msyql

- mysql

mysql

mysql配置

```
[client]
default-character-set=utf8mb4

[mysqld-配置]
bind-address=0.0.0.0
character_set_server=utf8mb4
collation-server = utf8mb4_unicode_ci
init_connect='SET NAMES utf8mb4'

datadir=/data/mysqldata/
socket=/data/mysqldata/mysql.sock

symbolic-links=0 #禁用符号链接

log-error=/data/mysqldata/mysqld.log
pid-file=/data/mysqldata/mysqld.pid

--initialize
--user
--datadir
--basedir
--defaults-file

mysql -uroot -p -S *.sock

lrzsz #文件传输工具
navicat #数据库工具包
sqyoy #数据库工具包

tar xf mysql-boost-8.0.45.tar.gz
```

三、 MySQL编译安装

```
mkdir build && cd build
cmake .. \
-DCMAKE_INSTALL_PREFIX=/usr/local/mysql-8.0.45\
-DDEFAULT_CHARSET=utf8mb4\
-DDEFAULT_COLLATION=utf8mb4_unicode_ci \
-DENABLED_LOCAL_INFILE=ON \
-DWITH_SSL=system\
-DMYSQL_DATADIR=/data/mysqldata\
-DMYSQL_TCP_PORT=3306 \
-DDOWNLOAD_BOOST=0 \
-DWITH_BOOST=../boost
make -j 4 #4代表CPU核心
make install
yum install -y elfutils-libelf-devel binutils
yum install -y ncurses-devel
```

四、开发工具

```
PL?SQL Developer #PL/SQL开发工具
toad for Oracle #Oracle数据库工具包
```

五、包环境

```
rpm -qa | grep glibc
```

起库

```
/usr/local/mysql-8.0.45/bin/mysql_safe --defaults-file=/etc/my3307.cnf &
```

关库

```
/usr/local/mysql-8.0.45/bin/mysqldadmin -uroot -p -S /data/mysqldata/mysql.sock shutdown
```

本地连接

```
/usr/local/mysql-8.0.45/bin/mysql -uroot -p -S /data/mysqldata/mysql.sock



select user,host,authentication_string,plugin from mysql.user;
```

数据类型

1.整数 tinyint 很小的整数 1字节 -128~~127 smallint~~ 小整整数 2字节 -32768~~32767~~ mediumint 中整数 3字节 -8688~~3067215 int~~ 整数 4字节 -2147483648~~2147483647~~ bigint 大整数 8字节 -9223372036854775808~9223372036854775807

2.浮点数与定位数 float 单精度浮点数 4字节 -3.4028234663852886E+38~~3.4028234663852886E+38~~ ~~double~~ 双精度浮点数 8字节 -1.7976931348623157E+308~~1.7976931348623157E+308~~ decimal 定位数 15字节 -10^38~10^38

```
create table ccc(x float(4,2),y double(5,3),z decimal(4,2));
insert into ccc values (8.666,8.666,8.666);
select * from ccc;
```

3.日期与时间 year 年 1字节 1901~~2155 time~~ 时间 3字节 ~~00:00:00~~23:59:59 date 日期 3字节 1000-01-01~~9999-12-31 datetime~~ 日期时间 8字节 ~~1000-01-01 00:00:00~~9999-12-31 23:59:59 timestamp 时间戳 4字节 1970-01-01 00:00:00~:00~2038-0

4.文本字符 char 字符 1字节 0~~255 varchar~~ 可变长度字符串 ~~1~~65535字节 text 大文本 ~~1~~65535字节 ~~enum~~ 枚举值 ~~1~~字节 ~~0~~255

```
set             集合值          1字节         0~65535
```

5.二进制字符 bit 位 1字节 0~~1 enum~~ 枚举值 ~~1~~字节 ~~0~~255 varchar 可变长度字符串 ~~1~~65535字节 ~~longblob~~ 大二进制大对象 ~~1~~45775206字节

数据库字符集

```
create database database_name charset='utf8mb4' collate='utf8mb4_unicode_ci';
```

查看用的什么字符集

```
select default_character_set_name,default_collation_name from information_schema.schemata
where schema_name='database_name';
drop database database_name;
```

十一、数据库改名

```
rename database database_name to test; 旧语法。8.0不再支持。

use test001;
create database test001;
create table ddd(id int, name varchar(255),age int,phone varchar(255));
insert into ddd values (1,'张三',18,'13800000000'),(2,'李四',19,'13900000000'),(3,'王五
',20,'13800000001');

create table fff(id int, name varchar(255),age int,phone varchar(255));
insert into fff values (1,'张三',18,'13800000000'),(2,'李四',19,'13900000000'),(3,'王五
',20,'13800000001');

select * from ddd;
select * from fff;

create database test002;

rename table test001.ddd to test002.ddd;
rename table test001.fff to test002.fff;

use test002;
select * from ddd;
select * from fff;

use test001;
show tables;

drop database test001;
drop database test002;
```

十二、查看当前在哪个数据库

```
create database aaa charset='utf8mb4' collate='utf8mb4_unicode_ci';
use aaa;
select database();
drop database aaa;
```

十三、变更表结构

```
create database aaa charset='utf8mb4' collate='utf8mb4_unicode_ci'; #创建数据库aaa
use aaa;
create table bbb(id int, name varchar(255),age int,phone varchar(255));

alter table bbb add column email varchar(255);
desc bbb;

drop database aaa; #删除数据库aaa
```

十四、临时表

```
create database aaa charset='utf8mb4' collate='utf8mb4_unicode_ci'; #创建数据库aaa
use aaa;
create temporary table ccc(id int, name varchar(255),age int,phone varchar(255));
show tables; #查看临时表
insert into ccc values (1,'张三',18,'13800000000'),(2,'李四',19,'13900000000'),(3,'王五
',20,'13800000001');
select * from ccc; #查看临时表ccc

drop database ccc; #删除临时数据库ccc
```

