

DBA基础

NSD DBA1

DAY01

内容

上午	09:00 ~ 09:30	数据库服务概述
	09:30 ~ 10:20	构建MySQL服务器
	10:30 ~ 11:20	
	11:30 ~ 12:00	数据库基本管理
下午	14:00 ~ 14:50	MySQL数据类型
	15:00 ~ 15:50	
	16:00 ~ 17:00	
	17:10 ~ 18:00	总结和答疑



数据库服务概述

数据库服务概述

什么是数据库?

什么是数据库

E-R 数据模型

常见的数据库服务软件

MySQL数据库介绍

MySQL的起源与发展过程

MySQL的特点及应用

什么是数据库？

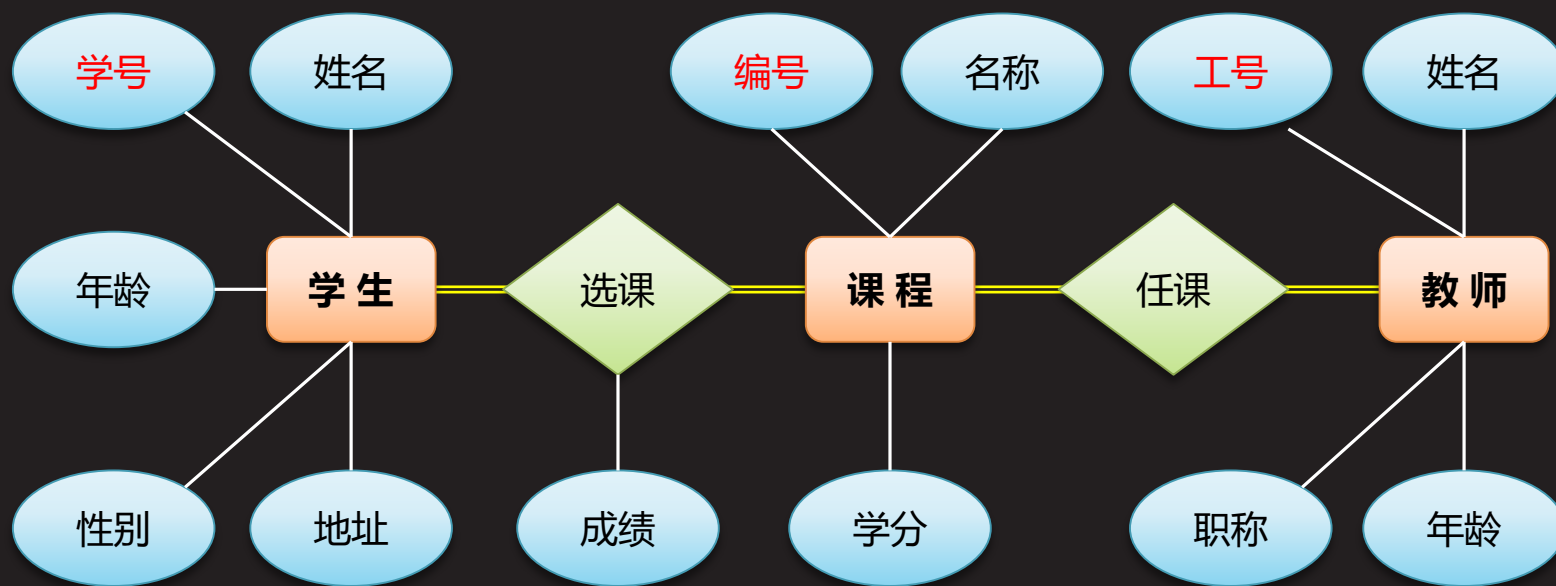
什么是数据库

- DB, DataBase
 - 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, DataBase Management System
 - 数据库管理系统：用来操纵和管理数据库的大型服务软件
- DBS, DataBase System
 - 数据库系统：即 DB+DBMS，指带有数据库并整合了数据库管理软件的计算机系统



E-R 数据模型

- 实体-关系 模型
 - Entity-Relationship Model



常见的数据库服务软件

类 型	厂 商
Oracle Database	Oracle（甲骨文）
MySQL/MariaDB	
SQL Server	Microsoft（微软）
Access	
DB2	IBM
Sybase	Sybase
PostgreSQL	加州大学伯克利分校



