

MySQL 系列课程

时间: 2017/11/17

作者:袁毅雄

邮箱: 896778954@qq.com

目录

MySQL 系列课程	1
MySQL 的获取	3
MySQL 安装	3
MySQL 卸载	4
MySQL 基础命令	4
连接数据库	4
退出	4
DCL	5
三建(建库,建表,建约束)	5
库	5
表	5
创建表	5
删除表	6
修改表	6
查看表的结构	7
显示所有表	7
属性	7
约束(属性)	8
约束作用	8
约束使用示例	8
索引	9
索引的作用	9
索引的分类	10
索引使用示例	10
DML	11
四句(增,删,改,查)	11
增—添加数据	11
单条添加	11
多条添加	11
缓存数据添加	12
删—删除数据	13
语法:	13
示例:	13

改—修改数据	13
语法:.....	13
示例:.....	13
查—查询数据	13
语法:	13
示例.....	14
条件表达式(删,改,查)	15
条件运算符.....	15
逻辑运算符.....	15
DQL.....	16
四句(查).....	16
连表查询.....	16
内链接.....	16
外连接.....	17
子查询.....	17
where	17
from.....	18
select	18
聚合函数.....	18
sum,avg,count,max,min	18
分组.....	19
排序.....	19
升序.....	19
降序.....	19
多列排序.....	19
分页.....	19
where/having	20
where	20
Having.....	20
TCL.....	20
事物.....	20
ACID	20
MySQL 自动事物处理.....	20
事物的开启.....	20
事物的回滚.....	20
事物的提交.....	21
示例.....	21
备份/还原	21
综合练习题.....	21
school.sql.....	21
1.统计出各个年级男生的人数.....	27
参考效果.....	27
参考答案.....	27
2.查询出参加“高等数学-4”及格和不及格的男生女生人数	28

参考效果.....	28
参考答案.....	28
3.查询出“郭靖”所学的所有科目的考试情况,并且根据考试成绩降序.....	28
参考效果.....	28
参考答案.....	29
4.查询出各班级各科目考试的男女及格比例.....	30
参考效果.....	30
参考答案.....	31

MySQL 的获取

MySQL 官网下载地址:

<http://www.mysql.com/downloads/> (很多都链接到了 Oracle 官网)

注: 1、在官方下载,是要先注册,然后才能下载。2、这里提前申请好了账号和密码,方便你下载:

ID/密码: mysqlpub@gmail.com/MySQLpub123 (注意大小写!)

Oracle 官网下载地址:

<https://edelivery.oracle.com>

ID/密码: mysqlpub@gmail.com/MySQLpub123 (注意大小写!)

MySQL 安装

第一步:

变量名: MYSQL_HOME

变量值: D:\Program Files\mysql-5.6.11-win64

再在 Path 中添加 %MYSQL_HOME%\bin

第二步:

1) 从控制台进入到 MySQL 解压目录下的 bin 目录下:

2) 输入服务安装命令:

```
mysqld install MySQL --defaults-file="D:\Program Files\mysql-5.6.11-win64\my-default.ini"
```

安装成功后会提示服务安装成功。

第三步:

刚安装完成时 root 账号默认密码为空,此时可以将密码修改为指定的密码。如: 123456

方法一:

```
c:>mysql -uroot
```

```
mysql>show databases;
```

```
mysql>use mysql;
```

```
mysql>UPDATE user SET password=PASSWORD("123456") WHERE user='root';
mysql>FLUSH PRIVILEGES;
mysql>QUIT
```

MySQL 卸载

第一步：

首先，先在服务（开始——>控制面板——>管理工具——>服务）里停掉 MySQL 的服务。
打开控制面板-添加删除程序，找到 MySQL，卸载。

第二步：

把安装好的 MYSQL 卸载了，但这对于卸载 MySQL 来说这只是一半，还有重要的另一半是要清理注册表。

开始-运行里面输入 regedit，打开注册表，

找到关于 MYSQL 的项把他们都删除，要一个项一个项的查找把他们都删除，这样在安装的时候就可以了。其实注册表里 MySQL 的项就是这三项：

HKEY_LOCAL_MACHINE/SYSTEM/ControlSet001/Services/Eventlog/Application/MySQL

HKEY_LOCAL_MACHINE/SYSTEM/ControlSet002/Services/Eventlog/Application/MySQL

HKEY_LOCAL_MACHINE/SYSTEM/CurrentControlSet/Services/Eventlog/Application/MySQL

这样，把上面的三项删除了之后，MySQL 就基本卸载完全了。如果你还不放心的话，可以在 C 盘查找 mysql，把相关的项都删除。

有可能你找不到注册表的第三项，有可能是把 mysql 的信息存在了一个叫 C:\ProgramData\的文件夹里了，找到 C:\ProgramData\Mysql 的文件夹删除

MySQL 基础命令

连接数据库

```
mysql -h 主机名/IP -u 账号 -p 密码
```

示例：

```
mysql -hlocalhost -uroot -proot
```

退出

```
exit
```

DCL

三建(建库,建表,建约束)

库

create	database	[if not exists]	《数据库名称》;	---创建数据库
drop	database	[if exists]	《数据库名称》;	---删除数据库
show	databases;			---查看当前 MySQL 下的所有数据库
use	《数据库名称》;			---使用数据库

表

创建表

语法

create	table	[if not exists]	《表名》(
列名	列类型	[属性]	[索引] [注释],
列名	列类型	[属性]	[索引] [注释],
.....			
列名	列类型	[属性]	[索引] [注释]
)[表类型][字符集][注释]			