3阶段 ---------

十五、 DCL

作用：授权、撤销、创建 /删除角色

```
grant #授权
revoke #撤销
create role / drop role #创建/删除角色
grant role / drop role
```

十六、 DDL

[illegible] create:[illegible] alter:[illegible] drop:[illegible] rename:[illegible] truncate:[illegible]

[illegible] DML

[illegible] select:[illegible] insert:[illegible] update:[illegible] delete:[illegible]

```
create database aaa charset='utf8mb4' collate='utf8mb4_unicode_ci'; #[illegible]aaa
use aaa;
show tables;
-- [illegible]
create table bbb (
    server_id int primary key auto_increment comment '[illegible]id([illegible])',
    hostname varchar(50) not null comment '[illegible]',
    ip varchar(255) not null comment '[illegible]ip[illegible]',
    os varchar(50) not null comment '[illegible]',
    cpu int comment 'cpu[illegible]',
    mem int comment '[illegible]([illegible]gb)',
    disk int comment '[illegible]([illegible]gb)',
    status enum('running','stopped','maintenance','error') default 'running' comment '[illegible]
',
    create_date date comment '[illegible]'
) engine=innodb default charset=utf8mb4 comment='[illegible]';

-- [illegible]
create table ccc (
    id int primary key auto_increment comment '[illegible]',
    server_id int not null comment '[illegible]id',
    record_time datetime comment '[illegible]',
    foreign key (server_id) references bbb(server_id) on delete cascade on update cascade
) engine=innodb default charset=utf8mb4 comment='[illegible]';

-- [illegible]
create table ddd (
    id int primary key auto_increment comment '[illegible]',
    server_id int not null comment '[illegible]id',
    record_time datetime comment '[illegible]',
    success tinyint(1) comment '[illegible] 0=[illegible] 1=[illegible]',
    foreign key (server_id) references bbb(server_id) on delete cascade on update cascade
```

```
) engine=innodb default charset=utf8mb4 comment='服务器操作日志表';

-- 插入服务器基础数据
insert into bbb (hostname, ip, os, cpu, mem, disk, status, create_date)
values
('linux-test-01', '120.92.11.101', 'centos 7.9', 2, 2, 40, 'running', '2026-07-01'),
('linux-note-web', '120.92.11.102', 'ubuntu 22.04', 2, 4, 60, 'running', '2026-07-05'),
('mysql-db-server', '120.92.11.103', 'centos stream 9', 4, 8, 100, 'maintenance', '2026-06-
20');

-- 插入监控采集记录
insert into ccc (server_id, record_time)
values
(1, '2026-07-18 08:30:00'),
(1, '2026-07-18 10:15:00'),
(2, '2026-07-18 09:20:00'),
(3, '2026-07-18 14:00:00');

-- 插入操作日志数据
insert into ddd (server_id, record_time, success)
values
(1, '2026-07-17 11:20:00', 1),
(1, '2026-07-17 15:40:00', 0),
(2, '2026-07-18 09:10:00', 1),
(3, '2026-07-16 16:30:00', 1);

select * from bbb;
select * from ccc;
select * from ddd;

desc bbb;
desc ccc;
desc ddd;

delete from ccc where server_id = 1;
select * from ccc;



is not null
like '%libai%'';
```

```
count(*) #统计记录数
sum(cpu) #统计cpu核心数总和
avg(mem) #统计内存大小平均值
max(disk) #统计磁盘大小最大值
min(disk) #统计磁盘大小最小值
```

十八、事务

```
start transaction; #开始事务
commit; #提交事务
rollback; #回滚事务

DDL truncate #不受事务影响
```

```
primary key                          #主键，自增
auto_increment                       #自增，每次插入新记录时，自动增加1
enum('枚举1','枚举2','枚举3')      #枚举类型，只能取指定的值
default '枚举1'                      #默认值
date                                     #日期类型，存储年月日信息
datetime                                 #日期时间类型，存储年月日时分秒信息
foreign key                           #外键，引用其他表的字段，用于维护数据的完整性和一致性。
```

十九、 mysql_secure_installation

```
Would you like to setup VALIDATE PASSWORD component?
Change the password for root ?
Remove anonymous users?
Disallow root login remotely?
Remove test database and access to it?
Reload privilege tables now?
```

二十、 mysql

二十一、 maridb

```
echo "skip-grant-tables=1" >> /etc/my.cnf.d/mysql-server.cnf
mysql -u root -p123456 -e {"show processlist;";"show master status;";"flush binary
logs;";"show binary logs;"}
```

□□□□□

```
create user "user001"@"%" identified by '123456';
alter user "user001"@"%" identified by '258069'; {set password for 'root'@'localhost' =
password('123456');□}
rename user "user002"@"%" to "user003"@"%";

select concat('"',user,'"','@','"',host,'"') from mysql.user;    -->select
concat('"',user,'"@"',host,'"') from mysql.user;
select concat('\'',user,'\'','@','\'',host,'\'') from mysql.user;-->select
concat('\'',user,'\'@\'',host,'\'') from mysql.user;
```

□□□□

```
create database class; && use class; | drop database class;
```

□□□□ □□

```
create table job(); | drop table job;
primary key=not null+unique key
auto_increment
unsigned
zerofill
age int check(age >= 0 and age <= 120)
score int check(score between 0 and 100)
default '□□'


int {auto_increment;primary key;comment '□□'} {-- ;#;/* */;}
decimal()
test()
varchar() {not null;unique}
```

```
char()
enum('男','女') default '男'
tinyint (-128 ~ 127)
amount decimal(10,2)
date
created_at datetime default current_timestamp
updated_at datetime default current_timestamp on update current_timestamp

show create table job;
```