MySQL数据库介绍

MySQL的起源与发展过程

- 最为著名、应用最广泛的开源数据库软件
 - 最早隶属于瑞典的MySQL AB公司
 - 2008年1月, MySQL AB被Sun收购
 - 2009年4月, SUN被Oracle收购
- 崭新的开源分支 MariaDB
 - 为应付 MySQL 可能会闭源的风险而诞生
 - 由MySQL原作者 Widenius 主导开发
 - 与MySQL保持最大程度兼容

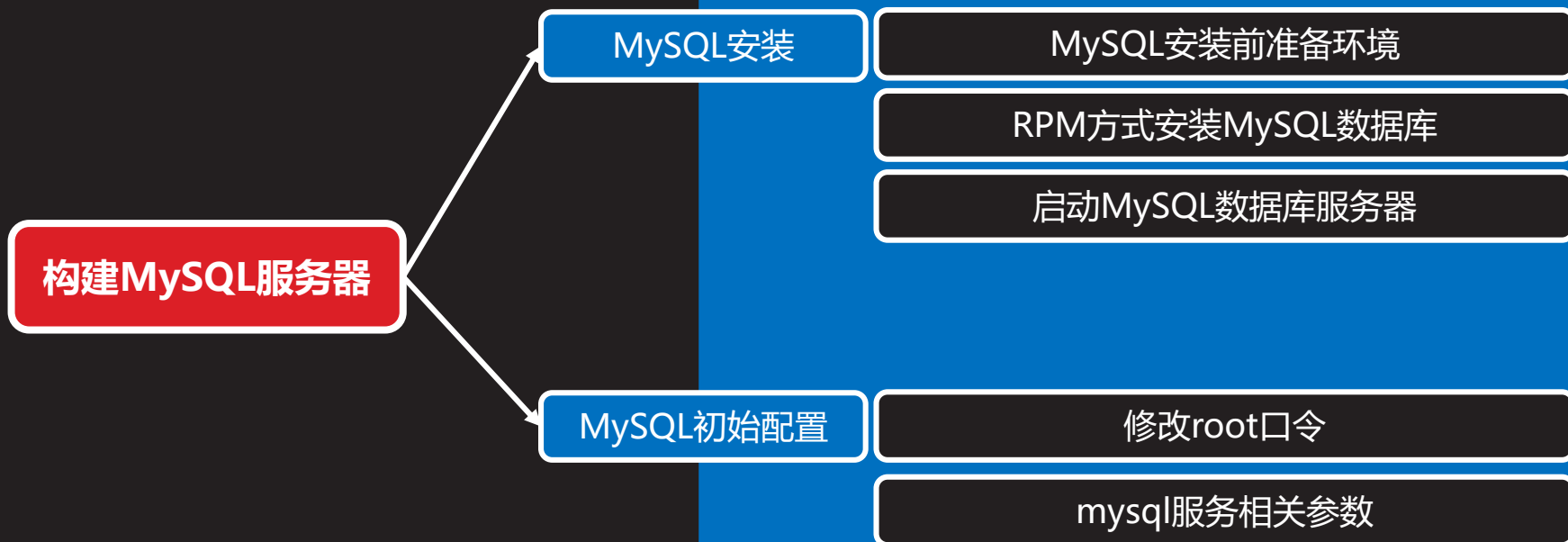


MySQL的特点及应用

- 主要特点
 - 适用于中小规模、关系型数据库系统
 - 支持Linux/Unix、Windows等多种操作系统
 - 使用C和C++编写，可移植性强
 - 通过API支持Python/Java/Perl/PHP等语言
- 典型应用环境
 - LAMP平台，与Apache HTTP Server组合
 - LNMP平台，与Nginx组合



构建MySQL服务器



MySQL安装



安装MySQL

- 基本需求
 - 1. 采用RHEL7.2系统搭建MySQL服务器
 - 2. 关闭iptables
 - 3. 关闭selinux
 - 4. 软件 mysql-5.7.17-1



安装MySQL (续1)

- 从官网下载RPM包
 - <http://dev.mysql.com/downloads/mysql/>
 - 选择适用于当前系统的 bundle 集合包

mysql-community-client //客户端应用程序
 mysql-community-common //数据库和客户端库共享文件
 mysql-community-devel //客户端应用程序的库和头文件
 mysql-community-embedded //嵌入式函数库
 mysql-community-embedded-compat //嵌入式兼容函数库
 mysql-community-embedded-devel //头文件和库文件作为Mysql的嵌入式库文件
 mysql-community-libs //MySQL 数据库客户端应用程序的共享库
 mysql-community-libs-compat //客户端应用程序的共享兼容库



