练习课（三）

正则表达式练习

1、匹配年月日日期 格式**2018-12-6**

^[1-9]\d{0,3}-(1[0-2]|0?[1-9])-(3[01]|[12]\d|0?[1-9])$

1.^[1-9]表示年是以数字1-9开头的,\d{0,3}表示年的位数,^[1-9]\d{0,3}就表示1-9999年之间

2.(1[0-2]|0?[1-9])中|前面的1[0-2]表示从10到12,后面的0?[1-9]表示01-09或者1-9,

(1[0-2]|0?[1-9])表示月,01-12或者1-12

3.(3[01]|[12]\d|0?[1-9])$其中3[01]表示30或31,[12]\d表示从10-29,最后的0?[1-9]表示从

01-09或者是从1-9.整体就表示从01-31或者1-31

2、长度为**8-10**位的用户密码 ： 包含数字字母下划线

\w{8,10}

3、匹配验证码：**4**位数字字母组成的

[\da-zA-Z]{4}或者**[0-9a-zA-Z]{4}**

[ ]里面的表示数字,或者a-z或者A-Z,{4}表示4位

4、从类似

```
<a>wahaha</a>
<b>banana</b>
<h1>qqxing</h1>
```

这样的字符串中，
1）匹配出**wahaha**，**banana**，**qqxing**内容。

\w{6}

>\w+<

2）匹配出**a,b,h1**这样的内容

<\w+>

5、**1-2*((60-30+(-40/5)*(9-2*5/3+7/3*99/4*2998+10*568/14))-(-4*3)/(16-3*2))**
1）从上面算式中匹配出最内层小括号以及小括号内的表达式

```
\([^()]+\)    \(和\)表示前后位( ),[^()]就表示外面的()里面没有()
```

6、从类似**9-2*5/3+7/3*99/4*2998+10*568/14**的表达式中匹配出从左到右第一个乘法或除法

\d+[*/]\d+　　*[/]前后的\d+表示*或/前面的整数,可能是多位数字,要加+

mysql面试题集锦

理清SQL语句的执行顺序

创建school数据库

```
create database school charset=utf8;
use school;
```

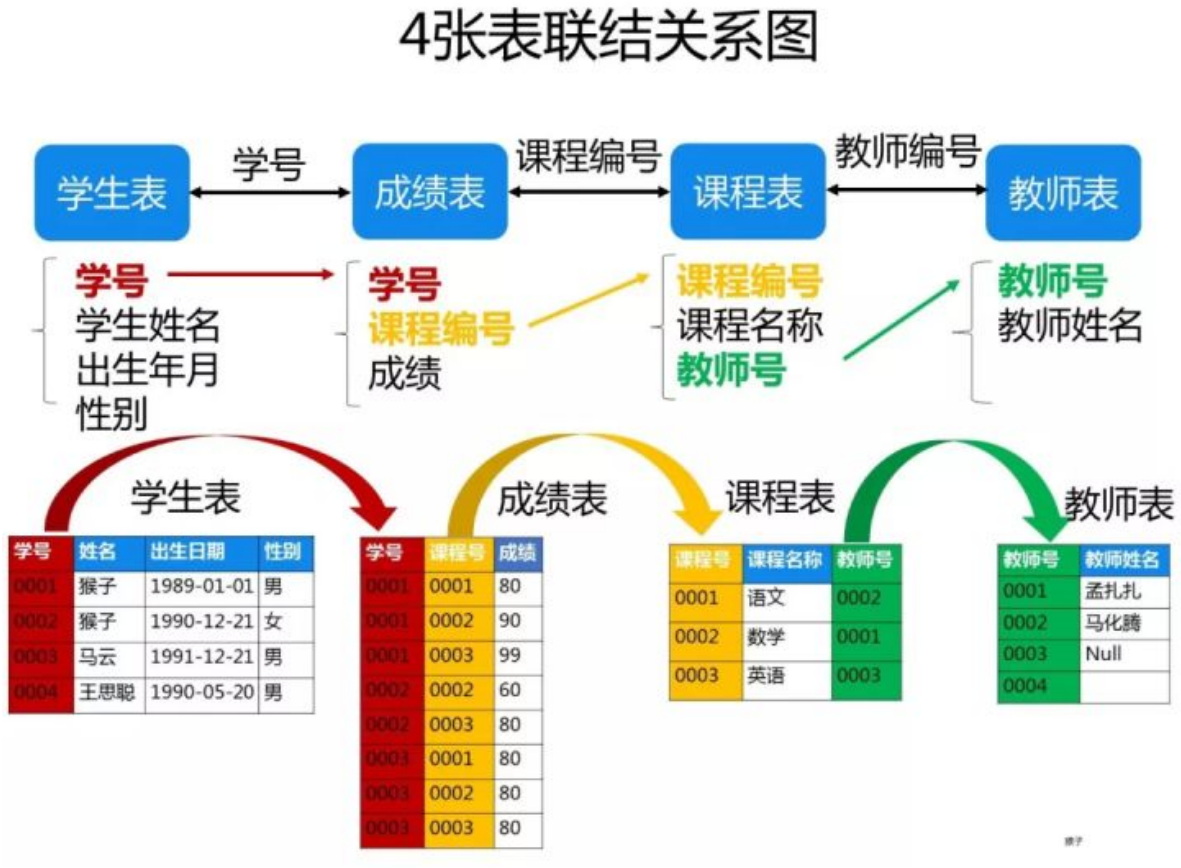
创建四张表

```
create table student(
    s_id varchar(10),
    s_name varchar(20),
    s_age date,
    s_sex varchar(10)
);

create table course(
    c_id varchar(10),
    c_name varchar(20),
    t_id varchar(10)
);


create table teacher (
t_id varchar(10),
t_name varchar(20)
);

create table score (
    s_id varchar(10),
    c_id varchar(10),
    score varchar(10)
);
```