二十五、插入 | 更改 | 索引 | 删除

```
insert into job() values();
alter table job {add;modify;drop;add unique index;drop index}
alter table job add age int comment "年龄" after username; | alter table job drop age;
alter table job add unique index aaa(mail); --> show {keys;indexes} from job;  | alter table
job drop index aaa;
alter table job modify column phone varchar(100);
update job set {username="libai";age=age+100}{,mail="heihei}" where
{username="user01";id=1;phone is null limit 1;id>10}
delete from job where mail="hei hei"; --> delete from job;
truncate table job;
```

二十六、权限

```
grant all privileges on *.* to "user001"@"%" with grant option; --> show grants for
"user001"@"%";-->select * from mysql.user where User='user001' \G
revoke {file; update; super} on *.* from "user001"@"%";
{select;insert;update;delete}
{create;drop;alter;index;}
{create user;drop user;grant option;all privileges}
{file}
flush privileges;
```

二十七、角色

```
create user a_role;
grant all privileges on *.* to a_role;
create user user001@'%' identified by '123456';
grant "a_role"@"%" to "user001"@"%";
set default role "a_role"@"%" to "user001"@"%";
```

□□□□□

```
show databases; && show tables;
!= # □□□
select * from class.job where id<10 and id>5; in(5,6,7,8,9)
select * from class.job where id between 5 and 10;
select * from class.job order by id {asc;desc;linuxnote-[desc,id2 asc]}
select * from class.job limit 5 {offset 10}
select * from class.job like '%libai%';
desc job; --> show full {columns;fields} from job;
select user,host from mysql.user;-->select concat('"',user,'"@"',host,'"') from mysql.user;
show keys from job;
show binary logs;
show variables like
{'bind_address';'log_bin';log_bin_basename;server_id;local_infile;log_error;relay_log};

select user from mysql.user where user='';
show databases like 'test%';
```

□□□□□

```
exit; && quit;
```

□□□□□□□□

```
yum install https://dev.mysql.com/get/mysql80-community-release-el7-6.noarch.rpm -y
yum install -y mysql-community-server.x86_64



systemctl start mysqld && systemctl enable mysqld && systemctl status mysqld
rm -rf /var/lib/mysql
```

```
mkdir /var/lib/mysql
chown -R mysql:mysql /var/lib/mysql
chmod 750 /var/lib/mysql
systemctl start mysqld && systemctl enable mysqld && systemctl status mysqld
grep "password" /var/log/mysqld.log
alter user root@localhost identified by 'Root@123456';



SET GLOBAL validate_password.policy = LOW;
SET GLOBAL validate_password.length = 4;
SET GLOBAL validate_password.mixed_case_count = 0;
SET GLOBAL validate_password.number_count = 0;
SET GLOBAL validate_password.special_char_count = 0;
```

validate_password.policy=LOW □ 0~2□ 2 □□□□

```
validate_password.length=4
validate_password.mixed_case_count=0
validate_password.number_count=0
validate_password.special_char_count=0
```

□□□ mysqld□□□

```
show binary logs;
show variables like {'bind_address';'log_bin';server_id;local_infile;'validate_password%'};

bind-address = 0.0.0.0
local_infile = 0
revoke file on *.* from "user001"@"%" ;

general_log = 1
general_log_file = /var/log/mysql/general.log
server-id = 1
log-bin = /var/log/mysql/mysql-bin



select user from mysql.user where user='';
select user, host from mysql.user where user='root' and host='%';
show databases like 'test%';
```

□□□□□□□□□ mysql

```
groupadd mysql && useradd -r -g mysql -s /bin/false mysql
https://downloads.mysql.com/archives/community/
wget https://downloads.mysql.com/archives/get/p/23/file/mysql-5.7.42-linux-glibc2.12-x86_64.tar.gz
md5sum mysql-5.7.42-linux-glibc2.12-x86_64.tar.gz
tar -zxvf mysql-5.7.42-linux-glibc2.12-x86_64.tar.gz
mkdir -p /usr/local/mysql  && mv mysql-5.7.42-linux-glibc2.12-x86_64/* /usr/local/mysql && rm -rf mysql-5.7.42-linux-glibc2.12-x86_64*
export PATH=/usr/local/mysql/bin:$PATH  --> source /etc/profile && echo $PATH && which mysqldump
echo 'export PATH=/usr/local/mysql/bin:$PATH' | tee -a /etc//profile && source /etc/profile
groupadd -M -s /sbin/nologin -g mysql mysql
mkdir -p /mysql && chown -R mysql:mysql /mysql/ && chmod -R 755 /mysql/
mysqld --initialize --user=mysql --basedir=/usr/local/mysql --datadir=/mysql

sudo tee /etc/systemd/system/mysql.service <<EOF
linuxnote-[Unit]
Description=□□□□□
After=network.target
linuxnote-[Service]
User=mysql
Group=mysql
ExecStart=/usr/local/mysql/bin/mysqld --defaults-file=/etc/my.cnf
Restart=on-failure
linuxnote-[Install]
WantedBy=multi-user.target
EOF
sudo systemctl daemon-reload  && sudo systemctl start mysql  && sudo systemctl enable mysql && systemctl status mysql

mysql -uroot -p
dnf install -y ncurses-compat-libs  && ldd /usr/local/mysql/bin/mysql
```

□□□ mysqldump|□□

```
mysql -u root -p123456 -e "show processlist;"
mysqldump -uroot -p123456 {--add-databases;class;class job;} > *.sql
mysqldump -u root -p123456 --comments --skip-lock-tables --single-transaction class | gzip >
class-$(date +%F).sql.gz
{--lock-tables;--no-data;--where=}

mysql -u root -p123456 -e "drop database class;"
mysql -u root -p123456 -e "create database class;"
mysql -uroot -p123456 < all.sql
mysql -uroot -p123456 class < class.sql
mysql -uroot -p123456 class < job.sql
```

