

一、数据库的发展历史

- 1、人工处理
- 2、文件系统
- 3、数据库管理

二、数据库的相关概念

1、数据库系统（DBS）

以数据库为核心的计算机系统，主要包含：

- 硬件
- 系统软件
- 数据库管理系统（DBMS）
- 数据库管理员（DBA）
- 数据

2、数据库（DB）

存放数据的仓库。

3、数据（Data）

符号记录，一般表现为文字、字符、图片、声音、视频...

4、数据库管理系统(DBMS)

用来定义数据、管理和维护数据的软件。

5、数据模型

对现实世界的模拟。

- 1) 概念模型：对现实世界的信息进行概括，抽象 ==> E-R模型(图)
- 2) 逻辑模型：系统分析设计人员对数据存储的观点，是对概念数据模型进一步的分解和细化。
 - 网状模型
 - 层次模型
 - 关系模型：以表格的方式来管理数据（关系表）
 - 面向对象模型
- 3) 物理模型

物理模型是对真实数据库的描述。

数据库中的一些对象如下：表，视图，字段，数据类型、长度、主键、外键、索引、是否可为空，默认值。

6、关系数据库

逻辑模型为关系模型的数据库管理系统，常用的关系数据库有：Oracle、DB2、Sybase、SQL Server、PostgreSQL、MySQL、Sqlite、H2、Access等。

7、SQL

Structured Query Language（简称SQL），结构化查询语言，数据库管理系统可以通过SQL语言来管理数据库中的数据。通俗地讲，它是用于与数据库通信的语言。主要有以下四种分类：

- DDL（Data Definition Language）：数据定义语言，主要用于定义数据库、表、视图、索引、触发器等。有 drop、create、alter等语句。
- DML（Data Manipulation Language）：数据操作语言，主要包括对数据的增、删、改。
- DQL（Data Query Language）：数据检索语言
- DCL（Data Control Language）：数据控制语言

三、数据库的设计过程

第一：收集信息

第二：标识对象（实体、实体对象）

第三：标识属性

第四：标识对象间的关系

一对一	1:1
1对多	1:M 、 1:N 、 1:*
多对多	M:N 、 **

第五：绘制E-R图

- 实体：矩形
- 属性：椭圆形
- 关系：菱形
- 连接：直线（箭头）

第六：把E-R图转换为数据表（关系表）

- 1.把实体名转换为表名
 - 2.把属性转换为表的字段
 - 3.标识主键
 - 4.标识表间的关系（外键）
- 第七：使用范式规范化数据表

案例

图书借阅管理系统的基本功能

用户注册和登录后，数据库中存储用户的注册信息

数据库中存储所有图书的详细信息

用户借书后，数据库中存储用户的借阅信息，如：借书时间、图书信息，用户信息等

用户还书后，数据库中存储归还的时间等信息

要求：根据以上需求描述，绘制E-R图，并使用T-SQL定义相关的表，每张表至少要录入三条记录。

四、范式

1、概念

规范化数据（表）

2、目的

- 为了"消除"数据的冗余

- "消除"CRUD的异常

3、分类

- 第一范式：保证每列不可再分（原子性）
- 第二范式：确保表中的每列都和主键相关（确保一张表只描述一件事情）
- 第三范式：确保每列都和主键列直接相关,而不是间接相关

注：有时为了性能和效率，可以放弃范式的约束。

五、MySQL特点

1. 免费、开源
2. 跨平台、支持多种编程语言，如Java、C/C++、Python、PHP等
3. 功能强大、使用方便
4. 标准的SQL
5. 版本
 1. 社区版（免费）
 2. 企业版（收费）

六、安装与配置

1、安装

1) Window系统安装

- 安装包：根据向导安装、配置
- 压缩包：解压到指定目录、配置

以下是以压缩包安装为例

第一：下载、并解压到指定位置

- 安装包：<https://dev.mysql.com/downloads/installer/>
- 压缩包：<https://dev.mysql.com/downloads/mysql/5.7.html#downloads>

解压位置

C:\dev\mysql-5.7.32-winx64

第二：配置环境变量

```
C:\dev\mysql-5.7.32-winx64\bin
```

第三：创建MySQL配置文件

在MySQL安装的根目录，创建 `my.ini` 配置文件，内容如下：

```
[mysqld]
# 设置3306端口
port=3306
# 设置mysql的安装目录
basedir=C:\dev\mysql-5.7.32-winx64
# 设置mysql数据库的数据的存放目录
datadir=C:\dev\mysql-5.7.32-winx64\data
# 允许最大连接数
max_connections=200
# 允许连接失败的次数。这是为了防止有人从该主机试图攻击数据库系统
max_connect_errors=10
# 服务端使用的字符集默认为UTF8
character-set-server=utf8
# 创建新表时将使用的默认存储引擎
default-storage-engine=INNODB
# 默认使用“mysql_native_password”插件认证
default_authentication_plugin=mysql_native_password

[mysql]
# 设置mysql客户端默认字符集
default-character-set=utf8

[client]
# 设置mysql客户端连接服务端时默认使用的端口
port=3306
default-character-set=utf8
```

