--mysql-day1

数据库 （组织和管理数据的软件）

qq

微信 账号 密码

游戏 账号 密码 皮肤 装备 人物属性

mysql 开源占有率第一

mariadb MySQL衍生版本，技术和MySQL相同

oracle 商业占有率第一

db2 IBM

sybase

sqlserver mssql

access

sqlite

nosql

virtual fox

postgresql

informix

概述讲义：

关系数据库，是建立在关系模型基础上的数据库，借助于集合代数等数学概念和方法来处理数据库中的数据。现实世界中的各种实体以及实体之间的各种联系均用关系模型来表示。关系模型是由埃德加·科德于1970年首先提出的，并配合“科德十二定律”。现如今虽然对此模型有一些批评意见，但它还是数据存储的传统标准。标准数据查询语言SQL就是一种基于关系数据库的语言，这种语言执行对关系数据库中数据的检索和操作。

关系模型就是指二维表格模型,因而一个关系型数据库就是由二维表及其之间的联系组成的一个数据组织。

关系型数据库的优势：

1.保持数据的一致性（事务处理）

2.由于以标准化为前提，数据更新的开销很小（相同的字段基本上都只有一处）

3.可以进行Join等复杂查询

其中能够保持数据的一致性是关系型数据库的最大优势。

关系型数据库，就是所有的表都可以进行关联起来并且数据拥有一致性

姓名 性别 年龄 部门 职位 入职时间 工资

张三 男 30 IT 经理 2005 20000

李四 女 35 sale 经理 2006 30000

数据库连接技术

ODBC open database connector 微软提出的

JDBC java database connector sun提出的

mysql特点

性能 它把最大的精力用于提高性能

可编程支持性 oracle PL/SQL sqlserver T-SQL postgresql pl/PGSQL

可靠性 web2.0标准网站

易用性 简单易用

MySQL有多种存储引擎，每种存储引擎有各自的优缺点：

MyISAM、InnoDB、MERGE、MEMORY(HEAP)、BDB(BerkeleyDB)、EXAMPLE、FEDERATED、ARCHIVE、CSV、BLACKHOLE

但常用的引擎只有MyISAM 和 InnoDB 两种

1、MyISAM管理非事务表。它提供高速存储和检索，以及全文搜索能力。MyISAM在所有MySQL配置里被支持，它是默认的存储引擎，除非你配置MySQL默认使用另外一个引擎。

2、InnoDB引擎提供事务安全表，默认被包括在所有MySQL 5.1二进制分发版里，你可以按照喜好通过配置MySQL来允许或禁止任一引擎。

mysql 5.5版本以前，默认是MyISAM引擎，5.5版本以后默认都是InnoDB引擎

mysql（mariadb）数据库的三种安装方式

1、rpm版 --安装简单，所有配置文件和文件目录都已经配置好，安装就能用

2、二进制 --相当于windows的绿色版(硬盘免安装版)

3、源码版 --安装复杂，所有配置文件都是原始状态未配置，需要人工配置。

下载地址：

# wget http://mirrors.163.com/.help/CentOS6-Base-163.repo

1、rpm版 --使用yum源安装，CentOS 6 自带的mysql是5.1版本

# yum install mariadb\* -y

# rpm -qa mariadb\* 安装完成后查询所有的包

mariadb-5.5.52-1.el7.x86\_64

mariadb-embedded-5.5.52-1.el7.x86\_64

mariadb-test-5.5.52-1.el7.x86\_64

mariadb-libs-5.5.52-1.el7.x86\_64

mariadb-server-5.5.52-1.el7.x86\_64

mariadb-embedded-devel-5.5.52-1.el7.x86\_64

mariadb-bench-5.5.52-1.el7.x86\_64

mariadb-devel-5.5.52-1.el7.x86\_64

--可使用rpm -qi 去查询每个包的详细说明

# id mysql

uid=27(mysql) gid=27(mysql) groups=27(mysql)

# systemctl restart mariadb --直接启动

mysql5.7以后重启需要查看初始密码

cat /var/log/mysqld.log |grep password

第一次启动数据库，要进行安全设置工作，因为数据库没有密码

# /usr/bin/mysql\_secure\_installation

Enter current password for root (enter for none): --输入数据库管理员root的密码，没有密码直接回车

Set root password? [Y/n] y --设置root的密码

New password:

Re-enter new password: --输入你的密码

Remove anonymous users? [Y/n] y --删除匿名用户

Disallow root login remotely? [Y/n] n --是否允许root用户远程登录

Remove test database and access to it? [Y/n] --删除测试库

Reload privilege tables now? [Y/n] --让上面的操作生效

# netstat -ntlup |grep 3306

tcp 0 0 0.0.0.0:3306 0.0.0.0:\* LISTEN 3720/mysqld

# lsof -i:3306

COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME

mysqld 20180 mysql 14u IPv4 44654 0t0 TCP \*:mysql (LISTEN)