□□□□□□□

```
rsync -av /mysql/ /opt/mysql$(date +%F)
rsync -av /opt/mysql2026-06-19/mysql/ /mysql

server-id = 1
log-bin = /var/lib/mysql/mysql-bin
```

□□□□□□□□

```
mysql -u root -p123456 -e "show processlist;"
mysql -u root -p123456 -e "show master status;"
mysql -u root -p123456 -e "flush binary logs;"
mysql -uroot -p123456 -e "show binary logs;"

rsync -av /mysql/mysql-bin.00000{1..4} /opt/mysql-bin-$(date +%F)
mysqlbinlog /opt/mysql-bin-2026-06-20/mysql-bin.000001 | mysql -uroot -p123456
```

□□□ XtraBackup|□□□

```
yum install -y https://repo.percona.com/yum/percona-release-latest.noarch.rpm
percona-release enable-only tools release
yum install -y percona-xtrabackup-24
create user 'user001'@'localhost' identified by '123456';
```

```
grant process,reload, lock tables, replication client, create, insert, select on *.* to
'user001'@'localhost';
flush privileges;
innobackupex --user=user001 --password=123456 --socket=/tmp/mysql.sock --no-timestamp
/opt/mysql-x
innobackupex --apply-log /opt/mysql-x/
innobackupex --prepare /opt/mysql-x
systemctl stop mysql
rm -rf /mysql/*
innobackupex --copy-back /opt/mysql-x
chown -R mysql:mysql /mysql
systemctl start mysql
mysql -uuser001 -p123456 -e "show databases;"
mysql -uroot -p123456
```

```
innobackupex --user=user001 --password=123456 --socket=/tmp/mysql.sock --no-timestamp --
databases="class" /opt/db_class
innobackupex --user=user001 --password=123456 --socket=/tmp/mysql.sock --no-timestamp --
databases="class student test_db" /opt/db_class_student_test_db
innobackupex --user=user001 --password=123456 --socket=/tmp/mysql.sock --no-timestamp --
include='test_db.*' /opt/db_test_db_all
innobackupex --user=user001 --password=123456 --socket=/tmp/mysql.sock --no-timestamp --
include="class.job" /opt/db_class.job
```

```
innobackupex --user=user001 --password=123456 --socket=/tmp/mysql.sock --no-timestamp --
incremental /opt/inc1 --incremental-basedir=/opt/mysql-x
innobackupex --user=user001 --password=123456 --socket=/tmp/mysql.sock --no-timestamp --
incremental /opt/inc2 --incremental-basedir=/opt/inc1
```

```
cat /opt/inc1/xtrabackup_checkpoints
innobackupex --apply-log --redo-only /opt/mysql-x
innobackupex --apply-log --redo-only /opt/mysql-x --incremental-dir=/opt/inc1
innobackupex --apply-log /opt/mysql-x --incremental-dir=/opt/inc2
systemctl stop mysql
rm -rf /mysql/*
innobackupex --copy-back /opt/mysql-x
chown -R mysql:mysql /mysql
systemctl start mysql

innobackupex --apply-log --export /opt/db_class.job
systemctl stop mysql
rm -f /mysql/class/job.*
systemctl start mysql
 -create table job
cp /opt/db_class.job/class/job.ibd /mysql/class/
cp /opt/db_class.job/class/job.cfg /mysql/class/
chown mysql:mysql /mysql/class/job.*
alter table job discard tablespace;
cp /opt/db_class.job/class/job.ibd /mysql/class/
cp /opt/db_class.job/class/job.cfg /mysql/class/
chown mysql:mysql /mysql/class/job.*
alter table job import tablespace;
select * from job;

innobackupex --verify
```

□□□ mysql□□□

```
ls -l /proc/11862/exe

mkdir -p /mysql
chown -R mysql.mysql /mysql
chmod -R 750 /mysql

whereis mysqld && which mysqld
msyqld --defaults-file=/etc/mysql2.cnf --initialize --user=mysql
vim /etc/systemd/system/mysql2.service
```

```
[Unit]
Description=MySQL□□□3307
After=network.target syslog.target

[Service]
User=mysql
Group=mysql
Type=notify
ExecStart=/usr/libexec/mysqld --defaults-file=/etc/mysql2.cnf
ExecReload=/bin/kill -HUP $MAINPID
KillMode=mixed
Restart=on-failure
LimitNOFILE=65535

[Install]
WantedBy=multi-user.target
systemctl start mysql2.service

mysql_secure_installation --defaults-file=/etc/mysql2.cnf

mysqld --defaults-file=/etc/mysql2.cnf -uroot -p123456
mysql -uroot -p123456 -h 127.0.0.1 -P 3307
mysql -u root -p123456 -S /mysql/mysql.sock

create user "root"@"%" identified by '123456';
grant all privileges on *.* to "root"@"%" with grant option;
flush privileges ;
mysql -uroot -p123456 -h 192.168.20.145 -P 3307
```

□□□□□□□

```
systemctl stop mysql2
systemctl disable mysql2
rm -f /etc/systemd/system/mysql2.service
systemctl daemon-reload
rm -f /etc/mysql2.cnf
```

□□□□□□□

```
vim /etc/my.cnf.d/mysql-server.cnf
server-id=1
log-bin=/var/log/mysql/mysql-bin.log
sudo systemctl enable --now mysqld && systemctl status mysqld --no-pager && mysql --version
```

```
vim /etc/my.cnf.d/mysql-server.cnf
server-id = 2
relay-log = /var/log/mysql/relay-bin.log
read_only = 1
```

```
mysqldump -uroot -p123456 --all-databases > all.sql && scp all.sql 192.168.20.147:~

create user repl@'%' identified by '123456';
grant replication slave on *.* to repl@'%';
flush privileges;
select * from mysql.user where User='repl' \G

show master status;
show binary logs;
```

