Lab 1
SQL 练习

姓名：_____高茂航_____

学号：_____PB22061161_____

日期：_____2024.4.20_____

Lab 1

1 数据库安装配置结果


```
C:\Windows\system32>mysqld install
The service already exists!
The current server installed: D:\mysql-8.0.36-winx64\bin\mysqld MySQL

C:\Windows\system32> net start mysql
The requested service has already been started.

More help is available by typing NET HELPMSG 2182.
```

2 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。

```
1   create schema s_t_c_s;
2  create table student
3   (SNO char(11) primary key,
4      NAME varchar(4),
5      GENDER varchar(6),
6      BIRTHDAY datetime,
7      DEPART int
8   );
9
10 insert into student (SNO, NAME, GENDER, BIRTHDAY, DEPART)
11 values
12 ('PB210000001', 'YH', 'male', '2002-03-29 00:00:00', 229),
13 ('PB210000002', 'ZY', 'male', '2001-09-12 00:00:00', 11),
14 ('PB210000003', 'FWJ', 'male', '2001-04-29 00:00:00', 12),
15 ('PB210000004', 'JTY', 'male', '2002-03-15 00:00:00', 11),
16 ('PB210000005', 'YY', 'female', '2002-08-12 00:00:00', 12),
17 ('PB210000006', 'HCC', 'male', '2002-06-25 00:00:00', 229),
18 ('PB210000007', 'RZJ', 'male', '2002-06-14 00:00:00', 11),
19 ('PB210000008', 'WCS', 'male', '2002-08-23 00:00:00', 13),
20 ('PB210000009', 'ZMS', 'female', '2002-06-23 00:00:00', 12),
21 ('PB210000010', 'WD', 'male', '2003-02-24 00:00:00', 13),
22 ('PB210000011', 'BL', 'female', '2002-05-08 00:00:00', 14),
23 ('PB210000012', 'ADN', 'male', '2004-06-26 00:00:00', 10),
24 ('PB210000013', 'HD', 'male', '2003-11-17 00:00:00', 14),
25 ('PB210000014', 'GNJ', 'male', '2001-01-28 00:00:00', 11),
26 ('PB210000015', 'XB', 'female', '2002-10-09 00:00:00', 12),
27 ('PB210000016', 'LC', 'female', '2003-11-30 00:00:00', 11),
28 ('PB210000017', 'TX', 'male', '2002-05-16 00:00:00', 12),
29 ('PB210000018', 'MY', 'male', '2003-12-02 00:00:00', 14),
30 ('PB210000019', 'MT', 'female', '2002-02-13 00:00:00', 10),
31 ('PB210000020', 'XY', 'female', '2001-02-14 00:00:00', 229),
32 ('PB210000021', 'LYH', 'male', '2002-09-30 00:00:00', 229),
33 ('PB210000022', 'MSW', 'male', '2003-06-17 00:00:00', 11),
34 ('PB210000023', 'HXY', 'male', '2003-02-18 00:00:00', 12),
35 ('PB210000024', 'YHS', 'female', '2003-04-01 00:00:00', 229),
36 ('PB210000025', 'YWB', 'male', '2003-08-12 00:00:00', 229);
37
38 create table teacher
39   (TNO char(7) primary key,
40    NAME varchar(4),
41      GENDER varchar(6),
42      BIRTHDAY datetime,
```
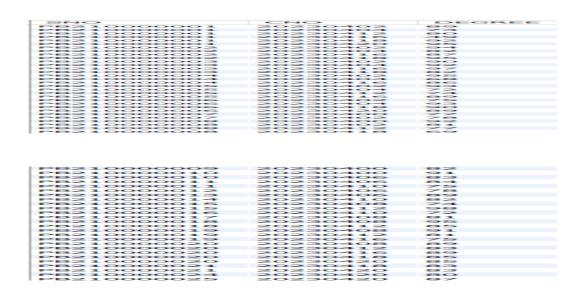
```
43       POSITION varchar(30),
44       DEPART int
45    );
46
47  insert into teacher (TNO, NAME, GENDER, BIRTHDAY, POSITION, DEPART)
48  values
49  ('TA90021', 'HMZ', 'male', '1994/12/23 00:00:00', 'Instructor', 11),
50  ('TA90022', 'HB', 'female', '1978/4/9 00:00:00', 'Associate Professor', 10),
51  ('TA90023', 'ZDH', 'male', '1986/11/17 00:00:00', 'Instructor', 229),
52  ('TA90024', 'HS', 'male', '1977/4/10 00:00:00', 'Associate Professor', 6),
53  ('TA90025', 'HTZ', 'female', '1969/7/28 00:00:00', 'Professor', 11),
54  ('TA90026', 'TJY', 'male', '1973/10/3 00:00:00', 'Associate Professor', 12),
55  ('TA90027', 'XR', 'male', '1970/5/16 00:00:00', 'Associate Professor', 11),
56  ('TA90028', 'ZXY', 'male', '1986/7/16 00:00:00', 'Instructor', 229),
57  ('TA90029', 'ZR', 'male', '1975/9/24 00:00:00', 'Associate Professor', 11),
58  ('TA90030', 'SY', 'male', '1972/11/6 00:00:00', 'Professor', 229),
59  ('TA90031', 'LQA', 'female', '1986/11/17 00:00:00', 'Associate Professor', 10),
