数据库第三次上机

> 22373386 高铭

TASK 2

关系模式1

职员（职员ID，姓名，性别，出生年月，职级，月薪，部门ID）(部门ID引用部门表的主键)
职员考勤（职员ID，出勤日期时间）（职员ID引用职员表的主键）
部门（部门ID，部门名称，部门经理ID）（部门经理ID引用职员表的主键）
监理（监理员ID，监理姓名）
工程（工程ID，工程工期，工程预算）（工程工期存的是天数，int型）
工程实施（工程ID，部门ID）（工程ID、部门ID分别引用工程、部门表的主键。）
工程监理（工程ID，监理员ID）（工程ID、监理员ID分别引用工程、监理表主键）

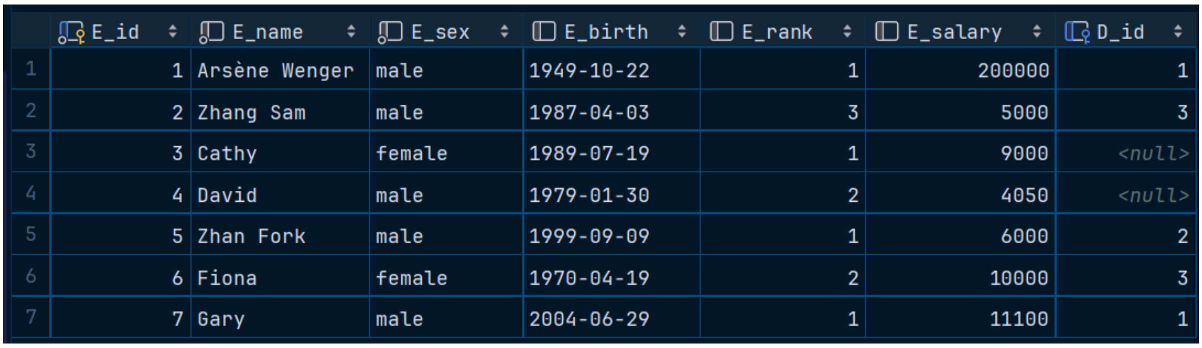
```sql
# 建立表格
create schema lab3_1 collate utf8mb4_0900_ai_ci;
# utf8mb4_0900_ai_ci - MySQL针对utf8mb4编码的排序规则：不区分音调、不区分大小写
create table Department(
    D_id int NOT NULL PRIMARY KEY,
    D_name varchar(255) NOT NULL,
    D_manager_id int NOT NULL,
);
create table Employee(
    E_id int NOT NULL PRIMARY KEY,
    E_name varchar(255) NOT NULL,
    E_sex char(10) NOT NULL,
    E_birth date,
    E_rank int,
    E_salary int,
    D_id int NOT NULL,
    constraint fk_Employee_D_id foreign key (D_id)
        references Department(D_id)
);

alter table Department add
    constraint fk_Department_E_id foreign key (D_manager_id)
            references Employee(E_id);

create table Attendance(
    E_id int NOT NULL PRIMARY KEY,
    A_time time NOT NULL PRIMARY KEY,
    constraint fk_Attendance_E_id foreign key (E_id)
        references Employee(E_id)
);
create table Supervisor(
    S_id int not null primary key,
    S_name varchar(255) not null
);
create table Project(
    P_id int not null primary key,
    P_schedule int not null ,
    P_budget int
);
create table P_Implement(
    P_id int not null primary key,
    D_id int NOT NULL PRIMARY KEY,
    constraint fk_ProjectImplement_D_id foreign key (D_id)
        references Department(D_id),
    constraint fk_ProjectImplement_P_id foreign key (P_id)
        references Project(P_id)
);
create table P_Supervision(
    P_id int not null primary key,
    S_id int not null primary key,
    constraint fk_P_id foreign key (P_id)
        references Project(P_id),
    constraint fk_S_id foreign key (S_id)
        references Supervisor(S_id)
);
```

