1. 什么是数据库？

提供数据存储的服务，功能是存储数据。

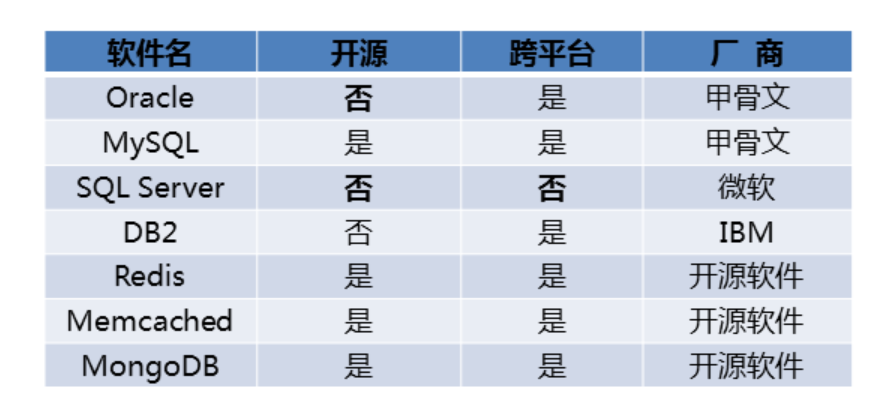
存储数据的仓库

什么是数据？

各种应用软件上的历史信息都是数据。

数据库的服务软件？

提供数据管理和服务的软件

常见的软件：

1. MySQL

专用术语：

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。

DBMS：数据库管理系统、用来操纵和管理数据库服务的软件。

DBS: 指带有数据库并整合了数据库管理软件的计算机系统。

MySQL介绍

最早属于瑞典公司MySQL AB

2008年1月 MySQL AB 公司被sun收购

2009年 4月 sun 被oracle 收购

MariaDB

为应付MySQL带来的闭源风险而诞生

由MySQL原作者主导开发，与Mysql保持最大范围兼容

特点：

适用于中小规模、关系型数据库系统

支持linux windows unix

支持Python Java Perl PHP语言

典型的应用环境、lamp 与appache http server结合

Lnmp平台 与nginx组合

并且mysql与mariadb不能同时安装。

1. mysql数据服务器的搭建

庞丽静 数据库

QQ 240398019

中心名\_班级名\_姓名

北京天坛\_nsd2005\_大熊

工作岗位 数据库管理员 DBA

课程量 大 （讲课 练习时间 排错时间）

笔 记 随堂文档

一 rdbms1 数据库基础 5天

rdbms1\_day01

学习任务如下：

1 相关概念

什么是数据库？ 提供数据存储的服务 功能（存储数据）

存储数据的仓库

数据是啥样的？ 各种应用软件的上历史信息 都是数据

数据库服务软件有哪些？提供服务和 对数据做管理的软件

软件名 商业软件 开源软件 到哪里去下载

是否跨平台（Linux windows unix）

专业术语： DB? DBMS? DBS?

mysql介绍 ？ 特点 应用场景

2 搭建MySQL数据库服务器

2.1 环境准备

虚拟机 克隆新虚拟机 192.168.4.50 关闭selinux firewalld 配置yum

软件 mysql-5.7.17.tar

2.2 具体步骤如下：

1 清除mariadb软件

systemctl stop mariadb

rpm -e --nodeps mariadb-server mariadb

rm -rf /etc/my.cnf

rm -rf /var/lib/mysql

2 安装MySQL服务软件

]# tar -xvf mysql-5.7.17.tar

]# yum -y install mysql-community-\*.rpm

]# rpm -qa | grep -i mysql

3 启动mysqld服务

]# systemctl start mysqld

]# systemctl enable mysqld

4 查看服务状态（进程 端口号 ）

]# systemctl status mysqld

]# ps -C mysqld

]# netstat -utnlp | grep 3306

5 了解与服务相关的信息

进程名 mysqld

数据传输协议 TCP

进程的所有者名和所属组名 mysql / mysql

grep mysql /etc/passwd

grep mysql /etc/group

ps aux | grep mysqld

默认端口号 3306

默认主配置文件 /etc/my.cnf

cat /etc/my.cnf

默认数据库目录 /var/lib/mysql/

ls /var/lib/mysql/

默认错误日志文件 /var/log/mysqld.log

cat /var/log/mysqld.log

6 查看初始密码(启动数据库服务器随机创建)

]# grep password /var/log/mysqld.log

root@"localhost" xdf<af&!j$dfaf)

7 使用初始密码登录

]# mysql -hlocalhost -uroot -p'xdf<af&!j$dfaf)'

MySQL> show databases; 报错

8 修改登录密码(密码要符合密码策略的要求)， 断开连接

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

mysql> exit

9 使用修改的密码登录， 查看数据

]# mysql -hlocalhost -uroot -p123qqq...A

mysql> show databases; 会显示默认的4个数据库

10 修改密码策略 练习 10分钟到 11:36

mysql> show variables like "%password%";

mysql> set global validate\_password\_policy=0;

mysql> set global validate\_password\_length=6;

mysql> show variables like "%password%";

11 修改新密码

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

12 使用新密码登录

mysql> exit

Bye

[root@host50 ~]# mysql -hlocalhost -uroot -ptarena

13 把密码策略写进配置文件，会永久有效

[mysqld]

validate\_password\_policy=0

validate\_password\_length=6

:wq

[root@host50 ~]# systemctl restart mysqld

[root@host50 ~]# mysql -hlocalhost -uroot -ptarena

mysql> show variables like "validate\_password\_policy";

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

| Variable\_name | Value |

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

| validate\_password\_policy | LOW |

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

1 row in set (0.00 sec)

mysql> show variables like "validate\_password\_length";

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

| Variable\_name | Value |

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

| validate\_password\_length | 6 |

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

1 row in set (0.00 sec)

mysql>

3 数据库服务的基本管理

3.1 连接数据库服务的连接方式？0

3.2 把数据存储到数据库服务器的步骤？

3.3 SQL命令使用规则？ (SQL 结构化查询语言)

3.4 SQL命令分类

3.5 基本管理命令

库的常用命令

库名的命名规则

表的常用命令 练习到14:50

创建新表

create table 库名.表名(列名 类型 ，列名 类型 ) ;

mysql> create database db1;

mysql> create table db1.stu(name char(5) , age int , class char(10) );

mysql> desc db1.stu;

记录常用命令

insert into库名.表名 values(值 ， 值);

mysql> insert into db1.stu values("bob" , 19 , "nsd2005");

mysql> select \* from db1.stu;

mysql> update db1.stu set name="jerry" where age=19 ;

mysql> update db1.stu set name="jerrya";

mysql> delete from db1.stu;

向表里添加行时可以写中文 （字符集utf8）

create table db1.t1(姓名 char(10) , 地址 char(20) , 性别 char(4) ) DEFAULT CHARSET=utf8;

mysql> show create table db1.t1 \G

mysql> insert into db1.t1 values ("张三","北京","男"),("小红","天津","女") ;

mysql> select \* from db1.t1;

4 MySQL数据类型 （重点 难点）

4.1 字符类型 （存储字符的类型） 姓名

char varchar blob text

4.2 数值类型

整数类型:根据存储数据的范围分为以下几种类型

有负数的范围 全都是正整数

有符号 无符号 unsigned

微小整形

小整形

中整形

大整形

极大整形

一位数的范围 0 - 9

二位数的范围 10 - 99

三位数的范围 100-999

浮点类型（有小数点的数） float double

小数的组成 整数.小数

118.22

练习+休息到 17：15

mysql> create table db1.t4(name char(5) , age tinyint unsigned , money float(7,2),pai double);

mysql> insert into db1.t4 values ("bob" , 21 , 99923.88 , 3.1415926);

mysql> insert into db1.t4 values ("john" , 28.5 , 23000 , 3.1415926);

mysql> insert into db1.t4 values ("tom" , 28.4 , 23000 , 3.1415926);

mysql> insert into db1.t4 values ("tomA" , 28.4 , 28.4 , 3.1415926);

4.3 枚举类型 给表头赋值时 ，值必须在定义的范围选择

字段名 enum(值列表) 单选

字段名 set(值列表) 多选项

练习5分钟 到 17：31

mysql> create table db1.t6 ( name char(10) , sex enum("boy" , "girl" , "no" ) , love set("eat","sleep","game","money"));

mysql> insert into db1.t6 values ("bob","man","it,football");

ERROR 1265 (01000): Data truncated for column 'sex' at row 1

mysql>

mysql> insert into db1.t6 values ("bob","boy","it,football");

ERROR 1265 (01000): Data truncated for column 'love' at row 1

mysql>

mysql> insert into db1.t6 values ("bob","boy","eat,sleep,money");

Query OK, 1 row affected (0.01 sec)

mysql> select \* from db1.t6;