示例

--基础结构	
create table student(
stu_no	int(4),
name	varchar(20),
age	tinyint(3),
sex	char(4) ,
birthday	date,
information	tinyText
) engine=mylsam,charset=utf8	

删除表

```
drop table 表名
```

修改表

修改表的名称

```
alter table 表名 rename 新表名
```

修改表的列

添加列

```
alter table 表名 add 列 类型
```

修改列名/类型

```
alter table 表名 change 原列名 新列名 数据类型
```

修改列的类型

```
alter table 表名 modify 列名 新数据类型
```

删除列

```
alter table 表名 drop 列名
```

列的类型

java	mysql	介绍
String	varchar(长度), char(长度), TinyText,Text,....	字符串
char	char(长度),	字符
short/int/long/float/double	tinyint,smallint,int,bigint,....	数字
date	date,time,dataTime,TimeStamp,Year,....	日期
Object	null	空的

查看表的结构

```
desc 表名
```

显示所有表

```
show tables;
```

属性

表类型：**engine=mylsam**

MyIsam	InnoDB	功能
不支持	支持	事物
不支持	支持	数据行锁定
不支持	支持	外键约束
支持	不支持	(列)全文索引
大,约 2 倍	较小	表空间大小

适用场合：

MyIsam： 结业空间及相应速度(查询业务较多的表)

InnoDB： 安全性，事物处理及多用户操作数据表(添加，删除，修改功能较多的表)

数据表的存储位置：**MySQL** 安装目录\data 下存放数据表。目录名对应数据库名，该目录下文件名对应数据表

MyIsam： *.frm -- 表结构定义文件/*.MYD -- 数据文件/*.MYI -- 索引文件

InnoDB： 类型数据表只有一个*.frm 文件，以及上一级目录的 ibdata1 文件

字符集: `charset=utf8`

注释: `comment '注释'`

自增: `auto_increment=100`

约束(属性)

约束作用

Unsigned

无符号，声明该数据列不允许为负数

ZeroFill

0 填充，不足位数用 0 来填充，
如: `int(3)`, 输入 5 结果为 005

Auto_Increment

自动增长，每条数据在上一条数据上加 1，
通常设置为主键(primary key)，且为整数类型，
可定义初始值和步长

Null 和 Not Null

默认为 null, 表示该列允许不插入值
如果设置为 not null 表示该列必须有值

Default

用于设置默认值
如: 性别列，默认为“男”

约束使用示例

```
--属性
create table student(
  stu_no int(4) zerofill,
  name varchar(20) not null,
  age tinyint(3) unsigned,
  sex char(4) default 'man',
  birthday date,
```



```
information tinyText  
  
) engine=mysam,charset=utf8
```

```
--属性  
create table student(  
    stu_no int(4) auto_increment primary key,  
    name varchar(20)  
) engine=mysam,charset=utf8, auto_increment=15
```

索引

索引的作用

- 1.提高查询速度
- 2.确保数据的唯一性
- 3.可以加快表和表之间的连接，实现表与表之间的参照完整性
- 4.使用分组和排序子句进行数据检索时，可以显著减少分组和排序的时间
- 5.全文检索字段进行搜索优化

索引的分类

primary key 主键索引

unique 唯一索引

index 常规索引

fulltext 全文索引

外键

索引使用示例

```
--索引
create table student(
  stu_no int(4) primary key,
  name varchar(20) unique,
  sex varchar(2),
  info text(2000),

  index(sex),
  fulltext key(info),
  unique key(name,sex) --组合索引

) engine=mysam,charset=utf8
```

DML

四句(增,删,改,查)

增—添加数据

单条添加

语法:

```
insert into 表名(列名 1, 列名 2, 列名 3,...)
values      (值 1 , 值 2 , 值 3,...)
```

示例:

```
insert into student(id,name,sex,birthday)
values
(1,'arvin','man','2012-12-12')

insert into student(sex,name,birthday)
values
('man','arvin','2012-12-12')
```

多条添加

语法:

```
insert into 表名(列名 1, 列名 2, 列名 3,...)
values
(值 1 , 值 2 , 值 3,...),
(值 1 , 值 2 , 值 3,...),
(值 1 , 值 2 , 值 3,...),
.....
```

示例:

```
insert into student(id,name,sex,birthday)
values
(2,'arvin-02','man','2012-12-12'),
```

```
(3,'arvin-03','man','2012-12-12'),  
(4,'arvin-04','man','2012-12-12')
```

缓存数据添加

查询出来的数据

语法:

```
insert into 表名(列名 1, 列名 2, 列名 3,...)  
select 列名 1, 列名 2, 列名 3,... from 表 [where 条件表达式]
```

示例:

```
insert into grade(gradename)  
select gradename from grade
```

内存拼接的数据

语法:

```
--拼接所有  
[sql 结果]  
union all  
[sql 结果]  
  
--去除重复  
[sql 结果]  
union  
[sql 结果]
```

示例:

```
insert into grade(gradename)  
select 'java4' as gradename  
union all  
select 'java3' as gradename  
union all  
select 'java3' as gradename
```

删—删除数据

语法:

```
delete from 表名 [where 条件表达式]
```

示例:

```
delete from student where id not in(1,2) and id not in(3,4)
```

改—修改数据

语法:

```
update 表名 set 列 1=值 1, 列 2=值 2,... [where 条件表达式]
```

示例:

```
update student set name='arvin-04' where id=4
```

```
update student set name='arvin-05',sex='woman' where id=5
```

```
update student set sex='woman' where id in(1,3,5)
```

