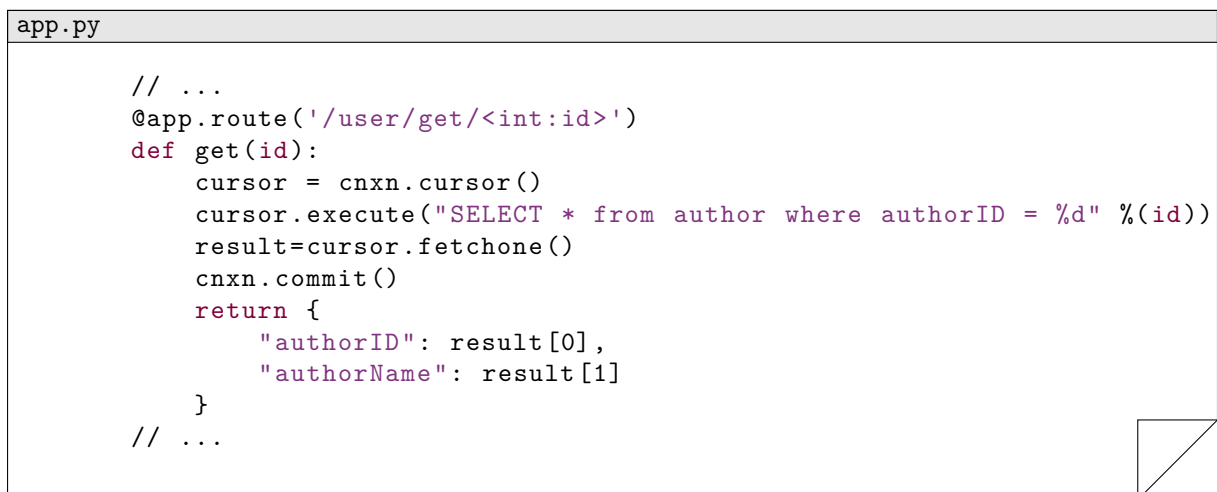


Figure 14: dba2 权限



在前端发送请求结果见图 15。

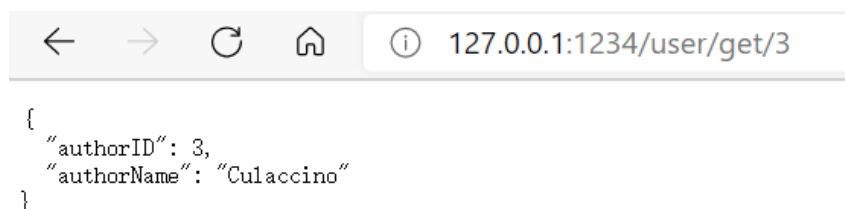


Figure 15: 在 author 表中检索 id 为 3 的采编者并且返回 name

以`@app.route('/user/add', methods=['GET', 'POST'])`路由为例,实现了在表author中添加操作。终端采用命令和具体代码如下,结果见图 16。

## Command Line

```
$ curl -d '{"name": "Karina", "ID": 40}' -H 'Content-Type:application/json' http://localhost:1234/user/add
```

## app.py

```
@app.route('/user/add', methods=['GET', 'POST'])
def add():
    if request.method == 'POST':
        data = request.get_json()
        cursor = cnxn.cursor()
        name = data['name']
        ID = data['ID']
        cursor.execute("insert into author(authorID, authorName)
            values(%d, '%s')" % (ID, name))
        cursor.execute("SELECT * from author")
        result=cursor.fetchall()
        cnxn.commit()
        for row in result:
            print(row)
        return {
            "ok": "true"
        }
```

	authorID	authorName
1	7	Alkaid
2	39	Ariana
3	3	Culaccino
4	5	Goya
5	40	Karina
6	4	Komorebi