4.4 日期时间类型

年 year YYYY 2020

日期 date YYYYMMDD 20200812

时间 time HHMMSS 052528

日期时间 datetime或 timestamp YYYYMMDDHHMMSS

20200812173758

mysql> create table db1.t7( name char(10) , up\_class time , start year , party datetime);

mysql> insert into db1.t7 values ("bob",090000 , 1990 , 20200913183000);

mysql> select \* from db1.t7 ;

year 类型 要求使用4位数赋值 当使用2位数赋值时：

01~69视为2001~2069

70~99视为1970~1999

mysql> insert into db1.t7 values ("tom",190000, 58 , 20200813100000 );

Query OK, 1 row affected (0.01 sec)

mysql> insert into db1.t7 values ("tomA",190000, 88 , 20200813100000 );

Query OK, 1 row affected (0.01 sec)

mysql> select \* from db1.t7;

时间函数的使用 （MySQL服务自带获取时间的命令）

mysql > select curtime();

mysql > select curdate();

mysql > select now();

mysql > select year(now());

可以使用时间函数获取的数据 给日期时间类型的表头赋值

mysql> insert into db1.t7 values ("john", curtime() , year(now()), now());

datetime 和 timestamp 日期时间类型的比较 (下次可讲)

周四额外练习题

在db1库下创建前任信息表 （表结构自定义）合理的使用学过的数据类型 把前任信息存放到表里。

二 rdbms2 数据库进阶 5天

三 NoSQL redis数据库服务 3天

四 project2 数据库项目 2天

1. [案例1：构建MySQL服务器](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/RDBMS1/DAY01/CASE/01/index.html#case1)
2. [案例2：数据库基本管理](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/RDBMS1/DAY01/CASE/01/index.html#case2)
3. [案例3：字符类型](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/RDBMS1/DAY01/CASE/01/index.html#case3)
4. [案例4：数值类型](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/RDBMS1/DAY01/CASE/01/index.html#case4)
5. [案例5：日期时间类型](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/RDBMS1/DAY01/CASE/01/index.html#case5)
6. [案例6：枚举类型](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/RDBMS1/DAY01/CASE/01/index.html#case6)

**1 案例1：构建MySQL服务器**

**1.1 问题**

要求如下：

* 在IP地址192.168.4.50主机上部署mysql服务
* 设置数据库管理员root本机登录密码为tarena

**1.2 方案**

克隆新的虚拟机：

eth0网卡:192.168.4.50

主机名称:host50

下载软件mysql-5.7.17.tar

关闭防火墙（如果有的话）

关闭SELinux（如果有的话）

**1.3 步骤**

实现此案例需要按照如下步骤进行。

**步骤一：准备工作**

1）如果之前有mariadb，则需要先卸载，并删除对应的配置与数据：

1. **[**root@localhost **~]**# systemctl stop mariadb

2）删除/etc/my.cnf配置文件

此配置文件由RHEL自带的mariadb-libs库提供：

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

3）删除数据

1. **[**root@localhost **~]**# rm **-**rf **/var**/lib/mysql/\*

4）卸载软件包（没有会显示未安装软件包）

1. **[**root@localhost **~]**# rpm **-**e **--**nodeps mariadb**-**server mariadb
2. 警告：/var/log**/**mariadb**/**mariadb**.**log 已另存为/var/log**/**mariadb**/**mariadb**.**log**.**rpmsave

**步骤二：安装mysql软件包**

1）解压mysql-5.7.17.tar 软件包

1. **[root@host50 ~]# tar -xvf mysql-5.7.17.tar** //解压mysql整合包
2. **./mysql-community-client-5.7.17-1.el7.x86\_64.rpm**
3. **./mysql-community-common-5.7.17-1.el7.x86\_64.rpm**
4. **./mysql-community-devel-5.7.17-1.el7.x86\_64.rpm**
5. **./mysql-community-embedded-5.7.17-1.el7.x86\_64.rpm**
6. **./mysql-community-embedded-compat-5.7.17-1.el7.x86\_64.rpm**
7. **./mysql-community-embedded-devel-5.7.17-1.el7.x86\_64.rpm**
8. **./mysql-community-libs-5.7.17-1.el7.x86\_64.rpm**
9. **./mysql-community-libs-compat-5.7.17-1.el7.x86\_64.rpm**
10. **./mysql-community-minimal-debuginfo-5.7.17-1.el7.x86\_64.rpm**
11. **./mysql-community-server-5.7.17-1.el7.x86\_64.rpm**
12. **./mysql-community-test-5.7.17-1.el7.x86\_64.rpm**

2）安装MySQL软件包

1. **[root@host50 ~]# yum -y install mysql-community-\*.rpm** //yum安装自动解决依赖
2. **Rpm -qa | grep -i mysql 查看安装数据包**
3. **./mysql-community-client-5.7.17-1.el7.x86\_64.rpm**
4. **./mysql-community-common-5.7.17-1.el7.x86\_64.rpm**
5. **./mysql-community-devel-5.7.17-1.el7.x86\_64.rpm**
6. **./mysql-community-embedded-5.7.17-1.el7.x86\_64.rpm**
7. **./mysql-community-embedded-compat-5.7.17-1.el7.x86\_64.rpm**
8. **./mysql-community-embedded-devel-5.7.17-1.el7.x86\_64.rpm**
9. **./mysql-community-libs-5.7.17-1.el7.x86\_64.rpm**
10. **./mysql-community-libs-compat-5.7.17-1.el7.x86\_64.rpm**
11. **./mysql-community-minimal-debuginfo-5.7.17-1.el7.x86\_64.rpm**
12. **./mysql-community-server-5.7.17-1.el7.x86\_64.rpm**
13. **./mysql-community-test-5.7.17-1.el7.x86\_64.rpm**

3）启动MySQL数据库服务并设置开机自启

提示：第一次启动，需要初始化数据，会比较慢

1. **[**root@host50 **~]**# systemctl start mysqld //启动mysql服务
2. **[**root@host50 **~]**# systemctl enable mysqld //设置开机自启
3. **[**root@host50 **~]**# systemctl status mysqld //查看mysql服务状态
4. ● mysqld**.**service **-** MySQL Server
5. Loaded**:** loaded **(**/usr/lib**/**systemd**/**system**/**mysqld**.**service**;** enabled**;** vendor preset**:** disabled**)**
6. Active**:** active **(**running**)** since 二 **2018-08-28** **10:03:24** CST**;** 8min ago
7. Docs**:** man**:**mysqld**(8)**
8. http**:**//dev.mysql.com/doc/refman/en/using-systemd.html
9. Main PID**:** **4284** **(**mysqld**)**
10. CGroup**:** /system.slice/mysqld**.**service
11. └─**4284** **/**usr**/**sbin**/**mysqld **--**daemonize **--**pid**-**file**=**/var/r**...**
12. **8**月 **28** **10:02:56** localhost**.**localdomain systemd**[1]:** Starting MySQ**...**
13. **8**月 **28** **10:03:24** localhost**.**localdomain systemd**[1]:** Started MySQL**...**
14. Hint**:** Some lines were ellipsized**,** use **-**l to show **in** full**.**

**步骤三：连接MySQL服务器，修改密码**

1）查看初始密码

1. **[**root@host50 **~]**#grep –i 'password' /var/log**/**mysqld**.**log
2. **2017-04-**01T18**:10:42.**948679Z **1** **[**Note**]** A temporary password is generated **for** root@localhost**:** mtoa**>**Av**<**p6Yk //随机生成的管理密码为mtoa>Av<p6Yk

2）使用初始密码连接mysql服务

1. **[**root@host50 **~]**# mysql **-**u root **-**p'mtoa>Av<p6Yk' //初始密码登录，
2. mysql**:** **[**Warning**]** Using a password on the command line **interface** can be insecure**.**
3. Welcome to the MySQL monitor**.** Commands end **with** **;** or **\**g**.**
4. Your MySQL connection id is **11**
5. Server version**:** **5.7.17**
6. Copyright **(**c**)** **2000,** **2016,** Oracle and**/**or its affiliates**.** All rights reserved**.**
7. Oracle is a registered trademark of Oracle Corporation and**/**or its
8. affiliates**.** Other names may be trademarks of their respective
9. owners**.**
10. Type 'help;' or '**\h**' **for** help**.** Type '**\c**' to clear the current input statement**.**
11. mysql**>**                                     //登录成功后，进入SQL操作环境

3）重置数据库管理员roo本机登录密码

1. mysql**>** show databases**;**
2. ERROR **1820** **(**HY000**):** You must reset your password using ALTER USER statement before executing **this** statement //提示必须修改密码
3. mysql**>** alter user root@”localhost” identified by "123qqq…A"**;** //修改登陆密码
4. Query OK**,** **0** rows affected **(0.00** sec**)**
5. mysql**>** exit //断开连接
6. **[**root@host50 **~]**#

