



武汉生物工程学院

# MySQL数据库

主讲教师：朱 华





## 多表连接查询

数据库设计时为了**避免数据冗余**，将大量数据**分成若干数据表存放**，表与表之间的数据关联性**通过设置外键实现**。

在检索数据时需要**对多张表的数据进行筛选**，筛选出满足用户要求的数据。查询时在查询语句中的**from子句指定多个数据源**，表连接的语法格式为：





## 多表连接查询

Select 字段列表 from 表名1 [连接类型]  
join 表名2 on 表1和表2之间的连接条件;

说明:

连接类型: 主要有inner连接（内连接）和  
outer连接（外连接），而外连接又分为left join  
（左外连接，简称左连接）、right join（右连接）  
以及full（完全外连接）。

on: 表与表之间的连接条件，例如表1. 学号=  
表2. 学号，其中学号字段类型及字长应完全一致。



## 全连接

表与表之间的连接没有任何条件筛选，连接后的结果集包含连接表的全部数据，结果集中的**字段数是连接表字段数总和，记录数是连接表记录数的乘积**。

例choose库中的表classes和表choose，其表结构为：





## 全连接

```
mysql> desc classes;
```

Field	Type	Null	Key	Default	Extra
班级编号	char(4)	NO	PRI	NULL	
班级名称	char(20)	NO	UNI	NULL	
学院编号	int(11)	YES	MUL	NULL	

3 rows in set (0.03 sec)

```
mysql> desc choose;
```

Field	Type	Null	Key	Default	Extra
学号	char(11)	NO	PRI		
课程号	int(11)	NO	PRI	0	
成绩	tinyint(3) unsigned	YES		NULL	
选课时间	datetime	YES		CURRENT_TIMESTAMP	

4 rows in set (0.02 sec)





## 全连接

例choose库中的表classes和表choose，表中的数据为：

```
mysql> select * from classes;
```

班级编号	班级名称	学院编号
01	19计算机科学与技术1班	3
02	19计算机科学与技术2班	3
03	19计算机科学与技术3班	3
04	19信息管理1班	4
05	19机电1班	1
06	19机电2班	1
07	19生工1班	5

```
7 rows in set (0.02 sec)
```

```
mysql> select * from choose;
```

学号	课程号	成绩	选课时间
01640401	1	50	2019-09-22 15:26:07
01640401	2	40	2019-09-22 15:26:07
01640401	3	60	2019-09-22 15:26:07
01640402	2	70	2019-09-22 15:26:07
01640403	1	80	2019-09-22 15:26:07
01640403	2	90	2019-09-22 15:26:07
01640404	3	0	2019-09-22 15:26:07
01640405	1	0	2019-09-22 15:26:07

```
8 rows in set (0.00 sec)
```



## 全连接

例choose库中的表classes和表choose，全连接的sql语句为：

```
select classes.*, choose.* from classes  
join choose;
```

from子句产生的是一个中间结果，表中的每条记录都是与其他表中记录交叉产生的所有可能的组合，也就是笛卡尔积。





## 全连接

```
mysql> select classes.*,choose.* from choose join classes;
```

班级编号 号	成绩	班级名称 选课时间	学院编号	学号	课程
01 1	50	19计算机科学与技术1班 2019-09-22 15:26:07	3	01640401	
02 1	50	19计算机科学与技术2班 2019-09-22 15:26:07	3	01640401	
03 1	50	19计算机科学与技术3班 2019-09-22 15:26:07	3	01640401	
04 1	0	19信息管理1班 2019-09-22 15:26:07	4	01640405	
05 1	0	19机电1班 2019-09-22 15:26:07	1	01640405	
06 1	0	19机电2班 2019-09-22 15:26:07	1	01640405	
07 1	0	19生工1班 2019-09-22 15:26:07	5	01640405	

56 rows in set (0.02 sec)