安装MySQL (续2)

- 准备工作
 - 停止mariadb服务
 - 删除文件 /etc/my.cnf
 - 删除数据
 - 卸载软件包

```
[root@localhost ~]# systemctl stop mariadb
[root@localhost ~]# rm -rf /etc/my.cnf
[root@localhost ~]# rm -rf /var/lib/mysql/*
[root@localhost ~]# rpm -e --nodeps mariadb-server mariadb
警告: /var/log/mariadb/mariadb.log 已另存为
/var/log/mariadb/mariadb.log.rpmsave
[root@localhost ~]#
```



安装MySQL (续3)

- 至少安装server、client、share* 包
 - 采用U升级安装，可替换冲突文件
 - 推荐将devel也装上，用于支持其他软件

```
[root@localhost ~]# yum -y install perl-Data-Dumper perl-JSON  
perl-Time-HiRes
```

```
[root@localhost ~]# tar -xf mysql-5.7.17-1.el7.x86_64.rpm-  
bundle.tar
```

```
[root@localhost ~]# rpm -Uvh mysql-community-*.rpm
```



启动MySQL数据库服务

- 服务脚本为
/usr/lib/systemd/system/mysqld.service

```
[root@localhost ~]# systemctl enable mysqld
```

```
[root@localhost ~]# systemctl start mysqld
```

```
[root@localhost ~]# systemctl status mysqld
```

- mysqld.service - MySQL Server

Loaded: loaded (/usr/lib/systemd/system/mysqld.service;
enabled; vendor preset: disabled)

Active: active (running) since Tue 2017-04-18 15:29:56 CST;
1min 12s ago

.....

Main PID: 15112 (mysqld)

CGroup: /system.slice/mysqld.service

└─15112 /usr/sbin/mysqld --daemonize --pid-
file=/var/run/mysqld/mysqld.pid



MySQL初始配置



修改root口令

- 默认的数据库管理账号
 - root, 允许从 localhost 访问
 - 首次登录密码在安装时随机生成
 - 存储在错误日志文件里

```
[root@localhost ~]# grep 'temporary password'
/var/log/mysqld.log
2017-04-18T07:29:27.634784Z 1 [Note] A temporary password is
generated for root@localhost: mtoa>Av<p6Yk
[root@localhost ~]#
```

随机生成的管理密码

```
[root@localhost ~]# mysql -uroot -p'mtoa>Av<p6Yk'
mysql>
```



修改root口令 (续1)

validate_password_policy 验证密码策略	
0 or LOW	长度
1 or MEDIUM(默认)	长度; 数字, 小写/大写, 和特殊字符
2 or STRONG	长度; 数字, 小写/大写和特殊字符; 字典文件

```
mysql> set global validate_password_policy=0; //只验证长度
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> set global validate_password_length=6; //修改密码长度
默认值是8个字符
```



修改root口令 (续2)

- 操作方法
 - 通过客户端工具 mysql 连接本服务器
 - 使用 alter user 重设口令

```
mysql> alter user user() identified by "123456"; //修改登陆密码  
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> quit
```

```
[root@localhost ~]# mysql -uroot -p123456 //使用新密码登陆
```



mysql服务相关参数

文件	说明
/etc/my.cnf	主配置文件
/var/lib/mysql	数据库目录
默认端口号	3306
进程名	mysqld
传输协议	TCP
进程所有者	mysql
进程所属组	mysql



案例1：构建MySQL服务器

1. 安装MySQL-server、MySQL-client
2. 修改数据库用户root的密码
3. 确认MySQL服务程序运行、root可控



数据库基本管理

数据库基本管理

```
graph LR; A[数据库基本管理] --> B[MySQL连接工具]; A --> C[MySQL基本操作]; B --> D[数据存储流程]; B --> E[连接MySQL服务器]; B --> F[使用 mysql> 管理环境]; C --> G[库管理命令]; C --> H[表管理命令]; C --> I[记录管理命令];
```

MySQL连接工具

数据存储流程

连接MySQL服务器

使用 mysql> 管理环境

MySQL基本操作

库管理命令

表管理命令

记录管理命令

MySQL连接工具



数据存储流程

- 客户端把数据存储到服务器上的步骤
 - 连接数据库服务器
 - 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
 - 断开连接



