关于腾讯云平台的建立和nodejs如何连接腾讯云mysql和mongoose数据库

1. 首先是登陆腾讯云，链接<https://passport.qcloud.com/>，登陆之后可以领取新手礼包，如图



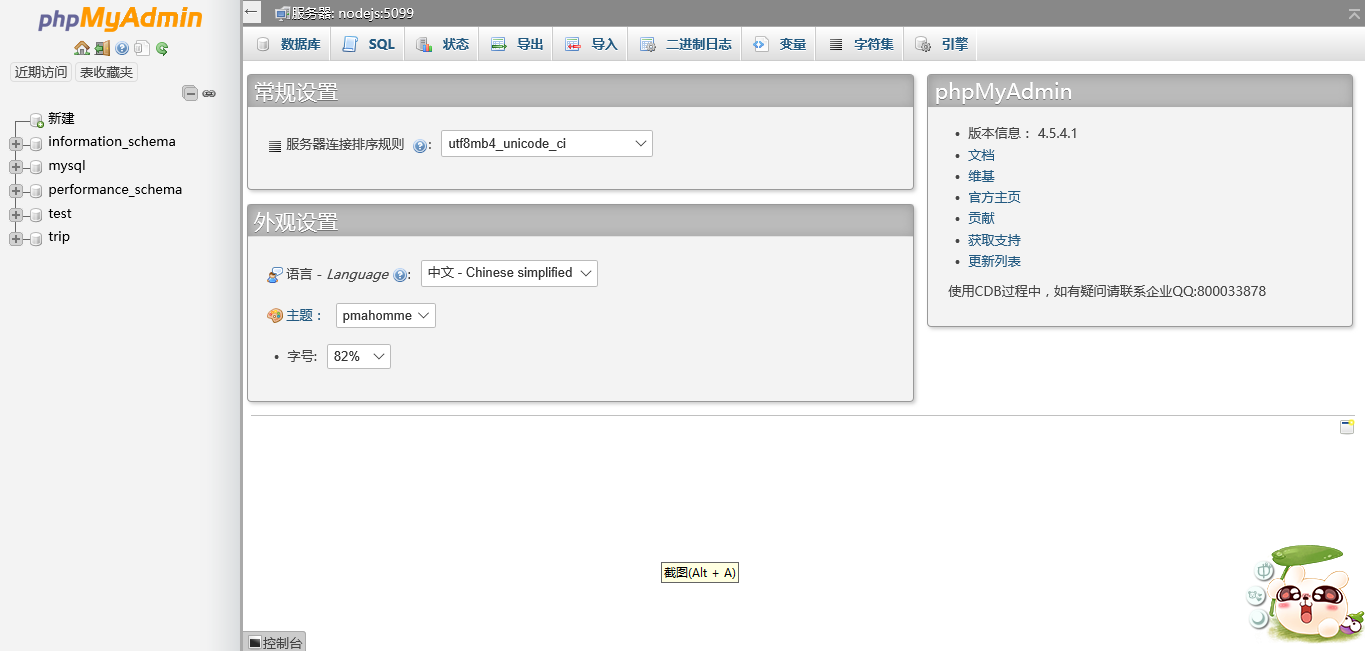
1. 进入云产品->数据库->CDB for MYSQL->立即选购，选购最低配的可用礼包抵扣使用一个月，购买完成
2. 主页->右上角产品管理->使用中的数据库,这时我们会看见一个实例，首先我们进入管理页面对实例进行一些基本的设置

实例名可以随意更改，外网地址必须开启，否则我们就不能在自己的项目中和本地访问云数据库。进入账号管理页面修改自己的root密码，然后点击右上角的登录数据库



4.登陆完成之后如下图

这个和本地客户端操作基本相同，我就不详细阐述了



二、使用一个demo来说明如何操作云mysql数据库

1、在实例上上新建一个为nodejs的数据库，然后新建一个为employee的表，新建四个字段 name sex age email除了年龄为int其它的全部为varchar格式

2、新建一个TimLiu的文件夹,打开cmd，cd TimLiu, 初始化项目npm init ,按照它的提示一步一步的操作即可，安装mysql模块，npm insitall mysql, 这个模块的作用主要是连接mysql数据库。

3、新建一个model.js 文件，

var mysql = require(‘mysql’);

var connection = mysql.createConnection({

host:'',

user:'root',

password:'123abc',

database:' nodej,

port:6445

})

connection.connect();

