# 数据库

数据库，可以理解为电子化的文件柜。用户可以对文件柜中的数据进行增、删、改、查的操作。

数据库管理系统，是为管理数据库而设计的电脑软件系统。一般会具有存储、截取、安全保证、备份等基本功能。

## 分类

|  |  |
| --- | --- |
| SQL：结构化查询语言 | NoSQL：泛指非关系型的数据库 |
| 单机：数据库运行在一台服务器上 | 分布式：数据库运行在服务器集群上 |
| 文件型：数据放在硬盘上 | 内存型：数据放在内存里 |
| 批处理：将SQL分解成MR任务 | 交互式：分级查询之后汇总 |

### SQL

MySQL、SQL Server、Postgre SQL、Oracle、Access

### No SQL

MongoDB、neo4j、elasticsearch、InfluxDB、BigTable、LevelDB

### 单机

MySQL、Postgre SQL

### 分布式

Hive、cloudera IMPALA

### 文件型

MySQL、MongoDB

### 内存型

Redis、memcache

### 批处理

Hive

### 交互式

cloudera IMPARA、HBASE

## DBMS

数据库管理系统作用：

减少记录编档的时间

减少记录检索的时间

灵活的查找序列

灵活的输出格式

多个用户同时访问记录

## SQL

SQL（Structured Query Language，结构化查询语言），是一种用于数据库中的标准数据查询语言。由IBM最早用在其开发的数据库系统中。1986年10月，称为关系型数据库管理系统的标准语言。不同数据库系统之间的SQL不能完全相互通用。

SQL可以完成数据定义、数据操作、数据控制的作用。

# MySQL

## 概述

MySQL是一个关系型数据库管理系统，由瑞典MySQL AB公司开发，目前属于Oracle公司。

MySQL是一个开放源码的小型关联式数据库管理系统，与其他的大型数据库（Oracle、DB2、SQL Server）等相比，对于一般的个人使用者和小型企业来说，MySQL提供的功能已经绰绰有余，而且由于MySQL是开放源码软件，因此可以大大降低总体拥有成本。

## 代码

from \_\_future\_\_ import print\_function

sql = ('SELECT \* from ipdata limit 10')

# mysql-connector

print('mysql-connector'.center(50, '='))

from mysql import connector

cnx = connector.Connect(host="pythontest.cgngr7chq0yp.ap-northeast-1.rds.amazonaws.com", user="jilu",

password="123456", database="pythontest", charset="utf8")

# cnx.autocommit = True

db0 = cnx.cursor()

db0.execute(sql)

for row in db0:

print(\*row) # print row[0], row[1], row[2], row[3]

# MySQLdb

print('MySQLdb'.center(50, '='))

import MySQLdb

def connect\_mysql(db\_host="pythontest.cgngr7chq0yp.ap-northeast-1.rds.amazonaws.com", user="jilu",

passwd="123456",db="pythontest", charset="utf8"):

conn = MySQLdb.connect(host=db\_host, user=user, passwd=passwd, db=db, charset=charset)

conn.autocommit(True)

return conn.cursor()

db1 = connect\_mysql()

db1.execute(sql)

for row in db1:

print(\*row)

# torndb1

print('torndb1'.center(50, '='))

import torndb

import simplejson as json

db2 = torndb.Connection(

host='pythontest.cgngr7chq0yp.ap-northeast-1.rds.amazonaws.com',

database='pythontest',

user='jilu',

password='123456',

charset="utf8")

rows = db2.query(sql)

for row in rows:

print(json.dumps(row, ensure\_ascii=False))

# # torndb2

# print('torndb3'.center(50, '='))

# row = db2.get(sql)

# print(json.dumps(row, ensure\_ascii=False))

#

# torndb3

print('torndb2'.center(50, '='))

row = db2.get('SELECT \* from ipdata limit 1')

print(json.dumps(row, ensure\_ascii=False))

## 应用

### Flask用户管理

# MySQL Python客户端

## sql-connector

官方客户端

## MySQLdb

第三方客户端

## Torndb

MySQLdb第二次封装