4）修改密码策略

1. **[**root@host50 **~]**# mysql **-**uroot –p123qqq…A
2. mysql**>** show variables like “%password%”;查看变量，密码策略。
3. mysql**>**set global validate\_password\_policy**=0;** //只验证长度
4. Query OK**,** **0** rows affected **(0.00** sec**)**
5. mysql**>**set global validate\_password\_length**=6**； //修改密码长度,默认值是8个字符
6. Query OK**,** **0** rows affected **(0.00** sec**)**
7. mysql**>** alter user root@”localhost” identified by "tarena"**;** //修改登陆密码
8. Query OK**,** **0** rows affected **(0.00** sec**)**
9. mysql**>**exit

5）使用修改后的密码登录

1. **[**root@host50 **~]**# mysql **-**uroot **-**ptarena        //登录
2. Welcome to the MySQL monitor**.** Commands end **with** **;** or **\**g**.**
3. Your MySQL connection id is **15**
4. Server version**:** **5.7.17** MySQL Community Server **(**GPL**)**
5. Copyright **(**c**)** **2000,** **2016,** Oracle and**/**or its affiliates**.** All rights reserved**.**
6. Oracle is a registered trademark of Oracle Corporation and**/**or its
7. affiliates**.** Other names may be trademarks of their respective
8. owners**.**
9. mysql**>** show databases**;** //查看数据库
10. **+--------------------+**
11. **|** Database **|**
12. **+--------------------+**
13. **|** information\_schema **|**
14. **|** mysql **|**
15. **|** performance\_schema **|**
16. **|** sys                **|**
17. **+--------------------+**
18. **4** rows **in** set **(0.00** sec**)**

