0406基础认识

```
(base) [yufc@VM-12-12-centos:~/Core]$ vim /etc/my.cnf
YouCompleteMe unavailable: requires Vim compiled with Python (2.7.1+ or 3.4+) support.
UltiSnips requires py >= 2.7 or py3
Error: LeaderF requires vim compiled with +python or +python3
Press ENTER or type command to continue
(base) [yufc@VM-12-12-centos:~/Core]$
```

```
1: /e/my.cnf
 1 # For advice on how to change settings please see
 2 # http://dev.mysql.com/doc/refman/5.7/en/server-configuration-defaults.html
 3
 4 [mysqld]
 5 #
 6 # Remove leading # and set to the amount of RAM for the most important data
 7 # cache in MySQL. Start at 70% of total RAM for dedicated server, else 10%.
 8 # innodb_buffer_pool_size = 128M
 9 #
10 # Remove leading # to turn on a very important data integrity option: loggi
11 # changes to the binary log between backups.
12 # log_bin
13 #
14 # Remove leading # to set options mainly useful for reporting servers.
15 # The server defaults are faster for transactions and fast SELECTs.
16 # Adjust sizes as needed, experiment to find the optimal values.
17 # join_buffer_size = 128M
18 # sort_buffer_size = 2M
19 # read_rnd_buffer_size = 2M
20 datadir=/var/lib/mysql
21 socket=/var/lib/mysql/mysql.sock
22
23 # Disabling symbolic-links is recommended to prevent assorted security risk
24 symbolic-links=0
25
26 log-error=/var/log/mysqld.log
27 pid-file=/var/run/mysqld/mysqld.pid
NORMAL  /etc/my.cnf ↔                                                    co
```

数据都放在这个
datadir里面

```
(base) [yufc@VM-12-12-centos:/var/lib/mysql]$ sudo ls
[sudo] password for yufc:
auto.cnf    client-cert.pem  ibdata1      ibtmp1      mysql.sock.lock     public_key.pem   sys
ca-key.pem  client-key.pem   ib_logfile0  mysql       performance_schema  server-cert.pem
ca.pem      ib_buffer_pool   ib_logfile1  mysql.sock  private_key.pem     server-key.pem
(base) [yufc@VM-12-12-centos:/var/lib/mysql]$
```

