

## 6月1号 课堂笔记

### 导入导出的问题

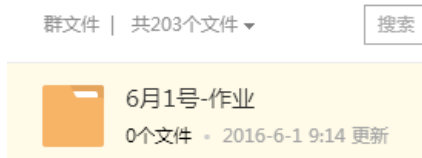
导出的sql文件里，可能字段或数据有冲突，导致不能导入的，  
可以删除表或清空表数据。

注意：这是我们学习和练习，删除表和清空表数据都非常慎重，需要确认和备份数据!

1. drop table 表名; // 删除表, 注意删除表的auto\_increment的值会保留
  2. delete from table ; // 删除所有数据
  3. truncate table 表名; // 清空表数据
- 或是使用phpmyadmin



大家把昨天的作业发给组长，组长统计汇总成 word文档，或云笔记，发到群文件里  
6月1号-作业



### 练习

1. 显示 cv 表中的男性平均年龄，保留一位小数。

```
select truncate ( avg(age), 1 ) from cv where sex = '男' limit 10;
```

2. 显示 cv 表中的女性的平均身高和体重，保留一位小数，四舍五入。

```
select fromat ( avg(age), 1 ) from cv where sex = '男' limit 10;
```

3. 显示cv表中的数据，其中 city和性别 放在一个字段里显示，用 / 连接显示

```
select concat( city, "/", sex ) from cv limit 10;
```

```
select concat_ws("/", city, sex ) from cv limit 10;
```

4. 统计出所有姓王的用户总数

```
select count(*) from cv where name like "王%";
```

- 5.统计所有名字中包含 ‘涵’ 子的用户，按性别分组显示总数

```
select sex, count(*) from cv where name like "%涵%" group by sex;
```

6. 统计所有名字中包含 ‘涵’ 子的用户，按性别分组显示总数, 并且分组总数大于10的

```
select sex, count(*) from cv where name like "%涵%" group by sex having count(*) > 10;
```

```
select sex, count(*) as total from cv where name like "%涵%" group by sex having total > 10;
```

5. 显示用户名，只显示前3个字符

```
select left(name, 3) from cv limit 10;
```

```
select substr(name, 1, 3) from cv limit 10;
```

6. 显示用户名，只显示名子，不显示姓

```
select substr(name, 2) from cv limit 10;
```

7. 显示用户生日的timestamp为多少

```
select unix_timestamp( birthday ) from cv limit 10;
```

## 8. 显示用户的生日离当前日期， 还没到生日的， 还差多少秒， 分， 小时， 还差多少天， 还差多少月

思路 1：

比如： 用户的生日是 1990-11-06， 那么他的生日离现在还有多少时间呢？

当前日期比如是： 2016-06-01

先忽略年， 直接按月日进行计算， 5个月在加5天

**忽略年**， 可以把生日的年当成是 2016年， 或是把2016年当成是 1990年， 然后计算2个日期的差。

`select to_days(now());` 当前的天数

`select concat("2016-", right(birthday, 5 )) from cv limit 10;` // 将用户的生日的年换成当前的年份

`select to_days( concat("2016-", right(birthday, 5 )) ) from cv limit 10;` // 计算当前年份下的生日的天数

计算2个的天数差

`select id, name, birthday, ( to_days( concat("2016-", right(birthday, 5 )) ) - to_days(now()) ) as diff_days  
from cv`

`where concat("2016-", right(birthday, 5 )) > now() limit 10;`

天数计算出来了， 那么 秒， 分， 小时， 天， 月(按30天)都可以算出来

秒 天 \* 86400

分 天\*24\*60

小时 天\*24

月 ceil(天/30)



id	name	birthday	diff_days
2	彭容伊2	1991-07-28	57
3	陈文嘉书宁3	1988-06-16	15
5	廉一毅5	1982-08-16	76
6	苏奕杉6	1976-11-05	157
7	奚衍煊7	1981-12-16	198
9	倪馨蔓9	1971-12-30	212
10	苏鑫媛10	1971-08-24	84
11	周孟宇11	1975-06-13	12
12	任天晴12	1996-08-29	89
13	毕羽姗13	1993-06-07	6