**将密码策略写进配置文件/etc/my.cnf**

**Vim /etc/my.cnf**

validate\_password\_policy**=0**

validate\_password\_length**=6**

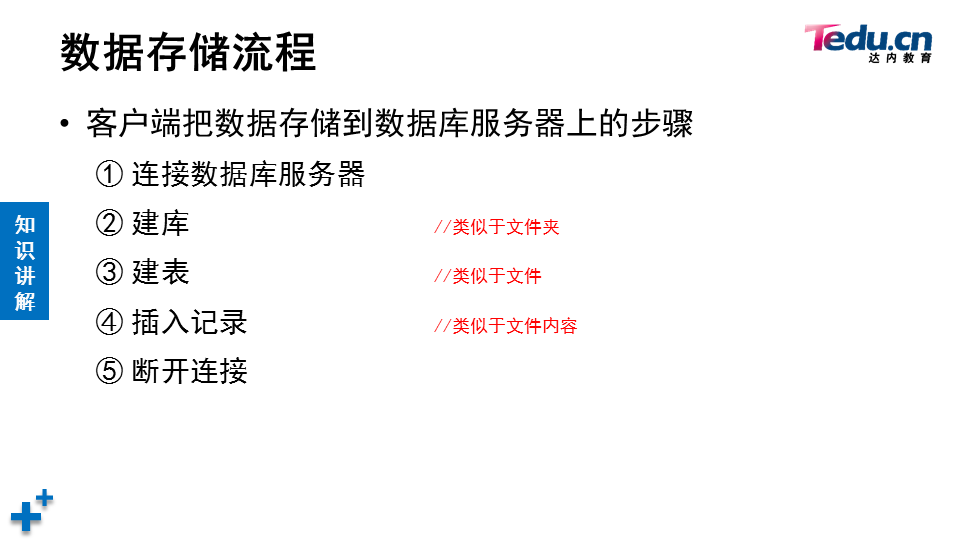
1. mysql**>**

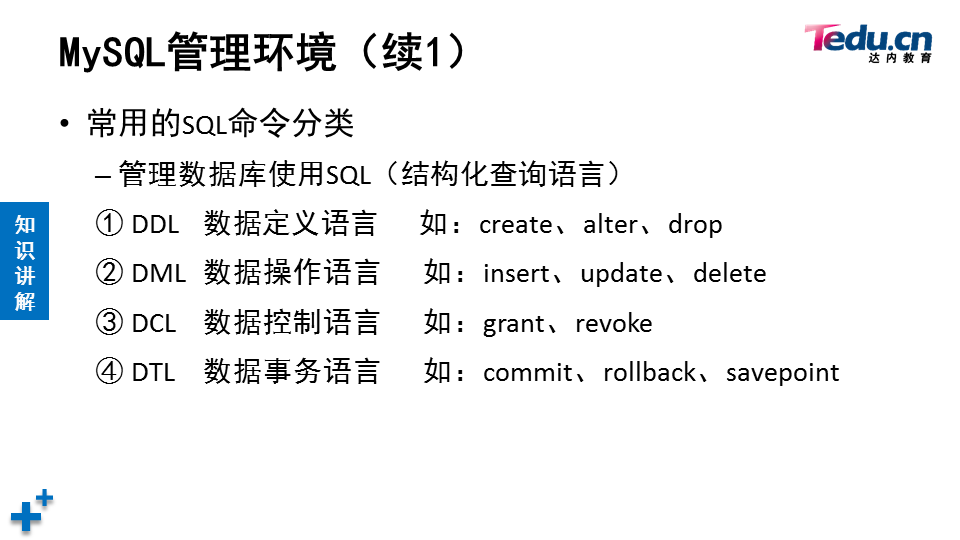
**2 案例2：数据库基本管理**

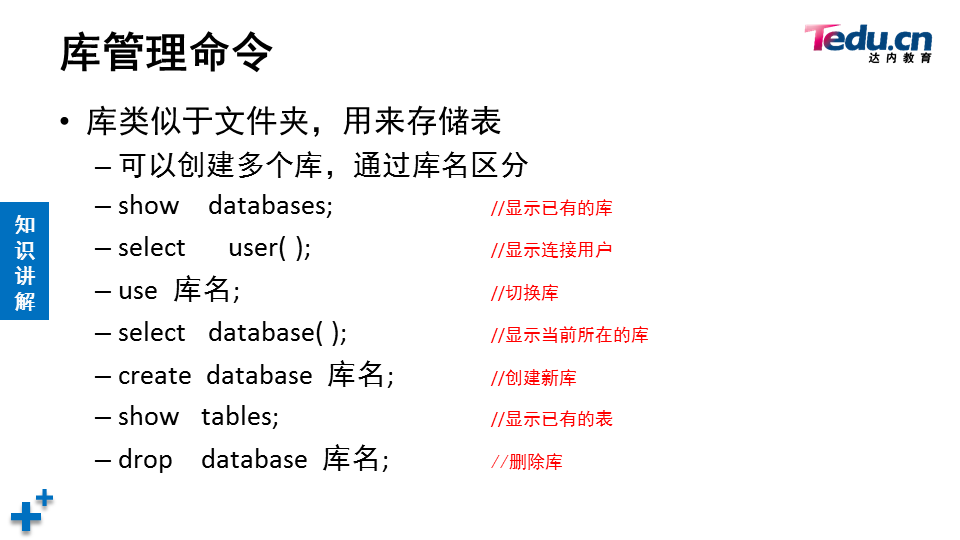
**2.1 问题**

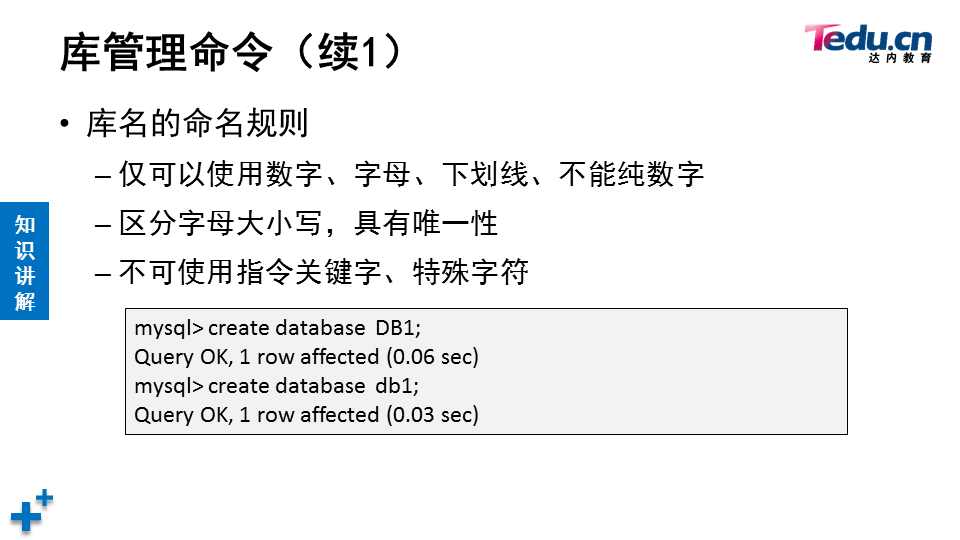
本案例练习对库、表、记录的基本管理，具体操作如下：

* 使用mysql命令、Web页面、安装图形软件、编写脚本



* 练习库管理命令（查看、删除、创建库、切换）
* 
* 练习表管理命令（查看、删除、创建表）

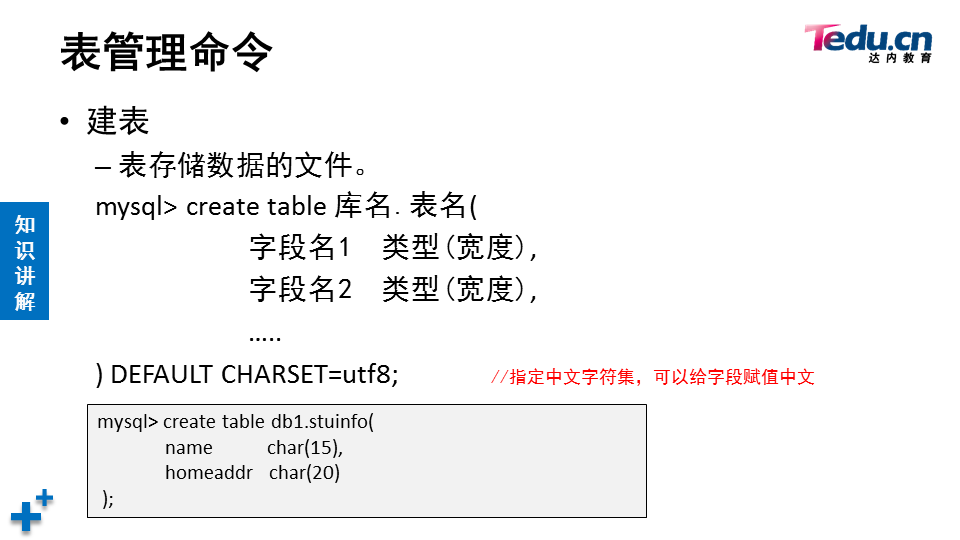
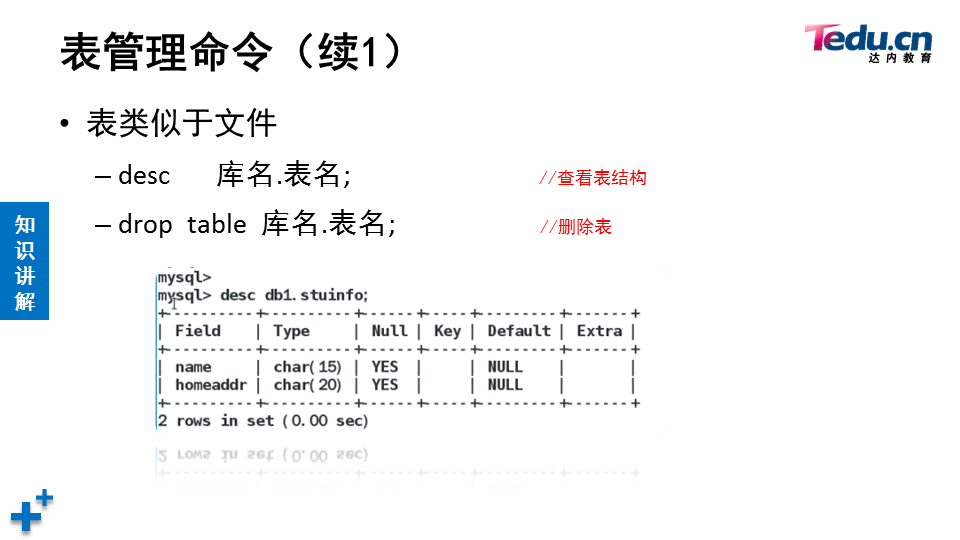


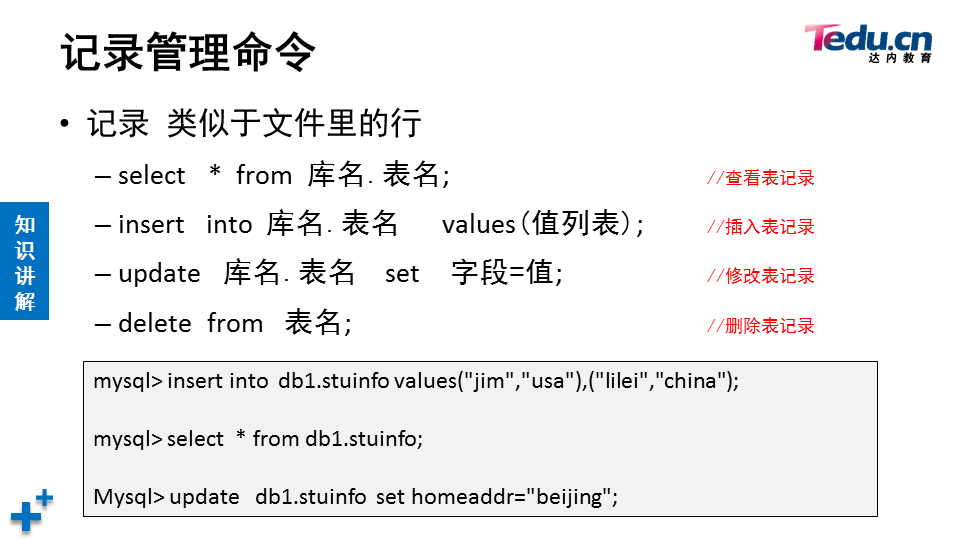


* 练习记录管理命令（插入、查看、修改、删除）

表－1 测试用表数据





**2.2 步骤**<http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/RDBMS1/DAY01/COURSE/LINUXNSD_V01RDBMS1DAY01_035.png>

实现此案例需要按照如下步骤进行。

**步骤一：使用mysql命令连接数据库**

连接MySQL服务器时，最基本的用法是通过 -u 选项指定用户名、-p指定密码。密码可以写在命令行（如果不写，则出现交互，要求用户输入），当然基于安全考虑一般不推荐这么做：

1. **[**root@dbsvr1 **~]**# mysql **-**uroot **-**p123456         //紧挨着选项，不要空格
2. mysql**:** **[**Warning**]** Using a password on the command line **interface** can be insecure**.**
3. Welcome to the MySQL monitor**.** Commands end **with** **;** or **\**g**.**
4. Your MySQL connection id is **16**
5. Server version**:** **5.7.17** MySQL Community Server **(**GPL**)**
6. Copyright **(**c**)** **2000,** **2016,** Oracle and**/**or its affiliates**.** All rights reserved**.**
7. Oracle is a registered trademark of Oracle Corporation and**/**or its
8. affiliates**.** Other names may be trademarks of their respective
9. owners**.**
10. Type 'help;' or '**\h**' **for** help**.** Type '**\c**' to clear the current input statement**.**
11. mysql**>** exit                                 //退出已登录的mysql> 环境
12. Bye

默认情况下，msyql命令会连接本机的MySQL服务。但在需要的时候，可以通过 -h 选项指定远程主机；

1. **[**root@dbsvr1 **~]**# mysql **-**h **127.0.0.1** –u root –p
2. Enter password**:**
3. Welcome to the MySQL monitor**.** Commands end **with** **;** or **\**g**.**
4. Your MySQL connection id is **17**
5. Server version**:** **5.7.17** MySQL Community Server **(**GPL**)**
6. Copyright **(**c**)** **2000,** **2016,** Oracle and**/**or its affiliates**.** All rights reserved**.**
7. Oracle is a registered trademark of Oracle Corporation and**/**or its
8. affiliates**.** Other names may be trademarks of their respective
9. owners**.**
10. Type 'help;' or '**\h**' **for** help**.** Type '**\c**' to clear the current input statement**.**
11. mysql**>** exit                                 //退出已登录的mysql环境
12. Bye

**步骤二：练习查看/删除/创建库的相关操作**

以root用户登入“mysql> ”环境后，可以执行各种MySQL指令、SQL指令。基本的用法事项如下：

* 操作指令不区分大小写（库名/表名、密码、变量值等除外）。
* 每条SQL指令以 ; 结束或分隔。
* 不支持 Tab 键自动补齐。
* \c 可废弃当前编写错的操作指令。

1）查看现有的库

1. mysql**>** show databases**;** //查看现有的库
2. **+--------------------+**
3. **|** Database **|**
4. **+--------------------+**
5. **|** information\_schema **|**                             //信息概要库
6. **|** mysql **|**                             //授权库
7. **|** performance\_schema **|**                             //性能结构库
8. **|** sys **|**                             //系统元数据库
9. **+--------------------+**
10. **4** rows **in** set **(0.15** sec**)**