注意：其中的data目录是用于存放数据的目录，用户可以不需要创建，下一步初始化工作中会自动创建！

第四：初始化数据库

以管理员身份运行CMD并进入MySQL安装目录下的bin目录，执行命令：`mysql --initialize [-user=root] --console。`

```
C:\dev\mysql-5.7.32-winx64\bin> mysql --initialize --user=mysql --console
...
... (root@localhost: aB4rvf335G,Z) # 初始密码
...
```

注意：

1) `--initialize`：初始化数据库，创建一个root帐号并指定了一个默认密码。

其中root@localhost后面的 `aB4rvf335G,Z` 就是初始密码。在没有更改密码前，需要记住这个密码，便于后续登录。

当然，如果忘记，则删除data目录，重新执行以上命令即可。

```
$ mysql -u root -p
# 输入初始密码: aB4rvf335G,Z
```

2) `--initialize-insecure`：功能同上，但不会生成默认密码。

```
# 跳过密码登录
$ mysql -u root -p --skip-password
```

3) `--user=mysql`：在 Unix 和类 Unix 系统上，数据库目录和文件由 `mysql` 登录帐户拥有是很重要的，这样服务器在您以后运行时可以对它们进行读写访问。如果不使用-user参数，则默认创建root帐号。

4) `--defaults-file=d:\my.ini`：指定默认配置文件

```
bin\mysqld --defaults-file=C:\my.ini --initialize --console
```

第五：安装服务

同样的，进入MySQL安装目录下的bin目录，执行命令：`mysqld --install [服务名称]`，如果不指定服务名称，则默认为 `mysql`

```
C:\dev\mysql-5.7.32-winx64\bin> mysqld --install MySQL5.7
```

第六：启动服务

```
# 启动服务
net start mysql5.7

# 停止服务
net stop mysql5.7

# 删除服务
sc delete 服务名称
mysqld -remove 服务名称
```

第七：登录MySQL

```
mysql -u帐号 -p[密码]
mysql -uroot -paB4rvf335G,Z
mysql -uroot -p
```

第八：修改初始密码

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY '新密码';
mysql> set password = password('密码')
```

第九：退出重新登录

```
exit
mysql -uroot -p
# 输入新密码
```

第十、安装图形界面客户端

2) Linux系统安装

```
# 查看进程
[zing@localhost ~]$ ps -ef | grep mysql
root      1371      1  0  2018 ?        00:00:00 /bin/sh /usr/bin/mysqld_safe --
datadir=/var/lib/mysql --socket=/var/lib/mysql/mysql.sock --pid-file=/var/run/mysqld/mysqld.pid
--basedir=/usr --user=mysql
mysql     1479  1371  0  2018 ?        00:20:46 /usr/libexec/mysqld --basedir=/usr --
datadir=/var/lib/mysql --user=mysql --log-error=/var/log/mysqld.log --pid-
file=/var/run/mysqld/mysqld.pid --socket=/var/lib/mysql/mysql.sock
zing      17405 17387  0  09:35 pts/0    00:00:00 grep mysql
```

3) Mac系统安装

2、目录结构

- bin目录：存储可执行文件
- data目录：存储数据文件
- include目录：存储包含的头文件
- lib目录：存储库文件
- docs目录：文档

- share目录：错误消息和字符集文件

3、登录MySQL

```
mysql -u 帐号 -p
```

其它参数：

-u: 用户名

-p: 密码

-h: 服务器名，默认是本地服务器

-P: 端口号，默认是3306

-D: 打开指定数据库

--prompt=name: 指定命令提示符

--delimiter=name: 指定分隔符

-V: 输出版本信息并退出

--version: 同-V

注意：如果远程连接失败，则进行以下授权

```
[zing:~]$ mysql -uroot -h192.168.0.1 -P3306 -p
Enter password:
ERROR 1045 (28000): Access denied for user 'root'@'192.168.0.1' (using password: YES)

####解决####
# 进入mysql数据库
use mysql

# 授权:授予此帐号允许所有IP访问全部数据库及相关的数据库表
# GRANT ALL PRIVILEGES ON *.* TO '帐号'@'%' IDENTIFIED BY '密码' WITH GRANT OPTION;
# 其中, *.*表示全部数据库的全部表, %表示所有的ip地址
GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' IDENTIFIED BY 'root' WITH GRANT OPTION;

