

# 数据库基础

**NSD RDBMS1**

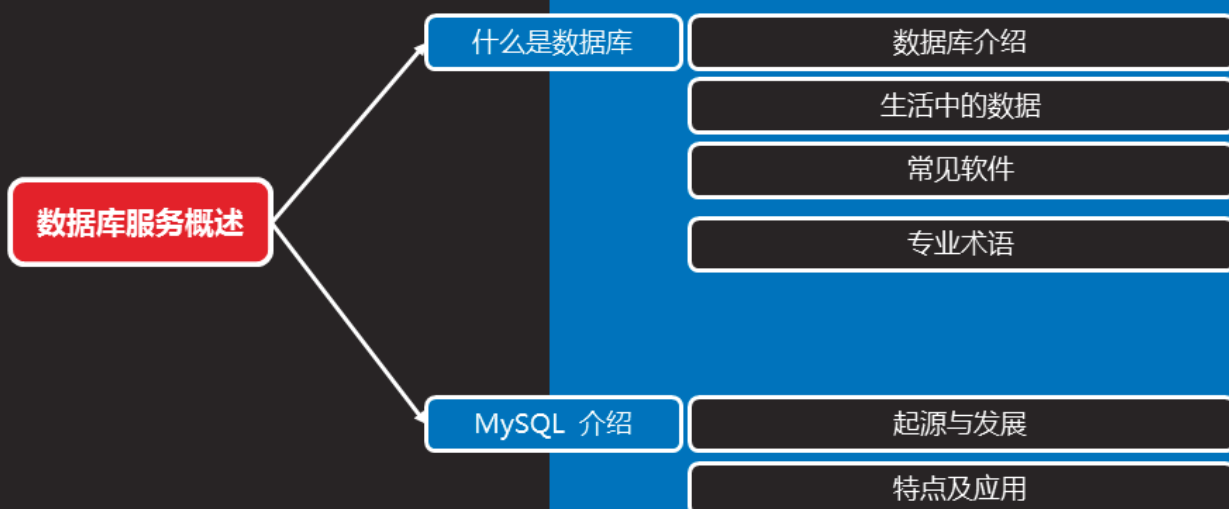
**DAY01**

# 内容

上午	09:00 ~ 09:30	数据库服务概述
	09:30 ~ 10:20	构建MySQL服务
	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



# 什么是数据库

## 数据库介绍

- 存储数据的仓库

知识讲解



## 生活中的数据

- 视频、音频、图片、文本

知识讲解



## 常见软件

- 主流操作系统：Unix、Linux、Windows

知识讲解

软件名	开源	跨平台	厂商
Oracle	否	是	甲骨文
MySQL	是	是	甲骨文
SQL Server	否	否	微软
DB2	否	是	IBM
Redis	是	是	开源软件
Memcache	是	是	开源软件
MongoDB	是	是	开源软件



