

数据库管理

NSD DATABASE

DAY03

内容

上午	09:00 ~ 09:30	作业讲解和回顾
	09:30 ~ 10:20	数据导入导出
	10:30 ~ 11:20	表记录基本操作
	11:30 ~ 12:00	查询及匹配条件
下午	14:00 ~ 14:50	
	15:00 ~ 15:50	多表查询
	16:10 ~ 17:00	
	17:10 ~ 18:00	总结和答疑



数据导入导出

数据导入导出

数据导入导出

设置数据导入/导出使用的目录

SQL数据导入

SQL数据导出

数据导入导出

设置数据导入/导出使用的目录

- 查看默认使用目录及目录是否存在

```
mysql> show variables like "secure_file_priv";
```

Variable_name	Value
secure_file_priv	/var/lib/mysql-files/

```
[root@localhost ~]# ls -ld /var/lib/mysql-files/  
drwxr-x---. 2 mysql mysql 31 4月 19 14:15 /var/lib/mysql-files/
```

设置数据导入/导出使用的目录（续1）

知识讲解

- 修改目录及查看修改结果

```
[root@localhost ~]# mkdir /myload ; chown mysql /myload
[root@localhost ~]# vim /etc/my.cnf
[mysqld]
secure_file_priv="/myload"
[root@localhost ~]# systemctl restart mysqld
mysql> show variables like "secure_file_priv";
+-----+-----+
| Variable_name | Value      |
+-----+-----+
| secure_file_priv | /myload/  |
+-----+-----+
```



SQL数据导入

知识讲解

- 基本用法
 - LOAD DATA INFILE “目录名/文件名”
 - INTO TABLE 表名
 - FIELDS TERMINATED BY “分隔符”
 - LINES TERMINATED BY “\n” ;
- 注意事项
 - 字段分隔符要与文件内的一致
 - 指定导入文件的绝对路径
 - 导入数据的表字段类型要与文件字段匹配
 - 禁用SELinux



SQL数据导入（续1）

- 案例需求
 - 将本地用户的记录导入userdb库的user表里
 - 为每条用户记录添加记录编号

知识讲解

```
mysql> create database userdb;
mysql> create table userdb.user(
    -> name char(50),
    -> password char(1),
    -> uid int(2),
    -> gid int(2),
    -> comment varchar(100),
    -> homedir char(60),
    -> shell char(50),
    -> index(name)
    -> );
```

表结构参考/etc/passwd 文件



SQL数据导入（续2）

```
mysql> load data infile "/myload/user.txt" into table userdb.user fields
terminated by ":" lines terminated by "\n";
```

知识讲解

```
mysql> alter table userdb.user add id int(2) zerofill primary key
auto_increment first;
```

```
mysql> select id ,name,uid from userdb.user limit 3;
```

```
+----+-----+-----+
| id  | name   | uid   |
+----+-----+-----+
| 01  | root   | 0     |
| 02  | bin    | 1     |
| 03  | daemon | 2     |
+----+-----+-----+
3 rows in set (0.00 sec)
```



SQL数据导出

知识讲解

- 基本用法
 - SELECT查询.. ..
INTO OUTFILE “目录名/文件名”
FIELDS TERMINATED BY “分隔符”
LINES TERMINATED BY “\n” ;
- 注意事项
 - 导出的内容由SQL查询语句决定
 - 禁用SELinux



SQL数据导出（续1）

知识讲解

- 案例需求
 - 导出userdb库user表中uid小于100的用户记录
 - 导出mysql库user表的前10条记录，只需包括user、host两个字段的的信息

```
mysql> select * from userdb.user where uid<100 into outfile  
"/myload/user2.txt";  
Query OK, 18 rows affected (0.00 sec)
```

```
mysql> select user,host from mysql.user into outfile  
"/myload/user3.txt";  
Query OK, 3 rows affected (0.00 sec)
```



案例1：SQL数据导入/导出

使用SQL语句完成下列导出、导入操作：

- 1) 将/etc/passwd文件导入userdb库user表并给每条记录加编号
- 2) 将userdb库user表中uid小于100的前10条记录导出，存为/myload/user2.txt文件