2）切换/使用指定的库

1. mysql**>** use sys**;** //切换到sys库
2. Database changed
3. mysql**>** select database**();**                          //确认当前所在的库
4. **+------------+**
5. **|** DATABASE**()** **|**
6. **+------------+**
7. **|** sys **|**
8. **+------------+**
9. **1** row **in** set **(0.00** sec**)**

切换到mysql库：

1. mysql**>** use mysql**;** //切换到mysql库
2. Reading table information **for** completion of table and column names
3. You can turn off **this** feature to get a quicker startup **with** **-**A
4. Database changed
5. mysql**>** select database**();**                         //确认当前所在的库
6. **+------------+**
7. **|** DATABASE**()** **|**
8. **+------------+**
9. **|** mysql **|**
10. **+------------+**
11. **1** row **in** set **(0.00** sec**)**
12. **5** rows **in** set **(0.00** sec**)**

3）新建名为newdb的库，确认结果：

1. mysql**>** create database newdb**;** //新建名为newdb的库
2. Query OK**,** **1** row affected **(0.00** sec**)**
3. mysql**>** show databases**;**
4. **+--------------------+**
5. **|** Database **|**
6. **+--------------------+**
7. **|** information\_schema **|**
8. **|** mydb **|**                         //新建的mydb库
9. **|** mysql **|**
10. **|** newdb **|**                         //新建的newdb库
11. **|** performance\_schema **|**
12. **|** sys **|**
13. **+--------------------+**
14. **6** rows **in** set **(0.00** sec**)**

4）删除指定的库

1. mysql**>** drop database newdb**;** //删除名为newdb的库
2. Query OK**,** **0** rows affected **(0.01** sec**)**
3. mysql**>** show databases**;**                         //确认删除结果，已无newdb库
4. **+--------------------+**
5. **|** Database **|**
6. **+--------------------+**
7. **|** information\_schema **|**
8. **|** mydb **|**
9. **|** mysql **|**
10. **|** performance\_schema **|**
11. **|** sys **|**
12. **+--------------------+**
13. **5** rows **in** set **(0.00** sec**)**

**步骤三：练习查看/删除/创建表的相关操作**

1）查看指定的库里有哪些表

查看mysql库里有哪些表：

1. mysql**>** use mysql**;**
2. Reading table information **for** completion of table and column names
3. You can turn off **this** feature to get a quicker startup **with** **-**A
4. Database changed
5. mysql**>** show tables**;**
6. **+---------------------------+**
7. **|** Tables\_in\_mysql **|**
8. **+---------------------------+**
9. **|** columns\_priv **|**
10. **|** db **|**
11. **|** engine\_cost **|**
12. **|** event **|**
13. **|** func **|**
14. **|** general\_log **|**
15. **|** gtid\_executed **|**
16. **|** help\_category **|**
17. **|** help\_keyword **|**
18. **|** help\_relation **|**
19. **|** help\_topic **|**
20. **|** innodb\_index\_stats **|**
21. **|** innodb\_table\_stats **|**
22. **|** ndb\_binlog\_index **|**
23. **|** plugin **|**
24. **|** proc **|**
25. **|** procs\_priv **|**
26. **|** proxies\_priv **|**
27. **|** server\_cost **|**
28. **|** servers **|**
29. **|** slave\_master\_info **|**
30. **|** slave\_relay\_log\_info **|**
31. **|** slave\_worker\_info **|**
32. **|** slow\_log **|**
33. **|** tables\_priv **|**
34. **|** time\_zone **|**
35. **|** time\_zone\_leap\_second **|**
36. **|** time\_zone\_name **|**
37. **|** time\_zone\_transition **|**
38. **|** time\_zone\_transition\_type **|**
39. **|** user **|** //存放数据库用户的表
40. **+---------------------------+**
41. **31** rows **in** set **(0.00** sec**)**

2）查看指定表的字段结构

当前库为mysql，查看columns\_priv表的结构，以列表形式展现：

1. mysql**>** desc columns\_priv**\**G         //查看表结构，以列表形式展现，末尾不用分号
2. **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*** **1.** row **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***
3. Field**:** Host
4. Type**:** char**(60)**
5. Null**:** NO
6. Key**:** PRI
7. Default**:**
8. Extra**:**
9. **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*** **2.** row **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***
10. Field**:** Db
11. Type**:** char**(64)**
12. Null**:** NO
13. Key**:** PRI
14. Default**:**
15. Extra**:**
16. **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*** **3.** row **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***
17. Field**:** User
18. Type**:** char**(32)**
19. Null**:** NO
20. Key**:** PRI
21. Default**:**
22. Extra**:**
23. **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*** **4.** row **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***
24. Field**:** Table\_name
25. Type**:** char**(64)**
26. Null**:** NO
27. Key**:** PRI
28. Default**:**
29. Extra**:**
30. **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*** **5.** row **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***
31. Field**:** Column\_name
32. Type**:** char**(64)**
33. Null**:** NO
34. Key**:** PRI
35. Default**:**
36. Extra**:**
37. **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*** **6.** row **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***
38. Field**:** Timestamp
39. Type**:** timestamp
40. Null**:** NO
41. Key**:**
42. Default**:** CURRENT\_TIMESTAMP
43. Extra**:** on update CURRENT\_TIMESTAMP
44. **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*** **7.** row **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***
45. Field**:** Column\_priv
46. Type**:** set**(**'Select'**,**'Insert'**,**'Update'**,**'References'**)**
47. Null**:** NO
48. Key**:**
49. Default**:**
50. Extra**:**
51. **7** rows **in** set **(0.01** sec**)**

查看columns\_priv表的结构，以表格形式展现：

1. mysql**>** desc columns\_priv**;**         //查看表结构，以表格形式展现末尾需要有分号
2. **+-------------+----------------------------------------------+------+-----+-------------------+-----------------------------+**
3. **|** Field **|** Type **|** Null **|** Key **|** Default **|** Extra **|**
4. **+-------------+----------------------------------------------+------+-----+-------------------+-----------------------------+**
5. **|** Host **|** char**(60)** **|** NO **|** PRI **|** **|** **|**
6. **|** Db **|** char**(64)** **|** NO **|** PRI **|** **|** **|**
7. **|** User **|** char**(32)** **|** NO **|** PRI **|** **|** **|**
8. **|** Table\_name **|** char**(64)** **|** NO **|** PRI **|** **|** **|**
9. **|** Column\_name **|** char**(64)** **|** NO **|** PRI **|** **|** **|**
10. **|** Timestamp **|** timestamp **|** NO **|** **|** CURRENT\_TIMESTAMP **|** on update CURRENT\_TIMESTAMP **|**
11. **|** Column\_priv **|** set**(**'Select'**,**'Insert'**,**'Update'**,**'References'**)** **|** NO **|** **|** **|** **|**
12. **+-------------+----------------------------------------------+------+-----+-------------------+-----------------------------+**
13. **7** rows **in** set **(0.00** sec**)**

上述操作中，当引用非当前库中的表时，可以用“库名.表名”的形式。比如，切换为mysql库再执行“desc columns\_priv;”，与以下操作的效果是相同的：

1. mysql**>** desc mysql**.**columns\_priv**;**
2. **+-------------+----------------------------------------------+------+-----+-------------------+-----------------------------+**
3. **|** Field **|** Type **|** Null **|** Key **|** Default **|** Extra **|**
4. **+-------------+----------------------------------------------+------+-----+-------------------+-----------------------------+**
5. **|** Host **|** char**(60)** **|** NO **|** PRI **|** **|** **|**
6. **|** Db **|** char**(64)** **|** NO **|** PRI **|** **|** **|**
7. **|** User **|** char**(16)** **|** NO **|** PRI **|** **|** **|**
8. **|** Table\_name **|** char**(64)** **|** NO **|** PRI **|** **|** **|**
9. **|** Column\_name **|** char**(64)** **|** NO **|** PRI **|** **|** **|**
10. **|** Timestamp **|** timestamp **|** NO **|** **|** CURRENT\_TIMESTAMP **|** on update CURRENT\_TIMESTAMP **|**
11. **|** Column\_priv **|** set**(**'Select'**,**'Insert'**,**'Update'**,**'References'**)** **|** NO **|** **|** **|** **|**
12. **+-------------+----------------------------------------------+------+-----+-------------------+-----------------------------+**
13. **7** rows **in** set **(0.00** sec**)**

3）在test库中创建一个名为pwlist的表

包括name、password两列，其中name列作为主键。两个字段值均不允许为空，其中密码列赋予默认空值，相关操作如下所述。

切换到mydb库：

1. mysql**>** use mydb**;**
2. Database changed

新建pwlist表：

1. mysql**>** create table pwlist**(**
2. **->** name char**(16)** not **null,**
3. **->** password char**(48)default** ''**,**
4. **->** primary key**(**name**)**
5. **->** **);**
6. Query OK**,** **0** rows affected **(0.38** sec**)**

