MySQL安装

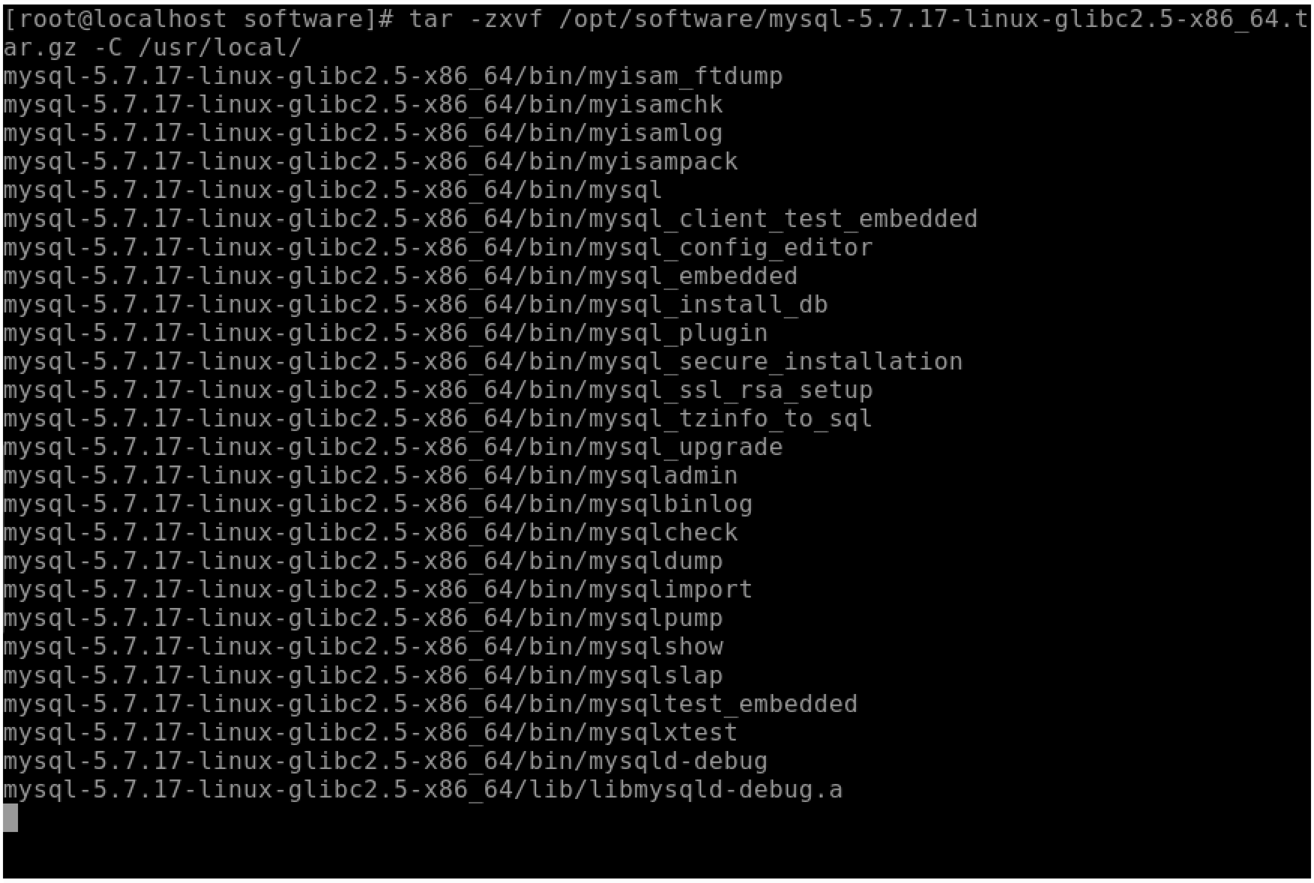
1. 解压安装包

进入opt/software目录下

```
cd /opt/software
```

执行如下命令解压MySQL安装包

```
tar -zxvf /opt/software/mysql-5.7.17-linux-glibc2.5-x86_64.tar.gz -C
/usr/local/
```

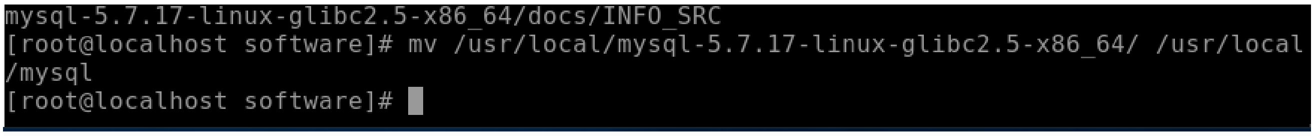


```
[root@localhost software]# tar -zxvf /opt/software/mysql-5.7.17-linux-glibc2.5-x86_64.t
ar.gz -C /usr/local/
mysql-5.7.17-linux-glibc2.5-x86_64/bin/myisam_ftdump
mysql-5.7.17-linux-glibc2.5-x86_64/bin/myisamchk
mysql-5.7.17-linux-glibc2.5-x86_64/bin/myisamlog
mysql-5.7.17-linux-glibc2.5-x86_64/bin/myisampack
mysql-5.7.17-linux-glibc2.5-x86_64/bin/mysql
mysql-5.7.17-linux-glibc2.5-x86_64/bin/mysql_client_test_embedded
mysql-5.7.17-linux-glibc2.5-x86_64/bin/mysql_config_editor
mysql-5.7.17-linux-glibc2.5-x86_64/bin/mysql_embedded
mysql-5.7.17-linux-glibc2.5-x86_64/bin/mysql_install_db
mysql-5.7.17-linux-glibc2.5-x86_64/bin/mysql_plugin
mysql-5.7.17-linux-glibc2.5-x86_64/bin/mysql_secure_installation
mysql-5.7.17-linux-glibc2.5-x86_64/bin/mysql_ssl_rsa_setup
mysql-5.7.17-linux-glibc2.5-x86_64/bin/mysql_tzinfo_to_sql
mysql-5.7.17-linux-glibc2.5-x86_64/bin/mysql_upgrade
mysql-5.7.17-linux-glibc2.5-x86_64/bin/mysqladmin
mysql-5.7.17-linux-glibc2.5-x86_64/bin/mysqlbinlog
mysql-5.7.17-linux-glibc2.5-x86_64/bin/mysqlcheck
mysql-5.7.17-linux-glibc2.5-x86_64/bin/mysqldump
mysql-5.7.17-linux-glibc2.5-x86_64/bin/mysqlimport
mysql-5.7.17-linux-glibc2.5-x86_64/bin/mysqlpump
mysql-5.7.17-linux-glibc2.5-x86_64/bin/mysqlshow
mysql-5.7.17-linux-glibc2.5-x86_64/bin/mysqlslap
mysql-5.7.17-linux-glibc2.5-x86_64/bin/mysqltest_embedded
mysql-5.7.17-linux-glibc2.5-x86_64/bin/mysqlxtest
mysql-5.7.17-linux-glibc2.5-x86_64/bin/mysqld-debug
mysql-5.7.17-linux-glibc2.5-x86_64/lib/libmysqld-debug.a
```

2. 目录重命名

执行如下命令将安装目录重命名为MySQL

```
mv /usr/local/mysql-5.7.17-linux-glibc2.5-x86_64/ /usr/local/mysql
```

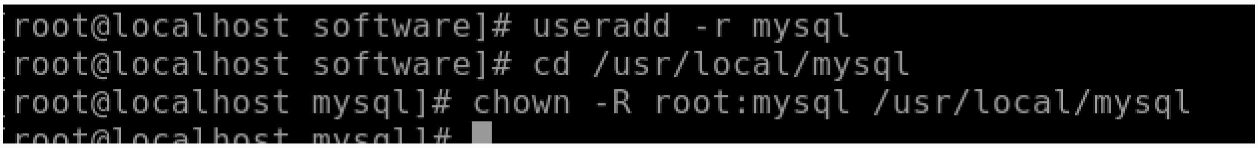


```
mysql-5.7.17-linux-glibc2.5-x86_64/docs/INFO_SRC
[root@localhost software]# mv /usr/local/mysql-5.7.17-linux-glibc2.5-x86_64/ /usr/local
/mysql
[root@localhost software]# 
```

3. 初始化用户

执行如下命令完成用户初始化

```
useradd -r mysql

cd /usr/local/mysql

chown -R root:mysql /usr/local/mysql
```



```
[root@localhost software]# useradd -r mysql
[root@localhost software]# cd /usr/local/mysql
[root@localhost mysql]# chown -R root:mysql /usr/local/mysql
[root@localhost mysql]# 
```