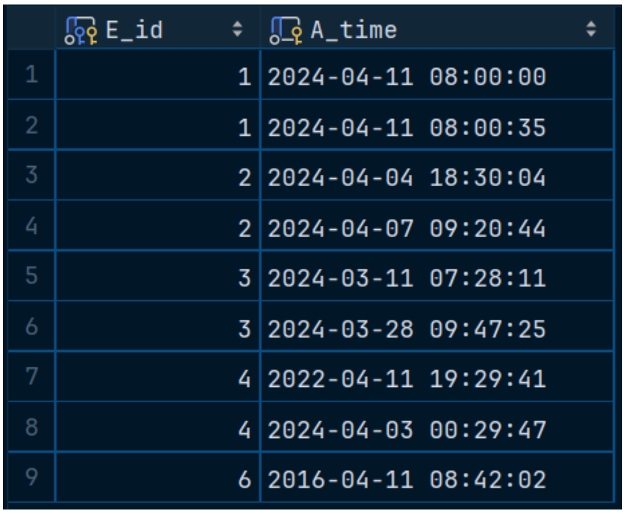
各表数据展示

employee



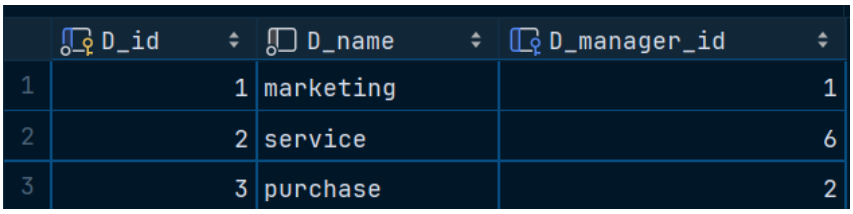
	E_id	E_name	E_sex	E_birth	E_rank	E_salary	D_id
1	1	Arsène Wenger	male	1949-10-22	1	200000	1
2	2	Zhang Sam	male	1987-04-03	3	5000	3
3	3	Cathy	female	1989-07-19	1	9000	<null>
4	4	David	male	1979-01-30	2	4050	<null>
5	5	Zhan Fork	male	1999-09-09	1	6000	2
6	6	Fiona	female	1970-04-19	2	10000	3
7	7	Gary	male	2004-06-29	1	11100	1

attendance



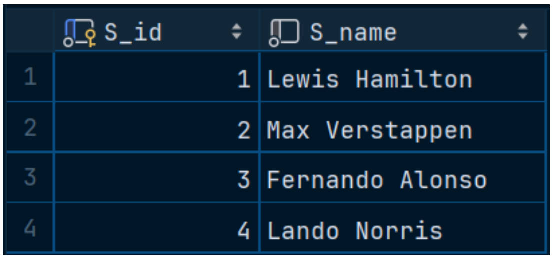
	E_id	A_time
1	1	2024-04-11 08:00:00
2	1	2024-04-11 08:00:35
3	2	2024-04-04 18:30:04
4	2	2024-04-07 09:20:44
5	3	2024-03-11 07:28:11
6	3	2024-03-28 09:47:25
7	4	2022-04-11 19:29:41
8	4	2024-04-03 00:29:47
9	6	2016-04-11 08:42:02

department



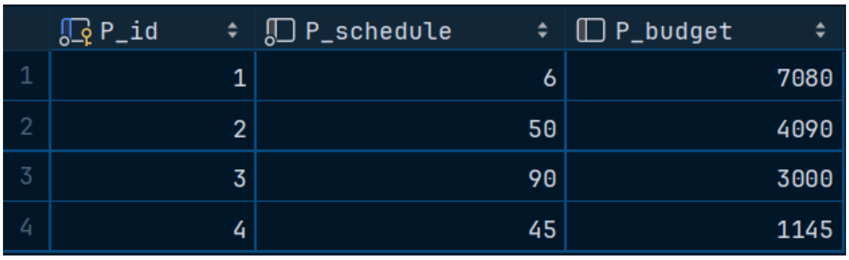
	D_id	D_name	D_manager_id
1	1	marketing	1
2	2	service	6
3	3	purchase	2