8.0.23+

```
change replication source to
    source_host='192.168.20.145',
    source_port=3306,
    source_user='repl',
    source_password='123456',
    source_log_file='mysql-bin.000001',
    source_log_pos=1527,
    source_ssl=1;
```

5.7 / 8.0.22-

```
change master to
    master_host='192.168.20.146',
```

```
    master_port=3306,
    master_user='repl',
    master_password='123456',
    master_log_file='mysql-bin.000003',
    master_log_pos=13956,
    master_ssl=1;

show variables like 'server_id';
```

8.0.22□

```
start slave;
stop slave;
show slave status\G
```

8.0.22□

```
start replica;
stop replica;
show replica status\G
reset replica all;
set global sql_slave_skip_counter = 1;
```

□□□□□ mysqld

```
systemctl stop mysqld && dnf remove mysql-server mysql -y
rm -rf /var/lib/mysql
rm -rf /etc/my.cnf
rm -rf /var/log/mysql
dnf install mysql-server -y
```

□□□ proxysql|□□□ |□□

```
cat > /etc/yum.repos.d/proxysql.repo <<'EOF'
linuxnote-[proxysql_repo]
name=ProxySQL 3.0.x Official Repo
baseurl=https://repo.proxysql.com/ProxySQL/proxysql-3.0.x/almalinux/$releasever
gpgcheck=1
```

```
gpgkey=https://repo.proxysql.com/ProxySQL/proxysql-3.0.x/repo_pub_key
EOF
dnf install proxysql -y
sudo systemctl enable --now proxysql && systemctl status proxysql --no-pager && proxysql --
version
```

□□□□□□□

```
show status like 'threads_connected';
--lock-all-tables
mysqlduml -uroot -p123456 --single-transaction  -h 192.168.20.146 class name > name.sql
mysqlduml -uroot -p123456 --single-transaction -h 192.168.20.146 class job > job.sql
create database name;
create database job;
mysql -uroot -p123456 name < name.sql
mysql -uroot -p123456 job < job.sql
```

□□□□□□□

```
create table namme1 like name;
create table namme2 like name;
create table namme3 like name;
create table namme4 like name;

insert into name1 select * from name where id between 1 and 10;
insert into name2 select * from name where id between 11 and 20;
insert into name3 select * from name where id between 21 and 30;
insert into name4 select * from name where id between 31 and 40;



insert into name1 select * from name where class_name='□□';
insert into name2 select * from name where class_name='□□';
insert into name3 select * from name where class_name='□□';
insert into name4 select * from name where class_name='□□';

select * from name1
union all
```

```
select * from name2
union all
select * from name3
union all
select * from name4;


create view all-name as
select * from name1
union all
select * from name2
union all
select * from name3
union all
select * from name4;
drop view if exists all-name;
```

□□□ proxysql□□□

□□ proxysql□

```
mysql -uadmin -padmin -h 127.0.0.1 -P 6032
insert into mysql_servers(hostgroup_id, hostname, port) values
(10, '192.168.20.146', 3306),
(20, '192.168.20.147', 3306);
save mysql variables to disk;
load mysql variables to runtime;
```

□

```
create user 'monitor'@'192.168.20.%' identified by '123456';
grant replication client on *.* to 'monitor'@'192.168.20.%';
flush privileges;
```

□□ proxysql□

```
set mysql-monitor_username='monitor';
set mysql-monitor_password='123456';
save mysql variables to disk;
load mysql variables to runtime;
```

□

```
create user appuser@'%' identified by '123456';
grant all on *.* to "appuser"@"%";
flush privileges;
```

□□ proxysql;

```
insert into mysql_users (username, password, default_hostgroup, transaction_persistent)
values ('appuser', '123456', 10, 1);
save mysql users to disk;
load mysql users to runtime;


insert into mysql_query_rules (rule_id, active, match_digest, destination_hostgroup, apply)
values (1, 1, '^select.*', 20, 1);
load mysql query rules to runtime;
save mysql query rules to disk;


load mysql variables to runtime;
save mysql variables to disk;

load mysql servers to runtime;
save mysql servers to disk;

load mysql users to runtime;
save mysql users to disk;


insert into mysql_servers(hostgroup_id, hostname, port) values
(10, '192.168.20.146', 3306),
(10, '192.168.20.147', 3306);

insert into mysql_replication_hostgroups (writer_hostgroup, reader_hostgroup, comment)
values (10, 20, '□□□□□');

load mysql servers to runtime;
save mysql servers to disk;
```

□□□□

```
update mysql_servers set weight=50 where hostname='192.168.20.146';
update mysql_servers set weight=100 where hostname='192.168.20.147';
load mysql servers to runtime;
save mysql servers to disk;
```

□□

```
update global_variables set variable_value='5000' where variable_name='mysql-
monitor_local_dns_cache_refresh_interval';

update mysql_servers set status='offline_soft' where hostname='172.22.4.4'
select * from stats_mysql_query_rules;
select * from stats_mysql_connection_pool;
select hostgroup,sum_time,count_star from stats_mysql_query_digest;
```

□□□□□

```
exit
mysql -uadmin -padmin -h127.0.0.1 -P6032 -e "select hostgroup_id, hostname, status from
runtime_mysql_servers;"
mysql -u appuser -p123456 -h 192.168.20.145 -P 6033 -e "select @@server_id, @@hostname;"
mysql -u appuser -p123456 -h 192.168.20.145 -P 6033 -e "select * from class.students;"
mysql -u appuser -p123456 -h 192.168.20.145 -P 6033 -e "insert into class.students (name, age,
class_name) values ('□□□□□□6', 25, '□□□');"
mysql -uadmin -padmin -h127.0.0.1 -P6032 -e "select hostgroup, count_star, digest_text from
stats_mysql_query_digest;"

mysql -uadmin -padmin -h127.0.0.1 -P6032 \
  -e "select hostgroup,
              digest_text,
              count_star,
              from_unixtime(last_seen) as last_seen_time
       from stats_mysql_query_digest
       where digest_text like 'select%'
       order by last_seen desc
       limit 5;"

dnf -y remove proxysql
```