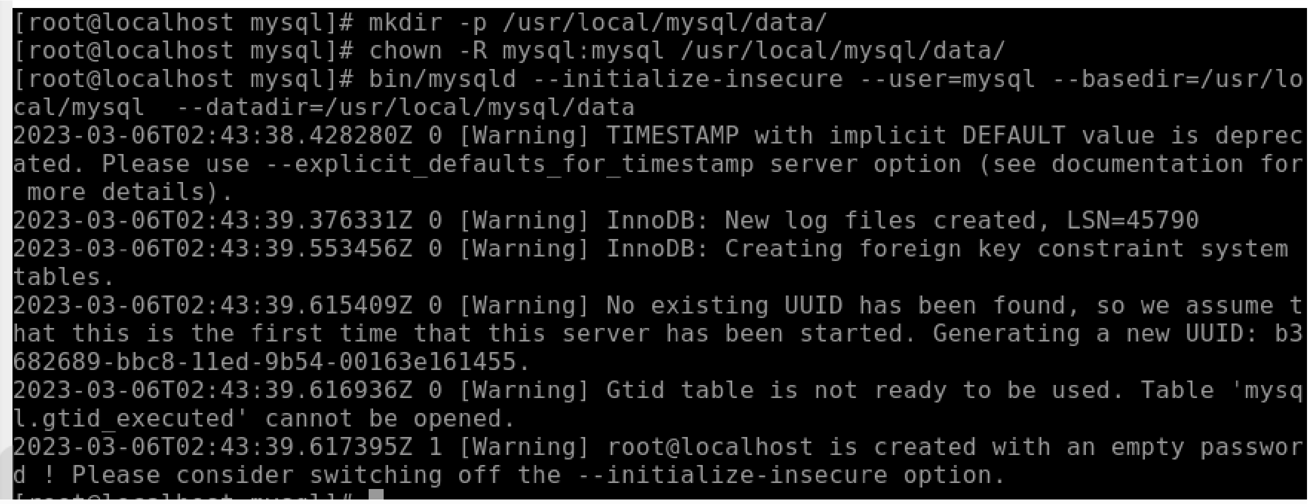
4. 创建数据目录

执行如下命令创建MySQL数据目录

```
mkdir -p /usr/local/mysql/data/
```

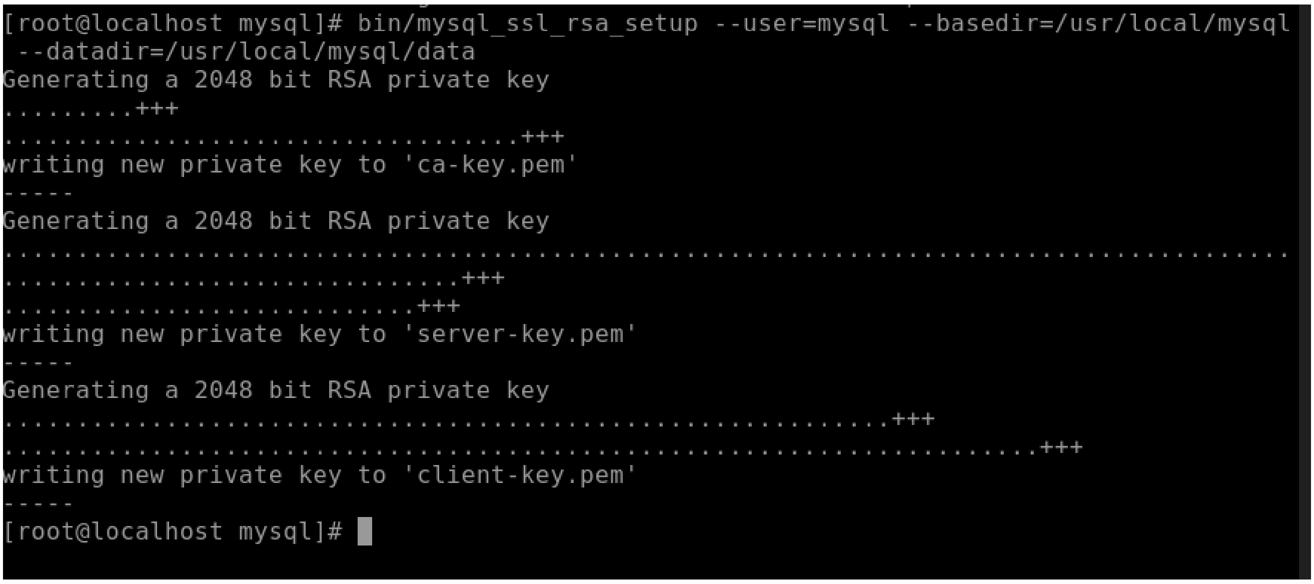
```
chown -R mysql:mysql /usr/local/mysql/data/
```

```
bin/mysqld --initialize-insecure --user=mysql --basedir=/usr/local/mysql     --datadir=/usr/local/mysql/data
```



```
[root@localhost mysql]# mkdir -p /usr/local/mysql/data/
[root@localhost mysql]# chown -R mysql:mysql /usr/local/mysql/data/
[root@localhost mysql]# bin/mysqld --initialize-insecure --user=mysql --basedir=/usr/lo
cal/mysql  --datadir=/usr/local/mysql/data
2023-03-06T02:43:38.428280Z 0 [Warning] TIMESTAMP with implicit DEFAULT value is deprec
ated. Please use --explicit_defaults_for_timestamp server option (see documentation for
 more details).
2023-03-06T02:43:39.376331Z 0 [Warning] InnoDB: New log files created, LSN=45790
2023-03-06T02:43:39.553456Z 0 [Warning] InnoDB: Creating foreign key constraint system
tables.
2023-03-06T02:43:39.615409Z 0 [Warning] No existing UUID has been found, so we assume t
hat this is the first time that this server has been started. Generating a new UUID: b3
682689-bbc8-11ed-9b54-00163e161455.
2023-03-06T02:43:39.616936Z 0 [Warning] Gtid table is not ready to be used. Table 'mysq
l.gtid_executed' cannot be opened.
2023-03-06T02:43:39.617395Z 1 [Warning] root@localhost is created with an empty passwor
d ! Please consider switching off the --initialize-insecure option.
[root@localhost mysql]# 
```

```
bin/mysql_ssl_rsa_setup --user=mysql --basedir=/usr/local/mysql --datadir=/usr/local/mysql/data
```



```
[root@localhost mysql]# bin/mysql_ssl_rsa_setup --user=mysql --basedir=/usr/local/mysql
 --datadir=/usr/local/mysql/data
Generating a 2048 bit RSA private key
.........+++
...................................+++
writing new private key to 'ca-key.pem'
-----
Generating a 2048 bit RSA private key
.......................................................................................
..............................+++
...........................+++
writing new private key to 'server-key.pem'
-----
Generating a 2048 bit RSA private key
............................................................+++
......................................................................+++
writing new private key to 'client-key.pem'
-----
[root@localhost mysql]#
```

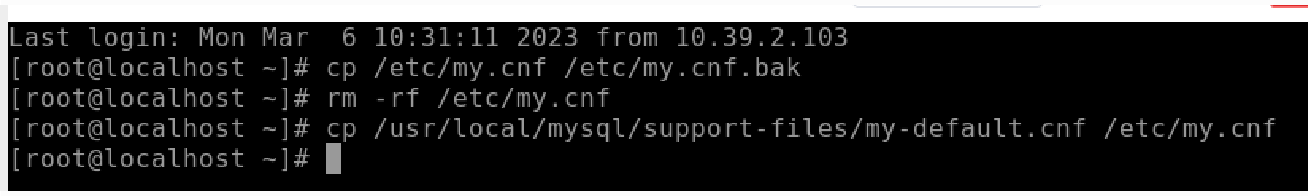
5. 配置MySQL

执行如下命令完成MYSQL配置

```
cp /etc/my.cnf /etc/my.cnf.bak
```

```
rm -rf /etc/my.cnf
```

```
cp /usr/local/mysql/support-files/my-default.cnf /etc/my.cnf
```

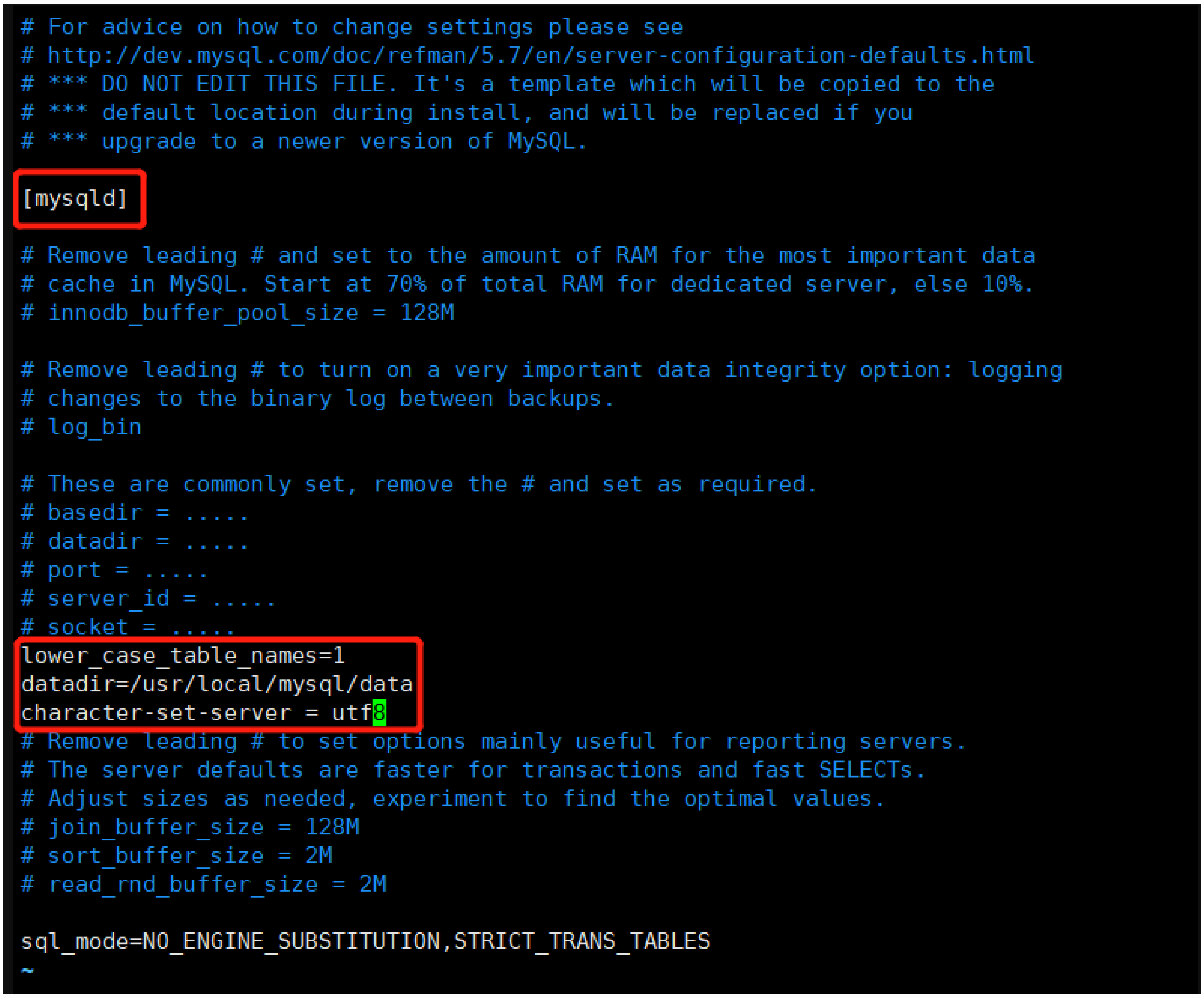


```
Last login: Mon Mar  6 10:31:11 2023 from 10.39.2.103
[root@localhost ~]# cp /etc/my.cnf /etc/my.cnf.bak
[root@localhost ~]# rm -rf /etc/my.cnf
[root@localhost ~]# cp /usr/local/mysql/support-files/my-default.cnf /etc/my.cnf
[root@localhost ~]#
```