supervisor



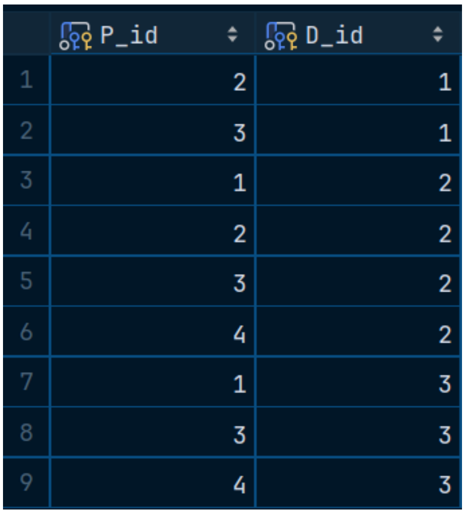
	S_id	S_name
1	1	Lewis Hamilton
2	2	Max Verstappen
3	3	Fernando Alonso
4	4	Lando Norris

project



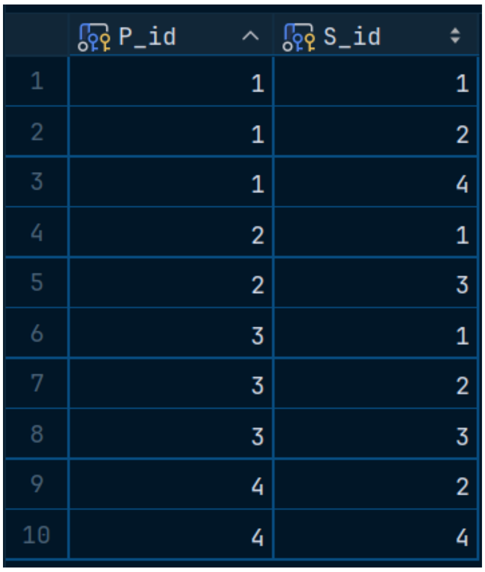
	P_id	P_schedule	P_budget
1	1	6	7080
2	2	50	4090
3	3	90	3000
4	4	45	1145

p_implement



	P_id	D_id
1	2	1
2	3	1
3	1	2
4	2	2
5	3	2
6	4	2
7	1	3
8	3	3
9	4	3

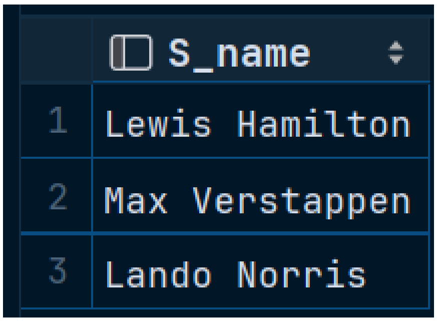
p_supervision



	P_id	S_id
1	1	1
2	1	2
3	1	4
4	2	1
5	2	3
6	3	1
7	3	2
8	3	3
9	4	2
10	4	4

1-1. 查找监理过工程ID为1的监理姓名。

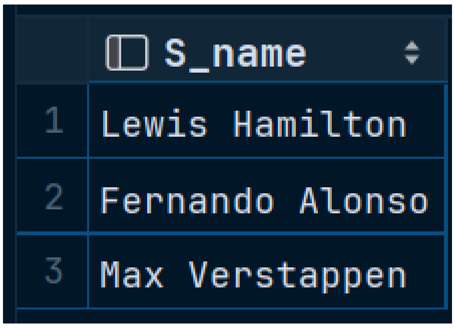
```sql
select supervisor.S_name
from supervisor,P_Supervision
where supervisor.S_id = P_Supervision.S_id and P_Supervision.P_id = 1
```