四十九、怎么收回权限

```
-- 收回 qianshan 用户所有库所有表的查询权限
revoke select on *.* from qianshan@'%';

-- 收回该用户全部所有权限
revoke all privileges on *.* from qianshan@'%';

-- 删除用户
drop user qianshan@'%';
```

五十、角色管理

```
mysql> CREATE ROLE 'qianshan_role';
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT ALL privileges ON *.* TO 'qianshan_role';
Query OK, 0 rows affected (0.00 sec)

mysql> create user qs001@'%' identified by 'Zhongguo@0755';
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT 'qianshan_role' TO 'qs001'@'%';
Query OK, 0 rows affected (0.00 sec)
```

数据库与数据库管理系统

1. 什么是数据库 数据库 =database=date+base=数据 +根源 (基本 /基础)
   数据库是存储在计算机系统内的，结构化的，共享的，可控制的数据集合。 常见的 excel
   表格，它有行，有列，是一有定的结构化，没有操作日志，恢复起来很困难。
   数据库：它可以存上千万行，上亿数据，建立索引之后，查询是很快的，有锁，有事务，可以多人使
   用，并且可以根 据日志进行灾难恢复。 数据库的三个基本特点： 永久存储：关机数据还在
   有组织：按固定格式（二维表）存放 可共享：多人可以同时使用，互不干扰。
2. 什么是数据库管理系统 数据库管理系统 =DBMS
   数据库管理系统是一种大型的软件，用于建立、使用和维护数据库。 DBMS的主要功能：

1、定义数据结构，规定表有几列，每列存什么类型的数据， DDL 2
、组织与存储，决定数据在磁盘上如何保存，如果存取，并提高效率。 3
、操作数据：增加，删除，修改，查询， DML 4、事务与运行管理：保证多人同时使用不出错，数据完整性 5

、建立与维护：初始数据输入，备份，恢复，性能监控

3. 数据库与数据库管理系统区别 数据库：类似为一个个具体的 excel 文件。举例：学生信息库，订单库，物流库 DBMS：类似为 EXCEL这个软件。举例： mysql，oracle，Sql server

常见的数据库系统

常见数据库排名

```
https://db-engines.com/en/ranking
```

oracle数据库： 197x年，美国人发明的，流行的版本， 11G，12C，19C。

mysql数据库： 瑞典的 MySQL AB公司开发， 1995年，在 2008被 SUN收购了， 2009，SUN公司被 Oracle 公司收购了。

```
Microsoft SQL Server
```

198x年， sybase公司发布 的 sql server

```
sql server 6.5 sql server 2000
```

sql server 2017版，版本支持 linux。

postgresql：美国人研发的，开源，它支持 windows和 linux下运行。并且的许可是 BSD。 gaussdb/opengauss，国内信创领域用得多。

```
mogodb
```

文档型数据库：不是关系型数据库，它是 NoSQL中的一种。

键值数据库： redis/memcached：

时序数据库： Prometheus、 InfluxDB

```
time host bytes_in bytes_out
```

MariDB与 Mysql区别

MariDB是一个开源关系型数据库管理系统（ RDBMS） 2009年， oracle公司收购了 sun。相当于它收购了 mysql。早期的 mysql开发人员，在 oracle收购之后，从 msyql

这里拷贝了数据，形成了一个分支。这个分支就是 maridb数据库。

```
https://mariadb.org/download/?t=mariadb
```

Mysql5.7与 8.x安装

mysql5.7与 mysql8.0主要区别 1、默认编码: mysql5.7 字符集 iatin1,mysql8的字符集是 utf8mdb4 2、身份验证插件: mysql5.7的 mysql_native_password，msyql8的 caching_sha2_password 3、性能: mysql8性能是 mysql5.7的 2倍。 4、授权: mysql5.7版本，grant all xxxx 一条指令就搞定了授权，mysql create user; grant all 2条指令 5、冷备，xtrabackup版本不一样

Mysql5.7安装

centos7.9环境 yum安装 第一步: 先增加 yum源

```
https://dev.mysql.com/downloads/repo/yum/
rpm -ivh mysql80-community-release-el7-6.noarch.rpm
```

第二步: 调整 mysql-server版本

```
yum repolist all|grep mysql #查看mysql可安装的版本
yum install -y yum-utils #安装yum工具箱
yum-config-manager --disable mysql80-community
yum-config-manager --enable mysql57-community
```

第三步:

```
yum install mysql-server #在线安装
```

第四步: 如报错，调整 mysql-community.repo仓库源，不让 rpm包进行 gpg检测

```
sed -i 's#file:///etc/pki/rpm-gpg/RPM-GPG-KEY-mysql-2022#https://repo.mysql.com/RPM-GPG-KEY-
mysql-2023#g' /etc/yum.repos.d/mysql-community.repo
sed -i 's#file:///etc/pki/rpm-gpg/RPM-GPG-KEY-mysql#https://repo.mysql.com/RPM-GPG-KEY-
mysql#g' /etc/yum.repos.d/mysql-community.repo
sed -i 's#RPM-GPG-KEY-mysql-2022#RPM-GPG-KEY-mysql-2023#g'
/etc/yum.repos.d/mysqlcommunity.repo
```