60  ('TA90032', 'GHJ', 'male', '1976/10/1 00:00:00', 'Associate Professor', 18);
61
62  create table course
63    (CNO char(8) primary key,
64     NAME varchar(50),
65       TYPE int,
66     TNO char(7),
67       foreign key(TNO) references teacher(TNO)
68       );
69
70  insert into course (CNO, NAME, TYPE, TNO)
71  values
72  ('20230400', 'Linear_Algebra', '1', 'TA90022'),
73  ('20230402', 'DB_Design', '1', 'TA90021'),
74  ('20230404', 'Machine_Learning', '1', 'TA90023'),
75  ('20230406', 'Operating_System', '1', 'TA90025'),
76  ('20230408', 'Natural_Language_Processing', '0', 'TA90027'),
77  ('20230410', 'Artificial_Intelligence', '1', 'TA90025'),
78  ('20230412', 'Data_Mining', '1', 'TA90025'),
79  ('20230414', 'Signal_Control', '1', 'TA90024'),
80  ('20230416', 'Computer_Network', '1', 'TA90029'),
81  ('20230418', 'Pattern_Recognition', '1', 'TA90023'),
82  ('20230420', 'Deep_Learning', '0', 'TA90030');
83
84  create table score (
85      SNO char(11),
86      CNO char(8),
87      DEGREE int,
88      primary key (SNO, CNO),
89      foreign key (SNO) references student(SNO),
90      foreign key (CNO) references course(CNO)
91  );
92
93
94  insert into score (SNO, CNO, DEGREE)
95  values
96  ('PB210000001', '20230402', 89),
97  ('PB210000002', '20230402', 94),
98  ('PB210000003', '20230402', 90),
99  ('PB210000004', '20230402', 95),
100 ('PB210000005', '20230402', 93),
101 ('PB210000006', '20230402', 75),
```

```
102 ('PB210000007', '20230402', 78),
103 ('PB210000008', '20230402', 81),
104 ('PB210000001', '20230404', 73),
105 ('PB210000002', '20230404', 82),
106 ('PB210000003', '20230404', 92),
107 ('PB210000004', '20230404', 68),
108 ('PB210000005', '20230404', 72),
109 ('PB210000006', '20230404', 93),
110 ('PB210000007', '20230400', 77),
111 ('PB210000008', '20230400', 92),
112 ('PB210000009', '20230400', 82),
113 ('PB210000010', '20230400', 91),
114 ('PB210000008', '20230418', 69),
115 ('PB210000001', '20230418', 92),
116 ('PB210000002', '20230418', 95),
117 ('PB210000003', '20230418', 82),
118 ('PB210000010', '20230406', 83),
119 ('PB210000011', '20230406', 84),
120 ('PB210000012', '20230406', 78),
121 ('PB210000013', '20230406', 89),
122 ('PB210000017', '20230408', 81),
123 ('PB210000015', '20230408', 74),
124 ('PB210000018', '20230408', 95),
125 ('PB210000019', '20230408', 91),
126 ('PB210000020', '20230408', 89),
127 ('PB210000014', '20230410', 82),
128 ('PB210000016', '20230410', 72),
129 ('PB210000011', '20230410', 75),
130 ('PB210000018', '20230412', 85),
131 ('PB210000020', '20230412', 83),
132 ('PB210000008', '20230412', 77),
133 ('PB210000005', '20230412', 74),
134 ('PB210000015', '20230412', 98),
135 ('PB210000001', '20230412', 80),
136 ('PB210000019', '20230412', 81),
137 ('PB210000019', '20230416', 75),
138 ('PB210000020', '20230416', 89),
139 ('PB210000001', '20230416', 49),
140 ('PB210000002', '20230416', 87),
141 ('PB210000003', '20230416', 97),
142 ('PB210000004', '20230416', 86),
143 ('PB210000005', '20230416', 89),
144 ('PB210000006', '20230418', 90),
145 ('PB210000020', '20230418', 85),
146 ('PB210000021', '20230418', 83),
147 ('PB210000001', '20230420', 95),
148 ('PB210000006', '20230420', 89),
149 ('PB210000020', '20230420', 85),
150 ('PB210000021', '20230420', 83),
151 ('PB210000024', '20230420', 80),
152 ('PB210000025', '20230420', 87);
153
154 select * from student;
155 select * from teacher;
156 select * from course;
157 select * from score;
158
159
```