课堂练习



表记录基本操作

表记录基本操作

表记录的基本操作

insert 插入记录

update 更新记录

delete 删除记录

表记录的基本操作

insert 插入记录

- 格式1，仅指定部分字段的值
 - INSERT INTO 表名 (字段1,...,字段N)

VALUES

(字段1值, ..., 字段N值), ——— 第1条表记录

(字段1值, ..., 字段N值), ——— 第2条表记录

(字段1值, ..., 字段N值), ——— 第3条表记录

...;

insert 插入记录 (续1)

知识讲解

- 格式2，指定每一个字段的值
 - INSERT INTO 表名 (字段1,... ..,字段N)
VALUES
(字段1值, 字段2值, 字段N值),
(字段1值, 字段2值, 字段N值),
(字段1值, 字段2值, 字段N值),
... ;

第1条表记录

第2条表记录

第3条表记录



insert 插入记录 (续2)

知识讲解

- 注意事项
 - 字段值要与字段类型相匹配
 - 对于字符类型的字段，要用双或单引号括起来
 - 依次给所有字段赋值时，字段名可以省略
 - 只给一部分字段赋值时，必须明确写出对应的字段名称



insert 插入记录 (续3)

- 示例

- 分别向stu_info表中插入3条记录、1条记录

知识讲解

```
mysql> CREATE TABLE stu_info(  
  -> name varchar(12) NOT NULL,  
  -> gender enum("boy","girl") DEFAULT "boy",  
  -> age int(3) NOT NULL );  
Query OK, 0 rows affected (0.05 sec)  
  
mysql> INSERT INTO stu_info VALUES  
  -> ('Jim','girl',24),  
  -> ('Tom','boy',21),  
  -> ('Lily','girl',20);  
Query OK, 3 rows affected (0.00 sec)  
Records: 3 Duplicates: 0 Warnings: 0  
  
mysql> INSERT INTO stu_info(name,age)  
  -> VALUES('Jerry',27);  
Query OK, 1 row affected (0.01 sec)
```

```
mysql> SELECT * FROM stu_info;  
+-----+-----+-----+  
| name | gender | age |  
+-----+-----+-----+  
| Jim   | girl   | 24  |  
| Tom   | boy    | 21  |  
| Lily  | girl   | 20  |  
| Jerry | boy    | 27  |  
+-----+-----+-----+  
4 rows in set (0.00 sec)
```

列出所有表记录



update 更新记录

- 格式1，更新表内的所有记录

- UPDATE 表名

SET

字段1=字段1值,

字段2=字段2值,

字段N=字段N值;

知识讲解



update 更新记录 (续1)

知识讲解

- 格式2，只更新符合条件的部分记录
 - UPDATE 表名
 - SET
 - 字段1=字段1值,
 - 字段2=字段2值,
 - 字段N=字段N值;
 - WHERE 条件表达式;



update 更新记录 (续2)

知识讲解

- 注意事项
 - 字段值要与字段类型相匹配
 - 对于字符类型的字段，要用双或单引号括起来
 - 若不使用WHERE限定条件，会更新所有记录
 - 限定条件时，只更新匹配条件的记录



update 更新记录 (续3)

知识讲解

- 示例
 - 将stu_info表中所有人的age设置为10
 - 然后将其中性别为boy的人的年龄设置为20

```
mysql> UPDATE stu_info SET age=10;  
Query OK, 4 rows affected (0.01 sec)  
Rows matched: 4  Changed: 4  Warnings: 0  
  
mysql> UPDATE stu_info SET age=20 WHERE gender='boy';  
Query OK, 2 rows affected (0.03 sec)  
Rows matched: 2  Changed: 2  Warnings: 0
```



delete 删除记录

知识讲解

- 格式1, 仅删除符合条件的记录
 - DELETE FROM 表名 WHERE 条件表达式;
- 格式2, 删除所有的表记录
 - DELETE FROM 表名;



delete 删除记录 (续1)