Figure 16: 在 author 表中添加采编者成功

## 6 前端网页展示

为了完成在前端展示学科的树状模型，以及点击相关学科可以展示出与该学科对应的思政素材，在app.py中，设置了@app.route("/subsubject/side-item-name-<int:id>")路由完成各级科学目录之下的学科获取，以完成网页侧边栏的树状课程目录展示；设置了@app.route("/course/<int:id>")路由完成了对应学科下的思政素材的获取功能，并且结合 JavaScript 代码将其展示在网页上。以下是网页的展示，具体的前端代码index.js; styles.css; index.html, 以及后端Flask代码app.py代码详见附录。

网页的主页见图 17。

点击学科，即可展示对应的思政素材，见图 18。

侧边栏点开，即可展示学科的树状结构见图 19。



Figure 17: 网页主页



Figure 18: 展示对应的思政素材

## 7 思考题

**Q: 如何在 Power Designer 中自动生成数据库创建脚本 SQL Script?**

点击生成的物理模型 → 上方的Database选项 → 点击Generate DataBase

**Q: 如何在 Power Designer 中自动生成数据库设计报告?**

选中想要生成报告的模型 → 上方的Report选项 → 点击Generate Report

**Q: 如何实现批量思政资源数据的加载?**

采用拼接BatchInsert插入语句或者采用SqlBulkCopy插入方案。

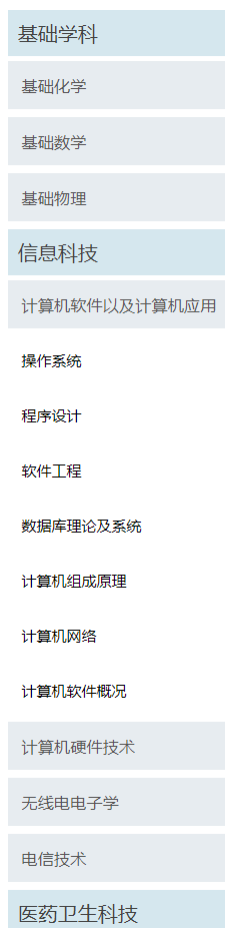


Figure 19: 侧边栏与学科树状结构

## 8 实验总结

本次实验完成了思政资源数据库创建与数据管理操作。在本次实验中，通过完成各种实验要求的操作，逐步熟练运用各种 SQL 语句。其次，为了完成前端界面的编写，去学习了Flack框架的相关知识，同时也进行了前端编写的学习。虽然由于时间原因最终完成的网页不够美观，排版也可以更加完善，但是在这期间学习到了很多新的知识。

以下是一些本次实验中可以继续完善的地方：

- 虽然在Flask框架中利用路由，可以实现数据的增删改查操作，但是还不能在前端进行可视化的操作。后续可以继续完善登录功能，根据不同用户的权限完成不同的操作。
- 网站只能完成素材展示的功能，下载功能还未实现。后续会学习有关POST请求的知识，完善网站功能。
- 多媒体数据的存储，目前实现的方式是在数据库中直接存放素材的地址，但是还未能能在网站上完成展示。