查—查询数据

语法:

```
select 列 from 表 [where 条件表达式]
```

```
select * from 表 [where 条件表达式]
```

```
select 列 1,列 2,... from 表 [where 条件表达式]
```

示例

```
select * from student
```

```
select id,name from student
```

```
select id from student
```

```
select * from student where id>3
```

```
select * from student where id>3 or id <2
```

```
select * from student where id!=3
```

```
select * from student where name <> 'arvin-02'
```

```
select * from student where id in(1,3)
```

```
select * from student where id not in(1,3)
```

```
select * from student where id between 1 and 4
```

```
select * from student where id >1 and id <=4
```

```
select * from student where id between 4 and 1 ---错误用法
```

```
select * from student where id >=4 and id <=1 ---错误用法
```

```
select * from student where name like '____n'
```

```
select * from student where name like 'a____'
```

```
select * from student where name like '___v___'
```

```
select * from student where name like '%2'
```

```
select * from student where name like 'a%'
```

```
select * from student where name like '%-%'
```

条件表达式(删,改,查)

条件运算符

关键字	示例	
基础用法		
>	age >18	
>=	age >=18	
<	age <18	
<=	age <=18	
==	age ==18	
!=,<>	age !=18 , age <>18	
高级用法		
in	age in(18,20,50,...)	
not in	age not in(18,20,50,...)	
between and	age between 18 and 50	
like	name like 通配符	
	通配符	
	_	_a , __a , a_ , _a_
	%	%a , a% , %a%

逻辑运算符

关键字 mysql	关键字 java	示例
and	&&	表达式 and 表达式
or		表达式 or 表达式

DQL

四句(查)

连表查询

内链接

where

特点:

等同于 inner join 连表

示例:

```
select
    gra.gradeName    as '年级',
    stu.studentName as '姓名'
from
    grade    as gra,
    student as stu
where
    gra.gradeId=stu.gradeId
```

inner join

特点:

inner 连接的表无所谓主从表，两边匹配上就出来，匹配不上换一个都不出来

示例:

```
select
    gra.gradeName    as '年级',
    stu.studentName as '姓名'
from
    student stu
inner join grade gra on gra.gradeId=stu.gradeId
```


外连接

left join

特点:

以 left 关键字为参考,

左边的表为主表,主表的数据全出来

右边的表为从表,从表的数据匹配上就出来匹配不上就用 null 补全

示例:

```
select
    gra.gradeName    as '年级',
    stu.studentName as '姓名'
from
    student stu
left join grade gra on gra.gradeId=stu.gradeId
```

right join

特点:

以 right 关键字为参考,

右边的表为主表,主表的数据全出来

左边的表为从表,从表的数据匹配上就出来匹配不上就用 null 补全

示例:

```
select
    gra.gradeName    as '年级',
    stu.studentName as '姓名'
from
    student stu
right join grade gra on gra.gradeId=stu.gradeId
```

子查询

where

```
select * from student where gradeid =(
    select gradeid from grade where gradename='大一'
)
```

```
select * from student where gradeid in (  
    select gradeid from grade where gradename='大一' or gradename='大二'  
)
```

from

```
select temp.* from  
(  
    select  
        stu.studentname as stu_name,  
        gra.gradename as gra_name  
    from student stu  
    inner join grade gra on gra.gradeid=stu.gradeid  
) temp  
where temp.stu_name like '%文%'
```

select

```
select  
    stu.studentName as studentName,  
    (select gra.gradeName from grade gra where gra.gradeid= stu.gradeid) as gradeName  
from  
    student stu
```

聚合函数

sum,avg,count,max,min

```
Select sum(studentResult) from result ----和  
Select avg(studentResult) from result ----平均值  
Select max(studentResult) from result ----最大值  
Select min(studentResult) from result ----最小值  
  
Select count(studentResult) from result ---数据个数  
Select count(*) from result ---数据个数  
Select count(1) from result ---数据个数
```

分组

```
select sex from student group by sex
select sex,count(1) from student group by sex

select gradeid,sex,count(1) from student group by gradeid ,sex

//分组后只能显示出分组的列和聚合函数
```

排序

升序

```
select studentNo,studentResult from result where studentNo=100 order by studentResult
select studentNo,studentResult from result where studentNo=100 order by studentResult asc
```

降序

```
select studentNo,studentResult from result where studentNo=100 order by studentResult desc
```

多列排序

```
select studentNo,studentResult from result where studentNo=100
order by
    studentNo asc ,
    studentResult desc
```

分页

```
select * from student limit (页数-1)*显示的条数 , 显示的条数
```

where/having

where

```
select * from student where phone like ' _%1%'
```

Having

```
select * from student where phone like ' _%1%' having studentName like '李%'
```

TCL

事物

ACID

Atomicity(原子性)/**Consistency**(一致性)/**Isolation**(隔离性)/**Durability**(持久性)

MySql 自动事物处理

```
set autocommit =0  //使用自动事物提交  
set autocommit =1  //不使用自动事物提交
```

事物的开启

```
begin;
```

事物的回滚

```
rollback;
```

事物的提交

```
commit;
```

示例

```
set autocommit =1  //不使用自动事物提交

begin; //开启事物

update tb_money set money=money+200 where name='jack';
update tb_money set money=money-200 where name='arvin';

commit;// rollback;

set autocommit =0  //使用自动事物提交
```

