Database

Th.S Trần Đức Lợi
Pythonvietnam.info

Ôn tập bài cũ

- Ôn lại nội dung đã học về **Class**
- Chữa bài **ConfigLoaderClass**

Mục đích bài học

- Tìm hiểu về làm việc với Database trong python
  - Database introduction
  - SQL
  - MySQL
  - Redis

Database

- Cơ sở dữ liệu là một tập hợp có tổ chức của dữ liệu
- Để quản lý cơ sở dữ liệu ta cần DBMS (Database Management System)
- Các cơ sở dữ liệu phổ biến: Oracle, MySQL, SQLite, SQLServer, Postgres, …

Database

- 4 loại CSDL chính:
  - Hierarchical
  - Network
  - Relational
  - Object Relational

Database

- Lưu trữ thông tin về tất cả các thành viên trong công ty, trường học, bệnh viện, …
- Relational DB:
  - Employee table
  - Department table
  - Salary table

Database

- Database: chứa nhiều bảng
- Table: còn được gọi là Relation, bao gồm tuples và attributes
- Tuple (row): Một tập hợp các trường đại diện cho một phần tử
- Attribute (column, field): Một trường đại diện cho một thuộc tính của các phần tử

Database

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPT_ID	EMAIL
1001	Loi	Tran	99	loitd@pythonvietnam.info
1002	John	Hei	98	john@pythonvietnam.info

Department_ID	Dept_name	Manager_ID
99	Kế Toán	1001
98	HT	1005

Database: Key

- Primary Key
  - Định nghĩa một bản ghi duy nhất
  - Phải là giá trị duy nhất
  - Không thể chứa giá trị NULL
  - Mỗi bảng đều nên có chỉ 1 primary key
- Foreign Key
  - Trỏ tới một primary key của một bảng khác

Database: SQL

- Để query dữ liệu từ DB
- Structured query language
- Là ngôn ngữ tuân theo chuẩn ANSI thao tác trên CSDL quan hệ (relational database)

SQL

- Query dữ liệu
- Thêm mới, sửa, xóa các dòng dữ liệu trong bảng
- Tạo mới, thay thế, chỉnh sửa và drop các đối tượng
- Quản lý truy nhập

SQL

- Data Manipulation Language
  - Select
  - Insert
  - Update
  - Delete
  - Merge
- Data Definition Language
  - Create
  - Alter
  - Drop
  - Rename
  - Truncate
  - comment

SQL

- Data Control Language
  - Grant
  - Revoke
- Transaction Control
  - Commit
  - Rollback
  - Savepoint

MySQL

- Open-source database
- Community Edition
- Standard Edition
- Enterprise Edition
- MySQL Cluster Carrier Grade Edition

MySQL

- Cài đặt:
  - http://www.wampserver.com/en/
  - https://www.apachefriends.org/index.html
  - https://www.mamp.info/en/

SQL: Select

- Lấy dữ liệu từ DB
- Column alias
- Distinct keyword
- Select * | distinct | column | expression alias from Table

SQL: where

- Giới hạn số lượng dữ liệu bằng điều kiện
- SELECT *column,column*
  FROM *table*
  WHERE *column = value*;

SQL: Update

- Cập nhật dữ liệu cho DB
- UPDATE *table*
  SET *column1=value1,column2=value2,...*
  WHERE *column=value*;

SQL: Insert

- Thêm dữ liệu vào trong bảng
- INSERT INTO *table*
  VALUES (*value1*,*value2*,*value3*,...);
- INSERT
  INTO *table* (*column1*,*column2*,*column3*,...)
  VALUES (*value1*,*value2*,*value3*,...);

SQL: Delete

- Xóa dữ liệu trong database
- DELETE FROM *table*
  WHERE *column*=*value*;

Python & MySQL

- Cài đặt thư viện MySQLdb để làm việc với MySQL
- Windows:
  - http://sourceforge.net/projects/mysql-python/
- Linux:
  - sudo apt-get install python-mysqldb

MySQLdb: connect()

- Mở kết nối tới DB:
  - connect(host="localhost",user="root",passwd="", db="cdcol")
- Lấy con trỏ:
  - cur = con.cursor()

MySQLdb: Close()