	S_name
1	Lewis Hamilton
2	Max Verstappen
3	Lando Norris

1-2. 查询监理过部门ID为1的部门干过的工程的监理姓名。

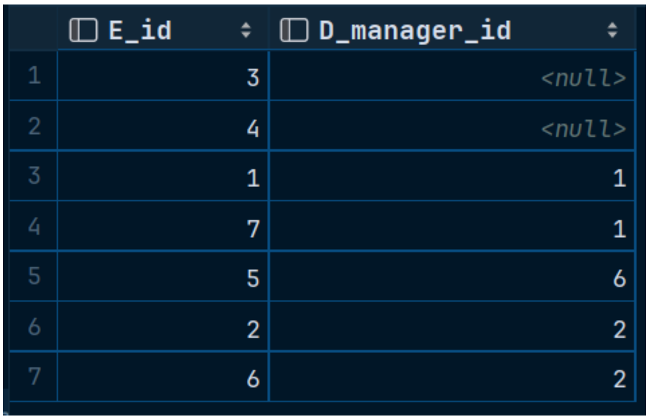
```sql
select s.S_name
from supervisor as s, p_supervision as ps, p_implement as pi
where s.S_id = ps.S_id and ps.P_id = pi.P_id and pi.D_id = 1
group by s.S_name
```



	S_name
1	Lewis Hamilton
2	Fernando Alonso
3	Max Verstappen

1-3. 查询所有职员ID及他们的经理ID（注意有的职员可能没有部门）。

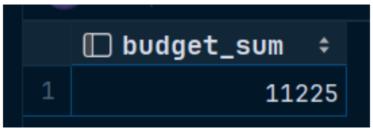
```sql
select e.E_id, d.D_manager_id
from employee as e
left join department d on e.D_id = d.D_id
```



	E_id	D_manager_id
1	3	<null>
2	4	<null>
3	1	1
4	7	1
5	5	6
6	2	2
7	6	2

1-4. 查询所有Zhang姓员工参与的工程的总预算。

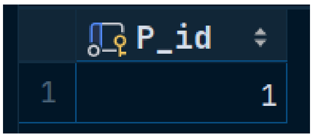
```sql
select sum(p0.P_budget) budget_sum
from project p0
where p0.P_budget in (
    select p.P_budget
    from project p, employee e, p_implement pi
    where e.E_name like 'Zhang%' and p.P_id = pi.P_id and e.D_id = pi.D_id
    group by p.P_id
)
```



	budget_sum
1	11225

1-5. 查询工程预算比所有工程工期大于10天的工程都要多的工程ID。

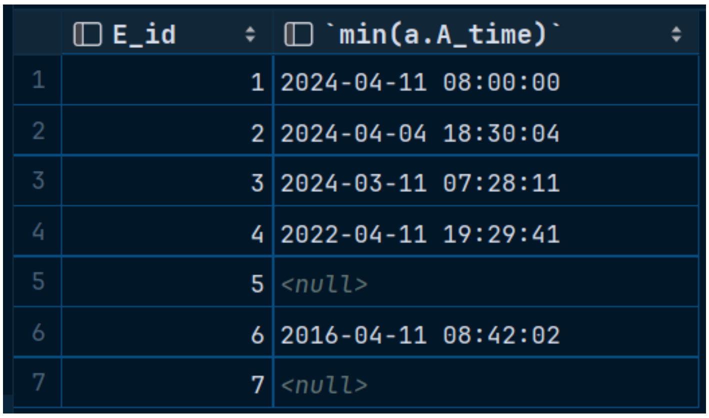
```sql
select p1.P_id
from project p1
where p1.P_budget > ALL (
    select p2.P_budget
    from project p2
    where p2.P_schedule > 10
)
```