- 示例
 - 删除stu_info表中未满16岁的人的记录
 - 然后再删除表中的所有记录

```
mysql> DELETE FROM stu_info WHERE age<=16;  
Query OK, 2 rows affected (0.00 sec)
```

```
mysql> DELETE FROM stu_info;  
Query OK, 2 rows affected (0.00 sec)
```

知识讲解



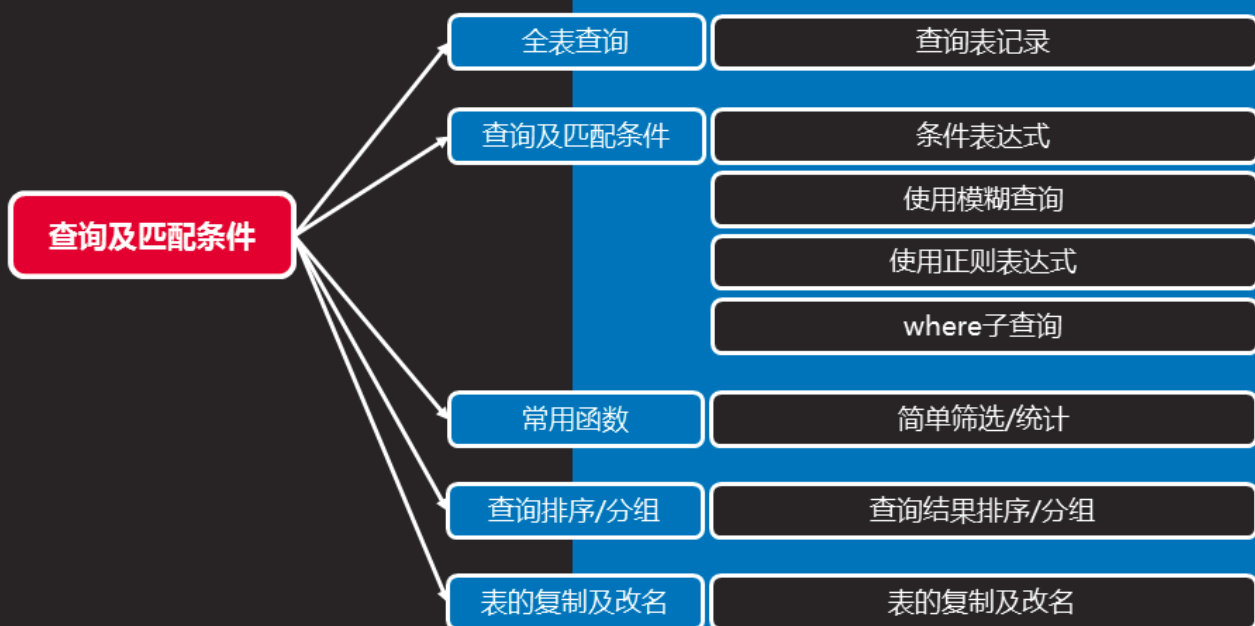
案例2：操作表记录

- 练习表记录的操作
 - 表记录的插入
 - 表记录的更新
 - 表记录的查询
 - 表记录的删除

课堂练习



查询及匹配条件



全表查询

查询表记录

知识讲解

- 格式1
 - SELECT 字段1, ..., 字段N FROM 表名;
- 格式2
 - SELECT 字段1, ..., 字段N FROM 表名
WHERE 条件表达式;
- 注意事项
 - 使用 * 可匹配所有字段
 - 指定表名时, 可采用 库名.表名 的形式



查询表记录 (续1)

知识讲解

- 应用示例
 - 列出stu_info表的所有记录
 - 列出stu_info表中女孩的名字、age信息

```
mysql> SELECT * FROM stu_info;
```

name	gender	age
Jim	girl	24
Tom	boy	21
Lily	girl	20
Jerry	boy	27

```
mysql> SELECT name,age FROM stu_info WHERE gender='girl';
```

name	age
Jim	24
Lily	20



查询表记录（续2）

- 应用示例

- 列出stu_info表中女孩的名字、age信息
- 其中name显示为“姓名”、age显示为“年龄”

```
mysql> SELECT name AS '姓名',age AS '年龄' FROM stu_info  
-> WHERE gender='girl';
```