连接MySQL服务器

- 使用 mysql 命令
 - mysql [-h服务器 -u用户名 -p密码 数据库]
 - quit 或 exit 退出

```

root@dbsvr1:~
[root@dbsvr1 ~]# mysql -h 127.0.0.1 -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5
Server version: 5.6.15 MySQL Community Server (GPL)

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> exit
Bye
[root@dbsvr1 ~]#
    
```



使用 mysql> 管理环境

- 操作指令类型
 - MySQL指令：环境切换、看状态、退出等控制
 - SQL指令：数据库定义/查询/操纵/授权语句
- 基本注意事项
 - 操作指令不区分大小写（密码、变量值除外）
 - 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



使用 mysql> 管理环境 (续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 数据事物语言 (commit、rollback、savepoint)



MySQL基本操作



库管理命令

- 库 类似于系统的文件夹
 - Show databases; //显示已有的库
 - Use 库名; //切换库
 - Select database(); //显示当前所在的库
 - Create database 库名; //创建新库
 - Show tables; //显示已有的表
 - Drop database 库名; //删除库



库管理命令(续1)

- 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



表管理命令

- 新建指定名称的表
 - CREATE TABLE 库名.表名(
 字段名1 字段类型(宽度) 约束条件,
 字段名2 字段类型(宽度) 约束条件,
 ..
 字段名N 字段类型(宽度) 约束条
);



表管理命令(续1)

