

# 数据库第三次作业-冯尔宁-2022302181149

## 设学生一选课数据库的关系 student、s\_c、course（见表 1~表 3）。

(1) 试定义下列完整性约束：

- 在关系 student 中插入的学生年龄值应在 15-25 岁之间。

```
ALTER TABLE student add constraint age_check CHECK(age between 15 and 25);
```

- 在关系 s\_c 中插入元组时，其 sno 值和 cno 值必须分别在关系 student 和关系 course 中出现。

```
ALTER TABLE s_c add constraint fk_student_sno FOREIGN KEY (sno) REFERENCES student(sno);  
ALTER TABLE s_c add constraint fk_course_cno FOREIGN KEY (cno) REFERENCES course(cno);
```

- 在关系 s\_c 中修改 grade 值时，必须仍在 0-100 之间。

```
ALTER TABLE s_c add constraint check_grade CHECK(grade between 0 and 100);
```

- 在删除关系 course 中一个元组时，首先要把关系 s\_c 中具有同样 cno 值的元组全部删去。

```
ALTER TABLE s_c add constraint fk_course_cno FOREIGN KEY (cno) REFERENCES course(cno) ON DELETE CASCADE;
```

- 在关系 student 中把某个 sno 值修改为新值时，必须同时把关系 s\_c 中那些同样的 sno 值也修改为新值。

```
ALTER TABLE s_c add constraint fk_student_sno FOREIGN KEY (sno) REFERENCES student(sno) ON UPDATE CASCADE;
```

(2) 试用 SQL 的断言机制定义下列完整性约束：

- 学生必须在选修“数学课”课后，才能选修其他课程。

```
CREATE ASSERTION Enroll_Math_First
CHECK (
    not exists(
        select 1 from s_c
        where cno not in (select cno from course where course_name = '数学课')
        and sno not in (select sno from s_c join course on s_c.cno = course.cno where course_name = '数学课')
    )
);
```

- 每个艺术系的学生最多选修 20 门课程。

```
CREATE ASSERTION Limit_Art_Student_Courses
CHECK (
    not exists (
        select 1 from student where dept = '艺术系'
        group by sno
        having COUNT(*) > 20
    )
);
```