在/etc/my.cnf目录下进入vim

输入

```
lower_case_table_names=1

datadir=/usr/local/mysql/data

character-set-server = utf8
```



```
# For advice on how to change settings please see
# http://dev.mysql.com/doc/refman/5.7/en/server-configuration-defaults.html
# *** DO NOT EDIT THIS FILE. It's a template which will be copied to the
# *** default location during install, and will be replaced if you
# *** upgrade to a newer version of MySQL.

[mysqld]

# Remove leading # and set to the amount of RAM for the most important data
# cache in MySQL. Start at 70% of total RAM for dedicated server, else 10%.
# innodb_buffer_pool_size = 128M

# Remove leading # to turn on a very important data integrity option: logging
# changes to the binary log between backups.
# log_bin

# These are commonly set, remove the # and set as required.
# basedir = .....
# datadir = .....
# port = .....
# server_id = .....
# socket = .....
lower_case_table_names=1
datadir=/usr/local/mysql/data
character-set-server = utf8
# Remove leading # to set options mainly useful for reporting servers.
# The server defaults are faster for transactions and fast SELECTs.
# Adjust sizes as needed, experiment to find the optimal values.
# join_buffer_size = 128M
# sort_buffer_size = 2M
# read_rnd_buffer_size = 2M

sql_mode=NO_ENGINE_SUBSTITUTION,STRICT_TRANS_TABLES
~
```

6. 设置自启动

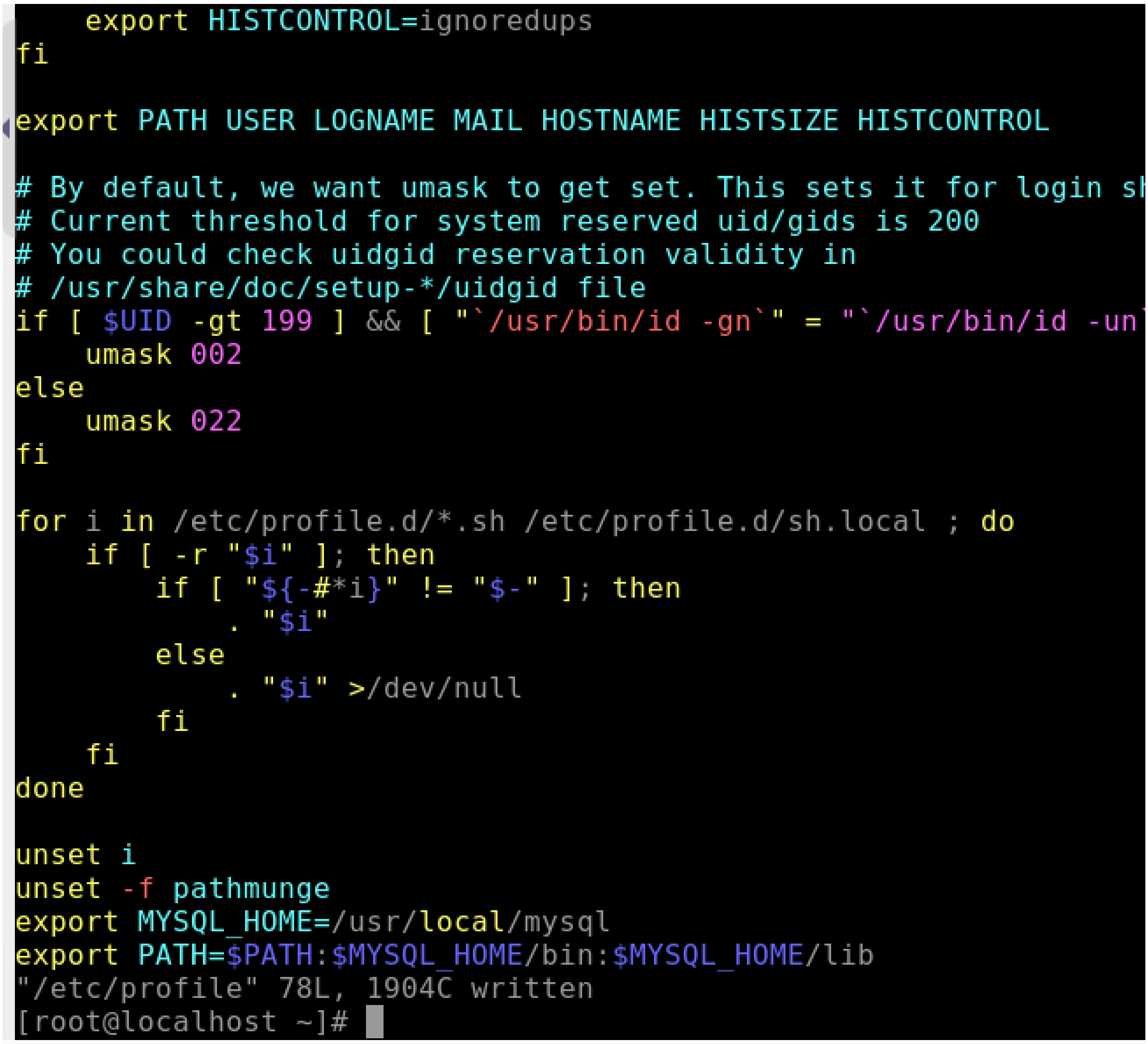
使用service命令管理mysql,设置mysql开机自启动,添加到环境变量中

```
cp /usr/local/mysql/support-files/mysql.server /etc/init.d/mysqld
```

```
chkconfig --add mysqld
```

添加mysql环境变量

输入 `vim /etc/profile`

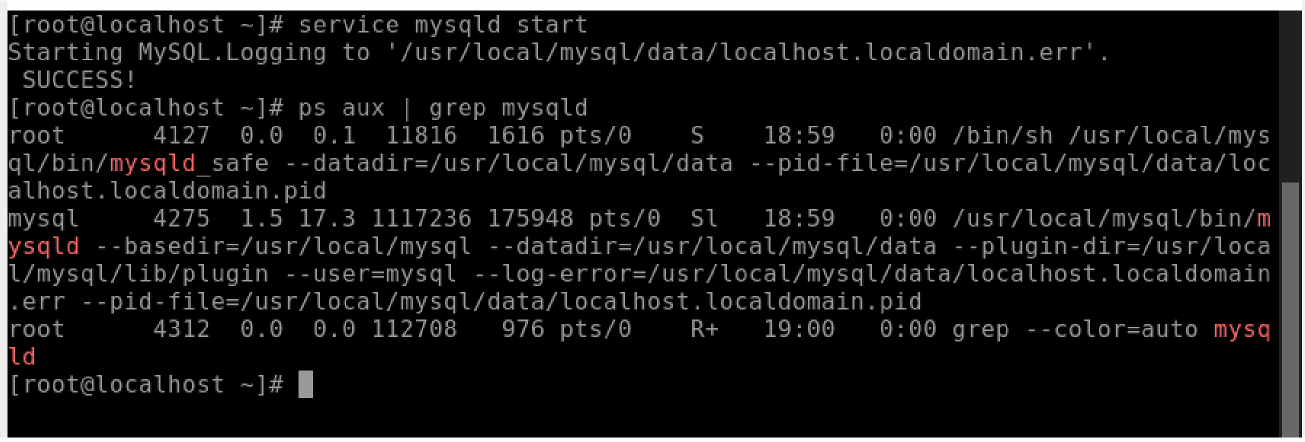


```
    export HISTCONTROL=ignoredups
fi

export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE HISTCONTROL

# By default, we want umask to get set. This sets it for login sh
# Current threshold for system reserved uid/gids is 200
# You could check uidgid reservation validity in
# /usr/share/doc/setup-*/uidgid file
if [ $UID -gt 199 ] && [ "`/usr/bin/id -gn`" = "`/usr/bin/id -un
    umask 002
else
    umask 022
fi

for i in /etc/profile.d/*.sh /etc/profile.d/sh.local ; do
    if [ -r "$i" ]; then
        if [ "${-#*i}" != "$-" ]; then
            . "$i"
        else
            . "$i" >/dev/null
        fi
    fi
done

unset i
unset -f pathmunge
export MYSQL_HOME=/usr/local/mysql
export PATH=$PATH:$MYSQL_HOME/bin:$MYSQL_HOME/lib
"/etc/profile" 78L, 1904C written
[root@localhost ~]#
```

