**DBA1-day01**

**搭建数据库服务器**

**在主机50client上部署mysql数据库服务器**

**装包**

**修改配置**

**启动服务**

**查看服务状态**

**客户端访问服务**

**在主机50上部署mysql数据库服务**

**删除系统自带的mariadb mysql数据库软件**

**[root@client ~]#rpm -qa | grep -i mariadb**

**[root@client ~]#systemctl stop mariadb**

**[root@client ~]#rpm -e --nodeps mariadb-server mariadb**

**[root@client ~]#rm -rf /etc/my.cnf**

**[root@client ~]#rm -rf /var/lib/mysql**

**[root@client ~]# tar -xf mysql-5.7.17.tar -C /mnt //解压压缩包**

**[root@client ~]# rpm -Uvh /mnt/mysql-community-\*.rpm**

**//安装mysql服务软件包**

**[root@client ~]# yum list | grep -i json | grep -i perl**

**//根据上一步的提示查询依赖包的包名**

**[root@client ~]# yum -y install perl-JSON //根据上一的查询结果安装依赖包**

**[root@client ~]# rpm -Uvh /mnt/mysql-community-\*.rpm**

**//再次安装mysql服务软件包**

**[root@client ~]# systemctl start mysqld**

**[root@client ~]# systemctl enable mysqld**

**//启动mysql服务并设置为开机自启**

**[root@client ~]# systemctl status mysqld //查看服务状态**

**[root@client ~]# ss -natpul | grep 3306 //查看mysql服务端口**

**[root@client ~]# ps -C mysqld //查看进程**

**与数据库服务相关的信息**

**[root@client ~]# vim /etc/my.cnf //服务主配置文件**

**datadir=/var/lib/mysql //数据库服务的目录**

**socket=/var/lib/mysql/mysql.sock //套接字目录（只有当服务启动之后才有）**

**log-error=/var/log/mysqld.log //服务的错误日志文件**

**pid-file=/var/run/mysqld/mysqld.pid //服务的pid号文件**

**[root@client ~]# ps aux | grep mysqld //查看mysql服务的进程号**

**命令选项**

**-h, --host=name Connect to host. //要连接的主机**

**客户端连接数据库服务存储数据**

**\*默认只允许数据服务器自己访问自己**

**登陆mysql**

**[root@client ~]# grep password /var/log/mysqld.log**

**2018-10-15T02:21:59.477549Z 1 [Note] A temporary password is generated for root@localhost: 9qksi9M3yS,H**

**//查询数据库管理员首次连接数据库的初始密码为红色部分**

**[root@client ~]# mysql -hlocalhost -uroot -p'9qksi9M3yS,H'**

**//利用初始密码连接本机的数据库服务**

**mysql> alter user root@localhost identified by "123asd...A";**

**//首次进入须修改密码**

**mysql>exit或quit**

**[root@client ~]# mysql -hlocalhost -uroot -p123asd...A**

**//修改密码后利用新密码重新登陆**

**SQL命令使用规则（结构化查询语言）**

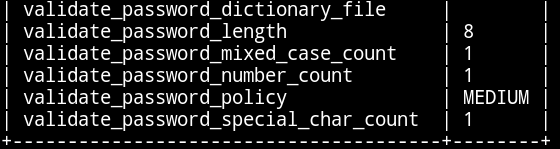
1. **命令结束必须以分号结尾**
2. **SQL命令不区分大小写，不能使用Tab键自动补齐**