SNO	NAME	GENDER	BIRTHDAY	DEPART
PB21000000001	YH	male	2002-03-29 00:00:00	229
PB21000000002	ZY	male	2001-09-12 00:00:00	11
PB21000000003	FWJ	male	2001-04-25 00:00:00	11
PB21000000004	JTY	male	2002-03-15 00:00:00	11
PB21000000005	WY	female	2002-08-15 00:00:00	12
PB21000000006	HCC	male	2002-06-25 00:00:00	229
PB21000000007	RZJ	male	2002-06-14 00:00:00	11
PB21000000008	WCS	male	2002-08-23 00:00:00	11
PB21000000009	ZMS	female	2002-06-23 00:00:00	12
PB21000000010	WD	male	2003-02-24 00:00:00	13
PB21000000011	RL	female	2002-05-08 00:00:00	14
PB21000000012	ADN	male	2004-06-26 00:00:00	10
PB21000000013	HD	male	2003-11-12 00:00:00	11
PB21000000014	GNJ	male	2001-01-28 00:00:00	11
PB21000000015	XB	female	2002-10-09 00:00:00	12
PB21000000016	TC	female	2003-11-30 00:00:00	12
PB21000000017	TX	male	2003-05-16 00:00:00	11
PB21000000018	MY	male	2003-12-02 00:00:00	14
PB21000000019	MT	female	2002-02-13 00:00:00	10
PB21000000020	XY	female	2001-02-11 00:00:00	229
PB21000000021	YH	male	2002-09-30 00:00:00	229
PB21000000022	MSW	male	2003-09-17 00:00:00	11
PB21000000023	HXY	male	2003-02-18 00:00:00	12
PB21000000024	YHS	female	2003-04-01 00:00:00	229
PB21000000025	YWB	male	2003-08-12 00:00:00	229

TNO	NAME	GENDER	BIRTHDAY	POSITION	DEPART
TA90021	HMZ	male	1994-12-23 00:00:00	Instructor	11
TA90022	HB	female	1978-04-09 00:00:00	Associate Professor	10
TA90023	ZDH	male	1986-11-17 00:00:00	Instructor	229
TA90024	HS	male	1977-04-10 00:00:00	Associate Professor	6
TA90025	HTZ	female	1969-07-28 00:00:00	Professor	11
TA90026	TJY	male	1973-10-03 00:00:00	Associate Professor	12
TA90027	XR	male	1970-05-16 00:00:00	Associate Professor	11
TA90028	ZXY	male	1986-07-16 00:00:00	Instructor	229
TA90029	ZR	male	1975-09-24 00:00:00	Associate Professor	11
TA90030	SY	male	1972-11-06 00:00:00	Professor	229
TA90031	LQA	female	1986-11-17 00:00:00	Associate Professor	10
TA90032	GHJ	male	1976-10-01 00:00:00	Associate Professor	18