执行 `source /etc/profile` 让环境变量生效

7. 启动MySQL

执行如下命令启动mysql并查看是否启动成功



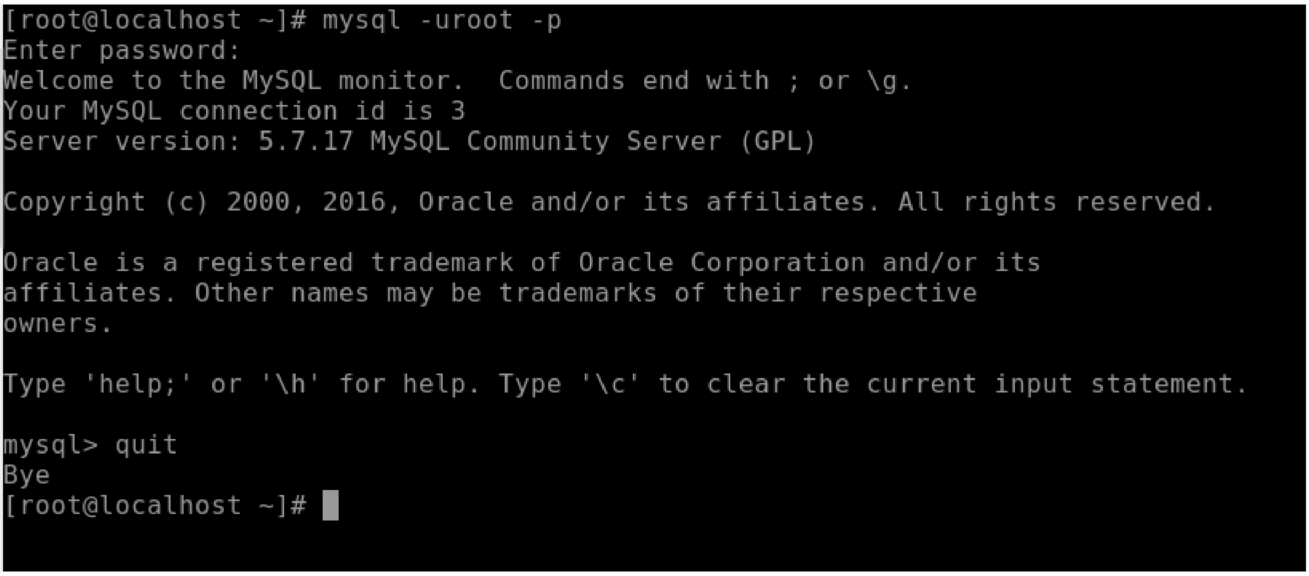
```
[root@localhost ~]# service mysqld start
Starting MySQL.Logging to '/usr/local/mysql/data/localhost.localdomain.err'.
 SUCCESS!
[root@localhost ~]# ps aux | grep mysqld
root      4127  0.0  0.1  11816  1616 pts/0    S    18:59   0:00 /bin/sh /usr/local/mys
ql/bin/mysqld_safe --datadir=/usr/local/mysql/data --pid-file=/usr/local/mysql/data/loc
alhost.localdomain.pid
mysql     4275  1.5 17.3 1117236 175948 pts/0  Sl   18:59   0:00 /usr/local/mysql/bin/m
ysqld --basedir=/usr/local/mysql --datadir=/usr/local/mysql/data --plugin-dir=/usr/loca
l/mysql/lib/plugin --user=mysql --log-error=/usr/local/mysql/data/localhost.localdomain
.err --pid-file=/usr/local/mysql/data/localhost.localdomain.pid
root      4312  0.0  0.0 112708   976 pts/0    R+   19:00   0:00 grep --color=auto mysq
ld
[root@localhost ~]#
```

8. 创建用户

执行如下命令创建一个新用户并设置root用户,root用户密码为root

输入 `mysql -uroot -p` ，执行后出现 `“Enter password: ”` 后直接按回车。

grant all on . to 'root'@'%' identified by 'root';flush privileges;



```
[root@localhost ~]# mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.7.17 MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> quit
Bye
[root@localhost ~]#
```

参考：

退出mysql: `quit`

启动mysql: `service mysqld start`

停止mysql: `service mysqld stop`

MySQL常用操作

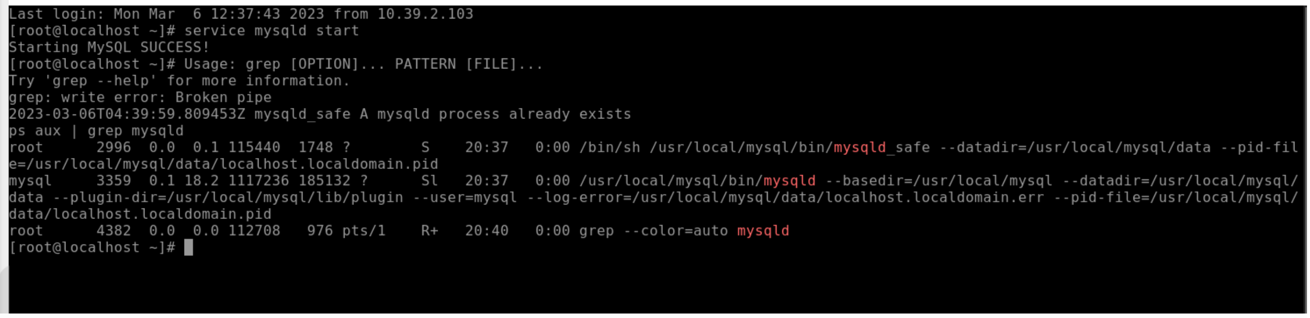
1. 登录mysql服务

执行

```
service mysqld start
```

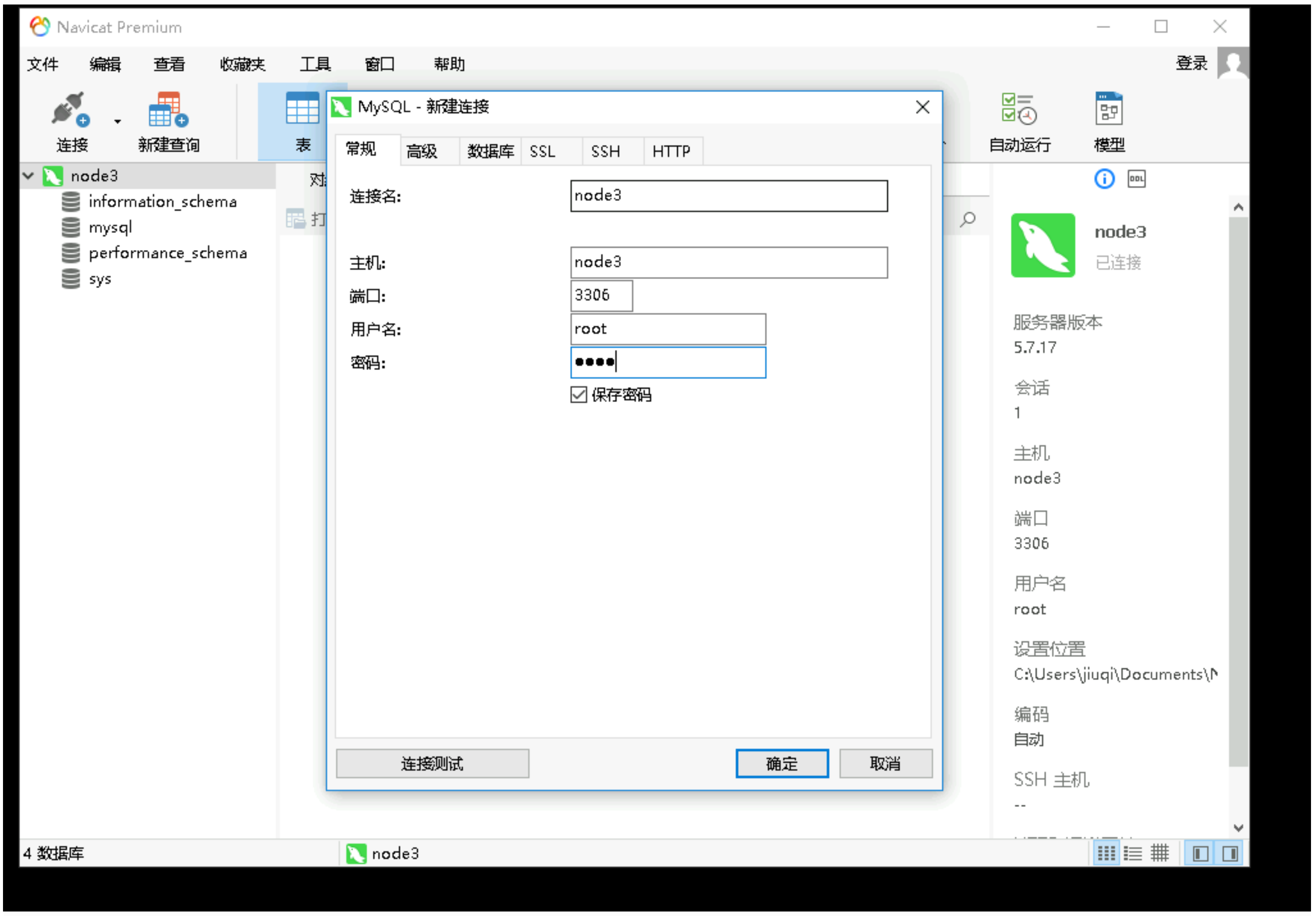
并确认是否启动成功

```
ps aux | grep mysqld
```



```
Last login: Mon Mar  6 12:37:43 2023 from 10.39.2.103
[root@localhost ~]# service mysqld start
Starting MySQL SUCCESS!
[root@localhost ~]# Usage: grep [OPTION]... PATTERN [FILE]...
Try 'grep --help' for more information.
grep: write error: Broken pipe
2023-03-06T04:39:59.809453Z mysqld_safe A mysqld process already exists
ps aux | grep mysqld
root      2996  0.0  0.1 115440  1748 ?        S    20:37   0:00 /bin/sh /usr/local/mysql/bin/mysqld_safe --datadir=/usr/local/mysql/data --pid-fil
e=/usr/local/mysql/data/localhost.localdomain.pid
mysql     3359  0.1 18.2 1117236 185132 ?      Sl   20:37   0:00 /usr/local/mysql/bin/mysqld --basedir=/usr/local/mysql --datadir=/usr/local/mysql/
data --plugin-dir=/usr/local/mysql/lib/plugin --user=mysql --log-error=/usr/local/mysql/data/localhost.localdomain.err --pid-file=/usr/local/mysql/
data/localhost.localdomain.pid
root      4382  0.0  0.0 112708   976 pts/1    R+   20:40   0:00 grep --color=auto mysqld
[root@localhost ~]# 
```