确认新创建的表：

1. mysql**>** show tables**;**
2. **+----------------+**
3. **|** Tables\_in\_mydb **|**
4. **+----------------+**
5. **|** pwlist **|**                                 //新建的pwlist表
6. **+----------------+**
7. **1** rows **in** set **(0.01** sec**)**

查看pwlist表的字段结构：

1. mysql**>** desc pwlist**;**
2. **+----------+----------+------+-----+---------+-------+**
3. **|** Field **|** Type **|** Null **|** Key **|** Default **|** Extra **|**
4. **+----------+----------+------+-----+---------+-------+**
5. **|** name **|** char**(16)** **|** NO **|** PRI **|** NULL **|** **|**
6. **|** password **|** char**(48)** **|** YES **|** **|** **|** **|**
7. **+----------+----------+------+-----+---------+-------+**
8. **2** rows **in** set **(0.01** sec**)**

4）删除指定的表

删除当前库中的pwlist表：

1. mysql**>** drop table pwlist**;**
2. Query OK**,** **0** rows affected **(0.01** sec**)**

确认删除结果：

1. mysql**>** show tables**;**
2. Empty set **(0.00** sec**)**

5）在mydb库中创建一个学员表

表格结构及数据内容如表-1所示。

在MySQL表内存储中文数据时，需要更改字符集（默认为latin1不支持中文），以便MySQL支持存储中文数据记录；比如，可以在创建库或表的时候，手动添加“DEFAULT CHARSET=utf8”来更改字符集。

如：create table db1.t1(内容) defaults charset=utf8；

根据上述表格结构，创建支持中文的student表：

Show create table db1.t1 \G 查看看建的数据表字符集类型。

1. mysql**>** CREATE TABLE mydb**.**student**(**
2. **->** 学号 char**(9)** NOT NULL**,**
3. **->** 姓名 varchar**(4)** NOT NULL**,**
4. **->** 性别 **enum(**'男'**,**'女'**)** NOT NULL**,**
5. **->** 手机号 char**(11)** DEFAULT ''**,**
6. **->** 通信地址 varchar**(64),**
7. **->** PRIMARY KEY**(**学号**)**
8. **->** **)** DEFAULT CHARSET**=**utf8**;**                 //手工指定字符集，采用utf8
9. Query OK**,** **0** rows affected **(0.**31sec**)**

查看student表的字段结构：

1. mysql**>** DESC mydb**.**student**;**
2. **+--------------+-------------------+------+-----+---------+-------+**
3. **|** Field **|** Type **|** Null **|** Key **|** Default **|** Extra **|**
4. **+--------------+-------------------+------+-----+---------+-------+**
5. **|** 学号 **|** char**(9)** **|** NO **|** PRI **|** NULL **|** **|**
6. **|** 姓名 **|** varchar**(4)** **|** NO **|** **|** NULL **|** **|**
7. **|** 性别 **|** **enum(**'男'**,**'女'**)** **|** NO **|** **|** NULL **|** **|**
8. **|** 手机号 **|** char**(11)** **|** YES **|** **|** **|** **|**
9. **|** 通信地址 **|** varchar**(64)** **|** YES **|** **|** NULL **|** **|**
10. **+--------------+-------------------+------+-----+---------+-------+**
11. **5** rows **in** set **(0.00** sec**)**

查看student表的实际创建指令：

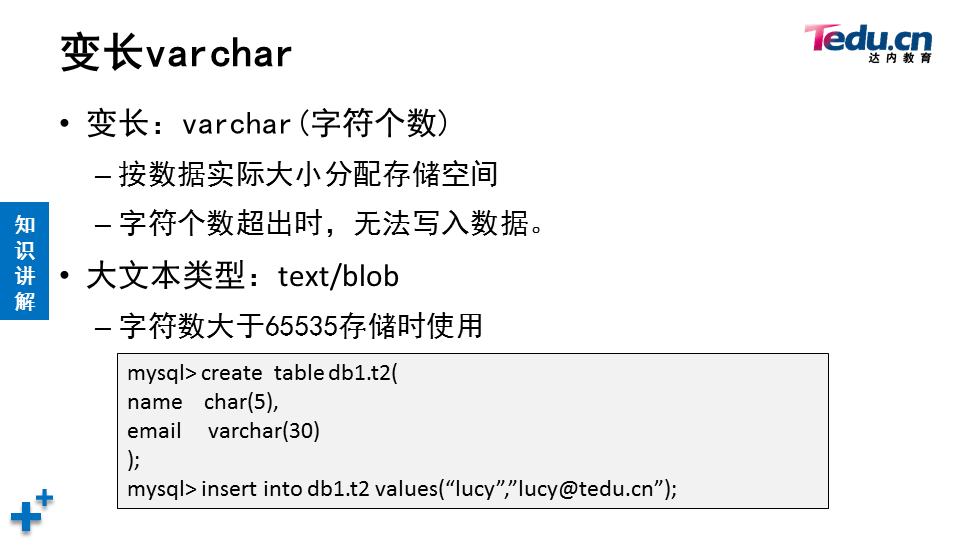
1. mysql**>** SHOW CREATE TABLE mydb**.**student**;**
2. **+---------+------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+**
3. **|**Table **|**Create Table **|**
4. **+---------+--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+**
5. **|** student **|** CREATE TABLE `student` **(**
6. `学号` char**(9)** NOT NULL**,**
7. `姓名` varchar**(4)** NOT NULL**,**
8. `性别` **enum(**'男'**,**'女'**)** NOT NULL**,**
9. `手机号` char**(11)** DEFAULT ''**,**
10. `通信地址` varchar**(64)** DEFAULT NULL**,**
11. PRIMARY KEY **(**`学号`**)**
12. **)** ENGINE**=**InnoDB DEFAULT CHARSET**=**utf8 **|**
13. **+---------+--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+**
14. **1** row **in** set **(0.00** sec**)**

***注意：若要修改MySQL服务的默认字符集，可以更改服务器的my.cnf配置文件，添加character\_set\_server=utf8 配置，然后重启数据库服务。***

1. **[**root@dbsvr1 **~]**# vim **/**etc**/**my**.**cnf                         //修改运行服务配置
2. **[**mysqld**]**
3. **..** **..**
4. character\_set\_server**=**utf8
5. **[**root@dbsvr1 **~]**# systemctl restart mysqld                 //重启服务
6. **..** **..**
7. **[**root@dbsvr1 **~]**# mysql –u root **-**p
8. Enter password**:**
9. **..** **..**
10. mysql**>** SHOW VARIABLES LIKE 'character%'**;**                 //确认更改结果
11. **+--------------------------+----------------------------+**
12. **|** Variable\_name **|** Value **|**
13. **+--------------------------+----------------------------+**
14. **|** character\_set\_client **|** utf8 **|**
15. **|** character\_set\_connection **|** utf8 **|**
16. **|** character\_set\_database **|** utf8 **|**
17. **|** character\_set\_filesystem **|** binary **|**
18. **|** character\_set\_results **|** utf8 **|**
19. **|** character\_set\_server **|** utf8 **|**
20. **|** character\_set\_system **|** utf8 **|**
21. **|** character\_sets\_dir **|** /usr/share**/**mysql**/**charsets**/** **|**
22. **+--------------------------+----------------------------+**
23. **8** rows **in** set **(0.03** sec**)**

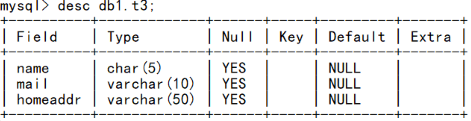
**3 案例3：字符类型**

****

****

**3.1 问题**

* 按照 图-1 所示建表。



图－1

**3.2 步骤**

实现此案例需要按照如下步骤进行。

**步骤一：创建a3表**

1）新建db1库，并切换到db1库

1. mysql**>** CREATE DATABASE db1**;**
2. Query OK**,** **1** row affected **(0.00** sec**)**
3. mysql**>** USE db1**;**
4. Database changed

