



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

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,
```

Lab 1

```
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),
```

Lab 1

```
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
```

Lab 1

SNO	NAME	GENDER	BIRTHDAY	DEPART
PB210000001	YH	male	2002-03-29 00:00:00	229
PB210000002	ZY	male	2001-09-12 00:00:00	11
PB210000003	FWJ	male	2001-04-29 00:00:00	12
PB210000004	JTY	male	2002-03-15 00:00:00	11
PB210000005	YY	female	2002-08-12 00:00:00	12
PB210000006	HCC	male	2002-06-25 00:00:00	229
PB210000007	RZJ	male	2002-06-14 00:00:00	11
PB210000008	WCS	male	2002-08-23 00:00:00	13
PB210000009	ZMS	female	2002-06-23 00:00:00	12
PB210000010	WD	male	2003-02-24 00:00:00	13
PB210000011	BL	female	2002-05-08 00:00:00	14
PB210000012	ADN	male	2004-06-26 00:00:00	10
PB210000013	HD	male	2003-11-17 00:00:00	14
PB210000014	GNJ	male	2001-01-28 00:00:00	11
PB210000015	XB	female	2002-10-09 00:00:00	12
PB210000016	LC	female	2003-11-30 00:00:00	11
PB210000017	TX	male	2002-05-16 00:00:00	12
PB210000018	MY	male	2003-12-02 00:00:00	14
PB210000019	MT	female	2002-02-13 00:00:00	10
PB210000020	XY	female	2001-02-14 00:00:00	229
PB210000021	LYH	male	2002-09-30 00:00:00	229
PB210000022	MSW	male	2003-06-17 00:00:00	11
PB210000023	HXY	male	2003-02-18 00:00:00	12
PB210000024	YHS	female	2003-04-01 00:00:00	229
PB210000025	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

Lab 1

SNO	CNO	DEGREE
PB210000001	20230402	89
PB210000001	20230404	73
PB210000001	20230412	80
PB210000001	20230416	49
PB210000001	20230418	92
PB210000001	20230420	95
PB210000002	20230402	94
PB210000002	20230404	82
PB210000002	20230416	87
PB210000002	20230418	95
PB210000003	20230402	90
PB210000003	20230404	92
PB210000003	20230416	97
PB210000003	20230418	82
PB210000004	20230402	95
PB210000004	20230404	68
PB210000004	20230416	86
PB210000005	20230402	93
PB210000005	20230404	72
PB210000005	20230412	74
PB210000005	20230416	89
PB210000006	20230402	75
PB210000006	20230404	93
PB210000006	20230418	90
PB210000006	20230420	89
PB210000007	20230400	77
PB210000007	20230402	78
PB210000008	20230400	92
PB210000008	20230402	81
PB210000008	20230412	77
PB210000008	20230418	69

PB210000009	20230400	82
PB210000010	20230400	91
PB210000010	20230406	83
PB210000011	20230406	84
PB210000011	20230410	75
PB210000012	20230406	78
PB210000013	20230406	89
PB210000014	20230410	82
PB210000015	20230408	74
PB210000015	20230412	98
PB210000016	20230410	72
PB210000017	20230408	81
PB210000018	20230408	95
PB210000018	20230412	85
PB210000019	20230408	91
PB210000019	20230412	81
PB210000019	20230416	75
PB210000020	20230408	89
PB210000020	20230412	83
PB210000020	20230416	89
PB210000020	20230418	85
PB210000020	20230420	85
PB210000021	20230418	83
PB210000021	20230420	83
PB210000024	20230420	80
PB210000025	20230420	87

Lab 1

3 任务二:实现以下各题功能的 SQL 语句

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

```
1 alter table student add AGE int;
```

3.1.2 计算每个学生的年龄 (AGE)

```
1 set sql_safe_updates = 0;
2 update student set AGE = YEAR(CURDATE()) - YEAR(BIRTHDAY);
3 select AGE from student;
```

AGE
22
23
23
22
22
22
22
22
22
21
22
20
21
23
22
21
22
21
22
23
22
21
21
21
21

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 update student set AGE = AGE + 2;
2 select AGE from student;
3
4 alter table student modify column AGE CHAR(2);
```

Lab 1

AGE
24
25
25
24
24
24
24
24
23
24
22
23
25
24
23
24
23
24
25
24
23
23
23
23

Column	Type	Default Value	Nulla...	Character ...	Collation	Privileges	Extra	Comments
AGE	char(2)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references		

3.1.4 删除属性列 AGE

```
1 alter table student drop column AGE;
2 select * from student;
```

Column	Type	Default Value	Nulla...	Character ...	Collation	Privileges	Extra	Comments
BIRTHDAY	datetime		YES			select,insert,update,references		
DEPART	int		YES			select,insert,update,references		
GENDER	varchar(6)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references		
NAME	varchar(4)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references		
SNO	char(11)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references		