- 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



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



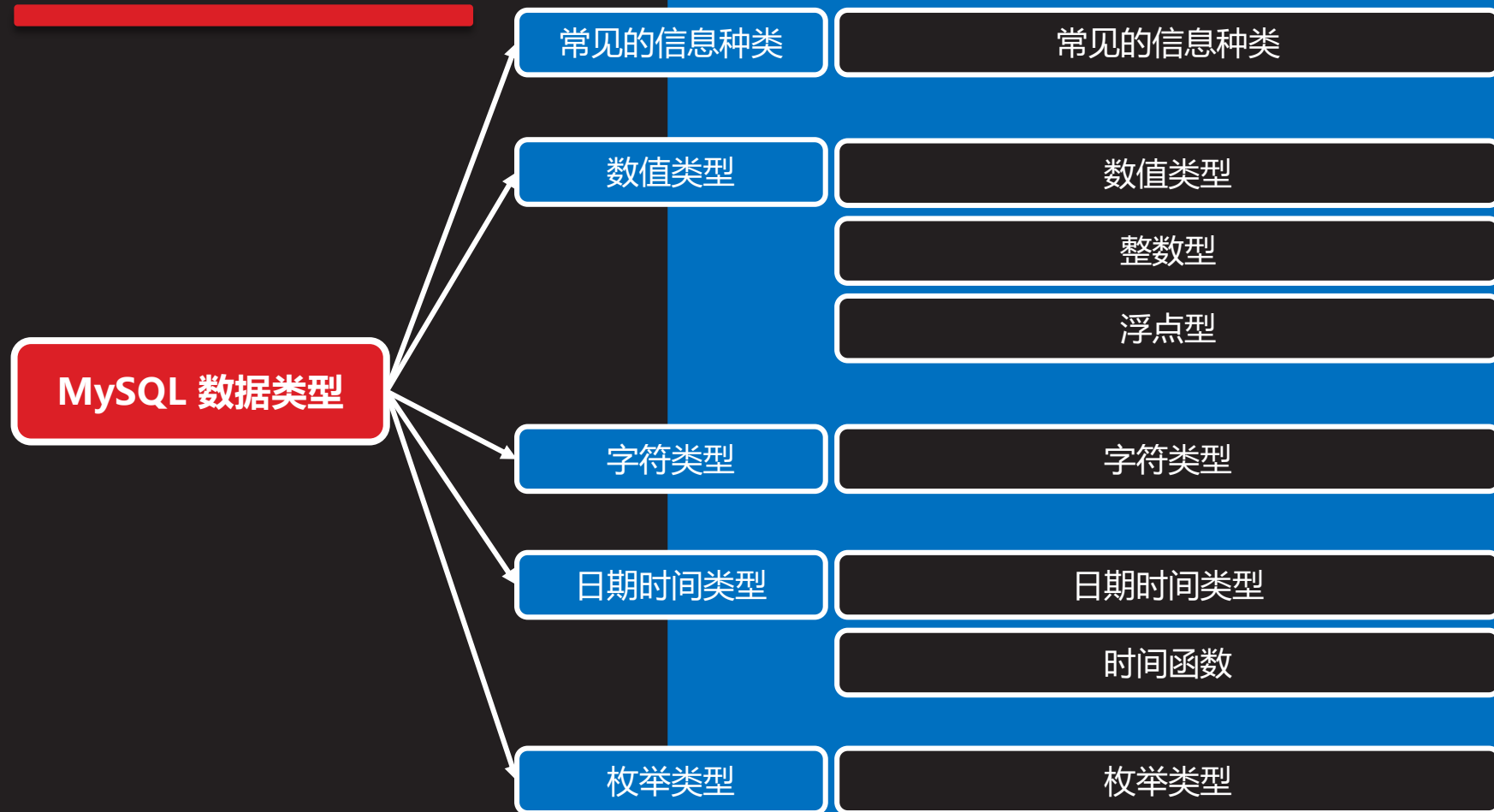
案例2：数据库基本管理

1. 使用mysql命令连接数据库
2. 查看/删除/创建库
3. 查看/删除/创建表（参考图示表格）

学号	姓名	性别	手机号	通信地址
NSD131201	张三	男	13012345678	朝阳区劲松南路
NSD131202	韩梅梅	女	13722223333	海淀区北三环西路
NSD131203	王五	男	18023445678	丰台区兴隆中街



MySQL 数据类型



常见的信息种类

常见的信息种类

- 数值型：体重、身高、成绩、工资
- 字符型：姓名、工作单位、通信住址
- 枚举型：兴趣爱好、性别
- 日期时间型：出生日期、注册时间



数值类型

数值类型

类 型	大 小	范围 (有符号)	范围 (无符号)	用 途
TINYINT	1字节	-128~127	0 ~ 255	微小整数
SMALLINT	2字节	-32768~32767	0 ~ 65535	小整数
MEDIUMINT	3字节	$-2^{23} \sim 2^{23}-1$	$0 \sim 2^{24}-1$	中整数
INT	4字节	$-2^{31} \sim 2^{31}-1$	$0 \sim 2^{32}-1$	大整数
BIGINT	8字节	$-2^{63} \sim 2^{63}-1$	$0 \sim 2^{64}-1$	极大整数
FLOAT	4字节			单精度浮点数
DOUBLE	8字节			双精度浮点数
DECIMAL	对DECIMAL(M,D), 其中M为有效位数、D为小数位数, M应大于D, 占用M+2字节			



整数型

- 关于整数型字段
 - 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，在左边填充空格补位
 - 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。



整数型（续1）

- 了解表的字段设置
 - DESC 或 DESCRIBE，查看表结构

```
mysql> USE test;
Database changed
mysql> CREATE TABLE t1(id tinyint unsigned zerofill);
Query OK, 0 rows affected (0.06 sec)
```

```
mysql> DESC t1;
```

Field	Type	Null	Key	Default	Extra
id	tinyint(3) unsigned zerofill	YES		NULL	

1 row in set (0.00 sec)

字段名 | 字段类型 | 是否为空 | 是否为主键 | 默认值 | 描述信息



浮点型

- 关于浮点型字段
 - 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 t2(pay float(7,2));
Query OK, 0 rows affected (0.06 sec)
```

```
mysql> DESC t2;
```

Field	Type	Null	Key	Default	Extra
pay	float(7,2)	YES		NULL	

```
1 row in set (0.04 sec)
```



字符类型

字符类型

- 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



日期时间类型

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



日期时间类型 (续1)

- 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



日期时间类型 (续2)

- 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2001~2069，而70~99视为1970~1999



日期时间类型 (续3)

- 创建一个学员表
 - 包括姓名、入学年份、生日、培训时间段

```
mysql> CREATE TABLE stu_info(
    -> name varchar(8),
    -> starty year,
    -> birth date,
    -> ttime1 time,
    -> ttime2 time
    -> );
Query OK, 0 rows affected (0.07 sec)
```



时间函数

类 型	用 途
now()	获取日期和时间
year()	获取年
sleep(N)	休眠N秒
curdate()	获取当前的系统日期
curtime()	获取当前的系统时刻
month()	获取指定时间中的月份
date()	获取指定时间中的日期
time()	获取指定时间中的时刻



时间函数（续1）

- 无需库、表，可直接调用
 - 使用SELECT指令输出函数结果