然后打开win10机器桌面的Navicat Premium 12软件，点击【连接】后，选择【MySQL】，在弹出窗口中输入连接名：node3，主机：node3，端口：3306，用户名：root，密码：root后，点击【连接测试】，确认连接成功后，点击【确定】。



node3
information_schema
mysql
performance_schema
sys

连接名: node3

主机: node3

端口: 3306

用户名: root

密码: ●●●●

☑ 保存密码

node3
已连接

服务器版本
5.7.17

会话
1

主机
node3

端口
3306

用户名
root

设置位置
C:\Users\jiuqi\Documents\N

编码
自动

SSH 主机
--

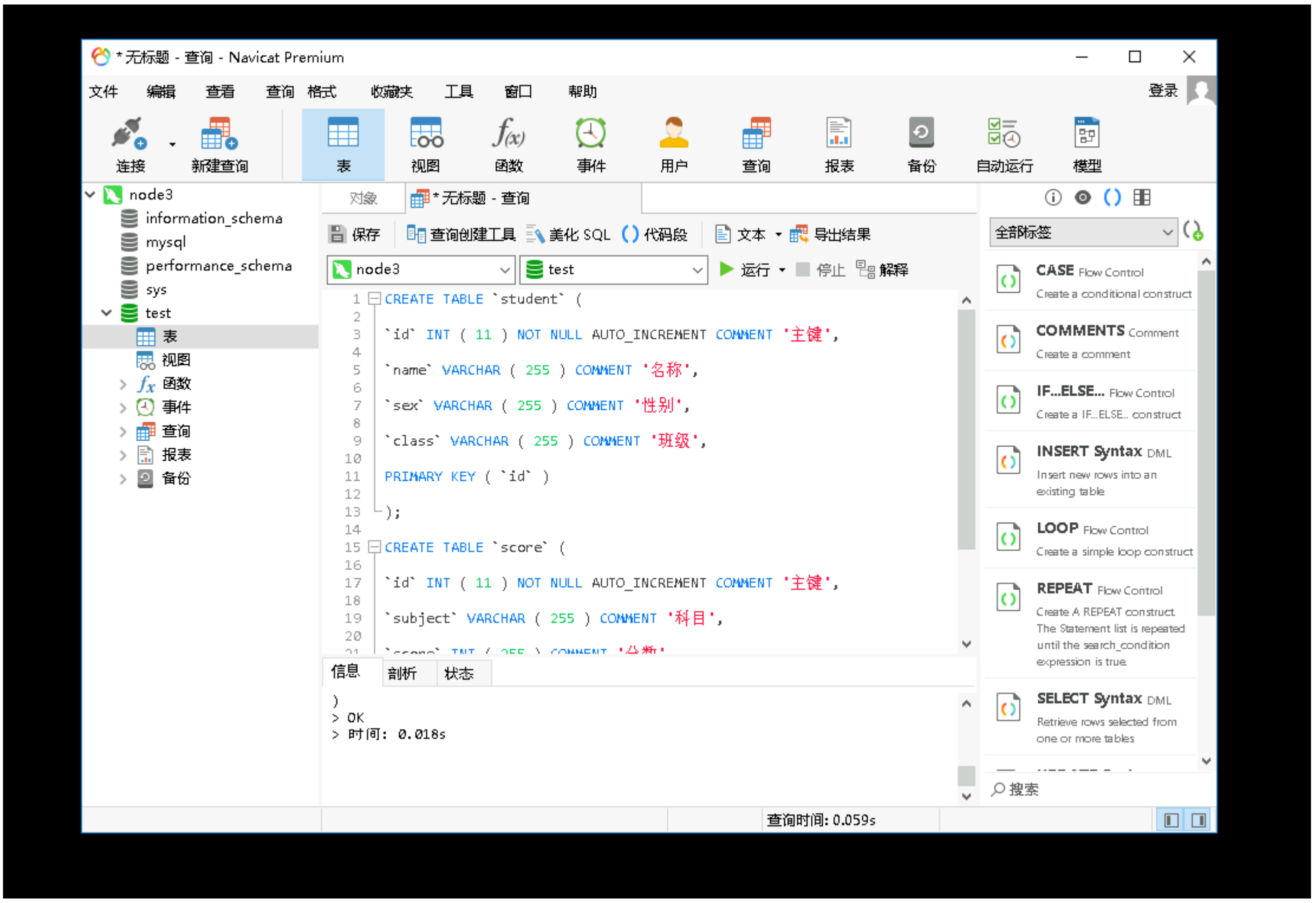
2. 创建数据库

新建查询->输入创建数据库语句

```
CREATE DATABASE test;
```

3. 创建表

选择新建的test数据库，输入创建学生表和成绩表的SQL语句，点击【运行】，创建表的SQL语句如下：


* 无标题 - 查询 - Navicat Premium
文件 编辑 查看 查询 格式 收藏夹 工具 窗口 帮助
登录
连接 新建查询 表 视图 函数 事件 用户 查询 报表 备份 自动运行 模型
node3
information_schema
mysql
performance_schema
sys
test
表
视图
函数
事件
查询
报表
备份
对象
* 无标题 - 查询
保存 查询创建工具 美化 SQL 代码段 文本 导出结果
node3
test
运行 停止 解释
全部标签
CASE Flow Control
Create a conditional construct
COMMENTS Comment
Create a comment
IF...ELSE... Flow Control
Create a IF...ELSE.. construct
INSERT Syntax DML
Insert new rows into an existing table
LOOP Flow Control
Create a simple loop construct
REPEAT Flow Control
Create A REPEAT construct. The Statement list is repeated until the search_condition expression is true.
SELECT Syntax DML
Retrieve rows selected from one or more tables
搜索
信息 剖析 状态
查询时间: 0.059s

```
CREATE TABLE `student` (

`id` INT ( 11 ) NOT NULL AUTO_INCREMENT COMMENT '主键',

`name` VARCHAR ( 255 ) COMMENT '名称',

`sex` VARCHAR ( 255 ) COMMENT '性别',

`class` VARCHAR ( 255 ) COMMENT '班级',

PRIMARY KEY ( `id` )

);

CREATE TABLE `score` (

`id` INT ( 11 ) NOT NULL AUTO_INCREMENT COMMENT '主键',

`subject` VARCHAR ( 255 ) COMMENT '科目',

`score` INT ( 255 ) COMMENT '分数',
```

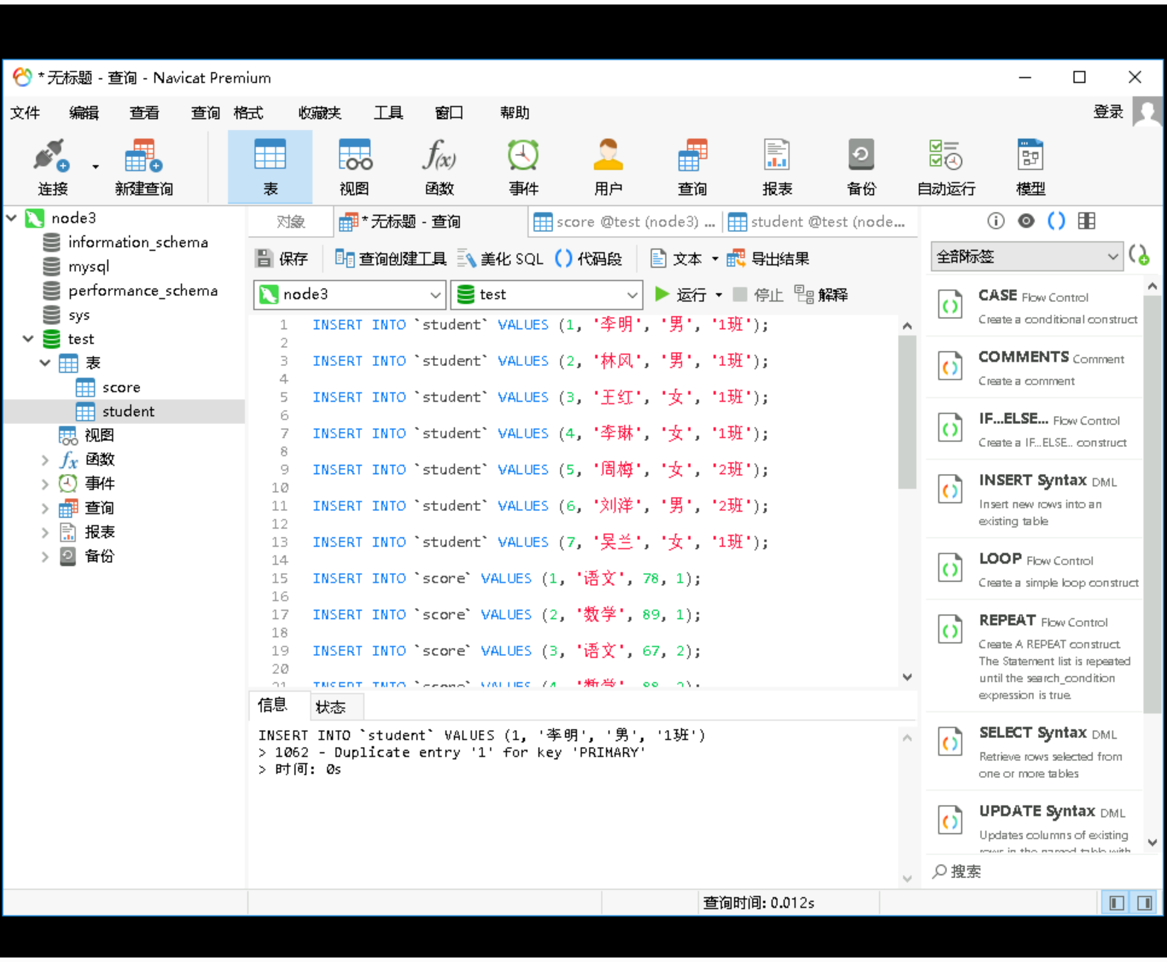
```
)
> OK
> 时间: 0.018s
```

```
CREATE TABLE `student` (

`id` INT ( 11 ) NOT NULL AUTO_INCREMENT COMMENT '主键',

`name` VARCHAR ( 255 ) COMMENT '名称',

`sex` VARCHAR ( 255 ) COMMENT '性别',

`class` VARCHAR ( 255 ) COMMENT '班级',

PRIMARY KEY ( `id` )

);

CREATE TABLE `score` (

`id` INT ( 11 ) NOT NULL AUTO_INCREMENT COMMENT '主键',

`subject` VARCHAR ( 255 ) COMMENT '科目',

