



中国科学技术大学  
University of Science and Technology of China

# Lab 1

## SQL 练习

姓名: 高茂航

学号: PB22061161

日期: 2024.4.20

# Lab 1

---

设有如下关系模式:Student(SNO, NAME, GENDER, BIRTHDAY, DEPART) SNO 为学生学号,DEPART 为系别 Teacher(TNO,NAME,GENDER,BIRTHDAY,POSITION,DEPART) TNO 为教师工号, POSITION 为职称, DEPART 为系别。Course(CNO, NAME, TYPE, TNO) CNO 为课程编号, TYPE 为课程类型, 1 表示必修课, 0 表示选修课, TNO 为教师工号。Score(SNO, CNO, DEGREE) SNO 为学生学号, CNO 为课程编号, DEGREE 为成绩。PS: 加下划线“   ” 的表示该字段为主键。

## 1 任务一

根据所给 Student.csv、Teacher.csv、Course.csv、Score.csv 表中的数据信息, 在数据库中创建对应的关系表并将数据录入到数据库中。可能涉及到的数据类型: varchar, char, int, float ,datetime。(也需要截图展示!)

## 2 任务二:实现以下各题功能的 SQL 语句 (部分题目的结果受前面题目影响)

### 2.1 修改基本表

2.1.1 在学生表 student 中增加一个新的属性列 AGE(年龄), 类型为 int

2.1.2 计算每个学生的年龄 (AGE, 即当前年份减去出生年份)

注意, 此操作可能需要关闭安全更新模式; 提示, 可使用 MySQL 的 YEAR 函数。

2.1.3 为每个学生的年龄加 2, 将 AGE(年龄)的数据类型由 int 改为 char

2.1.4 删除属性列 AGE

2.1.5 创建一个教师课程数量表: teacher\_course(TNO, NUM\_COURSE), 两个属性分别表示授课教师工号, 课程数量, 其中 TNO 是主键

2.1.6 用一条语句, 结合表 course 记录, 为表 teacher 中所有教师, 在表 teacher\_course 添加对应记录(用到查询, 而不是手动数 NUM 插入; 若是表 course 中未出现的教师, 则课程数量记为 NULL)

2.1.7 删除表 teacher\_course 中含有 NULL 的记录

2.1.8 删除表 teacher\_course

2.1.9 在学生表 student 、成绩表 score 中分别插入一些数据, 数据如下 (注意: 如果与原有数据冲突, 比如学号重复, 请修改一下自己的信息保证能够插入):

请保证选择的三门课成绩不一样。

# Lab 1

---

2.1.10 在 score 表中删除你所选的课程中成绩最低的一门课程的记录(可能会用到子查询)

## 2.2 索引

2.2.1 用 create 语句在 course 表的名称 NAME 上建立普通索引 NAME\_INDEX

2.2.2 用 create 语句在 teacher 表的工号 TNO 上建立唯一索引 TNO\_INDEX

2.2.3 用 create 语句在 score 表上的学号 SNO、成绩 DEGREE 上建立复合索引 RECORD\_INDEX, 要求学号为降序, 学号相同时成绩为升序

2.2.4 用一条语句查询表 score 的索引

2.2.5 删除 teacher 表字段 TNO 上的索引 TNO\_INDEX

## 2.3 查询

注: 以下每道题限制用一条 SQL 语句完成

2.3.1 查询和你属于同一个系的学生学号和姓名(包括你本人)

2.3.2 查询和你属于同一个系的学生学号和姓名(不包括你本人)

2.3.3 查询和你的某个好友属于同一个系的学生学号和姓名(9 题插入的某个好友)

2.3.4 查询和你的两个好友都不在一个系的学生学号和姓名(9 题插入的两个好友)

2.3.5 查询教过你的所有老师的工号和姓名

2.3.6 查询 11 系和 229 系教师的总人数

2.3.7 查询你的系中年龄(即当前年份减去出生年份)最大的学生的学号、姓名和年龄

2.3.8 查询你的系中年龄(即当前年份减去出生年份)最小的学生的学号、姓名和年龄

2.3.9 查询选修 DB\_Design 课程且成绩在 75 分以下(不包括 75)的学生的学号、姓名和分数

2.3.10 查询选修过“ZDH”老师课程的学生学号和姓名(去掉重复行)