## 内连接

全连接后的结果集存在**很多无效数据**，严重影响数据检索性能。如果按照某种条件筛选连接表时，结果集中都是满足要求的记录，这种表连接的方式是内连接。语法格式为：

**Select 字段列表 from 表名1 [inner] join 表名2 on 表1和表2之间的连接条件；**

其中inner关键字可以省略，只需指定on的条件。





## 内连接

例4.25 基于图书销售管理数据库，查询4号订单买家的详细信息。Sql语句如下：

```
select buyer.* from dingdan join buyer  
on dingdan.买家id=buyer.买家id where 订单  
id=201909004;
```





## 内连接

```
mysql> select * from buyer;
```

买家id	姓名	性别	家庭住址	联系电话
buyer1	彭万里	男	武生院学子南101	157895614
buyer2	高大山	男	武生院学子南102	1592345435
buyer3	孙子涵	女	武生院东八101	18879451689
buyer4	孙丹	女	武生院东八102	18295621876
buyer5	马建国	男	武生院北一101	15984651894
buyer6	王卫国	男	武生院北一102	18879456213
buyer7	周利人	男	武生院柏园101	18879136513

7 rows in set (0.04 sec)

```
mysql> select buyer.* from dingdan join buyer on dingdan.买家id=buyer.买家id where 订单id=201909004;
```

买家id	姓名	性别	家庭住址	联系电话
buyer4	孙丹	女	武生院东八102	18295621876

1 row in set (0.00 sec)



## 内连接

例4.26 基于学生选课数据库，查询学生的详细信息。Sql语句如下：

```
select 学号, 姓名, student. 班级编号, 班级名称  
from student join classes on student. 班级编号=classes. 班级编号;
```





## 内连接

```
mysql> select 学号,姓名,student.班级编号,班级名称 from student join classes on s  
tudent.班级编号=classes.班级编号;
```

学号	姓名	班级编号	班级名称
01640401	浅浅	01	19计算机科学与技术1班
01640402	猴子	01	19计算机科学与技术1班
01640403	魔兽	01	19计算机科学与技术1班
01640404	英雄	02	19计算机科学与技术2班
01640405	火王	03	19计算机科学与技术3班
01640406	马浅	04	19信息管理1班
01640407	田七	02	19计算机科学与技术2班

7 rows in set (0.00 sec)

```
mysql> select 学号,姓名,student.班级编号,班级名称 from classes join student on s  
tudent.班级编号=classes.班级编号;
```

学号	姓名	班级编号	班级名称
01640401	浅浅	01	19计算机科学与技术1班
01640402	猴子	01	19计算机科学与技术1班
01640403	魔兽	01	19计算机科学与技术1班
01640404	英雄	02	19计算机科学与技术2班
01640407	田七	02	19计算机科学与技术2班
01640405	火王	03	19计算机科学与技术3班
01640406	马浅	04	19信息管理1班

7 rows in set (0.00 sec)



## 内连接

select 查询语句中所有的子句仍然适用于表的连接查询，只是在from子句后多了数据源。

例4.27 基于学生选课数据库，查询班级人数少于2人的班级信息。Sql语句如下：

```
select count(学号), 班级名称 from  
student join classes on student. 班级编号  
=classes. 班级编号 group by student. 班级编号  
having count(学号)<2;
```





## 内连接

```
mysql> select count(学号),班级名称 from student join classes on student.班级编号=classes.班级编号 group by student.班级编号 having count(学号)<2;
```

count(学号)	班级名称
1	19计算机科学与技术3班
1	19信息管理1班

```
2 rows in set (0.00 sec)
```





## 内连接

例4.28 基于学生选课数据库，查询每个学生的选课门数。Sql语句如下：

必须加表名吗？

```
select student. 学号, 姓名, count(课程号)  
选课门数 from student join choose on  
student. 学号=choose. 学号 group by student.  
学号;
```