```
mysql> SELECT now(),sysdate(),curdate();
```

now()	sysdate()	curdate()
2013-12-27 10:42:16	2013-12-27 10:42:16	2013-12-27

1 row in set (0.00 sec)

```
mysql> SELECT date(now()),time(now());
```

date(now())	time(now())
2013-12-27	10:42:58

1 row in set (0.00 sec)



枚举类型

枚举类型

- 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(4),
    -> gender enum("boy","girl"),
    -> interest set("book","film","music","football")
    -> );
Query OK, 0 rows affected (0.38 sec)
```



案例3：MySQL 数据类型

1. 在studb库里创建stuinfo表，
2. 表结构自定义，并合理使用数据类型。



总结和答疑

总结和答疑

安装MySQL

问题现象1

故障分析及排除

问题现象2

故障分析及排除

MySQL基本使用

问题现象1

故障分析及排除

问题现象2

故障分析及排除

安装MySQL



问题现象1

- 执行升级安装失败

- 报错: `libaio.so.1()(64bit)` is
`libaio.so.1(LIBAIO_0.4)(64bit)` is

```
[root@dbsvr1 ~]# rpm -Uvh MySQL-*.rpm
```

```
error: Failed dependencies:
```

```
    libaio.so.1()(64bit) is needed by MySQL-  
embedded-5.6.15-1.el6.x86_64
```

```
    libaio.so.1(LIBAIO_0.1)(64bit) is needed by MySQL-  
embedded-5.6.15-1.el6.x86_64
```

```
    libaio.so.1(LIBAIO_0.4)(64bit) is needed by MySQL-
```



故障分析及排除

- 原因分析
 - 安装的RPM包有依赖
 - 安装的RPM包和系统自带的软件包 版本不匹配
- 解决办法
 - 安装依赖的软件

```
[root@dbsvr1 ~]#yum -y install libaio //安装依赖的包
[root@dbsvr1 ~]# rpm -q libaio //查看是否安装成功
libaio-0.3.107-10.el6.x86_64
```



问题现象2

- 安装依赖包后，执行升级安装依然失败
 - 报错： A MySQL server package (mysql-server-5.1.73-5.el6_6.x86_64) is installed

```
[root@dbsvr1 ~]# rpm -Uvh MySQL-*.rpm
```

```
...
```

```
error: %pre(MySQL-server-5.6.15-1.el6.x86_64) scriptlet failed,  
exit status 1
```

```
error: install: %pre scriptlet failed (2), skipping MySQL-server-  
5.6.15-1.el6
```



故障分析及排除

- 原因分析：
 - 系统已经安装了提供MySQL数据库服务器的软件包
 - 并且和当前安装的软件不兼容，所以升级失败
- 解决办法：
 - 卸载不兼容的低版本软件包，再单独安装高版本的提供数据库服务的软件包

```
[root@dbsvr1 ~]# rpm -qa | grep -i mysql-server
mysql-server-5.1.73-5.el6_6.x86_64
[root@dbsvr1 ~]# rpm -e --nodeps mysql-server
[root@dbsvr1 ~]# rpm -ivh MySQL-server-5.6.15-1.el6.x86_64.rpm
```



MySQL基本使用



问题现象1

- 创建新数据库失败
 - 报错: ERROR 1007 (HY000): Can't create

```
mysql> create database test;  
ERROR 1007 (HY000): Can't create database 'test'; database exists  
mysql>
```



故障分析及排除

- 原因分析
 - 提示数据库已经存在,
 - 没有遵守建库规则, 数据库名不符合唯一性
- 解决办法
 - 检查现有库, 新建库时使用其他名字

```
mysql> show databases;
| test          |
...
mysql> create database test_1;
Query OK, 1 row affected (0.07 sec)
```



问题现象2

- 登录数据库系统后，创建新表失败
 - 报错：ERROR 1046 (3D000): ...

```
[root@dbsvr1 ~]# mysql -uroot -p123
```

```
mysql> create table stu_tab(name char(10));  
ERROR 1046 (3D000): No database selected
```



故障分析及排除

- 原因分析
 - 建表之前没有选择库，导致无法确定创建位置
- 解决办法
 - 提前选库，或者建表时指定数据库

```
mysql> create table test_1.stu_tab(name char(10));  
Query OK, 0 rows affected (0.73 sec)
```

