### 数据库基础知识

> mysql 使用

```

// mysql 使用：

1.下载mysql模块(client)

2.连接

var db=mysql.createConnection(host, port, user, password, database) //?

3.查询

db.query(SQL, (err, data)=>{})

4.SQL语句

增删改查

// 连接池 ---为了保持用户在访问服务器时，数据库实时链接

// 我们需要预设链接指定的一些库，这样有利于提高相应效率

const db=mysql.createPool({host: 'localhost', user: 'root', password: '123456', database: 'blog'});

```

> 数据库类型

```

1).关系型数据库：

MySQL、Oracle、SQLServer、Access、db2、fox pro

MySQL：

免费、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

2).文件型：

sqlite、mongodb(偏向缓存)

3).空间型：------------------用的比较少

```

> 数据库有两个端

```

1）.mySQL Server端：

数据存在（mySQL服务器）

安装：

网址: www.mySql.com

下载：mysql server

// 百度应用中心: mysql server

skip to update > server only

配置：

看视频

development server

默认账户： root 密码自己设定

Add user

--------------------------------------

mySQL默认使用的是命令行

2）.mySQL Client端：

管理工具、Node(客户端去mySQL服务器取数据)

Navicat-管理工具

NodeJS-程序

客户端是：nivacat

// 百度应用中心: nivacat

// 有插件可破解

```

> 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

```

> navicat创建数据库

```

具体参阅：

（4）创建数据库

右键 > 创建数据库 -----

数据库名(database): nodejs创建链接数据库时，需要用到

字符集: utf-8

排序规则：utf-8\_general\_ci （几乎可以覆盖到所有语言）

--------------------------

打开 、关闭数据库

建表

用户名

密码

主键：唯一、性能高

唯一标识符

int 整数

ID int

username varchar（小字符串） 长度32（不能太长，影响性能）

保存 》命名表

重新设计表

创建表 user\_table

ID int

username varchar（小字符串） 长度32（不能太长，影响性能）

password varchar（小字符串） 长度32（不能太长，影响性能）

整数 int

小字符串 varchar

大字符串 text // 最多可支持2G

```

> NodeJS

```

NodeJS不支持MySQL

安装模块：

$ npm install mysql

const mysql=require('mysql');

使用数据库需要两步：

1.连接到服务器

//createConnection(哪台服务器, 用户名, 密码, 库)

// 用户名，默认root

// 数据库端口，默认：port: '3306'

var db=mysql.createConnection({host: 'localhost', port: '3306', user: 'root', password: '123456', database: '20161222'});

2.查询

// db.query('干啥'， 回调);

db.query('', (err, data) => {

});

--------------------------------------------------

4.大查询语句——增删改查

增--INSERT

INSERT INTO 表 (字段列表) VALUES(值列表)

// ID一般不需要指定，0表示数据库自动填充，自增

INSERT INTO `user\_table` (`ID`, `username`, `password`) VALUES (0, 'GLEN', '234234')

删--DELETE ()

查--SELECT

SELECT 什么 FROM 表

SELECT \* FROM `user\_table` // \* 匹配所有

//查询文章列表 查询具体字段：ID,title,summery

db.query('SELECT ID,title,summery FROM article\_table', (err, data)=>{

if(err){

res.status(500).send('database error').end();

}else{

res.articles=data;

next();

}

});

改--UPDATE

查询 》新建查询 》插入mysql 》 运行

表 》 F5刷新

--------------------------------------------------

5.数据库的标准写法

a. 关键字必须大写， 如：增--INSERT / 删--DELETE 等

b. 库，表，字段必须加`(不这么加也不会报错)，如：

`user\_table`

`ID`

`username`

`password`

// 实例

SELECT \* FROM `user\_table`

-------------------------------------------------- 实例：

// server.js

https://github.com/glenlzk/node\_demo/tree/master/09%E6%95%B0%E6%8D%AE%E5%BA%93/01demo

const mysql = require('mysql');

// 1.链接数据库

var db = mysql.createConnection({

host: 'localhost',

post: '3306',

user: 'root',

password: '123456', // 字符串

database: 'pms' // 必须

});

// console.log(db);

// 2.查询数据库

db.query('SELECT \* FROM `user\_table`', (err, data) => {

if (err) {

console.log('查询出错: ' + err);

} else {

console.log(JSON.stringify(data));

};

});

```