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 create table teacher_course
2 (TNO char(7) primary key,
3  NUM_COURSE int,
4  foreign key(TNO) references Teacher(TNO)
5  );
```

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 insert into teacher_course (TNO, NUM_COURSE)
2 select
3     teacher.TNO,
4     CASE
5         WHEN COUNT(course.CNO) > 0 THEN COUNT(course.CNO)
6         ELSE NULL
7     END AS NUM_COURSE
8 from teacher
9 left join course on teacher.TNO = course.TNO
10 group by teacher.TNO;
```


Lab 1

```
11
12 select * from teacher_course;
```

TNO	NUM_COURSE
TA90021	1
TA90022	1
TA90023	2
TA90024	1
TA90025	3
TA90026	NULL
TA90027	1
TA90028	NULL
TA90029	1
TA90030	1
TA90031	NULL
TA90032	NULL

3.1.7 删除表 teacher_course 中含有 NULL 的记录

```
1 delete from teacher_course
2 where NUM_COURSE is NULL;
3
4 select * from teacher_course;
```

TNO	NUM_COURSE
TA90021	1
TA90022	1
TA90023	2
TA90024	1
TA90025	3
TA90027	1
TA90029	1
TA90030	1

3.1.8 删除表 teacher_course

```
1 drop table teacher_course;
```

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

```
1 alter table score drop foreign key score_ibfk_1;
2
3 alter table student modify column Sno VARCHAR(11);
4 alter table score add foreign key (SNO) references Student(Sno);
5
6 insert into student (SNO, NAME, GENDER, BIRTHDAY, DEPART)
7 values
8 ('PB22061161', 'GMH', 'male', '2004-07-25 00:00:00', 229),
```

Lab 1

```
9 ('PB22081558', 'LYX', 'female', '2004-7-7 00:00:00', 229),
10 ('PB22061177', 'WYB', 'male', '2003-12-29 00:00:00', 11);
11
12 select * from student;
13
14 alter table score drop foreign key score_ibfk_3;
15
16 alter table score modify column Sno VARCHAR(11);
17
18 alter table score add foreign key (SNO) references Student(Sno);
19
20 insert into score (SNO, CNO, DEGREE)
21 values
22 ('PB22061161', '20230402', 96),
23 ('PB22061161', '20230410', 97),
24 ('PB22061161', '20230412', 99);
25
26 select * from score;
```

Sno	NAME	GENDER	BIRTHDAY	DEPART
PB210000013	HD	male	2003-11-17 00:00:00	14
PB210000014	GNJ	male	2001-01-28 00:00:00	11
PB210000015	XB	female	2002-10-09 00:00:00	12
PB210000016	LC	female	2003-11-30 00:00:00	11
PB210000017	TX	male	2002-05-16 00:00:00	12
PB210000018	MY	male	2003-12-02 00:00:00	14
PB210000019	MT	female	2002-02-13 00:00:00	10
PB210000020	XY	female	2001-02-14 00:00:00	229
PB210000021	LYH	male	2002-09-30 00:00:00	229
PB210000022	MSW	male	2003-06-17 00:00:00	11
PB210000023	HXY	male	2003-02-18 00:00:00	12
PB210000024	YHS	female	2003-04-01 00:00:00	229
PB210000025	YWB	male	2003-08-12 00:00:00	229
PB22061161	GMH	male	2004-07-25 00:00:00	229
PB22061177	WYB	male	2003-12-29 00:00:00	11
PB22081558	LYX	female	2004-07-07 00:00:00	229

Lab 1

Sno	CNO	DEGREE
PB210000010	20230406	83
PB210000011	20230406	84
PB210000011	20230410	75
PB210000012	20230406	78
PB210000013	20230406	89
PB210000014	20230410	82
PB210000015	20230408	74
PB210000015	20230412	98
PB210000016	20230410	72
PB210000017	20230408	81
PB210000018	20230408	95
PB210000018	20230412	85
PB210000019	20230408	91
PB210000019	20230412	81
PB210000019	20230416	75
PB210000020	20230408	89
PB210000020	20230412	83
PB210000020	20230416	89
PB210000020	20230418	85
PB210000020	20230420	85
PB210000021	20230418	83
PB210000021	20230420	83
PB210000024	20230420	80
PB210000025	20230420	87
PB22061161	20230402	96
PB22061161	20230410	97
PB22061161	20230412	99

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  select min(DEGREE) into @minDegree
2  from score
3  where SNO='PB22061161';
4
5  delete from score
6  where SNO='PB22061161' and DEGREE=@minDegree;
7
8  select * from score where SNO='PB22061161';

```

Sno	CNO	DEGREE
PB22061161	20230410	97
PB22061161	20230412	99

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 create index NAME_INDEX on course(name);
2 show index from course;

```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
teacher	0	PRIMARY	1	TNO	A	12				BTREE			YES	
teacher	0	TNO_INDEX	1	TNO	A	12				BTREE			YES	