备份/还原

综合练习题

school.sql

```
DROP TABLE IF EXISTS `grade`;

CREATE TABLE `grade` (
  `GradeID` int(11) NOT NULL AUTO_INCREMENT COMMENT '年级编号',
  `GradeName` varchar(50) NOT NULL COMMENT '年级名称',
  PRIMARY KEY (`GradeID`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8;

insert into `grade`(`GradeID`,`GradeName`) values (1,'大一'),(2,'大二'),(3,'大三'),(4,'大四'),(5,'
预科班');
```

```
DROP TABLE IF EXISTS `result`;
```

```
CREATE TABLE `result` (  
  `StudentNo` int(4) NOT NULL COMMENT '学号',  
  `SubjectNo` int(4) NOT NULL COMMENT '课程编号',  
  `ExamDate` datetime NOT NULL COMMENT '考试日期',  
  `StudentResult` int(4) NOT NULL COMMENT '考试成绩',  
  KEY `SubjectNo` (`SubjectNo`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
insert      into      `result`(`StudentNo`,`SubjectNo`,`ExamDate`,`StudentResult`)      values  
(1000,1,'2013-11-11 16:00:00',94),(1000,2,'2012-11-10 10:00:00',75),(1000,3,'2011-12-19  
10:00:00',76),(1000,4,'2010-11-18 11:00:00',93),(1000,5,'2013-11-11  
14:00:00',97),(1000,6,'2012-09-13 15:00:00',87),(1000,7,'2011-10-16  
16:00:00',79),(1000,8,'2010-11-11 16:00:00',74),(1000,9,'2013-11-21  
10:00:00',69),(1000,10,'2012-11-11 12:00:00',78),(1000,11,'2011-11-11  
14:00:00',66),(1000,12,'2010-11-11 15:00:00',82),(1000,13,'2013-11-11  
14:00:00',94),(1000,14,'2012-11-11 15:00:00',98),(1000,15,'2011-12-11  
10:00:00',70),(1000,16,'2010-09-11 10:00:00',74),(1001,1,'2013-11-11  
16:00:00',76),(1001,2,'2012-11-10 10:00:00',93),(1001,3,'2011-12-19  
10:00:00',65),(1001,4,'2010-11-18 11:00:00',71),(1001,5,'2013-11-11  
14:00:00',98),(1001,6,'2012-09-13 15:00:00',74),(1001,7,'2011-10-16  
16:00:00',85),(1001,8,'2010-11-11 16:00:00',69),(1001,9,'2013-11-21  
10:00:00',63),(1001,10,'2012-11-11 12:00:00',70),(1001,11,'2011-11-11  
14:00:00',62),(1001,12,'2010-11-11 15:00:00',90),(1001,13,'2013-11-11  
14:00:00',97),(1001,14,'2012-11-11 15:00:00',89),(1001,15,'2011-12-11  
10:00:00',72),(1001,16,'2010-09-11 10:00:00',90),(1002,1,'2013-11-11  
16:00:00',61),(1002,2,'2012-11-10 10:00:00',80),(1002,3,'2011-12-19  
10:00:00',89),(1002,4,'2010-11-18 11:00:00',88),(1002,5,'2013-11-11  
14:00:00',82),(1002,6,'2012-09-13 15:00:00',91),(1002,7,'2011-10-16  
16:00:00',63),(1002,8,'2010-11-11 16:00:00',84),(1002,9,'2013-11-21  
10:00:00',60),(1002,10,'2012-11-11 12:00:00',71),(1002,11,'2011-11-11  
14:00:00',93),(1002,12,'2010-11-11 15:00:00',96),(1002,13,'2013-11-11  
14:00:00',83),(1002,14,'2012-11-11 15:00:00',69),(1002,15,'2011-12-11  
10:00:00',89),(1002,16,'2010-09-11 10:00:00',83),(1003,1,'2013-11-11  
16:00:00',91),(1003,2,'2012-11-10 10:00:00',75),(1003,3,'2011-12-19  
10:00:00',65),(1003,4,'2010-11-18 11:00:00',63),(1003,5,'2013-11-11  
14:00:00',90),(1003,6,'2012-09-13 15:00:00',96),(1003,7,'2011-10-16  
16:00:00',97),(1003,8,'2010-11-11 16:00:00',77),(1003,9,'2013-11-21  
10:00:00',62),(1003,10,'2012-11-11 12:00:00',81),(1003,11,'2011-11-11  
14:00:00',76),(1003,12,'2010-11-11 15:00:00',61),(1003,13,'2013-11-11  
14:00:00',93),(1003,14,'2012-11-11 15:00:00',79),(1003,15,'2011-12-11  
10:00:00',78),(1003,16,'2010-09-11 10:00:00',96),(1004,1,'2013-11-11
```