## A 附录: lab\_1.sql

```
1  /*=====*/
2  /* DBMS name:      MySQL 5.0                      */
3  /* Created on:      2021/5/6 21:39:22              */
4  /*=====*/
5
6
7  if exists (select 1
8      from sys.sysreferences r join sys.sysobjects o on (o.id =
9  r.constid and o.type = 'F')
10     where r.fkeyid = object_id('CourseItem') and o.name =
11     'FK_author_course')
12  alter table CourseItem
13     drop constraint FK_author_course
14  go
15
16  if exists (select 1
17      from sys.sysreferences r join sys.sysobjects o on (o.id =
18  r.constid and o.type = 'F')
19     where r.fkeyid = object_id('media') and o.name =
20     'FK_author_media')
21  alter table media
22     drop constraint FK_author_media
23  go
24
25  if exists (select 1
26      from sys.sysreferences r join sys.sysobjects o on (o.id =
27  r.constid and o.type = 'F')
28     where r.fkeyid = object_id('Course_Media_Rel') and o.name =
29     'FK_Course_Media_Rel')
30  alter table Course_Media_Rel
31     drop constraint FK_Course_Media_Rel
32  go
33
34  if exists (select 1
35      from sys.sysreferences r join sys.sysobjects o on (o.id =
36  r.constid and o.type = 'F')
37     where r.fkeyid = object_id('Course_Media_Rel') and o.name =
38     'FK_Course_Media_Rel2')
39  alter table Course_Media_Rel
40     drop constraint FK_Course_Media_Rel2
41  go
42
43  if exists (select 1
44      from sys.sysreferences r join sys.sysobjects o on (o.id =
45  r.constid and o.type = 'F')
46     where r.fkeyid = object_id('course_subject') and o.name =
47     'FK_course_subject')
48  alter table course_subject
```

```

49     drop constraint FK_course_subject
50 go
51
52
53 IF OBJECT_ID(N'CourseItem', N'U') IS NOT NULL
54 drop table CourseItem;
55
56 IF OBJECT_ID(N'Course_Media_Rel', N'U') IS NOT NULL
57 drop table Course_Media_Rel;
58
59 IF OBJECT_ID(N'author', N'U') IS NOT NULL
60 drop table author;
61
62 IF OBJECT_ID(N'course_subject', N'U') IS NOT NULL
63 drop table course_subject;
64
65 IF OBJECT_ID(N'media', N'U') IS NOT NULL
66 drop table media;
67
68 IF OBJECT_ID(N'subject', N'U') IS NOT NULL
69 drop table subject;
70
71 /*=====*/
72 /* Table: CourseItem */
73 /*=====*/
74 create table CourseItem
75 (
76     courseID          integer not null,
77     authorID          integer,
78     courseDesc        varchar(1024),
79     courseName        varchar(1024),
80     keyword           varchar(1024),
81     author            varchar(1024),
82     source            varchar(1024),
83     type              varchar(10),
84     cnt               integer,
85     primary key (courseID)
86 );
87
88 /*=====*/
89 /* Table: Course_Media_Rel */
90 /*=====*/
91 create table Course_Media_Rel
92 (
93     courseID          integer not null,
94     mediaID           integer not null,
95     primary key (courseID, mediaID)
96 );
97
98 /*=====*/

```

```

99  /* Table: author                                                    */
100 /*=====*/
101 create table author
102 (
103     authorID            integer not null,
104     authorName          varchar(1024) not null,
105     primary key (authorID)
106 );
107
108 /*=====*/
109 /* Table: course_subject                                             */
110 /*=====*/
111 create table course_subject
112 (
113     courseID            integer not null,
114     subjectID           integer not null,
115     primary key (courseID, subjectID)
116 );
117
118 /*=====*/
119 /* Table: media                                                       */
120 /*=====*/
121 create table media
122 (
123     mediaID             integer not null,
124     authorID            integer,
125     mediaName           varchar(1024),
126     url                 VARCHAR(2083),
127     primary key (mediaID)
128 );
129
130 /*=====*/
131 /* Table: subject                                                     */
132 /*=====*/
133 create table subject
134 (
135     subjectID           integer not null,
136     subjectName         varchar(1024) not null,
137     parentID            integer,
138     primary key (subjectID)
139 );
140
141 alter table CourseItem add constraint FK_author_course foreign key
142 (authorID)
143     references author (authorID) on delete CASCADE on update
144     CASCADE;
145
146 alter table Course_Media_Rel add constraint FK_Course_Media_Rel
147 foreign key (courseID)
148     references CourseItem (courseID) on delete CASCADE on update

```



```

149 CASCADE;
150
151 alter table Course_Media_Rel add constraint FK_Course_Media_Rel2
152 foreign key (mediaID)
153     references media (mediaID) on delete CASCADE on update
154 CASCADE;
155
156 alter table course_subject add constraint FK_course_subject foreign key
157 (courseID)
158     references CourseItem (courseID) on delete CASCADE on update
159 CASCADE;
160
161 alter table course_subject add constraint FK_course_subject2 foreign key
162 (subjectID)
163     references subject (subjectID) on delete CASCADE on update
164 CASCADE;
165
166 alter table media add constraint FK_author_media foreign key (authorID)
167     references author (authorID) on delete NO ACTION on update
168 NO ACTION;