Lab 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 create unique index TNO_INDEX on teacher(tno);
2 show index from teacher;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
score	0	PRIMARY	1	Sno	A	24				BTREE			YES	
score	0	PRIMARY	2	CNO	A	60				BTREE			YES	
score	1	CNO	1	CNO	A	10				BTREE			YES	
score	1	RECORD_INDEX	1	Sno	D	24				BTREE			YES	
score	1	RECORD_INDEX	2	DEGREE	A	56			YES	BTREE			YES	

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 create index RECORD_INDEX on score (sno desc, degree);
```

3.2.4 用一条语句查询表 score 的索引

```
1 show index from score;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
course	0	PRIMARY	1	CNO	A	11				BTREE			YES	
course	1	TNO	1	TNO	A	8			YES	BTREE			YES	
course	1	NAME_INDEX	1	NAME	A	11			YES	BTREE			YES	

3.2.5 删除 teacher 表字段 TNO 上的索引 TNO_INDEX

```
1 drop index TNO_INDEX on teacher;
2 show index from teacher;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
teacher	0	PRIMARY	1	TNO	A	12				BTREE			YES	

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 select Sno, NAME
2 from student
3 where DEPART='229'
```

Sno	NAME
PB210000001	YH
PB210000006	HCC
PB210000020	XY
PB210000021	LYH
PB210000024	YHS
PB210000025	YWB
PB22061161	GMH
PB22081558	LYX

Lab 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 select Sno, NAME
2 from student
3 where DEPART='229' and NAME!='GMH';
```

Sno	NAME
PB210000001	YH
PB210000006	HCC
PB210000020	XY
PB210000021	LYH
PB210000024	YHS
PB210000025	YWB
PB22081558	LYX

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 select Sno, NAME
2 from student
3 where DEPART='11';
```

Sno	NAME
PB210000002	ZY
PB210000004	JTY
PB210000007	RZJ
PB210000014	GNJ
PB210000016	LC
PB210000022	MSW
PB22061177	WYB
SA210110023	DPC
NULL	NULL

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 select Sno, NAME
2 from student
3 where DEPART!='11' and DEPART!='229';
```

Sno	NAME
PB210000002	ZY
PB210000004	JTY
PB210000007	RZJ
PB210000014	GNJ
PB210000016	LC
PB210000022	MSW
PB22061177	WYB

Lab 1

3.3.5 查询教过你的所有老师的工号和姓名

```
1 select distinct teacher.Tno,teacher.NAME
2 from score,course,teacher
3 where course.CNO=score.CNO and course.TNO=teacher.TNO and score.SNO='PB22061161';
```

Tno	NAME
TA90025	HTZ

3.3.6 查询 11 系和 229 系教师的总人数

```
1 select count(TNO)
2 from teacher
3 where DEPART='11' or DEPART='229';
```

count(TNO)
7

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 select SNO,NAME,YEAR(CURDATE()) - YEAR(BIRTHDAY) as Maxage
2 from student
3 where DEPART='229'
4 and YEAR(CURDATE()) - YEAR(BIRTHDAY) = (
5     select max(YEAR(CURDATE()) - YEAR(BIRTHDAY))
6     from student
7     where DEPART='229');
```

SNO	NAME	Maxage
PB210000020	XY	23

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 select SNO,NAME,YEAR(CURDATE()) - YEAR(BIRTHDAY) as Minage
2 from student
3 where DEPART='229'
4 and YEAR(CURDATE()) - YEAR(BIRTHDAY) = (
5     select min(YEAR(CURDATE()) - YEAR(BIRTHDAY))
6     from student
7     where DEPART='229');
```

SNO	NAME	Minage
PB22061161	GMH	20
PB22081558	LYX	20

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 select student.SNO, student.NAME, score.DEGREE
2 from score, student, course
3 where student.SNO=score.SNO and score.CNO=course.CNO and course.NAME='DB_Design' and score.DEGREE<75;
```

Lab 1

SNO	NAME	DEGREE
-----	------	--------

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 select distinct student.SNO, student.NAME
2 from student, course, teacher, score
3 where student.SNO=score.SNO and score.CNO=course.CNO and course.TNO=teacher.TNO
4 and teacher.NAME='ZDH';
```

SNO	NAME
PB210000001	YH
PB210000002	ZY
PB210000003	FWJ
PB210000004	JTY
PB210000005	YY
PB210000006	HCC
PB210000008	WCS
PB210000020	XY
PB210000021	LYH

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 select score.SNO, score.DEGREE
2 from score, course
3 where score.CNO=course.CNO and course.NAME='Linear_Algebra'
4 order by score.DEGREE desc;
```