- Đóng kết nối:
  - cur.close()
  - con.close()
- Commit và rollback dữ liệu:
  - con.commit()
  - con.rollback()

MySQLdb: Bài tập

- Hãy sử dụng configloader để load thông số DB và thực hiện kết nối tới DB MySQL

MySQLdb: Select

- Thực thi câu select:
  - cur.execute(sql)
- Fetch dữ liệu
  - row1 = cur.fetchone()
  - rows = cur.fetchall()

MySQLdb: Insert

- Thực thi câu lệnh Insert
- try:
- cur.execute(sql3, data2insert)
- con.commit()
- except Exception, e:
- con.rollback()
- print e

MySQLdb: Update

```
try:
    cur.execute(sql4)
    con.commit()
except Exception, e:
    con.rollback()
    print e
```

MySQLdb: Bài tập

- Lấy dữ liệu từ file config, kết nối vào DB, in toàn bộ dữ liệu của DB, cho người dùng chọn lựa sửa, xóa 1 row bất kỳ
- (Quản lý học sinh)

Redis

- Cài đặt redis trên windows
  - Redis.io
  - https://github.com/rgl/redis/downloads
  - Apt-get install redis-server
- Pip install redis
- Import redis
- Hiểu 4 loại lưu trữ hay gặp trên redis
  - Strings
  - Lists
  - Sets
  - hashs

Redis

- Redis is an open source, BSD licensed, advanced **key-value cache** and **store**. It is often referred to as a **data structure server** since keys can contain strings, hashes, lists, sets,sorted sets, bitmaps and hyperloglogs.
    - *Nguồn redis.io*

Redis: key

- Có thể sử dụng bất kỳ binary sequence nào làm key
- Kích thước lớn nhất cho phép là 512MB
- Không nên đặt quá dài (1 key dài 1MB, …)
  - Tốn bộ nhớ
  - Tra cứu khó khăn
  - Truyền tải giá trị qua môi trường mạng -> tốn băng thông
  - Không cần thiết

Redis: Key TTL

- Cú pháp:
  - ***Expire key time***
  - Hoặc ***set key 'abc' ex time***
- Đơn vị tính seconds
- Lệnh ***ttl key*** để xem thời gian còn sống của 1 key

Redis: Key

- Xóa một key bằng lệnh ***del key***
- Xem kiểu dữ liệu của một key bằng lệnh: ***type key***