2）新建t3表

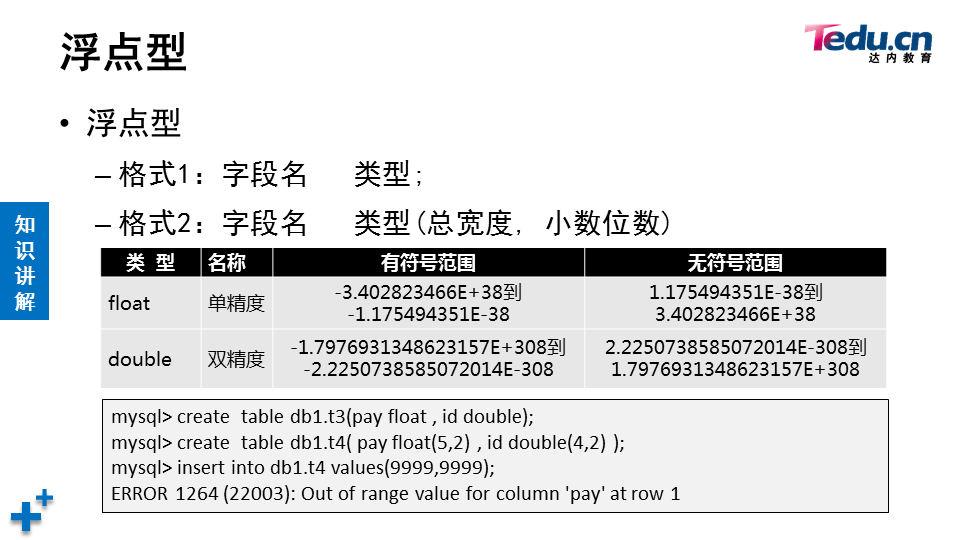
1. mysql**>** CREATE TABLE db1**.**t3 **(**
2. **->** name char**(5)** **,**
3. **->** mail varchar**(10),**
4. **->** homedir varchar**(50)**
5. **->** **);**
6. Query OK**,** **0** rows affected **(0.**61sec**)**

3) 查看a3表结构

1. mysql**>** DESC db1**.**a3**;**
2. **+----------+----------------------+------+-----+---------+-------+**
3. **|** Field **|** Type **|** Null **|** Key **|** Default **|** Extra **|**
4. **+----------+----------------------+------+-----+---------+-------+**
5. **|** name **|** char**(5)** **|** YES **|** **|** NULL **|** **|**
6. **|** mail **|** varchar**(10)** **|** YES **|** **|** NULL **|** **|**
7. **|** homedir **|** varchar**(50)** **|** YES **|** **|** NULL **|** **|**
8. **+----------+----------------------+------+-----+---------+-------+**
9. **3** rows **in** set **(0.00** sec**)**

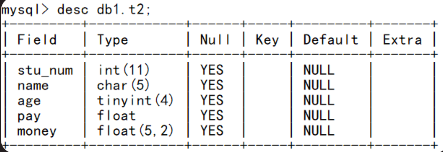
**4 案例4：数值类型**

****

****

**4.1 问题**

按照 图-2 所示建表。



图－2

**4.2 步骤**

实现此案例需要按照如下步骤进行。

**步骤一：创建t2表**

1）切换到db1库

1. mysql**>** USE db1**;**
2. Database changed

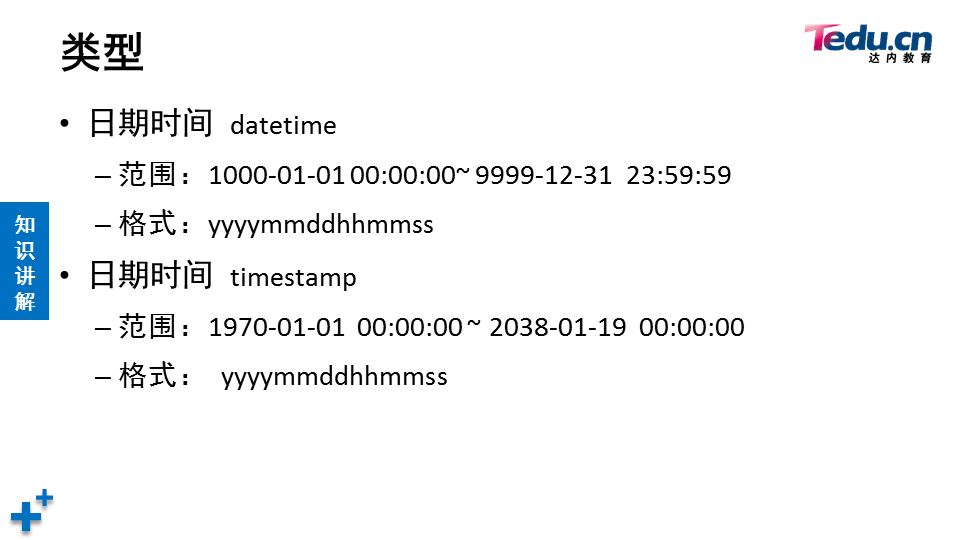
2）新建t2表

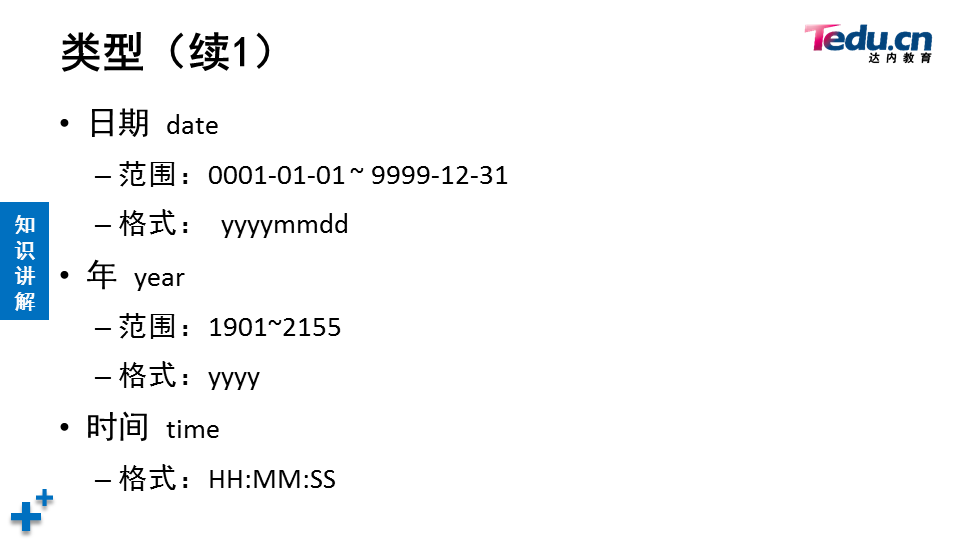
1. mysql**>** create table db1**.**t2**(**
2. **->** stu\_num int**,**
3. **->** name char**(5),**
4. **->** age tinyint**,**
5. **->** pay float**,**
6. **->** money float**(5,2)**
7. **->** **);**
8. Query OK**,** **0** rows affected **(0.03** sec**)**

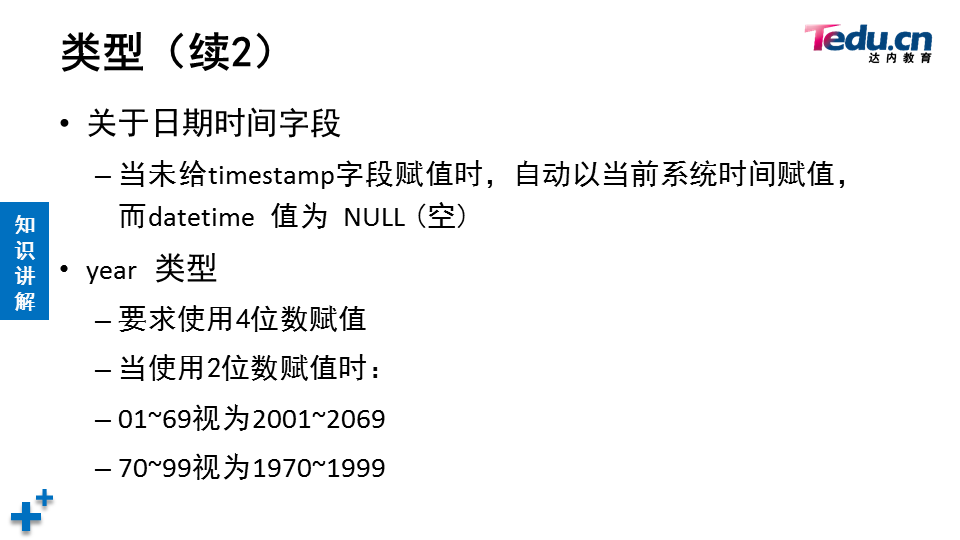
3) 查看t2表结构

1. mysql**>** desc db1**.**t2**;**
2. **+---------+------------+------+-----+---------+-------+**
3. **|** Field **|** Type **|** Null **|** Key **|** Default **|** Extra **|**
4. **+---------+------------+------+-----+---------+-------+**
5. **|** stu\_num **|** int**(11)** **|** YES **|** **|** NULL **|** **|**
6. **|** name **|** char**(5)** **|** YES **|** **|** NULL **|** **|**
7. **|** age **|** tinyint**(4)** **|** YES **|** **|** NULL **|** **|**
8. **|** pay **|** float **|** YES **|** **|** NULL **|** **|**
9. **|** money **|** float**(5,2)** **|** YES **|** **|** NULL **|** **|**
10. **+---------+------------+------+-----+---------+-------+**
11. **5** rows **in** set **(0.00** sec**)**
12. mysql**>**