**修改数据库服务密码策略**

1. **命令行临时修改**

**[root@client ~]# mysql -hlocalhost -uroot -p123asd...A**

**mysql> show variables like "%password%";**

****

**| validate\_password\_policy | MEDIUM | //默认**

**mysql> set global validate\_password\_policy=0; //降低密码复杂度策略**

**mysql> set global validate\_password\_length=6; //修改密码长度**

**mysql> alter user root@localhost identified by '123456';**

**//根据修改之后的密码策略修改密码**

**mysql>exit或quit**

**[root@client ~]# mysql -hlocalhost -uroot -p123456**

**//利用修改策略之后设置的密码进行登陆**

1. **永久修改（修改配置文件，保证服务重启之后密码策略长久有效）**

**[root@client ~]# vim /etc/my.cnf**

**[mysqld]**

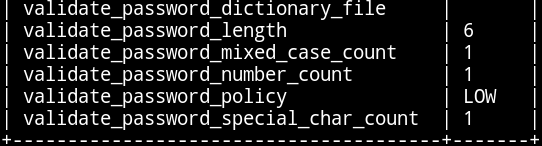
**validate\_password\_policy=0**

**validate\_password\_length=6**

**[root@client ~]# systemctl restart mysqld //重起服务**

**[root@client ~]# mysql -hlocalhost -uroot -p123456 //登陆查看**

**mysql> show variables like '%password%'; 查看之前的配置是否有效**

****

**| validate\_password\_policy | LOW |**

**数据库服务的基本使用：**

**客户端把数据存储到数据库服务器上的步骤**

1. **客户端连接数据库服务器**
2. **创建数据库（文件夹）**
3. **创建表（文件） 创建 查看 删除**
4. **存储数据（select insert update delete）**
5. **断开连接**