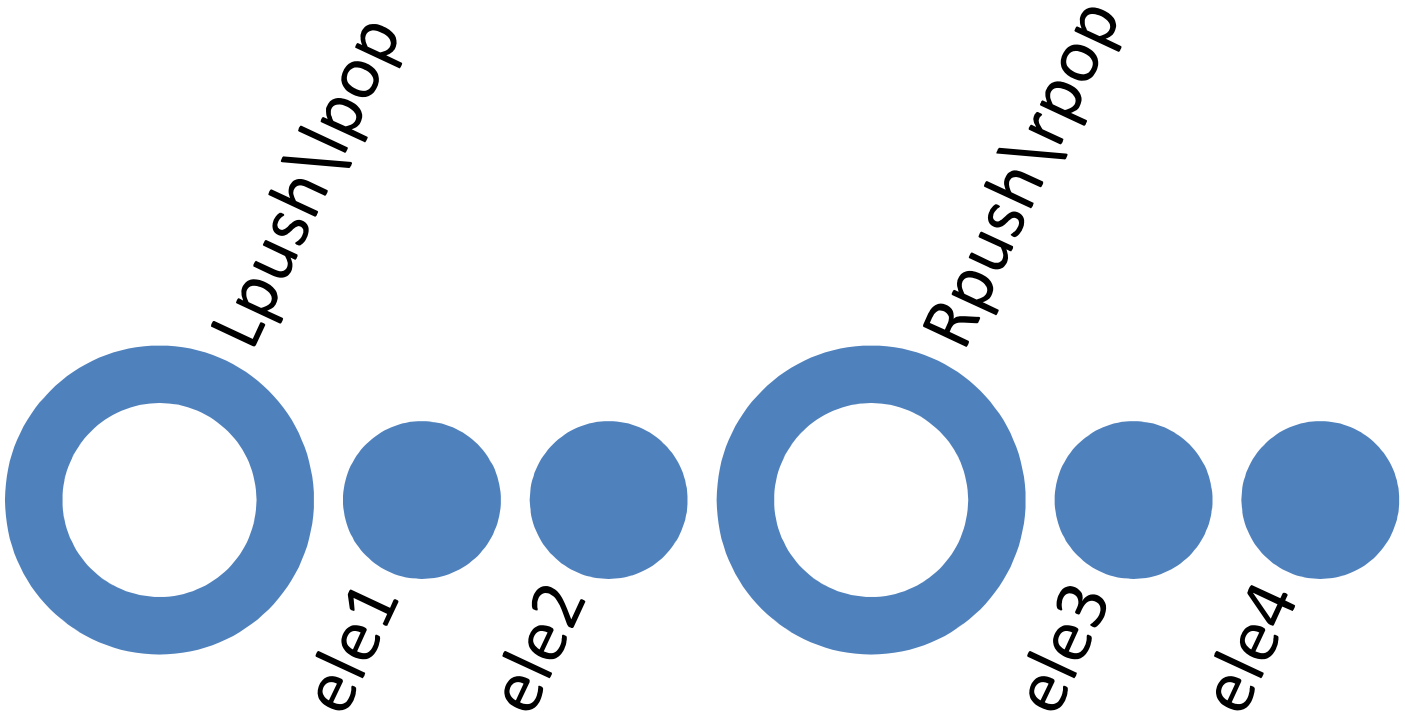
Redis: String types

- Là kiểu dữ liệu cơ bản nhất
- Giống với dữ liệu của Memcached
- Giá trị không lớn hơn 512MB
- Giá trị có thể là chuỗi, chuỗi binary, …
- Câu lệnh làm việc cơ bản: SET, GET, INCR, DECR, MSET, MGET, …

Redis: List

- List trong redis là 1 dãy các phần tử sắp xếp nối nhau: 1,2,3,4,5,…
- Linked list # Array: thời gian thêm mới phần tử không phụ thuộc vào số lượng phần tử
- Ưu điểm: tốc độ thêm mới phần tử
- Hạn chế của list: tốc độ truy cập phần tử (ngược lại với sorted sets)

Redis: list



Redis: List

- Ứng dụng
  - Lấy danh sách các công việc/ updates/ feeds mới nhất của 1 user
  - Giao tiếp giữa 2 processes theo mô hình consumer-producer

Redis: connect

- r = redis.StrictRedis(host='localhost', port=6379, db=0)
- Mỗi một instance của redis sẽ tự tạo một connection pool để quản lý kết nối tới redis

Redis: Connect

- ConnectionPool:
- pool = redis.ConnectionPool(host='localhost', port=6379, db=0)
- s= redis.Redis(**connection_pool**=pool)
- -> sử dụng pool chung

Redis: Key-value

- #single key
- r.set('f', 'you')
- print r.get('f')
- Sử dụng lệnh get/set tương ứng với redis client

Redis: list

- #list
- r.lpush('lst1', 'San Loeo')
- r.lpush('lst1', 'San Loeo')
- print r.lrange('lst1', 0, -1)
- r.rpop('lst1')
- print r.lrange('lst1', 0, -1)
- Sử dụng các lệnh lpush, rpop để làm việc với list

Redis: Bài tập

- Dựng module kết nối với redis và thực hiện các thao tác với list và key-value dưới dạng OOP

Redis: Pub/Sub

- Mô hình Publish/Subscribe
- Đối tượng PubSub trong redis-py
- Khai báo một pubsub mới
  - Ps = s.pubsub()
  - Ps.subscribe('pyvn-chnl', 'chnl-2')
  - Ps.get_message()

Redis: pub/sub

- Publish một message
  - S.publish(‘chnl’, ‘I am Loi’)
- Unsubcribe kênh:
  - Ps.unsubscribe(‘chnl’)
  - Ps.unsubscribe()

Redis: Pub/Sub

- Ps.close()
- Sau khi làm việc xong với một đối tượng pubsub

Redis: Pub/Sub

- Xây dựng lại bài tập redis với pub/sub kết nối vào redis sử dụng connection pool

Tổng kết bài học

- SQL
- MySQL
- Redis