```

```sql

`score` INT ( 255 ) COMMENT '分数',

`sid` INT ( 11 ) COMMENT '学生表主键',

PRIMARY KEY ( `id` )

);
```

4. 插入数据

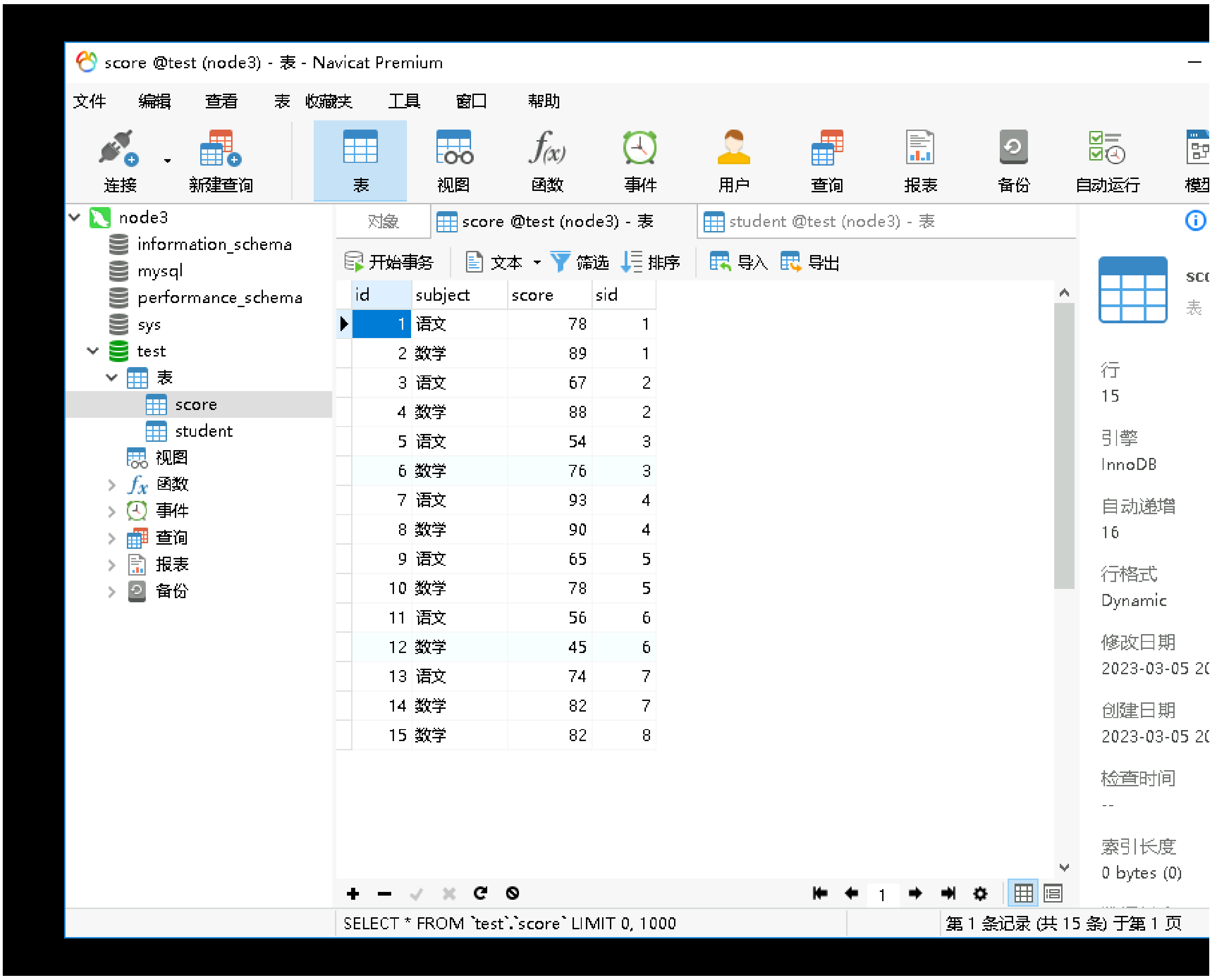
输入插入学生表和成绩表的SQL语句



```sql
INSERT INTO `student` VALUES (1, '李明', '男', '1班');

INSERT INTO `student` VALUES (2, '林风', '男', '1班');

INSERT INTO `student` VALUES (3, '王红', '女', '1班');

INSERT INTO `student` VALUES (4, '李琳', '女', '1班');

INSERT INTO `student` VALUES (5, '周梅', '女', '2班');

INSERT INTO `student` VALUES (6, '刘洋', '男', '2班');

INSERT INTO `student` VALUES (7, '吴兰', '女', '1班');

INSERT INTO `score` VALUES (1, '语文', 78, 1);

INSERT INTO `score` VALUES (2, '数学', 89, 1);

INSERT INTO `score` VALUES (3, '语文', 67, 2);

INSERT INTO `score` VALUES (4, '数学', 88, 2);
```

```
INSERT INTO `student` VALUES (1, '李明', '男', '1班')
> 1062 - Duplicate entry '1' for key 'PRIMARY'
> 时间: 0s
```

```sql
INSERT INTO `student` VALUES (1, '李明', '男', '1班');

INSERT INTO `student` VALUES (2, '林风', '男', '1班');

INSERT INTO `student` VALUES (3, '王红', '女', '1班');

INSERT INTO `student` VALUES (4, '李琳', '女', '1班');

INSERT INTO `student` VALUES (5, '周梅', '女', '2班');

INSERT INTO `student` VALUES (6, '刘洋', '男', '2班');

INSERT INTO `student` VALUES (7, '吴兰', '女', '1班');

INSERT INTO `score` VALUES (1, '语文', 78, 1);

INSERT INTO `score` VALUES (2, '数学', 89, 1);

INSERT INTO `score` VALUES (3, '语文', 67, 2);

INSERT INTO `score` VALUES (4, '数学', 88, 2);

INSERT INTO `score` VALUES (5, '语文', 54, 3);

INSERT INTO `score` VALUES (6, '数学', 76, 3);

INSERT INTO `score` VALUES (7, '语文', 93, 4);

INSERT INTO `score` VALUES (8, '数学', 90, 4);

INSERT INTO `score` VALUES (9, '语文', 65, 5);

INSERT INTO `score` VALUES (10, '数学', 78, 5);

INSERT INTO `score` VALUES (11, '语文', 56, 6);

INSERT INTO `score` VALUES (12, '数学', 45, 6);

INSERT INTO `score` VALUES (13, '语文', 74, 7);

INSERT INTO `score` VALUES (14, '数学', 82, 7);

INSERT INTO `score` VALUES (15, '数学', 82, 8);
```



id	subject	score	sid
1	语文	78	1
2	数学	89	1
3	语文	67	2
4	数学	88	2
5	语文	54	3
6	数学	76	3
7	语文	93	4
8	数学	90	4
9	语文	65	5
10	数学	78	5
11	语文	56	6
12	数学	45	6
13	语文	74	7
14	数学	82	7
15	数学	82	8