这里的host为我们在腾讯云上开通的外地址，端口号也是在外网地址里面，注意要把外网地址写分开，用户为默认的管理员用户，密码为我们在云平台上设置的密码，数据库为我们创建的nodejs数据库。

接下来我们对数据库进行增删改查操作

function addEmployee(){

var employeeInsertSql = 'INSERT INTO employee(name,sex,age,email) VALUES(?,?,?,?)';

var employeeInsertSql\_Params = ['Tim','男',22,'18818216454@163.com']

connection.query(employeeInsertSql,employeeInsertSql\_Params,function(err,result){

if(err) console.log('[INSERT ERR]-',err.message);

console.log(result);

})

}

addEmployee()

/\*\*

\* 更新员工

\*/

function insertEmployee(){

var employeeUpdateSql = "UPDATE employee SET name = ? WHERE age =?";

var employeeUpdateSql\_Params = ['Peter',22];

connection.query(employeeUpdateSql,employeeUpdateSql\_Params,function(err,result){

if(err) console.log('[UPDATE ERR]-',err.message);

console.log(result);

})

}

insertEmployee();

/\*\*

\* 查询员工

\*

\*/

function getEmployee(){

var employeeGetSql = "SELECT \* FROM employee";

connection.query(employeeGetSql,function(err,result){

if(err) console.log('[SELECT ERR]-',err.message);

console.log(result);

})

}

getEmployee();

/\*\*

\* 删除员工

\*/

function deleteEmployee(){

var employeeDeleteSql = "DELETE employee WHERE name = ?";

var employeeDeleteSql\_Params = 'Peter';

connection.query(employeeDeleteSql,employeeDeleteSql\_Params,function(err,result){

if(err) console.log('[DELETE ERR]-',err.message);

console.log(result);

})

}

deleteEmployee();

这样我们就可以愉快的操作数据库了

如果大家想把项目更模块化操作，可在model下面新建一个employee\_two.js,代码如下

var mysql = require('mysql');

var connection = mysql.createConnection({

host:'',

user:'root',

password:'',

database:'nodejs',

port:

})

connection.connect();

/\*\*

\* 增加员工

\* @param {String} employee

\* @param {Function} callback

\*/

exports.addEmployee=function(employee,callback){

var employeeInsertSql = 'INSERT INTO employee(name,sex,age,email) VALUES(?,?,?,?)';

var employeeInsertSql\_Params = [employee.name,employee.sex,employee.age,employee.email]

connection.query(employeeInsertSql,employeeInsertSql\_Params,callback)

}

/\*\*

\* 更新员工

\* @param {String} employee

\* @param {Function} callback

\*/

exports.updateEmployee = function(name,age){

var employeeUpdateSql = "UPDATE employee SET name = ? WHERE age =?";

var employeeUpdateSql\_Params = ['Peter',22];

connection.query(employeeUpdateSql,employeeUpdateSql\_Params,function(err,result){

if(err) console.log('[UPDATE ERR]-',err.message);

console.log(result);

})

}

/\*\*

\* 查询员工

\*

\*/

exports.getEmployee = function(){

var employeeGetSql = "SELECT \* FROM employee";

connection.query(employeeGetSql,callback)

}

/\*\*

\* 删除员工

\* @param {String} name

\*/

exports.deleteEmployee = function(name){

var employeeDeleteSql = "DELETE employee WHERE name = ?";

var employeeDeleteSql\_Params = 'Peter';

connection.query(employeeDeleteSql,employeeDeleteSql\_Params,callback)

}

再在根目录下新建index.js,代码如下

var db = require('./model/employee\_two');

//增加员工

var employee = {

name:'lisa',

age:22,

sex:"女",

email:'99533212@qq.com'

}

db.addEmployee(employee,function(err,result){

if(err) console.log("[INSERT err]-",err.message)

console.log(result);

})

//删除员工

db.deleteEmployee('Peter',function(err,result){

if(err) console.log("[DELETE err]-",err.message)

console.log(result);

})

//更新员工

db.updateEmployee('Tim',23,function(err,result){

if(err) console.log("[UPDATE err]-",err.message)

console.log(result);

})

//查询员工

db.getEmployee(function(err,result){

if(err) console.log("[GET err]-",err.message)

console.log(result);

})