以特定的格式保存好的文件，
我们叫做数据库

数据库的本质：是对文件的内容提供更基本的内容操作，不用程序员自己处理

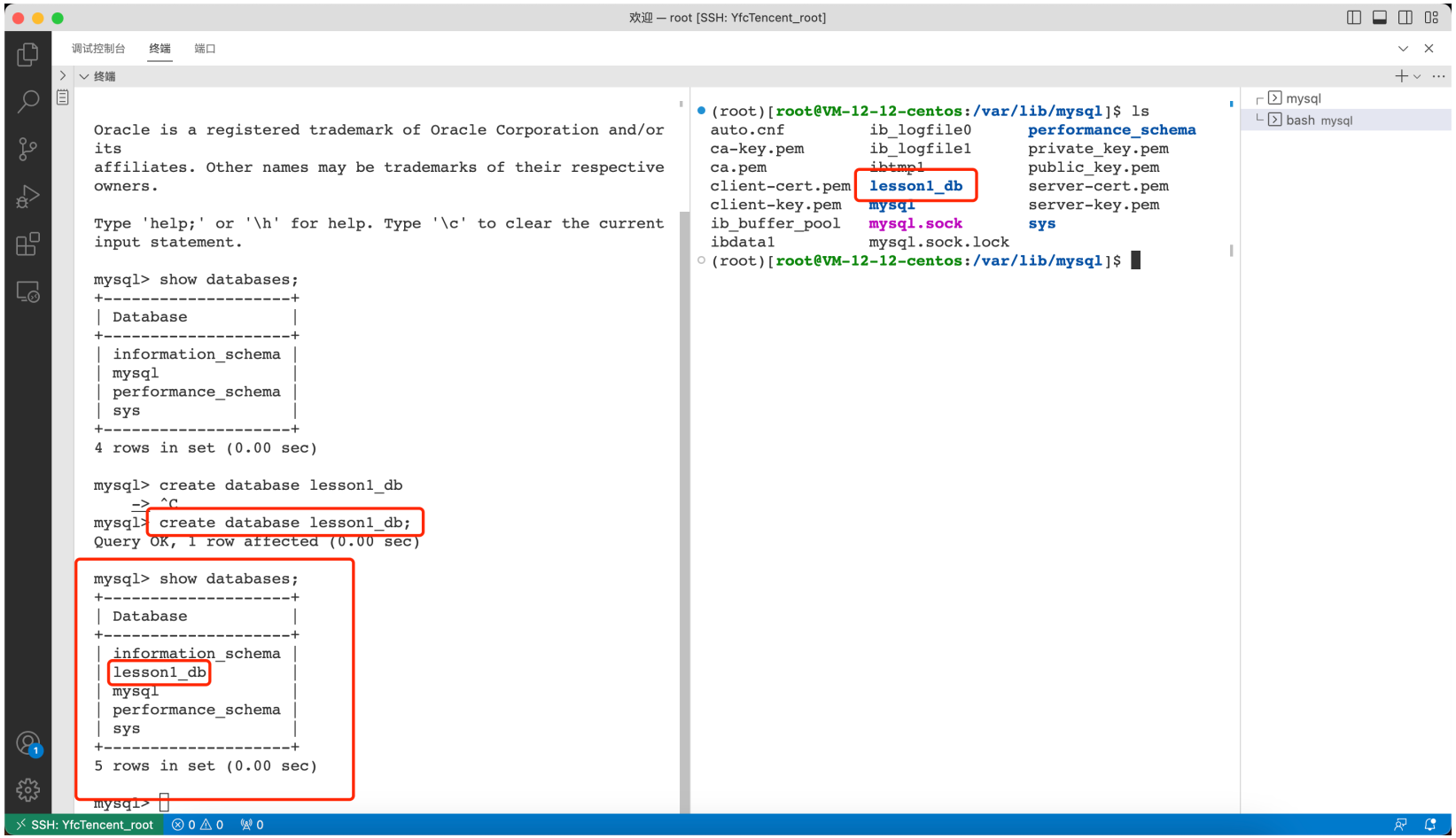
```
(root)[root@VM-12-12-centos:~]$ netstat -nltp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      1801/sshd
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      1619/master
tcp        0      0 127.0.0.1:43424         0.0.0.0:*               LISTEN      24668/node
tcp6       0      0 :::22                   :::*                    LISTEN      1801/sshd
tcp6       0      0 ::1:25                  :::*                    LISTEN      1619/master
tcp6       0      0 :::3306                 :::*                    LISTEN      10458/mysqld
(root)[root@VM-12-12-centos:~]$
```

```
tcp6       0      0 :::3306                 :::*                    LISTEN      10458/mysqld
(root)[root@VM-12-12-centos:~]$ mysql -h 127.0.0.1 -P 3306 -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.41 MySQL Community Server (GPL)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

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>
```