**• 数据库的命名规则**

**– 可以使用数字 / 字母 / 下划线,但不能纯数字**

**– 区分大小写,具有唯一性**

**– 不可使用指令关键字、特殊字符**

**记录管理命令**

**• 记录 类似于文件里的行**

**– Select \* from 表名 ; // 查看表记录**

**– Insert into 表名 values( 值列表 );// 插入表记录**

**– Update 表名 set 字段 = 值 ; // 修改表记录**

**– Delete from 表名 ; // 删除表记录**

**表 类似于系统的文件**

**– Desc 表名 ; // 查看表结构**

**– Select \* from 表名; // 查看表记录**

**– Drop table 表名 ; // 删除表**

1. **客户端连接数据库服务器**

**[root@client ~]# mysql -hlocalhost -uroot -p123456**

1. **创建数据库（文件夹） 创建 查看 切换 删除 更改**

**库管理命令**

**mysql> show databases; //查看当前所有的库**

**mysql> select database(); // 显示当前所在的库**

**mysql> create database gamedb; //创建新库**

**mysql> use gamedb; //切换库**

**mysql> show tables; //显示库中的表**

**mysql> drop database gamedb; //删除库**

1. **创建表（先建库——>再建表）**

**mysql> create database teadb;**

**mysql> use teadb;**

**mysql> create table teadb.stuinfo(**

**-> name char(20),**

**-> age int,**

**-> homeaddr char(50)**

**-> );**

**mysql> show tables;**

**mysql> select \* from teadb.stuinfo;**

**给表中插入数据（表管理）**

**mysql> insert into teadb.stuinfo //插入数据**

**-> values ("bob",19,"shengzhen"),**

**-> ("tom",21,"guangzhou");**

**mysql> select \* from teadb.stuinfo; //查看表中数据**

**修改**

**mysql> update teadb.stuinfo set age=35; //修改整列**

**mysql> update teadb.stuinfo set age=18 where name="bob";**

**//指定修改某一行的某一列**

**删除表**

**mysql> delete from teadb.stuinfo where name="bob"**

**//删除表中的数据**

**mysql> delete from teadb.stuinfo; //删除所有表记录**

**mysql> desc teadb.stuinfo; //查看表结构**

**mysql> drop tables teadb.stuinfo; //删除表**

**创建可以输入中文记录的表**

**mysql> create table 学生表(姓名 char(20),性别 char(6))DEFAULT CHARSET=utf8;**

**mysql> insert into 学生表 values("李斯","男"),("李华","女");**

**//给表中写入数据**

**MySQL数据类型**

**常见的信息种类**

**• 数值型:体重、身高、成绩、工资**

**数值类型：**

****

**整数型**

**使用 UNSIGNED 修饰时,对应的字段只保存正数**

**– 数值不够指定宽度时,在左边填空格补位**

**– 宽度仅是显示宽度,存数值的大小由类型决定**

**– 使用关键字 ZEROFILL 时,填 0 代替空格补位**

**– 数值超出范围时,报错。**

**mysql> create table t9( name char(3), lever int(3) zerofill );**

**mysql> insert into t9 values("tom",9);**

**mysql> insert into t9 values("lucy",20);//会失败，因name最长3位**

**mysql> select \* from t9;**

**+------+-------+**

**| name | lever |**

**+------+-------+**

**| tom | 009 |**

**| jim | 020 |**

**+------+-------+**

**浮点型**

**定义格式: float( 总宽度 , 小数位数 )**

**– 当字段值与类型不匹配时,字段值作为 0 处理**

**– 数值超出范围时,仅保存最大 / 最小值**

**mysql> create table t3(**

**-> age tinyint unsigned,**

**-> pay float(7,2)**

**-> );**

**mysql> insert into t3 values(18,38000.88);**

**mysql> select \* from t3;**

**+------+----------+**

**| age | pay |**

**+------+----------+**

**| 18 | 38000.88 |**

**+------+----------+**

**mysql> DESC t3;**

**+-------+---------------------+------+-----+---------+-------+**

**| Field | Type | Null | Key | Default | Extra |**

**+-------+---------------------+------+-----+---------+-------+**

**| age | tinyint(3) unsigned | YES | | NULL | |**

**| pay | float(7,2) | YES | | NULL | |**

**+-------+---------------------+------+-----+---------+-------+**

**字符型:姓名、工作单位、通信住址**

**• 定长: char( 字符数 )**

**– 最大长度 255 字符**

**– 不够指定字符数时在右边用空格补齐**

**– 字符数断超出时,无法写入数据。**

**• 变长: varchar( 字符数 )**

**– 按数据实际大小分配存储空间**

**– 字符数断超出时,无法写入数据。**

**• 大文本类型: text/blob**

**– 字符数大于 65535 存储时使用**

**mysql> create table t4(name char(4),homedir varchar(6));**

**mysql> insert into t4 values("jack","usa");**

**mysql> select \* from t4;**

**+------+---------+**

**| name | homedir |**

**+------+---------+**

**| jack | usa |**

**+------+---------+**

**枚举型:兴趣爱好、性别**

**• 从给定值集合中选择单个值, ENUM**

**– 定义格式: enum( 值 1, 值 2, 值 N)**

**• 从给定值集合中选择一个或多个值, SET**

**– 定义格式: set( 值 1, 值 2, 值 N)**

**mysql> create table tea( name varchar(10),likes set("eat","sleep","game","film"), sex enum("boy","girl","no"));**

**mysql> insert into tea values("bob","eat,sleep","boy");**

**mysql> select \* from tea;**

**+------+-----------+------+**

**| name | likes | sex |**

**+------+-----------+------+a**

**| bob | eat,sleep | boy |**

**+------+-----------+------+**

**日期时间型:出生日期、注册时间**

**日期时间类型：**

**• 日期时间, DATETIME**

**– 占用 8 个字节**

**– 范围: 1000-01-01 00:00:00.000000**

**~ 9999-12-31 23:59:59.999999**

**• 日期时间, TIMESTAMP**

**– 占用 4 个字节**

**– 范围: 1970-01-01 00:00:00.000000**

**~ 2038-01-19 03:14:07.999999**

**当为给TIMESTAMP字段赋值时，自动以当前系统时间赋值，而DATETIME字段默认赋值为NULL**

**mysql> create table t7(**

**-> meetting datetime,**

**-> oarty timestamp**

**-> );**

**mysql> insert into t7 values(now(),now());**

**mysql> select \* from t7;**

**+---------------------+---------------------+**

**| meetting | oarty |**

**+---------------------+---------------------+**

**| 2018-10-16 09:31:32 | 2018-10-16 09:31:32 |**

**+---------------------+---------------------+**

**mysql> insert into t7(meetting) values(now());**

**mysql> select \* from t7;**

**+---------------------+---------------------+**

**| meetting | oarty |**

**+---------------------+---------------------+**

**| 2018-10-16 09:31:32 | 2018-10-16 09:31:32 |**

**| 2018-10-16 09:32:21 | 2018-10-16 09:32:21 |**

**+---------------------+---------------------+**

**mysql> insert into t7(oarty) values(20181023223030);**

**mysql> select \* from t7;**

**+---------------------+---------------------+**

**| meetting | oarty |**

**+---------------------+---------------------+**

**| 2018-10-16 09:31:32 | 2018-10-16 09:31:32 |**

**| 2018-10-16 09:32:21 | 2018-10-16 09:32:21 |**

**| NULL | 2018-10-23 22:30:30 |**

**+---------------------+---------------------+**

**• 日期, DATE**

**– 占用 4 个字节**

**– 范围: 0001-01-01 ~ 9999-12-31**

**• 年份, YEAR**

**– 占用 1 个字节**

**– 范围: 1901~2155**

**• 时间, TIME**

**– 占用 3 个字节**

**– 格式: HH:MM:SS**

**mysql> create table t5( name char(10), birthday date, start year, uptime time, party datetime);**

**mysql> desc t5;**

**+----------+----------+------+-----+---------+-------+**

**| Field | Type | Null | Key | Default | Extra |**

**+----------+----------+------+-----+---------+-------+**

**| name | char(10) | YES | | NULL | |**

**| birthday | date | YES | | NULL | |**

**| start | year(4) | YES | | NULL | |**

**| uptime | time | YES | | NULL | |**

**| party | datetime | YES | | NULL | |**

**+----------+----------+------+-----+---------+-------+**

**mysql>insert into t5 values("bob",20181120,1990,083000,**

**20181224203018);**

**mysql> select \* from t5 ;**

**+------+------------+-------+----------+---------------------+**

**| name | birthday | start | uptime | party |**

**+------+------------+-------+----------+---------------------+**

**| bob | 2018-11-20 | 1990 | 08:30:00 | 2018-12-24 20:30:18 |**

**+------+------------+-------+----------+---------------------+**

**• 关于日期时间字段**

**– 当未给 TIMESTAMP 字段赋值时,自动以当前系统时**

**间赋值,而 DATETIME 字段默认赋值为 NULL**

**• YEAR 年份的处理**

**– 默认用 4 位数字表示**

**– 当只用 2 位数字赋值时, 01~69 视为 2000~2069 ,**

**而 70~99 视为 1970~1999**

**mysql> insert into t5(name,start) values("lucy",77);**

**mysql> select \* from t5;**

**+------+------------+-------+----------+---------------------+**

**| name | birthday | start | uptime | party |**

**+------+------------+-------+----------+---------------------+**

**| bob | 2018-11-20 | 1990 | 08:30:00 | 2018-12-24 20:30:18 |**

**| tom | 2018-11-20 | 2019 | 08:30:00 | 2018-12-24 20:30:18 |**

**| lucy | 2018-10-16 | 1990 | 09:20:12 | 2018-10-16 09:20:12 |**

**| lucy | NULL | 1977 | NULL | NULL |**

**+------+------------+-------+----------+---------------------+**

**mysql> insert into t5(name,start) values("tom",50);**

**mysql> select \* from t5;**

**+------+------------+-------+----------+---------------------+**

**| name | birthday | start | uptime | party |**

**+------+------------+-------+----------+---------------------+**

**| bob | 2018-11-20 | 1990 | 08:30:00 | 2018-12-24 20:30:18 |**

**| tom | 2018-11-20 | 2019 | 08:30:00 | 2018-12-24 20:30:18 |**

**| lucy | 2018-10-16 | 1990 | 09:20:12 | 2018-10-16 09:20:12 |**

**| lucy | NULL | 1977 | NULL | NULL |**

**| tom | NULL | 2050 | NULL | NULL |**

**+------+------------+-------+----------+---------------------+**

**时间函数：**

|  |  |  |
| --- | --- | --- |
| **类型** | **用途** | **例如** |
| **now()** | **获取系统当前日期和时间** | **mysql> select now();** |
| **year()** | **执行时动态获得系统日期时间** | **mysql> select year(now());** |
| **sleep(N)** | **休闲N秒** | **mysql> select sleep(2);** |
| **curdate()** | **获取当前的系统日期** | **mysql> select curtime();** |
| **curtime()** | **获取当前的系统时刻** | **mysql> select curdate();** |
| **month()** | **获取指定时间中的月份** | **mysql> select month( now() );** |
| **date()** | **获取指定时间中的日期** | **mysql> select date( now() );** |
| **time** | **获取指定时间中的时刻** | **mysql> select time(now());** |

**mysql> insert into t5 values("lucy",date(now()),year(19901008),curtime(),now());**

**mysql> select \* from t5 where name='lucy';**

**+------+------------+-------+----------+---------------------+**

**| name | birthday | start | uptime | party |**

**+------+------------+-------+----------+---------------------+**

**| lucy | 2018-10-16 | 1990 | 09:20:12 | 2018-10-16 09:20:12 |**

**+------+------------+-------+----------+---------------------+**