Linux 运维必会 MySQL 30 道基础命令

老男孩老师教学与培训核心思想：重目标、重思路、重方法、重实践、重习惯、重总结。

联系方式：

意见投诉信箱:oldboy521@gmail.com 网站运维交流群：114580181 246054962 45039636 37081784

==========

老男孩 linux 实训联系方式

咨询：QQ: 70271111 41117397 41117483 80042789 电话：18600338340 18911718229 网站地址：http://www.etiantian.org http://bbs.etiantian.org blog http://oldboy.blog.51cto.com

文档信息：

文档版本：Version 1.6

文档作用：课后总结

修改记录：2015-12-21

系统环境：CentOS release 6.7 (Final)

技术交流：http://yulianhui.blog.51cto.com/

目　录

数据库的重要性是所有技术里最核心最需要掌握的(理解原理，并且被面试时能清晰的表达出来)，直接决定运维人员薪水的高低！

所有题都要给出专业的解答方案，不是很水的那种泛泛的解答。

1. 登陆数据库。

```
mysql -uroot -poldboy123 -S /data/3306/mysql.sock
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 5.5.32-log Source distribution
```

```
Copyright (c) 2000, 2013, 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>
```

2. 查看数据库版本及当前登录用户是什么。

```
#查看数据库版本
mysql> select version();
+------------+
| version()  |
+------------+
| 5.5.32-log |
+------------+
1 row in set (0.00 sec)
#查看当前登录用户
mysql> select user();
+----------------+
| user()         |
```

```
+----------------+
| root@localhost |
+----------------+
1 row in set (0.00 sec)
```

3.创建 GBK 字符集的数据库 oldboy，并查看已建库的完整语句。

```
mysql> create database oldboy character set gbk collate gbk_chinese_ci;
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| oldboy             |
| performance_schema |
+--------------------+
4 rows in set (0.00 sec)

mysql>
#查看已建库的完整语句
mysql> show create database oldboy\G
```

```
*************************** 1. row ***************************
           Database: oldboy
Create Database: CREATE DATABASE `oldboy` /*!40100 DEFAULT CHARACTER SET gbk */
1 row in set (0.00 sec)
```

4.创建用户 oldboy，使之可以管理数据库 oldboy。

```
mysql> grant all on oldboy.* to oldboy@'localhost' identified by '123';
Query OK, 0 rows affected (0.00 sec)
```

5.查看创建的用户 oldboy 拥有哪些权限。

```
mysql> show grants for oldboy@'localhost'\G
*************************** 1. row ***************************
Grants for oldboy@localhost: GRANT USAGE ON *.* TO 'oldboy'@'localhost' IDENTIFIED BY PASSWORD '*23AE809DDACAF96AF0FD78ED04B6A265E05AA257'
*************************** 2. row ***************************
Grants for oldboy@localhost: GRANT ALL PRIVILEGES ON `oldboy`.* TO 'oldboy'@'localhost'
2 rows in set (0.00 sec)
```

5.查看当前数据库里有哪些用户。

```
mysql> select user,host from mysql.user;
+--------+-----------+
| user   | host      |
+--------+-----------+
| rep    | 10.0.0.%  |
| root   | 127.0.0.1 |
| oldboy | localhost |
| root   | localhost |
+--------+-----------+
4 rows in set (0.00 sec)
```

6. 进入 oldboy 数据库

```
mysql> use oldboy;
Database changed
mysql> select database();
+------------+
| database() |
+------------+
| oldboy     |
+------------+
1 row in set (0.00 sec)
```

8.创建一 innodb 引擎字符集为 GBK 表 test，字段为 id 和 namevarchar(16)，查看建表结构及 SQL 语句。

```
mysql> create table test(
    -> id int(4),
    -> name varchar(16)
    -> ) ENGINE=innodb default charset=gbk;
Query OK, 0 rows affected (0.01 sec)
mysql> show tables;
+-----------------+
| Tables_in_oldboy |
+-----------------+
| test             |
+-----------------+
1 row in set (0.00 sec)
#查看表结构：方法一
mysql> desc test;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| id    | int(4)      | YES  |     | NULL    |       |
| name  | varchar(16) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
#查看表结构：方法二
mysql> show columns from test;
```

```
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| id    | int(4)      | YES  |     | NULL    |       |
| name  | varchar(16) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
#查看表结构 sql 语句
mysql> show create table test\G
*************************** 1. row ***************************
       Table: test
Create Table: CREATE TABLE `test` (
  `id` int(4) DEFAULT NULL,
  `name` varchar(16) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=gbk
1 row in set (0.00 sec)
```