行
15

引擎
InnoDB

自动递增
16

行格式
Dynamic

修改日期
2023-03-05 20

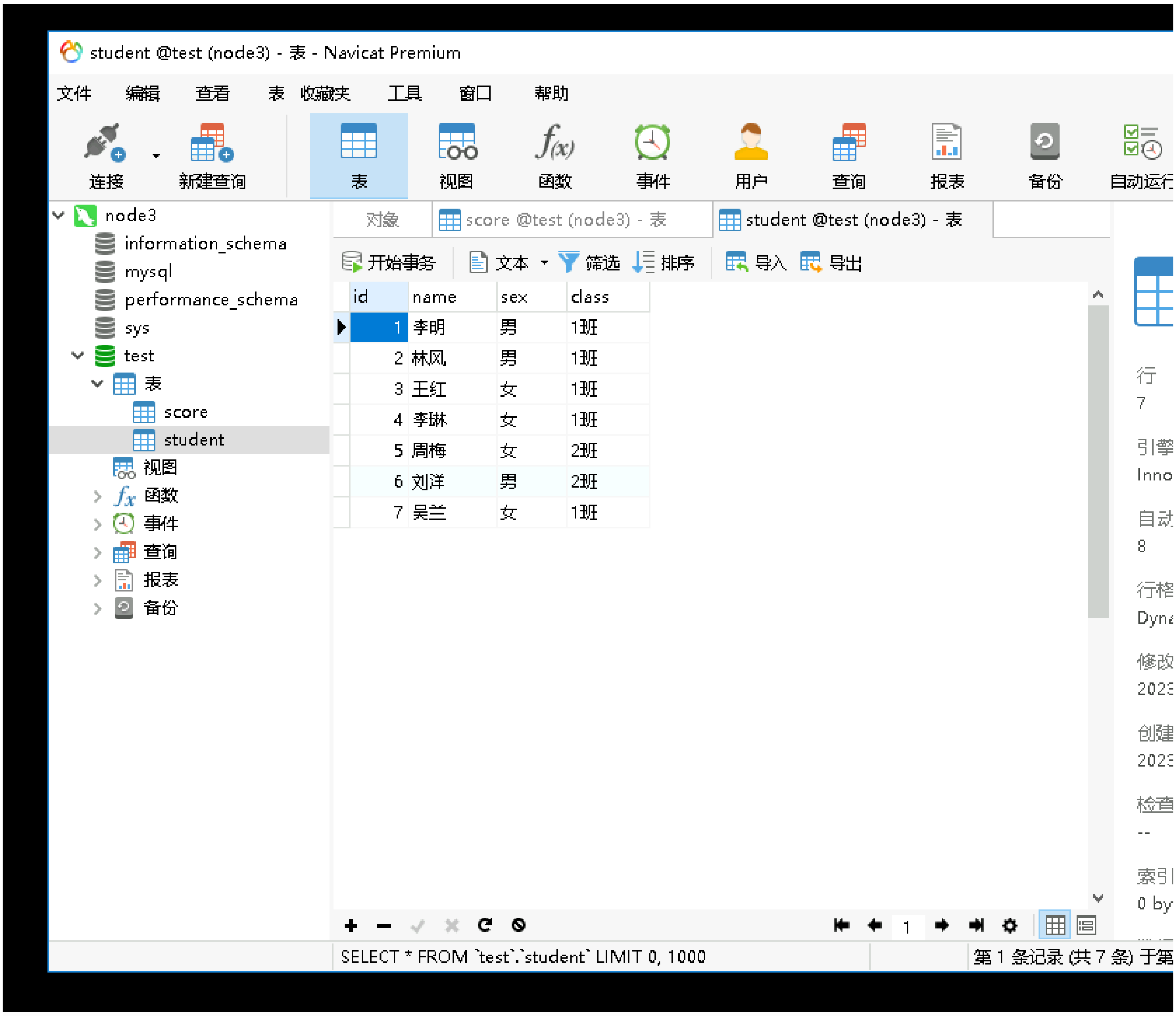
创建日期
2023-03-05 20

检查时间
--

索引长度
0 bytes (0)

```
SELECT * FROM `test`.`score` LIMIT 0, 1000
```

第 1 条记录 (共 15 条) 于第 1 页


student @test (node3) - 表 - Navicat Premium
文件 编辑 查看 表 收藏夹 工具 窗口 帮助
连接 新建查询 表 视图 函数 事件 用户 查询 报表 备份 自动运行
node3
information_schema
mysql
performance_schema
sys
test
表
score
student
视图
函数
事件
查询
报表
备份
对象
score @test (node3) - 表
student @test (node3) - 表
开始事务 文本 筛选 排序 导入 导出
行
7
引擎
Inno
自动
8
行格
Dyna
修改
2023
创建
2023
检查
--
索引
0 by

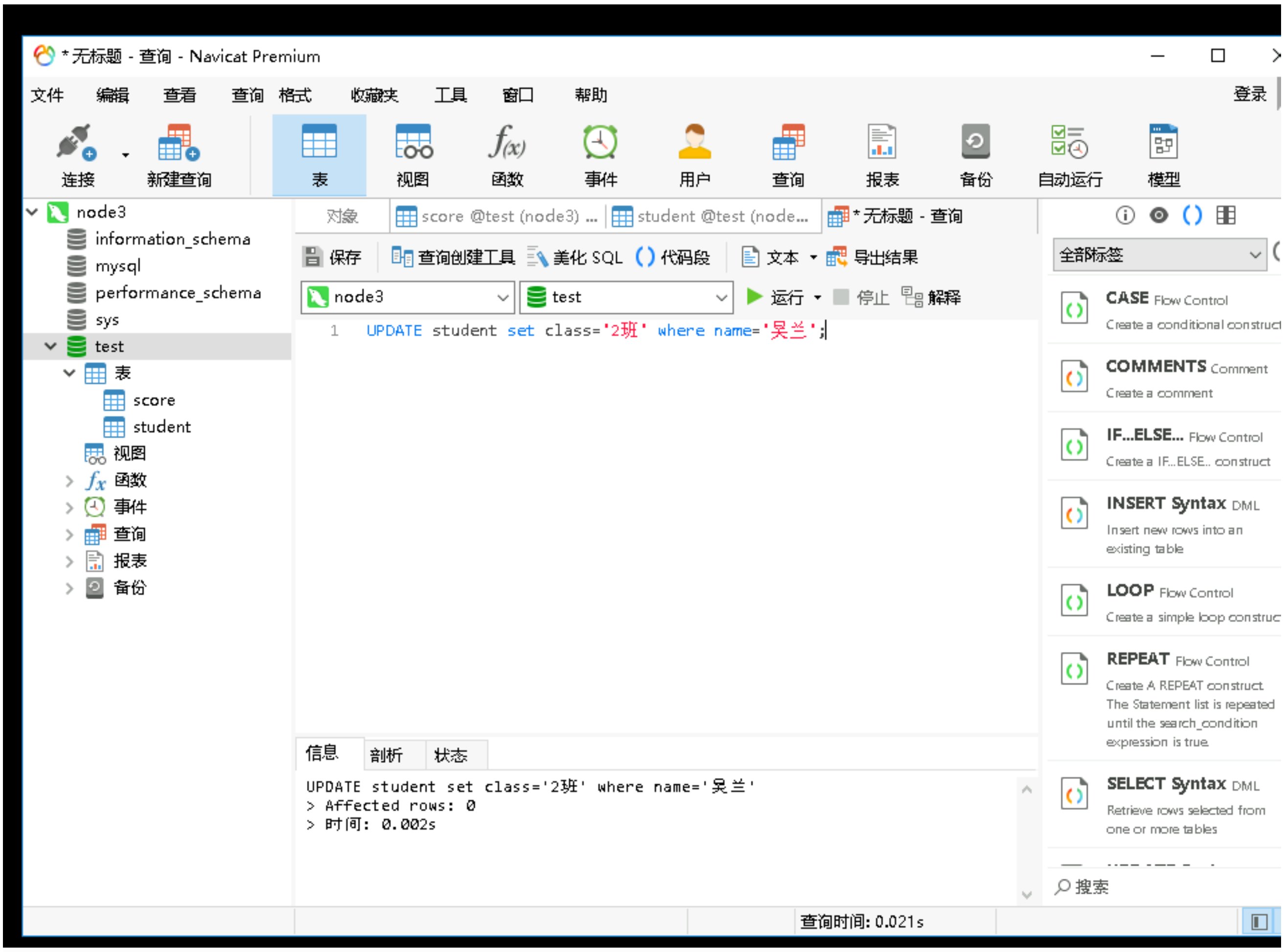
id	name	sex	class
1	李明	男	1班
2	林风	男	1班
3	王红	女	1班
4	李琳	女	1班
5	周梅	女	2班
6	刘洋	男	2班
7	吴兰	女	1班

```
SELECT * FROM `test`.`student` LIMIT 0, 1000
```

第 1 条记录 (共 7 条) 于第

5. 更新数据

修改学生“吴兰”的班级为“2班”，更新SQL语句如下：


* 无标题 - 查询 - Navicat Premium
文件 编辑 查看 查询 格式 收藏夹 工具 窗口 帮助 登录
连接 新建查询 表 视图 函数 事件 用户 查询 报表 备份 自动运行 模型
node3
information_schema
mysql
performance_schema
sys
test
表
score
student
视图
函数
事件
查询
报表
备份
对象
score @test (node3) ...
student @test (node...
* 无标题 - 查询
保存 查询创建工具 美化 SQL 代码段 文本 导出结果
node3
test
运行 停止 解释
信息 剖析 状态
全部标签
CASE Flow Control
Create a conditional construct
COMMENTS Comment
Create a comment
IF...ELSE... Flow Control
Create a IF...ELSE... construct
INSERT Syntax DML
Insert new rows into an existing table
LOOP Flow Control
Create a simple loop construc
REPEAT Flow Control
Create A REPEAT construct The Statement list is repeated until the search_condition expression is true
SELECT Syntax DML
Retrieve rows selected from one or more tables
搜索

```
UPDATE student set class='2班' where name='吴兰';
```