姓名	年龄
Jim	24
Lily	20

为指定字段显示别名

查询及匹配条件

条件表达式

知识讲解

- 使用 WHERE 子句引导
 - 适用于更新、删除、查询等场合
 - 符合条件的记录才被操作
- 常见的匹配条件
 - 检查字段的数值大小
 - 字符串是否匹配
 - 逻辑测试、范围测试



条件表达式（续1）

知识讲解

- 数值的比较

类 型	用 途
=	等于
>、>=	大于、大于或等于
<、<=	小于、小于或等于
!=	不等于
BETWEEN .. AND ..	在 .. 与 .. 之间



条件表达式（续2）

- 逻辑比较，多条件组合

知识讲解

类 型	用 途
OR、AND、!	逻辑或、逻辑与、逻辑非
IN ..、NOT IN ..	在 .. 范围内、不在 .. 范围内
IS NULL	字段的值为空
IS NOT NULL	字段的值不为空
DISTINCT	不显示重复的值



条件表达式（续3）

- 运算操作

知识讲解

类 型	用 途
+	加法
-	减法
*	乘法
/	除法
%	取余数（求模）



使用模糊查询

知识讲解

- 基本用法
 - WHERE 字段名 LIKE '通配字符串'
 - 通配符 _ 匹配单个字符、% 匹配0~N个字符
- 示例
 - 列出name值“以 J 开头或以 Y 结尾”的记录

```
mysql> SELECT * FROM stu_info  
-> WHERE name LIKE 'J%' OR name LIKE '%y';
```

name	gender	age
Jim	girl	24
Lily	girl	20
Jerry	boy	27



使用正则表达式

知识讲解

- 基本用法
 - WHERE 字段名 REGEXP '正则表达式'
- 示例
 - 列出name值“以 J 开头或以 Y 结尾”的记录

```
mysql> SELECT * FROM stu_info WHERE name REGEXP '^J|Y$';
```

name	gender	age
Jim	girl	24
Lily	girl	20
Jerry	boy	27



where子查询

知识讲解

- 使用where子查询
 - 把内层查询结果作为外层查询的查询条件



where子查询（续1）

知识讲解

- 格式1
 - select 字段名列表 from 表A where 条件 (select 字段名列表 from 表A);
 - select 字段名列表 from 表A where 条件 (select 字段名列表 from 表A where 条件);
- 示例
 - 输出年龄小于平均年龄的学生的名字和年龄

```
select name,age from student where age < (select avg(age) from \ student);
```



where子查询 (续2)

知识讲解

- 格式2
 - select 字段名列表 from 表A where 条件
(select 字段名列表 from 表B);
 - select 字段名列表 from 表A where 条件
(select 字段名列表 from 表B where 条件);
- 示例
 - 输出成绩在80分以上学生的学号和名字

```
select stu_id,name from 学生表 where stu_id in  
(select stu_id \ from 成绩表 where 成绩 >80);
```



常用函数

简单筛选/统计

知识讲解

- 常用的统计函数
 - avg() : 集合的平均值
 - sum() : 对集合中的各参数求和
 - min() : 集合中的最小值
 - max() : 集合中的最大值
 - count() : 记录的个数



简单筛选/统计（续1）

知识讲解

- 应用示例
 - 输出stu_info表的记录个数、各学员年龄总和
 - 计算各学员的平均年龄、最大/小的岁数

```
mysql> SELECT count(*),sum(age) FROM stu_info;
```

count(*)	sum(age)
4	92

```
mysql> SELECT avg(age),max(age),min(age) FROM stu_info;
```

avg(age)	max(age)	min(age)
23.0000	27	20



查询排序/分组

查询结果排序/分组

- 基本用法
 - ORDER BY 字段名 [asc/desc]
 - GROUP BY 字段名 [HAVING 条件表达式]
 - LIMIT N,M 限制显示记录条目数