10 rows in set (0.00 sec)

思路2：

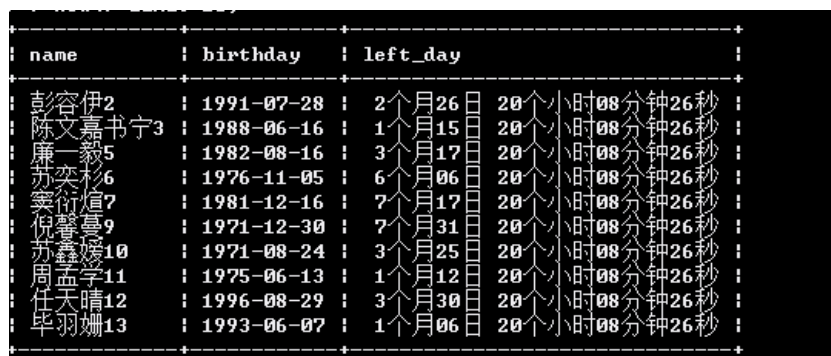
算出日期的时间戳， 然后2个时间戳相减， 将得到的时间戳， 在根据date\_format转成

m-d H:i:s

练习

写 当前日期和生日（ 转成同一年的 ）的时间戳 相减

`select name, birthday, date_format ( from_unixtime ( unix_timestamp ( concat("2016-", right(birthday, 5 )) ) - unix_timestamp ( now() ) ), " %c个月%d日 %H个小时%i分钟%S秒 " ) as left_day from cv where concat("2016-", right(birthday, 5 )) > now() limit 10;`



name	birthday	left_day
彭容伊2	1991-07-28	2个月26日 20个小时08分钟26秒
陈文嘉书宁3	1988-06-16	1个月15日 20个小时08分钟26秒
廉一毅5	1982-08-16	3个月17日 20个小时08分钟26秒
苏奕杉6	1976-11-05	6个月06日 20个小时08分钟26秒
奚衍煊7	1981-12-16	7个月17日 20个小时08分钟26秒
倪馨蔓9	1971-12-30	7个月31日 20个小时08分钟26秒
苏鑫媛10	1971-08-24	3个月25日 20个小时08分钟26秒
周孟宇11	1975-06-13	1个月12日 20个小时08分钟26秒
任天晴12	1996-08-29	3个月30日 20个小时08分钟26秒
毕羽姗13	1993-06-07	1个月06日 20个小时08分钟26秒

注意： 多了1个月， 其实在php显示的时候去处理

思路3

使用 timestampdiff 函数

知识点

to\_days // 根据日期取天数，从公元0年开始

```
mysql> select to_days('2016-01-01');
+-----+
| to_days('2016-01-01') |
+-----+
|          736329 |
+-----+
1 row in set (0.00 sec)

mysql> select to_days('0000-01-01');
+-----+
| to_days('0000-01-01') |
+-----+
|          1 |
+-----+
1 row in set (0.00 sec)
```

concat / left / right / substr

unix\_timestamp // 将时间转成时间戳 就是从1970年开始的秒数

from\_unixtime // 将时间戳该成日期，y-m-d his

date\_format // 格式化一个日期，按我们的格式显示

now() // 当前的时间

1.ceil () /ceiling () 向上取整

```
mysql> select ceil(1.2);
+-----+
| ceil(1.2) |
+-----+
|          2 |
+-----+
1 row in set (0.01 sec)
```

2.floor () 向下取整

ex: floor (1.2) = 1

```
mysql> select floor(1.2);
+-----+
| floor(1.2) |
+-----+
|          1 |
+-----+
1 row in set (0.00 sec)
```