```

## B 附录: app.py

```
1 import pyodbc
2 from flask import Flask, jsonify, request, abort, render_template
3 import json
4 app = Flask(__name__, static_folder='./template/static',
5 template_folder='./template')
6
7 server = 'DESKTOP-BBKEC14\\SQLEXPRESS'
8 database = 'EX'
9 username = 'sa'
10 password = '123'
11 cnxn = pyodbc.connect('DRIVER={SQL Server Native Client 11.0};SERVER=' +
12 server+';DATABASE='+database+';UID='+username+';
13 PWD=' + password)
14
15 @app.route('/', defaults={'path': ''})
16 @app.route('/<path:path>')
17 def catch_all(path):
18     return render_template('index.html')
19
20 @app.route("/subject")
21 def home():
22     cursor = cnxn.cursor()
23     cursor.execute("select subjectName, subjectID from dbo.subject where
24 parentID = 0")
25     result=cursor.fetchall()
26     cnxn.commit()
27     return jsonify([{'name': r[0], 'id': r[1] } for r in result])
28
29 @app.route("/subsubject/side-item-name-<int:id>")
30 def subhome(id):
31     cursor = cnxn.cursor()
32     cursor.execute("select subjectName, subjectID from dbo.subject
33 where parentID = %d" %(id))
34     result=cursor.fetchall()
35     cnxn.commit()
36     for row in result:
37         print(row)
38     return jsonify([{'name': r[0], 'id': r[1] } for r in result])
39
40 @app.route("/course/<int:id>")
41 def course(id):
42     cursor = cnxn.cursor()
43     cursor.execute("select courseID from dbo.course_subject where
44 subjectID = %d" %(id))
45     result1=cursor.fetchall()
46     finalres = []
47     for r in result1:
48         cursor.execute("select courseID, courseName, author, source
```

```

49         from dbo.CourseItem where courseID = %d" %(r[0]))
50         tempres = cursor.fetchone()
51         finalres.append({'id':tempres[0], 'name':tempres[1], 'author':
52             tempres[2], 'src':tempres[3]})
53     cnxn.commit()
54     for row in result1:
55         print(row)
56     return jsonify(finalres)
57
58 @app.route("/subsubsubject/sub-item-name-<int:id>")
59 def subsubhome(id):
60     cursor = cnxn.cursor()
61     cursor.execute("select subjectName, subjectID from dbo.subject where
62         parentID = %d" %(id))
63     result=cursor.fetchall()
64     cnxn.commit()
65     for row in result:
66         print(row)
67     return jsonify([{'name': r[0], 'id': r[1] } for r in result])
68
69 @app.route('/user/get/<int:id>')
70 def get(id):
71     cursor = cnxn.cursor()
72     cursor.execute("SELECT * from author where authorID = %d" %(id))
73     result=cursor.fetchone()
74     cnxn.commit()
75     return {
76         "authorID": result[0],
77         "authorName": result[1]
78     }
79
80 @app.route('/user/del/<int:id>')
81 def dele(id):
82     cursor = cnxn.cursor()
83     cursor.execute("DELETE from author where authorID = %d" %(id))
84     cursor.execute("SELECT * from author")
85     result=cursor.fetchall()
86     cnxn.commit()
87     for row in result:
88         print(row)
89     return "Delect successfully!"
90
91 # @app.route('/post-sum', methods=['GET', 'POST'])
92 # def index():
93 #     if request.method == 'POST':
94 #         data = request.get_json()
95 #         return {
96 #             "res": data['x'] + data['y']
97 #         }
98