	P_id
1	1

1-6. 查询所有职员最早的考勤记录。（给出查询结果：职员ID，最早考勤时间）

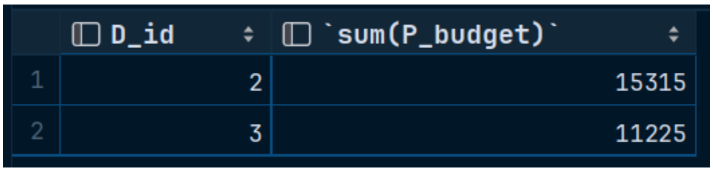
```sql
select e.E_id, min(a.A_time)
from employee e
left join attendance a on e.E_id = a.E_id
group by e.E_id
```



	E_id	`min(a.A_time)`
1	1	2024-04-11 08:00:00
2	2	2024-04-04 18:30:04
3	3	2024-03-11 07:28:11
4	4	2022-04-11 19:29:41
5	5	<null>
6	6	2016-04-11 08:42:02
7	7	<null>

1-7. 查询参加过的工程的总预算额在10000以上的部门ID，及其预算额。

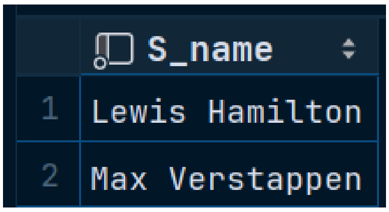
```sql
select d.D_id, sum(P_budget)
from department d, project p, p_implement pi
where d.D_id = pi.D_id and p.P_id = pi.P_id
group by d.D_id
having sum(P_budget) > 10000
```



	D_id	`sum(P_budget)`
1	2	15315
2	3	11225

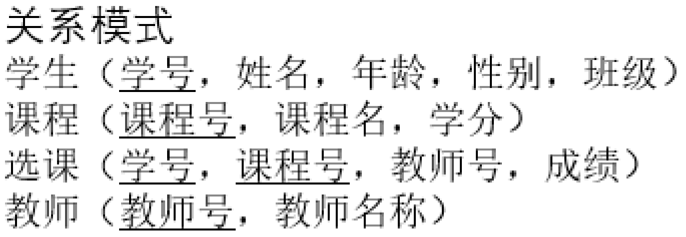
1-8. 请查询至少监理了三个工程的监理姓名。

```sql
select S_name
from supervisor s, p_supervision ps
where s.S_id = ps.S_id
group by s.S_id, s.S_name
having count(p_id) >= 3
```



	S_name
1	Lewis Hamilton
2	Max Verstappen

关系模式2


关系模式
学生（学号，姓名，年龄，性别，班级）
课程（课程号，课程名，学分）
选课（学号，课程号，教师号，成绩）
教师（教师号，教师名称）

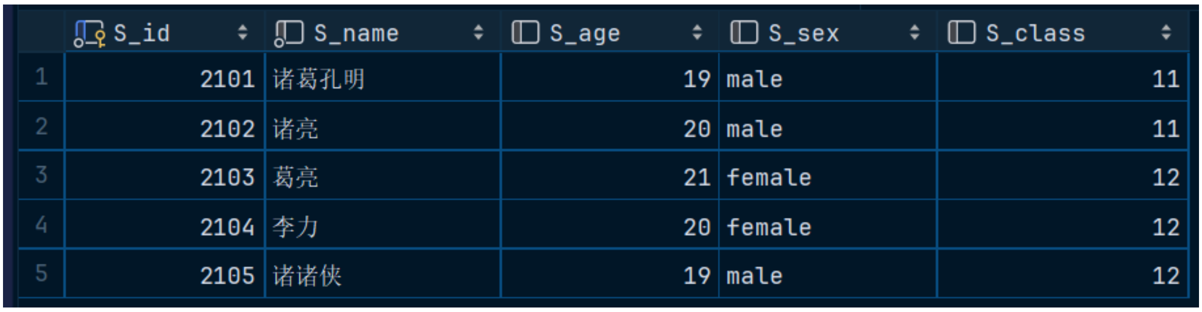
```sql
# 建立表格
create schema lab3_2
create table Student(
    S_id int not null primary key,
    S_name varchar(255) not null,
    S_age int,
    S_sex char(30),
    S_class int
);
create table Course(
    C_id int not null primary key,
    C_name varchar(255) not null,
    C_credit int
);
create table SelectCourse(
    S_id int not null,
    C_id int not null,
    T_id int,
    score int
);
alter table SelectCourse add primary key (S_id,C_id);

create table Teacher(
    T_id int not null primary key,
    T_name varchar(255) not null
);

alter table SelectCourse add foreign key (S_id)
    references Student(S_id);
alter table SelectCourse add foreign key (C_id)
    references Course(C_id);
alter table SelectCourse add foreign key (T_id)
    references Teacher(T_id);
```