SNO	DEGREE
PB210000008	92
PB210000010	91
PB210000009	82
PB210000007	77

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 select distinct score.CNO, course.NAME, avg(score.DEGREE) as ave_degree
2 from score, course
3 where score.CNO=course.CNO and course.TYPE=1
4 group by score.CNO, course.NAME;
```

Lab 1

CNO	NAME	ave_degree
20230410	Artificial_Intelligence	81.5000
20230416	Computer_Network	81.7143
20230412	Data_Mining	84.6250
20230402	DB_Design	86.8750
20230420	Deep_Learning	86.5000
20230400	Linear_Algebra	85.5000
20230404	Machine_Learning	80.0000
20230408	Natural_Language_Processing	86.0000
20230406	Operating_System	83.5000
20230418	Pattern_Recognition	85.1429

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 select distinct score.CNO, course.NAME, avg(score.DEGREE) as ave_degree
2 from score, course
3 where score.CNO=course.CNO and course.TYPE=1
4 group by score.CNO;
```

CNO	NAME	ave_degree
20230400	Linear_Algebra	85.5000
20230402	DB_Design	86.8750
20230404	Machine_Learning	80.0000
20230406	Operating_System	83.5000
20230410	Artificial_Intelligence	81.5000
20230412	Data_Mining	84.6250
20230416	Computer_Network	81.7143
20230418	Pattern_Recognition	85.1429

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 select distinct student.SNO, student.NAME
2 from student
3 where not exists
4   (select *
5    from course
6     where course.TNO='TA90023' and not exists
7       (select *
8        from score
9         where student.SNO=score.SNO and score.CNO=course.CNO));
```

SNO	NAME
PB210000001	YH
PB210000002	ZY
PB210000003	FWJ
PB210000006	HCC

Lab 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 select course.CNO, course.NAME, max(score.DEGREE), min(score.DEGREE), max(score.DEGREE)-min(
    score.DEGREE) as score_diffence
2 from score, course
3 where score.CNO=course.CNO
4 group by course.CNO;
```

CNO	NAME	max(score.DEGREE)	min(score.DEGREE)	score_diffence
20230400	Linear_Algebra	92	77	15
20230402	DB_Design	95	75	20
20230404	Machine_Learning	93	68	25
20230406	Operating_System	89	78	11
20230408	Natural_Language_Processing	95	74	21
20230410	Artificial_Intelligence	97	72	25
20230412	Data_Mining	99	74	25
20230416	Computer_Network	97	49	48
20230418	Pattern_Recognition	95	69	26
20230420	Deep_Learning	95	80	15

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 select CNO, count(CNO) as num
2 from score
3 where DEGREE<75
4 group by CNO;
```

CNO	num
20230404	3
20230408	1
20230410	1
20230412	1
20230416	1
20230418	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 select distinct teacher.TNO, teacher.NAME
2 from teacher, score, course
3 where teacher.TNO=course.TNO
4 and exists(
5     select *
6     from score
7     where course.CNO=score.CNO and score.DEGREE<75);
```

TNO	NAME
TA90029	ZR
TA90027	XR
TA90025	HTZ
TA90023	ZDH
TA90021	HMZ

Lab 1

3.3.18 查询选修少于 2 门课程的学生的学号、姓名

```
1 select student.SNO, student.NAME
2 from student, score
3 where student.SNO=score.SNO
4 group by score.SNO
5 having count(score.SNO)<2;
```

SNO	NAME
PB210000009	ZMS
PB210000012	ADN
PB210000013	HD
PB210000014	GNJ
PB210000016	LC
PB210000017	TX
PB210000022	MSW
PB210000023	HXY
PB210000024	YHS
PB210000025	YWB
PB22061177	WYB
PB22081558	LYX
SA210110021	QXY
SA210110023	DPC

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 select distinct student.SNO, student.NAME
2 from student
3 where not exists
4   (select *
5     from score score1
6     where score1.SNO='PB210000001' and not exists
7       (select *
8         from score score2
9         where student.SNO=score2.SNO and score1.CNO=score2.CNO));
```

SNO	NAME
PB210000001	YH

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 select distinct course.NAME, avg(score.DEGREE) as ave_degree
2 from score
3 left join course on score.CNO=course.CNO
4 group by score.CNO;
```

Lab 1

NAME	ave_degree
Artificial_Intelligence	81.5000
Computer_Network	81.7143
Data_Mining	84.6250
DB_Design	86.8750
Deep_Learning	86.5000
Linear_Algebra	85.5000
Machine_Learning	80.0000
Natural_Language_Processing	86.0000
Operating_System	83.5000
Pattern_Recognition	85.1429
Signal_Control	NULL