往表里插值

```
insert into student (s_id, s_name, s_age, s_sex)
values  ('01' , '赵雷' , '1990-01-01' , '男'),
        ('02' , '钱电' , '1990-12-21' , '男'),
        ('03' , '孙风' , '1990-05-20' , '男'),
        ('04' , '李云' , '1990-08-06' , '男'),
        ('05' , '周梅' , '1991-12-01' , '女'),
        ('06' , '吴兰' , '1992-03-01' , '女'),
        ('07' , '郑竹' , '1989-07-01' , '女'),
        ('08' , '王菊' , '1990-01-20' , '女');

insert into course (c_id, c_name, t_id)
values  ('01' , '语文' , '02'),
        ('02' , '数学' , '01'),
        ('03' , '英语' , '03');

insert into teacher (t_id, t_name)
values  ('01' , '张三'),
        ('02' , '李四'),
        ('03' , '王五');

insert into score (s_id, c_id, score)
values  ('01' , '01' , 80),
        ('01' , '02' , 90),
        ('01' , '03' , 99),
        ('02' , '01' , 70),
        ('02' , '02' , 60),
        ('02' , '03' , 80),
        ('03' , '01' , 80),
        ('03' , '02' , 80),
        ('03' , '03' , 80),
```

```
        ('04' , '01' , 50),
        ('04' , '02' , 30),
        ('04' , '03' , 20),
        ('05' , '01' , 76),
        ('05' , '02' , 87),
        ('06' , '01' , 31),
        ('06' , '03' , 34),
        ('07' , '02' , 89),
        ('07' , '03' , 98);
```

创建一张总总表

```
create table total(
select a.s_id as s_id,a.s_name as s_name,a.s_age as s_age,a.s_sex as s_sex,
b.c_id as c_id,b.score as score,c.t_id as t_id,d.t_name as t_name
from student a
left join
score  b on a.s_id=b.s_id
left join
course c on b.c_id=c.c_id
left join
teacher d on c.t_id=d.t_id
);
select * from total;
```

1、查询"01"课程比"02"课程成绩高的学生的信息及课程分数

```
select a.s_id as s_id,score1,score2 from
(select s_id, score as score1 from score where c_id='01') a
inner join
(select s_id, score as score2 from score where c_id='02') b
on a.s_id=b.s_id
where score1>score2;
```

2、查询"01"课程比"02"课程成绩低的学生的信息及课程分数

```
select a.s_id as s_id,score1,score2 from
(select s_id, score as score1 from score where c_id='01') a
inner join
(select s_id, score as score2 from score where c_id='02') b
on a.s_id=b.s_id
where score1<score2;
```

3、查询平均成绩大于等于60分的同学的学生编号和学生姓名和平均成绩

```
select student.s_id as s_id,student.s_name as s_name,b.avg_score as avg_score
from student
right join
(select s_id,avg(score) as avg_score from score
group by s_id having avg_score>60) b
on student.s_id=b.s_id;
```

4、查询所有同学的学生编号、学生姓名、选课总数、所有课程的总成绩

```
select s_id, s_name, count(c_id) as c_num, sum(score) as total_score
from total
group by s_id ;
```

5、查询"李"姓老师的数量

```
select count(t_name) from teacher
where t_name like '李%';
```

6、查询学过"张三"老师授课的同学的信息

```
select distinct s_id,s_name,s_age,s_sex
from total
where t_name='张三';
```

7、查询学过编号为"01"并且也学过编号为"02"的课程的同学的信息

```
select * from student
where s_id in
(select s_id from score where c_id='01')
and s_id in
(select s_id from score where c_id='02');
```

8、查询没有学全所有课程的同学的信息

```
select s.s_id,s.s_name,s.s_age,s.s_sex from student as s inner join (select s_id
from total group by s_id having count(c_id) <3) as a on s.s_id = a.s_id;
```

9、查询至少有一门课与学号为"01"的同学所学相同的同学的信息

思路：先找出'01'同学学过的c_id，再找出学过任一门的s_id，再根据s_id在student找学生信息。

```
select * from student
where s_id in
(select distinct s_id from score
where c_id in
(select c_id from score where s_id='01'));
```