9. 插入一条数据 1,oldboy

```
mysql> insert into test values(1,'oldboy');
Query OK, 1 row affected (0.00 sec)

mysql> select * from test;
+------+--------+
```

```
| id     | name     |
+------+--------+
|      1 | oldboy |
+------+--------+
1 row in set (0.00 sec)
```

10.批量插入数据 2,老男孩，3,etiantian。ç要求中文不能乱码。

```
mysql> insert into test values(2,'老男孩'),(3,'etiantian');
Query OK, 2 rows affected (0.01 sec)
Records: 2    Duplicates: 0    Warnings: 0

mysql> select * from test;
+------+-----------+
| id     | name     |
+------+-----------+
|      1 | oldboy    |
|      2 | 老男孩  |
|      3 | etiantian |
+------+-----------+
3 rows in set (0.00 sec)
```

11. 查询插入的所有记录，查询名字为 oldboy 的记录。查询 id 大于 1 的记录。

（1）查询插入的所有记录

```
mysql> select * from test;
+------+-----------+
| id   | name      |
+------+-----------+
|    1 | oldboy    |
|    2 | 老男孩    |
|    3 | etiantian |
+------+-----------+
3 rows in set (0.00 sec)
```

（2）查询名字为 oldboy 的记录

```
mysql> select * from test where name='oldboy';
+------+--------+
| id   | name   |
+------+--------+
|    1 | oldboy |
+------+--------+
1 row in set (0.01 sec)
```

（3）查询 id 大于 1 的记录

```
mysql> select * from test where id>1;
+------+-----------+
| id   | name      |
```

```
+------+-----------+
|      2 | 老男孩 |
|      3 | etiantian |
+------+-----------+
2 rows in set (0.00 sec)
```

12. 把数据 id 等于 1 的名字 oldboy 更改为 oldgirl。

```
mysql> update test set name='oldgirl' where id=1;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from test;
+------+-----------+
| id   | name      |
+------+-----------+
|      1 | oldgirl  |
|      2 | 老男孩 |
|      3 | etiantian |
+------+-----------+
3 rows in set (0.00 sec)
```

13. 在字段 name 前插入 age 字段，类型 tinyint(2)。

```
mysql> alter table test add age tinyint(2) after id;
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> desc test;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| id    | int(4)      | YES  |     | NULL    |       |
| age   | tinyint(2)  | YES  |     | NULL    |       |
| name  | varchar(16) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
rows in set (0.00 sec)
```

14. 备份 oldboy 库及 MySQL 库。

退出 mysql 数据库在命令行输入

```
mysqldump -uroot -poldboy123 -S /data/3306/mysql.sock --events -B oldboy
mysql >/opt/bak.sql
[root@db02 ~]# ll /opt/bak.sql
-rw-r--r-- 1 root root 529737 Dec 21 16:55 /opt/bak.sql
```

15. 删除表中的所有数据，并查看。

```
delete 是逻辑删除表中的数据，一列一列的删除表中数据，速度比较慢
mysql> delete from test;
truncate 是物理删除表中的数据，一次性全部都给清空表中数据，速度很快
mysql> truncate table test;
Query OK, 0 rows affected (0.00 sec)
##查看结果
mysql> select * from test;
Empty set (0.00 sec)
```

16. 删除表 test 和 oldboy 数据库并查看

```
#删除表 test
mysql> show tables;
+-----------------+
| Tables_in_oldboy |
+-----------------+
| test             |
+-----------------+
1 row in set (0.00 sec)

mysql> drop table test;
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> show tables;
Empty set (0.00 sec)
#删除数据库 oldboy
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| oldboy             |
| performance_schema |
+--------------------+
4 rows in set (0.00 sec)

mysql> drop database oldboy;
Query OK, 0 rows affected (0.00 sec)
##查看结果
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
+--------------------+
3 rows in set (0.00 sec)
```

