

数据库

数据库

常见数据库

关系型

文件型

两端

数据基本概念

数据库操作

命令行

数据类型

可视化工具

常用命令

查

增

删

改

排序

个数

平均分

总数

LIMIT 分页

导入导出数据库

enter `code` here

常见数据库

关系型

- MySQL

- 免费、

enter code here

中小网站

优点：性能非常不错

缺点：集群、容灾稍微弱一些

- Oracle
 - 挺贵、大型应用、金融级
 - 优点：性能非常不错、集群、容灾非常强
 - 缺点：挺贵
- SQLServer
- Access

文件型

- sqlite
 - 小巧
- mongodb
 - 大数据
 - 不容易描述两表关系

两端

服务端：数据在的地方

客户端：获取数据的第一关口（管理工具，node）

数据基本概念

两种单位：

- 1.库：文件夹-用来管理，本身没法存数据
- 2.表：文件-存数据的

表-Excel：

行-一条数据

列(字段、域)-一个数据项

数据库操作

命令行

直接用mysql需要配置环境变量

1. 登录mysql

```
mysql -h host -u user -p
Enter password: ****
```

-h host 是远程服务器的ip地址，**-u user** 是登陆数据库的用户名，需要注意的是 **-p** 后面并没有跟随密码（明文密码不安全），敲下这行命令之后，会提示输入密码。

如果数据就是安装在本机，那么不需要 **-h host**，比如：

```
mysql -u user -p
```

2. 查看当前数据库版本和当前时间

注意：在成功登陆数据库之后，所有的操作都必须以 “;” 号结尾的，否则不能执行，mysql会把回车理解成换行。（一定要注意“;”号结尾）

```
mysql> SELECT VERSION(), CURRENT_DATE;
```

3. 查看当前所有库

```
mysql> show databases;
```

4. 创建一个数据库

```
mysql> CREATE DATABASE newKU;
```

5. 使用某个数据库 (没有;号结尾)

```
mysql> USE newKU
```

6. 查看当前数据库下的所有的表

```
mysql> SHOW TABLES;
```

7. 创建一张表

```
mysql> CREATE TABLE pet (name VARCHAR(20), owner VARCHAR(20),  
-> species VARCHAR(20), sex CHAR(1), birth DATE, death  
DATE);
```

8. 查看一张表的结构

```
mysql> DESCRIBE pet;
```

9. 退出数据库

```
mysql> QUIT
```

数据类型

MySQL 数据类型

MySQL中定义数据字段的类型对你数据库的优化是非常重要的。
MySQL支持多种类型，大致可以分为三类：数值、日期/时间和字符串(字符)类型。

数值类型

MySQL支持所有标准SQL数值数据类型。
这些类型包括严格数值数据类型(INTEGER、SMALLINT、DECIMAL和NUMERIC)，以及近似数值数据类型(FLOAT、REAL和DOUBLE PRECISION)。
关键字INT是INTEGER的同义词，关键字DEC是DECIMAL的同义词。
BIT数据类型保存位字段值，并且支持MyISAM、MEMORY、InnoDB和BDB表。
作为SQL标准的扩展，MySQL也支持整数类型TINYINT、MEDIUMINT和BIGINT。下面的表显示了需要的每个整数类型的存储和范围。

类型	大小	范围 (有符号)	范围 (无符号)	用途
TINYINT	1 字节	(-128 , 127)	(0 , 255)	小整数值
SMALLINT	2 字节	(-32 768 , 32 767)	(0 , 65 535)	大整数值
MEDIUMINT	3 字节	(-8 388 608 , 8 388 607)	(0 , 16 777 215)	大整数值
INT或 INTEGER	4 字节	(-2 147 483 648 , 2 147 483 647)	(0 , 4 294 967 295)	大整数值
BIGINT	8 字节	(-9 223 372 036 854 775 808 , 9 223 372 036 854 775 807)	(0 , 18 446 744 073 709 551 615)	极大整数值
FLOAT	4 字节	(-3.402 823 466 E+38 , -1.175 494 351 E-38) , 0 , (1.175 494 351 E-38 , 3.402 823 466 351 E+38)	0 , (1.175 494 351 E-38 , 3.402 823 466 E+38)	单精度 浮点数值
DOUBLE	8 字节	(-1.797 693 134 862 315 7 E+308 , -2.225 073 858 507 201 4 E-308) , 0 , (2.225 073 858 507 201 4 E-308 , 1.797 693 134 862 315 7 E+308)	0 , (2.225 073 858 507 201 4 E-308 , 1.797 693 134 862 315 7 E+308)	双精度 浮点数值
DECIMAL	对 DECIMAL(M,D) ，如果M>D，为 M+2否则为D+2	依赖于M和D的值	依赖于M和D的值	小数值