第四步: 启动服务

```
systemctl start mysqld
```

Mysql5.7二进制安装

```
#先从官网下载二进制包，上传到服务器
tar xf mysql-5.7.44-linux-glibc2.12-x86_64.tar.gz
#新建mysql用户
[root@c79-s01 opt]# groupadd mysql
[root@c79-s01 opt]# useradd -M -s /sbin/nologin -g mysql mysql
[root@c79-s01 opt]# mkdir -p /data/mysqldata
[root@c79-s01 opt]# chown -R mysql.mysql /data/mysqldata/
#创建mysql的配置文件
[root@c79-s01 opt]# more /etc/my.cnf
[client]
default-character-set=utf8mb4
[mysqld]
bind-address=0.0.0.0
character_set_server=utf8mb4
collation-server = utf8mb4_unicode_ci
init_connect='SET NAMES utf8mb4'
datadir=/data/mysqldata/
socket=/data/mysqldata/mysql.sock
symbolic-links=0
log-error=/data/mysqldata/mysqld.log
pid-file=/data/mysqldata/mysqld.pid
```

数据库的初始化

```
/usr/local/mysql-5.7.44/bin/mysqld --initialize --user=mysql --datadir=/data/mysqldata --
basedir=/usr/local/mysql-5.7.44
```

找到初始密码

```
more /data/mysqldata/mysqld.log
```

启动数据库

```
/usr/local/mysql-5.7.44/bin/mysqld_safe --defaults-file=/etc/my.cnf &
```

连接数据库

```
mysql -uroot -p -S /data/mysqldata/mysql.sock
mysql -uroot -p -h 127.0.0.1
```

设置开机自启

```
cp /usr/local/mysql-5.7.44/support-files/mysql.server /etc/init.d/mysqld
sed -i '46s/basedir=/basedir=\/usr\/local\/mysql-5.7.44/g' /etc/rc.d/init.d/mysqld
sed -i '47s/datadir=/datadir=\/data\/mysqldata/g' /etc/rc.d/init.d/mysqld
chmod +x /etc/init.d/mysqld
chkconfig -add mysqld
systemctl daemon-reload
systemctl start mysqld
systemctl enable mysqld
```

Mysql8.0安装

yum在线安装

centos7.9上安装

第一步: 先增加 yum源

```
https://dev.mysql.com/downloads/repo/yum/
rpm -ivh mysql80-community-release-el7-6.noarch.rpm
```

第二步: 修改 gpg检测

```
vi /etc/yum.
```

第三步: 安装

```
yum install msyql-server
```

rockylinux8.10上安装

```
dnf install mysql-server
systemctl start mysqld
mysql -uroot -p
```

RPM包安装

rockylinux8.10环境

官网下载 rpm包集合，并上传到服务器

```
tar xf mysql-8.0.45-1.el8.x86_64.rpm-bundle.tar
```

安装刚下线的 rpm包文件

```
dnf localinstall \
mysql-community-server-8.0.45-1.el8.x86_64.rpm \
mysql-community-client-8.0.45-1.el8.x86_64.rpm \
mysql-community-common-8.0.45-1.el8.x86_64.rpm \
mysql-community-icu-data-files-8.0.45-1.el8.x86_64.rpm \
mysql-community-client-plugins-8.0.45-1.el8.x86_64.rpm \
mysql-community-libs-8.0.45-1.el8.x86_64.rpm
```

启动服务

```
systemctl start mysqld
```

找初始密码，并登录测试

```
grep 'password' /var/log/mysqld.log
mysql -uroot -p
```

二进制安装

rockylinux8.10环境

```
groupadd mysql
useradd -M -s /sbin/nologin -g mysql mysql
#下载文件，注意glibc版本
cd /opt

lrzsz mysql-8.0.45-linux-glibc2.28-x86_64.tar.xz

tar -xf mysql-8.0.45-linux-glibc2.28-x86_64.tar.xz
mv mysql-8.0.45-linux-glibc2.28-x86_64 /usr/local/mysql-8.0.45
```

```
chown mysql.mysql -R /usr/local/mysql-8.0.45/

#设置数据目录权限
mkdir -p /data/mysqldata
chown -R mysql.mysql /data/mysqldata

#设置my.cnf配置文件
[client]
default-character-set=utf8mb4
[mysqld]
bind-address=0.0.0.0
character_set_server=utf8mb4
collation-server = utf8mb4_0900_ai_ci
init_connect='SET NAMES utf8mb4'
datadir=/data/mysqldata/
socket=/data/mysqldata/mysql.sock
symbolic-links=0
log-error=/data/mysqldata/mysqld.log
pid-file=/data/mysqldata/mysqld.pid

[mysql]
default-character-set=utf8mb4

#数据库初始化
/usr/local/mysql-8.0.45/bin/mysqld --defaults-file=/etc/my.cnf --initialize --user=mysql --
datadir=/data/mysqldata --basedir=/usr/local/mysql-8.0.45

#找初始密码
[root@alma88-01 soft]# grep 'password' /data/mysqldata/mysqld.log
```

启库

```
/usr/local/mysql-8.0.45/bin/mysqld_safe &
```

登录并修改密码与新建用户

```
[root@alma88-01 opt]# mysql -uroot -p'XXXXXXXXX' -P3306 -h 127.0.0.1
mysql> alter user 'root'@'localhost' identified by 'Zhongguo@0755';
mysql> create user 'root'@'%' identified by 'Zhongguo@0755';
mysql> grant all privileges on *.* to rootd@'%';
```

```
mysql> flush privileges;


#关库
/usr/local/mysql-8.0.45/bin/mysqladmin -uroot -p -S /data/mysqldata/mysql.sock shutdown
```