## 专业术语

知识讲解

- **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



## MySQL 介绍

---

## 起源与发展

知识讲解

- 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



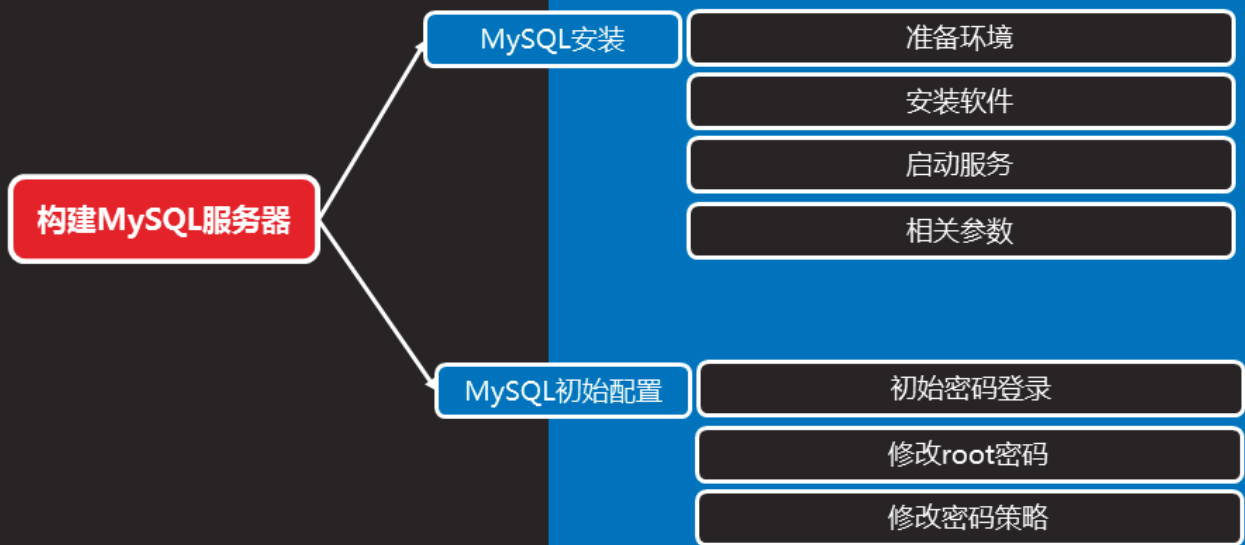
## 特点及应用

知识讲解

- 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
  - 支持Python、Java、Perl、PHP等编程语言
- 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

## 环境准备

知识讲解

- 基本需求
  1. 创建CentOS系统虚拟机1台
  2. 配置IP地址192.168.4.50
  3. 关闭firewalld
  4. 禁用SELinux
  5. 拷贝软件 mysql-5.7.17.tar
  6. 配置yum 源



## 安装软件

知识讲解

- 从官网下载RPM包
  - <http://dev.mysql.com/downloads/mysql>

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





## 安装软件（续1）

- 必须安装的软件包server、client
  - yum安装，自动解决软件依赖

知识讲解

```
]# tar -xvf mysql-5.7.17.tar
```

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

```
]# rpm -qa | grep -i mysql
```



## 相关参数

- 软件安装后自动创建相关目录与文件

知识讲解

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
错误日志文件	/var/log/mysqld.log



# 启动服务

知识讲解

- 首次启动服务，会执行数据初始化

```
]# systemctl start    mysqld    //启动服务  
]# systemctl enable   mysqld    //开机运行
```

```
]# systemctl status   mysqld    //查看状态
```

- 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
```

```
.....
```

```
]# netstat -utnlp | grep :3306 //查看服务信息  
]# ps -C mysqld                //查看进程
```



## MySQL初始配置

## 初始密码登录

知识讲解

- 数据库管理员名为 root
  - 默认仅允许root本机连接
  - 首次登录密码在安装软件时随机生成
  - 随机密码存储在日志文件 /var/log/mysqld.log 里
  - 连接命令 ]# mysql -h数据库地址 -u用户 -p密码

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

```
]# mysql -hlocalhost -uroot -p' mtoa>Av<p6Yk ' //连接
```



## 修改root密码

知识讲解

- 具体操作如下：
  - 使用 alter user 命令修改登录密码
  - 新密码必须满足密码策略
  - 管理员root 使用新密码连接服务

```
//修改登陆密码
```

```
mysql> alter user root@" localhost " identified by "密码";
```

```
]# mysql -hlocalhost -uroot -p密码 //使用新密码登陆
```



## 修改密码策略

知识讲解

策略名称	验证方式
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> show variables like "%password%" ; //查看变量
mysql> set global validate_password_policy=0; //修改密码策略
mysql> set global validate_password_length=6; //修改密码长度
```

```
]# vim /etc/my.cnf //永久配置
[mysqld]
validate_password_policy=0
validate_password_length=6
:wq
```