3.round () 四舍五入

```
mysql> select round(1.6);
+-----+
| round(1.6) |
+-----+
|          2 |
+-----+
1 row in set (0.00 sec)

mysql> select round(1.4);
+-----+
| round(1.4) |
+-----+
|          1 |
+-----+
1 row in set (0.00 sec)
```

9. 显示用户的信息，按年龄从大到小，身高从高到低，男性，显示前10个用户。

select \* from cv where sex = '男' order by age desc, height desc limit 10;

10. 显示用户的信息，按体重从高到低，身高从高到低，女性，显示从第20个，到第30个户

select \* from cv where sex = '男' order by age desc, height desc limit 19, 10;

我们要学习相关的思路和知识点，然后不太清楚的，去查手册。

## 数据表之间的关系

我们创建了cv表用来保存简历的基本信息，如果现在需要添加新的简历信息，譬如，教育经历，请问如何设计？

一份简历的教育经历可能会有很多条，从小学到大学，甚至研究生；教育经历还包括了多个信息：哪年开始，哪年结束，所在学校名称.....

上面的信息如果仍然保存在cv表里，明显不合适，因为一条记录仅仅是一份简历，又不能让记录进行嵌套

那么我们可以创建一个新的数据表：edu来保存这些信息

请大家来设计这个表

服务器: mysql wampserver » 数据库: xinan » 表: edu									
浏览 结构 SQL 搜索 插入 导出 导入 权限									
#	名字	类型	排序规则	属性	空	默认	额外		
1	id	int(11)			否	无	AUTO_INCREMENT		
2	degree	varchar(10)	utf8_general_ci		否				
3	from_date	date			否	0000-00-00			
4	end_date	date			否	0000-00-00			
5	name	varchar(50)	utf8_general_ci		否				
6	comment	text	utf8_general_ci		否	无			
7	created_at	datetime			否	0000-00-00 00:00:00			

查看建表语句：

```
mysql> use xinan;
mysql> show create table edu;
+-----+-----+
| Table | Create Table Statement |
+-----+-----+
| edu   | CREATE TABLE `edu` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `degree` varchar(10) NOT NULL DEFAULT '',
  `from_date` date NOT NULL DEFAULT '0000-00-00',
  `end_date` date NOT NULL DEFAULT '0000-00-00',
  `name` varchar(50) NOT NULL DEFAULT '',
  `comment` text NOT NULL,
  `created_at` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 |
+-----+-----+
```

但是，从结构上来看，cv和edu两个表并没有关联，我们需要对edu表做个小小的修改，让他们建立关联（edu里的记录分别数据哪条cv的）

**ALTER TABLE edu ADD cv\_id INT NOT NULL DEFAULT 0 AFTER id;**

关系数据库设计中有三种关系：

- 一对一 （ 一张表 ）
- 一对多（最常用） （ 2张表 ）
- 多对多 （3张表 ）

我们前面的cv和edu表之间是什么关系？

一对多

假设，当前有一份简历，它的id是1，那么如何才能获取到用户简历的信息？


```
select * from cv where id=1;
```

那么获取id等于1的那份简历中录入的**所有**教育经历呢？

```
select * from edu where cv_id=1;
```

现在，再提高点难度，edu表里的school\_name不再使用文本保存，我们另外有一张表school专门用来保存学校的信息，school\_name现在变成了school\_id