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 select student.DEPART, count(distinct student.SNO) as num_student,
2    avg(case when score.DEGREE is not null then score.DEGREE else null end) as avg_score
3 from student
4 left join score
5 on student.SNO=score.SNO
6 group by student.DEPART;

```

DEPART	num_student	avg_score
10	2	81.2500
11	7	83.2727
12	6	85.3333
13	2	82.1667
14	3	85.6000
229	8	85.0000

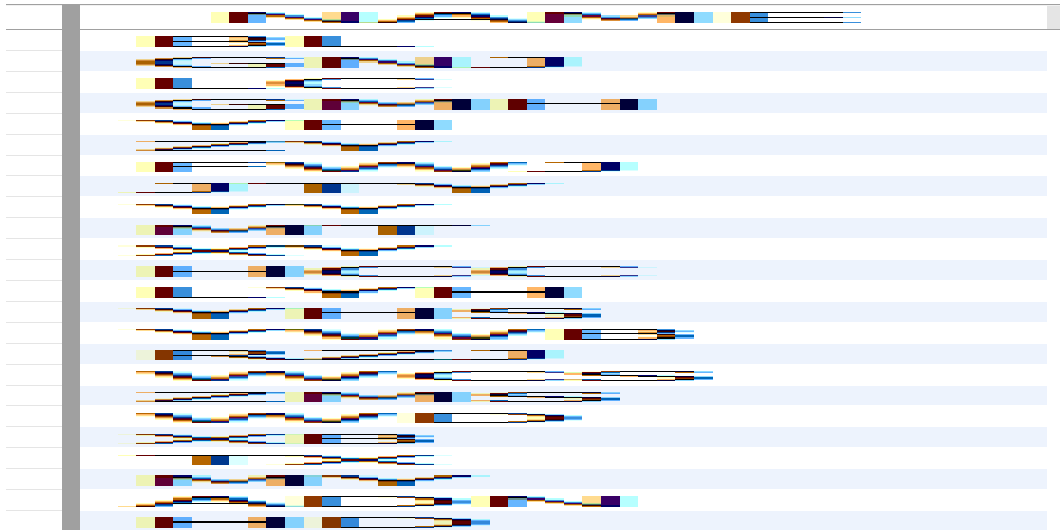
3.3.22 查询所有未选修 DB_Design 课程或者 Data_Mining 课程的学生姓名 (去掉重复行)

```

1 select distinct student.NAME
2 from student, score, course
3 where student.SNO=score.SNO and score.CNO=course.CNO and course.NAME!='DB_Design' and course
   .NAME!='Data_Mining';

```

Lab 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 select course.NAME, min(year(current_date()-year(student.BIRTHDAY)) as Minage, max(year(  
    current_date()-year(student.BIRTHDAY)) as Maxage, avg(year(current_date()-year(student  
    .BIRTHDAY)) as aveage  
2 from score  
3 left join student  
4 on score.SNO=student.SNO  
5 left join course  
6 on course.CNO=score.CNO  
7 group by course.NAME;
```

NAME	Minage	Maxage	Aveage
Artificial_Intelligence	20	23	21.5000
Computer_Network	22	23	22.4286
Data_Mining	20	23	21.7500
DB_Design	22	23	22.2500
Deep_Learning	21	23	21.8333
Linear_Algebra	21	22	21.7500
Machine_Learning	22	23	22.3333
Natural_Language_Processing	21	23	22.0000
Operating_System	20	22	21.0000
Pattern_Recognition	22	23	22.4286
Signal_Control	NULL	NULL	NULL

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 select student.SNO, student.NAME  
2 from score, student, course  
3 where score.SNO=student.SNO and course.CNO=score.CNO and course.NAME like 'Computer%';
```

Lab 1

SNO	NAME
PB2100000001	YH
PB2100000002	ZY
PB2100000003	FWJ
PB2100000004	JTY
PB2100000005	YY
PB2100000019	MT
PB2100000020	XY

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 select score1.*
2 from score score1
3 where abs((select avg(DEGREE) from score where CNO=score1.CNO) - score1.DEGREE) <= 12;

```

Sno	CNO	DEGREE
PB210000025	20230420	87
PB210000024	20230420	80
PB210000021	20230418	83
PB210000021	20230420	83
PB210000020	20230412	83
PB210000020	20230418	85
PB210000020	20230420	85
PB210000020	20230408	89
PB210000020	20230416	89
PB210000019	20230416	75
PB210000019	20230412	81
PB210000019	20230408	91
PB210000018	20230412	85
PB210000018	20230408	95
PB210000017	20230408	81
PB210000016	20230410	72
PB210000015	20230408	74
PB210000014	20230410	82
PB210000013	20230406	89
PB210000012	20230406	78
PB210000011	20230410	75
PB210000011	20230406	84
PB210000010	20230406	83
PB210000010	20230400	91
PB210000009	20230400	82
PB210000008	20230412	77
PB210000008	20230402	81
PB210000008	20230400	92
PB210000007	20230400	77
PB210000007	20230402	78

Lab 1

PB210000006	20230402	75
PB210000006	20230420	89
PB210000006	20230418	90
PB210000005	20230404	72
PB210000005	20230412	74
PB210000005	20230416	89
PB210000005	20230402	93
PB210000004	20230404	68
PB210000004	20230416	86
PB210000004	20230402	95
PB210000003	20230418	82
PB210000003	20230402	90
PB210000003	20230404	92
PB210000002	20230404	82
PB210000002	20230416	87
PB210000002	20230402	94
PB210000002	20230418	95
PB210000001	20230404	73
PB210000001	20230412	80
PB210000001	20230402	89
PB210000001	20230418	92
PB210000001	20230420	95

