**Morning：**

1. 数据库概述

-什么是数据库？

DB：数据库，依照某种数据模型进行组织并存放到存储器的数据集合

SQL：结构化管理语言

DBMS：数据库管理系统，如Mysql

DBS：数据库系统，DB+DBMS

-提供数据库服务的软件：

开源软件：Mysql

商业软件：Oracle、SQL server、DB2

-Mysql特点：

适合中小型企业的关系型数据库服务软件；跨平台；支持多种开发语言（php perl python ）；可移植性好。

1. 搭建数据库服务

**步骤1：卸载mariadb**

systemctl stop mariadb

rpm -e --nodeps mariadb-server #忽略依赖关系卸载mariadb服务

rpm -e --nodeps mariadb #卸载提供mariadb指令的软件

rm -rf /etc/my.cnf #移除配置文件

rm -rf /var/lib/mysql/\* #清空数据库目录

**步骤2：安装Mysql**

[root@mysql50 ~]# tar -xvf mysql-5.7.17.tar

[root@mysql50 ~]# yum -y install perl-JSON #安装依赖包

[root@mysql50 ~]# rpm -Uvh mysql-community\*.rpm

#-Uvh升级安装（覆盖旧版本）

[root@mysql50 ~]# rpm -qa | grep -i mysql

**步骤3：启动Mysqld服务**

[root@mysql50 ~]# systemctl {start|enable|status} mysqld

#首次启动对目录/var/lib/mysql做数据初始化；下次启动直接加载

[root@mysql50 ~]# netstat -antplu | grep :3306

[root@mysql50 ~]# ps aux | grep -i mysqld ; ll /var/lib/mysql/\*

#进程所有者为mysql，数据库目录下的文件所有者/组都是mysql

**步骤4：连接Mysql服务器**

三种方式：命令行、脚本、图形工具

命令：mysql -h 数据库服务器IP或主机名 -u数据库用户名 -p密码

[root@mysql50 ~]# grep 'password' /var/log/mysqld.log

A temporary password is generated for root@localhost: :-XrI-:#,5Kp

#首次启动登陆密码记录在mysqld.log中，正常情况记录日志

[root@mysql50 ~]# mysql -hlocalhost -uroot -p':-XrI-:#,5Kp'

mysql> \h #列出帮助信息

mysql> show databases \c #取消本行输入

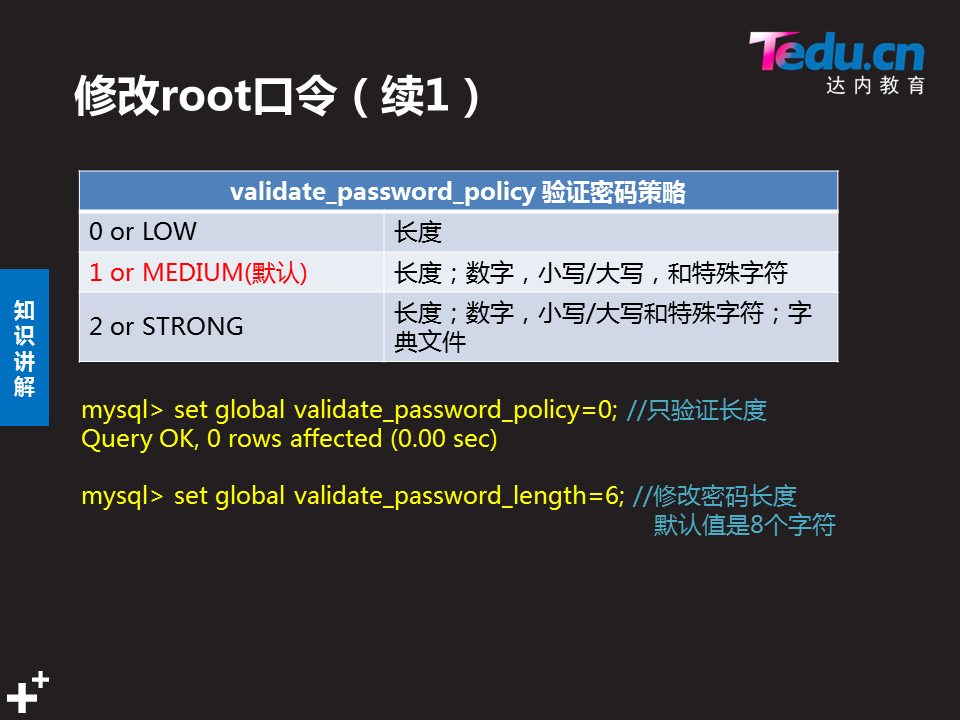
mysql> alter user root@"localhost" identified by "123qwE==";

#修改密码，密码需要符合复杂度

**Afternoon：**

1、Mysql的基本使用

**-修改密码策略&长度**



mysql> show variables like "%password%";