## 内连接

```
mysql> select 学号,姓名,count(课程号) 选课门数 from student join choose on student.学号=choose.学号 group by student.学号;  
ERROR 1052 (23000): Column '学号' in field list is ambiguous
```

```
mysql> select student.学号,姓名,count(课程号) 选课门数 from student join choose  
on student.学号=choose.学号 group by student.学号;  
+-----+-----+-----+  
| 学号      | 姓名      | 选课门数 |  
+-----+-----+-----+  
| 01640401  | 浅浅      | 3        |  
| 01640402  | 猴子      | 1        |  
| 01640403  | 魔兽      | 2        |  
| 01640404  | 英雄      | 1        |  
| 01640405  | 火王      | 1        |  
+-----+-----+-----+  
5 rows in set (0.01 sec)
```





## 外连接

与内连接不同，外连接（左连接或右连接）的连接条件值**筛选**一张表的数据，对另一张表**不进行筛选**（该表的所有记录出现在结果集中）。





## 左连接

语法格式为：

`select 字段列表 from 表1 left join 表2  
on 表1和表2之间的连接条件；`

表1左连接表2，查询结果集中须**包含表1的全部记录**，然后表1按指定的连接条件与表2进行连接，若表2中没有满足连接条件的记录，则结果集中表2相应的字段填入null。





## 左连接

例4. 29 将例4. 28中的连接改为左连接查询每个学生的选课门数。Sql语句如下：

```
select student. 学号, 姓名, count(课程号)  
选课门数 from student left join choose on  
student. 学号=choose. 学号 group by student.  
学号;
```





## 左连接

```
mysql> select student.学号,姓名,count(课程号) 选课门数 from student left join choose on student.学号=choose.学号 group by student.学号;
```

学号	姓名	选课门数
01640401	浅浅	3
01640402	猴子	1
01640403	魔兽	2
01640404	英雄	1
01640405	火王	1
01640406	马浅	0
01640407	田七	0

7 rows in set (0.00 sec)

```
mysql> select student.学号,姓名,count(课程号) 选课门数 from student join choose on student.学号=choose.学号 group by student.学号;
```

学号	姓名	选课门数
01640401	浅浅	3
01640402	猴子	1
01640403	魔兽	2
01640404	英雄	1
01640405	火王	1

5 rows in set (0.00 sec)



## 右连接

语法格式为：

`select 字段列表 from 表1 right join 表2  
on 表1和表2之间的连接条件；`

表1右连接表2，意味着查询结果集中须**包含表2的全部记录**，然后表2按指定的连接条件与表1进行连接，若表1中没有满足连接条件的记录，则结果集中表1相应的字段填入null。





## 右连接

例4.30 基于学生信息表student和班级信息表classes查询每个班级的男生人数，即使班级没有学生也要输出该班级信息。Sql语句如下：

```
select 班级名称, 性别, count(学号) 人数  
from student right join classes on student.  
班级编号=classes.班级编号 and 性别='男'  
group by 班级名称;
```





## 右连接

```
mysql> select 班级名称,性别,count(学号) 人数 from student right join classes on  
student.班级编号=classes.班级编号 and 性别='男' group by 班级名称;
```

班级名称	性别	人数
19机电1班	NULL	0
19机电2班	NULL	0
19计算机科学与技术1班	男	2
19计算机科学与技术2班	男	1
19计算机科学与技术3班	NULL	1
19生工1班	NULL	0
19信息管理1班	NULL	0

```
7 rows in set (0.14 sec)
```

```
mysql> select 班级名称,性别,count(学号) 人数 from student join classes on stu  
nt.班级编号=classes.班级编号 and 性别='男' group by 班级名称;
```

班级名称	性别	人数
19计算机科学与技术1班	男	2
19计算机科学与技术2班	男	1
19计算机科学与技术3班	男	1

```
3 rows in set (0.02 sec)
```





## 表的自身连接

两张表连接时如果表名相同称为表自身连接，  
语法格式为：

`select 字段列表 from 表1 a join 表1 b`  
`on a和b之间的连接条件；`

a和b是表1的重命名，连接时相当于表1被使用  
两次。





## 表的自身连接

例4.31 查询学生信息表student的出生日期小于“马浅”出生日期的学生学号、姓名、出生日期。Sql语句如下：

```
select a.学号, a.姓名, a.出生日期, b.姓名, b.  
出生日期 from student a join student b on a.  
出生日期 < b.出生日期 where b.姓名 = '马浅';
```





