

1 数据库安装

1.1 基础知识

数据库：数据库就是一个用来存储各种数据的容器

数据库管理系统：专门用于创建和管理数据库的统一软件，介于应用和操作系统之间。比如MySQL、SQL Server、Oracle等等，它们不仅有基础的数据管理功能，还能保证数据的完整性、安全性和可靠性。

数据库应用程序：通过数据库管理系统提供的接口与之通信，访问和管理数据的应用程序。

SQL语言：它是一种数据库查询语言和程序设计语言，主要用于管理数据库中的数据，如存储数据、查询数据、更新数据等等。

1.2 下载与安装

本章节我们所要学的就是MySQL，你可以通过以下链接去下载安装：

① MySQL下载链接
<https://dev.mysql.com/downloads/mysql/>
历史版本下载链接（选择MySQL Community Server）
<https://downloads.mysql.com/archives/>

当然，你也可以安装phpStudy等集成环境。

1.3 MySQL 数据库的层次

MySQL数据库的层次是这样的：数据库->表->列->数据，可以有多个数据库，每个数据库中也有多个表，每个表中也有多个列，所有列查询出一行数据（数据是有多行的）。

2 数据库和表操作

2.1 MySQL 数据库的操作

1. 查看全部数据库

```
1 show databases
```

2. 创建数据库

```
1 create database 数据库名
```

3. 查询创建好的数据库

```
1 show create database 数据库名
```

4. 修改数据库

```
1 alter database 数据库名称 default character set 编码方式 collate 编码方式_bin
2 alter database yang default character set gbk collate gbk_bin
```

5. 删除数据库

```
1 drop database 数据库名称
```

2.2 MySQL 数据类型

INT类型：

数据类型	字节数
TINYINT	1
SMALLINT	2
MEDIUMINT	3
INT	4

数据类型	字节数
BIGINT	8

FLOAT类型：

数据类型	字节数
FLOAT	4
DOUBLE	8
DECIMAL(M,D)	M+2

时间日期类型：

数据类型	字节数	日期格式	取值范围
YEAR	1	YYYY	1901~2155
DATE	4	YYYY-MM-DD	1000-01-01~9999-12-03
TIME	3	HH:MM:SS	-838:59:59~838:59:59
DATETIME	8	YYYY-MM-DD HH:MM:SS	1000-01-01 00:00:00~ 9999-12-31 23:59:59
TIMESTAMP	4	YYYY-MM-DD HH:MM:SS	1970-01-01 00:00:01~ 2038-01-19 03:14:07

字符串和文本类型：

数据类型	字节数
CHAR	用于表示固定长度的字符串

数据类型	字节数
VARCHAR	用于表示可变长度的字符串
BINARY	用于表示固定长度的二进制数据
VARBINARY	用于表示可变长度的二进制数据
TINYTEXT	0-255字节
TEXT	0~65535字节
MEDIUMTEXT	0~16777215字节
LONGTEXT	0~4294967295字节

2.3 MySQL 数据表的操作

1.创建数据表

```
1 create table 表名 (字段1 数据类型, 字段2 数据类型...)
```

2.查询当前数据库的所有表

```
1 show tables 表名
```

3.查看数据表

```
1 show create table 表名  
2 desc 表名
```

4.修改表

1	alter table 表名 default character set 编码方式 collate 编码方式_bin(修改字符集)
2	alter table 表名 change 旧字段名 新字段名 新类型 (修改字段名)
3	alter table 旧表名 rename 新表名 (修改表名)
4	alter table 表名 modify 字段名 数据类型 (修改字段属性)
5	alter table 表名 add 新字段名 数据类型 (添加新字段)
6	alter table 表名 drop 字段名 (删除字段)
7	alter table 表名 modify 字段名1 数据类型 first/after 字段名2

5.删除表

1	drop table 表名
---	---------------

3 表数据的增加更新和删除

1.查看表的数据

```
1 select * from 表名
2 select 列名 from 表名
```

2.为表中所有字段添加数据

```
1 insert into 表名(字段1,字段2 ...) values(值1,值2 ...)
2
3 // 不指定字段添加数据
4
5 insert into 表名 Values(值1,值2,值3 ...)
```

3.更新数据

```
1 update 表名 Set 字段名=值 where 条件表达式
```

4.删除数据

```
1 delete from 表名 where 条件表达式
```

4 VC6连接MySQL

了解完基础的语法和知识点之后，接下来我们需要使用VC6通过SDK去连接MySQL，在之前你已经安装过了MySQL Server应用了，接下来只需要将include和lib目录填入VC6的配置中即可，在正式使用之前还需要包含几个头文件：

```
1 #include <WINSOCK2.H> // 因为在这里SDK的实现，就是基于TCP/IP协议，所以我们需要包含这个头
2 #include <mysql.h>
3 #pragma comment(lib, "libmysql.lib")
```

4.1 常用函数

我们来了解一下这个SDK常用的函数：

1.分配或初始化与mysql_real_connect()相适应的MYSQL对象：

```
1 MYSQL * STDCALL mysql_init(
2     MYSQL *mysql // 可以理解就是MySQL的一个句柄
3 );
```

如果传入的参数mysql是NULL指针，该函数将分配、初始化、并返回新对象，否则，将初始化对象，并返回对象的地址；如果mysql_init()分配了新的对象，当调用mysql_close()来关闭连接时，将释放该对象。

2.与MySQL数据库引擎建立连接：

```
1 MYSQL *mysql_real_connect (
2     MYSQL *mysql, // MySQL句柄
3     const char *host, // 目标host
4     const char *user, // 用户名
5     const char *passwd, // 密码
6     const char *db, // 数据库名, 当db为null的时候, 函数连接到默认数据库
7     unsigned int port, // 目标端口
8     const char *unix_socket, // unix_socket为null时, 表明不使用socket或管道机制
9     unsigned long client_flag // 设置为0
10 );
```

3.选择一个MySQL数据库：

```
1 int STDCALL mysql_select_db(
2     MYSQL *mysql, // MySQL句柄
3     const char *db // 数据库名字
4 );
```

4.SQL语句查询：

```
1  int STDCALL mysql_query(  
2      MYSQL *mysql, // MySQL句柄  
3      const char *q // SQL语句  
4  );
```

5. SQL语句查询返回结果集：

```
1  MYSQL_RES * STDCALL mysql_store_result(  
2      MYSQL *mysql // MySQL句柄  
3  );
```

6. 关闭连接：

```
1  void STDCALL mysql_close(  
2      MYSQL *sock // MySQL句柄  
3  );
```

7. 释放结果集使用的内存：

```
1  void STDCALL mysql_free_result(  
2      MYSQL_RES *result // 结果集  
3  );
```