edu表， school表

 edu.sql

根据sql文件导入，生成表和相关数据

1. 如果数据名称不一样，可以修改sql文件，将数据库名改成你本地的名称

```
>
--
-- Database: `xinan`
--
--
```

2. 或是将数据库名称属性删除

3. 或是直接复制相关sql，在phpmyadmin 里，sql 或导入

左连接

```
SELECT * FROM edu LEFT JOIN school ON edu.school_id = school.id;
```

此时，edu为左表，school为右表

LEFTJOIN（左连接）以edu表的记录为基础，左表的记录会全部显示出来，右表如果没有能够关联上的数组，则记录中显示NULL

也就是说，如果edu表中的school\_id与school表中的id有对应不上情况，学校名称会为空（譬如学校信息被意外删除）

```
select * from edu left join school on edu.school_id = school.id limit 2\G;
```

```
mysql> select * from edu left join school on edu.school_id = school.id limit 2\G;
***** 1. row *****
      id: 1
     cv_id: 1
    degree: 大学
   from_date: 2010-09-01
     end_date: 2014-09-01
    school_id: 1
     comment: 很牛。。。
created_at: 2016-06-01 02:22:06
      id: 1
     name: 北京大学
    address:
      type: 1
     city: 北京
***** 2. row *****
      id: 2
     cv_id: 1
    degree: 研究生
   from_date: 2014-09-01
     end_date: 2016-07-01
    school_id: 1
     comment: 研究生 .....
created_at: 0000-00-00 00:00:00
      id: 1
     name: 北京大学
    address:
      type: 1
     city: 北京
2 rows in set (0.00 sec)

ERROR:
No query specified
```

\G: 将命令行的字段， 按行显示

指定需要显示的字段

注意:

因为2张表会join成一张表，所以，如果字段名称有重名的话， 需要指定表名. 字段名

显示用户的教育经历信息

简历id, 学校类型, 学校名称, 起始时间, 结束时间

```
select edu.cv_id, edu.id as edu_id, edu.degree, school.name, school.id as school_id, edu.from_date,
edu.end_date from edu left join school on edu.school_id = school.id limit 2;
```

```
mysql> select edu.cv_id, edu.id as edu_id, edu.degree, school.name, school.id as s_id, edu.from_date, edu
edu left join school on edu.school_id = school.id limit 2;
+-----+-----+-----+-----+-----+-----+-----+
| cv_id | edu_id | degree | name      | s_id | from_date | end_date |
+-----+-----+-----+-----+-----+-----+-----+
| 1     | 1     | 大学   | 北京大学 | 1    | 2010-09-01 | 2014-09-01 |
| 1     | 2     | 研究生 | 北京大学 | 1    | 2014-09-01 | 2016-07-01 |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

注意下:

如果两张表里有字段相同的，可以使用as，将字段重命名， where 语句里，还是使用原表中的字段名。

左连接



## 右连接

**SELECT \* FROM edu RIGHT JOIN school ON edu.school\_id = school.id;**

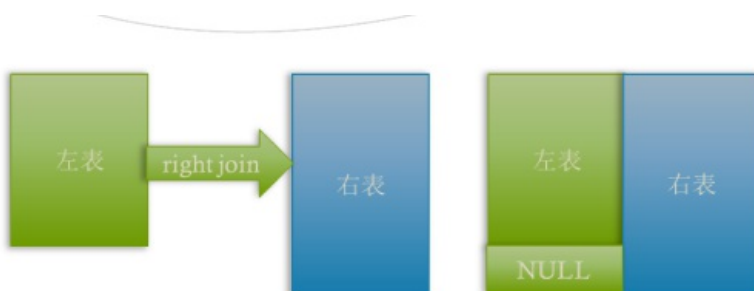
RIGHTJOIN（右连接）与左连接相反，查询出的结果以右表为主，如果edu表中的school\_id与school表中的id有对应不上的情况，school\_id错误的简历信息将无法显示