登陆数据库：

# mysql -u root -p123456 --p参数后没有空格 u=user p=password

MariaDB [(none)]> show variables like '%engine%'; --查看数据库的引擎

+---------------------------+--------+

| Variable\_name | Value |

+---------------------------+--------+

| default\_storage\_engine | InnoDB |

| engine\_condition\_pushdown | OFF |

| storage\_engine | InnoDB |

+---------------------------+--------+

3 rows in set (0.00 sec))

MariaDB [(none)]> show variables like 'datadir'; --查看数据库的数据目录

+---------------+-----------------+

| Variable\_name | Value |

+---------------+-----------------+

| datadir | /var/lib/mysql/ |

+---------------+-----------------+

1 row in set (0.00 sec)

MariaDB [(none)]> show databases; --查看当前所有数据库 或者 show schemas;

+--------------------+

| Database |

+--------------------+

| information\_schema |

| mysql |

| performance\_schema |

| test |

+--------------------+

4 rows in set (0.01 sec)

MariaDB [(none)]> use information\_schema; --使用这个库，或者叫进入这个库

MariaDB [(none)]> select database(); --查看现在use进入了哪个库,类似pwd

+--------------------+

| database() |

+--------------------+

| information\_schema |

+--------------------+

一个数据库（show databases;）对应一个目录/var/lib/mysql

数据库里存放表格 对引目录里的文件

MariaDB [(none)]> show tables; --查看当前库所拥有的表 --对应相应的目录下的文件

MariaDB [(none)]> desc 表名; --描述表的结构（元数据），description

MariaDB [(none)]> select \* from 表名; --查看表的内容

MariaDB [(none)]> select \* from 表名\G

MariaDB [(none)]> select user(); --查看当前登录用户

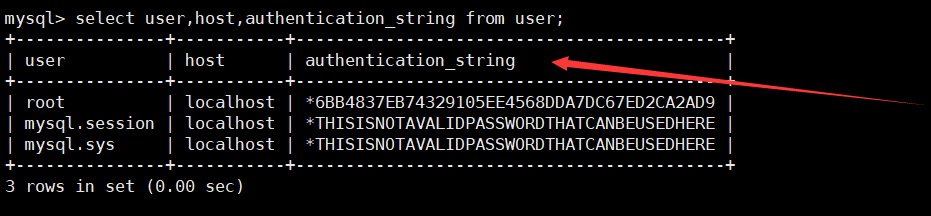
+----------------+

| user() |

+----------------+

| root@localhost | --此用户不是系统的root用户，他们的密码无关

MariaDB [(none)]> select user,host,password from mysql.user; --查看当前数据库所有用户和密码，以及各用户授权信息



# ls /var/lib/mysql/ --rpm数据文件

ibdata1 --innodb引擎数据文件，information\_schema一些信息在此

ib\_logfile1

ib\_logfile0

mysql.sock --用于本地连接;是unix沿用下来的一种连接机制。远程连接用tcp/ip，本地也可以用tcp/ip。socket连接速度相对快一点，tcp/ip相对稳定点

mysql --对应mysql库

test --对应test库

# rm /var/lib/mysql/mysql.sock

rm: remove socket `/var/lib/mysql/mysql.sock'? y

# mysql --连接报错，找不到socket文件

ERROR 2002 (HY000): Can't connect to local MySQL server through socket '/var/lib/mysql/mysql.sock' (2)

# mysql -h 127.0.0.1 --但是通过tcp/ip连接

--解决方法，重启mariadb

--socket和tcp/ip连接的区别:

socket是unix的一种连接机制，用于本地程序连接

tcp/ip用于远程程序连接，也可以用于本地连接(当使用IP为本地IP时)

如果是本地连接.socket效果要好，从优化角度讲，本地最好用socket(可以对其备份)

---------------------------------------------------------

修改密码三种方法：

第一种：

# mysqladmin -u root password "123" --把root用户登录密码改为123，这种方式只适用于没有密码的情况

# mysql

ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: NO)

# mysql -p123 --参数与密码之间没有空格

# mysql -p --不想明文登录就使用这种方式

Enter password:

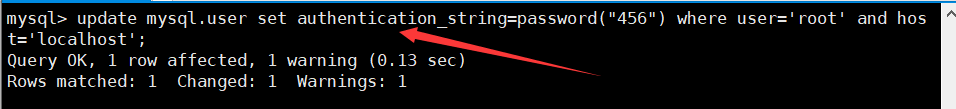
如果已经有密码为123，想要修改密码：

# mysqladmin -u root password '456' -p123 --如果之前有密码，要修改密码，需要提供之前的密码 ，8.0 后版本也可以

# mysqladmin -u root password 'Mysql137#$' -p123

第二种：

MariaDB [(none)]> update mysql.user set password=password("456") where user="root" and host="localhost"; --使用sql语句在数据库内部直接修改用户密码表，password()是一个密码加密函数



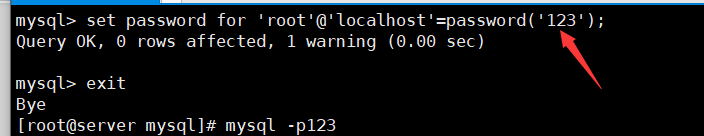
上图是mysql5.7源码版

MariaDB [(none)]> update mysql.user set authentication\_string=password(“新密码”) where User=”root” and Host=”localhost”;  --mysql5.7修改密码的命令 ，mysql8.0后不可以

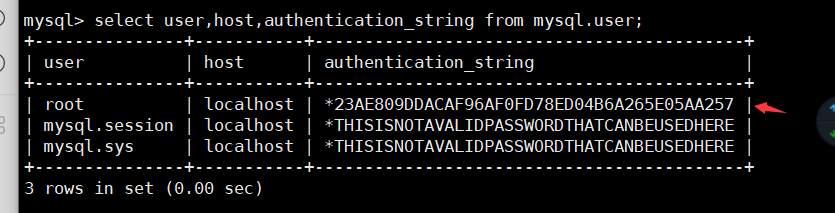
MariaDB [(none)]> flush privileges; --修改过密码后都要记得刷新权限表，不刷新就不会生效

第三种：

MariaDB [(none)]> set password for 'root'@'localhost'=password('123'); --使用此操作语句也可以修改密码，修改后不需要刷新权限表

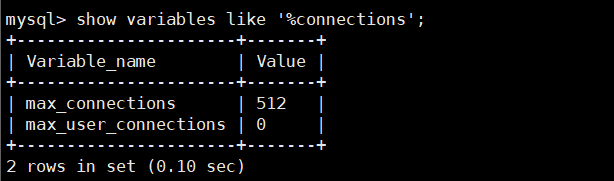


MariaDB [(none)]> select user,host,password from mysql.user;



--修改参数例子

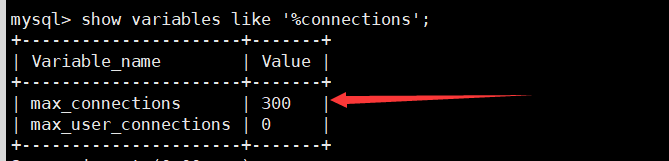
MariaDB [(none)]> show variables like "%connections"; --like模糊查询，%是通配符，表示n个字符



MariaDB [(none)]> set global max\_connections=300; --把最大允许的链接数改为300

Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> show variables like "%connections";



--再次查看变成300

--但注意上面的修改重启又回变回到151

========================================================================

mysql的配置文件，rpm 版默认为/etc/my.cnf

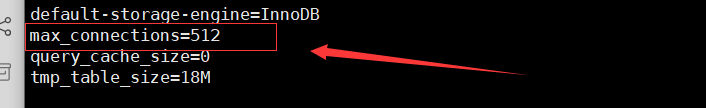
[mysqld] --用中括号括起来的叫参数组，用来针对不同的工具设定参数

datadir=/var/lib/mysql --数据文件存放的目录

socket=/var/lib/mysql/mysql.sock --socket文件是用于本地连接mysql数据库的接口文件，远程连接的话就要通过TCP／IP协议

user=mysql --管理mysql的系统用户

max\_connections=300 --加到这里，重启后，这个值就不会变回151了



[mysqld\_safe]

log-error=/var/log/mysqld.log --错误日志文件

pid-file=/var/run/mysqld/mysqld.pid --pid文件

--注意，pid文件和socket文件是服务启动成功后才会有的

在/etc/my.cnf 后面加上一段

[client] --针对客户端登录时的参数组

password=123 --客户端登录时直接使用123这个密码了 ，不用重新启动也会生效

# mysql --再次登录就不需要密码了

忘记密码如何修改

1，可以在配置文件里加上

# vim /etc/my.cnf

skip-grant-tables --注意写到[mysqld]参数组下，表示跳过授权表，登录时，不验证密码

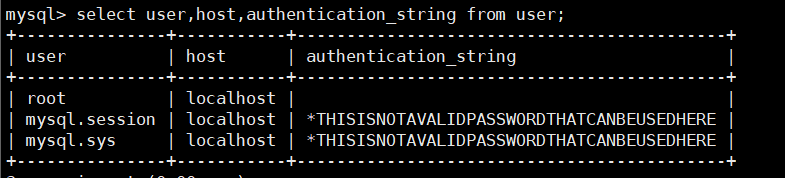
2，systemctl restart mariadb 这时候需要重启

mysql --登录就不需要密码，进去改密码,改完后,直接flush privileges就可以使用新密码来登录了

MySQL8.0版本，进去把mysql 库 user表密码字段设置为空，

mysql> use mysql

mysql>update user set authentication\_string='' where user='root' and host='localhost'; 然后退出