#查看所有包含password的变量

mysql> set global validate\_password\_policy=0; #密码策略

mysql> set global validate\_password\_length=6; #密码长度

mysql> alter user root@"localhost" identified by "123456";

vim /etc/my.conf #永久设置

validate\_password\_policy=0

validate\_password\_length=6

**-数据存储到过程**

连接数据库服务器-->建库-->建表-->插入记录-->断开连接

-建库 #类似于系统文件夹

-建表 #类似于系统文件

-插入记录 #类似于文件里的行

**-SQL操作指令**

数据库服务器 <-- 库 <-- 表 <-- 记录 <-- 列

1. 常用的SQL操作指令：

DDL数据定义语言（create、alter、drop）

DML数据操作语言（insert、update、delete）

DCL数据控制语言（grant、revoke）

DTL数据事务语言（commit、rollback、savepoint）

2）库管理命令

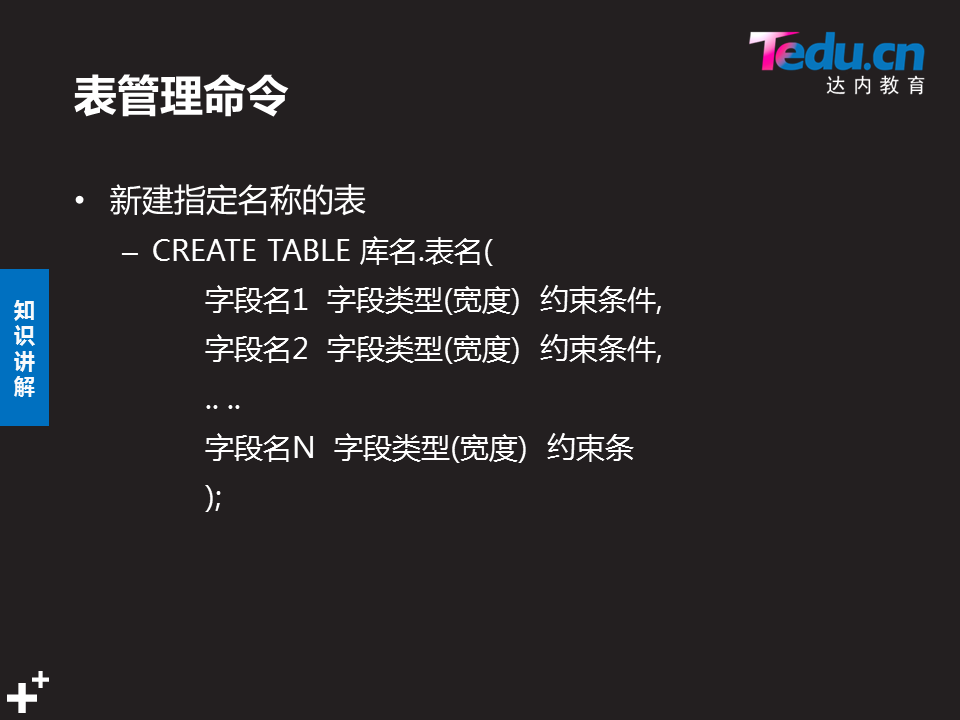
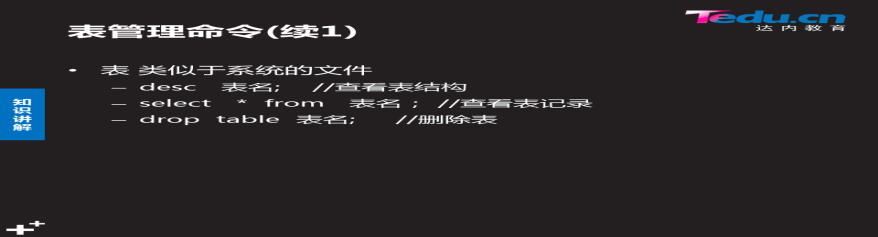