```

```

99 @app.route('/user/add', methods=['GET', 'POST'])
100 def add():
101     if request.method == 'POST':
102         data = request.get_json()
103         cursor = cnxn.cursor()
104         name = data['name']
105         ID = data['ID']
106         cursor.execute("insert into author(authorID, authorName)
107             values(%d, '%s')" % (ID, name))
108         cursor.execute("SELECT * from author")
109         result=cursor.fetchall()
110         cnxn.commit()
111         for row in result:
112             print(row)
113         return {
114             "ok": "true"
115         }
116
117 @app.route('/user/update', methods=['GET', 'POST'])
118 def update():
119     if request.method == 'POST':
120         data = request.get_json()
121         cursor = cnxn.cursor()
122         name = data['name']
123         ID = data['ID']
124         cursor.execute("UPDATE author SET authorID = '%d' WHERE authorName
125             = '%s'" % (ID, name))
126         cursor.execute("SELECT * from author")
127         result=cursor.fetchall()
128         cnxn.commit()
129         for row in result:
130             print(row)
131         return {
132             "ok": "true"
133         }
134
135 app.run(host='127.0.0.1', port=1234, debug=True)

```

## C 附录: index.html

```
1 <!doctype html>
2 <html lang="en">
3
4 <head>
5     <link rel="stylesheet" href="./static/styles.css">
6     <meta charset="utf-8">
7     <title>Getting started with CSS</title>
8 </head>
9
10 <body>
11
12     <ul class="topmenu">
13         <!-- <li><a href="#基础科学" class = "active">基础科学</a></li>
14         <li><a href="#信息科学">信息科学</a></li>
15         <li><a href="#医药卫生科技">医药卫生科技</a></li>
16         <li><a href="#哲学与人文科学">哲学与人文科学</a></li> -->
17     </ul>
18
19     <div class = "clearfix">
20         <div class = "column sidemenu">
21             <ul class = "sidemenu">
22                 <!--
23                     <a class="item-name" onclick="loadAndRenderSubItem
24                     (event)">随便什么</a>
25                     <div class="item-content">...</div>
26
27                     <a href="#信息科学">信息科学</a>
28                     <a href="#医药卫生科技">医药卫生科技</a>
29                     <a href="#哲学与人文科学">哲学与人文科学</a> -->
30             </ul>
31         </div>
32
33         <div class = "column content">
34             <div class = "header">
35                 <h1>课程思政素材中心</h1>
36             </div>
37             <h1>简介</h1>
38             <p>这是一个收集课程思政素材的网站，请点击左侧导航选择想查看
39             的课程对应的思政素材。</p>
40             <p class = "tips">以下是查找到的思政素材：</p>
41             <div class = "courselist"></div>
42         </div>
43     </div>
44
45     <div class="footer">
46         <li><b>Contact Me</b></li>
47         <li>Email: <a class = "color1" href="#2333333@qq.com">2333333@qq.com</a>
48         </li>
```

```
49         <li>Github: <a class = "color1" href="#https://github.com/KKKirino">
50             https://github.com/KKKirino</a></li>
51     </div>
52 <script src="./static/index.js"></script>
53 </body>
54
55 </html>
```