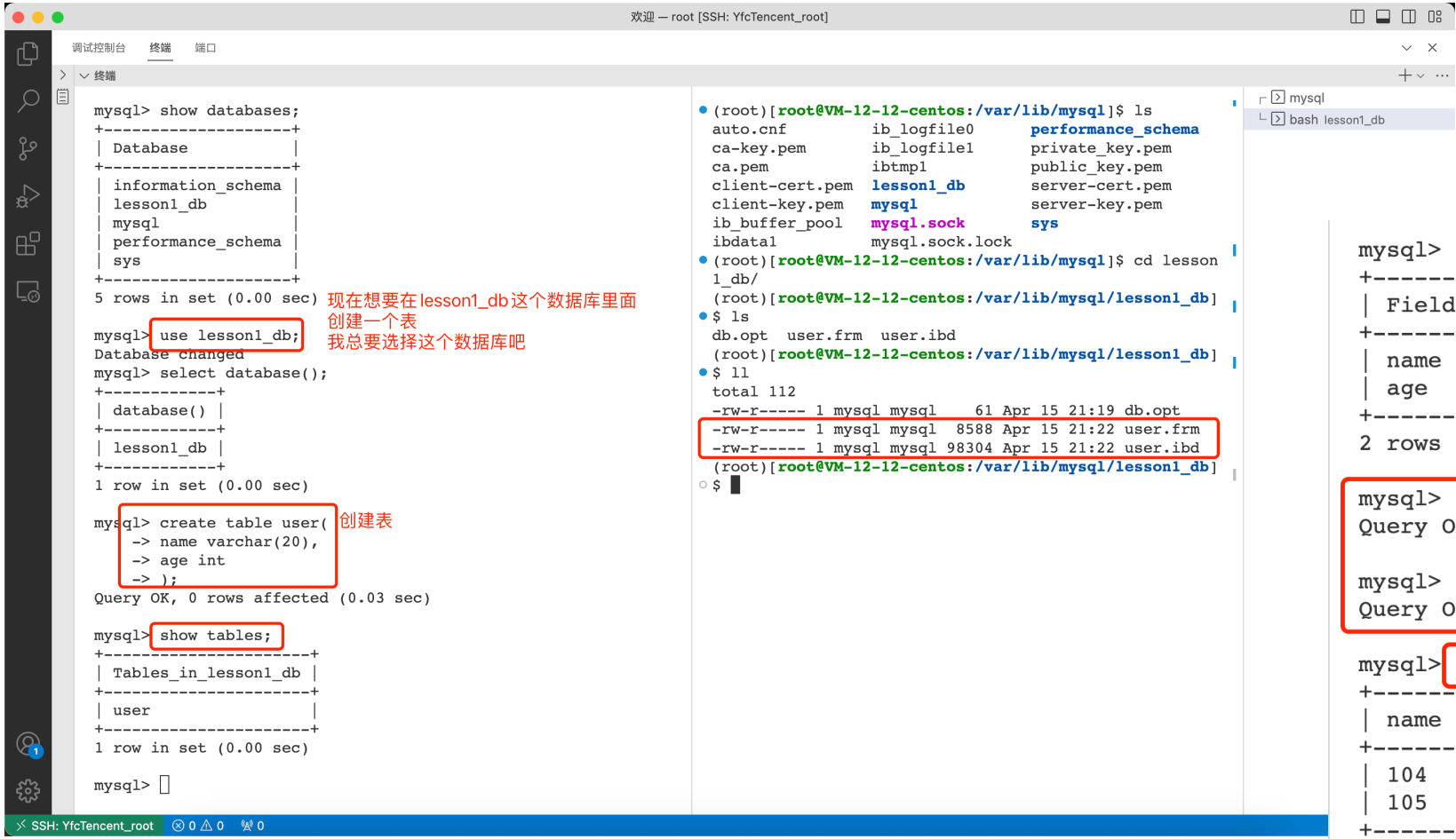
这里后面
可以跟密码
也可以回车再输入
密码

创建一个数据库本质是在 linux ß上做什么？

建表本质在 linux 上是在做什么？



在mysql中，建立一个数据库
的本质是在linux下建立了一个目录！



所以创建表的本质
就是在数据库文件夹里面
创建一些文件！

```
mysql> desc user;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| name  | varchar(20) | YES  |     | NULL    |       |
| age   | int(11)     | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)

mysql> insert into user (name, age) values('104',14);
Query OK, 1 row affected (0.01 sec)

mysql> insert into user (name, age) values('105',16);
Query OK, 1 row affected (0.01 sec)

mysql> select * from user;
+------+------+
| name | age  |
+------+------+
| 104  |   14 |
| 105  |   16 |
+------+------+
2 rows in set (0.00 sec)

mysql>
```