源码编译安装

rockylinux8.10环境

官网下载文件

```
#解压并编译
tar xf mysql-boost-8.0.45.tar.gz
cd mysql-8.0.45
mkdir build && cd build

cmake .. \
-DCMAKE_INSTALL_PREFIX=/usr/local/mysql-8.0.45 \
-DDEFAULT_CHARSET=utf8mb4 \
-DDEFAULT_COLLATION=utf8mb4_unicode_ci \
-DENABLED_LOCAL_INFILE=ON \
-DWITH_SSL=system \
-DMYSQL_DATADIR=/data/mysqldata \
-DMYSQL_TCP_PORT=3306 \
-DDOWNLOAD_BOOST=0 \
-DWITH_BOOST=../boost

make -j8 #8代表CPU核心
make install
```

cmake时出错，缺什么补什么

```
yum install cmake -y
yum install gcc-toolset-12-gcc gcc-toolset-12-gcc-c++ gcc-toolset-12-binutils gcc-toolset-
12-annobin-annocheck gcc-toolset-12-annobin-plugin-gcc
yum install -y elfutils-libelf-devel binutils
yum install -y openssl-devel
yum install -y ncurses-devel
yum install -y libtirpc-devel
tar xf rpcsvc-proto-1.4.4.tar.xz
```

```
cd rpcsvc-proto-1.4.4
./configure && make && make install
```

创建用户及目录授权

```
groupadd mysql
useradd -M -s /sbin/nologin -g mysql mysql
#下载文件，注意glibc版本

mkdir -p /data/mysqldata
chown -R mysql.mysql /data/mysqldata
```

设置配置文件

```
[root@c79-s01 opt]# more /etc/my.cnf
[client]
default-character-set=utf8mb4

[mysqld]
bind-address=0.0.0.0
character_set_server=utf8mb4
collation-server = utf8mb4_unicode_ci
init_connect='SET NAMES utf8mb4'

datadir=/data/mysqldata/
socket=/data/mysqldata/mysql.sock

symbolic-links=0

log-error=/data/mysqldata/mysqld.log
pid-file=/data/mysqldata/mysqld.pid
```

初始化

```
/usr/local/mysql-8.0.45/bin/mysqld --initialize --user=mysql --datadir=/data/mysqldata --
basedir=/usr/local/mysql-8.0.45
```

起库

```
/usr/local/mysql-8.0.45/bin/mysqld_safe --defaults-file=/etc/my.cnf &
```

或

```
/usr/local/mysql-8.0.45/bin/mysqld_safe &
```

打包，可迁移到目标机直接使用

```
cd /usr/local/
tar -zcf mysql-8.0.45.rocky8.cmake.tar.gz /usr/local/mysql-8.0.45
#保存
sz /usr/local/mysql-8.0.45.rocky8.cmake.tar.gz
```

数据库客户端安装

```
navicat


#先建立用户，用于远程连接
mysql> create user 'qsadmin'@'%' identified by 'Zhongguo@0755';
Query OK, 0 rows affected (0.02 sec)
mysql> grant all privileges on *.* to 'qsadmin'@'%';
Query OK, 0 rows affected (0.01 sec)
mysql> flush privileges;
Query OK, 0 rows affected (0.01 sec)
mysql>

sqlyog
www.webyog.com
```

免费版本: https://github.com/webyog/sqlyog-community/wiki/Downloads 支持 mysql

```
dbeaver
https://dbeaver.io
```

支持 mysql, 支持 pg, 支持 oracle, 支持 sql server等

PL/SQL Developer和 TOAD for Oracle，针对 oracle数据库管理的软件

开关数据库及连接

yum&&rpm安装类

数据库起库

```
systemctl start mysqld
```

第一次执行的时候，它做了一个初始化的事，起库的事（找 /etc/my.cnf）。 3306: 传统 mysql 协议的监听端口 33060: 支持 x协议的监听端口

数据库关库:

```
systemctl stop mysqld
```

背后 kill -15 ${mysqld_pid}

数据库的本地连接:

```
mysql -uroot -p
mysql -uroot -p -S /var/lib/mysql/mysql.sock
mysql -uroot -p -h 127.0.0.1
```

手工安装类 二进制安装 源码编译安装

```
#数据库起库
/usr/local/mysql-8.0.45/bin/mysqld_safe --defaults-file=/etc/my3307.cnf &
#数据库关库:
/usr/local/mysql-8.0.45/bin/mysqladmin -uroot -p -S /data/mysqldata/mysql.sock shutdown
#数据库本地连接:
/usr/local/mysql-8.0.45/bin/mysql -uroot -p -S /data/mysqldata/mysql.sock
```

数据库授权与撤销

DCL语句概述 DCL=数据控制语言，它是属于 SQL语言中的一种， DDL,DML,TCL。 DCL 专门用于定义数据库的安全策略和访问权限。

grant：授权 revoke: 撤销授权

```
create role/drop role
grant role/revoke role
```

全局级: .(所有的库表)，权限存在 mysql.user表中。一般是分配给 DBA。数据库级: db_name.*，权限存在 mysql.db表中。最常见的。 表级: db_name.table_name，权限存在 mysql.tables_priv表中。 列级: col1 ， mysql.权限存在 columns_priv表中。 存储过程 /函数级: 权限存在 procs_priv表中。

授权原则 1、最小权限。 2、创建用户（帐户）的时候，最好是带登录主机或 IP段。 3、密码要有一定的复杂度。大小写加一些特殊字符，长度大于 8位，不要用弱口令。 4、定期清理用户，不需要的用户及时收回权限或删除用户。

怎么授权

```
#mysql8.x
create user username@'IP段' identified by "密码";
grant all on *.* to username@'IP段';
flush privileges;
```

IP段 =%，允许任意地方。 IP段 =192.168.1.1，允许 1.1这个主机。 IP段 =192.168.1.%，允许 192.168.1 网段。

```
#mysql 5.7
grant all on *.* to username@'IP段' identified by "密码";

-- 修改root本地用户密码
alter user root@'localhost' identified by 'Zhongguo@0755';

-- 查询所有用户、登录主机、加密密码、密码认证插件
select user,host,authentication_string,plugin from mysql.user;

-- 查看 qianshan 用户任意主机登录权限，\G 竖行展示结果
show grants for qianshan@'%'\G;

-- 直接修改mysql系统表，将用户名qianshan改为qs
update mysql.user set user='qs' where user='qianshan';
-- 修改系统表后必须刷新权限方可生效
flush privileges;
```