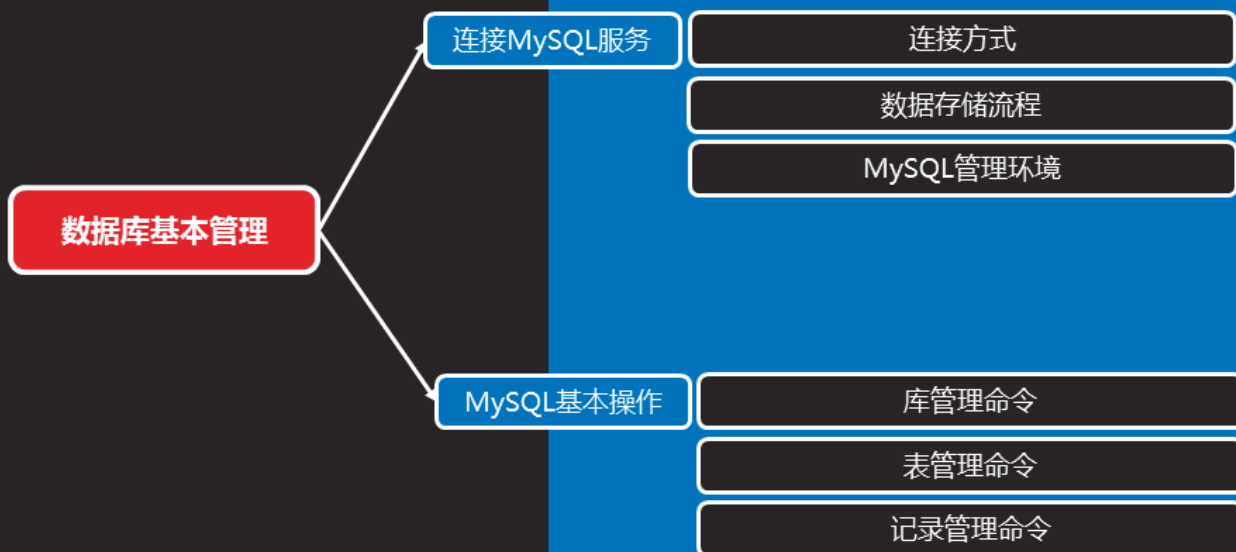
## 案例1：构建MySQL服务器

- 要求如下：
  - 在IP地址192.168.4.51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

课堂练习



## 数据库基本管理



## 连接MySQL服务

## 连接方式

知识讲解

- 客户端连接MySQL服务的方法
  - ① 命令行
  - ② web页面
  - ③ 安装图形软件
  - ④ 编写脚本（php、Java、python .....）
- 使用 mysql 命令
  - mysql -h服务器IP -u用户名 -p密码 [ 数据库名 ]
  - quit 或 exit 退出



## 数据存储流程

知识讲解

- 客户端把数据存储到数据库服务器上的步骤
  - ① 连接数据库服务器
  - ② 建库 //类似于文件夹
  - ③ 建表 //类似于文件
  - ④ 插入记录 //类似于文件内容
  - ⑤ 断开连接



# mysql管理环境

知识讲解

- SQL命令使用规则
  - SQL命令不区分字母大小写（密码、变量值除外）
  - 每条SQL命令以；结束
  - 默认命令不支持Tab键自动补齐
  - \c 终止sql命令



## MySQL管理环境（续1）

知识讲解

- 常用的SQL命令分类
  - 管理数据库使用SQL（结构化查询语言）
  - ① DDL 数据定义语言 如：create、alter、drop
  - ② DML 数据操作语言 如：insert、update、delete
  - ③ DCL 数据控制语言 如：grant、revoke
  - ④ DTL 数据事物语言 如：commit、rollback、savepoint



# MySQL基本操作

## 库管理命令

- 库 类似于文件夹，用来存储表
  - 可以创建多个库，通过库名区分
  - `show databases;` //显示已有的库
  - `select user( );` //显示连接用户
  - `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)

知识讲解

- 库名 命名规则
  - 仅可以使用数字、字母、下划线、不能纯数字
  - 区分字母大小写，具有唯一性
  - 不可使用指令关键字、特殊字符

```
mysql> create database DB1;  
Query OK, 1 row affected (0.06 sec)
```

```
mysql> create database db1;  
Query OK, 1 row affected (0.03 sec)
```



## 表管理命令

知识讲解

- 建表
    - 表存储数据的文件。
- ```
Mysql> create table 库名.表名(  
    字段名1 类型(宽度),  
    字段名2 类型(宽度),  
    ....
```

```
) DEFAULT CHARSET=utf8 ; //指定中文字符集，可以给字  
段赋值中文
```

```
mysql> create table db1.stuinfo(  
    name char(15),  
    homeaddr char(20)  
);
```



## 表管理命令(续1)

知识讲解

- 表 类似于文件
  - desc 库名.表名; //查看表结构
  - drop table 库名.表名; //删除表

```
mysql>
mysql> desc db1.stuinfo;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| name  | char(15) | YES | | NULL | |
| homeaddr | char(20) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```



## 记录管理命令

知识讲解

- 记录 类似于文件里的行
  - select \* from 库名.表名; //查看表记录
  - insert into 库名.表名 values(值列表); //插入表记录
  - update 库名.表名 set 字段=值; //修改表记录
  - delete from 表名; //删除表记录