16:00:00',84),(1004,2,'2012-11-10	10:00:00',79),(1004,3,'2011-12-19
10:00:00',76),(1004,4,'2010-11-18	11:00:00',78),(1004,5,'2013-11-11
14:00:00',81),(1004,6,'2012-09-13	15:00:00',90),(1004,7,'2011-10-16
16:00:00',63),(1004,8,'2010-11-11	16:00:00',89),(1004,9,'2013-11-21
10:00:00',67),(1004,10,'2012-11-11	12:00:00',100),(1004,11,'2011-11-11
14:00:00',94),(1004,12,'2010-11-11	15:00:00',65),(1004,13,'2013-11-11
14:00:00',86),(1004,14,'2012-11-11	15:00:00',77),(1004,15,'2011-12-11
10:00:00',82),(1004,16,'2010-09-11	10:00:00',87),(1005,1,'2013-11-11
16:00:00',82),(1005,2,'2012-11-10	10:00:00',92),(1005,3,'2011-12-19
10:00:00',80),(1005,4,'2010-11-18	11:00:00',92),(1005,5,'2013-11-11
14:00:00',97),(1005,6,'2012-09-13	15:00:00',72),(1005,7,'2011-10-16
16:00:00',84),(1005,8,'2010-11-11	16:00:00',79),(1005,9,'2013-11-21
10:00:00',76),(1005,10,'2012-11-11	12:00:00',87),(1005,11,'2011-11-11
14:00:00',65),(1005,12,'2010-11-11	15:00:00',67),(1005,13,'2013-11-11
14:00:00',63),(1005,14,'2012-11-11	15:00:00',64),(1005,15,'2011-12-11
10:00:00',99),(1005,16,'2010-09-11	10:00:00',97),(1006,1,'2013-11-11
16:00:00',82),(1006,2,'2012-11-10	10:00:00',73),(1006,3,'2011-12-19
10:00:00',79),(1006,4,'2010-11-18	11:00:00',63),(1006,5,'2013-11-11
14:00:00',97),(1006,6,'2012-09-13	15:00:00',83),(1006,7,'2011-10-16
16:00:00',78),(1006,8,'2010-11-11	16:00:00',88),(1006,9,'2013-11-21
10:00:00',89),(1006,10,'2012-11-11	12:00:00',82),(1006,11,'2011-11-11
14:00:00',70),(1006,12,'2010-11-11	15:00:00',69),(1006,13,'2013-11-11
14:00:00',64),(1006,14,'2012-11-11	15:00:00',80),(1006,15,'2011-12-11
10:00:00',90),(1006,16,'2010-09-11	10:00:00',85),(1007,1,'2013-11-11
16:00:00',87),(1007,2,'2012-11-10	10:00:00',63),(1007,3,'2011-12-19
10:00:00',70),(1007,4,'2010-11-18	11:00:00',74),(1007,5,'2013-11-11
14:00:00',79),(1007,6,'2012-09-13	15:00:00',83),(1007,7,'2011-10-16
16:00:00',86),(1007,8,'2010-11-11	16:00:00',76),(1007,9,'2013-11-21
10:00:00',65),(1007,10,'2012-11-11	12:00:00',87),(1007,11,'2011-11-11
14:00:00',69),(1007,12,'2010-11-11	15:00:00',69),(1007,13,'2013-11-11
14:00:00',90),(1007,14,'2012-11-11	15:00:00',84),(1007,15,'2011-12-11
10:00:00',95),(1007,16,'2010-09-11	10:00:00',92),(1008,1,'2013-11-11
16:00:00',96),(1008,2,'2012-11-10	10:00:00',62),(1008,3,'2011-12-19
10:00:00',97),(1008,4,'2010-11-18	11:00:00',84),(1008,5,'2013-11-11
14:00:00',86),(1008,6,'2012-09-13	15:00:00',72),(1008,7,'2011-10-16
16:00:00',67),(1008,8,'2010-11-11	16:00:00',83),(1008,9,'2013-11-21
10:00:00',86),(1008,10,'2012-11-11	12:00:00',60),(1008,11,'2011-11-11
14:00:00',61),(1008,12,'2010-11-11	15:00:00',68),(1008,13,'2013-11-11
14:00:00',99),(1008,14,'2012-11-11	15:00:00',77),(1008,15,'2011-12-11
10:00:00',73),(1008,16,'2010-09-11	10:00:00',78),(1009,1,'2013-11-11
16:00:00',67),(1009,2,'2012-11-10	10:00:00',70),(1009,3,'2011-12-19
10:00:00',75),(1009,4,'2010-11-18	11:00:00',92),(1009,5,'2013-11-11
14:00:00',76),(1009,6,'2012-09-13	15:00:00',90),(1009,7,'2011-10-16
16:00:00',62),(1009,8,'2010-11-11	16:00:00',68),(1009,9,'2013-11-21