```
mysql> select edu.cv_id, edu.degree, edu.school_id, school.name, edu.from_date, edu.to_date
on edu.school_id = school.id ;
```

cv_id	degree	school_id	name	from_date	end_date
1	研究生	1	北京大学	2014-09-01	2016-07-01
2	大学	2	清华大学	1997-09-01	2001-07-01
4	大学	11	合肥工业大学	2011-10-11	2016-05-08
NULL	NULL	NULL	复旦大学	NULL	NULL
NULL	NULL	NULL	武汉大学	NULL	NULL
NULL	NULL	NULL	中国人民大学	NULL	NULL
NULL	NULL	NULL	浙江大学	NULL	NULL
NULL	NULL	NULL	上海交通大学	NULL	NULL
NULL	NULL	NULL	南京大学	NULL	NULL
NULL	NULL	NULL	中国科学技术大学	NULL	NULL
NULL	NULL	NULL	东南大学	NULL	NULL
NULL	NULL	NULL	安徽大学	NULL	NULL
NULL	NULL	NULL	安徽师范大学	NULL	NULL
NULL	NULL	NULL	安徽医科大学	NULL	NULL
NULL	NULL	NULL	合肥学院	NULL	NULL

15 rows in set (0.01 sec)

## 右连接



## 内连接

INNER JOIN（内连接）既不以左表为准，也不以右表为准，只显示两边完全符合条件的记录



**SELECT \* FROM edu INNER JOIN school ON edu.school\_id = school.id;**

```
mysql> select edu.cv_id, edu.degree, edu.school_id, school.name, edu.from_date,
on edu.school_id = school.id ;
+-----+-----+-----+-----+-----+-----+
| cv_id | degree | school_id | name       | from_date | end_date |
+-----+-----+-----+-----+-----+-----+
| 1 | 研究生 | 1 | 北京大学 | 2014-09-01 | 2016-07-01 |
| 2 | 大学 | 2 | 清华大学 | 1997-09-01 | 2001-07-01 |
| 4 | 大学 | 11 | 合肥工业大学 | 2011-10-11 | 2016-05-08 |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

可以直接连在一起写

```
select * from table_name1, table_name2,..... where table_name1.id = table_name2.id and .....
limit 10;
```

```
select * from edu , school where edu.school_id = school.id limit 2;
```

注意:

1. 多张表关联显示， where 语句里一定要有关联的语句，否则，会将表中的记录全部取出来！
2. 如果关联的表很多， 可以使用as 给表定义别名， 给表定义别名以后， 字段的引用需要使用别名。

```
select t1.id, t1.from_date,t1.end_date, t2.name from edu as t1, school as t2 where t1.school_id =
t2.id;
```

(注: as 省略也可以, mysql5.5)

### 3张表关联

显示简历信息，并显示简历的教育经历信息

查询显示id为1的用户的，简历信息，包括教育经历信息

```
select cv.id, cv.name, cv.sex, cv.city,cv.birthday, edu.degree, edu.from_date,edu.end_date, school.name,
school.city as school_city from cv, edu, school where cv.id = edu.cv_id and edu.school_id =
school.id
```

id	name	sex	city	birthday	degree	from_date	end_date	name	school_city
1	花殇霏1	其他	阜阳	1973-01-13	大学	2010-09-01	2014-09-01	北京大学	北京
1	花殇霏1	其他	阜阳	1973-01-13	研究生	2014-09-01	2016-07-01	北京大学	北京
2	彭容伊2	男	安庆	1991-07-28	大学	1997-09-01	2001-07-01	清华大学	北京
4	孙孟学4	女	芜湖	1989-04-06	大学	2011-10-11	2016-05-08	合肥工业大学	安徽