17. Linux 命令行恢复以上删除的数据。

```
#退出 mysql 数据库在命令输入命令
mysql -uroot -poldboy123 -S /data/3306/mysql.sock </opt/bak.sql
#登录数据库并查看结果
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| oldboy             |
| performance_schema |
+--------------------+
4 rows in set (0.00 sec)
#查看 test 表的数据
mysql> select * from oldboy.test;
+------+------+-----------+
| id   | age  | name      |
+------+------+-----------+
|    1 | NULL | oldgirl   |
|    2 | NULL |  老男孩|
|    3 | NULL | etiantian |
+------+------+-----------+
```

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

18. 把 GBK 字符集修改为 UTF8（可选，注意，此题有陷阱）。

```
#首先查看一下
mysql> show variables like 'character_set%';
+--------------------------+----------------------------------------+
| Variable_name            | Value                                  |
+--------------------------+----------------------------------------+
| character_set_client     | utf8                                   |
| character_set_connection | utf8                                   |
| character_set_database   | utf8                                   |
| character_set_filesystem | binary                                 |
| character_set_results    | utf8                                   |
| character_set_server     | utf8                                   |
| character_set_system     | utf8                                   |
| character_sets_dir       | /application/mysql-5.5.32/share/charsets/ |
+--------------------------+----------------------------------------+
8 rows in set (0.00 sec)
###人品好字符集都是 utf8 0(∩_∩)0 哈哈~
#现在开始模拟 gbk 字符集测试环境
mysql> set global character_set_database = gbk;
mysql> quit;
[root@db02 ~]# mysql -uroot -poldboy -S /data/3306/mysql.sock
```

```
mysql> show variables like 'character_set%';
+--------------------------+----------------------------------------+
| Variable_name            | Value                                  |
+--------------------------+----------------------------------------+
| character_set_client     | utf8                                   |
| character_set_connection | utf8                                   |
| character_set_database   | gbk                                    |
| character_set_filesystem | binary                                 |
| character_set_results    | utf8                                   |
| character_set_server     | utf8                                   |
| character_set_system     | utf8                                   |
| character_sets_dir       | /application/mysql-5.5.32/share/charsets/ |
+--------------------------+----------------------------------------+
8 rows in set (0.00 sec)
##现在把字符集修改成 utf8 会了吗？
操作一下吧-_-!!!
mysql> set global character_set_database = utf8;
Query OK, 0 rows affected (0.00 sec)

mysql> quit;
Bye
mysql> show variables like 'character_set%';
+--------------------------+----------------------------------------+
| Variable_name            | Value                                  |
+--------------------------+----------------------------------------+
| character_set_client     | utf8                                   |
```

```
| character_set_connection | utf8                                       |
| character_set_database   | utf8                                       |
| character_set_filesystem | binary                                     |
| character_set_results    | utf8                                       |
| character_set_server     | utf8                                       |
| character_set_system     | utf8                                       |
| character_sets_dir       | /application/mysql-5.5.32/share/charsets/ |
+--------------------------+--------------------------------------------+
8 rows in set (0.00 sec)
#这就完成了！
```