10:00:00',70),(1009,10,'2012-11-11	12:00:00',83),(1009,11,'2011-11-11
14:00:00',88),(1009,12,'2010-11-11	15:00:00',65),(1009,13,'2013-11-11
14:00:00',91),(1009,14,'2012-11-11	15:00:00',99),(1009,15,'2011-12-11
10:00:00',65),(1009,16,'2010-09-11	10:00:00',83),(1010,1,'2013-11-11
16:00:00',83),(1010,2,'2012-11-10	10:00:00',87),(1010,3,'2011-12-19
10:00:00',89),(1010,4,'2010-11-18	11:00:00',99),(1010,5,'2013-11-11
14:00:00',91),(1010,6,'2012-09-13	15:00:00',96),(1010,7,'2011-10-16
16:00:00',72),(1010,8,'2010-11-11	16:00:00',72),(1010,9,'2013-11-21
10:00:00',98),(1010,10,'2012-11-11	12:00:00',73),(1010,11,'2011-11-11
14:00:00',68),(1010,12,'2010-11-11	15:00:00',62),(1010,13,'2013-11-11
14:00:00',67),(1010,14,'2012-11-11	15:00:00',69),(1010,15,'2011-12-11
10:00:00',71),(1010,16,'2010-09-11	10:00:00',66),(1011,1,'2013-11-11
16:00:00',62),(1011,2,'2012-11-10	10:00:00',72),(1011,3,'2011-12-19
10:00:00',96),(1011,4,'2010-11-18	11:00:00',64),(1011,5,'2013-11-11
14:00:00',89),(1011,6,'2012-09-13	15:00:00',91),(1011,7,'2011-10-16
16:00:00',95),(1011,8,'2010-11-11	16:00:00',96),(1011,9,'2013-11-21
10:00:00',89),(1011,10,'2012-11-11	12:00:00',73),(1011,11,'2011-11-11
14:00:00',82),(1011,12,'2010-11-11	15:00:00',98),(1011,13,'2013-11-11
14:00:00',66),(1011,14,'2012-11-11	15:00:00',69),(1011,15,'2011-12-11
10:00:00',91),(1011,16,'2010-09-11	10:00:00',69),(1012,1,'2013-11-11
16:00:00',86),(1012,2,'2012-11-10	10:00:00',66),(1012,3,'2011-12-19
10:00:00',97),(1012,4,'2010-11-18	11:00:00',69),(1012,5,'2013-11-11
14:00:00',70),(1012,6,'2012-09-13	15:00:00',74),(1012,7,'2011-10-16
16:00:00',91),(1012,8,'2010-11-11	16:00:00',97),(1012,9,'2013-11-21
10:00:00',84),(1012,10,'2012-11-11	12:00:00',82),(1012,11,'2011-11-11
14:00:00',90),(1012,12,'2010-11-11	15:00:00',91),(1012,13,'2013-11-11
14:00:00',91),(1012,14,'2012-11-11	15:00:00',97),(1012,15,'2011-12-11
10:00:00',85),(1012,16,'2010-09-11	10:00:00',90),(1013,1,'2013-11-11
16:00:00',73),(1013,2,'2012-11-10	10:00:00',69),(1013,3,'2011-12-19
10:00:00',91),(1013,4,'2010-11-18	11:00:00',72),(1013,5,'2013-11-11
14:00:00',76),(1013,6,'2012-09-13	15:00:00',87),(1013,7,'2011-10-16
16:00:00',61),(1013,8,'2010-11-11	16:00:00',77),(1013,9,'2013-11-21
10:00:00',83),(1013,10,'2012-11-11	12:00:00',99),(1013,11,'2011-11-11
14:00:00',91),(1013,12,'2010-11-11	15:00:00',84),(1013,13,'2013-11-11
14:00:00',98),(1013,14,'2012-11-11	15:00:00',74),(1013,15,'2011-12-11
10:00:00',92),(1013,16,'2010-09-11	10:00:00',90),(1014,1,'2013-11-11
16:00:00',64),(1014,2,'2012-11-10	10:00:00',81),(1014,3,'2011-12-19
10:00:00',79),(1014,4,'2010-11-18	11:00:00',74),(1014,5,'2013-11-11
14:00:00',65),(1014,6,'2012-09-13	15:00:00',88),(1014,7,'2011-10-16
16:00:00',86),(1014,8,'2010-11-11	16:00:00',77),(1014,9,'2013-11-21
10:00:00',86),(1014,10,'2012-11-11	12:00:00',85),(1014,11,'2011-11-11
14:00:00',86),(1014,12,'2010-11-11	15:00:00',75),(1014,13,'2013-11-11
14:00:00',89),(1014,14,'2012-11-11	15:00:00',79),(1014,15,'2011-12-11
10:00:00',73),(1014,16,'2010-09-11	10:00:00',68),(1015,1,'2013-11-11


```

16:00:00',99),(1015,2,'2012-11-10
10:00:00',60),(1015,4,'2010-11-18
14:00:00',78),(1015,6,'2012-09-13
16:00:00',84),(1015,8,'2010-11-11
10:00:00',93),(1015,10,'2012-11-11
14:00:00',74),(1015,12,'2010-11-11
14:00:00',63),(1015,14,'2012-11-11
10:00:00',67),(1015,16,'2010-09-11
16:00:00',97),(1016,2,'2012-11-10
10:00:00',77),(1016,4,'2010-11-18
14:00:00',75),(1016,6,'2012-09-13
16:00:00',96),(1016,8,'2010-11-11
10:00:00',62),(1016,10,'2012-11-11
14:00:00',98),(1016,12,'2010-11-11
14:00:00',62),(1016,14,'2012-11-11
10:00:00',76),(1016,16,'2010-09-11
16:00:00',100),(1017,2,'2012-11-10
10:00:00',86),(1017,4,'2010-11-18
14:00:00',96),(1017,6,'2012-09-13
16:00:00',81),(1017,8,'2010-11-11
10:00:00',76),(1017,10,'2012-11-11
14:00:00',73),(1017,12,'2010-11-11
14:00:00',85),(1017,14,'2012-11-11
10:00:00',99),(1017,16,'2010-09-11 10:00:00',76);

```

```

10:00:00',60),(1015,3,'2011-12-19
11:00:00',75),(1015,5,'2013-11-11
15:00:00',78),(1015,7,'2011-10-16
16:00:00',95),(1015,9,'2013-11-21
12:00:00',79),(1015,11,'2011-11-11
15:00:00',65),(1015,13,'2013-11-11
15:00:00',74),(1015,15,'2011-12-11
10:00:00',65),(1016,1,'2013-11-11
10:00:00',90),(1016,3,'2011-12-19
11:00:00',75),(1016,5,'2013-11-11
15:00:00',97),(1016,7,'2011-10-16
16:00:00',92),(1016,9,'2013-11-21
12:00:00',83),(1016,11,'2011-11-11
15:00:00',94),(1016,13,'2013-11-11
15:00:00',97),(1016,15,'2011-12-11
10:00:00',82),(1017,1,'2013-11-11
10:00:00',88),(1017,3,'2011-12-19
11:00:00',73),(1017,5,'2013-11-11
15:00:00',64),(1017,7,'2011-10-16
16:00:00',66),(1017,9,'2013-11-21
12:00:00',95),(1017,11,'2011-11-11
15:00:00',82),(1017,13,'2013-11-11
15:00:00',68),(1017,15,'2011-12-11