```
mysql> insert into db1.stuinfo values("jim","usa"),("lilei","china");
```

```
mysql> select * from db1.stuinfo;
```

```
Mysql> update db1.stuinfo set homeaddr= "beijing" ;
```



## 案例2：数据库基本管理

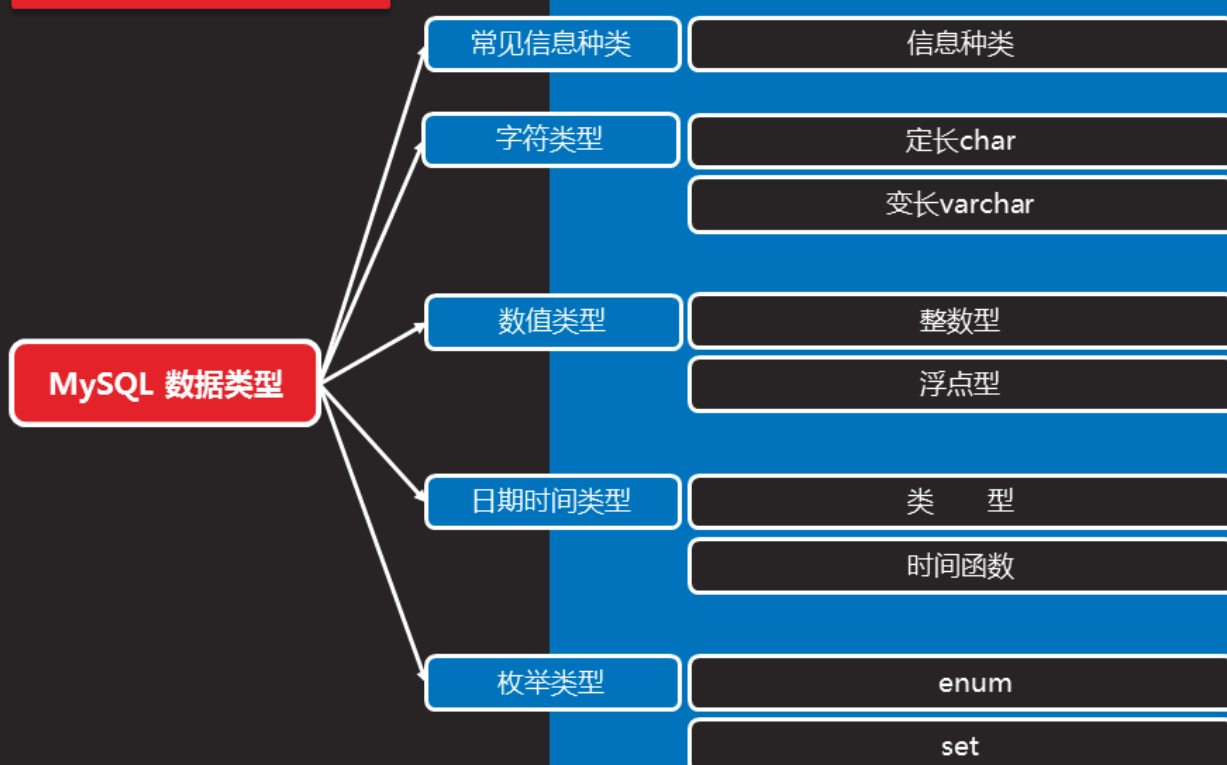
课堂练习

1. 使用mysql命令连接数据库
2. 练习库管理命令（查看、删除、创建库、切换）
3. 练习表管理命令（查看、删除、创建表）
4. 练习记录管理命令（插入、查看、修改、删除）

| 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



# 常见信息种类

## 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、专业
- 日期时间型：出生日期、注册时间

知识讲解

\* 邮件地址  @ 163.com ▼

6~18个字符，可使用字母、数字、下划线，需以字母开头

\* 密码

6~16个字符，区分大小写

\* 确认密码

请再次填写密码

请输入密码

\* 确认密码



# 字符类型

## 定长char

知识讲解

- 定长：char(字符个数)
  - 最大字符个数255
  - 不够指定字符个数时在右边用空格补全
  - 字符个数超出时，无法写入数据。

```
mysql> create table db1.t1(  
name char(5),  
homedir char(50)  
);
```

```
mysql> insert into db1.t1 values( "bob" ," USA" );
```



## 变长varchar

### 知识讲解

- 变长：varchar(字符个数)
  - 按数据实际大小分配存储空间
  - 字符个数超出时，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 db1.t2(  
name char(5),  
email varchar(30)  
);  
mysql> insert into db1.t2 values( "lucy" ," lucy@tedu.cn" );
```