查询结果排序/分组（续1）

- 应用示例
 - 查询stu_info表，按年龄排序，输出前2条结果
 - 再输出降序排列的前2条结果

```
mysql> SELECT * FROM stu_info ORDER BY age LIMIT 2;
```

name	gender	age
Lily	girl	20
Tom	boy	21

```
mysql> SELECT * FROM stu_info ORDER BY age DESC LIMIT 2;
```

name	gender	age
Jerry	boy	27
Jim	girl	24

知识讲解



查询结果排序/分组（续2）

- 应用示例
 - 查询stu_info表，按性别进行分组
 - 分别统计boy、girl的人数

```
mysql> SELECT gender AS '性别', count(*) AS '人数' FROM stu_info  
GROUP BY gender;
```

性别	人数
boy	2
girl	2

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

知识讲解



表的复制及改名

表的复制及改名

- 将源表 xxx 复制为新表 yyy
 - CREATE TABLE yyy SELECT * FROM xxx;
- 将指定的查询结果复制为新表 zzz
 - CREATE TABLE zzz SQL查询语句;

```
mysql> CREATE TABLE yyuser SELECT * FROM mysql.user;  
Query OK, 4 rows affected (0.02 sec)  
Records: 4 Duplicates: 0 Warnings: 0
```

```
mysql> CREATE TABLE zzuser  
-> SELECT user,host,password FROM mysql.user;  
Query OK, 4 rows affected (0.04 sec)  
Records: 4 Duplicates: 0 Warnings: 0
```

表的复制及改名（续1）

知识讲解

- 复制源表 xxx 的结构到新表 vv
– CREATE TABLE vv SELECT * FROM xxx WHERE FALSE;
- 将源表 vv 的名称改为 ww
– ALTER TABLE vv RENAME TO ww;

```
mysql> CREATE TABLE vvuser SELECT * FROM mysql.user  
-> WHERE FALSE;  
Query OK, 0 rows affected (0.03 sec)  
Records: 0 Duplicates: 0 Warnings: 0  
  
mysql> ALTER TABLE vvuser RENAME TO wwuser;  
Query OK, 0 rows affected (0.00 sec)
```