## 表的自身连接

```
mysql> select a.学号,a.姓名,a.出生日期,b.姓名,b.出生日期 from student a join student b on a.出生日期<b.出生日期 where b.姓名='马浅';
```

学号	姓名	出生日期	姓名	出生日期
01640403	魔兽	2001-02-20 00:00:00	马浅	2001-03-04 00:00:00
01640404	英雄	1998-03-18 00:00:00	马浅	2001-03-04 00:00:00
01640405	火王	1999-04-01 00:00:00	马浅	2001-03-04 00:00:00

3 rows in set (0.11 sec)





## 表的自身连接

例4.31中如果提前查询出“马浅”的出生日期，  
则Sql语句可以分为：

```
select 出生日期 from student where 姓名  
='马浅' ;
```

```
select 学号, 姓名, 出生日期 from student  
where 出生日期<'2001-03-04' ;
```





## 表的自身连接

```
mysql> select 出生日期 from student where 姓名='马浅';
```

```
+-----+  
| 出生日期 |  
+-----+  
| 2001-03-04 00:00:00 |  
+-----+  
1 row in set (0.00 sec)
```

```
mysql> select 学号,姓名,出生日期 from student where 出生日期<'2001-03-04';
```

```
+-----+-----+-----+  
| 学号 | 姓名 | 出生日期 |  
+-----+-----+-----+  
| 01640403 | 魔兽 | 2001-02-20 00:00:00 |  
| 01640404 | 英雄 | 1998-03-18 00:00:00 |  
| 01640405 | 火王 | 1999-04-01 00:00:00 |  
+-----+-----+-----+  
3 rows in set (0.00 sec)
```





## 小结

本节主要介绍了多表连接查询中全连接、内连接（inner join）、外连接（outer join）的用法，其中外连接中又分为左连接、右连接和完全连接，并且初步了解了对于查询中间接条件使用的方法——表的自身连接。

同学们需要熟练掌握上述连接的方法，通过合适的连接查询语句获取需要的信息。





## Sql语句拓展—别名

字段取别名：

字段名    as    别名

表取别名：

表名   别名

表的自身连接中首次涉及到表的别名概念

(p57)，使用表的别名对我们书写sql语句有何便利？





## Sql语句拓展—别名

例4.26 基于学生选课数据库，查询学生的详细信息。Sql语句如下：

```
select 学号, 姓名, student. 班级编号, 班级名称  
from student join classes on student. 班级编号=classes. 班级编号;
```







## Sql语句拓展—别名

例4.26 基于学生选课数据库，查询学生的详细信息。别名Sql语句如下：

```
select a.学号, a.姓名, a.班级编号, b.班级名称  
from student a, classes b where a.班级编号=b.班级编号;
```





## Sql语句拓展—别名

```
mysql> select a.学号,a.姓名,a.班级编号,b.班级名称 from student a,classes b where  
a.班级编号=b.班级编号;
```

学号	姓名	班级编号	班级名称
01640401	浅浅	01	19计算机科学与技术1班
01640402	猴子	01	19计算机科学与技术1班
01640403	魔兽	01	19计算机科学与技术1班
01640404	英雄	02	19计算机科学与技术2班
01640405	火王	03	19计算机科学与技术3班
01640406	马浅	04	19信息管理1班
01640407	田七	02	19计算机科学与技术2班

```
7 rows in set (0.14 sec)
```





## Sql语句拓展—别名

列出参加“C语言程序设计”或“mysql数据库”考试的所有学生的学号、姓名、课程名和成绩。

内连接参考答案：SELECT student. 学号, student. 姓名, grade. 课程名, grade. 成绩 FROM student join grade on student. 学号=grade. 学号 WHERE grade. 课程名='C语言程序设计' or grade. 课程名='mysql数据库'；

如果使用表的别名该如何书写sql语句？