# 从 MySQL 8 开始, 不能支持一段式授权用户, 需要先创建再授权。
mysql> CREATE USER 'root'@'%' IDENTIFIED BY '密码';
mysql> GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' WITH GRANT OPTION;

# 刷新生效
mysql> flush privileges;

# 查看配置是否成功
select host,user from user;
+-----+-----+
```



```
| host                | user  |
+-----+-----+
| %                  | root  |
| .....             | ....  |
+-----+-----+
```

4、退出MySQL

```
exit
quit
\q
```

5、操作技巧

- 1) 命令行结束符以;或\g结束
- 2) \c取消当前命令执行
- 3) 可以通过help、?、\h加上相关关键字查看手册

```
#语法: ? 关键字 | help 关键字 | \h 关键字
? create database;
? int
\h use
```

4) 注释

- 单行注释

```
#单行注释

-- 单行注释 (空格)

/*
多行注释
*/
```

5) 查看MySQL版本

```
# 查看MySQL版本 (登录MySQL)
select version();

# 查看MySQL版本 (退出MySQL)
mysql --version
mysql -V
```

6、配置文件

1、查看MySQL配置

注意：必须在MySQL命令下执行

```
mysql> \s  
mysql> status # 功能同上
```

2、创建配置文件

在MySQL安装目录下，创建my.ini文件，格式如下所示：

```
#客户端配置  
[mysql]  
...  
  
#服务端配置  
[mysqld]  
...
```

3、乱码问题

在MySQL安装目录下，创建my.ini文件，并配置如下：

```
#客户端字符编码配置  
[mysql]  
default-character-set=utf8  
...  
  
#服务端字符编码配置  
[mysqld]  
character-set-server=utf8  
...
```

查看字符编码

```
mysql> show variables like 'character%';
```

Variable_name	Value
character_set_client	latin1
character_set_connection	latin1
character_set_database	utf8

```
| character_set_filesystem | binary |
| character_set_results   | latin1 |
| character_set_server    | utf8   |
| character_set_system    | utf8   |
| character_sets_dir      | /usr/share/mysqlCharsets/ |
+-----+-----+
8 rows in set (0.00 sec)
```

```
#character_set_database 默认是latin1
```

7、卸载

1) 停止服务

```
# 停止服务
$ net stop MySQL5.7          # 其中, MySQL5.7为安装时的服务名称
$ mysqld -remove            # 删除后台进程, 否则在删除文件时可能会提示文件被占用
```

2) 删除文件

- 在控制面板中, 卸载以mysql开头的软件
- 在硬盘中, 删除MySQL的安装目录
- 删除注册表 (win+R, 输入regedit)
 - HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\Eventlog\Application\MySQL 目录删除
 - HKEY_LOCAL_MACHINE\SYSTEM\ControlSet002\Services\Eventlog\Application\MySQL 目录删除
 - HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\MySQL 目录删除

3) 删除mysql服务

```
# 其中, MySQL5.7为安装时的服务名称
$ sc delete MySQL5.7
```

4) 删除注册表 win+R 然后输入regedit回车进入注册表编辑

- HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\Eventlog\Application\MySQL 目录删除
- HKEY_LOCAL_MACHINE\SYSTEM\ControlSet002\Services\Eventlog\Application\MySQL 目录删除
- HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\MySQL 目录删除

七、图形界面

1、Navicat

目前开发者用的最多的一款mysql图形用户管理工具，界面简洁、功能也非常强大，与微软的SQLServer管理器很像，简单易学，支持中文，提供免费版本。

2、Workbench

Mysql官方的一个图形管理工具，支持数据库的创建、设计、迁移、备份、导出、导入等功能，支持windows、linux、mac主流的操作系统，使用起来非常的棒。

3、SQLyog

Webyog公司的产品，一款易于使用、快速而简洁的mysql数据库图形化管理工具，可以直观的在任何地点高效地管理你的数据库。

八、数据库管理

1、创建数据库

```
#语法: create database | schema [if not exists] 数据库名称 [[default] character set [=] 字符集名称];  
create database if not exists pms default character set = 'UTF8';
```

2、查看当前服务器下的数据库列表

```
#语法: show databases | schemas ;  
show databases;
```

3、查看指定数据库的定义

```
#语法: show create database | schema 数据库名称;  
show create database test_db;
```

4、修改指定数据库的编码方式

```
#语法: alter database | schema 数据库名称 [default] character set [=] 字符集名称;  
alter database test_db default character set = 'utf8';
```

5、打开指定数据库

```
#语法: use 数据库名;  
use test_db;  
  
#查看当前打开的数据库, 如果没打开则显示null  
#语法: select database() | schema() ;  
select database();
```

6、删除指定数据库

```
#语法: drop database | schema [if exists] 数据库名称;  
drop database if exists test_db;
```

7、查看上一步操作产生的警告信息

```
show warnings ;
```