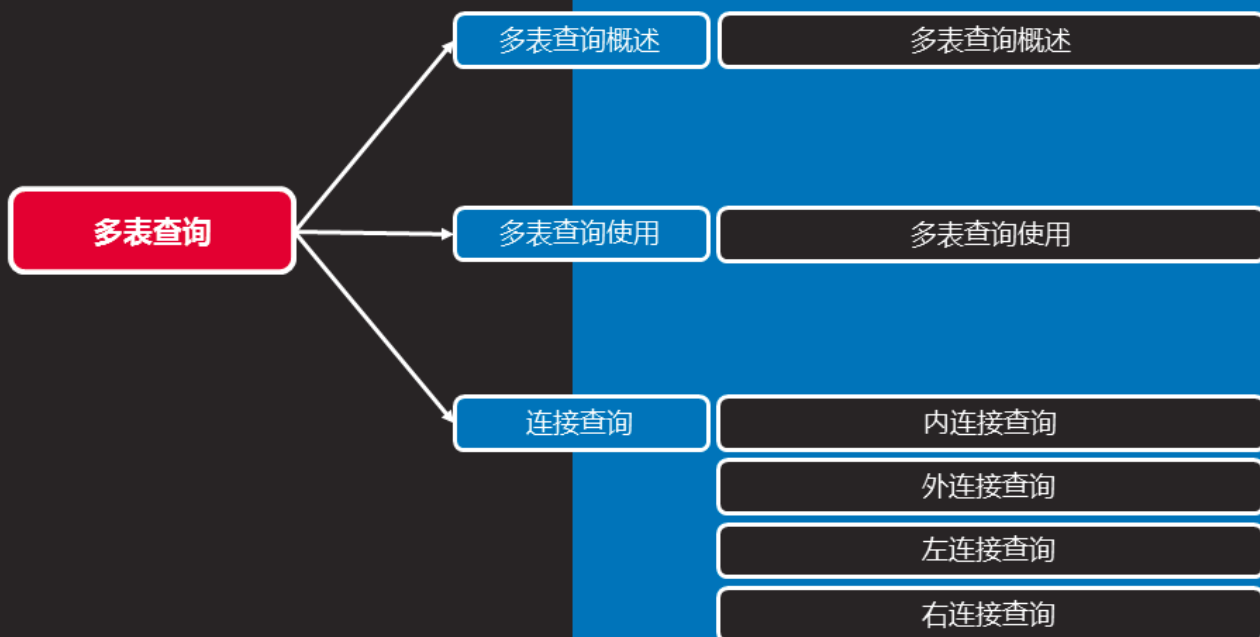
案例3：查询及匹配条件

课堂练习

- 练习常见的SQL查询及条件设置
 - 创建stu_info表，并插入数据
 - 练习常见SQL查询及条件设置



多表查询



多表查询概述

多表查询概述

知识讲解

- 多表查询（又称 连接查询）
 - 将2个或2个以上的表 按某个条件连接起来，从中选取需要的数据
 - 当多个表中 存在相同意义的字段（字段名可以不同）时，可以通过该字段连接多个表



多表查询使用

多表查询使用

知识讲解

- 格式1
 - select 字段名列表 from 表a,表b ;
- 格式2
 - select 字段名列表 from 表a,表b where 条件 ;

以上格式的查询结果叫笛卡尔集 显示查询结果的总条目数是 (表a的纪录数 * 表b的纪录数)



多表查询使用 (续1)

知识讲解

- 查询单表记录

```
mysql> select * from user;
```

id	username	password	uid	gid	shell
35	rpc	x	32	32	/sbin/nologin
36	rpcuser	x	29	29	/sbin/nologin
37	nfsnobody	x	65534	65534	/sbin/nologin
38	dhcpcd	x	177	177	/sbin/nologin
39	named	x	25	25	/sbin/nologin

5 rows in set (0.00 sec)

```
mysql> select * from user2;
```

id	username	comment	homedir	shell
1	root	root	/root	/bin/bash
2	bin	bin	/bin	/sbin/nologin
3	daemon	daemon	/sbin	/sbin/nologin
4	adm	adm	/var/adm	/sbin/nologin
5	lp	lp	/var/spool/lpd	/sbin/nologin

5 rows in set (0.00 sec)



多表查询使用（续2）

- 格式1示例

```
mysql> select * from user,user2;
```

id	username	password	uid	gid	shell	id	username	comment	homedir	shell
35	rpc	x	32	32	/sbin/nologin	1	root	root	/root	/bin/bash
36	rpcuser	x	29	29	/sbin/nologin	1	root	root	/root	/bin/bash
37	nfsnobody	x	65534	65534	/sbin/nologin	1	root	root	/root	/bin/bash
38	dhcpd	x	177	177	/sbin/nologin	1	root	root	/root	/bin/bash
39	named	x	25	25	/sbin/nologin	1	root	root	/root	/bin/bash
35	rpc	x	32	32	/sbin/nologin	2	bin	bin	/bin	/sbin/nologin
36	rpcuser	x	29	29	/sbin/nologin	2	bin	bin	/bin	/sbin/nologin
37	nfsnobody	x	65534	65534	/sbin/nologin	2	bin	bin	/bin	/sbin/nologin
38	dhcpd	x	177	177	/sbin/nologin	2	bin	bin	/bin	/sbin/nologin
39	named	x	25	25	/sbin/nologin	2	bin	bin	/bin	/sbin/nologin
35	rpc	x	32	32	/sbin/nologin	3	daemon	daemon	/sbin	/sbin/nologin
36	rpcuser	x	29	29	/sbin/nologin	3	daemon	daemon	/sbin	/sbin/nologin
37	nfsnobody	x	65534	65534	/sbin/nologin	3	daemon	daemon	/sbin	/sbin/nologin
38	dhcpd	x	177	177	/sbin/nologin	3	daemon	daemon	/sbin	/sbin/nologin
39	named	x	25	25	/sbin/nologin	3	daemon	daemon	/sbin	/sbin/nologin
35	rpc	x	32	32	/sbin/nologin	4	adm	adm	/var/adm	/sbin/nologin
36	rpcuser	x	29	29	/sbin/nologin	4	adm	adm	/var/adm	/sbin/nologin
37	nfsnobody	x	65534	65534	/sbin/nologin	4	adm	adm	/var/adm	/sbin/nologin
38	dhcpd	x	177	177	/sbin/nologin	4	adm	adm	/var/adm	/sbin/nologin
39	named	x	25	25	/sbin/nologin	4	adm	adm	/var/adm	/sbin/nologin
35	rpc	x	32	32	/sbin/nologin	5	lp	lp	/var/spool/lpd	/sbin/nologin
36	rpcuser	x	29	29	/sbin/nologin	5	lp	lp	/var/spool/lpd	/sbin/nologin
37	nfsnobody	x	65534	65534	/sbin/nologin	5	lp	lp	/var/spool/lpd	/sbin/nologin
38	dhcpd	x	177	177	/sbin/nologin	5	lp	lp	/var/spool/lpd	/sbin/nologin
39	named	x	25	25	/sbin/nologin	5	lp	lp	/var/spool/lpd	/sbin/nologin