日期和时间类型

表示时间值的日期和时间类型为DATETIME、DATE、TIMESTAMP、TIME和YEAR。
每个时间类型有一个有效值范围和一个"零"值，当指定不合法的MySQL不能表示的值时使用"零"值。
TIMESTAMP类型有专有的自动更新特性，将在后面描述。

类型	大小 (字节)	范围	格式	用途
DATE	3	1000-01-01/9999-12-31	YYYY-MM-DD	日期值
TIME	3	'-838:59:59'/'838:59:59'	HH:MM:SS	时间值或持续时间
YEAR	1	1901/2155	YYYY	年份值

DATETIME	8	1000-01-01 00:00:00/9999-12-31 23:59:59	YYYY-MM-DD HH:MM:SS	混合日期和时间值
TIMESTAMP	4	1970-01-01 00:00:00/2038 结束时间是第 2147483647 秒，北京时间 2038-1-19 11:14:07 ，格林尼治时间 2038年1月19日凌晨 03:14:07	YYYYMMDD HHMMSS	混合日期和时间值，时间戳

字符串类型

字符串类型指CHAR、VARCHAR、BINARY、VARBINARY、BLOB、TEXT、ENUM和SET。该节描述了这些类型如何工作以及如何在查询中使用这些类型。

类型	大小	用途
CHAR	0-255字节	定长字符串
VARCHAR	0-65535 字节	变长字符串
TINYBLOB	0-255字节	不超过 255 个字符的二进制字符串
TINYTEXT	0-255字节	短文本字符串
BLOB	0-65 535字节	二进制形式的长文本数据
TEXT	0-65 535字节	长文本数据
MEDIUMBLOB	0-16 777 215字节	二进制形式的中等长度文本数据
MEDIUMTEXT	0-16 777 215字节	中等长度文本数据
LOBLOB	0-4 294 967 295字节	二进制形式的极大文本数据
LONGTEXT	0-4 294 967 295字节	极大文本数据

CHAR和VARCHAR类型类似，但它们保存和检索的方式不同。它们的最大长度和是否尾部空格被保留等方面也不同。在存储或检索过程中不进行大小写转换。

BINARY和VARBINARY类类似于CHAR和VARCHAR，不同的是它们包含二进制字符串而不要非二进制字符串。也就是说，它们包含字节字符串而不是字符串。这说明它们没有字符集，并且排序和比较基于列值字节的数值值。

BLOB是一个二进制大对象，可以容纳可变数量的数据。有4种BLOB类型：TINYBLOB、BLOB、MEDIUMBLOB和LOBLOB。它们只是可容纳值的最大长度不同。

有4种TEXT类型：TINYTEXT、TEXT、MEDIUMTEXT和LONGTEXT。这些对应4种BLOB类型，有相同的最大长度和存储需求。

可视化工具

Navicat Premium

常用命令

查

```
SELECT column_name,column_name
FROM table_name
[WHERE Clause]
[OFFSET M ][LIMIT N]
```

```
SELECT * FROM `user` WHERE ID=1
```

增

```
INSERT INTO table_name ( field1, field2,...fieldN ) VALUES
( value1, value2,...valueN );
```

删

```
DELETE FROM table_name [WHERE Clause]
```

改

```
UPDATE table_name SET field1=new-value1, field2=new-value2
[WHERE Clause]
```

排序

升序

```
SELECT * FROM `user` ORDER BY age ASC
```

降序

```
SELECT * FROM `user` ORDER BY age DESC
```

价格(price)升序排序，如果价格相同，再按销量(sales)降序排序

```
ORDER BY price ASC, sales DESC
```

个数

查询user所有数据的个数

```
SELECT COUNT(*) FROM `user`
```

相同class只留一条数据

```
SELECT * FROM student_table GROUP BY class;
```

找到每个班级的总人数

```
SELECT class, COUNT(class) FROM student_table GROUP BY class;
```

平均分

以班级分组 查询每组的平均分

```
SELECT class, AVG(score), MAX(score) FROM student GROUP BY class
```

总数

班级总分数

```
SELECT class, SUM(score) FROM student GROUP BY class
```

班级总分数排序

```
SELECT class, SUM(score) FROM student GROUP BY class ORDER BY SUM(score) ASC
```


LIMIT 分页

LIMIT-限制输出

分页：

- 1.所有数据给前端
- 2.后台只给一丁点数据

LIMIT 10; 前10条

LIMIT 5,8; 从5开始，要8个

分页：

每页20条

第1页 0,20 0~19

第2页 20,20 20~39

第3页 40,20

第n页 $(n-1)*20,20$

子句之间是有顺序

WHERE GROUP ORDER LIMIT

筛选 合并 排序 限制

```
SELECT class,COUNT(class) FROM student_table
```

```
WHERE score>60
```

```
GROUP BY class
```

```
ORDER BY COUNT(class) DESC
```

```
LIMIT 2;
```

导入导出数据库

.sql