## 案例3：字符类型

- 按照如下结构建表：

### 课堂练习

```
mysql> desc db1.t3;
```

| Field    | Type         | Null | Key | Default | Extra |
|----------|--------------|------|-----|---------|-------|
| name     | char (5)     | YES  |     | NULL    |       |
| mail     | varchar (10) | YES  |     | NULL    |       |
| homeaddr | varchar (50) | YES  |     | NULL    |       |



# 数值类型

## 整数型

- 只能存储整数

| 类 型       | 名称        | 有符号范围                   | 无符号范围             |
|-----------|-----------|-------------------------|-------------------|
| tinyint   | 微小整数      | -128~127                | 0 ~ 255           |
| smallint  | 小整数       | -32768~32767            | 0 ~ 65535         |
| mediumint | 中整型       | $-2^{23} \sim 2^{23}-1$ | $0 \sim 2^{24}-1$ |
| int       | 大整型       | $-2^{31} \sim 2^{31}-1$ | $0 \sim 2^{32}-1$ |
| bigint    | 极大整型      | $-2^{63} \sim 2^{63}-1$ | $0 \sim 2^{64}-1$ |
| unsigned  | 使用无符号存储范围 |                         |                   |

## 浮点型

- 浮点型

- 格式1：字段名 类型;
- 格式2：字段名 类型(总宽度, 小数位数)

知识讲解

| 类 型    | 名称  | 有符号范围                  | 无符号范围 |
|--------|-----|------------------------|-------|
| float  | 单精度 | 0 ~ 2 <sup>32</sup> -1 |       |
| double | 双精度 | 0 ~ 2 <sup>64</sup> -1 |       |

```
mysql> create table db1.t3(pay float , id double);
mysql> create table db1.t4( pay float(5,2) , id double(4,2) );
mysql> insert into db1.t4 values(9999,9999);
ERROR 1264 (22003): Out of range value for column 'pay' at row 1
```



## 案例4：数值类型

- 按照如下结构建表：

课堂练习

```
mysql> desc db1.t2;
```

| Field   | Type        | Null | Key | Default | Extra |
|---------|-------------|------|-----|---------|-------|
| stu_num | int(11)     | YES  |     | NULL    |       |
| name    | char(5)     | YES  |     | NULL    |       |
| age     | tinyint(4)  | YES  |     | NULL    |       |
| pay     | float       | YES  |     | NULL    |       |
| money   | float(5, 2) | YES  |     | NULL    |       |





# 日期时间类型

## 类型

知识讲解

- 日期时间 datetime
  - 范围：1000-01-01 00:00:00 ~ 9999-12-31 23:59:59
  - 格式：yyyymmddhhmmss
- 日期时间 timestamp
  - 范围：1970-01-01 00:00:00 ~ 2038-01-19 00:00:00
  - 格式：yyyymmddhhmmss



## 类型（续1）

知识讲解

- 日期 date
  - 范围：0001-01-01 ~ 9999-12-31
  - 格式：yyyymmdd
- 年 year
  - 范围：1901~2155
  - 格式：yyyy
- 时间 time
  - 格式：HH:MM:SS



## 类型（续2）

知识讲解

- 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 值为 NULL (空)
- 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



## 时间函数

知识讲解

- MySQL服务内置命令
  - 可以使用时间函数给字段赋值

| 类 型       | 用 途         |
|-----------|-------------|
| curtime() | 获取当前的系统时间   |
| curdate() | 获取当前的系统日期   |
| now()     | 获取当前系统日期和时间 |
| year()    | 获取年         |
| month()   | 获取月         |
| day()     | 获取日         |
| date()    | 获取日期        |
| time()    | 获取时间        |



## 案例5：日期时间类型

课堂练习

- 练习如下时间函数的使用：
  - now() year() month() day() date() time()
  - curtime() curdate()
- 按照如下结构建表：

```
mysql> desc db1.t4;
```

| Field      | Type      | Null | Key | Default | Extra |
|------------|-----------|------|-----|---------|-------|
| name       | char (10) | YES  |     | NULL    |       |
| your_start | year (4)  | YES  |     | NULL    |       |
| up_time    | time      | YES  |     | NULL    |       |
| birthday   | date      | YES  |     | NULL    |       |
| party      | datetime  | YES  |     | NULL    |       |