```

```
DROP TABLE IF EXISTS `student`;
```

```

CREATE TABLE `student` (
  `StudentNo` int(4) NOT NULL COMMENT '学号',
  `LoginPwd` varchar(20) DEFAULT NULL,
  `StudentName` varchar(20) DEFAULT NULL COMMENT '学生姓名',
  `Sex` tinyint(1) DEFAULT NULL COMMENT '性别，取值 0 或 1',
  `GradeId` int(11) DEFAULT NULL COMMENT '年级编号',
  `Phone` varchar(50) NOT NULL COMMENT '联系电话，允许为空，即可选输入',
  `Address` varchar(255) NOT NULL COMMENT '地址，允许为空，即可选输入',
  `BornDate` datetime DEFAULT NULL COMMENT '出生时间',
  `Email` varchar(50) NOT NULL COMMENT '邮箱账号，允许为空，即可选输入',
  `IdentityCard` varchar(18) DEFAULT NULL COMMENT '身份证号',
  PRIMARY KEY (`StudentNo`),
  UNIQUE KEY `IdentityCard` (`IdentityCard`),
  KEY `Email` (`Email`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;

```

```

insert into
`student`(`StudentNo`,`LoginPwd`,`StudentName`,`Sex`,`Gradeld`,`Phone`,`Address`,`BornDate`,`
Email`,`IdentityCard`) values (1000,'111111','郭靖',1,1,'13500000001','北京海淀区中关村大街 1
号 ','1986-12-11 00:00:00','test1@bdqn.cn','450323198612111234'),(1001,'123456','李文才
',1,2,'13500000002','河南 洛 阳 ','1981-12-31
00:00:00','test1@bdqn.cn','450323198112311234'),(1002,'111111','李斯文',1,1,'13500000003','
天津市和平区','1986-11-30 00:00:00','test1@bdqn.cn','450323198611301234'),(1003,'123456','
武 松 ',1,3,'13500000004','上 海 卢 湾 区 ','1986-12-31
00:00:00','test1@bdqn.cn','450323198612314234'),(1004,'123456','张三',1,4,'13500000005','北
京市通州','1989-12-31 00:00:00','test1@bdqn.cn','450323198612311244'),(1005,'123456','张秋
丽 ',2,1,'13500000006','广 西 桂 林 市 灵 川 ','1986-12-31
00:00:00','test1@bdqn.cn','450323198612311214'),(1006,'123456','肖梅',2,4,'13500000007','地
址不详','1986-12-31 00:00:00','test1@bdqn.cn','450323198612311134'),(1007,'111111','欧阳峻
峰 ',1,1,'13500000008','北 京 东 城 区 ','1986-12-31
00:00:00','test1@bdqn.cn','450323198612311133'),(1008,'111111','梅超风',1,1,'13500000009','
河南洛阳','1986-12-31 00:00:00','test1@bdqn.cn','450323198612311221'),(1009,'123456','刘毅
',1,2,'13500000011','安 徽 ','1986-12-31
00:00:00','test1@bdqn.cn','450323198612311231'),(1010,'111111','大凡',1,1,'13500000012','河
南洛阳','1986-12-31 00:00:00','test1@bdqn.cn','450323198612311044'),(1011,'111111','奥丹斯
',1,1,'13500000013','北 京 海 淀 区 中 关 村 大 街 * 号 ','1984-12-31
00:00:00','test1@bdqn.cn','450323198412311234'),(1012,'123456','多伦',2,3,'13500000014','广
西 南 宁 中 央 大 街 ','1986-12-31
00:00:00','test1@bdqn.cn','450323198612311334'),(1013,'123456','李梅',2,1,'13500000015','上
海卢湾区','1986-12-31 00:00:00','test1@bdqn.cn','450323198612311534'),(1014,'123456','张得
',2,4,'13500000016','北 京 海 淀 区 中 关 村 大 街 * 号 ','1986-12-31
00:00:00','test1@bdqn.cn','450323198612311264'),(1015,'123456','李东方',1,4,'13500000017','
广 西 桂 林 市 灵 川 ','1976-12-31
00:00:00','test1@bdqn.cn','450323197612311234'),(1016,'111111','刘奋斗',1,1,'13500000018','
上海卢湾区','1986-12-31 00:00:00','test1@bdqn.cn','450323198612311251'),(1017,'123456','可
可 ',2,3,'13500000019','北 京 长 安 街 1 号 ','1981-09-10
00:00:00','test1@bdqn.cn','450323198109108311'),(10066,'"Tom',1,1,'13500000000','2017-11
-21 17:26:35','email@22.com','33123123123123123');

```

```

DROP TABLE IF EXISTS `subject`;

```

```

CREATE TABLE `subject` (
  `SubjectNo` int(11) NOT NULL AUTO_INCREMENT COMMENT '课程编号',
  `SubjectName` varchar(50) DEFAULT NULL COMMENT '课程名称',
  `ClassHour` int(4) DEFAULT NULL COMMENT '学时',
  `GradeID` int(4) DEFAULT NULL COMMENT '年级编号',
  PRIMARY KEY (`SubjectNo`)
) ENGINE=InnoDB AUTO_INCREMENT=18 DEFAULT CHARSET=utf8;