**5 案例5：日期时间类型**

****

****

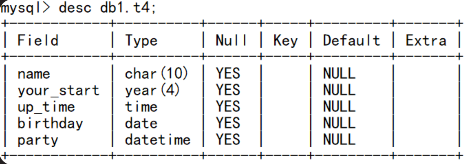
****

****

**5.1 问题**

练习如下时间函数的使用：

* now( ) year( ) month( ) day( ) date( ) time( )
* curtime( ) curdate( )
* 按照图-3所示建表



图－3

**5.2 步骤**

实现此案例需要按照如下步骤进行。

**步骤一：练习时间函数的使用**

1）使用now()查看当前的日期和时间

1. mysql**>** SELECT now**();**
2. **+---------------------+**
3. **|** now**()** **|**
4. **+---------------------+**
5. **|** **2019-07-03** **05:00:15** **|**
6. **+---------------------+**
7. **1** row **in** set **(0.00** sec**)**
8. mysql**>**

2）使用curdate()获得当前的日期

1. mysql**>** SELECT curdate**();**
2. **+------------+**
3. **|** curdate**()** **|**
4. **+------------+**
5. **|** **2019-07-03** **|**
6. **1** row **in** set **(0.00** sec**)**
7. mysql**>**

3）使用curtime()获得当前的时间

1. mysql**>** SELECT curtime**();**
2. **+-----------+**
3. **|** curtime**()** **|**
4. **+-----------+**
5. **|** **04:04:55** **|**
6. **+-----------+**
7. **1** row **in** set **(0.00** sec**)**

4）分别获取当前日期时间中的年份、月份、日

1. mysql**>** SELECT year**(**now**())** **,** month**(**now**())** **,** day**(**now**());**
2. **+-------------+--------------+------------+**
3. **|** year**(**now**())** **|** month**(**now**())** **|** day**(**now**())** **|**
4. **+-------------+--------------+------------+**
5. **|** **2019** **|** **7** **|** **3** **|**
6. **+-------------+--------------+------------+**
7. **1** row **in** set **(0.00** sec**)**
8. mysql**>**

5）获取系统日期

1. mysql**>** select date**(**now**());**
2. **+-------------+**
3. **|** date**(**now**())** **|**
4. **+-------------+**
5. **|** **2019-07-03** **|**
6. **+-------------+**
7. **1** row **in** set **(0.00** sec**)1** row **in** set **(0.00** sec**)**
8. Mysql**>**

**步骤二：创建t4表**

1）建表

1. mysql**>** create table db1**.**t4**(**
2. **->** name char**(10),**
3. **->** your\_start year**,**
4. **->** up\_time time**,**
5. **->** birthday date**,**
6. **->** party datetime
7. **->** **);**
8. Query OK**,** **0** rows affected **(0.04** sec**)**
9. mysql**>**

2) 查看表结构

1. Mysql**>**
2. mysql**>** desc db1**.**t4**;**
3. **+------------+----------+------+-----+---------+-------+**
4. **|** Field **|** Type **|** Null **|** Key **|** Default **|** Extra **|**
5. **+------------+----------+------+-----+---------+-------+**
6. **|** name **|** char**(10)** **|** YES **|** **|** NULL **|** **|**
7. **|** your\_start **|** year**(4)** **|** YES **|** **|** NULL **|** **|**
8. **|** up\_time **|** time **|** YES **|** **|** NULL **|** **|**
9. **|** birthday **|** date **|** YES **|** **|** NULL **|** **|**
10. **|** party **|** datetime **|** YES **|** **|** NULL **|** **|**
11. **+------------+----------+------+-----+---------+-------+**
12. **5** rows **in** set **(0.00** sec**)**
13. mysql**>**

3）插入记录

1. mysql**>**
2. mysql**>** insert into db1**.**t4 values**(**"bob"**,1990,083000,20191120,2019082820000);**
3. Query OK**,** **1** row affected**,** **1** warning **(0.01** sec**)**
4. mysql**>** insert into db1**.**t4 values**(**"tom"**,1991,090000,20191120,**now**());**
5. Query OK**,** **1** row affected **(0.02** sec**)**
6. mysql**>**

4）查看表记录

mysql>

mysql> select \* from db1.t4;

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

| name | your\_start | up\_time | birthday | party |

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

| bob | 1990 | 08:30:00 | 2019-11-20 | 0000-00-00 00:00:00 |

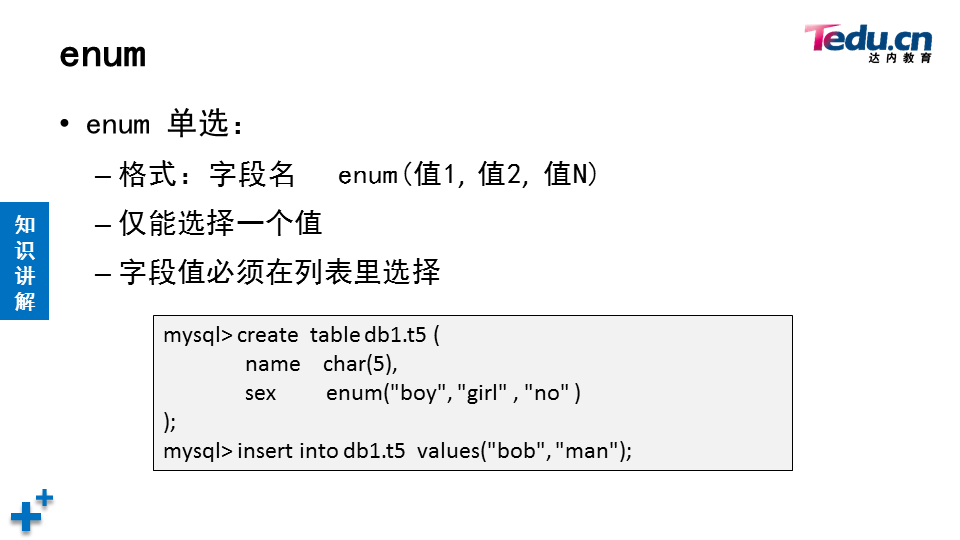
| tom | 1991 | 09:00:00 | 2019-11-20 | 2019-07-03 05:12:41 |

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

2 rows in set (0.00 sec)

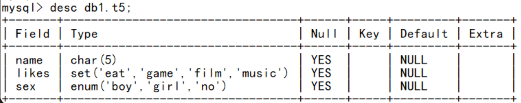
mysql>

**6 案例6：枚举类型**

****

**6.1 问题**

* 按照图-4所示建表



图－4

**6.2 步骤**

实现此案例需要按照如下步骤进行。

**步骤一：创建t5表**

1）建表

1. mysql**>**
2. mysql**>** create table db1**.**t5 **(**
3. **->** name char**(5),**
4. **->** likes set**(**"eat"**,**"game"**,**"film"**,**"music"**),**
5. **->** sex **enum(**"boy"**,**"girl"**,**"no"**)**
6. **->** **);**
7. Query OK**,** **0** rows affected **(0.04** sec**)**
8. Mysql**>**

2）查看表结构

1. mysql**>**
2. mysql**>** desc db1**.**t5**;**
3. **+-------+----------------------------------+------+-----+---------+-------+**
4. **|** Field **|** Type **|** Null **|** Key **|** Default **|** Extra **|**
5. **+-------+----------------------------------+------+-----+---------+-------+**
6. **|** name **|** char**(5)** **|** YES **|** **|** NULL **|** **|**
7. **|** likes **|** set**(**'eat'**,**'game'**,**'film'**,**'music'**)** **|** YES **|** **|** NULL **|** **|**
8. **|** sex **|** **enum(**'boy'**,**'girl'**,**'no'**)** **|** YES **|** **|** NULL **|** **|**
9. **+-------+----------------------------------+------+-----+---------+-------+**
10. **3** rows **in** set **(0.00** sec**)**
11. mysql**>**

3）插入表记录

1. mysql**>**
2. mysql**>** insert into db1**.**t5 values **(**"bob"**,**"eat,film,game"**,**"boy"**);**
3. Query OK**,** **1** row affected **(0.03** sec**)**
4. mysql**>**

4）查看表记录

1. mysql**>** select **\*** from db1**.**t5**;**
2. **+------+---------------+------+**
3. **|** name **|** likes **|** sex **|**
4. **+------+---------------+------+**
5. **|** bob **|** eat**,**game**,**film **|** boy **|**
6. **+------+---------------+------+**
7. **1** rows **in** set **(0.00** sec**)**