CNO	NAME	TYPE	TNO
20230400	Linear_Algebra	1	TA90022
20230402	DB_Design	1	TA90021
20230404	Machine_Learning	1	TA90023
20230406	Operating_System	1	TA90025
20230408	Natural_Language_Processing	0	TA90027
20230410	Artificial_Intelligence	1	TA90025
20230412	Data_Mining	1	TA90025
20230414	Signal_Control	1	TA90024
20230416	Computer_Network	1	TA90029
20230418	Pattern_Recognition	1	TA90023
20230420	Deep_Learning	0	TA90030

3 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)

3.1 修改基本表

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

3.1.2 计算每个学生的年龄 (AGE)

1

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

1

3.1.4 删除属性列 AGE

1

3.1.5 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

1

3.1.6 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(若表 **course** 中未出现的教师,则课程数量记为 **NULL**)

1

3.1.7 删除表 **teacher_course** 中含有 **NULL** 的记录

1

3.1.8 删除表 teacher_course

1

3.1.9 在学生表 student 、成绩表 score 中分别插入一些数据

1

3.1.10 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）

1

3.2 索引

3.2.1 用 create 语句在 course 表的名称 NAME 上建立普通索引 NAME_INDEX

1

3.2.2 用 **create** 语句在 **teacher** 表的工号 **TNO** 上建立唯一索引 **TNO_INDEX**

1

3.2.3 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

1

3.2.4　用一条语句查询表 score 的索引

1

3.2.5　删除 teacher 表字段 TNO 上的索引 TNO_INDEX

1

3.3 查询

3.3.1 查询和你属于同一个系的学生学号和姓名 (包括你本人)

1

3.3.2 查询和你属于同一个系的学生学号和姓名 (不包括你本人)

1

3.3.3 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）

1

3.3.4 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）

1

3.3.5 查询教过你的所有老师的工号和姓名

1

3.3.6 查询 11 系和 229 系教师的总人数

1

3.3.7 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

1

3.3.8 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

1

3.3.9 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

1

3.3.10 查询选修过"ZDH"老师课程的学生学号和姓名（去掉重复行）

1

3.3.11 查询选过某课程的学生学号和分数，并按分数降序展示

1

3.3.12 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）

1

3.3.13 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）

1

3.3.14 查询至少选修了 **ZDH** 老师（**TNO="TA90023"**）开设的所有课程的学生学号

1

3.3.15 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)

1

3.3.16 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

1

3.3.17 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）

1

3.3.18 查询选修少于 **2** 门课程的学生的学号、姓名

1

3.3.19 查询至少选修了 YH 同学 (SNO="PB210000001") 所选全部课程的学生学号

1

3.3.20 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）

1

3.3.21 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

1

3.3.22 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）

1

3.3.23 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）

1

3.3.24 查询选修了课程名中包含"Computer"课程的学生的学号和姓名

1

3.3.25 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**

1

3.4 视图

3.4.1 建立 **229** 系的学生视图（db_22_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

1

3.4.2 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)

1

3.4.3 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

1

3.4.4 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

1

3.4.5 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？

1

3.4.6 删除视图 db_229_student

```
1    drop view db_229_student;
```

3.5 触发器

3.5.1 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）

```
1
```


3.5.2 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。请展示出成功创建触发器和测试抛出错误信息的截图。

```
1
```

3.5.3 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

1

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

3.5.4 删除刚刚创建的所有触发器

1

3.6 空值

3.6.1 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

1

3.7 开放题

3.7.1

1

3.7.2

1

4   Conclusion