3.4 视图

3.4.1 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

```
1 create view db_229_student as
2 select *
3 from student
4 where DEPART=229
5 with check option;
6
7 select *
8 from db_229_student;
```

Sno	NAME	GENDER	BIRTHDAY	DEPART
PB210000001	YH	male	2002-03-29 00:00:00	229
PB210000006	HCC	male	2002-06-25 00:00:00	229
PB210000020	XY	female	2001-02-14 00:00:00	229
PB210000021	LYH	male	2002-09-30 00:00:00	229
PB210000024	YHS	female	2003-04-01 00:00:00	229
PB210000025	YWB	male	2003-08-12 00:00:00	229
PB22061161	GMH	male	2004-07-25 00:00:00	229
PB22081558	LYX	female	2004-07-07 00:00:00	229

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 update db_229_student
2 set NAME='G'
3 where SNO='PB210000020';
```

Lab 1

```
4
5 select *
6 from db_229_student;
```

Sno	NAME	GENDER	BIRTHDAY	DEPART
PB210000001	YH	male	2002-03-29 00:00:00	229
PB210000006	HCC	male	2002-06-25 00:00:00	229
PB210000020	G	female	2001-02-14 00:00:00	229
PB210000021	LYH	male	2002-09-30 00:00:00	229
PB210000024	YHS	female	2003-04-01 00:00:00	229
PB210000025	YWB	male	2003-08-12 00:00:00	229
PB22061161	GMH	male	2004-07-25 00:00:00	229
PB22081558	LYX	female	2004-07-07 00:00:00	229

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 select SNO, NAME
2 from db_229_student
3 where year(current_date())-year(BIRTHDAY) < 22;
```

Sno	NAME
PB210000024	YHS
PB210000025	YWB
PB22061161	GMH
PB22081558	LYX

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 insert into student (SNO, NAME, GENDER, BIRTHDAY, DEPART)
2 values
3 ('SA210110021', 'QXY', 'female', '2007-07-27', '229');
4
```

Lab 1

```
5 select *
6 from db_229_student;
```

Sno	NAME	GENDER	BIRTHDAY	DEPART
PB210000001	YH	male	2002-03-29 00:00:00	229
PB210000006	HCC	male	2002-06-25 00:00:00	229
PB210000020	G	female	2001-02-14 00:00:00	229
PB210000021	LYH	male	2002-09-30 00:00:00	229
PB210000024	YHS	female	2003-04-01 00:00:00	229
PB210000025	YWB	male	2003-08-12 00:00:00	229
PB22061161	GMH	male	2004-07-25 00:00:00	229
PB22081558	LYX	female	2004-07-07 00:00:00	229
SA210110021	QXY	female	2007-07-27 00:00:00	229

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 insert into db_229_student (SNO, NAME, GENDER, BIRTHDAY, DEPART)
2 values
3 ('SA210110023', 'DPC', '男', '1997-04-27', '11');
```

176 23:04:16 insert into db_229_student(SNO, NAME, GENDER, BIRTHDAY, DEPART) values ('SA210110023', 'DPC', '男', '1997-04-27', '11') Error Code: 1369, CHECK OPTION failed 't_c_s.db_229_student' 0.016 sec

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 create table teacher_sal
2 (TNO char(7) primary key,
3 SAL float
4 );
```

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”。

```
1 DELIMITER //
2 create trigger insert_sal
3 before insert on teacher_sal
4 for each row
5 begin
```


Lab 1