## D 附录: index.js

```
1 function post(url, postData = {}) {
2     return fetch({
3         url,
4         method: 'POST',
5         headers: {
6             'Content-Type': 'application/json'
7         },
8         body: JSON.stringify(postData)
9     }).then(res => res.json())
10 }
11
12 function get(url) {
13     return fetch(url).then(res => res.json())
14 }
15
16 function renderHeaderItem(data = {}) {
17     let headerTemplate = '<li id="header-item-$id"><a href="#$id" class =
18 "$isActive">$name</a></li>'
19     Object.keys(data).forEach(key => {
20         headerTemplate = headerTemplate.replaceAll('$' + key, data[key])
21     })
22     return headerTemplate
23 }
24
25
26 /**
27  * 渲染头部列表
28  */
29
30 function loadAndRenderHeader() {
31     // 获取数据并渲染
32     get('/subject').then(subjects => {
33         const header = document.querySelector('.topmenu')
34         const headerContent = subjects.map(renderHeaderItem).join('\n')
35         header.innerHTML = headerContent
36         console.log('[header] loaded!', headerContent)
37     })
38 }
39
40
41 function renderSiderItem(data = {}) {
42     let sideTemplate = '<a class="side-item-name-$id" onclick="
43 loadAndRenderCourseItem(event);loadAndRenderSubItem(event);">$name</a>' +
44 '\n' + '<div class="item-content-$id"></div>'
45     Object.keys(data).forEach(key => {
46         sideTemplate = sideTemplate.replaceAll('$' + key, data[key])
47     })
48     return sideTemplate
49 }
```

```

49 }
50
51 /**
52  * 渲染旁边列表
53  */
54
55 function loadAndRenderSide() {
56     // 获取数据并渲染
57     get('/subject').then(subjects => {
58         const sider = document.querySelector('.sidemenu')
59         const siderContent = subjects.map(renderSiderItem).join('\n')
60         sider.innerHTML = siderContent
61         console.log('[sider] loaded!', siderContent)
62     })
63 }
64
65 function renderSubSiderItem(data = {}) {
66     let subsideTemplate = '<div class="sub-item-name-$id" onclick=
67 "loadAndRenderCourseItem(event);loadAndRenderSubSubItem(event);">$name</div>
68 + '\n' + '<div class="sub-sub-content-name-$id"></div>'
69     Object.keys(data).forEach(key => {
70         subsideTemplate = subsideTemplate.replaceAll('$' + key, data[key])
71     })
72     return subsideTemplate
73 }
74
75 function loadAndRenderSubItem(evt) {
76     const dom = evt.target
77     var cont = dom.className;
78     console.log(dom.dataset)
79     console.log(cont)
80     const targetEle = dom.nextElementSibling
81     get('/subsubject/' + cont).then(subjects => {
82         // const subsider = document.querySelector('.' + cont)
83         const subsiderContent = subjects.map(renderSubSiderItem).join('\n')
84         targetEle.innerHTML = subsiderContent
85         console.log('[subsider] loaded!', subsiderContent)
86     })
87
88     // targetEle.innerHTML = new Array(10).fill(0).map(v => '<div class=
89 "sub-item">我是子类</div>').join('\n')
90 }
91
92 function renderSubSubSiderItem(data = {}) {
93     let subsubsideTemplate = '<div class="sub-sub-item-name-$id" onclick=
94 "loadAndRenderCourseItem(event)">$name</div>' + '\n' + '<div class=
95 "sub-sub-sub-item-name-$id"></div>'
96     Object.keys(data).forEach(key => {
97         subsubsideTemplate = subsubsideTemplate.replaceAll('$' + key, data[key])
98     })

```



```

99     return subsubsideTemplate
100 }
101
102 function loadAndRenderSubSubItem(evt) {
103     const dom = evt.target
104     var cont = dom.className;
105     console.log(dom.dataset)
106     console.log(cont)
107     const targetEle = dom.nextElementSibling
108     get('/subsubsubject/' + cont).then(subjects => {
109         // const subsider = document.querySelector('.' + cont)
110         const subsubsiderContent = subjects.map(renderSubSubSiderItem).join(
111             '\n')
112         targetEle.innerHTML = subsubsiderContent
113         console.log('[subsubsider] loaded!', subsubsiderContent)
114     })
115
116     // targetEle.innerHTML = new Array(10).fill(0).map(v => '<div class=
117     "sub-item">我是子类</div>').join('\n')
118 }
119
120 function renderCourseItem(data = {}) {
121     let courseTemplate = '<p class="course-$id"><b>$name</b><a style="color:
122     rgb(66, 66, 66); font-size:13px;"> 作者: $author 来源: $src</a></p>'
123     Object.keys(data).forEach(key => {
124         courseTemplate = courseTemplate.replaceAll('$' + key, data[key])
125     })
126     return courseTemplate
127 }
128
129 function loadAndRenderCourseItem(evt) {
130     const dom = evt.target
131     var cont = dom.className;
132     var index = cont.search(/\d/)
133     var num1 = cont.substring(index, cont.length)
134     console.log(num1)
135     targetEle = document.getElementsByClassName("courselist")[0]
136     console.log(targetEle.innerHTML)
137     get('/course/' + num1).then(subjects => {
138         // const subsider = document.querySelector('.' + cont)
139         const courseContent = subjects.map(renderCourseItem).join('\n')
140         targetEle.innerHTML = courseContent
141         console.log('[course] loaded!', courseContent)
142     })
143 }
144
145 function init() {
146     loadAndRenderHeader()
147     loadAndRenderSide()
148 }