注意：

3张表，关联显示，那么where条件里，至少有2个关联语句。（即多张表(n)关联显示，where条件里的关联语句为  $n - 1$ ）。

## 外键

实际上，理想状况下，无论是左连接，还是右连接，他们的结果都应该与内连接一样，只是因为某些原因，导致两边数据不对称时，才会出现差异

如何能在技术上保证数据不出现差异呢？我们可以使用外键。

### MySQL外键

如果一张数据表中的某个字段的值是另外一张表的主键，那么我们可以称该字段为表的外键

因此，我们可以称：cv\_id是edu表的外键；school\_id也是edu表的外键

外键主要是用来保证数据的完整性，表里是否设置外键不影响其使用。

### 添加外键

如果我们给数据表添加外键，需要注意，表中的外键所保存的值，与关联表里的id是否真正的能够对应

更直白的说就是，shool\_id里保存的id在school表里是否存在，如果不存在，添加外键时，系统会报错，因此，在测试时，我们可以直接执行truncate来清空错误数据，但是如果是在产品线上，需要通过批处理脚本，来统一修正错误数据，然后才可以添加外键

```
ALTER TABLE edu ADD FOREIGN KEY(cv_id) REFERENCES cv(id) ON DELETE CASCADE;
ALTER TABLE edu ADD FOREIGN KEY(school_id) REFERENCES school(id) ON DELETE CASCADE;
```

约束条件：ON DELETE

当主表的数据被删除时，关联表的从数据如何处理。譬如school表里的记录被删除时，edu表里保存同样school\_id的数据如何被系统自动处理

- 1、RESTRICT：约束  
如果存在从数据，不允许删除主数据。
- 2、NO ACTION （mysql中 no action 等于 RESTRICT ）  
如果存在从数据，不允许删除主数据。

<http://dev.mysql.com/doc/refman/5.7/en/create-table-foreign-keys.html>

NO ACTION: A keyword from standard SQL. In MySQL, equivalent to RESTRICT. The MySQL Server rejects the delete or update operation for the parent table if there is a related foreign key value in the referenced table. Some database systems have deferred checks, and NO ACTION is a deferred check. In MySQL, foreign key constraints are checked immediately, so NO ACTION is the same as RESTRICT.

- 3、CASCADE：级联  
删除主数据，顺便也删掉从数据。
- 4、SET NULL  
删除主数据，从数据外键的值设为NULL。

建议大家使用约束，保证数据一致性，使用级联时，有一定风险。

### 如何快速的检索数据

我们都使用过英文字典用来查询单词，如何快速的检索一个单词？

首先搜索首字母  
在基本确定位置后，再按第二个字母检索  
甚至第三、第四……直到查找到需要的单词

数据库中检索数据没有这么聪明，默认情况下，如果我想搜索name= 'andy' 的记录，数据库会把表里所有的数据挨个比对一遍，发现一个匹配则返回一个，直到数据全部匹配完

假设表里有上百万的记录，那么这种匹配是非常消耗时间的。

一条sql, 如果没有建立索引的，可能是要几十秒， 建立合理的索引，可能0点0几秒就能出结果了。

## HASH

我们可以把数据库中的索引理解为一本字典

数据库维护字典，每条记录都按照对应的顺序在字典中进行登记  
当有新的记录插入时，分析数据内容，然后将数据进行归类  
当有查询需求时，不会匹配全部数据，而是按字典进行检索

以英文单词举例，26个字母构成英文单词，当系统存入某个字段值为andy时，先将andy存入a字母保存的字典，在a字母的字典中，又划分出26个字母的子字典，数据进一步保存到n中……

当我们检索andy时，按照a->n->d->y的顺序可以非常快速的检索到记录，这就是我们常说的HASH（哈希）

## 索引

数据库的索引就是基于hash来实现的

有同学会有疑问，既然索引这么强大，为什么MySQL不默认给每个字段都设置索引就好了

大家要注意，一个新的记录保存到数据库中，分析其数据并归档至字典也是要消耗时间的，一条记录里有N个字段，每个字段都执行一次字典归类会大大影响数据插入的速度，因此索引默认是不添加的，需要人工来设置

**通常，常用来作为where查询条件的字段才需要设置索引，其他字段没有必要。**

## 创建索引

```
ALTER TABLE tbl_name ADD INDEX index_name (col_name);
```

检查select语句是否使用了正确的索引，我们一般在select之前添加explain

```
ALTER TABLE cv ADD INDEX idx_sex(sex);  
EXPLAIN SELECT * FROM cv WHERE sex = '男';
```

使用explain可以分析sql语句的性能和索引使用情况

+ 选项

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	cv	ref	name	name	62	const	1	Using index condition