```

```
insert into `subject`(`SubjectNo`,`SubjectName`,`ClassHour`,`GradeID`) values (1,'高等数学-1',110,1),(2,'高等数学-2',110,2),(3,'高等数学-3',100,3),(4,'高等数学-4',130,4),(5,'C 语言-1',110,1),(6,'C 语言-2',110,2),(7,'C 语言-3',100,3),(8,'C 语言-4',130,4),(9,'JAVA 第一学年',110,1),(10,'JAVA 第二学年',110,2),(11,'JAVA 第三学年',100,3),(12,'JAVA 第四学年',130,4),(13,'数据库结构-1',110,1),(14,'数据库结构-2',110,2),(15,'数据库结构-3',100,3),(16,'数据库结构-4',130,4),(17,'C#基础',130,1);
```

1.统计出各个年级男生的人数

参考效果

年级	性别	人数
大一	男	8
大二	男	2
大三	男	1
大四	男	2

参考答案

```
SELECT
    (SELECT gar.gradename FROM grade gar WHERE gar.gradeid=stu.GradeID) AS 年级,
    (
        CASE sex
            WHEN 1 THEN '男'
            WHEN 2 THEN '女'
        END
    ) AS 性别,
    COUNT(1) AS 人数
FROM
    student stu
WHERE
    stu.sex=1
GROUP BY
    stu.gradeid
```

2.查询出参加“高等数学-4”及格和不及格的男生女生人数

参考效果

SubjectName	及格男	及格女	不及格男	不及格女
高等数学-4	2	2	0	0

参考答案

```
SELECT
    sub.`SubjectName`,
    SUM(res.`StudentResult`>=60 AND stu.`Sex`=1) AS 及格男,
    SUM(res.`StudentResult`>=60 AND stu.`Sex`=2) AS 及格女,
    SUM(res.`StudentResult`<60 AND stu.`Sex`=1) AS 不及格男,
    SUM(res.`StudentResult`<60 AND stu.`Sex`=2) AS 不及格女
FROM
    result res
INNER JOIN student stu ON stu.`StudentNo`=res.`StudentNo`
INNER JOIN `SUBJECT` sub ON sub.`SubjectNo`=res.`SubjectNo`
INNER JOIN grade gra ON gra.`GradeID`=stu.`GradeId` AND gra.`GradeID`=sub.`GradeID`
WHERE sub.`SubjectName`='高等数学-4'
```

3.查询出“郭靖”所学的所有科目的考试情况,并且根据考试成绩降序

参考效果

StudentName	SubjectName	StudentResult
郭靖	C语言-1	97
郭靖	数据库结构-1	94
郭靖	高等数学-1	94
郭靖	JAVA第一学年	69

参考答案

```
SELECT stu.`StudentName`,sub.`SubjectName`,res.`StudentResult` FROM student stu  
INNER JOIN SUBJECT  sub ON sub.`GradeID`=stu.`GradeId`  
INNER JOIN result res ON res.`StudentNo`=stu.`StudentNo` AND res.`SubjectNo`=sub.`SubjectNo`  
WHERE stu.`StudentName`='郭靖'  
ORDER BY res.`StudentResult` DESC
```

4.查询出各班级各科目考试的男女及格比例

参考效果

年级	科目	性别	合格率
大一	C语言-1	男	71.4286%
大一	C语言-1	女	50.0000%
大一	JAVA第一学年	男	42.8571%
大一	JAVA第一学年	女	50.0000%
大一	数据库结构-1	男	57.1429%
大一	数据库结构-1	女	50.0000%
大一	高等数学-1	男	71.4286%
大一	高等数学-1	女	50.0000%
大三	C语言-3	男	100.0000%
大三	C语言-3	女	100.0000%
大三	JAVA第三学年	男	0.0000%
大三	JAVA第三学年	女	50.0000%
大三	数据库结构-3	男	0.0000%
大三	数据库结构-3	女	100.0000%
大三	高等数学-3	男	0.0000%
大三	高等数学-3	女	100.0000%
大二	C语言-2	男	50.0000%
大二	JAVA第二学年	男	50.0000%
大二	数据库结构-2	男	100.0000%
大二	高等数学-2	男	50.0000%
大四	C语言-4	男	100.0000%
大四	C语言-4	女	50.0000%
大四	JAVA第四学年	男	0.0000%
大四	JAVA第四学年	女	0.0000%
大四	数据库结构-4	男	50.0000%
大四	数据库结构-4	女	50.0000%
大四	高等数学-4	男	0.0000%
大四	高等数学-4	女	0.0000%

参考答案

```
SELECT
    gra.`GradeName` 年级,
    sub.`SubjectName` 科目,
    (
        CASE stu.`Sex`
            WHEN 1 THEN '男'
            WHEN 2 THEN '女'
        END
    ) AS 性别,
    CONCAT((SUM(res.`StudentResult`>=60)/COUNT(1)*100),'%') AS 合格率
FROM
    result res
INNER JOIN student stu ON stu.`StudentNo`=res.`StudentNo`
INNER JOIN `SUBJECT` sub ON sub.`SubjectNo`=res.`SubjectNo`
INNER JOIN grade gra ON gra.`GradeID`=stu.`GradeID` AND gra.`GradeID`=sub.`GradeID`
GROUP BY gra.`GradeName`,sub.`SubjectName`,stu.`Sex`
```