Tips：

-初始的4个库用于Mysql服务的加载，切勿删除；

-create/drop数据库，对应/var/lib/mysql/下的文件夹。

1. 表管理命令

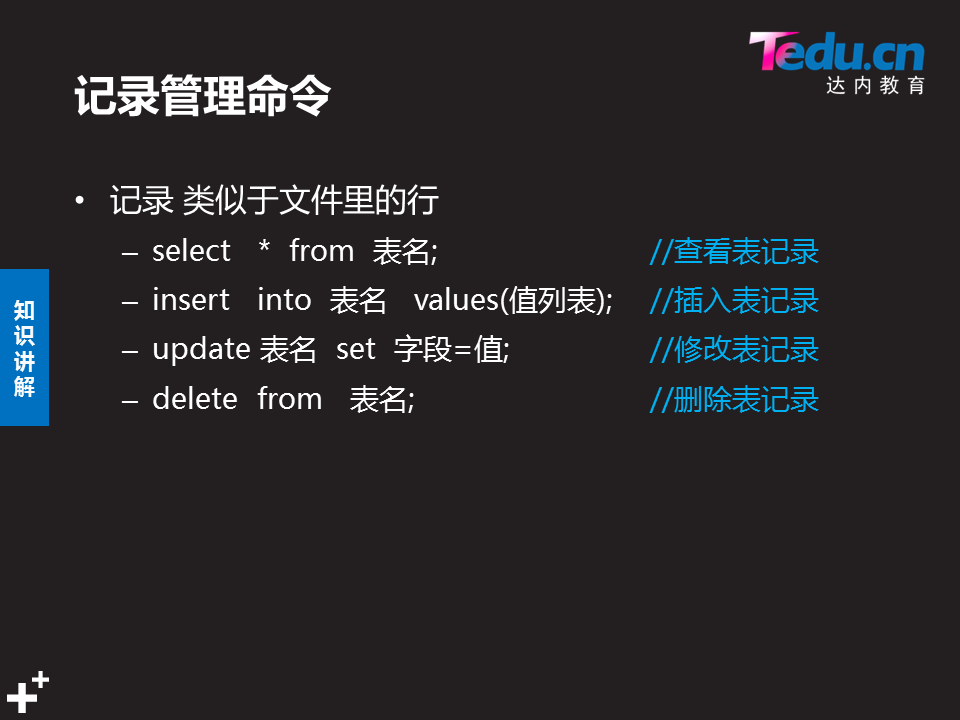
 

Tips：

-创建table后，对应/var/lib/mysql/库名/ 下生成文件.frm、.ibd，文件数量与存储引擎有关。

-删除table，/var/lib/mysql/xxxdb/下相应的文件会被清除。

4）表记录管理命令



Tips：

-插入表记录时，凡是字符都需要加””

-可配合where条件语句，对表记录进行查询、修改、删除管理

5）创建支持中文的表

show create table gamedb.t1; #查看创建t1表时设置的参数

create table gamedb.社团清单(绰号 char(10),刀龄 int)

DEFAULT CHARSET=utf8;

[root@dbsvr1 ~]# vim /etc/my.cnf #永久配置

.. ..

character\_set\_server=utf8

注意：一个汉字3个字节，但指定utf8后char(n)可以存n个汉字。

/var/lib/mysql下的文件名仍为乱码。

2、数据类型及案例

数值型、字符型、日期时间型、枚举型。

**-数值类型**

整型、浮点型



案例：

mysql> create table t1(age tinyint unsigned,pay float(7,2));

mysql> insert into t1 values(255,28000);

#float(总位数,小数位)，当整型数写入浮点数时，四舍五入。

**-字符类型**

|  |  |  |
| --- | --- | --- |
| char | 固定长度，不够右边补空格  最大长度255字符  超出长度，无法写入  节省系统资源，处理速度快 | 定长类型 |
| varchar | 按数据实际大小分配存储空间  最大长度65535字符  超出长度，无法写入  节省容量，但速度较慢 | 变长类型 |
| text/blob | 字符数大于65535存储时使用 | 大文本类型 |

案例：

mysql> create table t3(name char(5),homedir varchar(15),email varchar(30) );

mysql> insert into t3 values("bob","guangzhou","bob@qq.com");

**-日期时间类型**

1）日期时间类型

|  |  |  |
| --- | --- | --- |
| year | 范围：1901～2155  只用2位数字赋值时：  01～69视为2001～2069  70～99视为1970～1999 | 年份 |
| time | 格式：HH:MM:SS | 时间 |
| date | 范围：0001-01-01～9999-12-31 | 日期 |
| timestamp | 范围：1000-01-01 00:00:00  ～9999-12-31 23:59:59  若未赋值，默认为当前系统时间 | 日期时间 |
| datetime | 范围：1970-01-01 00:00:00  ～2038-01-19 03:14:07  若未赋值，默认为NULL | 日期时间 |

案例1：

mysql> create table t4( name char(3), birthday date, upwork time, party datetime);

mysql> insert into t4 values("b",19950418,083000,20181118203000);

#数据输入后，会自动切割成相应格式

mysql> select \* from t4;

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

| name | birthday | start | upwork | party |

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

| b | 1995-04-18 | 1900 | 08:30:00 | 2018-11-16 20:30:00 |

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

案例2：

mysql> create table timedb.t1( date1 timestamp, date2 datetime );

mysql> insert into timedb.t1(date2) values(19990909111111);

#指定某字段进行赋值

mysql> select \* from timedb.t1;

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

| date1 | date2 |

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

| 2018-11-17 09:30:03 | 1999-09-09 11:11:11 |

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

2）时间函数



案例：

mysql> SELECT now(); #获取当前时间

mysql> SELECT year(now()),month(now()),day(now());

#获取当前日期的年、月、日

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

| year(now()) | month(now()) | day(now()) |

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

| 2018 | 11 | 16 |

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

**-枚举类型**

|  |  |  |
| --- | --- | --- |
| enum | enum(值1,值2,值n) | 从给定值集合中选单个值 |
| set | set(值1,值2,值n) | 从给定值集合中选一个或多个值 |

案例：

mysql> create table t5( name char(5), hobby set("eat","game","movie"), sex enum("man","woman","no") );

mysql> insert into t5 values("bob","eat,game","man");

mysql> select \* from t5;

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

| name | hobby | sex |

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

| bob | eat,game | man |

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