下面给出如何将上述关系代数题目转换为 MySQL 上可执行的 SQL 查询，并通过 Python+Flask 应用来实现每个查询的详细步骤。下面示例中假定你已经在 MySQL 中创建了如下三张表（注意字段名称、数据类型需要与你实际的数据库保持一致）：

* **Student(Sno, Sname, Ssex, Sbirthdate, Smajor)**
* **Course(Cno, Cname, Ccredit, Cpno)**
* **SC(Sno, Cno, Grade, Semester, Teachingclass)**

**一、SQL 查询转换**

1. **题目1：查询专业为“信息安全”的所有学生记录**  
   **SQL语句：**
2. SELECT \* FROM Student WHERE Smajor = '信息安全';
3. **题目2：查询出生日期在2001年1月1日及以后的学生记录**  
   **SQL语句：**
4. SELECT \* FROM Student WHERE Sbirthdate >= '2001-01-01';
5. **题目3：查询所有学生的姓名和主修专业**  
   **SQL语句：**
6. SELECT DISTINCT Sname, Smajor FROM Student;
7. **题目4：查询所有不同的主修专业**  
   **SQL语句：**
8. SELECT DISTINCT Smajor FROM Student;
9. **题目5：查询选修课程号为“81003”的学生的学号、姓名和成绩**  
   **SQL语句：**
10. SELECT SC.Sno, Student.Sname, SC.Grade
11. FROM SC
12. INNER JOIN Student ON SC.Sno = Student.Sno
13. WHERE SC.Cno = '81003';
14. **题目6：查询至少选修了课程集合 K = { '81005', '81007' } 中所有课程的学生的学号**  
    **SQL语句（利用分组和 HAVING 实现“除运算”）：**
15. SELECT Sno
16. FROM SC
17. WHERE Cno IN ('81005', '81007')
18. GROUP BY Sno
19. HAVING COUNT(DISTINCT Cno) = 2;

（说明：若学生选修了这两门课程，则其在 SC 中对应的记录数（去重后）必为 2。）

1. **题目7：查询至少选修了“81001”和“81003”两门课程的学生的学号**  
   **SQL语句：**
2. SELECT Sno
3. FROM SC
4. WHERE Cno IN ('81001', '81003')
5. GROUP BY Sno
6. HAVING COUNT(DISTINCT Cno) = 2;
7. **题目8：查询选修课程号为“81002”的学生的学号**  
   **SQL语句：**
8. SELECT DISTINCT Sno FROM SC WHERE Cno = '81002';

**二、使用 Python + Flask 实现详细步骤**

下面的示例代码展示了如何利用 Flask 构建一个 Web 应用，每个路由对应一个查询接口。你可以通过浏览器或 API 工具（如 Postman）访问这些接口，查看查询结果。

**1. 环境准备**

* 安装 Flask 与 PyMySQL（用于连接 MySQL）：
* pip install flask pymysql
* 确保你的 MySQL 数据库中已创建好 Student、Course、SC 三个表，并插入相应的数据。

**2. 完整代码示例**

创建一个文件（例如 app.py），写入以下代码：

from flask import Flask, jsonify

import pymysql

app = Flask(\_\_name\_\_)

# 配置 MySQL 数据库连接参数

app.config['MYSQL\_HOST'] = 'localhost'

app.config['MYSQL\_USER'] = 'root'

app.config['MYSQL\_PASSWORD'] = 'yourpassword' # 请替换为你的密码

app.config['MYSQL\_DB'] = 'yourdatabase' # 请替换为你的数据库名称

def get\_connection():

return pymysql.connect(

host=app.config['MYSQL\_HOST'],

user=app.config['MYSQL\_USER'],

password=app.config['MYSQL\_PASSWORD'],

db=app.config['MYSQL\_DB'],

cursorclass=pymysql.cursors.DictCursor

)

# 题目1：查询专业为“信息安全”的所有学生记录

@app.route('/query1')

def query1():

conn = get\_connection()

try:

with conn.cursor() as cursor:

sql = "SELECT \* FROM Student WHERE Smajor = %s"

cursor.execute(sql, ('信息安全',))

result = cursor.fetchall()

return jsonify(result)

finally:

conn.close()