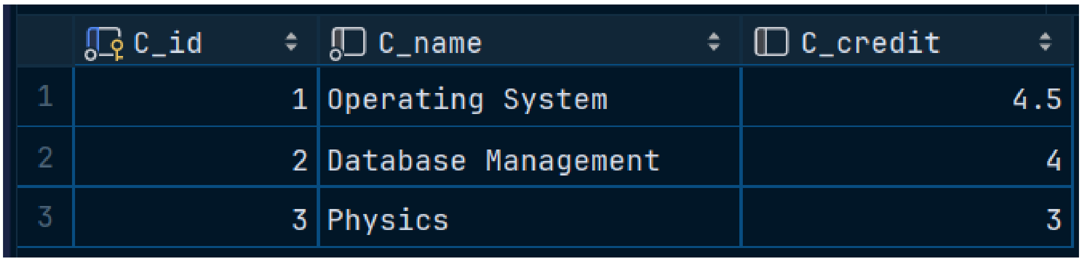
各表数据展示

student



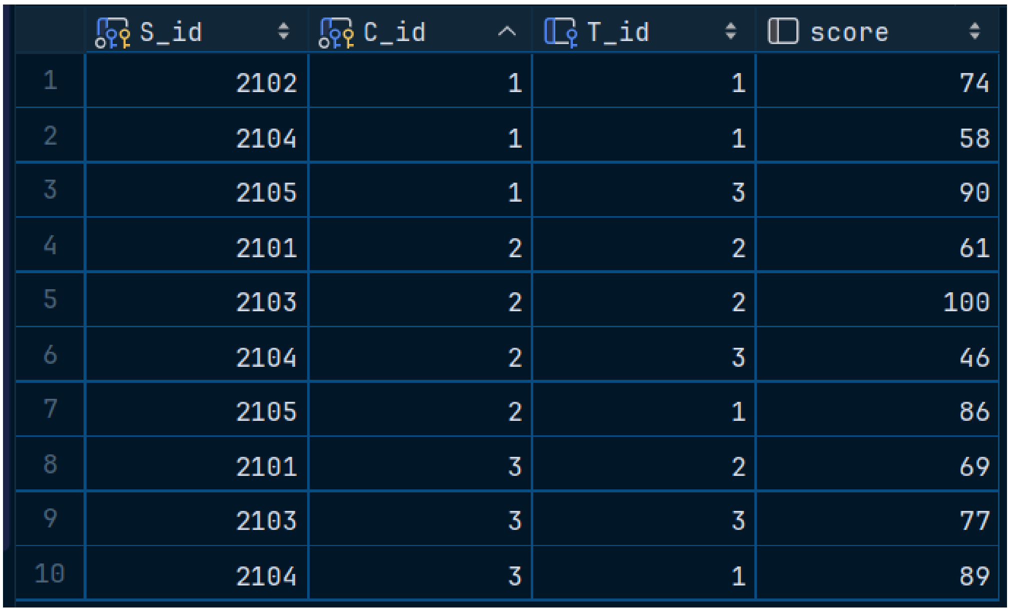
	S_id	S_name	S_age	S_sex	S_class
1	2101	诸葛孔明	19	male	11
2	2102	诸亮	20	male	11
3	2103	葛亮	21	female	12
4	2104	李力	20	female	12
5	2105	诸诸侠	19	male	12