2.3.11 查询选过某课程的学生学号和分数, 并按分数降序展示

(某课程是指 course 表中的某一课程名 NAME, 你自行选择; 分数指的是选的这门课的成绩, 不包括选这门课的同学的其他成绩)。

## Lab 1

---

- 2.3.12 查询每门课的平均成绩, 其中每行包含课程号、课程名和平均成绩(包括平均成绩为 NULL, 即该课没有成绩)
- 2.3.13 查询每门必修课的平均成绩, 其中每行包含课程号、课程名和平均成绩(包括平均成绩为 NULL, 即该课没有成绩)
- 2.3.14 查询至少选修了 ZDH 老师(TNO="TA90023")开设的所有课程的学生学号
- 2.3.15 查询每门课程的最高分和最低分, 并计算其分数差。其中每行包含课程号、课程名和最高分、最低分和分数差(课程无成绩的不用包括)
- 2.3.16 查询存在考试成绩低于 75 分的学生学号, 以及低于 75 分的课程数量
- 2.3.17 查询所教过的课程中有学生考试成绩低于 75 分的教师的工号和姓名(去掉重复行)
- 2.3.18 查询选修少于 2 门课程的学生的学号、姓名
- 2.3.19 查询至少选修了 YH 同学(SNO="PB210000001") 所选全部课程的学生学号
- 2.3.20 查询 Course 表中各个课程名称与相应的平均成绩(没有选课的学生平均成绩为 NULL)
- 2.3.21 查询每个系的学生人数和每个系的平均分, 其中每行包含系号、系的人数和平均成绩

(计算人数的时候需要包括那些没有成绩的同学, 但是计算成绩的时候不需要包括这些同学)

# Lab 1

---

- 2.3.22 查询所有未选修 DB\_Design 课程或者 Data\_Mining 课程的学生的学生姓名(去掉重复行)
- 2.3.23 查询各个课程的课程名及选该课的学生的最小年龄、最大年龄和平均年龄。(包括没有人选的课)
- 2.3.24 查询选修了课程名中包含"Computer"课程的学生的学号和姓名
- 2.3.25 设课程平均成绩为  $x$ , 查询各个课程成绩处于  $[x-12, x+12]$  区间的同学的成绩表, 即包含 SNO、CNO、DEGREE

## 2.4 视图

- 2.4.1 建立 229 系的学生视图(db\_229\_student), 属性与 student 表一样, 并要求对该视图进行修改和插入操作时仍需保证该视图只有 229 系的学生
- 2.4.2 将 229 系学生视图(db\_229\_student)中学号为"PB210000020"的学生姓名改为你的姓名(英文首字母)
- 2.4.3 在 229 系学生视图(db\_229\_student)中找出年龄小于 22 岁的学生, 包含 SNO、NAME
- 2.4.4 向 student 表中插入一名"学号 SA210110021, 姓名 QXY, 性别女, 生日 2007/7/27, 229 系"的学生。然后查询视图 db\_229\_student 的所有学生, 验证其是否更新
- 2.4.5 向视图 db\_229\_student 中插入一名"学号 SA210110023, 姓名 DPC, 性别男, 生日 1997/4/27, 11 系"的学生, 观察到了什么现象
- 2.4.6 删除视图 db\_229\_student

## 2.5 触发器

- 2.5.1 创建关系表:teacher\_salary(TNO, SAL), 其中 TNO 是教师工号(主键), SAL 是教师工资(类型 float)
- 2.5.2 定义一个 BEFORE 行级触发器, 为关系表 teacher\_salary 定义完整性规则:"表中出现的工号必须也出现在 teacher 表中, 否则报错"。

注:该规则实际上就是外键约束;MySQL 中可使用 SIGNAL 抛出错误;需要为 INSERT 和 UPDATE 分别定义触发器。请展示成功创建触发器和测试抛出错误信息的截图。

- 2.5.3 定义一个 BEFORE 行级触发器, 为关系表 teacher\_salary 定义完整性规则:"Instructor/Associate Professor/Professor 的工资不能低于 4000/7000/10000, 且不能高于 7000/10000/13000, 如果低于, 则改为 4000/7000/10000", 如果高于, 则改为 7000/10000/13000

注: 需要为 INSERT 和 UPDATE 分别定义触发器。并检验触发器是否工作: 为 teacher\_salary 构造 INSERT 和 UPDATE 语句并激活所定义过的触发器, 将过程截图展示

# Lab 1

---

2.5.4 删除刚刚创建的所有触发器

2.6 空值

2.6.1 将 score 表中的 Data\_Mining 课程成绩设为空值, 然后在 score 表查询学生学号和分数, 并按分数升序展示。观察 NULL 在 MySQL 中的大小是怎样的

2.7 开放题

2.7.1

2.7.2

## 3 Conclusion