25 rows in set (0.00 sec)

知识讲解



多表查询使用（续3）

- 格式2示例

```
mysql> select * from user2,user where user2.shell != user.shell;
```

id	username	comment	homedir	shell	id	username	password	uid	gid	shell
1	root	root	/root	/bin/bash	35	rpc	x	32	32	/sbin/nologin
1	root	root	/root	/bin/bash	36	rpcuser	x	29	29	/sbin/nologin
1	root	root	/root	/bin/bash	37	nfsnobody	x	65534	65534	/sbin/nologin
1	root	root	/root	/bin/bash	38	dhcpd	x	177	177	/sbin/nologin
1	root	root	/root	/bin/bash	39	named	x	25	25	/sbin/nologin

5 rows in set (0.00 sec)

```
mysql> select * from user2,user where user2.shell != user.shell and user.uid<100;
```

id	username	comment	homedir	shell	id	username	password	uid	gid	shell
1	root	root	/root	/bin/bash	35	rpc	x	32	32	/sbin/nologin
1	root	root	/root	/bin/bash	36	rpcuser	x	29	29	/sbin/nologin
1	root	root	/root	/bin/bash	39	named	x	25	25	/sbin/nologin

3 rows in set (0.00 sec)

知识讲解



多表查询使用（续4）

- 格式2示例

```
mysql> select user.id,user.username,user2.username,user2.shell from user,user2;
```

	id	username	username	shell
35	rpc	root	/bin/bash	
36	rpcuser	root	/bin/bash	
37	nfsnobody	root	/bin/bash	
38	dhcpd	root	/bin/bash	
39	named	root	/bin/bash	
35	rpc	bin	/sbin/nologin	
36	rpcuser	bin	/sbin/nologin	
37	nfsnobody	bin	/sbin/nologin	
38	dhcpd	bin	/sbin/nologin	
39	named	bin	/sbin/nologin	
35	rpc	daemon	/sbin/nologin	
36	rpcuser	daemon	/sbin/nologin	
37	nfsnobody	daemon	/sbin/nologin	
38	dhcpd	daemon	/sbin/nologin	
39	named	daemon	/sbin/nologin	
35	rpc	adm	/sbin/nologin	
36	rpcuser	adm	/sbin/nologin	
37	nfsnobody	adm	/sbin/nologin	
38	dhcpd	adm	/sbin/nologin	
39	named	adm	/sbin/nologin	
35	rpc	lp	/sbin/nologin	
36	rpcuser	lp	/sbin/nologin	
37	nfsnobody	lp	/sbin/nologin	
38	dhcpd	lp	/sbin/nologin	
39	named	lp	/sbin/nologin	

25 rows in set (0.00 sec)

知识讲解



连接查询

内连接查询

- 基本用法（只显示与查询条件匹配的记录）
 - select 字段名列表 from 表a,表b；

知识讲解



外连接查询

- 外连接查询
 - 左连接查询（以左表纪录为主）
 - 右连接查询（以右表纪录为主）

知识讲解



左连接查询

- 基本用法
 - select 字段名列表 from 表a LEFT JOIN 表b ON 条件表达式；

知识讲解



右连接查询

- 基本用法
 - select 字段名列表 from 表a RIGHT JOIN 表b ON 条件表达式；

知识讲解



总结和答疑