```
6  declare TNO_num int;
7  select count(*) into TNO_num
8      from teacher
9      where TNO = new.TNO;
10
11  if TNO_num = 0 then
12      signal sqlstate '45000'
13          set message_text = '工号必须存在于teacher表中';
14  end if;
15 end;
16 //
17 DELIMITER ;
18
19 show create trigger insert_sal;
20
21 insert into teacher_sal (TNO, SAL)
22 values
23 ('TA90021', 6000.4);
24 insert into teacher_sal (TNO, SAL)
25 values
26 ('TA12345', 4900.2);
27
28 DELIMITER //
29 create trigger update_sal
30 before update on teacher_sal
31 for each row
32 begin
33     declare TNO_num int;
34     select count(*) into TNO_num
35         from teacher
36         where TNO = new.TNO;
37
38     if TNO_num = 0 then
39         signal sqlstate '45000'
40             set message_text = '工号必须存在于teacher表中';
41     end if;
42 end;
43 //
44 DELIMITER ;
45
46 show create trigger update_sal;
47
48 update teacher_sal
49 set TNO = 'TA12345'
50 where TNO = 'TA90021';
```

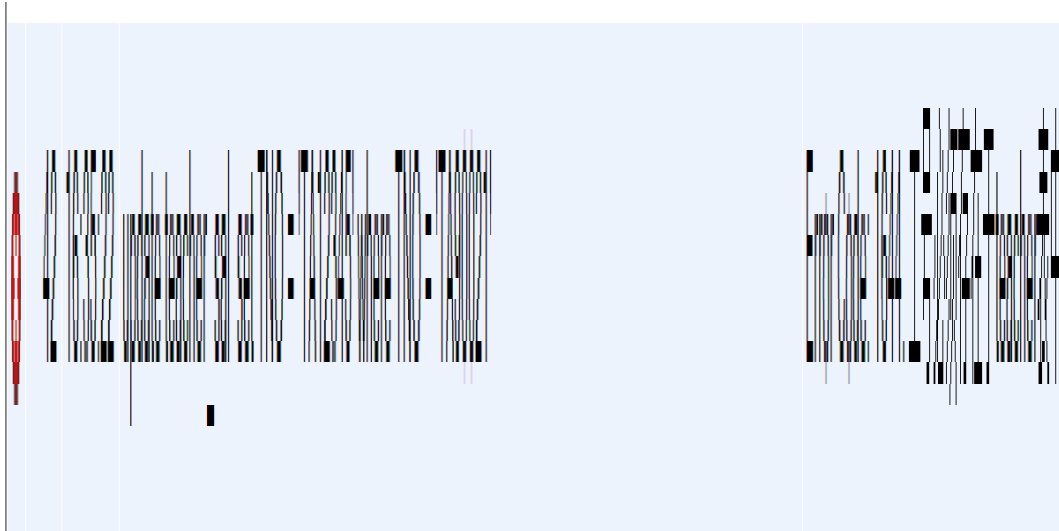
Lab 1

Trigger	SQL mode	SQL Original Statement	character set client	collation connection	Database Collation	Created
insert.sql	ONLY_FULL_GROUP_BY,STRICT_TRANS...	CREATE DEFINER='root'@'localhost'...	utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0900_ai_ci	2024-04-26 16:08:49.70

--	--	--	--	--	--	--

Trigger	SQL mode	SQL Original Statement	character set client	collation connection	Database Collation	Created
update.sql	ONLY_FULL_GROUP_BY,STRICT_TRANS...	CREATE DEFINER='root'@'localhost'...	utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0900_ai_ci	2024-04-26 16:13:46.15

Lab 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 DELIMITER //
2 create trigger insert2_sal
3 before insert on teacher_sal
4 for each row
5 begin
6     declare pos varchar(30);
7     select POSITION into pos
8     from teacher
9     where teacher.TNO = NEW.TNO;
10
11 if new.SAL < 4000 and pos = 'Instructor' then
12     set new.SAL = 4000;
13 elseif new.sal < 7000 and pos = 'Associate professor' then
14     set new.sal = 7000;
15 elseif new.sal < 10000 and pos = 'Professor' then
16     set new.sal = 10000;
17 elseif new.sal > 7000 and pos = 'Instructor' then
18     set new.sal = 7000;
19 elseif new.sal > 10000 and pos = 'Associate professor' then
20     set new.sal = 10000;
21 elseif new.sal > 13000 and pos = 'Professor' then
22     set new.sal = 13000;
23 end if;
24 end;
25 //
26 DELIMITER ;
27
28 show create trigger update_sal;
29
30 insert into teacher_sal (TNO, SAL)
31 values
32 ('TA90022', 6500.4),
33 ('TA90025', 9500.4),
34 ('TA90023', 3500.4),
```

Lab 1

```
35 ('TA90028', 7500.4),
36 ('TA90031', 10500.4),
37 ('TA90030', 13500.4);
38
39 select * from teacher_sal;
40
41 DELIMITER //
42 create trigger update2_sal
43 before update on teacher_sal
44 for each row
45 begin
46     declare pos varchar(30);
47     select POSITION into pos
48     from teacher
49     where teacher.TNO = NEW.TNO;
50
51     if new.SAL < 4000 and pos = 'Instructor' then
52         set new.SAL = 4000;
53     elseif new.sal < 7000 and pos = 'Associate professor' then
54         set new.sal = 7000;
55     elseif new.sal < 10000 and pos = 'Professor' then
56         set new.sal = 10000;
57     elseif new.sal > 7000 and pos = 'Instructor' then
58         set new.sal = 7000;
59     elseif new.sal > 10000 and pos = 'Associate professor' then
60         set new.sal = 10000;
61     elseif new.sal > 13000 and pos = 'Professor' then
62         set new.sal = 13000;
63     end if;
64 end;
65 //
66 DELIMITER ;
67
68 update teacher_sal
69 set SAL = '6500'
70 where TNO = 'TA90022';
71
72 update teacher_sal
73 set SAL = '9500'
74 where TNO = 'TA90025';
75
76 update teacher_sal
77 set SAL = '7500'
78 where TNO = 'TA90028';
79
80 update teacher_sal
81 set SAL = '3500'
82 where TNO = 'TA90023';
83
84 update teacher_sal
85 set SAL = '10500'
86 where TNO = 'TA90031';
87
88 update teacher_sal
89 set SAL = '13500'
90 where TNO = 'TA90030';
91
92 select * from teacher_sal;
```