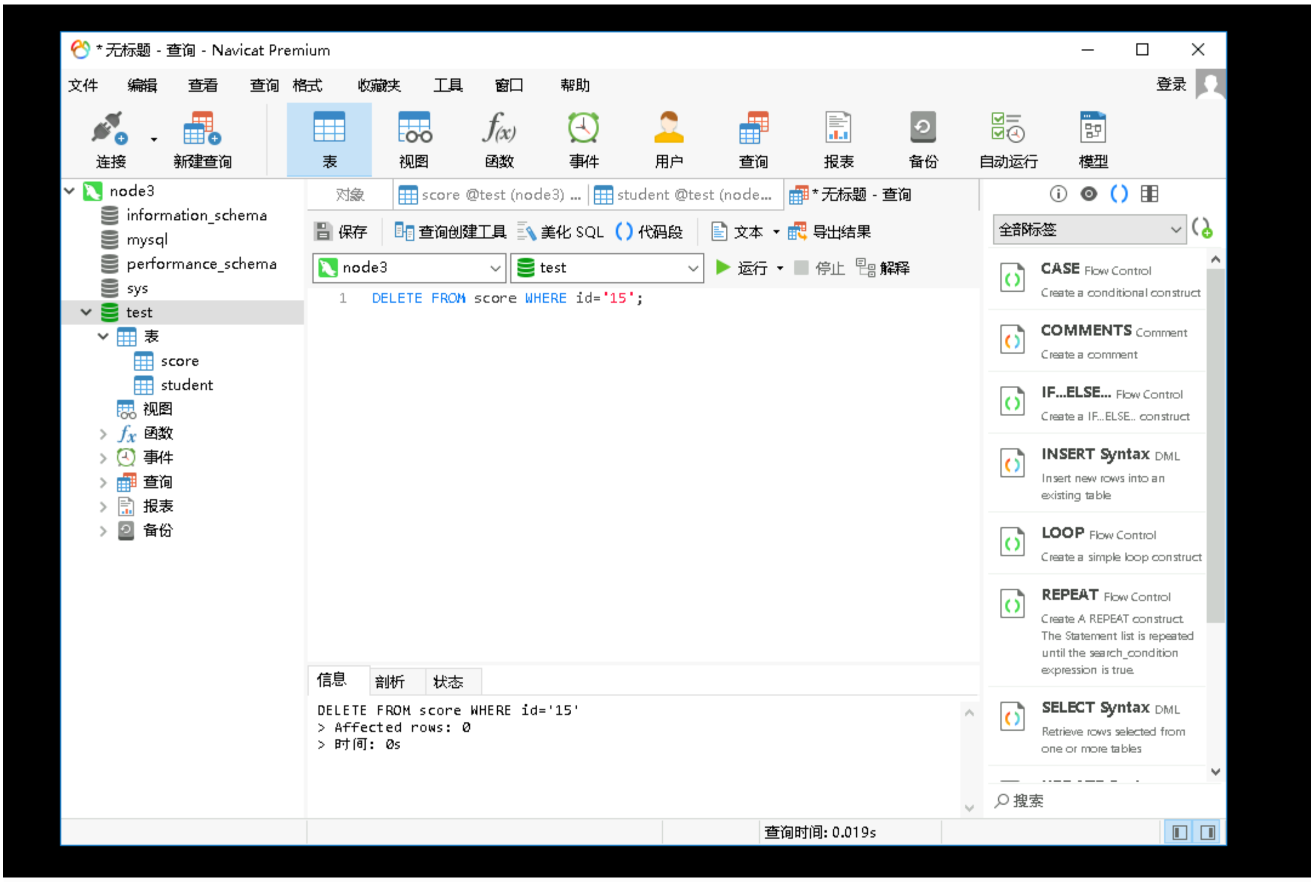
```
UPDATE student set class='2班' where name='吴兰'
> Affected rows: 0
> 时间: 0.002s
```

查询时间: 0.021s

```
UPDATE student set class='2班' where name='吴兰';
```

6. 删除数据

删除成绩表中主键为“15”的数据，删除SQL语句如下：


* 无标题 - 查询 - Navicat Premium
文件 编辑 查看 查询 格式 收藏夹 工具 窗口 帮助 登录
连接 新建查询 表 视图 函数 事件 用户 查询 报表 备份 自动运行 模型
node3
information_schema
mysql
performance_schema
sys
test
表
score
student
视图
函数
事件
查询
报表
备份
对象 score @test (node3) ... student @test (node... * 无标题 - 查询
保存 查询创建工具 美化 SQL 代码段 文本 导出结果
node3 test 运行 停止 解释
信息 剖析 状态
查询时间: 0.019s
全部标签
CASE Flow Control
Create a conditional construct
COMMENTS Comment
Create a comment
IF...ELSE... Flow Control
Create a IF...ELSE.. construct
INSERT Syntax DML
Insert new rows into an existing table
LOOP Flow Control
Create a simple loop construct
REPEAT Flow Control
Create A REPEAT construct The Statement list is repeated until the search_condition expression is true
SELECT Syntax DML
Retrieve rows selected from one or more tables
搜索

```
DELETE FROM score WHERE id='15';
```

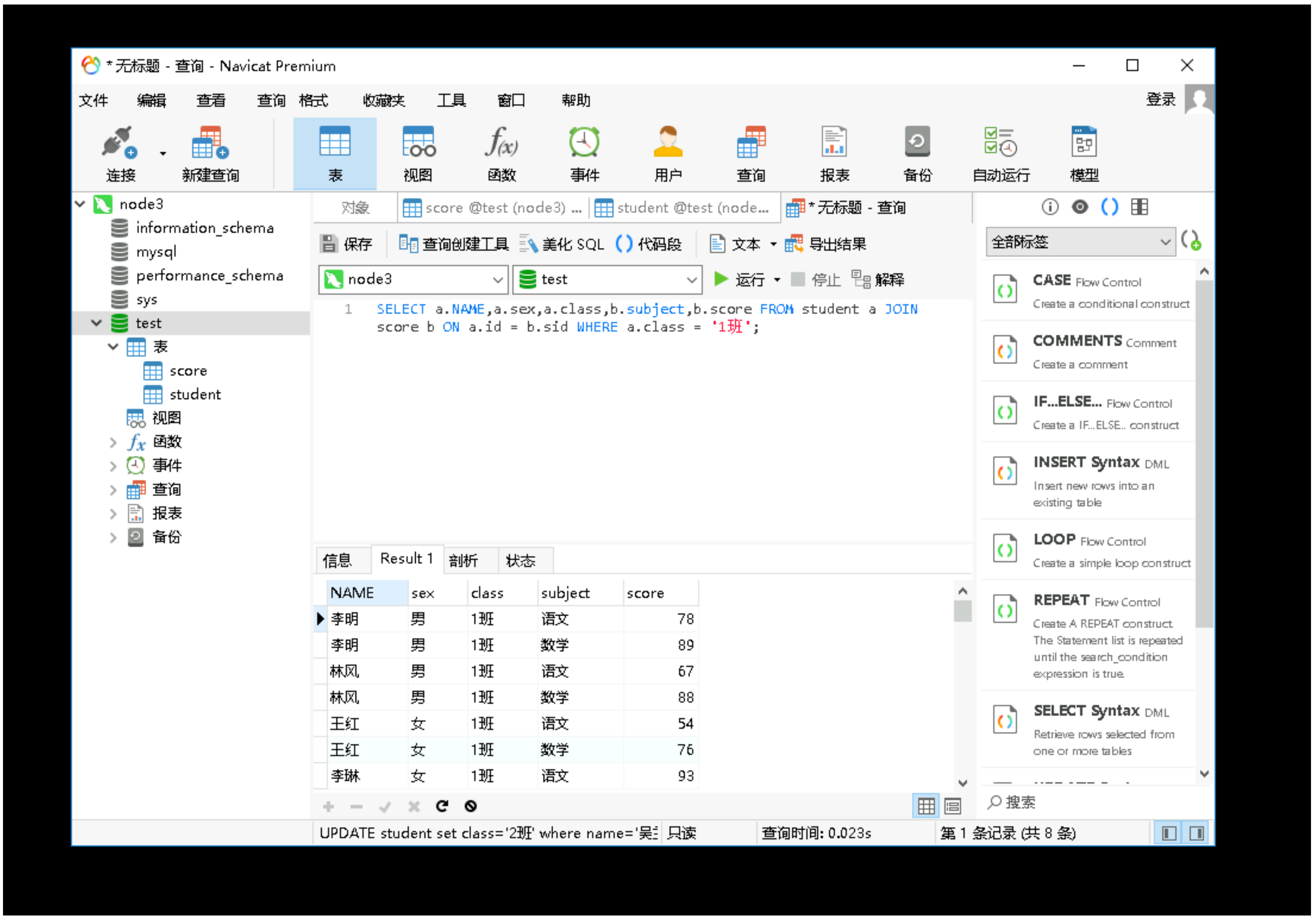
```
DELETE FROM score WHERE id='15'
> Affected rows: 0
> 时间: 0s
```

```
DELETE FROM score WHERE id='15';
```

7. 查询表数据

查询1班中每个学生的基本信息和每科的成绩，查询SQL如下：

```
SELECT a.NAME,a.sex,a.class,b.subject,b.score FROM student a JOIN
score b ON a.id = b.sid WHERE a.class = '1班';
```



```sql
SELECT a.NAME,a.sex,a.class,b.subject,b.score FROM student a JOIN
score b ON a.id = b.sid WHERE a.class = '1班';
```

NAME	sex	class	subject	score
李明	男	1班	语文	78
李明	男	1班	数学	89
林风	男	1班	语文	67
林风	男	1班	数学	88
王红	女	1班	语文	54
王红	女	1班	数学	76
李琳	女	1班	语文	93

UPDATE student set class='2班' where name='吴 只读 查询时间: 0.023s 第 1 条记录 (共 8 条)