course



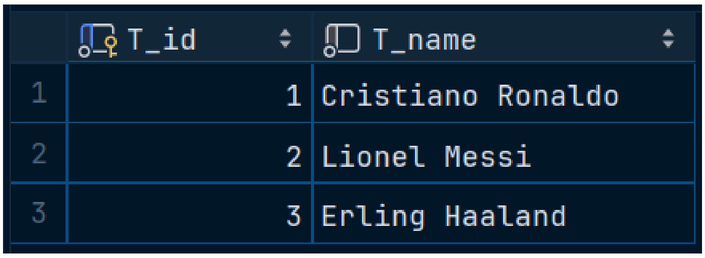
	C_id	C_name	C_credit
1	1	Operating System	4.5
2	2	Database Management	4
3	3	Physics	3

selectcourse



	S_id	C_id	T_id	score
1	2102	1	1	74
2	2104	1	1	58
3	2105	1	3	90
4	2101	2	2	61
5	2103	2	2	100
6	2104	2	3	46
7	2105	2	1	86
8	2101	3	2	69
9	2103	3	3	77
10	2104	3	1	89

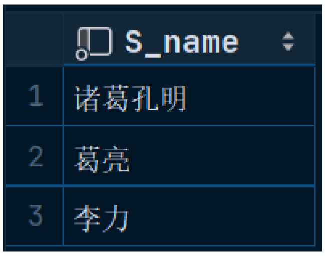
teacher



	T_id	T_name
1	1	Cristiano Ronaldo
2	2	Lionel Messi
3	3	Erling Haaland

2-1. 查找选修了物理课的学生姓名

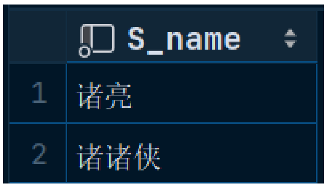
```sql
select s.S_name
from student s, course c, selectcourse sc
where s.S_id = sc.S_id and c.C_id = sc.C_id and c.C_name = 'Physics'
```



	S_name
1	诸葛孔明
2	葛亮
3	李力

2-2. 找出所有姓诸的学生姓名（排除姓‘诸葛’的学生）

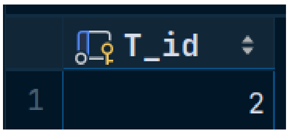
```sql
select S_name
from student s
where S_name like '诸%' and S_name not like '诸葛%'
```



	S_name
1	诸亮
2	诸诸侠

2-3. 查找教的学生的成绩都大于60分的教师（给出教师号即可）

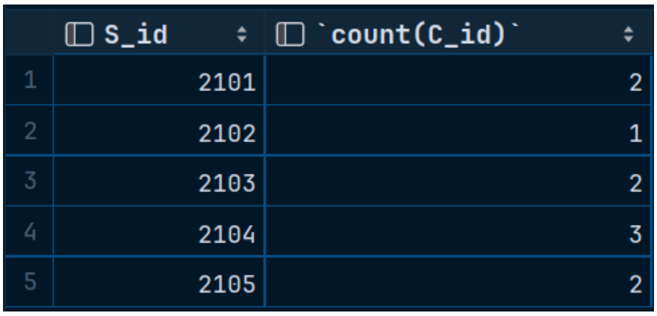
```sql
select t.T_id
from teacher t, selectcourse sc
where t.T_id = sc.T_id
group by t.T_id
having min(sc.score) >= 60
```