## 组合索引及注意事项

对多个字段组合创建的索引

```
ALTER TABLE tbl_name ADD INDEX index_name (column1, column2);
```

当where条件里不仅按一个字段去进行检索，还使用了其他字段，如果这种检索是经常性的操作，那么我们需要创建组合索引

假设有组合索引（A,B,C）三个字段，相当于创建了三个索引可以使用（以左为准）

（A,B,C）

（A,B）

（A）

所以，查询的顺序很重要

## 创建索引

可以使用sql语句，或是使用phpmyadmin

```
ALTER TABLE cv ADD INDEX idx_sex(sex);
```



查看索引语句

语法:

```
show index from table_name;
```

```
show index from cv;
```

练习

需求：网络教育很火，可以在网上选课学习，我们来练习创建学生选课相关的表。

学生&课程

学生表 (数据库或是模块名前缀\_表名 )

id, 用户名, 密码, 用户姓名, 性别, 生日, 注册时间, 注册ip, 状态 (正常, 禁用)

老师表

id, 老师姓名, 生日, 性别, 简介

课程表

ID, 课程名称, 课程类型, 课程简介

课程类型表

ID, 名称, 简介

老师课程关系表

id, 老师id, 课程id, 课程开始时间, 课程结束时间, 课程状态 (正常|失效), 选课人数

学生选课表

ID, 学生ID, 课程id, 老师课程关系表ID

表名, 主键, 自增, 字段名, 字段类型, 表与表的关系, 索引

表创建好后, 需要初始化数据

学生, 老师, 课程, 课程类型 ...

表创建好, 初始化一些数据, 然后在继续学习和练习sql的写法。

课程类型:

如:

php开发课程

java开发课程

Android开发课程

IOS开发课程

## 插入数据

1. 手写 insert into
2. 使用phpmyadmin
3. 写脚本

'''

模拟真实的数据，插入数据库  
最少几十条数据

1. 创建表
2. 写出相关查询相关SQL语句
  - 1) 查看男性学生的用户信息
  - 2) 查看男性学生，小于18岁或这大于35岁的用户
  - 3) 查看选课人数最多的10门课程
  - 4) 查看选课人数最多的10个老师
  - 5) 查看学生id为1的，选择的课程信息以及课程老师信息
  - 6) 查看php开发课程类型下，
    - a. 所有的课程信息
    - b. 所有选择了php开发课程的学生信息
    - c. 所有的老师信息
  7. 按课程类型，进行分组显示，学生人数
  8. 按课程类型，进行分组显示，老师人数

做完的，可以在做下，下面的练习

## 话题&回复

用户表

id, 用户名, 密码, 用户昵称, 性别, 生日, 注册时间, 注册ip, 状态（正常，禁用）

话题讨论表

id, 用户id, 话题标题, 话题内容, 发布时间, 编辑时间, ip地址, 状态（正常，禁止）

话题讨论回复表

id, 用户ID, 话题id, 用户ID, 回复内容, 回复时间, ip地址, 状态

1. 查看所有话题信息，包含话题的发布者信息
2. 查看话题id为1下，所有的回复信息以及回复者昵称。
3. 查看用户id为1，发布的所有话题信息
4. 查看用户id为1的，所有回复的信息。

## 圈子表-关系表-用户表

用户表

id, 用户名, 密码, 用户昵称, 性别, 生日, 注册时间, 注册ip, 状态（正常，禁用）

群组表

ID, 群名称, 群类型id, 群简介, 状态, 群等级, 创建人, 创建日期, 群人数

群类型表

ID, 类型名称, 说明

群内用户表

id, 群id, 用户ID, 加入时间

1. 显示群，以及群类型的具体名称和群说明
2. 显示指定群id下，所有的用户信息
3. 显示指定的一个用户，加入的所有群的信息
4. 显示群人数最多10个群
5. 显示群人数最多的3个群类型
6. 显示群信息，按群类型分组