19. MySQL 密码丢了，如何找回实战？

```
#查看一下进程是否服务在运行
[root@db02 ~]# ps -ef |grep 3306
root      1466     1  0 10:36 ?        00:00:00 /bin/sh /application/mysql/bin/mysqld_safe --defaults-file=/data/3306/my.cnf
mysql     2190  1466  0 10:36 ?        00:00:11 /application/mysql-5.5.32/bin/mysqld --defaults-file=/data/3306/my.cnf --basedir=/application/mysql --datadir=/data/3306/data --plugin-dir=/application/mysql/lib/plugin --user=mysql --log-error=/data/3306/mysql_oldboy3306.err --open-files-limit=1024 --pid-file=/data/3306/mysqld.pid --socket=/data/3306/mysql.sock --port=3306
root      3935  3703  0 16:16 pts/3    00:00:00 mysql -uroot -px xxxxxxx -S /data/3306/mysql.sock
```

```
root      4099  4008  0 17:17 pts/1    00:00:00 grep --color=auto 3306
#杀死进程 pid 号
 [root@db02 ~]# kill 1466
 [root@db02 ~]# kill 1466
-bash: kill: (1466) - No such process
[root@db02 ~]# ps -ef |grep 3306
root      3935  3703  0 16:16 pts/3    00:00:00 mysql -uroot -px xxxxxxx -S
 /data/3306/mysql.sock
root      4104  4008  0 17:17 pts/1    00:00:00 grep --color=auto 3306
[root@db02 ~]# pkill mysql   ###<====这里别忘了杀死-_-!!
#启动 mysql 实例 3306 跳过输入用户密码文件
[root@db02 ~]# mysqld_safe --defaults-file=/data/3306/my.cnf --skip-grant-
tables &>/dev/null  &    ##<=====放在后台运行
[1] 4697
[root@db02 ~]# ps -ef |grep 3306
root      4697  4008  1 17:24 pts/1    00:00:00 /bin/sh /application/mysql/
bin/mysqld_safe --defaults-file=/data/3306/my.cnf --skip-grant-tables
mysql     5434  4697  2 17:24 pts/1    00:00:00 /application/mysql-5.5.32/b
in/mysqld --defaults-file=/data/3306/my.cnf --basedir=/application/mysql -
-datadir=/data/3306/data --plugin-dir=/application/mysql/lib/plugin --user
=mysql --skip-grant-tables --log-error=/data/3306/mysql_oldboy3306.err --o
pen-files-limit=1024 --pid-file=/data/3306/mysqld.pid --socket=/data/3306/
mysql.sock --port=3306
root      5453  4008  0 17:24 pts/1    00:00:00 grep --color=auto 3306
[root@db02 ~]# netstat -ltunp |grep 3306
```

```
tcp        0      0 0.0.0.0:3306                0.0.0.0:*                   LI
STEN      5434/mysqld
#登录到数据库 3306 实例
[root@db02 ~]# mysql -S /data/3306/mysql.sock
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.5.32-log Source distribution
Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserve
d.
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 stateme
nt.
mysql>
#修改 mysql 超级管理 root 密码
mysql> update mysql.user set password=PASSWORD('oldboy') where user='root'
 and host='localhost';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0
mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)
#修改 mysql 启动脚本中用户密码
[root@db02 ~]# grep "pwd" /data/3306/mysql
mysql_pwd="oldboy"
#停止服务
```

```
[root@db02 ~]# /data/3306/mysql stop
Stoping MySQL...
[1]+  Done                    mysqld_safe --defaults-file=/data/3306/my.cnf
 --skip-grant-tables &>/dev/null
[root@db02 ~]# netstat -lutnp |grep 3306
#在启动服务
[root@db02 ~]# /data/3306/mysql start
Starting MySQL...
#登录数据库
[root@db02 ~]# mysql -uroot -poldboy -S /data/3306/mysql.sock
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.5.32-log Source distribution
Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserve
d.
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 stateme
nt.
mysql>
#完成!
```

20.MySQL 内中文数据乱码的原理及如何防止乱码？（可选）。

```
数据库服务的字符集
|character_set_client         | utf8                                |
| character_set_connection | utf8                                 |
| character_set_database      | utf8                              |
| character_set_filesystem | binary                                |
| character_set_results       | utf8                              |
| character_set_server         | utf8                               |
| character_set_system         | utf8                                |
系统字符集配置文件
[root@db02 ~]# cat /etc/sysconfig/i18n
LANG="en_US.UTF-8"
SYSFONT="latarcyrheb-sun16"
客户端软件 CRT/xshell 等软件字符集要用 utf8
```

21.在把 id 列设置为主键，在 Name 字段上创建普通索引。

```
mysql> alter table oldboy.test add primary key(id);
Query OK, 3 rows affected (0.01 sec)
Records: 3    Duplicates: 0    Warnings: 0

mysql> desc oldboy.test;
```

```
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| id    | int(4)      | NO   | PRI | 0       |       |
| age   | tinyint(2)  | YES  |     | NULL    |       |
| name  | varchar(16) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

#name 字段上创建普通索引

方法一：