Lab 1

Trigger	sql_mode	SQL Original Statement	character_set_client	collation_connection	Database Collation	Created
insert2_sql	ONLY_FULL_GROUP_BY,STRICT_TRANS...	CREATE DEFINER='root'@'localhost'...	utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0900_ai_ci	2024-04-27 09:39:13.70

TNO	SAL
TA900021	6000.4
TA900022	7000
TA900023	4000
TA900025	10000
TA900028	7000
TA900030	13000
TA900031	10000
NULL	NULL

Trigger	sql_mode	SQL Original Statement	character_set_client	collation_connection	Database Collation	Created
update2_sql	ONLY_FULL_GROUP_BY,STRICT_TRANS...	CREATE DEFINER='root'@'localhost'...	utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0900_ai_ci	2024-04-27 09:41:54.94

Lab 1

TNO	SAL
TA900021	6000.4
TA900022	7000
TA900023	4000
TA900025	10000
TA900028	7000
TA900030	13000
TA900031	10000
NULL	NULL

3.5.4 删除刚刚创建的所有触发器

```
1 drop trigger insert_sal;
2 drop trigger update_sal;
3 drop trigger insert2_sal;
4 drop trigger update2_sal;
```

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 update score
2 set DEGREE = NULL
3 where CNO in
4   (select CNO
5     from Course
6     where NAME = 'Data_Mining'
7   );
8
9 select SNO, DEGREE
10 from score
11 order by DEGREE;
```

Lab 1

SNO	DEGREE
PB22061161	NULL
PB210000001	NULL
PB210000020	NULL
PB210000018	NULL
PB210000005	NULL
PB210000019	NULL
PB210000008	NULL
PB210000015	NULL
PB210000001	49
PB210000004	68
PB210000008	69
PB210000005	72
PB210000016	72
PB210000001	73
PB210000015	74
PB210000011	75
PB210000006	75
PB210000019	75
PB210000007	77
PB210000012	78
PB210000007	78
PB210000024	80
PB210000008	81
PB210000017	81
PB210000009	82
PB210000003	82
PB210000002	82
PB210000014	82
PB210000021	83
PB210000021	83

PB210000010	83
PB210000011	84
PB210000020	85
PB210000020	85
PB210000004	86
PB210000025	87
PB210000002	87
PB210000005	89
PB210000020	89
PB210000020	89
PB210000006	89
PB210000013	89
PB210000001	89
PB210000006	90
PB210000003	90
PB210000019	91
PB210000010	91
PB210000008	92
PB210000003	92
PB210000001	92
PB210000006	93
PB210000005	93
PB210000002	94
PB210000004	95
PB210000018	95
PB210000002	95
PB210000001	95
PB210000003	97
PB22061161	97

3.7 开放题

3.7.1 查询每个学生的选课数量

```
1 select student.SNO, student.NAME, count(score.DEGREE) as num
2 from student
3 left join score
4 on student.SNO = score.SNO
5 group by student.Sno;
```

Lab 1

SNO	NAME	num
PB210000001	YH	5
PB210000002	ZY	4
PB210000003	FWJ	4
PB210000004	JTY	3
PB210000005	YY	4
PB210000006	HCC	4
PB210000007	RZJ	2
PB210000008	WCS	3
PB210000009	ZMS	1
PB210000010	WD	2
PB210000011	BL	2
PB210000012	ADN	1
PB210000013	HD	1
PB210000014	GNJ	1
PB210000015	XB	1
PB210000016	LC	1
PB210000017	TX	1
PB210000018	MY	1
PB210000019	MT	2
PB210000020	G	4
PB210000021	LYH	2
PB210000022	MSW	0
PB210000023	HXY	0
PB210000024	YHS	1
PB210000025	YWB	1
PB22061161	GMH	1
PB22061177	WYB	0
PB22081558	LYX	0
SA210110021	QXY	0
SA210110023	DPC	0

3.7.2 查询没有选 GMH 选过的任意一门课的学生名字

```
1 select distinct NAME
2 from student, score
3 where student.SNO = score.Sno and score.CNO not in
4     (select CNO
5      from student1, score score1
6      where student1.NAME = 'GMH' and student1.Sno = score1.Sno);
```

NAME
YH
ZY
FWJ
JTY
YY
HCC
RZJ
WCS
ZMS
WD
BL
ADN
HD
XB
TX
MY
MT
G
LYH
YHS
YWB

4 Conclusion