	T_id
1	2

2-4. 查询每个学生选修的课程数量，（给出查询结果：学号，选修课程数量）

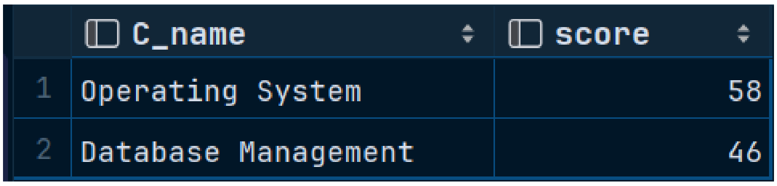
```sql
select s.S_id, count(C_id)
from student s
left join selectcourse sc on s.S_id = sc.S_id
group by s.S_id
```



	S_id	\`count(C_id)\`
1	2101	2
2	2102	1
3	2103	2
4	2104	3
5	2105	2

2-5. 查找李力的所有不及格的课程名称和成绩，按成绩降序排列

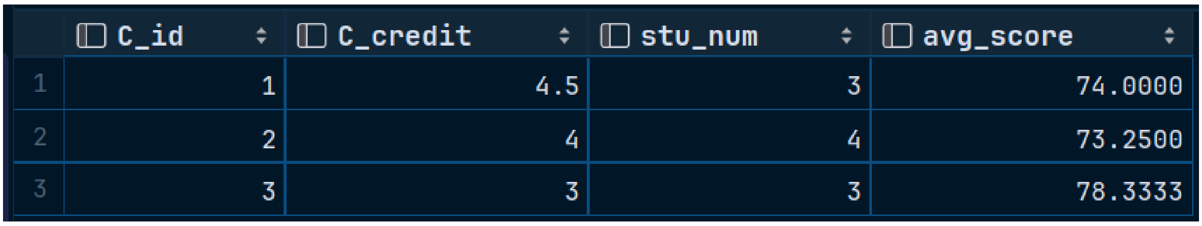
```sql
select c.C_name, sc.score
from course c, student s, selectcourse sc
where s.S_id = sc.S_id and c.C_id = sc.C_id and S_name = '李力' and score < 60
order by score DESC
```



	C_name	score
1	Operating System	58
2	Database Management	46

2-6. 列出每门课的学分，选修的学生人数，及学生成绩的平均分

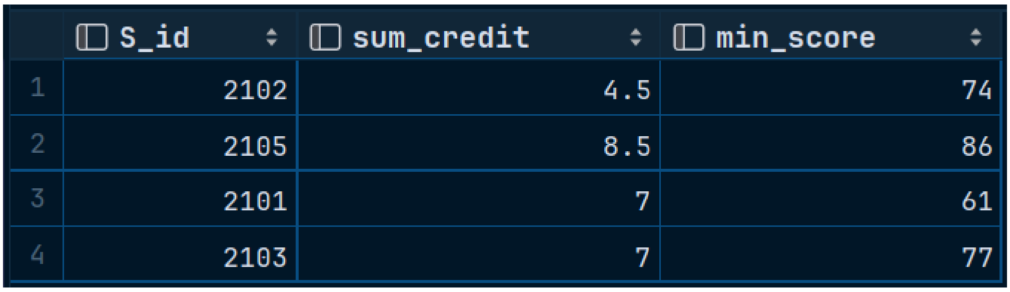
```sql
select c.C_id, C_credit, count(S_id) stu_num, avg(score) avg_score
from course c
left join selectcourse sc on c.C_id = sc.C_id
group by c.C_id, C_credit
```



	C_id	C_credit	stu_num	avg_score
1	1	4.5	3	74.0000
2	2	4	4	73.2500
3	3	3	3	78.3333

2-7. 选出所修课程总学分在10分以下的学生（注：不及格的课程没有学分）

```sql
select s.S_id, sum(c.C_credit) sum_credit, min(sc.score) min_score
from student s, selectcourse sc, course c
where s.S_id = sc.S_id and c.C_id = sc.C_id
group by s.S_id
having sum_credit < 10 and min_score > 60
```



	S_id	sum_credit	min_score
1	2102	4.5	74
2	2105	8.5	86
3	2101	7	61
4	2103	7	77