往表里面插点东西

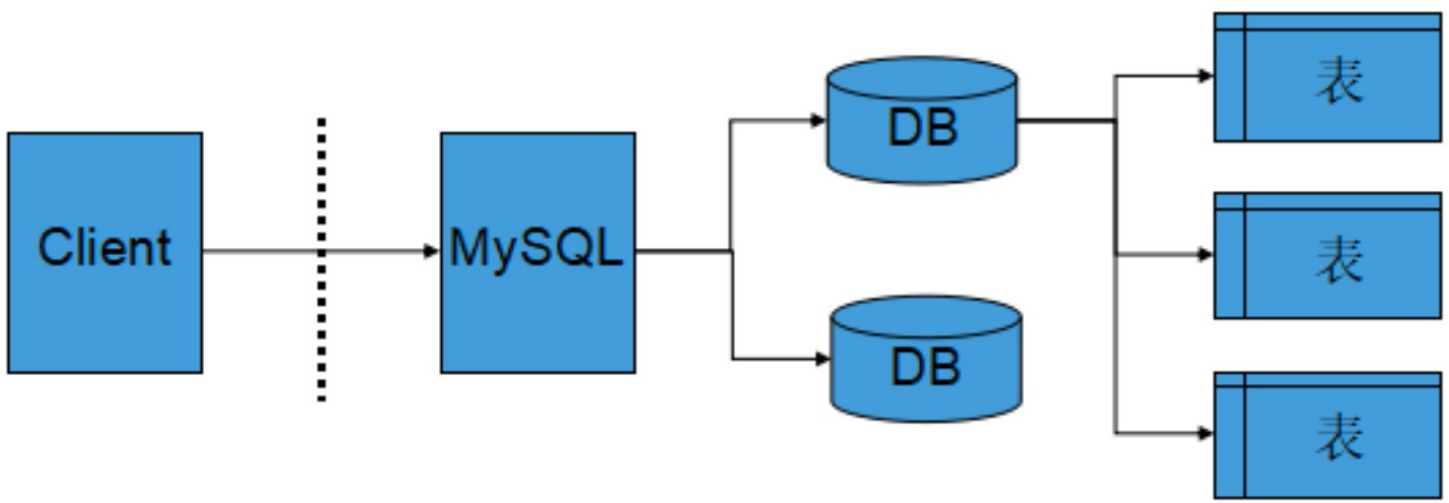
```
mysql> desc user;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| name  | varchar(20) | YES  |     | NULL    |       |
| age   | int(11)     | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)

mysql> insert into user (name, age) values('104',14);
Query OK, 1 row affected (0.01 sec)

mysql> insert into user (name, age) values('105',16);
Query OK, 1 row affected (0.01 sec)

mysql> select * from user;
+------+------+
| name | age  |
+------+------+
| 104  |   14 |
| 105  |   16 |
+------+------+
2 rows in set (0.00 sec)
```

```
(root)[root@VM-12-12-centos:/var/lib/mysql/lesson1_db]
$ ll
total 112
-rw-r----- 1 mysql mysql    61 Apr 15 21:19 db.opt
-rw-r----- 1 mysql mysql  8588 Apr 15 21:22 user.frm
-rw-r----- 1 mysql mysql 98304 Apr 15 21:22 user.ibd
(root)[root@VM-12-12-centos:/var/lib/mysql/lesson1_db]
$ ll
total 112
-rw-r----- 1 mysql mysql    61 Apr 15 21:19 db.opt
-rw-r----- 1 mysql mysql  8588 Apr 15 21:22 user.frm
-rw-r----- 1 mysql mysql 98304 Apr 15 21:26 user.ibd
(root)[root@VM-12-12-centos:/var/lib/mysql/lesson1_db]
$
```

这个不用说也能猜到，mysql肯定有自己的缓冲区
不然每次插入都进行IO，这样效率是很低的

不是说数据库的插入就是往文件里面写入吗？

为什么没有变化？



Mysql的生态中
也会提供第三方库

让语言能直接访问mysql

```
mysql> select * from student;
+------+------+--------+
|  id  | name | gender |
+------+------+--------+
|    1 | 张三 | 男     |
|    2 | 李四 | 女     |
|    3 | 王五 | 男     |
|    1 | 张三 | 男     |
|    2 | 李四 | 女     |
|    3 | 王五 | 男     |
+------+------+--------+
6 rows in set (0.00 sec)
```

列

行