10、查询没学过"张三"老师讲授的任一门课程的学生姓名

```
select s_id,s_name from student
where s_id not in
(select distinct s_id from total
where t_name='张三');
```

11、查询两门及其以上不及格课程的同学的学号，姓名及其平均成绩

思路：先找不及格超过两门的s_id，为表a，再根据表a连接学生信息表student和平均分表b。

```
select a.s_id,student.s_name,b.avg_score from
(select s_id from score
where score<60
group by s_id having count(*)>=2) a
left join
student on a.s_id=student.s_id
left join
(select s_id,avg(score) as avg_score
from score
group by s_id) b
on a.s_id=b.s_id;
```

12、检索"01"课程分数小于60，按分数降序排列的学生信息

```
select a.s_id,student.s_name,student.s_age,student.s_sex,a.score from
(select s_id,score from score
where c_id='01' and score<60
order by score desc) a
left join student on a.s_id=student.s_id;
```

13、查询不同老师所教不同课程平均分从高到低显示

```
select t_id,t_name,c_id,avg(score) as avg_score
from total
group by t_id,c_id
order by avg_score desc;
```

14、查询每门课程被选修的学生数

```
select c_id,count(s_id) as '选修人数'
from score group by c_id;
```

15、查询出只有两门课程的全部学生的学号和姓名

```
select student.* from
(select s_id from score
group by s_id having count(c_id)=2) a
left join student on a.s_id=student.s_id;
```

16、查询男生、女生人数

```
select s_sex as '性别',count(1) as '人数'
from student group by s_sex;
```

17、查询名字中含有"风"字的学生信息

```
select * from student
where s_name like '%风%';
```

18、查询同名同姓学生名单，并统计同名人数

```
select distinct s_name,num as '同名人数' from student,(select count(s_id) as num
 from student group by s_name) a;
```

19、查询1990年出生的学生名单(注：Student表中Sage列的类型是datetime)

```
select s_name from student where year(s_age)='1990';
```

20、查询每门课程的平均成绩，结果按平均成绩降序排列，平均成绩相同时，按课程编号

```
select c_id,avg(score) as '平均成绩'
from score group by c_id
order by 平均成绩 desc,c_id;
```

21、查询平均成绩大于等于85的所有学生的学号、姓名和平均成绩

```
select a.s_id,s_name,avg_score from
(select s_id,avg(score) as avg_score from score
group by s_id having avg(score)>=85) a
left join student on a.s_id=student.s_id;
```

22、查询课程名称为"数学"，且分数低于60的学生姓名和分数

```
select s_name,c_name,score from total
where c_name='数学' and score<60;
```

23、查询所有学生的课程及分数情况

```
select s_id,
sum(case when c_id='01' then score else 0 end) as '语文',
sum(case when c_id='02' then score else 0 end) as '数学',
sum(case when c_id='03' then score else 0 end) as '英语'
from total
group by s_id;
```

24、查询任何一门课程成绩在70分以上的姓名、课程名称和分数

```
select s_name,c_name,score
from total where score>70;
```

25、查询不及格的课程

```
select score.c_id,course.c_name,score
from score left join course
on score.c_id=course.c_id
where score<60;
```

26、查询课程编号为01且课程成绩在80分以上的学生的学号和姓名

```
select student.s_id,s_name from student
right join score on student.s_id=score.s_id
where c_id='01' and score>80;
```

因为'01'课程最高分为80，所以查询结果为空。

27、求每门课程的学生人数

```
select c_id,count(1) as '选课人数'
from score group by c_id;
```

28、查询选修"张三"老师所授课程的学生中，成绩最高的学生信息及其成绩

```
select student.*,a.score from
(select s_id,score
from total where t_name='张三'
order by score desc limit 1) a
left join student on a.s_id=student.s_id;
```

29、查询不同课程成绩相同的学生的学生编号、课程编号、学生成绩

```
select a.s_id,a.c_id,a.score
from score a,score b
where a.c_id=b.c_id and a.s_id!=b.s_id and a.score=b.score;
```

30、查询每门功成绩最好的前两名

```
(select c_id,s_id from score where c_id='01' order by score limit 2)
union
(select c_id,s_id from score where c_id='02' order by score limit 2)
union
(select c_id,s_id from score where c_id='03' order by score limit 2);
```

31、统计每门课程的学生选修人数（超过5人的课程才统计）。要求输出课程号和选修人数，查询结果按人数降序排列，若人数相同，按课程号升序排列

```
select c_id,count(s_id) as 选修人数 from score
group by c_id having 选修人数>5
order by 选修人数 desc,c_id;
```

32、检索至少选修两门课程的学生学号

```
select s_id from score group by s_id having count(c_id)>=2;
```

33、查询选修了全部课程的学生信息

```
select * from student
where s_id in
(select s_id from score
group by s_id having count(c_id)=(select count(*) from course));
```

复习内容

- 网络编程
- 进程线程
- 网络并发模型
- http协议
- 聊天室,文件服务器