# 题目2：查询出生日期在2001年1月1日及以后的学生记录

@app.route('/query2')

def query2():

conn = get\_connection()

try:

with conn.cursor() as cursor:

sql = "SELECT \* FROM Student WHERE Sbirthdate >= %s"

cursor.execute(sql, ('2001-01-01',))

result = cursor.fetchall()

return jsonify(result)

finally:

conn.close()

# 题目3：查询所有学生的姓名和主修专业

@app.route('/query3')

def query3():

conn = get\_connection()

try:

with conn.cursor() as cursor:

sql = "SELECT DISTINCT Sname, Smajor FROM Student"

cursor.execute(sql)

result = cursor.fetchall()

return jsonify(result)

finally:

conn.close()

# 题目4：查询所有不同的主修专业

@app.route('/query4')

def query4():

conn = get\_connection()

try:

with conn.cursor() as cursor:

sql = "SELECT DISTINCT Smajor FROM Student"

cursor.execute(sql)

result = cursor.fetchall()

return jsonify(result)

finally:

conn.close()

# 题目5：查询选修课程号为“81003”的学生的学号、姓名和成绩

@app.route('/query5')

def query5():

conn = get\_connection()

try:

with conn.cursor() as cursor:

sql = """

SELECT SC.Sno, Student.Sname, SC.Grade

FROM SC

INNER JOIN Student ON SC.Sno = Student.Sno

WHERE SC.Cno = %s

"""

cursor.execute(sql, ('81003',))

result = cursor.fetchall()

return jsonify(result)

finally:

conn.close()

# 题目6：查询至少选修了课程集合 K = {'81005', '81007'} 中所有课程的学生的学号

@app.route('/query6')

def query6():

conn = get\_connection()

try:

with conn.cursor() as cursor:

sql = """

SELECT Sno

FROM SC

WHERE Cno IN (%s, %s)

GROUP BY Sno

HAVING COUNT(DISTINCT Cno) = 2

"""

cursor.execute(sql, ('81005', '81007'))

result = cursor.fetchall()

return jsonify(result)

finally:

conn.close()

# 题目7：查询至少选修了“81001”和“81003”两门课程的学生的学号

@app.route('/query7')

def query7():

conn = get\_connection()

try:

with conn.cursor() as cursor:

sql = """

SELECT Sno

FROM SC

WHERE Cno IN (%s, %s)

GROUP BY Sno

HAVING COUNT(DISTINCT Cno) = 2

"""

cursor.execute(sql, ('81001', '81003'))

result = cursor.fetchall()

return jsonify(result)

finally:

conn.close()

# 题目8：查询选修课程号为“81002”的学生的学号

@app.route('/query8')

def query8():

conn = get\_connection()

try:

with conn.cursor() as cursor:

sql = "SELECT DISTINCT Sno FROM SC WHERE Cno = %s"

cursor.execute(sql, ('81002',))

result = cursor.fetchall()

return jsonify(result)

finally:

conn.close()

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

**3. 代码说明**

* **数据库连接配置：**  
  修改 MYSQL\_HOST、MYSQL\_USER、MYSQL\_PASSWORD 和 MYSQL\_DB 为你自己的 MySQL 数据库连接信息。  
  函数 get\_connection() 用于创建数据库连接，并使用 DictCursor 以字典格式返回查询结果。
* **各查询接口：**  
  每个 @app.route('/queryX') 定义一个 URL 路径，访问该路径时，Flask 会调用对应的函数。  
  在每个函数中：
  + 先建立数据库连接；
  + 使用 cursor.execute() 执行相应的 SQL 查询语句（其中使用了参数化查询，防止 SQL 注入）；
  + 用 cursor.fetchall() 获取所有结果，并用 jsonify 将结果转换为 JSON 格式返回给客户端；
  + 最后关闭连接。
* **启动 Flask 应用：**  
  运行 python app.py 后，应用会在 http://localhost:5000 启动。  
  你可以在浏览器中访问例如 http://localhost:5000/query1 来查看题目1的查询结果，其它查询类似。

**三、总结**

1. 首先将关系代数表达式转换为对应的 SQL 查询语句。
2. 利用 Python 的 Flask 框架构建 Web 接口，并通过 PyMySQL 模块连接 MySQL 数据库，执行 SQL 查询。
3. 每个路由对应一个具体查询，最终以 JSON 格式将结果返回给客户端。

这样一来，你就可以通过实际的 MySQL 数据库和 Python+Flask 应用来验证和学习各个关系代数查询在实际数据库中的实现方法。