# 枚举类型

## enum

- enum 单选：
  - 格式：字段名 enum(值1, 值2, 值N)
  - 仅能选择一个值
  - 字段值必须在列表里选择

```
mysql> create table db1.t5 (  
        name char(5),  
        sex enum( "boy" , " girl" , "no" )  
);
```

```
mysql> insert into db1.t5 values( "bob" , "man" );
```

## set

### 知识讲解

- set 多选：
  - 格式：字段名 set(值1, 值2, 值N)
  - 选择一个或多个值
  - 字段值必须在列表里选择

```
mysql> create table db1.t6(  
name char(5),  
likes set( "eat" , "game" , "music" , "money" )  
);
```

```
mysql> insert into db1.t6 values( "jerry" , "eat,film" );
```



## 案例6：枚举类型

- 按照如下结构建表：

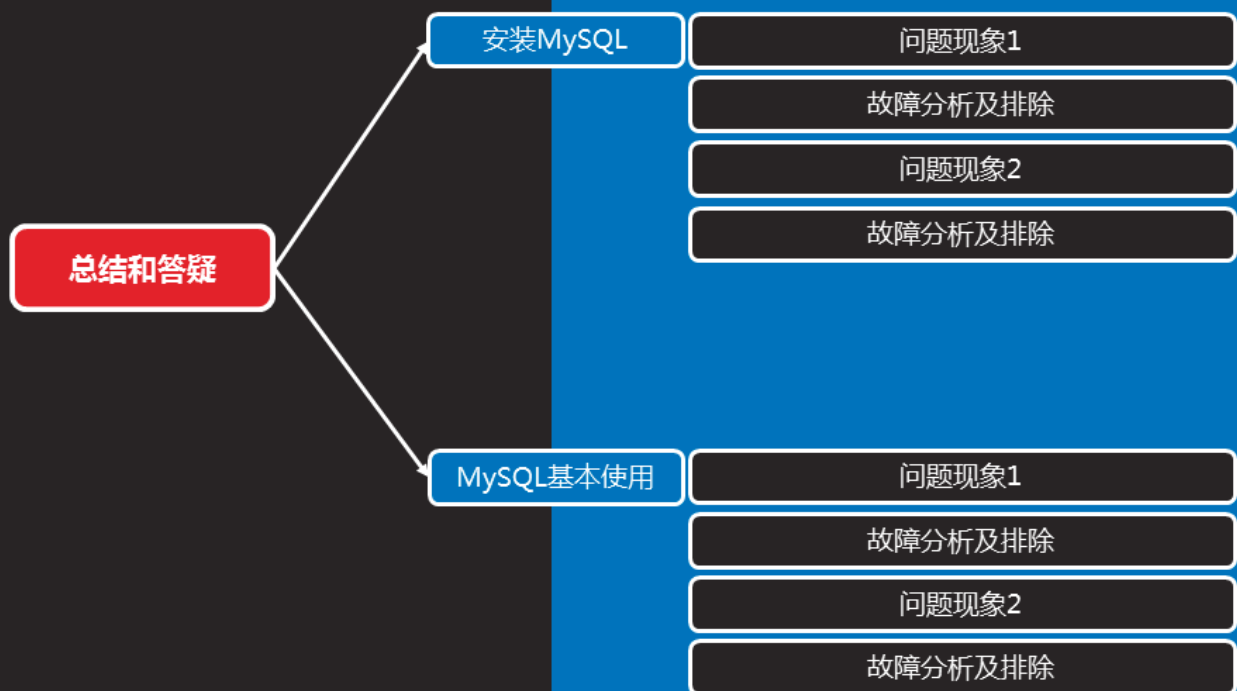
### 课堂练习

```
mysql>  
mysql> desc db1.t5;
```

| Field | Type                                | Null | Key | Default | Extra |
|-------|-------------------------------------|------|-----|---------|-------|
| name  | char(5)                             | YES  |     | NULL    |       |
| likes | set('eat', 'game', 'film', 'music') | YES  |     | NULL    |       |
| sex   | enum('boy', 'girl', 'no')           | YES  |     | NULL    |       |



## 总结和答疑



# 安装MySQL

## 问题现象1

知识讲解

- 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

知识讲解

- 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

知识讲解

- 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

知识讲解

- 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 .....

知识讲解

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



## 故障分析及排除

知识讲解

- 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

知识讲解

- 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

知识讲解

- 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)
```