```

149

150 `window.onload = init`

## E 附录: styles.css

```
1 * {
2     box-sizing: border-box;
3 }
4
5 body {
6     margin: 0;
7 }
8
9 .header {
10     background-color: #2196F3;
11     color: white;
12     text-align: center;
13     padding: 15px;
14 }
15
16 .footer {
17     background-color: #444;
18     color: white;
19     padding: 10px;
20 }
21
22 .tips {
23     font-size: 13px;
24 }
25
26 .courselist {
27     margin: 0;
28     padding: 0;
29     overflow: hidden;
30     background-color: rgb(226, 226, 226);
31     color: rgb(0, 0, 0)
32 }
33
34 .topmenu {
35     list-style-type: none;
36     margin: 0;
37     padding: 0;
38     overflow: hidden;
39     background-color: #777;
40 }
41
42 .topmenu li {
43     float: right;
44 }
45
46 .topmenu li a {
47     display: inline-block;
48     color: white;
```

```

49     text-align: center;
50     padding: 20px;
51     text-decoration: none;
52 }
53
54 .topmenu li a:hover {
55     background-color: #222;
56 }
57
58 .topmenu li a.active {
59     color: white;
60     background-color: #4CAF50;
61 }
62
63
64 .column {
65     /* float: left; */
66     padding: 15px;
67     height: auto;
68     min-height: 80vh;
69 }
70
71 .clearfix {
72     content: "";
73     clear: both;
74     display: flex;
75 }
76
77 .sidemenu ul {
78     list-style-type: none;
79     margin: 0;
80     padding: 0;
81 }
82
83 .sidemenu a{
84     list-style-type: none;
85     margin-bottom: 4px;
86     display: block;
87     padding: 8px;
88     background-color: rgb(213, 231, 238);
89     text-decoration: none;
90     color: rgb(66, 66, 66);
91 }
92
93 .sidemenu div[class^="sub-item-name-"] {
94     list-style-type: none;
95     margin-bottom: 6px;
96     display: block;
97     padding: 11px;
98     background-color: rgb(231, 236, 240);

```

```

99     text-decoration: none;
100     color: rgb(87, 86, 86);
101     font-size:13px
102 }
103
104 .sidemenu div[class^="sub-sub-item-name-"] {
105     list-style-type: none;
106     margin-bottom: 6px;
107     display: block;
108     padding: 11px;
109     background-color: rgb(255, 255, 255);
110     text-decoration: none;
111     color: rgb(0, 0, 0);
112     font-size:12px
113 }
114
115
116 .sidemenu a:hover {
117     background-color: #555;
118     color: white;
119 }
120 .sidemenu div[class^="sub-item-name-"]:hover {
121     background-color: rgb(160, 157, 157);
122     color: white;
123 }
124
125 .sidemenu div[class^="sub-sub-item-name-"]:hover {
126     background-color: rgb(92, 86, 86);
127     color: white;
128 }
129
130 .sidemenu a.active {
131     background-color: #008CBA;
132     color: white;
133 }
134
135
136 .footer li {
137     margin-bottom: 4px;
138     display: block;
139     text-decoration: none;
140     color: white;
141 }
142
143 a.color1:link {
144     color: white ;
145     text-decoration:underline;
146 }
147
148 a.color1:hover {

```

```
149     color: #008CBA;
150     text-decoration:underline;
151 }
152
153 /* a.color1:visited {
154     color: #FFFFFF;
155     text-decoration:none;
156 } */
```