把免密码参数注释掉，要不然等下设置不了新密码

重启mysqld

3，改完后记得去掉skip-grant-tables重新reload

MariaDB [(none)]> update mysql.user set password=password("123") where user="root" and host="localhost"; mysql 8.0以后不可以

mysql>alter user 'root'@'localhost' identified by 'Mysql137#$'; 密码不能过于简单， mysql 8.0 以后的密码设置

mysql> flush privileges; 需要刷新

4，再使用新的密码登录

关闭mysql的方法：

# pkill mysql --实际环境不建议用

# systemctl stop mariadb

# mysqladmin shutdown -p123

-----------------------

将数据关闭，使用以下命令启动数据库

# mysqld\_safe --defaults-file=/etc/my.cnf --skip-grant-tables &

登录时就不需要密码了

mysql

==========================================================================

mysql 数据目录迁移：

在现实工作中，数据库目录迁移是很常见的事，有的是因为原有磁盘空间不足需要迁移，有的是因为要做主从再迁移，有的可能是为了磁盘性能问题而迁移。

这里教大家数据迁移。

MariaDB [(none)]> create database a; ---创建数据库

MariaDB [(none)]> show databases; ---查询数据库列表

MariaDB [(none)]> use a; ---插入a数据库

MariaDB [(none)]> create table a1(id int(1),name char(10)); ---创建a1表有id和name字段，int(1)的int代表是整数类型，(1)代表限制一位数；char(10)的char代表是字符串类型，(10)代表限制10个字符串。

MariaDB [(none)]> insert into a1(id,name) values(1,"测试库");

MariaDB [(none)]> insert into a1(id,name) values(2,"mysql库"); --往a1表里插入数据。

MariaDB [(none)]> select \* from a1; ---查询a1表数据

1、登陆数据库

[root@web1 mysql]# mysql -uroot -p123456

2、查看数据库的数据目录

MariaDB [(none)]> show variables like 'datadir';

+---------------+--------------+

| Variable\_name | Value |

+---------------+--------------+

| datadir |/var/lib/mysql|

+---------------+--------------+

1 row in set (0.00 sec)

3、退出并将mysql数据库停止

MariaDB [(none)]> exit

Bye

[root@web1 mysql]

[root@web1 mysql]# systemctl stop mariadb

Stopping mysqld: [ OK ]

4、将新建新的数据目录,并授权

[root@web1 mysql] mkdir /data

5、将第2步查出来的mysql目录复制到第4步新建的data目录下 也可以cp -a 复制，

[root@web1 mysql] cp -r /var/lib/mysql /data/

[root@web1 mysql] chown -R mysql.mysql /data

6、修改mariadb配置文件/etc/my.cnf

# vim /etc/my.cnf

[mysqld]

datadir=/data/ --修改成新的数据目录

socket=/data/mysql.sock --修改到新的数据目录

port=3306

user=mysql

[mysqld\_safe]

log-error=/var/log/mariadb/mariadb.log

pid-file=/var/run/mariadb/mariadb.pid

[client] --新增

socket=/data/mysql.sock --增加这句

7、启动数据库

[root@web1 mysql]# systemctl restart mariadb

Stopping mysqld: [ OK ]

Starting mysqld: [ OK ]

8、登陆验证

[root@web1 ~]# mysql -uroot -p123456

MariaDB [(none)]>

MariaDB [(none)]> show databases;

+--------------------+

| Database |

+--------------------+

| information\_schema |

| a |

| mysql |

| performance\_schema |

| test |

+--------------------+

5 rows in set (0.00 sec)

MariaDB [(none)]> show variables like 'datadir'; --数据库目录已经是新的目录了

+---------------+--------------+

| Variable\_name | Value |

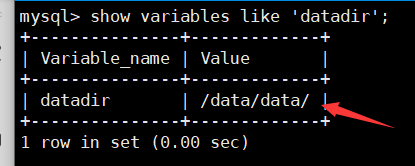
+---------------+--------------+

| datadir | /data/ |

+---------------+--------------+

1 row in set (0.00 sec)

MariaDB [(none)]>



到此迁移完成。