```
mysql> alter table oldboy.test add index index_name(name);
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

方法二：

```
mysql> create index index_name on oldboy.test(name);
#查看一下
mysql> desc oldboy.test;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| id    | int(4)      | NO   | PRI | 0       |       |
| age   | tinyint(2)  | YES  |     | NULL    |       |
| name  | varchar(16) | YES  | MUL | NULL    |       |
+-------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

22.在字段 name 后插入手机号字段(shouji)，类型 char(11)。

```
mysql> alter table oldboy.test add shouji char(11) after name;
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> desc oldboy.test;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| id     | int(4)      | NO   | PRI | 0       |       |
| age    | tinyint(2)  | YES  |     | NULL    |       |
| name   | varchar(16) | YES  | MUL | NULL    |       |
| shouji | char(11)    | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

23.所有字段上插入 2 条记录（自行设定数据）

```
mysql> insert into oldboy.test
values(4,21,'yifeny','15511024633'),(5,38,'oldboy','13512569874');
Query OK, 2 rows affected (0.00 sec)
Records: 2  Duplicates: 0  Warnings: 0
```

```
mysql> select * from oldboy.test;
+----+------+-----------+-------------+
| id | age  | name      | shouji      |
+----+------+-----------+-------------+
|  1 | NULL | oldgirl   | NUL         |
|  2 | NULL | 老男孩    | NULL        |
|  3 | NULL | etiantian | NUL         |
|  4 |   21 | yifeny    | 15511024633 |
|  5 |   38 | oldboy    | 13512569874 |
+----+------+-----------+-------------+
5 rows in set (0.00 sec)
```

24. 在手机字段上对前 8 个字符创建普通索引。

```
mysql> alter table oldboy.test add index index_shouji(shouji(8));
Query OK, 0 rows affected (0.15 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc oldboy.test;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| id     | int(4)      | NO   | PRI | 0       |       |
| age    | tinyint(2)  | YES  |     | NULL    |       |
```

```
| name     | varchar(16) | YES    | MUL | NULL |       |
| shouji | char(11)       | YES    | MUL | NULL     |    |
+--------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

25. 查看创建的索引及索引类型等信息。

```
mysql> show index from oldboy.test\G
*************************** 1. row ***************************
        Table: test
   Non_unique: 0
     Key_name: PRIMARY
  Seq_in_index: 1
  Column_name: id
    Collation: A
  Cardinality: 5
     Sub_part: NULL
       Packed: NULL
         Null:
   Index_type: BTREE
      Comment:
Index_comment:
*************************** 2. row ***************************
        Table: test
   Non_unique: 1
     Key_name: index_name
```

```
  Seq_in_index: 1
  Column_name: name
    Collation: A
  Cardinality: 5
     Sub_part: NULL
       Packed: NULL
         Null: YES
   Index_type: BTREE
      Comment:
Index_comment:
*************************** 3. row ***************************
        Table: test
   Non_unique: 1
     Key_name: index_shouji
  Seq_in_index: 1
  Column_name: shouji
    Collation: A
  Cardinality: 5
     Sub_part: 8
       Packed: NULL
         Null: YES
   Index_type: BTREE
      Comment:
Index_comment:
3 rows in set (0.00 sec)
```

26. 删除 Name，shouji 列的索引。

```
mysql> alter table oldboy.test drop index index_name;
Query OK, 0 rows affected (0.00 sec)
Records: 0  Duplicates: 0  Warnings: 0
mysql> drop index index_shouji on oldboy.test;
Query OK, 0 rows affected (0.00 sec)
Records: 0  Duplicates: 0  Warnings: 0


mysql> show index from oldboy.test\G
*************************** 1. row ***************************
        Table: test
   Non_unique: 0
     Key_name: PRIMARY
  Seq_in_index: 1
  Column_name: id
    Collation: A
  Cardinality: 5
     Sub_part: NULL
       Packed: NULL
         Null:
   Index_type: BTREE
      Comment:
Index_comment:
1 row in set (0.00 sec)
```

27.对 Name 列的前 6 个字符以及手机列的前 8 个字符组建联合索引。

```
mysql> alter table oldboy.test add index index_name_shouji(name(6),shouji(8));
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> show index from oldboy.test\G
*************************** 1. row ***************************
        Table: test
   Non_unique: 0
     Key_name: PRIMARY
  Seq_in_index: 1
  Column_name: id
    Collation: A
  Cardinality: 5
     Sub_part: NULL
       Packed: NULL
         Null:
   Index_type: BTREE
      Comment:
Index_comment:
*************************** 2. row ***************************
        Table: test
   Non_unique: 1
     Key_name: index_name_shouji
```

```
  Seq_in_index: 1
   Column_name: name
     Collation: A
   Cardinality: 5
      Sub_part: 6
        Packed: NULL
          Null: YES
    Index_type: BTREE
       Comment:
Index_comment:
*************************** 3. row ***************************
         Table: test
    Non_unique: 1
      Key_name: index_name_shouji
  Seq_in_index: 2
   Column_name: shouji
     Collation: A
   Cardinality: 5
      Sub_part: 8
        Packed: NULL
          Null: YES
    Index_type: BTREE
       Comment:
Index_comment:
3 rows in set (0.00 sec)
```

28.查询手机号以135开头的，名字为oldboy的记录（此记录要提前插入）。

```
mysql> select * from oldboy.test where name='oldboy' and shouji like '135%';
+----+------+--------+-------------+
| id | age  | name   | shouji      |
+----+------+--------+-------------+
|  5 |   38 | oldboy | 13512569874 |
+----+------+--------+-------------+
1 row in set (0.00 sec)
```

29.查询上述语句的执行计划（是否使用联合索引等）。

```
mysql> explain select * from oldboy.test where name='oldboy' and shouji like '135%';
+----+-------------+-------+-------+-----------------+-----------------+---------+------+------+-------------+
| id | select_type | table | type  | possible_keys   | key             | key_len | ref  | rows | Extra       |
+----+-------------+-------+-------+-----------------+-----------------+---------+------+------+-------------+
| 1 | SIMPLE | test | range | index_name_shouji | index_name_shouji | 32 | NULL |  1 |Using where |
+----+-------------+-------+-------+-----------------+-----------------+---------+------+------+-------------+
1 row in set (0.00 sec)
mysql> explain select * from oldboy.test where name='oldboy' and shouji like '135%'\G
*************************** 1. row ***************************
           id: 1
  select_type: SIMPLE
```

```
            table: test
             type: range
possible_keys: index_name_shouji
              key: index_name_shouji
          key_len: 32
              ref: NULL
             rows: 1
            Extra: Using where
1 row in set (0.00 sec)
```

30.把 test 表的引擎改成 MyISAM。

MySQL 数据库 5.1 版本以前默认的引擎是 MyISAM

MySQL 数据库 5.5 版本以后默认的引擎都是 InnoDB

```
    #首先现在查看一下 test 表的引擎
mysql> show create table oldboy.test\G
*************************** 1. row ***************************
         Table: test
Create Table: CREATE TABLE `test` (
  `id` int(4) NOT NULL DEFAULT '0',
  `age` tinyint(2) DEFAULT NULL,
  `name` varchar(16) DEFAULT NULL,
  `shouji` char(11) DEFAULT NULL,
  PRIMARY KEY (`id`),
```

```
  KEY `index_name_shouji` (`name`(6),`shouji`(8))
) ENGINE=InnoDB DEFAULT CHARSET=gbk
1 row in set (0.00 sec)
      #修改默认引擎为 MyISAM
mysql> alter table oldboy.test ENGINE=MyISAM;
Query OK, 5 rows affected (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 0


mysql> show create table oldboy.test\G
*************************** 1. row ***************************
       Table: test
Create Table: CREATE TABLE `test` (
  `id` int(4) NOT NULL DEFAULT '0',
  `age` tinyint(2) DEFAULT NULL,
  `name` varchar(16) DEFAULT NULL,
  `shouji` char(11) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `index_name_shouji` (`name`(6),`shouji`(8))
) ENGINE=MyISAM DEFAULT CHARSET=gbk
1 row in set (0.00 sec)
```